Sequence analysis

FunSiP: a modular and extensible classifier for the prediction of functional sites in DNA

Michiel Van Bel[1,2], Yvan Saeys[1,2] and Yves Van de Peer[1,2,*]

[1]Department of Plant Systems Biology, VIB and [2]Department of Molecular Genetics, Ghent University, Technologiepark 927, 9052 Gent, Belgium

ABSTRACT

Motivation: Many problems in genome annotation are tackled by using a classification model to predict functional sites such as splice sites, translation start sites or stop codons. Locating the correct position of these sites remains one of the most important but also one of the most difficult issues in the structural annotation of genomes. Most of the software currently in use is written for a very specific problem, thereby limiting the possibilities for reuse.

Summary: We developed a software platform that uses a very general approach towards the classification of functional sites in DNA sequences. The program uses an *ab initio* approach towards the identification of these sites, and extends SpliceMachine, a previously developed splice site predictor that shows state-of-the-art performance for both donor and acceptor splice site recognition in the human and *Arabidopsis thaliana* genome.

Availability: The program is developed as a stand-alone Java application, and is available as GPLv3 open-source software. The program, source and documentation can be obtained from the 'Software' section at http://bioinformatics.psb.ugent.be/

Contact: Yves.VandePeer@psb.ugent.be

Supplementary information: Supplementary data is available at *Bioinformatics* online.

1 INTRODUCTION

Identifying functional sites such as splice sites, translation start sites or stop codons in a DNA sequence is the first step in the genome annotation process. Each of these sites can be identified by a short conserved sequence, namely GT/GC for the donor sites, AG/AC for the acceptor sites, ATG for the start sites and TAG/TGA/TAA for stop codons. Evidently, context information is of vital importance when distinguishing between true functional sites and pseudo sites. By extracting various features around each putative site, we gain the ability to classify all sites with a high level of confidence, as was shown in the state-of-the-art performance of SpliceMachine (Degroeve *et al.*, 2005). We now generalized the approach of SpliceMachine and built a modular platform which can predict any type of functional site in a sequence, provided there exists a short conserved sequence (e.g. AG for acceptors) characterizing the site, and provided some training data for this site is available.

The results obtained from the classifier can then be used by a gene prediction tool, such as EuGène (Schiex *et al.*, 2001) to obtain the full gene structure.

2 METHODS

2.1 Technology

The program is implemented as a stand-alone Java application, making use of the WEKA machine learning library (Witten and Frank, 2005) to perform feature selection and classification, and of the Log4j library (Gupta, 2005) for formatted and directed output. The program requires at least Java version 1.5.0 to work and runs on a whole range of platforms including MacOS, Windows and different flavors of Linux and Unix. A modular design was chosen for the program, leading to an extensible platform that can be used for a broad range of classifications types.

2.2 General outline

The program has several modes of operation, but the ones most frequently used will be either building a classification model or using a previously built model to classify sites of interest in a sequence. Both modes of operation rely on several variables, which are defined in an editable configuration file. The main steps when building a classification model are: (1) collect training data, (2) extract information from the data and (3) build a classification model. The main steps when evaluating a genomic sequence are: (1) extract information from the sequence and (2) classify using the classification model.

2.3 Features

The program provides numerous features, including:

(1) Built-in support for feature selection. By applying univariate feature selection techniques, the predictive performance of the classifier increases significantly (Saeys *et al.*, 2007). Support for saving and loading the results of feature selection itself is also included.

(2) Built-in support for feature extraction from the DNA sequence: the set of possible (but not limiting) features include *k*-mer patterns, amino acids and RNA secondary structure features.

(3) A general framework and workflow, which is the same for all genomes: (a) collect training data, (b) optimize the extraction methods, (c) build a classifier and (d) use the classifier to distinguish between true or pseudo sites. Detailing the extraction methods and the type of site of interest is implemented in a configurable and extensible way.

(4) The optimization of the extraction methods and the classification of large numbers of genome files can take quite some time on a single
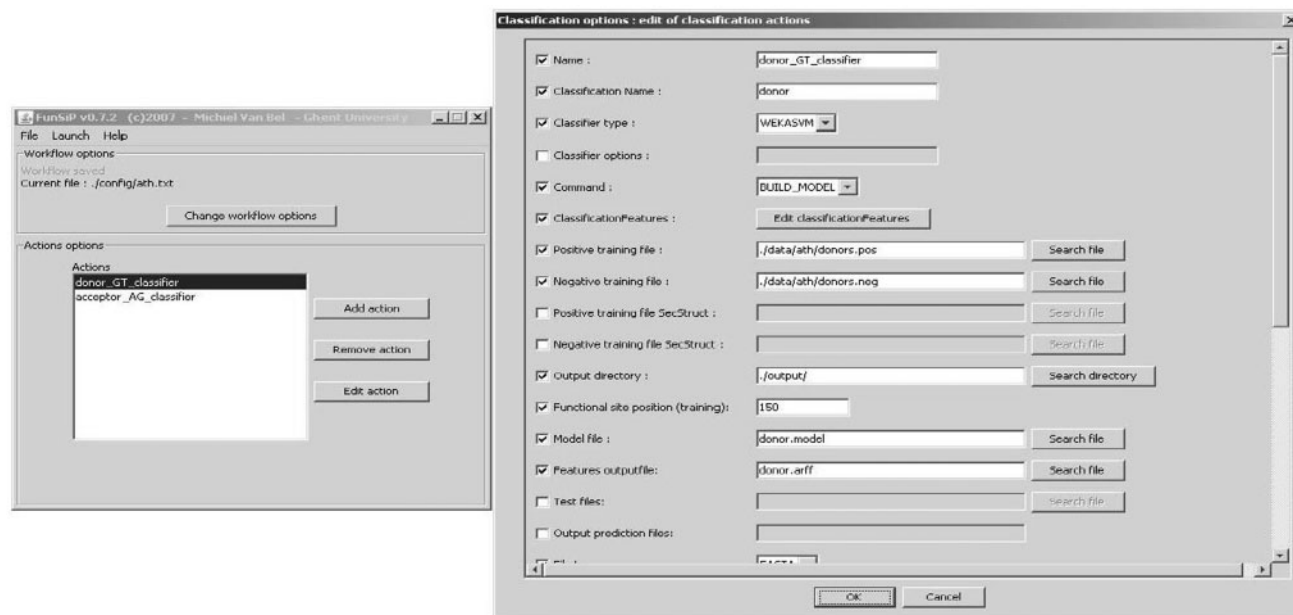
---

[*]To whom correspondence should be addressed.

Fig. 1. Display of the graphical user interface of Fun Si P, showing a wide variety of configurable options.

machine. These problems are largely parallelizable, and are thus very well suited for distributed computing. Therefore, support for Grid Engine (Sun Microsystems, 2008), a well-known software package for distributed computing and is supported by FunSiP.

(5) A graphical user interface to launch the program, edit the configuration options and evaluate the results is available (Fig. 1). Because the number of configuration options is quite large, extensive documentation and a number of example configuration files is supplied with the program.

2.4 Availability

The program is available at http://bioinformatics.psb.ugent.be/ under 'Software'. The program is open source and is licensed under the GPLv3 license or later. The program, source code and full documentation are also available at the website. The documentation includes installation notes, a guide on editing configuration files, a description of the internal working of the platform and a guide for extending the platform with user-defined methods for data extraction.

3 DISCUSSION

The program presented here is an extensible platform for recognizing and classifying functional sites in a genome sequence. Each functional site is assigned a score, which reflects how well it compares with both positive and negative training data. These results can then be used as input for a gene-prediction tool. Table 1 compares the results of FunSip with and without feature selection to SpliceMachine (Degroeve *et al.*, 2005), a state-of-the-art splice site predictor. Values in the table denote the F-measure (harmonic mean of precision and recall; Van Rijsbergen, 1979) on a 10-fold cross-validation experiment using 1000 positive and 10 000 negative examples.

Table 1. Performace comparison of FunSiP

| | Arabidopsis | |
	Donor	Acceptor
SpliceMachine	0.8752	0.8301
FunSiP	0.8664	0.8313
FunSiP with *gain ratio* FS	0.9013	0.8529

Values are F-measures obtained by 10-fold cross-validation of 1000 positive and 10 000 negative training examples.

REFERENCES

Degroeve,S. *et al.* (2005) Predicting splice sites from high-dimensional local context representations. *Bioinformatics*, **21**, 1332–1340.

Gupta,S. (2005) *Pro Apache log4j*. 2nd edn. APress, Berkeley, CA.

Schiex,T. *et al.* (2001) EuGène, an eukaryotic gene finder that combines several types of evidence. *Lect. Notes Comput. Sci.*, Vol. 2066, pp. 111–125.

Saeys,Y. *et al.* (2007) A review of feature selection techniques in bioinformatics. *Bioinformatics*, **23**, 2507–2517.

Sun Microsystems (2008) Grid Engine. Available at http://gridengine.sunsource.net/ (Last accessed date May 29, 2008).

Witten,I.-H. and Frank,E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd edn. Morgan Kaufmann, San Francisco.

Van Rijsbergen,C. (1979) *Information Retrieval.* 2nd edn. Butterworths, London.