ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

E. Weitschek, G. Fiscon, G. Felici

SUPERVISED LEARINING MEETS DNA BARCODING SPECIES CLASSIFICATION

R. 13-16 2013

Emanuel Weitschek - Institute for System Analysis and Computer Science "Antonio Ruberti" (IASI), CNR, Viale Manzoni 30, 00185 Rome, Italy. Email: emanuel.weitschek@iasi.cnr.it

Giulia Fiscon - Institute for System Analysis and Computer Science "Antonio Ruberti" (IASI), CNR, Viale Manzoni 30, 00185 Rome, Italy. Email: giulia.fiscon@iasi.cnr.it.

Giovanni Felici - Institute for System Analysis and Computer Science "Antonio Ruberti" (IASI), CNR, Viale Manzoni 30, 00185 Rome, Italy. Email: giovanni.felici@iasi.cnr.it

ISSN: 1128-3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

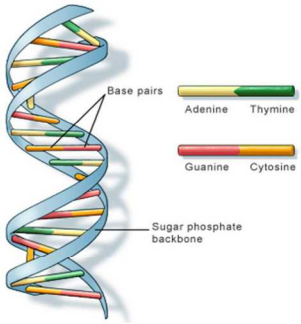
Abstract

Specific gene regions known as DNA Barcode are the main players in the technique of *DNA Barcoding* to identify species of the most life kingdoms. Reliable methods and algorithms support the solving of the DNA Barcode sequence classification problem whose aim is to assign an unknown specimen to a known species through the analysis of its Barcode.

In this work we propose the use of supervised machine learning methods as an effective approach to address the task of species classification through the DNA Barcode sequences. The Weka machine learning classifiers are selected to carry out the DNA Barcode analysis. Specifically, the trees-based J48, the rules-based RIPPER (JRIP), the Bayesian approach Naïve Bayes and functions-based method support vector machines (SMO) are evaluated on simulated and experimental datasets, i.e., public available datasets that belong to the animals, fungi and plants kingdoms. Then, well-established DNA Barcode classification methods, (e.g., phylogenetic trees based NJ and PAR, the similarity based BLAST, and the character based DNA-BAR and BLOG are compared to the Weka classification results.

The classification analysis on synthetic and real data sets shows that supervised machine learning methods are promising candidates for handling with success the DNA Barcoding species classification task. Indeed, the results show that Support Vector Machines and Naïve Bayes methods outperform on average the other analyzed classifiers. However, they do not provide a clear human interpretable classification model. Instead, the ruled-based ones give the diagnostics positions and nucleotide assignments, even if they show slightly lower classification performances.

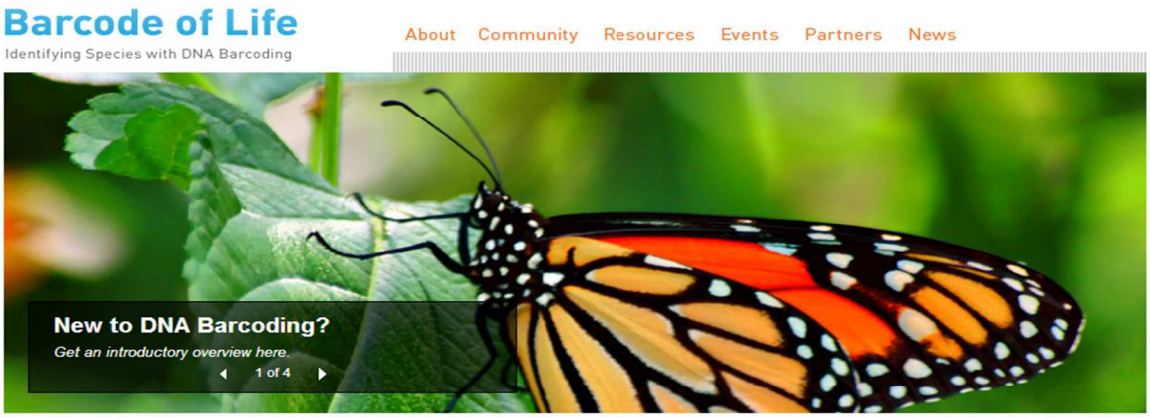
1. Introduction: “Barcode of life”



A DNA molecule is a long sequence made up of two paired strings (i.e., sequence of four characters called *nucleotides*), that build up a double helix structure. The DNA molecule brings hereditary genetic information on which the development of all organism living on Earth is based.

A specific fragment, coming from short portions of mitochondrial DNA, has been defined as *DNA Barcode* and it can be used as marker for organisms of the main life kingdoms. Indeed, in 2003 Hebert [1, 2] proposed the DNA barcoding as a novel technique to identify species. Till then, biological specimens were identified using morphological features, however in some tough cases the identification became complex even for experts. DNA Barcoding solves this problem because it is able to distinguish species and identify specimens (also incomplete, damaged or immature ones) using a very short gene sequence, that can be easily obtained from tiny amounts of tissue.

The DNA Barcode sequences are typically easy to align, show an high variability even among strictly related species, providing all adequate information needed to classify a specimen to species.



Thus, since 2004 the *Consortium for the Barcode of Life* (CBOL) promotes an international initiative devoted to developing DNA barcoding as a global standard for the identification of biological species, aiming to build up an online freely available sequence database (http://www.barcodinglife.org).

2. Methods

2.1 The learning problem statement

As mentioned in the previous section, species classification with DNA Barcode sequences has been proven to be effective on different organisms [3, 4]. Indeed, specific gene regions have been shown as Barcode in the main life kingdoms (e.g., animals, plants and fungi).

The classification problem can be formulated as follows: given a reference library composed of DNA Barcode specimen sequences of known species, an unknown DNA Barcode sequence has to be recognized in a species of the library.

Rather, the classification problem is supervised machine learning problem whose goal is to assign an unknown specimen to a known species starting from its Barcode.

Given

(i) a *set of training* examples $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, containing specimens (x_i) with a priori known species membership (y_i); and

(ii) a *test set* containing specimens which require classification;

the learning function is the following:

$$f : X \rightarrow Y$$

where X is the input space (i.e., the DNA Barcode sequences attributes) and Y is the output space (i.e., the species labels in which input data has to be classified).

The *test set* is made up by either query specimens with unknown species membership, or specimens with a priori known species membership, allowing validation of the specimen classifications correctness.

In general, training and test sets are provided separately. However, it could be even provide only one dataset, which is automatically randomly divided over a training and testing data. The ratio of the number of specimens in the training and testing dataset can be specified.

2.2 Datasets

The classification analysis is performed using a selection of published empirical datasets and simulated DNA Barcode datasets (also used in [5, 6, 7]).

2.2.1 Empirical DATA

Three published empirical datasets (available from "GenBank") have been chosen basing on (i) data with high phylogenetic diversity, which allow to keep a general applicability; (ii) data with complexity in the DNA Barcode classification due to the incomplete clustering in Barcode tree; and (iii) data coming from different genomic compartments.

The selected empirical datasets, summarized in Table 1, are the following:

- *Cypraeidae* [8]

Cypraeidae (Mollusca) are taxonomically one of the most extensively studied marine gastropods. Estimates of the effective population size for Cypraeidae species are not available. The dataset comprises 2008 DNA Barcode sequences with a length of 618 bases and coming from 211 species, whose 112 are represented by five or more sequences.

- *Drosophila* [9]

Drosophila was usually selected as dataset to test the algorithms performance whose goal was assigning sequences to species without Barcode gap. The dataset is made up of 615 DNA Barcode sequences of 19 species, which show a huge number of effective population size. Sequence length is 663 bases and there exist 15 species with a number of representing sequence is higher than five.

- *Inga* [10]

Inga (Fabaceae) is a large genus of tropical leguminous trees. Estimates of the effective population size for Inga species are not available. Lots of Inga species collected in southwestern Amazon are sorted in incomplete DNA Barcode tree. The dataset is made up of 913 DNA Barcodes of length 1838. Such sequences come from 56 species, whose 35 are represented by more than five sequences.

Table 1 Summary of chosen empirical datasets

Dataset	#Sequences	Seq.length	#Species	#Species > 4	Ref
Cypraeidae	2008	614	211	112	[8]
Drosophila	615	663	19	15	[9]
Inga	913	1838	56	35	[10]

Legend: #Sequences = number of dataset sequences comprised in the dataset; Seq.length = length of sequences; #Species = global number of species in the datasets; #Species>4 = species represented by more than 4 sequences; Ref= reference to original publication.

2.2.2 Simulated DATA

Realistic DNA Barcode datasets were simulated with Coalescent package in Mesquite version 2.75 (see the related work [5]). The data were simulated considering time of species divergence and the effective population size, or rather the number of individuals in a population (of a specie) that are contributing genes to the succeeding generations (*Ne*). Firstly, according to the Yule coalescence model [11], gene trees with 1000, 10000 and 50000 individuals of effective population size were simulated, generating datasets composed of 50 species each of 20 individuals. Each simulation was replicated 100-fold. The dataset complexity increases with such a population size. Then, DNA Barcode sequences were simulated on the addictive gene trees, with a sequence length of 650 bases, close to the real size of a standard DNA Barcode.

Table 2 Training set and test set of empirical and simulated data

Dataset	#Sequences	#Instances Training set	#Instances Test set
Cypraeidae	2008	1656	352
Drosophila	615	499	116
Inga	913	785	128
Simulated	20	16	4

2.3 The learning problem solution

Species classification with DNA Barcode sequences has been handled with several reliable approaches. So far, following *ad-hoc* methods have been used [5]: (i) tree-based methods; (ii) similarity-based methods; (iii) character-based methods (also named "diagnostic methods"). Tree-based methods (e.g., PAR [12], NJ [13]) assign unidentified Barcodes (query) to species based on their membership of clusters in a DNA Barcode tree. This approach can be achieved with Parsimony (i.e., PAR [12]), or Neighbor Joining (i.e., NJ [13]). Similarity-based methods (BLAST [14], NN [15]) assign query Barcodes to species based on how much DNA Barcode characters they have in common. Lastly, character-based methods (e.g. DNA-BAR [16], BLOG [6]) rely on the presence/absence of particular characters in DNA Barcode sequences for identification, instead of using them all.

In this work, the efficacy of supervised machine learning methods to classify species with DNA Barcode sequences is shown, through the performance comparison with the traditional DNA Barcode matching methods. The *Weka* machine learning software [17], which includes a collection of supervised classification methods, is adopted to address the task of DNA Barcode analysis. Classifiers families (trees, rules, lazy learners, Bayesian and functions) are tested on public available synthetic and real datasets belonging to the animals and plants kingdoms. In particular, the function-based method Support Vector Machines (SMO), the rule-based RIPPER (JRIP), the classification trees-based C4.5 (J48), and the Bayesian based method Naïve Bayes are considered.

2.3.1 DATA selection

The sequences of the empirical selected datasets are randomly divided into a *reference* set (80% per species), including the sequences with *a priori* assigned species membership (i.e., training set), and a *query* set (20% per species), comprising the unknown DNA Barcode sequences (i.e., test set).

Moreover, each dataset comprise species with 5 or more representing sequences.

Even simulated DNA Barcode sequences are divided into *reference* dataset (training set) and *query* dataset (test set), comprised of 16 and 4 sequences for species, respectively. It is worth noting that in our case since species membership of query dataset are simulated together with the reference dataset, they are also known, allowing *a posteriori* evaluation of their identification accuracy.

2.3.2 DATA pre-processing

An integrated software that converts the DNA Barcode FASTA sequences to the ARFF Weka format was developed, adapting different input formats to permit the execution of the experiments.

Indeed, our input files are DNA Barcode sequence alignments in the standard FASTA format [18], which need to be converted in the Weka input format (ARFF). The FASTA format shows the heading line of each sequence, that is formed by the starting character ">", following by the "specimen ID" and the "species name field" (dived by a vertical bar), respectively. The following lines contain the nucleotides sequences (a string of A, C, G, or T characters).

An example of FASTA format is shown:

```
>CC.MZ_9_ID316|Inga_chartacea

AAACTGCATGCATTTGCCATGACTAGCATTTG
```

Therefore, the algorithm converts FASTA format into the standard Weka format, called ARFF. The latter is composed by two parts. A first part of the file includes the name of the dataset (starting with "@relation") and a heading line (starting with "@attribute") for each attribute, where it is specified the type of attribute (e.g., numeric-a number- or categorical-a string of characters) and the classes enclosed in braces. The following part (starting with "@data") comprises a line for each instance, that shows the attribute values. Fields representing the line are comma-separated.

According to the ARFF format, in our case the attributes represent the nucleotide positions in the sequence and their number is equal to the sequence length in addition to the last one representing the class (i.e., the specie) for each dataset. In particular, the number of attributes is equal to 615, 664 and 1839 for Cypraeidae, Drosophila and Inga, respectively. Each dataset shows the last attribute heading line (starting with "@attribute class") comprising the specie of the analyzed sequences. Moreover, the attribute values are the nucleotides (A, C, G, T) and they are mapped in the set of number from 1 to 4 (1=A, 2=C, 3=G, 4=T). Indeed, since Weka requires the same positions and the same order for the categorical attributes, the latter (A, C, G, T) needed to be converted and mapped into numeric ones (1, 2, 3, 4).

An example of file in ARFF format is shown:

```
@relation Drosophila_test

@attribute pos1 numeric
@attribute pos2 numeric
@attribute pos3 numeric
@attribute class {Drosophila_angor,Drosophila_arizonae}

@data
1,3,4,Drosophila_angor
4,2,4,Drosophila_arizonae
```

2.4 The learning algorithms

The Weka tools collection for Machine Learning and Data Mining analysis is used to solve the species identification problem.

Weka (Waikato Environment for Knowledge Analysis) is a Java open source package that collects the most widespread algorithms to handle mostly classification problems, numeric prediction, or clustering problems. Among the several packages collected in Weka (e.g., either the central system package "weka.core", or the clustering package "weka.cluster", or the package to estimate different probability distributions "weka.estimators"), the "weka.classifier" package includes the implementation of classification and prediction algorithms, comprising the most important class called "Classifier". The latter defines the structure of any schema of classification or prediction assessment and it is made up by two methods, *buildClassifier()* and *classifyIstances()*, whose implementation is necessary for all learning algorithms.

In Table 3 all the available algorithms for classification, numeric prediction and clustering assessments are summarized. In greater detail, Table 4 highlights the Weka classifiers.

Table 3 Weka algorithms collection

Classification	Prediction	Meta	Clustering
Decision trees	Linear regression	Bagging	EM
Support Vector Machine	Model tree generators	Boosting	Coweb
Naïve Bayes	Instance-based learners	Stacking	-
Decision tables	Decision tables	Regression via classification	-
Locally weighted regression	Locally weighted regression	Classification via regression	-
Rule learners	Multi-layer perceptron	Cost sensitive classification	-

Table 4 Weka Classifiers

Kind of classification	Description
Bayes	Bayesian network (e.g. Naive Bayes)
Functions	Linear regression, Neural networks, Support Vector Machine
Lazy	Instance-based similarity (e.g., Nearest neighbor algorithm)
Meta	Bagging, Boosting, Stacking, Regression through classification, Classification through regression, Cost sensitive classification
Rules	Rule-based classifiers
Trees	Tree classifier (e.g., decision tree)
Mi	Algorithms that handle multi-instance data
Misc	Various classifiers that don't fit in any another category

2.4.1 Algorithms description

Among the Weka array of classifiers the following methods are tested: (i) the function-based method support vector machines (SMO); (ii) the rule-based RIPPER (JRIP); (iii) the classification trees-based C4.5 (J48); and (iv) the Bayesian-based method Naïve Bayes.

- *SMO (SVM)*

SMO is the Weka implementation of the supervised learning methods Support Vector Machines (SVM). SMO is a discriminative classifier formally defined by a separating hyperplane, i.e. given labeled training data, the algorithm outputs an optimal hyperplane that identifies classes for new examples. Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. For example, for a linearly separable set of 2D-points which belong to one of two classes, SVM finds a separating straight line that pass as far as possible from all points.

- *Jrip*

Jrip (RIPPER) implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction, which was proposed by William W. Cohen. Classes are examined in increasing size and an initial set of rules for the class is generated using incremental reduced error JRip. Then, all the examples of a particular judgment in the training data are treated as a class, and a set of rules that covers all the members of that class is found. Thereafter, it proceeds to the next class, repeating the same procedure until all classes have been covered.

- *J48*

J48 is the Weka implementation of the decision tree C4.5. It is a supervised tree-based classification method. A decision tree is a simple tree structure whose non-terminal vertexes represent tests on one or more attributes, while the terminal ones reflect the results of the decision. The key advantages of decision trees are the following: (i) they are simple and easily convertible into a set of rules; (ii) both numerical and categorical data can be classified (even if the output attribute must be categorical); (iii) there are no a priori assumptions about the kind of data. However, decision trees are unstable (i.e., variations in the training data can produce different set of attributes to be chosen) and generally multiple output attributes are not allowed.

- *Naïve Bayes*

Naïve Bayes is a Bayesian-based classifier using estimator classes. A Bayesian Network (BN) is the joint probability distribution of a set of variables. BN is solved by specific algorithms, based on the state of the observable variables and *a priori* probabilities represented by the edge in the relations between variables, evaluating the *a posteriori* probabilities of the unknown states. In this way, BN can be considered as a tool of investigation and forecasting. Mathematically, a Bayesian network is a directed acyclic graph whose vertexes are variables or states, while the edges are statistical dependencies between the variables and local probability distributions of the leaf vertexes compared to the values of the parent ones. The absence of an edge between two vertexes reflects their conditional independence. Contrarily, the presence of an edge from a vertex X_i to a vertex X_j can be explained as X_i is a direct cause of X_j.

2.4.2 Parameters configuration

The classification algorithms explained in the previous section are tested first using a standard configuration of parameters and then changing some value in order to compare the different performances (see the

following “Comparative Analysis” subsection for the obtained results). In particular, in Table 5 the standard parameters for each analyzed method are listed.

Table 5 Parameters Configuration

Classifier	Parameters	Description	Value
SMO (SVM)	build logistic models	whether to fit logistic models to the outputs	FALSE
	c	complexity parameter C	1.0
	epsilon	epsilon for round-off error	10^{-12}
	filterType	determines how/if the data will be transformed	normalized training data
	kernel	kernel to use	polyKernel
	numFolds	number of folds for cross-validation used to generate training data for logistic models	-1 (training data)
	tolerance parameters	tolerance parameter	0.001
	random seed	number seed for cross-validation	1
Jrip	folds	amount of data used for pruning. (one fold is used for pruning, the rest for growing the rules)	3
	seed	seed used for randomizing the data	1
	minNO	minimum total weight of the instances in a rule	2.0
	optimizations	number of optimization runs	2
	use pruning	whether pruning is performed	TRUE
J48	confidence factor	confidence factor used for pruning	0.25
	minNumOb	minimum number of instances per leaf.	2
	numFolds		3
	reduced-error pruning	whether reduced-error pruning is used instead of C.4.5 pruning.	FALSE
	sub tree raising	whether to consider the subtree raising operation when pruning	TRUE
	seed	seed used for randomizing the data when reduced-error pruning is used	1
	unpruned	whether pruning is performed	FALSE
	use Laplace	whether counts at leaves are smoothed based on Laplace	FALSE
Naïve Bayes	supervised discretization	use supervised discretization to convert numeric attributes to nominal ones	FALSE
	kernel estimator	use a kernel estimator for numeric attributes rather than a normal distribution	FALSE

3. Results

3.1 Empirical DATA

The results of Weka tested methods are shown for the three selected real datasets: Cypraeidae, Drosophila, and Inga [1]. The performances on query set (test set) and reference set (training set) for each selected dataset are drawn in Figure 1, Figure 2 and Figure 3, respectively. Each figure depicts results on real data through histograms that provide the accuracy rate for all analyzed methods on the test set (panel a of each picture) and training set (panel b of each picture).

SVM and Naive Bayes reach the highest classification performances on the all tested datasets. The reached accuracy is on average around 92%, 95% and 89% for Cypraeidae, Drosophila and Inga query sets, respectively.

[1] Note that even Multi layer perceptron had been run on three datasets, however it required a very high computational cost, not providing the demanded output. Therefore, the results are not shown.

- *Cypraeidae*

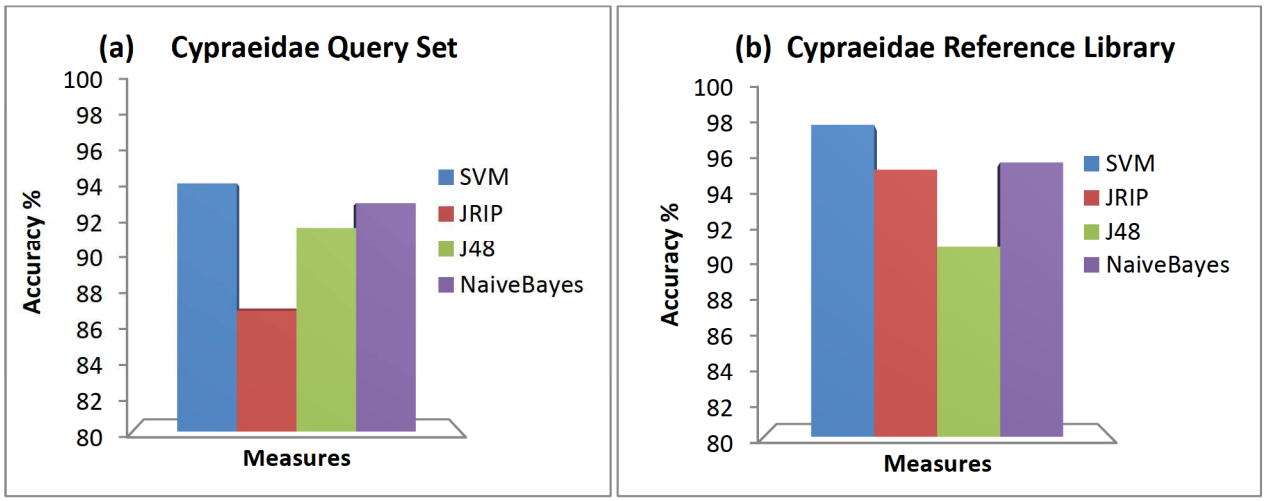


Figure 1 Results on Cypraeidae dataset: (a) DNA Barcode query identification success scores of four methods applied to Barcode sequence dataset; (b) DNA Barcode reference success scores of four methods applied to Barcode sequence dataset.

- *Drosophila*

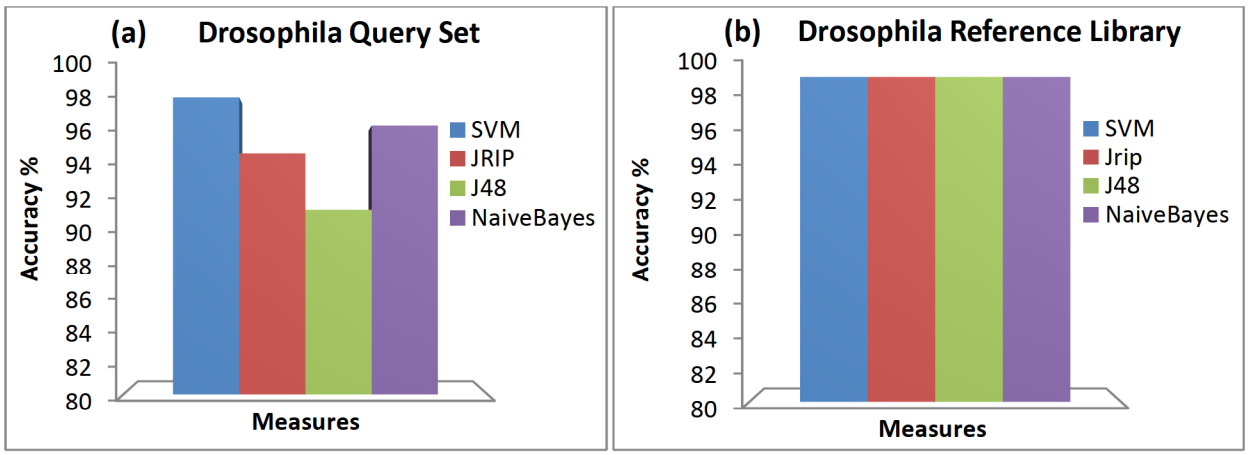


Figure 2 Results on Drosophila dataset: (a) DNA Barcode query identification success scores of four methods applied to Barcode sequence dataset; (b) DNA Barcode reference success scores of four methods applied to Barcode sequence dataset.

- *Inga*

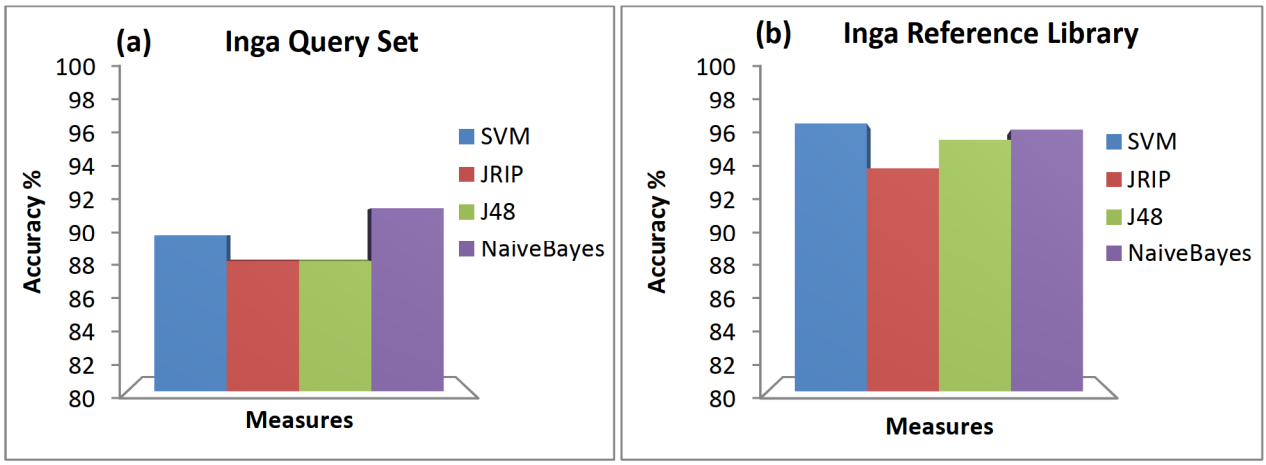


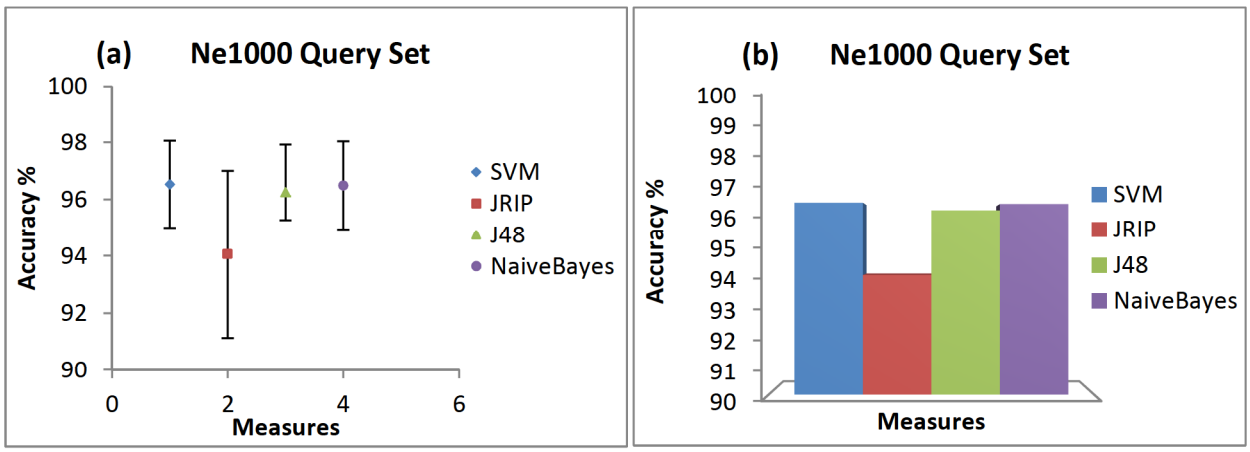
Figure 3 Results on Inga dataset: (a) DNA Barcode query identification success scores of four methods applied to Barcode sequence dataset; (b) DNA Barcode reference success scores of four methods applied to Barcode sequence dataset.

3.2 Simulated DATA

The classification performances on query (test) and reference (training) sets of simulated datasets with *Ne* equal 1000, 10000, 50000 are shown in Figure 4, Figure 5 and Figure 6, respectively. Each figure depicts results on simulated data through histograms and bar-plots, in order to highlight the averaged performances (panels b and d of each picture) together with the standard deviation (panels a and c of each picture).

The results based on synthetic data are largely consistent with results based on empirical ones: SVM and Naive Bayes outperforms the other methods. The reached accuracy is on average around 96% for both *Ne* 1000 and 10000, and 91% for *Ne* 50000.

- *Ne 1000*



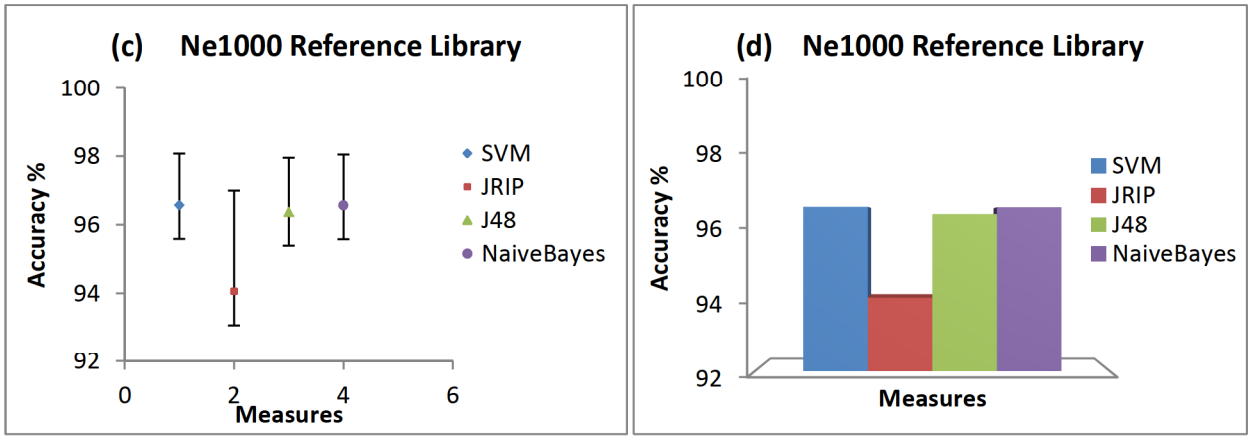


Figure 4 Results of synthetic dataset with *Ne* equal to 1000: (a) bar-plot of DNA Barcode query identification success scores considering for each methods the averaged accuracy and its standard deviation; (b) DNA Barcode query identification success scores of four methods applied to Barcode sequence dataset; (c) bar-plot of DNA Barcode reference success scores considering for each methods the averaged accuracy and its standard deviation; (d) DNA Barcode reference success scores of four methods applied to Barcode sequence dataset.

- *Ne 10000*

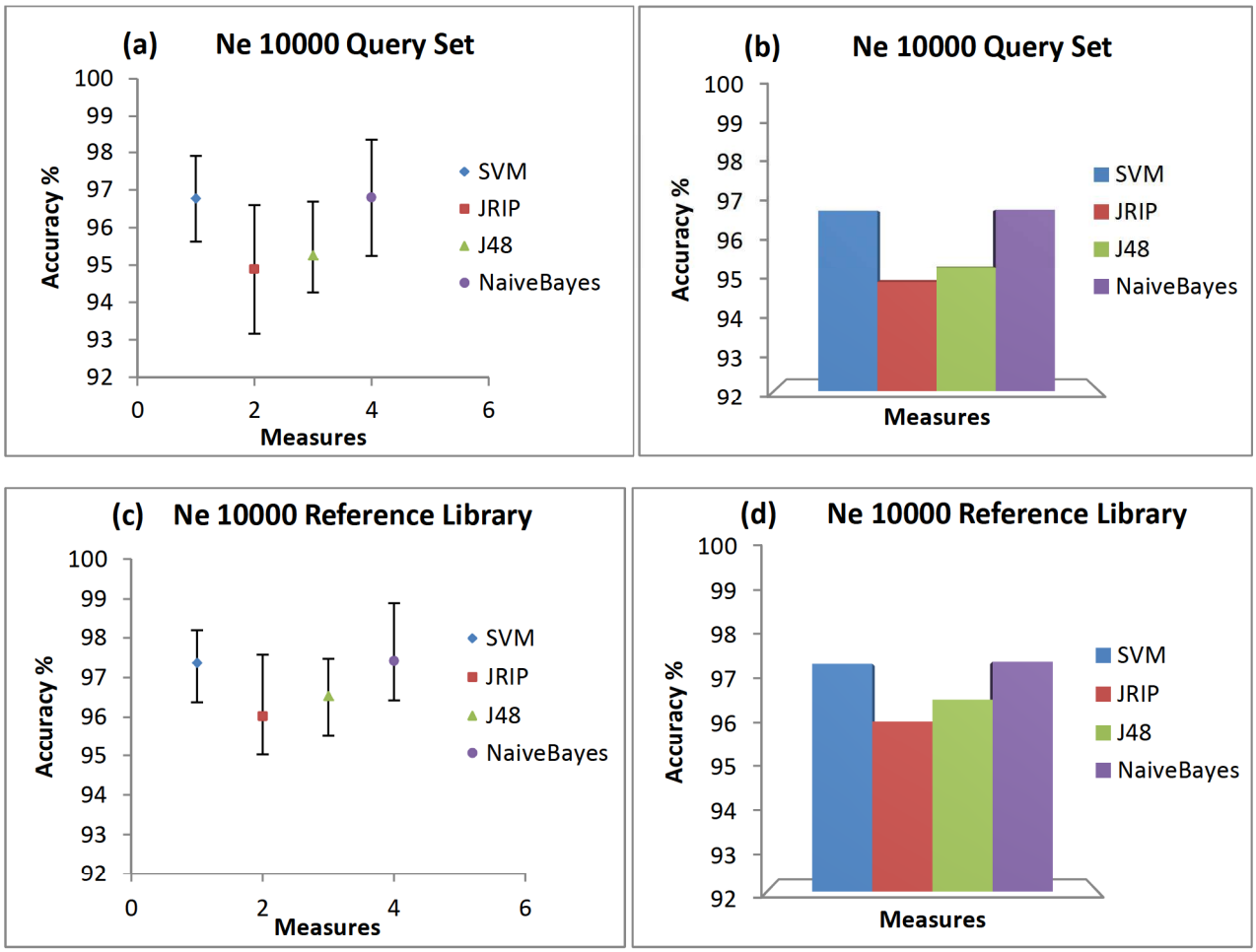


Figure 5 Results of synthetic dataset with *Ne* equal to 10000: (a) bar-plot of DNA Barcode query identification success scores considering for each methods the averaged accuracy and its standard deviation; (b) DNA Barcode query identification success scores of four methods applied to Barcode sequence dataset; (c) bar-plot of DNA Barcode reference success scores considering for each methods the averaged accuracy and its standard deviation; (d) DNA Barcode reference success scores of four methods applied to Barcode sequence dataset.

- *Ne 50000*

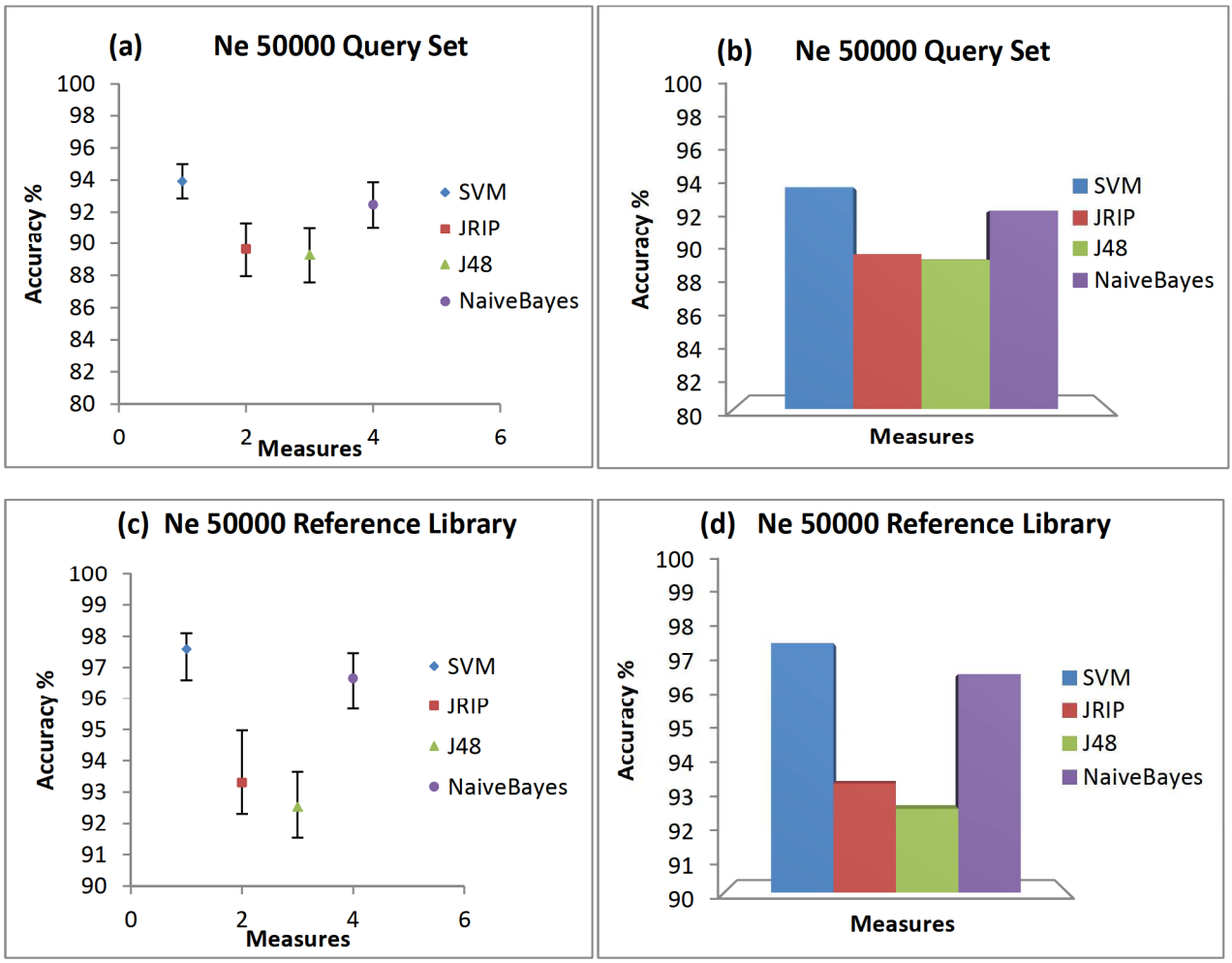


Figure 6 Results of synthetic dataset with *Ne* equal to 50000: (a) bar-plot of DNA Barcode query identification success scores considering for each methods the averaged accuracy and its standard deviation; (b) DNA Barcode query identification success scores of four methods applied to Barcode sequence dataset; (c) bar-plot of DNA Barcode reference success scores considering for each methods the averaged accuracy and its standard deviation; (d) DNA Barcode reference success scores of four methods applied to Barcode sequence dataset.

3.3 Comparative Analysis

A comparative evaluation of the classification results is performed (i) using the different machine learning algorithms chosen between the collection of Weka classifiers; (ii) using same algorithms with different parameters configurations; and (iii) comparing learning results with traditional and well-established DNA Barcode classification methods, as phylogenetic trees (NJ, PAR), similarity based (BLAST), and character based (DNA-BAR, BLOG).

3.3.1 Comparison of Weka learning algorithms

The comparative evaluation of Weka classifiers shows as Support Vector Machines (SVM) and Naïve Bayes methods outperform on average the other classifiers (JRip, J48), both on real (Figure 7) and

synthetic (Figure 8) datasets, although a precise and human interpretable classification model is not provided, as the one of rules-based methods like Jrip.

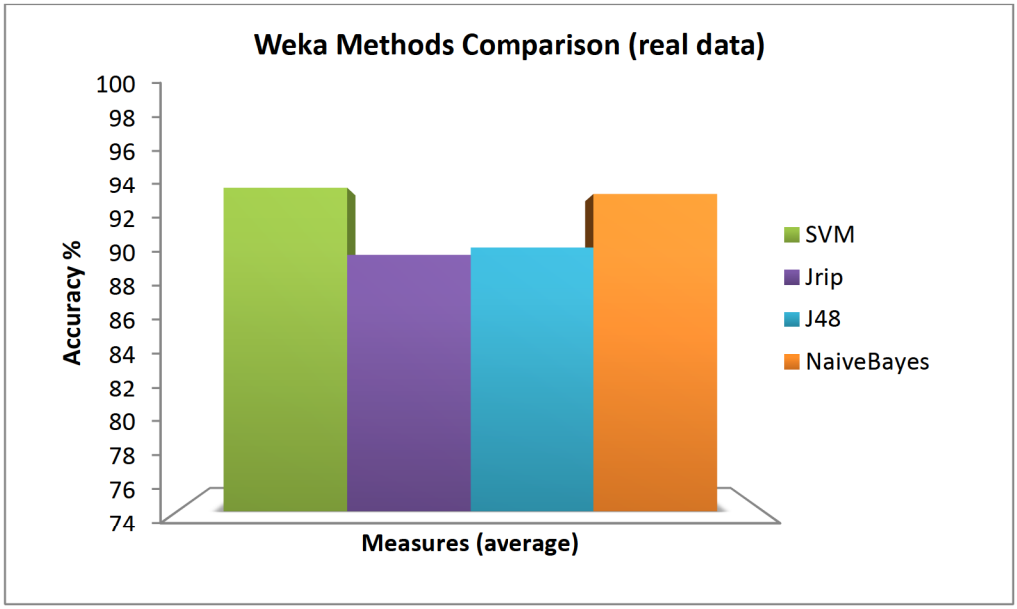


Figure 7 Histogram of sequence identification success of four methods that were applied to empirical query datasets.

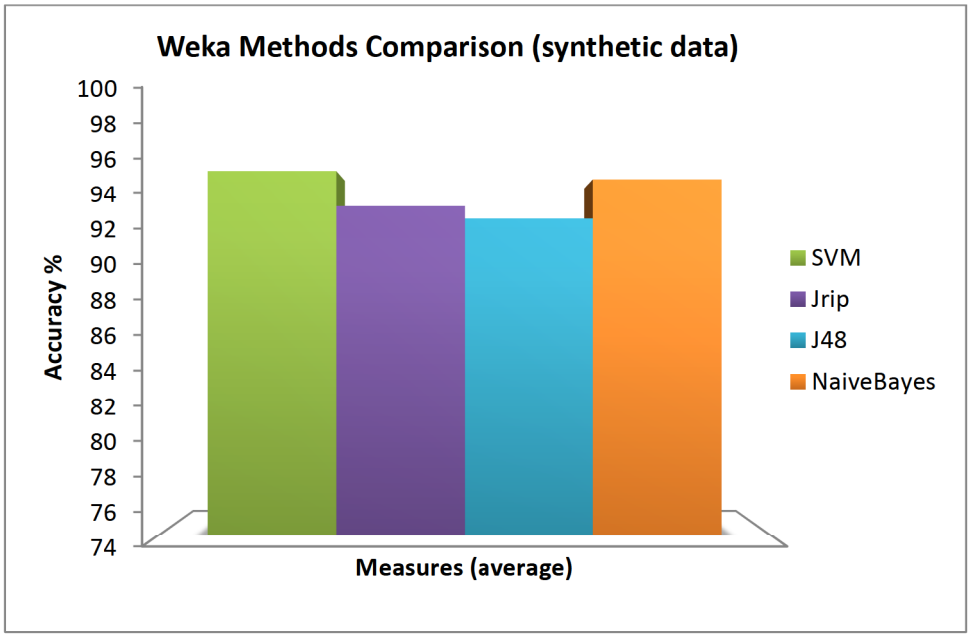


Figure 8 Histogram of sequence identification success of four methods that were applied to simulated query datasets.

3.3.2 Default VS Different parameter configurations of Weka classifiers

The classification performances of machine learning methods tested on real datasets using the parameters configuration of Table 4 are compared with the ones obtained using other configurations, listed in Table 6, 7, 8 for Cypraeidae, Drosophila and Inga, respectively. Figure 9, Figure 10 and Figure 11 show the results of

comparative analysis for the three empirical datasets. No meaning differences among the analyzed configurations does exist, except to the configuration of Drosophila and Inga (number 3 in Table 7 and Table 8), whose kernel parameter for SVM is set to Radial Basic Function (RBF).

- *Cypraeidae*

Table 6 Cypraeidae parameters configuration (see Table 4 for explanation of the default parameters configuration)

#Configuration	Jrip	SVM	J48	Naïve Bayes
1	default	default	default	default
2	pruning = FALSE	built logistic model	reduce error pruning = TRUE	kernel estimator = TRUE
3	optimization = 4	-	unpruned = TRUE	supervised discretization = TRUE

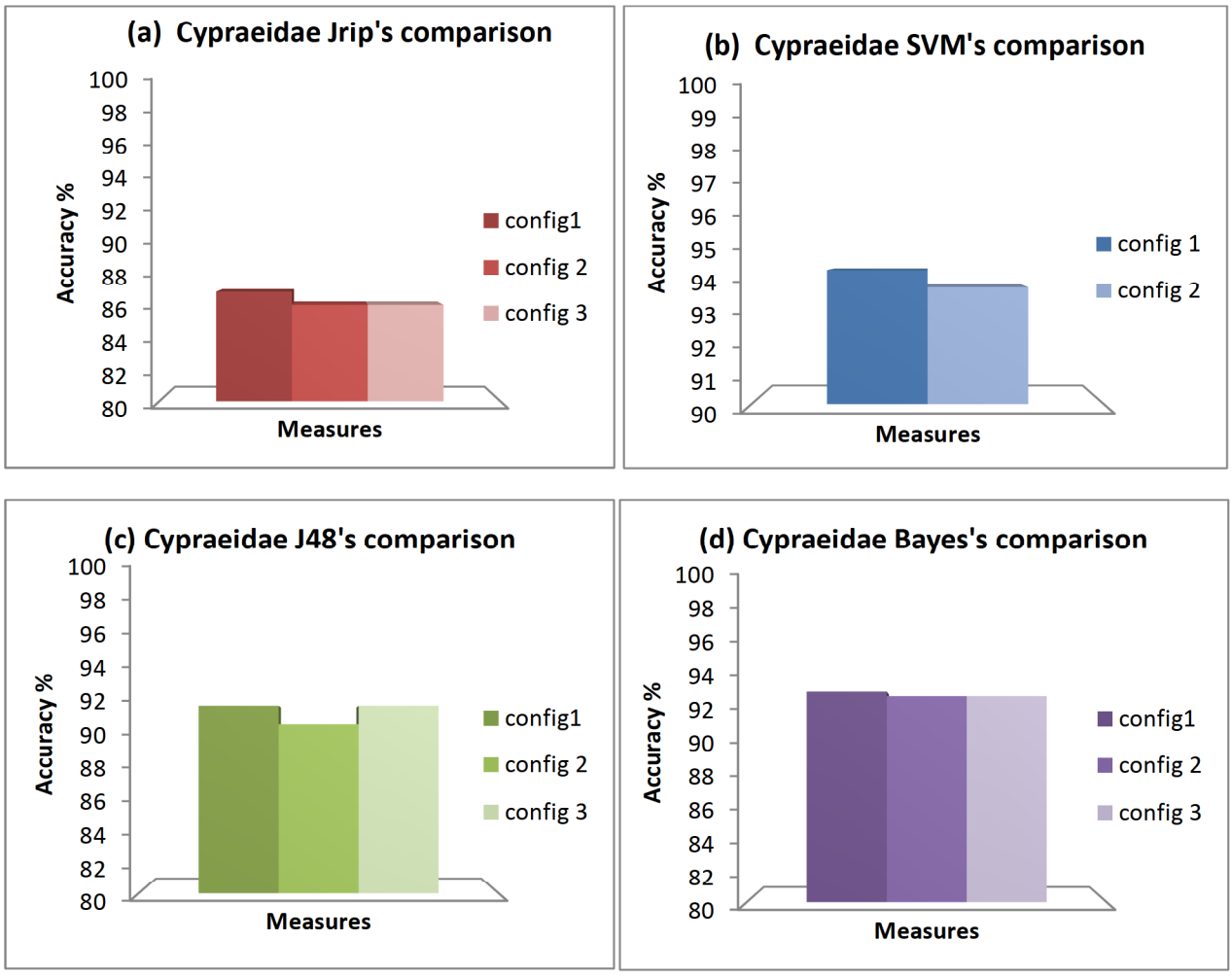


Figure 9 Results of comparative analysis performed on Cypraeidae empirical dataset: (a) Configurations comparison of rules-based method Jrip; (b) Configurations comparison of function-based method SVM; (c) Configurations comparison of tree-based method J48; (d) Configurations comparison of Bayesian-based method Naive Bayes.

- *Drosophila*

Table 7 Drosophila parameters configuration (see Table 4 for explanation of the default parameters configuration)

#Configuration	Jrip	SVM	J48	Naïve Bayes
1	default	default	default	default
2	pruning = FALSE	Standardization	reduce error pruning = TRUE	kernel estimator = TRUE
3	optimization = 10	kernel = RBF kernel	use Laplace = TRUE	supervised discretization = TRUE
4	folds = 20	built logistic model	sub tree raising = FALSE	-
5	optimization = 5 folds = 5	kernel = normalized polyKernel	unpruned = TRUE	-

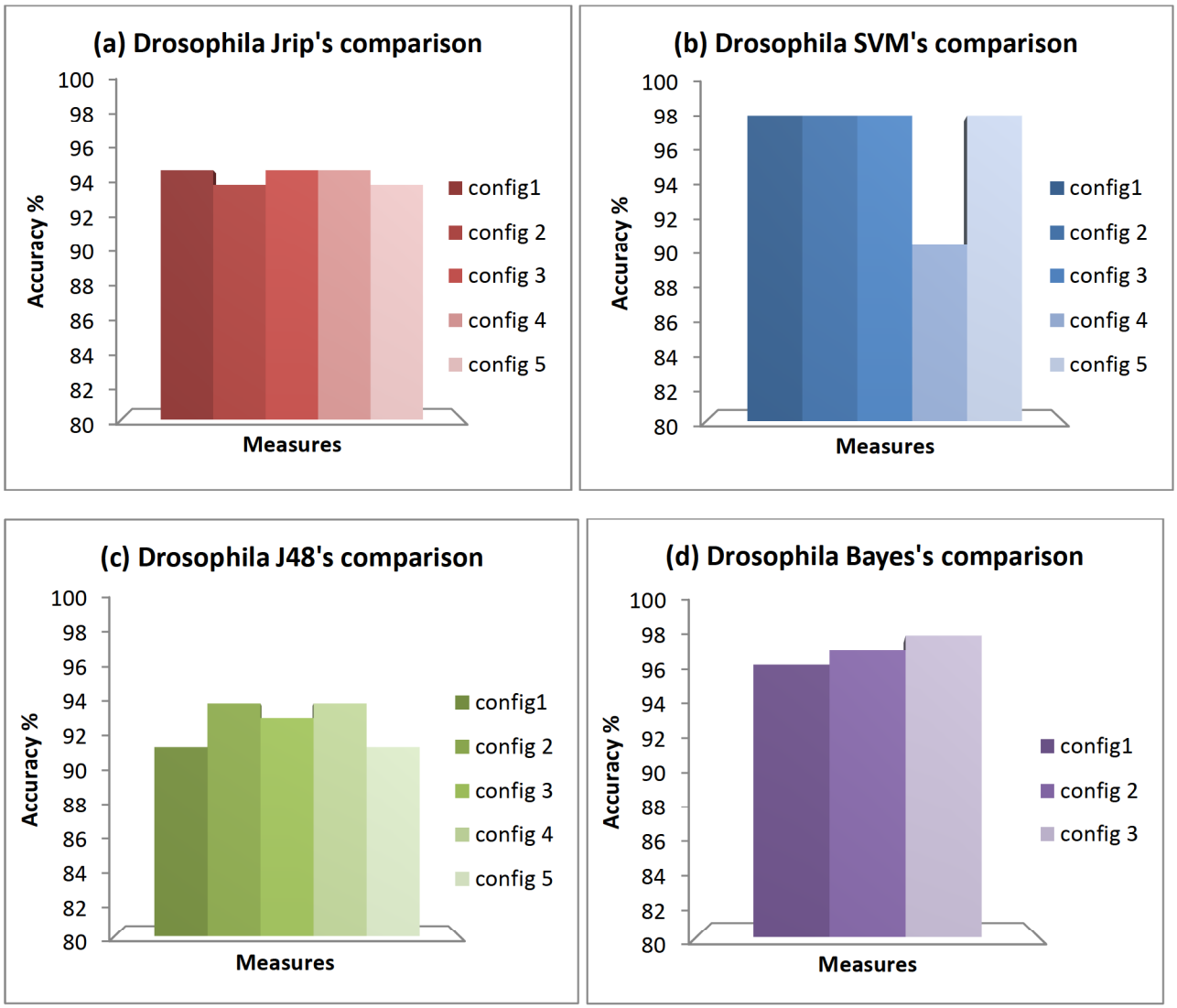


Figure 10 Results of comparative analysis performed on Drosophila empirical dataset: (a) Configurations comparison of rules-based method Jrip; (b) Configurations comparison of function-based method SVM; (c) Configurations comparison of tree-based method J48; (d) Configurations comparison of Bayesian-based method Naive Bayes.

- *Inga*

Table 8 Inga parameters configuration (see Table 4 for explanation of the default parameters configuration)

#Configuration	Jrip	SVM	J48	Naïve Bayes
1	default	default	default	default
2	pruning = FALSE	filter Type = standardize training data	reduce error pruning = TRUE	kernel estimator = TRUE
3	optimization = 5	kernel = RBF kernel	seed ≠ 1	supervised discretization = TRUE
4	-	built logistic model	-	-

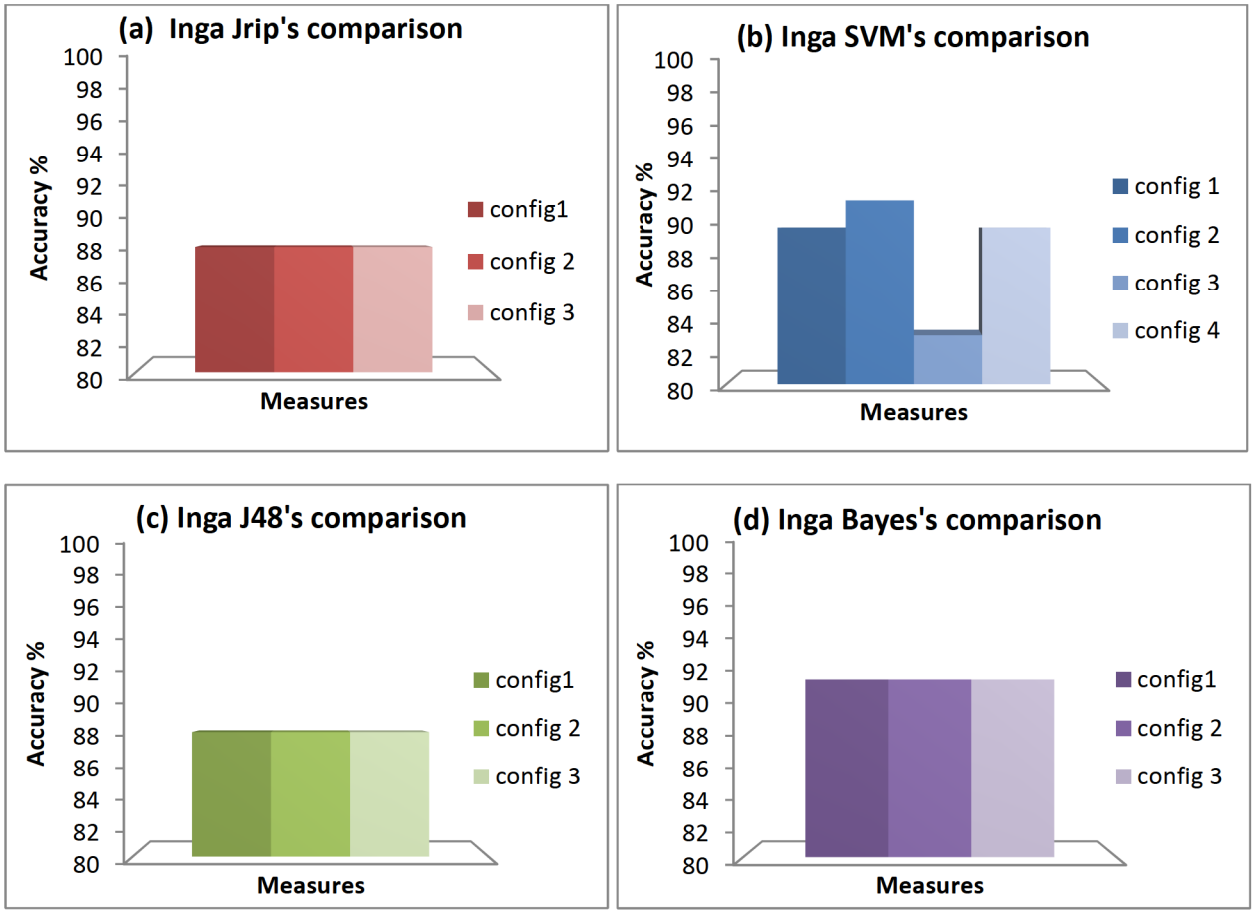


Figure 11 Results of comparative analysis performed on Inga empirical dataset: (a) Configurations comparison of rules-based method Jrip; (b) Configurations comparison of function-based method SVM; (c) Configurations comparison of tree-based method J48; (d) Configurations comparison of Bayesian-based method Naive Bayes.

3.3.3 Weka algorithms VS other consolidated methods

Analysis results on real (Figure 12) and synthetic (Figure 13) datasets show that Weka classifiers as Naïve Bayes and Support Vector Machines (SVM) reach on average the highest classification performances with respect to the other consolidated methods mostly developed to handle the DNA Barcode analysis [5]. However, Naïve Bayes and SVM do not provide a clear and compact human interpretable classification

model. Ruled based methods, as BLOG and RIPPER (JRip) have lower classification performances, but the user is provided with the diagnostics positions and nucleotide assignments (e.g, "If pos3=A and pos150=C then the sequence is X").

Summarizing, on simulated data the supervised machine learning methods outperform the traditional DNA Barcode classification methods (Figure 12)[2]. On real data the classification performances are at a comparable level to the other methods (Figure 13).

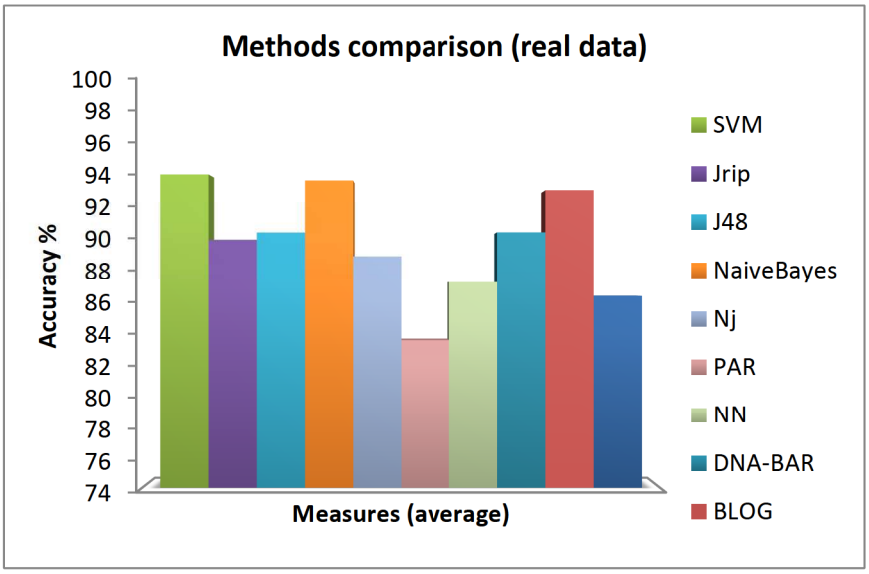


Figure 12 Classification performances (accuracy) for the empirical datasets

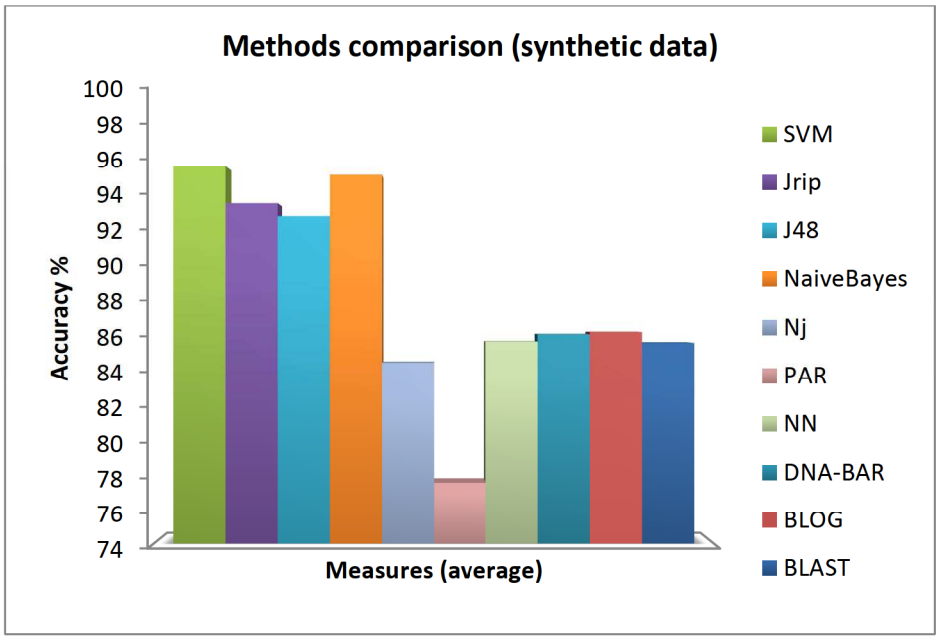


Figure 13 Classification performances (accuracy) for the synthetic datasets

[2] Note that in addition to the three selected data sets (Cypraeidae, Drosophila and Inga), in this experiment the classification performances are obtained testing all analyzed methods even on other published empirical datasets (birds, bats, fishes, algae, and funghi), in order to have more consistent results.

4. Conclusions

The classification analysis shows that supervised machine learning methods are promising candidates for handling with success the DNA Barcode species classification problem, obtaining excellent classification performances. Finally, the DNA Barcoding community is provided with a novel tool to perform species classification.

References

[1] P. Hebert, A. Cywinska, S.L.Ball and J. DeWaard, «Biological identifications through DNA barcodes,» *Proc R Soc B,* vol. 270, p. 313–321, 2003.

[2] P. Hebert, S. Ratnasingham and J. deWaard, «Barcoding animal life: cytochrome c oxidase subunit 1 divergences among closely related species,» *Proc R Soc B,* vol. 270, p. S96–S99., 2003.

[3] D. Schindel and S. Miller, «DNA barcoding a useful tool for taxonomists.,» *Nature,* vol. 435, pp. 17-17, 2005.

[4] P. Hebert and T. Gregory, «The promise of DNA barcoding for taxonomy,» *Syst Bio,* vol. 54, p. 852–859, 2005.

[5] R. v. Velzen, E. Weitschek, G. Felici and F. T. Bakker, «DNA Barcoding of Recently Diverged Species: Relative Performance of Matching Methods,» *PLOS one,* vol. 7, n. 1, 2012.

[6] E. Weitschek, R. V. Velzen, G. Felici and P. Bertolazzi, «BLOG 2.0: a software system for character-based species classification with DNA Barcode sequences. What it does, how to use it,» *Molecular ecology resource,* 2013.

[7] P. Bertolazzi, G. Felici and E. Weitschek, «Learning to classify species with barcodes,» *BMC Bioinformatics,* vol. 10, n. 7, 2009.

[8] C.P.Meyer and G.Paulay, «DNA barcoding: Error rates based on comprehensive sampling,» *PLoS Biol,* vol. 3, p. 2229–2238, 2005.

[9] M. Lou and G. Golding, «Assigning sequences to species in the absence of large interspecific differences,» *Mol Phylogenet Evol,* vol. 56, p. 187–194, 2010.

[10] K. Dexter, T. Pennington and C. Cunningham, «Using DNA to assess errors in tropical tree identifications: How often are ecologists wrong and when does it matter?,» *Ecol Monogr,* vol. 80, p. 267–286, 2010.

[11] M. Steel and A. McKenizie, «Properties of phylogenetic trees generated by Yule-type speciation models,» *Math Biosci,* vol. 170, pp. 91-112, 2001.

[12] A. Edwards and L.Cavalli-Sforza, «The recostruction of evolution,» *Ann Hum Genet,* vol. 27, pp. 105-106, 1963.

[13] N.Saitou and M. Nei, «The neighbor joining method - a new method for reconstructing phylogenetic

tree,» *Mol Biol Evol,* vol. 4, pp. 406-425, 1987.

[14] S. Altschul, T. Madden, A. Schaffer, J. Zhang and Z. Zhang, «Gapped BLAST and PSI-BLAST: a new generation of protein database search program,» *Nucleic Acids Res,* vol. 25, p. 3389–3402, 1997.

[15] F.Austerlitz, O. David, B. Schaeffer, K. Bleakley and M. Olteanu, «DNA barcode analysis: a comparison of phylogenetic and statistical classification methods,» *BMC Bioinformatics,* vol. 10 (suppl14), p. S10, 2009.

[16] B. DasGupta, K. Konwar, I. Mandoiu and A. Shvartsman, «DNA-BAR: distinguisher selection for DNA barcoding,» *Bioinformatics,* vol. 21, p. 3424–3426, 2005.

[17] I. H. Witten, «Weka: Practical machine learning tools and techniques with Java implementations,» *Computer Science Working Paper,* 1999.

[18] W. Pearson, «Rapid and sensitive sequence comparison with FASTP and FASTA,» *Methods in Enzymology,* vol. 183, pp. 63-98, 1990.