This is the accepted manuscript of the article, which has been published in International Journal of Data Science. 2017,2(4), 273-300. http://doi.org/10.1504/IJDS.2017.10009003

A comparison of classification methods in automated taxa identification of benthic macroinvertebrates

Henry Joutsijoki

School of Information Sciences,
University of Tampere,
Tampere, Finland
Fax: +358-3-2191001 E-mail: henry.joutsijoki@uta.fi

Martti Juhola

School of Information Sciences,
University of Tampere,
Tampere, Finland
Fax: +358-3-2191001 E-mail: martti.juhola@uta.fi

Abstract: In this research we examined the automated taxa identification of benthic macroinvertebrates. Benthic macroinvertebrates play an important role in biomonitoring. They can be used in water quality assessments. Identification of benthic macroinvertebrates is made usually by highly trained experts, but this approach has high costs and, hence, the automation of this identification process could reduce the costs and would make wider biomonitoring possible. The automated taxa identification of benthic macroinvertebrates returns to image classification. We applied altogether 11 different classification methods to the image dataset of eight taxonomic groups of benthic macroinvertebrates. Wide experimental tests were performed. The best results, around 94% accuracies, were achieved when Quadratic Discriminant Analysis, Radial Basis Function network and Multi-Layer Perceptron were used. On the basis of the results, it can be said that the automated taxa identification of benthic macroinvertebrates is possible with high accuracy.

Keywords: Benthic macroinvertebrates; Classification; Machine learning; Water quality

Reference

Biographical notes: Henry Joutsijoki received his M.Sc. and Phil.Lic. degrees in mathematics from the University of Tampere in 2008 and 2010. In 2012 he received Ph.D. degree in computer science and is a member of Data Analysis Research Group at the School of Information Sciences. His research interests include machine learning, support vector machines and data mining.

Martti Juhola received his M.Sc. and Ph.D. degrees in computer science from the University of Turku, Finland, in the 1980s, where he was an academic assistant, lecturer, and researcher, later becoming a professor at the University of Kuopio, Finland. Since 1997, he is a professor at the University of Tampere. His research

interests include medical informatics, signal analysis, pattern recognition, and information retrieval.

1 Introduction

Freshwater areas are in a minority position when considered all aquatic environments in the Globe. Hence, it is important to keep the current freshwater areas in good condition. Water quality monitoring has gained more and more interest when the environmental issues have come into the centre in all levels of the society. One way to monitor the water quality is to use benthic macroinvertebrates. Benthic macroinvertebrates are excellent indicators of the freshwater ecosystems such as rivers (Riverlife, 2012) and they are suitable for environmental research as articles Ambelu et al. (2010); Dominguez-Granda et al. (2011); Hoang et al. (2010); Song et al. (2006), for instance, show. Benthic macroinvertebrates contain a great variety of organisms which can be seen in their sensitivity towards water quality. In this research the examined image material consists of only images from EPT (*Ephemeroptera* (mayflies), *Plecoptera* (stoneflies) and *Trichoptera* (caddiflies)) orders which are known to be sensitive for water quality changes.

A common way to investigate the water quality is to take water samples and perform a chemical analysis but this approach only gives a short-term point of view about the condition of a freshwater ecosystem (Tirronen et al., 2009). Benthic macroinvertebrates instead can give not only a view of current situation, but also a broader perspective about the changes in water quality. The life cycle of benthic macroinvertebrates is usually between one to two years (Tirronen et al., 2009) which supports the use of benthic macroinvertebrates in water quality assessments. Benthic macroinvertebrates have several advantages, why they should be used in biomonitoring. Firstly, benthic macroinvertebrates appear in all aquatic habitats and we know plenty about the consequences of environmental effects to them (Riverlife, 2012). Secondly, benthic macroinvertebrates are relatively immobile, so they express well localized environmental conditions (Riverlife, 2012). Thirdly, benthic macroinvertebrates are easy to collect and, thus, they are suitable for experimental purposes (Riverlife, 2012).

Benthic macroinvertebrates are diversed organisms, since there are thousands of different species of them. A common way to interpret the condition of a freshwater ecosystem is to present a general measure called taxa richness instead of giving a complete list of species encountered (Riverlife, 2012). In practice from time to time benthic macroinvertebrates occur which cannot be identified to a species level which makes impossible to give an accurate list of species from a specific freshwater ecosystem. Taxa richness and water quality are connected to each other. If the taxa richness suddenly decreases, it may indicate that the water quality has gotten worse and something unnatural might have occurred. On the other hand, if the taxa richness or the number of benthic macroinvertebrates in several species has increased, it can point out that the water quality has improved.

Species or more generally taxa identification is a well-defined and specific problem. Attempts to automate the identification process have encountered problems in practice such as “It is too costly” or “It is too different” based on Gaston and O’Neill (2004). Nevertheless, we have tackled the identification problem in the case of automated taxa identification of benthic macroinvertebrates in our earlier research by applying machine learning method called Support Vector Machines (SVMs). Support Vector Machines (Cortes

and Vapnik, 1995) have become a very popular classification method. SVM was developed for binary classification, but soon the interest moved to expand SVM to also concern multi-class cases. Different multi-class extensions were quickly developed and from these commonly used methods are one-vs-one (OVO) (Hsu and Lin, 2002), one-vs-all (OVA) (Lorena et al., 2008; Rifkin and Klautau, 2004) and Directed Acyclic Graph Support Vector Machines (DAGSVM) (Lorena et al., 2008; Jian et al., 2008; Platt et al., 2000). In Joutsijoki and Juhola (2013); Joutsijoki (2013a) OVO strategy was applied to benthic macroinvertebrate classification and the problem of tie situations in OVO was examined. The OVO strategy was used also in Tirronen et al. (2009); Kiranyaz et al. (2011) for automated taxa identification of benthic macroinvertebrates. Moreover, in Joutsijoki and Juhola (2011a) OVO and OVA strategies were used in benthic macroinvertebrate classification and ties were closely concerned. DAGSVM was applied to the same application with a great success in Joutsijoki and Juhola (2011b). Lastly, in Joutsijoki (2012, 2013b, 2014) a bit rarely used variant of multi-class SVMs, the half-against-half strategy (Lei and Govindaraju, 2005), was used for the benthic macroinvertebrate classification. All these articles showed that the automated taxa identification of benthic macroinvertebrates is possible to made with a high accuracy. Furthermore, SVM proved to be a very good choice for the automated taxa identification of benthic macroinvertebrates. Since the previous researches on this dataset have focused on mainly the use of SVM, there is a lack of an extensive research where existing baseline classification methods are examined.

Generally speaking, the automated taxa identification of benthic macroinvertebrates (Tirronen et al., 2009; Kiranyaz et al., 2011, 2010a,b; Larios et al., 2008; Lytle et al., 2010; Ärje et al., 2010) is a relatively new application compared to applications such as handwritten digit recognition (Bottou et al., 1994; Liu et al., 2003), text classification (Joachims, 2001; Mitra et al., 2011) or ECG classification (Bortolan et al., 1991; Mar et al., 2011) for instance. The research around automated taxa identification of benthic macroinvertebrates has many advantages. Usually, the identification of benthic macroinvertebrate specimens is made by highly trained taxonomists. Due to human-made identification costs of identification are high and the identification is a laborious process. Hence, the automation of the identification process would cut costs greatly. Often the identification of benthic macroinvertebrates can be routine work for human experts (Joutsijoki et al., 2014). Thus, the automation of this process could relieve the workload of taxonomists to solve some other more specialized problems (Joutsijoki et al., 2014). An automated identification process would also enable biologists to collect a larger numbers of samples, which is recommended when benthic macroinvertebrates are used in biomonitoring. However, the need of human-made identification cannot be totally removed since human expertise is required when constructing a training set for the use of classification methods (Joutsijoki et al., 2014).

Identifying benthic macroinvertebrates from images is a demanding task from the pattern recognition point of view since differences between species or even genera can be small. There are still some taxonomic groups which are difficult to define even for taxonomists (Riverlife, 2012), so it makes the problem even harder. Furthermore, the intra-class variability of the benthic macroinvertebrates can be high and the positions and sizes of benthic macroinvertebrates may vary in each image. The classification of benthic macroinvertebrates need to be reliable because, if samples are classified wrong, this can give a wrong view of the current situation of an aquatic environment.

In this research the goal is to compare different existing classification methods in the automated taxa identification of benthic macroinvertebrates. Feature extraction, feature selection and other preprocessing stages of images are left outside of this research.

Altogether 11 different classification methods are used. These are: k-Nearest-Neighbour method (with four different measures) (Cover, 1967), Linear Discriminant Analysis (Xie and Qiu, 2007), Quadratic Discriminant Analysis (Yu and Ekström, 2003), Minimum Mahalanobis Distance Classifier (Zhang and Zhou, 2003), Classification Tree (Bittencourt et al., 2003), Multinomial Logistic Regression (Agresti, 1990; Barros et al., 2012), Naïve Bayes (Huang et al., 2003), K-Means (Jain, 2010), Self-Organizing Map (Chen et al., 2010; Kohonen, 1995; Saarikoski et al., 2009, 2011), Multi-Layer Perceptron (Haykin, 1999; Venkatesh et al., 2003) and Radial Basis Function network (Haykin, 1999; Picton, 2000). Experiments with Learning Vector Quantization (Kohonen, 1995, 1998) were so poor that it was left out from this research.

This research has a following structure. In Section 2 the theory of used classification methods are presented briefly. Section 3 explains the design of experiments, data description and the experimental results and their analysis. Section 4 concludes the research.

2 Method

2.1 k-Nearest-Neighbour

The k-Nearest-Neighbour (k-NN) method (Duda et al., 2001) is one of the most used classification methods according to Ougiaroglou et al. (2007) and Wu et al. (2008). In k-NN the classes of k nearest examples are investigated by using some distance measure. The class label of a new example is defined by the max-win principle. That is, the class having the most examples within the k-nearest training examples with respect to a new example assigns the final class label. To decrease the opportunity of a tie, it is a common habit to use only the odd values of k. There are no any exact rules for choosing the best k value. Thus, the usual approach is to try different values and to choose the value which gives the best performance. Another important aspect in k-NN method is the choice of distance measure. There are numerous alternatives to choose, likewise Euclidean metric

$$D(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2},$$

Cityblock metric

$$D(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m}|x_i - y_i|,$$

or more general L_∞ metric for example. Furthermore, Cosine measure

$$D(\mathbf{x},\mathbf{y}) = 1 - \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

and Correlation measure

$$D(\mathbf{x},\mathbf{y}) = 1 - \frac{(\mathbf{x}-\overline{\mathbf{x}})\cdot(\mathbf{y}-\overline{\mathbf{y}})}{\|\mathbf{x}-\overline{\mathbf{x}}\|\|\mathbf{y}-\overline{\mathbf{y}}\|},$$

where $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ are the mean vectors are commonly used distance measures in k-NN classification. In the presentation of the distance measures we assumed that $\mathbf{x},\mathbf{y} \in \mathbb{R}^m$ and the norm in the cosine and correlation measures is Euclidean.

2.2 Quadratic Discriminant Analysis

Let us have a set of classes $\{c_1, c_2, \ldots, c_N\}$ and let $P(c_i)$ denote a priori probability of the ith class, $i = 1, 2, \ldots, N$ (Cios et al., 2007; Ärje et al., 2010). Bayes theorem gives us

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{P(\mathbf{x})} = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{\sum_{i=1}^{N} P(\mathbf{x} \mid c_i)P(c_i)},\ i = 1, 2, \ldots, N \tag{1}$$

where $P(\mathbf{x})$ is the unconditional probability density function for an example $\mathbf{x} \in \mathbb{R}^m$ and $P(\mathbf{x} \mid c_i)$ is the class conditional probability density function for class c_i (Cios et al., 2007). Bayes' classification rule assigns the example using winner-takes-all principle to $P(c_i \mid \mathbf{x})$, $i = 1, 2, \ldots, N$ (Cios et al., 2007).

Based on Cios et al. (2007) Bayes classification rule can be presented by means of discriminant functions

$$d_i(\mathbf{x}) = \ln P(\mathbf{x} \mid c_i) + \ln P(c_i),\ i = 1, 2, \ldots, N \tag{2}$$

where example $\mathbf{x}$ follows a multivariate normal Gaussian distribution within each class and constant $P(\mathbf{x})$ has been neglected. Now the probability density function in the class c_i is

$$P(\mathbf{x} \mid c_i) = (2\pi)^{-\frac{m}{2}} |\mathbf{\Sigma}_i|^{-\frac{1}{2}} \exp[-\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)], \quad i = 1, 2, \ldots, N, \tag{3}$$

where $\boldsymbol{\mu}_i$ is the mean vector of the ith class feature vector and $|\mathbf{\Sigma}_i|$ is the determinant of the ith class covariance matrix (Cios et al., 2007). By substituting Eq. (3) into Eq. (2) and after eliminating the constant term $\frac{m}{2} \ln 2\pi$ we obtain

$$d_i(\mathbf{x}) = -\frac{1}{2} \ln|\mathbf{\Sigma}_i| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) + \ln P(c_i), \tag{4}$$

when $i = 1, 2, \ldots, N$ (Cios et al., 2007). Bayes classifier can now be called Quadratic Discriminant Analysis (QDA) on the basis of Eq. (4). QDA assumes that $\mathbf{\Sigma}_i \neq \mathbf{\Sigma}_j$ when $i \neq j$ (Cios et al., 2007; Ärje et al., 2010). In the discriminant form a new sample will be assigned to the class having the greatest discriminant value.

2.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a very important special case from Quadratic Discriminant Analysis (Cios et al., 2007). In LDA we assume that $\mathbf{\Sigma}_i = \mathbf{\Sigma}$, $i = 1, 2, \ldots, N$ (Cios et al., 2007). Then the discriminant function from Eq. (4) can be expressed as follows:

$$d_i(\mathbf{x}) = -\frac{1}{2} \ln|\mathbf{\Sigma}| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) + \ln P(c_i), \quad i = 1, 2, \ldots, N \text{ (Cios et al., 2007).} \tag{5}$$

Dropping out the class independent terms and multiplying the vector-matrix-vector-product open from Eq. (5) we obtain

$$d_i(\mathbf{x}) = \boldsymbol{\mu}_i^T \mathbf{\Sigma}^{-1} \mathbf{x} - \tfrac{1}{2} \boldsymbol{\mu}_i^T \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_i + \ln P(c_i), \tag{6}$$

where $i = 1, 2, \ldots, N$ (Cios et al., 2007). Equation (6) states now a linear discriminant function of $\mathbf{x}$. The class i, which has the greatest discriminant value $d_i(\mathbf{x})$, will be chosen as a final class for $\mathbf{x}$ (Cios et al., 2007).

2.4 Minimum Mahalanobis Distance Classifier

Assume that we have equal covariance matrix determinants for all classes c_i, $i = 1,2,\ldots,N$ and for all classes $P(c_i) = P$ (Bohling, 2006; Cios et al., 2007). Since $\ln P$ and $-\frac{1}{2}\ln|\mathbf{\Sigma}_i|$ are constants they can be left out from Eq. (4) and we can neglect the constant $\frac{1}{2}$ (Bohling, 2006; Cios et al., 2007). Hence, the discriminant function is

$$d_i(\mathbf{x}) = -(\mathbf{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i), i = 1,2,\ldots,N$$

according to Bohling (2006); Cios et al. (2007). Now $d_i(\mathbf{x})$ defines the squared Mahalonobis distance of $\mathbf{x}$. The classifier selects the class c_i for which $\mathbf{x}$ is the closest (when dealing with the Mahalanobis distance) to the mean vector $\boldsymbol{\mu}_i$ (Cios et al., 2007). In other words we seek based on Bohling (2006); Cios et al. (2007)

$$\arg\min_i (\mathbf{x}-\boldsymbol{\mu}_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i),\ i = 1,2,\ldots,N.$$

2.5 Classification Tree

Classification tree (CT) is a general classification method which can be used with numeric and categorical variables and one of the most well-known algorithm is CART (classification and regression trees) (Duda et al., 2001) according to Wu et al. (2008). A decision tree consists of nodes where in each one of them a split is made (Duda et al., 2001). Splits can be made as a binary decision or multiway decisions but in CART only binary splits is used (Duda et al., 2001). Construction of a decision tree begins at the root node where the whole training set is split and in other nodes splitting can be performed recursively and the number of split is not unambiguously determined (Duda et al., 2001).

The goal is to keep decision tree as simple as possible when constructing it. To achieve this objective we need to find features which divide the data well and one measure called impurity can help in this subject (Duda et al., 2001). We define $i(A)$ to denote the impurity of a node A and we want that $i(A) = 0$ for all examples that reach the node having the same class label (Duda et al., 2001). Moreover, we want $i(A)$ to be large, if the classes are equally represented (Duda et al., 2001). Very often entropy impurity is used as a measure $i(A) = -\sum_j P(c_j)\log_2 P(c_j)$, where $P(c_j)$ is the fraction of samples at node A that are in class c_j, $j = 1,2,\ldots,N$ (Duda et al., 2001). Another and a more general way to define impurity is to use Gini's impurity $i(A) = \sum_{i\neq j} P(c_i)P(c_j) = \frac{1}{2}[1-\sum_j P^2(c_j)]$ that CART algorithm also uses (Duda et al., 2001).

An important question that arises in the case of classification trees is when to stop splitting. Cross-validation is a way to avoid overfitting of data which can be encountered if CT is constructed so that every leaf node corresponds to the lowest impurity (Duda et al., 2001). Another way is to set a threshold value for impurity or to use statistical significance testing as a stopping criterion (Duda et al., 2001). Pruning is a relevant topic when CTs are considered (Duda et al., 2001). In pruning nodes linked to a common antecedent node are considered for elimination based on (Duda et al., 2001). Any pair whose elimination decreases the impurity is eliminated and, hence, the common antecedent node becomes a leaf (Duda et al., 2001). By this means we can simplify the structure of CT and to increase the generalization ability. After pruning a tree, it is common that it can be unbalanced (Duda et al., 2001).

Assigning class labels for the leaves is easy in CT. In the case of complete CT each leaf corresponds to samples in a single class (Duda et al., 2001). When pruning, for instance, is used and the leaves have positive impurity, each one of them should be labeled using majority voting (Duda et al., 2001). For missing attribute values we can evaluate the impurity at a node A using only present attribute information (Duda et al., 2001). There are also other methods to handle cases with missing attribute values and more information about this subject can be found from Duda et al. (2001).

2.6 Multinomial Logistic Regression

Multinomial Logististic Regression (MNLR) belongs to the group of multinomial logit models and is a generalization of the logistic regression where a response can have only two values (Agresti, 1990). MNLR can have both categorical and ordinal responses and the explanatory variables can be continuous or discrete (Agresti, 1990). Let $\pi_j(\mathbf{x}_i)$ denote the probability of response j, $j = 1,2,\ldots,N$, at the ith setting of values of m explanatory variables $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{im})'$ (Agresti, 1990). Now the generalized logit model in terms of the response probabilities is

$$\pi_j(\mathbf{x}_i) = \frac{\exp(\boldsymbol{\beta}_j' \mathbf{x}_i)}{\sum_{h=1}^{N} \exp(\boldsymbol{\beta}_h' \mathbf{x}_i)} \tag{7}$$

where $\boldsymbol{\beta}$'s are vectors for the regression coefficients (Agresti, 1990). When $\boldsymbol{\beta}_N = \mathbf{0}$, Eq. (7) obtains the form $\pi_N(\mathbf{x}_i) = (\sum_{h=1}^{N} \exp(\boldsymbol{\beta}_h' \mathbf{x}_i))^{-1}$(Agresti, 1990). Moreover, we obtain

$$\log\left[\frac{\pi_j(\mathbf{x}_i)}{\pi_N(\mathbf{x}_i)}\right] = \boldsymbol{\beta}_j' \mathbf{x}_i,\ j = 1,2,\ldots,N-1$$

(Agresti, 1990). Hence, we need $N-1$ logit equations in order to define response variable with $N-1$ classes.

When seeking the maximum likelihood estimates (parameter values which maximizes this function), we need to maximize the independent multinomial likelihood with respect to constraint in Eq. (7). By taking logarithm from the multinomial likelihood function we obtain the log likelihood function $L = \sum_{i=1}^{N} \#_i \log \pi_i$, where $\#_i$ is the number of responses in class i (Agresti, 1990). We can find the estimate for log likelihood function by using, for instance, Newton-Raphson method (Agresti, 1990). More information and details about MNLR can be found from (Agresti, 1990).

2.7 Naïve Bayes

Assume that we have a set of classes $C = \{c_1, c_2, \ldots, c_N\}$ and an example $\mathbf{x} \in \mathbb{R}^m$. The goal is to find class c_i, $i = 1,2,\ldots,N$, which has the highest posterior probability for $\mathbf{x}$. Naïve Bayes can be derived from the Eq. (1), i.e., Bayes theorem. Because $P(c_i \mid \mathbf{x})$ is unknown, it must be estimated from the data (Lewis, 1998). Bayes' theorem recommends to estimate probabilities $P(\mathbf{x} \mid c_i)$, $P(c_i)$ and $P(\mathbf{x})$ in order to evaluate $P(c_i \mid \mathbf{x})$ (Lewis, 1998). However, estimation of $P(\mathbf{x} \mid c_i)$ consists of a problem, since $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ can include an arbitrary number of different values (Lewis, 1998). Hence, a following decomposition is assumed

$$P(\mathbf{x} \mid c_i) = \prod_{j=1}^{m} P(x_j \mid c_i)$$

where the feature value x_j is statistically independent of any other $x_{j'}$ when the example $\mathbf{x}$ is from the class c_i (Lewis, 1998). Thus, we obtain according to Bayes' rule the form

$$P(c_i \mid \mathbf{x}) = \frac{P(c_i)\prod_{j=1}^{m} P(x_j \mid c_i)}{P(\mathbf{x})}. \tag{8}$$

In classification problems we choose the class c_i for the new sample such that $P(c_i \mid \mathbf{x})$ is the highest. Equation (8) now defines the Naïve Bayes classifier and the denominator can dropped out since it is class independent (Lewis, 1998). Thus, we obtain

$$P(c_i \mid \mathbf{x}) = P(c_i)\prod_{j=1}^{m} P(x_j \mid c_i).$$

2.8 K-Means

K-Means algorithm (Cios et al., 2007) is one of the first clustering methods and it has a very simple basic idea. Assume that we have examples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ and $\mathbf{x}_l \in \mathbb{R}^m$, $l = 1, 2, \ldots, n$ and we are interested in dividing them into c clusters. Before we can represent *K*-Means algorithm, we need to define some concepts. Firstly, we need the performance index

$$Q = \sum_{i=1}^{c}\sum_{l=1}^{n} u_{il}\|\mathbf{x}_l - \mathbf{v}_i\|^2 \tag{9}$$

where the squared norm is the Euclidean distance between $\mathbf{x}_l$ and cluster centroid $\mathbf{v}_i$ (Cios et al., 2007). Secondly, in the Eq. (9) $U = [u_{il}]$ is the partition matrix, which assigns the examples to the clusters and has the property of

$$u_{il} = \begin{cases} 1, \text{ if } \mathbf{x}_l \text{ belongs to cluster } i, \\ 0, \text{ otherwise} \end{cases}$$

(Cios et al., 2007). Matrix U satisfies the following two conditions:

$$0 < \sum_{l=1}^{n} u_{il} < n,\ i = 1, 2, \ldots, c \quad \text{and} \quad \sum_{i=1}^{c} u_{il} = 1,\ l = 1, 2, \ldots, n$$

based on Cios et al. (2007). Our task is to minimize Q and to construct partition matrix U and a set of cluster centroids.

According to Cios et al. (2007) *K*-Means algorithm can be represented as follows:

1. Choose randomly one centroid for each cluster. Hence, we have a set of centroids $\mathbf{v}_i$, $i = 1, 2, \ldots, c$.
2. Iterate.
   2.1. Construct a partition matrix U such that
   $$u_{il} = \begin{cases} 1, \text{ if } d(\mathbf{x}_l, \mathbf{v}_i) = \min_{i \neq j} d(\mathbf{x}_l, \mathbf{v}_j) \\ 0, \text{ otherwise.} \end{cases}$$
   2.2. Update the centroids by evaluating weighted average
   $$\mathbf{v}_i = \frac{\sum_{l=1}^{n} u_{il}\mathbf{x}_l}{\sum_{l=1}^{n} u_{il}}$$
   until Q does not change anymore, or until the changes are within acceptable limit.

2.9 Self-Organizing Map

Self-Organizing Map (SOM) (Haykin, 1999; Saarikoski et al., 2009, 2011) is a widely used clustering method in which the basic idea is to transform an input vector into a one- or two-dimensional lattice and to make the transform adaptively in a topologically ordered fashion (Haykin, 1999). Assume that the input space is m-dimensional. Let $\mathbf{x} = (x_1, \ldots, x_m)^T$ be an input vector and $\mathbf{w}_j = (w_{j1}, \ldots, w_{jm})$ be the synaptic weight vector of neuron j (Haykin, 1999). After the initialization of a network, there are three processes called competition, cooperation and synaptic adaptation which are included into SOM algorithm.

Overall, according to Haykin (1999) there are four important properties in SOM:

1. A continuous input space of activation samples.
2. Network topology is normally in the form of one- or two-dimensional lattice of neurons defining a discrete output space.
3. A time-varying neighborhood function $h_{j,i(\mathbf{x})}(t)$ defined around a winning neuron.
4. A learning-rate parameter $\eta(t)$ which has the initial value of η_0 and decreases when $t \to \infty$, but never reaches zero.

For the neighborhood function we define

$$h_{j,i(\mathbf{x})}(t) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(t)}\right), t = 0, 1, 2, \ldots,$$

where $d_{j,i}^2 = \|\mathbf{r}_j - \mathbf{r}_i\|^2$ is the squared Euclidean distance between the position of activated neuron j and the discrete position of the winning neuron i (Haykin, 1999). Moreover, we have

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right)$$

where σ_0 is the initial value of the SOM algorithm and τ_1 is a time constant Haykin (1999). For the learning parameter $\eta(t)$ we have

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_2}\right), t = 0, 1, 2, \ldots,$$

where τ_2 is another time constant in SOM algorithm (Haykin, 1999).

SOM algorithm can be summarized with five steps according to Haykin (1999):

1. Choose randomly initial values for weight vectors $\mathbf{w}_j(0)$ such that $\mathbf{w}_j(0)$ is different for each $j = 1, 2, \ldots, l$ where l is the number of neurons in the lattice. Weight vectors can also be chosen randomly from the set of input vectors.
2. Take some input vector $\mathbf{x} \in \mathbb{R}^m$ from the input space. This vector $\mathbf{x}$ represents the activation sample, which is applied in the lattice.
3. Seek the winning neuron $i(\mathbf{x})$ at the time step t by using criterion

$$i(\mathbf{x}) = \arg\min_j \|\mathbf{x}(t) - \mathbf{w}_j\|, j = 1, 2, \ldots, l.$$

4. Update the weight vectors of all neurons by the formula

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t) h_{j,i(\mathbf{x})}(t)(\mathbf{x}(t) - \mathbf{w}_j(t))$$

where $\eta(t)$ is a learning-rate parameter and $h_{j,i(\mathbf{x})}$ is the neighborhood function centred around the winning neuron $i(\mathbf{x})$. Both of these parameters are dynamically changed during learning.

5. Repeat the steps 2-4 until the stopping criterion has been reached.

2.10 Levenberg-Marquardt Backpropagation

Levenberg-Marquardt algorithm and the use of it together with backpropagation is one of the most used training algorithms with Multi-Layer Perceptron (MLP) networks according to Mohd et al. (2013); Ranganathan (2004). Generally speaking, MLP is feed-forward network which has an input layer, one or more hidden layers and an output layer (Haykin, 1999). An input layer is only a passive layer where no computations are made (Kiranyaz et al., 2011). Hidden layers contain neurons which include nonlinear smooth activation function such as sigmoid or hyperbolic tangent (Haykin, 1999). The use of a nonlinear activation function in MLP is important since linear activation function would reduce MLP to a single-layer-perceptron (Kiranyaz et al., 2011). All elements in an input layer are connected to the first hidden layer neurons with the corresponding weights. Moreover, hidden layer neurons are connected with all neurons in the next hidden layer or with the neurons in the output layer Joutsijoki et al. (2014). Examples on MLP network can be found from Haykin (1999).

Levenberg-Marquardt (LM) backpropagation originates from the optimization theory given by Levenberg (Levenberg, 1944) and Marquardt (Marquardt, 1963) independently from each other. LM is an approximation for Newton's method (Hagan and Menhaj, 1994) and the starting point for LM is that we have a sum of squares function $V(\mathbf{x})$ to be minimized with respect to $\mathbf{x}$ which is the vector of network's parameters (Hagan and Menhaj, 1994). Newton's method gives an update formula $\Delta\mathbf{x} = -[\nabla^2 V(\mathbf{x})]^{-1}\nabla V(\mathbf{x})$ where $\nabla^2 V(\mathbf{x})$ is Hessian matrix and $\nabla V(\mathbf{x})$ is gradient (Hagan and Menhaj, 1994).

However, Hessian matrix and gradient can be approximated as follows: $\nabla^2 V(\mathbf{x}) = J^T(\mathbf{x})J(\mathbf{x})$ and $\nabla V(\mathbf{x}) = J^T(\mathbf{x})e(\mathbf{x})$ where $e(\mathbf{x})$ is a vector of errors and $J(\mathbf{x})$ is Jacobian matrix (Hagan and Menhaj, 1994). LM backpropagation applies an update formula $\Delta\mathbf{x} = [J^T(\mathbf{x})J(\mathbf{x}) + \mu\mathbf{I}]^{-1}J^T(\mathbf{x})e(\mathbf{x})$ where μ is a damping factor which is adjusted after every iteration (Hagan and Menhaj, 1994). According to Hagan and Menhaj (1994) LM algorithm can be reduced to Gauss-Newton or gradient descent algorithm depending on the choice of μ. Moreover, Jacobian matrix can be evaluated using gradient descent backpropagation (Hagan and Menhaj, 1994).

2.11 Radial Basis Function Network

Radial Basis Function network (RBFN) was introduced by Broomhead and Loewe (1988). Compared to MLP, RBFN has a slightly different structure. RBFN has an input layer, one hidden layer and a linear output layer (Haykin, 1999). Now the neurons in the hidden layer apply a nonlinear transformation to input signals. A difference to MLP is also that RBFN has linear weights only between the hidden layer and the output layer (Haykin, 1999). Every

Figure 1 An example image from each taxonomical group in the dataset. The order of taxonomic groups from top left to bottom right is BAE, DIU, HEP, PEL, SIL, ISO, RHY and TAE.

neuron in the hidden layer contains a nonlinear activation function. Activation function for the ith RBF unit is

$$y_i = \varphi\left(\frac{\|\mathbf{x}-\mu_i\|}{\sigma_i^2}\right)$$

where φ is the radial basis function, μ_i is the center of radial basis function and σ_i is the width of the peak around the center μ_i (Haykin, 1999; Kiranyaz et al., 2010b; Picton, 2000). Often Gaussian basis function is used as an activation function. More information on RBFN can be found for instance from Haykin (1999); Picton (2000).

3 Experimental Results

3.1 Test arrangements and the data description

Our dataset (1350 images) contains images from eight different taxonomic groups of benthic macroinvertebrates. These are: *Baetis rhodani, Diura nanseni, Heptagenia sulphurea, Hydropsyche pellucidulla, Hydropsyche siltalai, Isoperla* sp., *Rhyacophila nubila* and *Taeniopteryx nebulosa*. Seven of these taxonomic groups were identified to a species level and one, *Isoperla* sp., was recognized to a genus level. We will refer to the groups in tables and in the following text with the abbreviations BAE, DIU, HEP, PEL, SIL, ISO, RHY and TAE. The corresponding group sizes were 116, 129, 172, 102, 271, 311, 83 and 166. In Figure 1 there is an example image from each taxonomical group included to the dataset.

In the testing phase we used 10 times 10-fold cross-validation to the dataset. Hence, we obtained 100 training and test sets. Cross-validation distributions were selected so that every training set had as equal number of training examples from every group as possible. The same cross-validation distributions were used with all classification methods. In the case of RBF network and Multi-Layer Perceptron we divided every training set into a smaller training set and validation set. The best configuration and parameter values were selected according to the mean accuracy of the validation sets. Levenberg-Marquardt algorithm was used as a training algorithm for MLPs. When the best configuration was found, RBF

network and Multi-Layer Perceptron were trained again with the full training set (union of smaller training set and validation set). Finally, RBFN and MLP were tested with the test sets obtained by cross-validation and a mean of the results was evaluated as a final result.

In RBF network we varied the value of σ (width of the Gaussian basis function) from $0.5, 1.0, 1.5, \ldots, 20.0$ and the best value of σ was determined according to the mean accuracy of the validation sets. Thus, the step between values of σ was 0.5 and we tested altogether 40 different values of σ. In addition, the number of neurons was set to 100. Further, mean squared error goal was 0.0. Moreover, MLP was tested with a single hidden layer and with two hidden layer configurations. More specifically, we tested MLP with configurations $15 \times i \times 8$ where $i = 1, 2, \ldots, 15$ and $15 \times i \times j \times 8$ where $i, j = 1, 2, \ldots, 15$. Altogether, MLP was tested with 240 different configurations. In MLP we set the maximum number of epochs to 150, performance goal was 0.0 and the target value for gradient was 1.00e-05.

We tested k-NN method with four different distance functions: Euclidean and cityblock metrics and cosine and correlation measures presented in Subsection 2.1. The k-NN method was tested with the odd integers k from 1 to 51 and if a tie situation occurred, it was solved such that the nearest sample from training set subject to test example assigned the final class label. In the case of NB normal distribution was used when modeling the data and in estimating prior probabilities relative frequencies of the classes from the training set was used. Moreover, LDA, QDA and MMDC are parameter free methods and for MNLR type of model to fit was nominal.

In the case of CT, where the pruning of trees was made automatically such that the splitting criterion was 10 or more observations in impure node and the minimal number of observations per tree leaf was 1. Furthermore, Gini's diversity index was used as a criterion for choosing a split and weights for all observations were 1 and surrogated splits at each branch node was not used. We performed the classification with SOM altogether for different 43 lattices. The number of neurons varied from 8 to 50 in a lattice. Hexagonal topology was used and initial neighborhood size was 3. In addition, 200 iterations was used.

K-Means was tested with the cluster numbers ranging from 8 to 100 and squared Euclidean distance was used as a distance measure and 100 replicates (number of times to repeat the clustering) was used and empty cluster was treated as an error. Because SOM is an unsupervised method, we had to define the class tag for each neuron in a lattice. This was made according to the majority principle where the class tags were determined based on the number of class members in the neuron. A class having the most samples in a neuron determined the class tag. Class tags were determined with a similar way in K-Means and, furthermore, if a tie situation happened when using the majority principle, the closest sample to the centroid in the cluster determined the final class tag for the cluster.

The dataset had altogether 25 features where 15 of them were selected to this paper. These features were divided into a geometrical and statistical features (can also be described as intensity-based features). These features were selected due to the excellent results in previous researches (see Joutsijoki (2012); Joutsijoki and Juhola (2013); Kiranyaz et al. (2010a, 2011, 2010b); Ärje et al. (2010)). Features are grayscale features. Geometrical features included {Area, Perimeter, Width, Height, Feret's Diameter, Major, Minor, Circularity} and statistical features included {Mean, Standard Deviation, Mode, Median, Integrated Density, Kurtosis, Skewness}. According to Joutsijoki et al. (2014) where the same dataset was examined the features used can be defined as follows:

1. Area is the size of the mask in square pixels.

2. Perimeter is the length of the outside boundary of the mask.

3. Width and Height are the width and height of the smallest rectangle enclosing the object in mask.
4. Feret's Diameter is the longest distance between any two point inside the sample.
5. Major and Minor are the length of primary and secondary axis of the best fitting ellipse. Furthermore, in mask ellipse has the 0th, 1st and 2nd image moments.
6. Circularity is $4\pi \times \frac{Area}{Perimeter^2}$.
7. Mean is the average gray value within the object.
8. Standard deviation is the variation of average gray value.
9. Mode is the most frequent gray value occurred in the object.
10. Median is the centermost value of the pixels in the object when the pixel values from the selection are ordered to a vector in increasing order.
11. Integrated Density is the sum of the gray values of a selection.
12. Kurtosis and Skewness are the fourth and the third standardized moment of the intensity values of a selection.

Other 10 features which are not used in this paper and they are included to the dataset are: Min, Max, XM, YM, X, Y, BX, BY, Angle, Area Fraction. More accurate information about these features and the ImageJ program can be found from ImageJ (2014). Before presenting the data to the classifiers, the columns of dataset were standardized to have zero mean and unit variance. We did not make any other transformations such as scaling the features into intervals $[-1,1]$ or $[0,1]$, because we wanted to keep the classification process as natural as possible. Every transformation moves the data farther from the input space and makes the analysis of results harder to understand with respect to original data. About the preprocessing stage of the data, i.e., how the features were extracted from the images and how the scanning of the benthic macroinvertebrate were made can be found from Joutsijoki et al. (2014); Ärje et al. (2010). All the tests were made with Matlab 2010b together with Statistics Toolbox, Neural Network Toolbox and Bioinformatics Toolbox of Matlab. Tests were performed using laptop having Pentium i7-2630QM, 2.0GHz processor and Win7 operating system with 16GB of memory.

3.2 Results

In the result tables the boldfaced numbers in the diagonal are the classification rates (also known as sensitivity or true positive rate). Moreover, the rows of the results tables indicate the true classes and the columns indicate predicted classes. Because the group sizes vary, the contents of tables are not symmetric. In the case of k-NN we present the classwise classification rates with all k values used and we do not present the complete mean confusion matrices. Accuracies were determined by evaluating the trace of a confusion matrix (not changed into percentages) divided by the sum of the elements in a confusion matrix. Equations and definitions of accuracy and classification rate (sensitivity) can be found from Cios et al. (2007). Moreover, in Table 11 we present the standard deviations of classification rates with different classification methods used. In the case of k-NN standard deviations are presented in a graphical form in Figures 2 and 3. Other metrics such as F1 score, area under

ROC curve or Matthews correlation coefficient were not used in this research because we are interested in only classification rates and accuracies and other measures do not bring any essential new information from the application point of view. Measures used are adequate for the research because accuracy can be used to compare classification methods with the human-made taxa identification accuracy and classification rate (sensitivity) explains which taxonomical groups are difficult to identify.

Firstly, we consider the results of k-NN. Figures 2 and 3 shows interesting results. The x axis presents the specific k value and the y axis is the corresponding classification rate with the k value. Class BAE was identified with all measures very well and the classification rates were similar with all measures. Generally, the level of classification in BAE was around 90% with all measures. A small increase to the classification rates came with correlation and cosine measures when $k > 20$. The second class, class DIU, was classified nearly perfectly with all measures. The same tendency continued although the k value was increased. Class HEP had a bit different kind of curves with the classification rates. Now for the first time, we obtained clear differences between the measures. Euclidean and cityblock measures were the best ones in the case of class HEP. Both of them obtained the best classification rates with small k values, likewise $k = 1,3,5$. The best classification rate was obtained when $k = 1$. Classification rates decreased when the k value became larger and this occurred with all measures. The order of the measures was that Euclidean and cityblock measures were the best ones. The third was cosine measure and the poorest results were achieved with correlation measure. The interval, in which the results spread, was quite wide since the best classification rate was above 90% and the lowest classification rate was around 50%.

For class PEL, classification rates formed interesting curves. All measures achieved similar results with all k values. When $k = 1$, classification rates were as their highest being nearly 100% and after that the classification rates decreased almost linearly until for $k > 11$ the classification rates stabilized with all measures to a level of around 80%. Class SIL again obtained very good results and the level of classification remained steady being within the interval of 90%-100%. The change of a distance did not bring any crucial differences to the results. Compared to the classes earlier analyzed, SIL managed likewise BAE and DIU. In class SIL correlation and cosine measures were slightly worse than Euclidean and cityblock measures but the differences were minimal. The next taxonomical group was *Isoperla* sp. (identified only to a genus level). Class ISO managed from the classification relatively well except with the correlation measure which obtained about 20% lower classification rates with every k value compared to other measures used. An interesting detail is that cosine measure achieved the highest classification rates when $k > 7$. Otherwise, Euclidean and cityblock measures were equally good and the results were above 90%.

Class RHY obtained very different results compared to the previous taxonomical groups. Now the diversity of the results was wider than before. A noticeable detail is that class RHY is the smallest taxonomical group in the data. With small k values Euclidean metric and cityblock and cosine measures obtained similar results, whereas for larger k value the results with cosine measure dropped dramatically. Classification rates with Euclidean and cityblock measures remained similar despite k value, but the general trend was downwards, when k value was increased. In the beginning the results with correlation measure were the poorest, but when $k > 35$ the roles between correlation and cosine measures changed. Then the classification rates with correlation measure were higher than with cosine. The drop in the classification rates when using cosine was significant. The highest results were achieved when $k = 1$, and it was then above 90%. On the contrary the lowest classification rate was obtained when $k = 51$, and it was then below 40%. The last class to analyze in

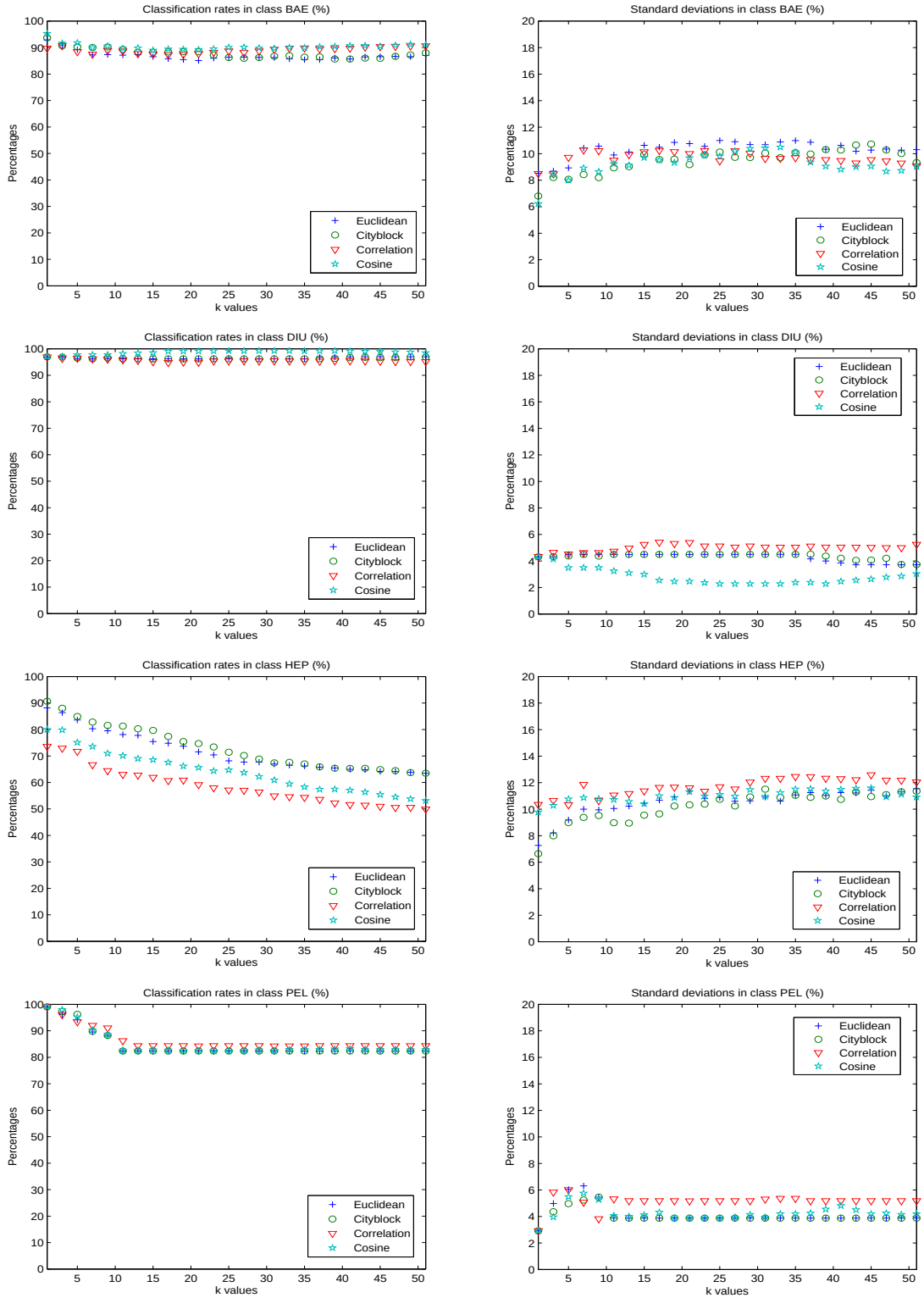

Figure 2 Classification rates and standard deviations (%) when k-NN used with different k values and measures in classes BAE, DIU, HEP and PEL.

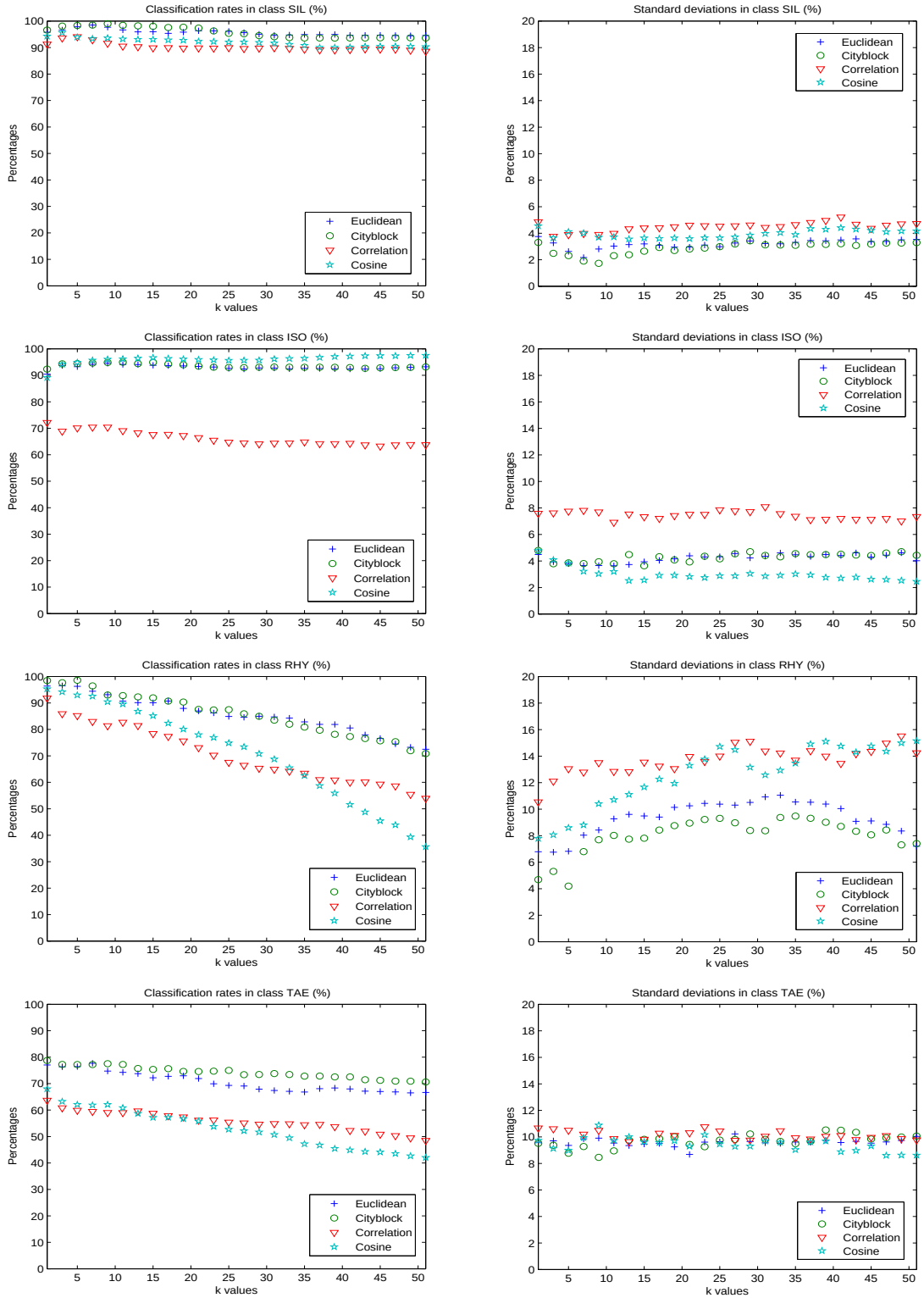

Figure 3 Classification rates and standard deviations (%) when k-NN used with different k values and measures in classes SIL, ISO, RHY and TAE.

Table 1 Results (%) when Linear Discriminant Analysis used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**94.2**	0.0	0.0	0.0	5.8	0.0	0.0	0.0
DIU	0.0	**92.7**	6.9	0.0	0.4	0.0	0.0	0.0
HEP	1.2	5.7	**78.5**	0.0	14.6	0.0	0.0	0.0
PEL	0.0	0.0	0.0	**82.4**	0.0	2.9	14.7	0.0
SIL	3.0	0.0	3.9	0.0	**93.1**	0.0	0.0	0.0
ISO	0.0	0.0	0.0	0.0	0.3	**90.2**	0.0	9.5
RHY	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0
TAE	0.0	0.0	0.0	0.0	0.0	7.7	0.0	**92.3**

k-NN results was TAE. TAE was the only class which did not exceed 90% classification rate with any k value or distance measure. The results of TAE were dichotomous. Euclidean and cityblock metrics remained within 70%-80% interval when the results with correlation and cosine were located in the interval of 50%-70%. Classification rates of these measures did not alter despite the increment to the k values. When considering all classes together, we noticed that the smaller k values were better than the larger k values. Especially when $k = 1$ almost every class obtained the highest classification rate.

Results, when Linear Discriminant Analysis (LDA) was applied to the benthic macroinvertebrate classification, can be seen from Table 1. LDA proved to be a good choice as the results showed. Altogether six classes from eight possible were classified above 90% classification rate. The smallest class RHY was classified perfectly and this can mean that the class RHY could be a totally separate cluster in the input space compared to the other taxonomical groups. From the first and the third row of Table 1 it can be seen that the majority of the misclassified samples of the classes BAE and HEP were located to the class SIL. Moreover, nearly 15% of the misclassified samples of class PEL were identified as class RHY samples. Classes HEP and PEL were the hardest classes to classify and these two classes were the only ones which remained below 90% classification rates. In the case of class ISO nearly all misclassified samples were classified as class TAE samples and nearly 8% of TAE samples were identified incorrectly to class ISO. Overall, LDA classified well the benthic macroinvertebrate samples and the highest number of classes where the misclassified samples were spread was three and this happened in the case of class HEP which had also the lowest classification rate.

In Table 2 there are the results given by Minimum Mahalanobis Distance Classifier (MMDC). MMDC achieved very good classification rates in seven classes. Class SIL was the only class having below 90% classification rate. It obtained around 83% classification rate and the misclassified points of class SIL spread among classes BAE, DIU and HEP. From these classes BAE and HEP were the same as in the results of LDA. Classes DIU, HEP and SIL were identified with nearly perfect score. There was a significant improvement, over 16%, in classes HEP and PEL compared to the corresponding results in Table 1. Moreover, in class RHY the classification rate decreased nearly 10% from LDA results being now a bit over 90%. Moreover, all misclassified points were located in class PEL. In classes ISO and TAE all misclassified samples were identified to the same classes as in LDA results and these classes were classified better than in Table 1.

Next we have the results of Quadratic Discriminant Analysis (QDA) in Table 3. An interesting detail is that the class TAE had identical results than in Table 2. Furthermore, in

Table 2 Results (%) when Minimum Mahalanobis Distance Classifier used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**94.5**	0.0	1.0	0.0	4.5	0.0	0.0	0.0
DIU	0.0	**97.3**	2.7	0.0	0.0	0.0	0.0	0.0
HEP	0.0	0.4	**99.0**	0.0	0.6	0.0	0.0	0.0
PEL	0.0	0.0	0.0	**99.0**	0.0	0.0	1.0	0.0
SIL	2.3	2.0	13.1	0.0	**82.6**	0.0	0.0	0.0
ISO	0.0	0.0	0.2	0.0	0.0	**92.3**	0.0	7.5
RHY	0.0	0.0	0.0	9.6	0.0	0.0	**90.4**	0.0
TAE	0.0	0.0	0.0	0.0	0.0	4.6	0.0	**95.4**

Table 3 Results (%) when Quadratic Discriminant Analysis used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**96.6**	0.0	0.8	0.0	2.6	0.0	0.0	0.0
DIU	0.0	**97.1**	2.5	0.0	0.4	0.0	0.0	0.0
HEP	0.0	0.2	**94.5**	0.0	5.3	0.0	0.0	0.0
PEL	0.0	0.0	0.0	**98.6**	0.0	0.0	1.4	0.0
SIL	6.5	1.6	3.4	0.0	**88.5**	0.0	0.0	0.0
ISO	0.0	0.0	0.0	0.0	0.0	**92.0**	0.0	8.0
RHY	0.0	0.0	0.0	2.7	0.0	0.0	**97.3**	0.0
TAE	0.0	0.0	0.0	0.0	0.0	4.6	0.0	**95.4**

the rest of the classes, except class DIU, the misclassified samples were classified identically into the same classes as in Table 2. This might stem from the reason that QDA and MMDC are related to each other in theoretical sense. QDA obtained, generally speaking, better results than LDA or MMDC. More closely considered QDA obtained above 95% classification rates in six classes. The only exceptions were classes SIL and ISO. When compared to MMDC results, improvements were achieved in classes RHY and SIL and the greatest decrease in classification rates was in class HEP which was classified into MMDC results with nearly perfect score.

Table 4 shows the results when Classification Tree method (more specifically CART algorithm) was applied. Compared to the previous tables we can notice immediately a phenomenon that there was much more diversity in the results than before. Firstly, a majority of the classes obtained below 90% classification rates. Secondly, the misclassified samples were spread into more classes than in Tables 1-3. We still got some similarities with the previous tables. Firstly, the majority of the misclassified samples in class HEP were classified into class SIL as in Tables 1 and 3. Secondly, the classes of misclassified samples in classes ISO and TAE were the same as in Tables 1 and 3. Thirdly, nearly 8% of the class SIL samples were identified as class HEP members and this confusion was also in the results of MMDC. When considering the diagonal elements of Table 4 it can be noticed that class TAE was identified below 80% classification rate and this result was the lowest classification rate hitherto. Classes DIU and PEL were the only ones which rose above 90% classification rate. Also, classes BAE, SIL and ISO obtained nearly identical classification rates. Although the

Table 4 Results (%) when Classification Tree used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**85.4**	0.0	1.6	1.0	5.9	3.4	0.0	2.7
DIU	0.0	**95.8**	1.3	0.0	2.9	0.0	0.0	0.0
HEP	0.3	0.8	**80.2**	0.3	11.8	1.2	2.9	2.5
PEL	2.0	0.0	1.9	**91.6**	0.8	2.2	1.4	0.1
SIL	1.7	0.8	7.6	0.6	**85.3**	1.7	0.0	2.3
ISO	1.4	0.0	1.2	0.5	2.4	**85.8**	0.6	7.9
RHY	0.0	0.0	5.3	3.6	0.4	3.0	**87.4**	0.3
TAE	1.4	0.0	3.0	0.0	3.3	14.8	0.2	**77.3**

Table 5 Results (%) when Naïve Bayes used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**93.1**	0.0	0.0	0.0	1.7	2.6	0.0	2.6
DIU	0.0	**97.0**	3.0	0.0	0.0	0.0	0.0	0.0
HEP	1.7	4.0	**66.1**	0.0	26.1	0.9	0.0	1.2
PEL	3.0	0.0	0.0	**82.4**	8.8	0.0	5.8	0.0
SIL	13.1	1.2	6.1	0.4	**76.6**	1.4	0.0	1.2
ISO	8.2	0.0	0.6	0.0	1.3	**74.0**	0.0	15.9
RHY	2.9	2.1	0.1	7.0	8.5	5.1	**62.1**	12.2
TAE	2.4	0.0	3.0	0.0	0.0	15.6	0.0	**79.0**

general level of the results decreased from the previous ones, the results were still reasonably good.

Next we had the results of the Naïve Bayes (NB) method. At the first sight we can notice that only two classes, classes BAE and DIU, had above 90% classification rate, which can be thought as a limit for very good result. Especially, class DIU with 97% identification was a high-class result. Moreover, only class PEL together with the aforementioned ones reached above 80% classification rate. The rest of the classes remained below 80%. Classes HEP and RHY were classified around 62% and 66% classification rates. These results were the lowest ones. Classes SIL and ISO were identified with very close results to each other. The analysis of the misclassified samples is again an important thing. From Table 5 it can be seen the same phenomenon as from Tables 1, 3 and 4: a great number of the misclassified samples in class HEP were classified as class SIL members. Furthermore, a majority of the misclassified samples in class ISO were located to class TAE and the same in vice versa. Overall, NB did not contrive very well from the classification compared to the previous classification methods.

Multinomial Logistic regression (MNLR) is never before used in the benthic macroinvertebrate classification. Results from Table 6 showed that MNLR was a relatively good choice for this classification problem. There were three classes, HEP, RHY and TAE, having below 90% classification rate. Now the best class was SIL recognized with nearly 96% classification rate and, also, DIU was identified very well since the classification rate achieved nearly 95%. Table 6 indicated that misclassified samples from classes BAE, DIU, HEP and PEL spread into exactly the same classes as in Table 1 where LDA was used.

Table 6 Results (%) when Multinomial Logistic Regression used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**92.2**	0.0	0.0	0.0	7.8	0.0	0.0	0.0
DIU	0.0	**94.7**	3.8	0.0	1.5	0.0	0.0	0.0
HEP	1.5	1.2	**83.6**	0.0	12.7	0.0	0.0	1.0
PEL	0.0	0.0	0.0	**92.6**	0.0	3.2	4.2	0.0
SIL	0.8	0.2	3.1	0.0	**95.8**	0.0	0.0	0.1
ISO	0.0	0.0	0.0	0.6	0.0	**92.8**	0.0	6.6
RHY	0.0	0.0	0.0	4.4	0.0	5.7	**89.9**	0.0
TAE	2.2	0.0	0.0	0.5	0.0	15.2	0.4	**81.7**

Furthermore, the tendency that the majority of the wrong classified samples from class ISO were identified as TAE members and vice versa, happened again. This phenomenon can also be seen from the result tables in article Joutsijoki and Juhola (2013) where SVM together with one-vs-one method was applied to the benthic macroinvertebrate classification.

The first clustering method applied to the benthic macroinvertebrate classification was K-Means and the corresponding results can be seen in Table 7. This table was achieved by using 100 clusters. Results with K-Means were promising but they did not manage to win LDA, QDA or MMDC results. The results were comparable with the obtained Classification Tree results. Now there were three classes (DIU, PEL and SIL) which gained classification rates over 90%. Otherwise, the classification rates remained to 80%-90% except with class HEP, which achieved below 80% classification rate and class TAE having below 70% classification rate. The same phenomenon occured with the misclassified examples of classes BAE, HEP, ISO and TAE as in many previous result tables. Compared to the results in Table 4, the diagonal entries of classes BAE, DIU, ISO and RHY were quite close to each other. Moreover, in both methods class TAE was identified with the lowest classification rate. A noticeable detail was that class RHY was classified quite well, although it was the smallest class in the dataset.

Another clustering method used in the benthic macroinvertebrate classification was SOM and the corresponding results can be seen from Table 8. The results showed similar behavior as in Table 7. Now classes DIU and SIL obtained over 90% classification rates. Classes ISO and TAE were classified in the same way. When considering the misclassified samples, we noticed the similar phenomena in the classes BAE, HEP, ISO and TAE as before. Moreover, we obtained similarity between K-Means and SOM when examined more closely misclassified samples in class RHY. The majority of these samples were located to class ISO. So, we obtained generally interesting patterns, how some of the classes interfere with each other despite the classification method. Overall, SOM achieved a bit worse results than K-Means. There were two classes below 80% classification rate and one class yielded below 70% classification rate. The class with the lowest result was the same as in Tables 4, 6 and 7.

The last two result tables considered artificial neural networks and the first one of them was Multi-Layer Perceptron. Compared to the previous result tables we obtained a significant improvement to the results. The results of QDA in contrast to Table 9 are similar since in both cases the results are very good. Class TAE was the only class having below 90% classification rate and it was 89.6% result. In a misclassified sample analysis there did not happen any dramatic changes. Classes DIU, PEL, SIL and ISO were identified above

Table 7 Results (%) when K-means with 100 clusters used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**84.6**	0.0	0.3	0.0	14.1	0.0	0.0	1.0
DIU	0.0	**94.7**	3.4	0.0	1.9	0.0	0.0	0.0
HEP	1.6	1.1	**77.9**	0.0	18.7	0.6	0.0	0.1
PEL	0.1	0.0	0.3	**90.8**	2.5	3.1	3.1	0.1
SIL	2.8	0.1	5.3	0.2	**91.1**	0.5	0.0	0.0
ISO	0.0	0.0	0.0	0.3	1.4	**83.8**	1.9	12.6
RHY	0.0	0.0	1.0	2.8	0.0	10.3	**85.5**	0.4
TAE	0.7	0.0	0.0	0.4	0.2	29.6	0.3	**68.8**

Table 8 Results (%) when Self-Organizing Map with 50 clusters used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**80.1**	0.0	0.2	0.0	18.7	0.0	0.0	1.0
DIU	0.0	**92.5**	6.2	0.0	1.3	0.0	0.0	0.0
HEP	1.2	1.7	**72.1**	0.0	23.4	1.0	0.0	0.6
PEL	1.3	0.0	0.2	**83.4**	3.0	4.9	7.2	0.0
SIL	1.8	0.2	5.4	0.0	**92.2**	0.3	0.0	0.1
ISO	0.0	0.0	0.0	0.0	1.8	**84.9**	1.3	12.0
RHY	0.0	0.0	1.0	1.7	0.0	16.2	**79.8**	1.3
TAE	0.7	0.0	0.0	0.0	0.1	30.7	0.2	**68.3**

95% classification rates which is always a noticeable detail. Compared to Table 3, the results contained some individual differences. Firstly, the first four classes were classified better than with QDA, but classes SIL and ISO were, on the contrary, classified better with MLP. Especially, in the case of class SIL the difference was significant being nearly 8%. The last two classes were again recognized better with QDA.

The last classification method was RBF networks which achieved results at quite the same level as MLP did. Class TAE was the worst class to identify as in Tables 4, 6, 7 and 8. Misclassified samples of the classes BAE, HEP, ISO and TAE were located in the similar manner as before. Compared to Table 9 individual differences appeared. The greatest improvement happened in class RHY where RBF network classified it nearly 4% better than MLP. The other, but smaller, improvements occurred in classes BAE, HEP and SIL. Classes DIU and PEL were recognized quite evenly with both ANN methods. Class TAE was identified worse with RBF network than MLP. Overall, the results of RBF network were similar to the QDA results.

From Figure 4 we can see the accuracies and standard deviations of the k-NN method with different k values and measures. Cityblock and Euclidean measures were very close to each other with all k values, but cityblock was little better than Euclidean measure. The best accuracy was obtained with the cityblock measure together with $k = 1$ being a bit over 90%. Cosine measure was below Euclidean measure all the time but it still achieved relatively good results. The poorest results were obtained with the correlation measure, which had over 80% accuracy as its best. A common fact for measures was that the increase in k value decreased the accuracy. Overall, it can be said that $k = 1$ is the best k value for this dataset.

Table 9 Results (%) when Multi-Layer Perceptron with configuration $15 \times 15 \times 7 \times 8$ used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**93.4**	0.0	0.8	0.0	5.2	0.3	0.1	0.2
DIU	0.2	**95.9**	1.7	0.0	1.4	0.1	0.4	0.3
HEP	1.3	0.5	**92.6**	0.0	4.9	0.2	0.1	0.4
PEL	0.1	0.0	0.4	**95.8**	0.1	0.4	3.1	0.1
SIL	1.3	0.5	2.1	0.0	**96.0**	0.0	0.0	0.1
ISO	0.1	0.0	0.1	0.0	0.1	**95.3**	0.2	4.2
RHY	0.0	0.0	0.3	6.0	0.0	1.2	**92.2**	0.3
TAE	0.0	0.0	0.4	0.0	0.1	9.8	0.1	**89.6**

Table 10 Results (%) when RBF network with $\sigma = 3.0$ used. The rows of the results tables indicate the true classes and the columns indicate predicted classes.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
BAE	**90.7**	0.0	0.0	0.0	9.3	0.0	0.0	0.0
DIU	0.0	**96.0**	2.5	0.0	1.5	0.0	0.0	0.0
HEP	1.7	0.0	**90.7**	0.0	7.0	0.0	0.0	0.6
PEL	0.0	0.0	0.0	**96.3**	1.0	1.5	1.2	0.0
SIL	0.6	0.3	1.0	0.0	**98.0**	0.1	0.0	0.0
ISO	0.0	0.0	0.0	0.3	0.0	**94.0**	0.0	5.7
RHY	0.1	0.0	0.0	2.5	0.0	1.4	**96.0**	0.0
TAE	0.0	0.0	0.0	0.0	0.0	12.7	0.0	**87.3**

Table 11 Standard deviations of classification rates (%) with different classification methods.

	BAE	DIU	HEP	PEL	SIL	ISO	RHY	TAE
LDA	7.1	6.5	9.2	3.9	4.2	5.4	0.0	5.7
MMDC	7.0	3.8	2.3	2.9	7.0	4.6	11.1	4.4
QDA	5.2	3.8	4.8	3.4	4.9	4.8	5.3	4.3
CT	9.9	4.6	10.0	6.7	7.7	6.0	11.5	9.0
NB	7.7	3.7	9.0	3.9	6.1	6.9	10.2	7.4
MNLR	8.1	5.3	9.5	6.5	3.5	4.0	10.0	8.0
K-Means	9.6	6.1	10.8	7.1	5.3	6.9	10.5	12.4
SOM	11.3	7.4	12.0	4.7	4.8	7.6	11.9	11.8
MLP	6.9	6.0	6.1	6.2	3.8	3.1	13.7	7.5
RBFN	9.0	4.7	7.3	4.8	2.8	4.3	6.9	6.7

Table 12 Obtained mean accuracies (%) and their standard deviations with different classification methods.

Method	Accuracy	Method	Accuracy
LDA	90.1 ± 2.1	MMDC	92.6 ± 2.0
QDA	93.7 ± 1.8	CT	85.4 ± 2.7
NB	77.8 ± 2.5	MNLR	90.8 ± 2.3
K-means (100 clusters)	84.4 ± 3.1	SOM	82.6 ± 3.0
RBFN	93.7 ± 1.9	MLP	94.1 ± 2.0

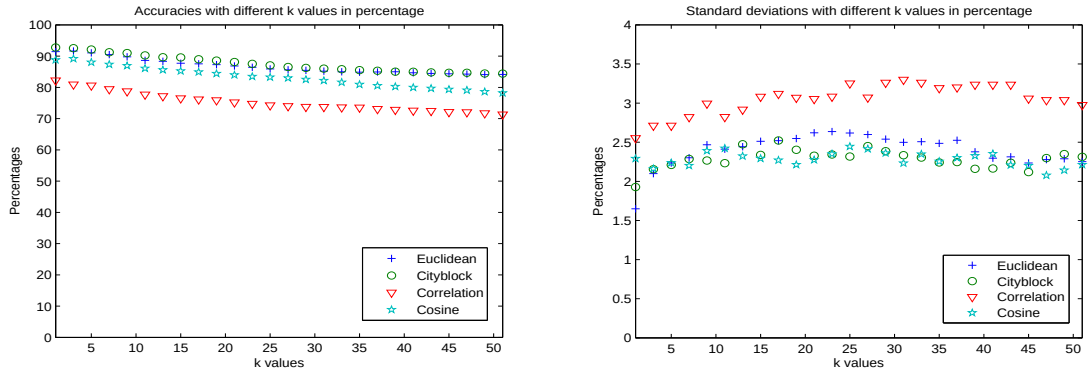

Figure 4 Accuracies and their standard deviations (%) with different k values and used measures.

Table 13 Approximate running times (in seconds) for testing with the optimal parameter values.

Method	Running time	Method	Running time
LDA	0.3	MMDC	0.4
QDA	0.4	CT	7.7
NB	1.1	MNLR	2241
K-means (100 clusters)	2450	SOM	243
RBFN	442	MLP	1250
k-NN (Euclidean and $k = 3$)	0.7	k-NN (Cityblock and $k = 1$)	0.6
k-NN (Correlation and $k = 1$)	0.7	k-NN (Cosine and $k = 3$)	0.7

From Table 12 the obtained mean accuracies of the 10 different classification methods can be seen. The analysis of Tables 1-10 can be confirmed with the observations in Table 12. Naïve Bayes obtained the lowest accuracy among all classification methods. Self-Organizing Map, K-Means and Classification Tree (CT) obtained mean accuracies close to each other, whereas CT achieved the highest score. The rest of the classification methods reached above 90% accuracy. LDA and MNLR had less than 1% difference between their accuracies. Moreover, QDA and RBF network obtained the same accuracy and their difference to the best classification method, Multi-Layer Perceptron, was only 0.4%. Although QDA, RBF network and MLP achieved very good results together with the high accuracies, they did not manage to beat SVM together with one-vs-one method, which obtained above 96% accuracy with 15D features in Joutsijoki and Juhola (2013). Finally, Table 13 shows the approximated running times with the optimal parameter values.

4 Conclusion

In this research we examined the automated taxa identification of benthic macroinvertebrates. This application is an infrequently researched area. In this research we applied altogether 11 different classification methods consisting of both unsupervised and supervised methods. The dataset included 25 features from which 15 were selected to the classifications. These features were the same as used in Joutsijoki and Juhola (2013, 2011b);

Joutsijoki (2012); Kiranyaz et al. (2010a, 2011, 2010b); Tirronen et al. (2009); Ärje et al. (2010) and they are the union of geometrical and statistical features.

We made extensive experimental tests where k-NN was tested with 26 different k values and with four different measures. Moreover, the tests with K-Means were repeated with 93 different K values. Self-Organizing Map was tested with 43 different numbers of neurons and RBF network with 40 different values of σ. Finally, Multi-Layer Perceptrons were tested with configurations $15 \times i \times 8$, when $i = 1, 2, \ldots, 15$ and $15 \times i \times j \times 8$, when $i, j = 1, 2, \ldots, 15$. Altogether MLP was tested with 240 different configurations.

The obtained results were good. Many of the classification methods reached above 90% accuracy. Especially, Quadratic Discriminant Analysis, RBF network, Multi-Layer Perceptron and Minimum Mahalanobis Distance Classifier showed their power in the classification. MLP achieved the best mean accuracy being over 94% and RBF network and QDA obtained nearly 94% accuracies. Furthermore, MMDC reached nearly 93% accuracy. Although the results were good, they did not managed to win SVM used in Joutsijoki and Juhola (2013, 2011a,b); Joutsijoki (2012).

Our future research will concentrate on a larger, 50 species, dataset of benthic macroinvertebrates. With this dataset SVM together with different multi-class extension will be tested. Also, other classification alternatives, employed in this research, will be applied to the larger dataset. An interesting research topic is how different multi-class extensions which are developed for SVM will work on other classification methods and with larger dataset. This topic is an infrequently researched area. Furthermore, hybrid approaches Tulyakov et al. (2008); Wan et al. (2012); Wozniak et al. (2014) for classification will be one interesting research topic to be concerned in future.

Acknowledgements

We thank Finnish Environment Institute, Jyväskylä, Finland for the data. The first author is also thankful to the Maj and Tor Nessling Foundation for the support.

References

Agresti, A. (1990) *Categorical Data Analysis*, John Wiley & Sons, New York.

Ambelu, A., Lock, K., Goethals, P. (2010) 'Comparison of modelling techniques to predict macroinvertebrate community composition in rivers of Ethiopia', *Ecological Informatics*, Vol. 5, No. 2, pp. 147–152.

Barros, A.P., de Assis Tenorio de Carvalho, F., de Andrade Lima Neto, E. (2012) 'A pattern classifier for interval-valued data based on multinomial logistic regression model', *Proceedings of 2012 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 541–546.

Bittencourt, H.R., Clarke, R.T. (2003) 'Use of classification and regression trees (CART) to classify remotely-sensed digital images', *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing*, Vol. 6, pp. 3751–3753.

Bohling, G. (2006) 'Classical normal-based discriminant analysis', *Technical Report*, Kansas Geological Survey. http://people.ku.edu/ gbohling/EECS833 Accessed 11.3.2015.

Bortolan, G., Degani, R., Willlems, J.L. (1991) 'ECG classification with neural networks and cluster analysis', *Proceedings of Computers in Cardiology*, pp. 177–180.

Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Jackel, L.D., LeCun, Y., Müller, U.A., Sackinger, E., Simard, P, Vapnik, V (1994) 'Comparison of classifiers methods: a case study in handwritten digit recognition', *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, Vol. 2, pp. 77–82.

Broomhead, D.S., Loewe, D. (1988) 'Multivariable functional interpolation and adaptive networks', *Complex Systems*, Vol. 2, pp. 321–355.

Chen, Y., Qin, B., Liu, T., Liu, Y., Li, S. (2010) 'The comparison of SOM and K-means for text clustering', Computer and Information Science Vol. 2, No. 3, pp. 268–274.

Cios, K.J., Pedrycz, W., Swiniarski, R.W., Kurgan, L.A. (2007) *Data Mining: A Knowledge Discovery Approach*, Springer-Verlag, New York

Cortes, C., Vapnik, V. (1995) 'Support-vector networks', *Machine Learning*, Vol 20, No. 3, pp. 273–297.

Cover, T.M. (1967) 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory*, Vol. IT-13, No. 1, pp. 21–27.

Dominguez-Granda, L., Lock, K., Goethals, P.L.M. (2011) 'Using multi-target clustering trees as a tool to predict biological water quality indices based on benthic macroinvertebrates and environmental parameters in the Chaguana watershed (Equador)', *Ecological Informatics*, Vol. 6, No. 5, pp. 303–308.

Duda, R.O., Hart, P.E., Stork, D.G. (2001) *Pattern Classification*, 2nd edition. John-Wiley & Sons.

Gaston, K.J., O'Neill, M.A. (2004) 'Automated species identification: why not?', *Philosophical Transactions of the Royal Society B*, Vol. 359, No. 1444, pp. 655–667.
Phil. Trans. R. Soc. Lond. B

Hagan, M.T., Menhaj, M.B. (1994) 'Training feedforward networks with the Marquardt algorithm', *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, pp. 989–993.

Haykin, S. (1999) *Neural Networks: A Compherensive Foundation*, 2nd edition. Prentice-Hall, New Jersey.

Hoang, T.H., Lock, K., Mouton, A., Goethals, P.L.M. (2010) 'Application of classification trees and support vector machines to model the presence of macroinvertebrates in rivers Vietnam', *Ecological Informatics*, Vol. 5, No. 2, pp. 140–146.

Huang, J., Lu, J., Ling, C.X. (2003) 'Comparing naive Bayes, decision trees, and SVM with AUC and accuracy', *Proceedings of the Third IEEE International Conference on Data Mining*, pp. 553–556.

Hsu, C.-W., Lin, C.-J. (2002) 'A comparison of methods for multiclass support vector machines', *IEEE Transactions on Neural Networks* Vol. 13, No. 2, pp. 415–425.

ImageJ: public domain Java-based image processing program, Available: http://rsb.info.nih.gov/ij/

Jain, A.K. (2010) 'Data clustering: 50 years beyond K-means' *Pattern Recognition Letters*, Vol. 31, No. 8, pp. 651–666.

Jian, C., Hongsheng, H., Suxiang, Q., Xiaojun, G. (2008) 'Research on water quality assessment method based on multi-class support vector machines', *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision*, pp. 1661-1665.

Joachims, T. (2001) 'A statistical learning learning model of text classification for support vector machines', *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 128–136.

Joutsijoki, H., Juhola, M. (2013) 'Kernel selection in multi-class support vector machines and its consequence to the number of ties in majority voting method', *Artificial Intelligence Review*, Vol. 40, No. 3, pp. 213–230.

Joutsijoki, H., Juhola, M. (2011a) 'Comparing the one-vs-one and one-vs-all methods in benthic macroinvertebrate image classification', *Proceedings of 7th International Conference on Machine Learning and Data Mining*, 2011, *Lecture Notes in Computer Science*, Vol. 6871, pp. 399–413.

Joutsijoki, H., Juhola, M. (2011b) 'Automated benthic macroinvertebrate identification with decision acyclic graph support vector machines', *Proceedings of 2nd IASTED International Conference on Computational Bioscience (CompBio)*, pp. 323–328.

Joutsijoki, H. (2013a) 'An application of one-vs-one method in automated taxa identification of macroinvertebrates', *Proceedings of the 4th Global Congress on Intelligent Systems*, pp. 125–130.

Joutsijoki, H. (2013b) 'Half-Against-Half structure in classification of benthic macroinvertebrate images', *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3646–3649.

Joutsijoki, H. (2012) 'Half-Against-Half multi-class support vector machines in classification of benthic macroinvertebrate images', *Proceedings of the International Conference on Computer & Information Science*, Vol. 1, pp. 424–429.

Joutsijoki, H. (2014) 'Half-Against-Half structure with SVM and k-NN classifiers in benthic macroinvertebrate image classification', *Journal of Computers*, Vol. 9, No. 2, pp. 454-462.

Joutsijoki, H., Meissner, K., Gabbouj, M., Kiranyaz, S., Raitoharju, J., Ärje, J., Kärkkäinen, S., Tirronen, V., Turpeinen, T., Juhola, M. (2014) 'Evaluating the performance of artificial neural networks for the classification of freshwater benthic macroinvertebrates', *Ecological Informatics*, Vol. 20, pp. 1-12.

Kiranyaz, S., Gabbouj, M., Pulkkinen, J., Ince, T., Meissner, K. (2010a) 'Classification and retrieval on macroinvertebrate image databases using evolutionary RBF neural network', *Proceedings of the International Workshop on Advanced Image Technology (IWAIT)*.

Kiranyaz, S., Ince, T., Pulkkinen, J., Gabbouj, M., Ärje, J., Kärkkäinen, S., Tirronen, V., Juhola, M., Turpeinen, T., Meissner, K. (2011) 'Classification and retrieval on macroinvertebrate image databases', *Computers in Biology and Medicine*, Vol. 41, No. 7, pp. 463–472.

Kiranyaz, S., Gabbouj, M., Pulkkinen, J., Ince, T., Meissner, K. (2010) 'Network of evolutionary binary classifiers for classification and retrieval in macroinvertebrate databases' *Proceedings of 2010 IEEE 17th International Conference on Image Processing (ICIP)*, pp. 2257-2260.

Kohonen, T. (1995) *Self-Organizing Maps*, Springer-Verlag, Berlin.

Kohonen, T., Somervuo, P. (1998) 'Self-Organizing Maps of symbol strings', *Neurocomputing* Vol. 21, No. 1–3, pp. 19-30.

Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., Moldenke, A., Lytle, D.A., Correa, S.R., Mortensen, E.N., Shapiro, L.G., Dietterich, T.G. (2008) 'Automated insect identification through concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects', *Machine Vision and Applications*, Vol. 19, No. 2, pp. 105–123.

Lei, H., Govindaraju, V. (2005) 'Half-Against-Half multi-class support vector machines', *Lecture Notes in Computer Science*, Vol. 3541, pp. 156-164.

Levenberg, K. (1944) 'A method for the solution of certain non-linear problems in least squares', *Quaterly of Applied Mathematics*, Vol. 2, No. 2, pp. 164–168.

Lewis, D.D. (1998) 'Naive (Bayes) at forty: The independence assumption in information retrieval', *Lecture Notes in Computer Science*, Vol. 1398, pp. 4–15.

Liu, C.-L., Nakashima, K., Sako, H., Fujisawa, H. (2003) 'Handwritten digit recognition: benchmarking of state-of-the-art techniques', *Pattern Recognition*, Vol. 36, No. 10, pp. 2271-2285.

Lorena, A.C., de Carvalho, A.C.P.L.F., Gama, J.M.P. (2008) 'A review on the combination of binary classifiers in multiclass problems', *Artificial Intelligence Review*, Vol. 30, pp. 19–37.

Lytle, D.A., Martínez-Muñoz, G., Zhang, W., Larios, N., Shapiro, L., Paasch, R., Moldenke, A., Mortensen, E.N., Todorovic, S., Diettrich, T.G. (2010) 'Automated processing and identification of benthic invertebrate samples', Journal of the North American Benthological Society, Vol. 29, No. 3, pp. 867–874.

Mar, T., Zaunseder, S., Martinéz, J.P., Llamedo, M., Poll, R. (2011) 'Optimization of ECG classification by means of feature selection', *IEEE Transactions on Biomedical Engineering*, Vol. 58, No. 8, pp. 2168-2177.

Marquardt, D.W. (1963) 'An algorithm for least-squares estimation of nonlinear parameters', *Journal ofthe Society for Industrial and Applied Mathemtics*, Vol. 11, No. 2, pp. 431-441.

Mohd, N., Khan, A., Rehman, M.Z. (2013) A New Cuckoo Search Based Levenberg-Marquardt (CSLM) Algorithm, Proceedings of the 13th International Conference on Computational Science and Its Applications (ICCSA) *Lecture Notes in Computer Science*, Vol. 7971, pp. 438–451.

Ougiaroglou, S., Nanopoulos, A., Papadopoulos, A.N., Manolopoulos, Y., Welzer-Druzovec, T. (2007) Adaptive *k*-nearest-neighbor classification using a dynamic number of nearest neighbors, Proceedings of the Eleventh East-European Conference on Advances in Databases and Information Systems (ADBIS) *Lecture Notes in Computer Science*, Vol. 4690, pp. 66–82.

Picton, P. (2000) *Neural Networks*, Second edition. PALGRAVE, New York.

Platt, J.C., Christiani, N., Shawe-Taylor, J. (2000) 'Large margin dags for multiclass classification', *Advances in Neural Information Processing Systems (NIPS 1999)*, Vol. 12, pp. 547–553.

Ranganathan, A. (2004) 'The Levenberg-Marquardt algorithm', http://users-phys.au.dk/jensjh/numeric/project/10.1.1.135.865.pdf. Accessed 20.7.2015.

Rifkin, R., Klautau, A. (2004) 'In defense of one-vs-all classification', *Journal of Machine Learning Research*, Vol. 5, pp. 101–141.

Riverlife project. Available in Finnish: http://www.ymparisto.fi/riverlife (partly English). Accessed 14.2.2012.

Saarikoski, J., Laurikkala, J., Järvelin, K., Juhola, M. (2009) 'A study of the use of self-organising maps in information retrieval', *Journal of Documentation*, Vol. 65, No. 2, pp. 304–322.

Saarikoski, J., Laurikkala, J., Järvelin, K., Juhola, M. (2011) 'Self-organising maps in document classification: A comparison with six machine learning methods' Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA 2011), *Lecture Notes in Computer Science*, Vol. 6593, pp. 260–269.

Song, M.-Y., Park, Y.-S., Kwak, I.-S., Woo, H., Chon, T.-S. (2006) 'Characterization of benthic macroinvertebrate communities in a restored stream by using self-organizing map', *Ecological Informatics*, Vol. 1, No. 3, pp. 295–305.

Tirronen, V., Caponio, A., Haanpää, T., Meissner, K. (2009) 'Multiple order gradient feature for macroinvertebrate identification using support vector machines' *Lecture Notes in Computer Science*, Vol. 5495, pp. 489-497.

Tulyakov, S., Jaeger, S., Govindaraju, V., Doermann, D. (2008) 'Review of classifier combination methods' *Machine Learning in Document Analysis and Recognition Studies in Computational Intelligence*, Vol. 90, pp. 361–386.

Venkatesh, Y.V., Raja, S.K. (2003) 'On the classification of multispectral satellite images using the multilayer perceptron', *Pattern Recognition*, Vol. 36, No. 9, pp. 2161–2175.

Vikramjit, M., Wang, C.-J., Banerjee, S. (2011) 'Text classification: A least square support vector machine approach', *Applied Soft Computing*, Vol. 7, No. 3, pp. 908-914.

Xie, J., Qiu, Z. (2007) 'The effect of imbalanced data sets on LDA: A theoretical and empirical analysis', *Pattern Recognition*, Vol. 40, No. 2, pp. 557–562.

Yu, J., Ekström, M. (2003) 'Multispectral image classification using wavelets: A simulation study', *Pattern Recognition*, Vol. 36, No. 4, pp. 889–898.

Zhang, Y., Zhou, J. (2003) 'A study on content-based music classification', *Proceedings of Seventh IEEE International Symposium on Signal Processing and Its Applications*, Vol. 2, pp. 113–116.

Wan, C.-H., Lee, L.-H., Rajkumar, R., Isa, D. (2012) 'A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine', *Expert Systems with Applications*, Vol. 39, pp. 11880–11888.

Woźniak, M., Graña, M., Corchado, E. (2014) 'A survey of multiple classifier systems as hybrid systems', *Information Fusion*, Vol. 16, pp. 3–17.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D. (2008) 'Top 10 algorithms in data mining', *Knowledge and Information Systems*, Vol. 14, No. 1, pp. 1–37.

Ärje, J., Kärkkäinen, S., Meissner, K., Turpeinen, T. (2010) 'Statistical classification and proportion estimation - an application to a macroinvertebrate image database' *Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 373-378.