BioWeka—extending the Weka framework for bioinformatics

Jan E. Gewehr*,†, Martin Szugat†, Ralf Zimmer
Practical Informatics and Bioinformatics Group, Department of Informatics, Ludwig-Maximilians-University Munich, Amalienstrasse 17, D-80333 Munich, Germany
Received on September 14, 2006; revised on November 25, 2006; accepted on January 03, 2007
Advance Access publication January 19, 2007
Associate Editor: Thomas Lengauer

ABSTRACT
Summary: Given the growing amount of biological data, data mining methods have become an integral part of bioinformatics research. Unfortunately, standard data mining tools are often not sufficiently equipped for handling raw data such as e.g. amino acid sequences. One popular and freely available framework that contains many well-known data mining algorithms is the Waikato Environment for Knowledge Analysis (Weka). In the BioWeka project, we introduce various input formats for bioinformatics data and bioinformatics methods like alignments to Weka. This allows users to easily combine them with Weka's classification, clustering, validation and visualization facilities on a single platform and therefore reduces the overhead of converting data between different data formats as well as the need to write custom evaluation procedures that can deal with many different programs. We encourage users to participate in this project by adding their own components and data formats to BioWeka.

Availability: The software, documentation and tutorial are available at http://www.bioweka.org.

Contact: support@bioweka.org

1 INTRODUCTION
The tremendous amount of biological data available nowadays leads inevitably to the application of data mining methods for tasks like classification and clustering. However, for many bioinformatics applications, the data (e.g. sequences) have to be transformed into a feature-based representation first. For instance, the well-known fold recognition server GenTHREADER (Jones, 1999) computes a number of scores based on alignments in a first step and then combines these using a neural network. Other applications like ECLAT (Friedel et al., 2005) generate feature representations for biological sequences by e.g. counting codons.

The popular data mining framework Weka (Witten and Frank, 2005) offers a broad variety of useful tools for machine learning purposes. The BioWeka project extends the Weka framework with additional bioinformatics functionalities including new input formats and alignments. These extensions can be combined with the built-in functionalities of Weka. This enables the user to employ all the useful facilities Weka has to offer together with well-known bioinformatics algorithms in a consistent way on a single platform. Figure 1 shows an overview of the way the BioWeka components can be used together with the underlying Weka software. Further, the extendability of BioWeka and its base classes allows for rapid development and evaluation of new methods.

2 OVERVIEW OF BIOWEKA

2.1 The Weka software
Weka is a widely accepted machine learning toolkit in bioinformatics (Frank et al., 2004) implemented in Java. It offers many state-of-the-art approaches in an object-oriented framework, including classifiers (SVMs, decision trees, rule learners, etc.) and clustering methods. Weka also provides a rich graphical user interface and a simple but powerful command line interface. The software contains standard validation methods like e.g. cross-validation. Further, it allows for visualization and statistical evaluation of the results.

2.2 Input formats
Weka uses a special format (ARFF) for its datasets. Since biological data comes in a lot of different formats, BioWeka contains an input layer for converting well-known formats into ARFF (and vice versa for some formats). So far, the following data formats are supported:

- MAGE-ML (Spellman et al., 2002) and CSV compatible formats for gene expression data,
- FASTA (Pearson and Lipman, 1988), EMBL (Kulikova et al., 2004), Swiss-Prot (Bairoch and Boeckmann, 1991) and GenBank (Benson et al., 1993) for the storage of biological sequences in ASCII files.
- InterProScan (Zdobnov and Apweiler, 2001) for the annotation of sequence patterns.

In addition to these formats already provided by BioWeka, users can easily extend BioWeka by adding their own converters. Custom XML formats can be incorporated into BioWeka using XSL stylesheets.

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.
2.3 Bioinformatics extensions

In Weka, all classes that modify a dataset are called filters. BioWeka contains new filters for handling sequences like the annotation of symbol properties (see bioweka.org for a full list of features). Another large part of BioWeka enables users to align sequences with each other using different alignment methods, including BLAST (Altschul et al., 1990), PSI-BLAST (Altschul et al., 1997) and JAligner (Moustafa et al., 2006). For alignment-based classification, a couple of different evaluation mechanisms are provided (e.g. by selecting the class with the highest average alignment score or the class with the highest single alignment score). Furthermore, custom alignment score evaluation schemes can be plugged in.

2.4 Extending and contributing to BioWeka

BioWeka is licensed under the GNU General Public License. This ensures that any contributions made to BioWeka are free to anyone. New components can be rapidly built on top of the existing base classes of BioWeka. For sequence formats, it is also possible to build on BioJava classes (see http://www.biojava.org). We encourage bioinformatics developers and users of Weka to participate in the BioWeka project by contributing code or exemplary datasets.

2.5 Using BioWeka

One has to download both the Weka and the BioWeka distribution and include the Weka JAR in the CLASSPATH variable for BioWeka. The BioWeka startup script provides access to Weka as well as BioWeka. For the BLAST and PSI-BLAST classifiers, a BLAST installation is necessary. In the Explorer GUI, users can import the new data formats listed above using BioWeka’s converters and apply BioWeka’s filters and classifiers.

3 DISCUSSION

In bioinformatics research, often (newly developed) classifiers have to be compared to other, well-known classifiers. In order to use many methods, it may be necessary to deal with many different input and output formats. Further, it may be inevitable to implement a customized evaluation framework around different programs.

Weka is a well-known framework that offers many standard machine learning methods. BioWeka makes it easy to use a number of data formats relevant for bioinformatics with Weka. Everything from classification to validation can be done with such data without further overhead using the standard workflow in Weka. In addition, some bioinformatics-specific methods have been integrated into Weka via BioWeka.

An example that illustrates BioWeka’s strengths: given a dataset such as a FASTA-file containing protein sequences with protein class annotations as provided e.g. by ASTRAL (Chandonia et al., 2004), one well-known bioinformatics task is to build a classifier that is able to classify as many sequences within this set correctly as possible in a cross-validation setup. With BioWeka, users can directly input such data. One way to do classification on sequences can be to derive features using BioWeka’s symbol filters or to import InterProScan results for the sequences via BioWeka’s loader for use with Weka’s classifiers. Further, users have the option of using alignment-based classification directly on the sequences with alignment methods such as e.g. BLAST.

In addition, the multifactor dimensionality reduction of the Weka-CG project (Moore et al., 2006) and the Weka LibSVM project (EL-Manzalawy and Honavar, 2005) come with the distribution.

To conclude, the integration of bioinformatics methods and other useful tools into Weka allows users to perform many bioinformatics standard tasks without the overhead of parsing data formats or writing code that combines different software packages. Developers can make use of BioWeka’s abstract classes and interfaces in order to prototype and test new algorithms. Again, this reduces the overhead of writing converter as well as evaluation classes and allows to concentrate directly on the methods. Comparison with many other methods can be done directly in BioWeka.

Finally, BioWeka is highly configurable and available free of charge.

ACKNOWLEDGEMENTS

We thank all contributors to the BioWeka project. J.G. was funded by the DFG under grant PROSEQO II (Zi 616/2). M.S. was partly funded in the HOBIT project by the Helmholtz-Gemeinschaft.

Conflict of Interest: none declared.

REFERENCES


