A New Machine Learning Dataset for Hierarchical Classification of Transposable Elements

Bruna Zamith Santos[1], **Ricardo Cerri**[1]

[1] Federal University of São Carlos
Department of Computer Science
Rod. Washington Luís – km 235 – São Carlos – SP – Brazil

bruna.zamith@hotmail.com, cerri@dc.ufscar.br

Abstract. *Transposable Elements (TEs) are DNA sequences that can change their location within the genome. They make up a large portion of the DNA in organisms, and contribute to genetic diversity within and across species. Furthermore, they increase the size of the genome and may affect the functionality of genes. Accurate classification of TEs present in a genome is an important step towards understanding their effects on genes and their role in genome evolution. Usually, TEs classification is performed using homology-based Bioinformatics tools, comparing a sequence with a database with many sequences belonging to previously known TE classes. This is a limited strategy, since it ignores the sequences' biochemical properties, and also the hierarchical relationships that may exist between the different TE classes. Based on existing proposals to establish a hierarchical TE taxonomy, we propose a new dataset for TE classification, having features that try to consider sequence properties that cannot be represented only by character sequences. Furthermore, the proposed dataset is hierarchically structured, facilitating its use by conventional and hierarchical classification methods. Focusing on investigating the interpretabiliy potential of our features, we tested our new dataset using decision trees and rule induction algorithms. The experiments showed promising results.*

1. Introduction

Transposable Elements (TEs) are DNA sequences that have the ability to move within the genome of a cell, changing, thereby, the activity of certain genes. According to [Wicker et al. 2007], TEs can be classified in a hierarchically taxonomy containing subclasses and superclasses.

In general terms, TEs can be divided into two major classes, according to their transposition mechanisms: Class I (retrotransposons) and Class II (transposons). The retrotransposons are transposed via RNA "copy and paste" mechanism — they are transcribed into RNA and then, again, transcribed into DNA. On the other hand, transposons use a "cut and paste" mechanism, and do not rely on RNA for the transposition process. Retrotransposons are the most abundant elements in eukaryotic genomes, contributing to increase their size. When TEs are embedded into other genes, they can modify and even reduce the activity of proteins. These alterations generate changes that either make impossible the organism survival, or contribute to genetic variability. Thus, the study of methods for TE classification is very important for TE's behavior comprehension, in order

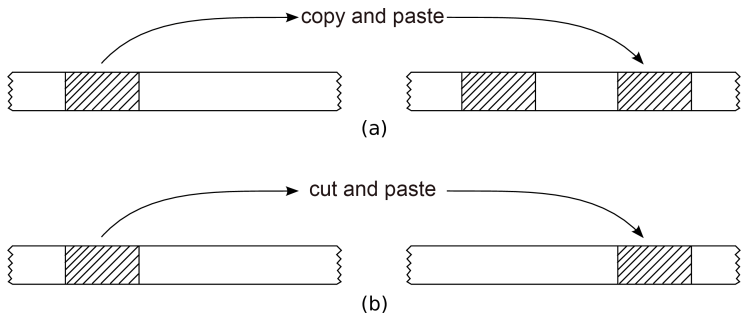

Figure 1. Difference between the mechanisms of transposition Class I and Class II TEs. (a) transposition of retrotransposons ("copy and paste"); (b) transposition of transposons ("cut and paste").

to help understanding how they affect genes functional mechanisms. Figure 1 illustrates the process of TE's transposition.

Currently, the identification and classification of TEs still employs a lot of manual work, despite the existence of automated methods that seek for TE's candidate sequences in the genomes. According to [Bergman and Quesneville 2007], there are many methods proposed to classify TEs, both using homology, structural information, and also seeking for repetitions in sequences. However, all those methods have limitations, because either they assume that the sequences involved are very similar, thus propagating incorrectly made classifications, or they are specific to a particular type of element [Costa et al. 2013]. Furthermore, the use of homology between sequences ignores many of their biochemical properties. Additionally, with the exception of few methods like [Costa et al. 2013, Abrusán et al. 2009], none of them induce models automatically from the data provided, using Machine Learning (ML).

In [Wicker et al. 2007], a TE's classification taxonomy is proposed, where their families and superfamilies are structured in a hierarchy. Part of this hierarchy is shown in Figure 2. As can be seen in the figure, TEs are divided into four hierarchical levels plus the root. At each level, each node corresponds to a TE family/superfamily.

In this work, an hierarchically structured TE dataset was proposed, based on [Wicker et al. 2007] taxonomy. This dataset has been formatted for use by machine learning algorithms, with attributes that try to incorporate knowledgement that cannot be encoded by simple character sequences. In possession of this database, the performance of different ML algorithms can be investigated. The advantage of using ML is based upon the fact that the methods learn how to automatically identify TEs through the use of datasets with previously identified TEs. It is also possible to extract interpretable knowledge by the means of classification rules. To the best of the author's knowledge, this is the first attempt to build a ML hierarchical dataset for TEs classification considering the whole [Wicker et al. 2007] taxonomy.

The remainder of this work is organized as follows. Section 2 presents some works proposed for TE classification; our new ML dataset for hierarchical classification of TEs is presented in Section 3; the experiments performed to evaluated the discriminative power of our attributes are presented in Section 4, together with a discution with the results; finally, Section 5 presents our conclusions and future research directions.

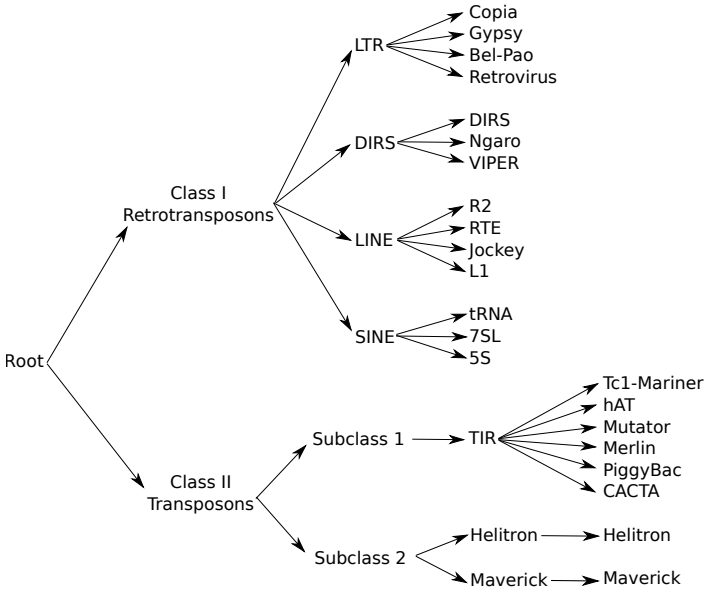

Figure 2. Example of hierarchical taxonomy classification of TEs. Adapted from [Wicker et al. 2007].

2. Related Work

Some methods were proposed for TE classification. However, with very few exceptions, these methods use homology, structural information, or search for repetitions in the sequences [Bergman and Quesneville 2007]. They assume that involved sequences are very similar, and also are specific for given types of TEs [Costa et al. 2013]. In addition, with exception of few methods [Abrusán et al. 2009, Costa et al. 2013, Loureiro et al. 2013], none of them automatically induces classification models from the provided data.

The LTRDigest method [Steinbiss et al. 2009], as an example, is specific for Long Terminal Repeats (LTR) retrotransposons. The method is initiated with a list of LTR-Retrotransposons, and then annotate these sequences with protein domains (using Hidden Markov Models) and other structural regions. It then finds groups in the LTR-retrotransposons, which can be checked to see if they correspond to known TEs.

TEClass [Abrusán et al. 2009] classifies sequences in Classes I and II. Class I elements can then be classified in LTRs and non-LTRs. The non-LTR elements can be classified in LINE and SINE. The classification is performed using binary Support Vector Machines (SVMs), and k-mers as attributes, but no hierarchical relationships were considered for the SVMs induction. Also, the dataset used does not consider the TEs [Wicker et al. 2007] taxonomy.

[Feschotte et al. 2009] proposed RepClass, a method consisting of three different classification modules: homology based, search for structural characteristics, and search for target site duplications. The three modules provide classification in different granularities, which are them integrated for the obtention of a single classification.

The Pastec method [Hoede et al. 2014] uses HMMs and different characteristics to classify TEs. Some of these characteristics are structural (sequence length, presence of LTRs and DIRs), homology and conserved domains.

3. A new Hierarchical Machine Learning Dataset for TEs Classification

This section presents the procedures performed to construct the new proposed ML dataset. In order to build a dataset with attributes that give biologically relevant and discriminative characteristics, we used signatures from the PROSITE database [Sigrist et al. 2012] of protein families and domains. PROSITE consists of biologically significant sites, patterns and profiles, which help in the identification of which protein family a sequence belongs to.

For the development of the dataset, we propose a pipeline which integrates different Bioinformatic tools. We first collected nucleotide sequences from the PSGB Repeat Element Database [Nussbaumer et al. 2013]. This database provides an extensive collection of biological data. The extracted sequences have their characterization represented as depicted in Figure 3, according to [Wicker et al. 2007]

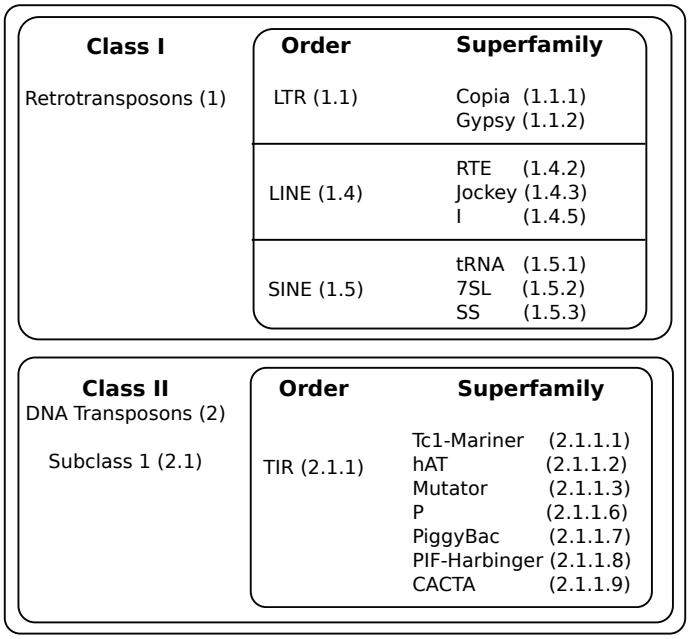

Figure 3. PSGB TEs classification according to Wicker's taxonomy.

As can be seen in Figure 3, the Wicker's taxonomy classify TEs into Class, Order and Superfamily. Class II DNA Transposons can be further divided into Subclass 1 and Subclass 2. Only Subclass 1 is shown in the figure. To hierarchically structure the MIPS sequences, we associated an id to each of the classes in each level. Thus, 2.1.1.3 (Mutator) is a subclass of 2.1.1 (TIR), which is a subclass of 2.1 (Subclass 1), which in turn a subclass of 2 (DNA Transposons). The symbol "·" is then used as a level divisor in our proposal.

Because PROSITE is a protein family database, and PSGB database contains only nucleotide sequences, we needed to translate our DNA sequences into amino acid sequences considering all six reading frames, in order to search them against PROSITE. However, as not all TE sequences are guaranteed to be translated into proteins, one alternative is to retrieve Open Reading Frames (ORFs) from our nucleotide sequences. An ORF is a nucleotide sequence with great potential to be translated to amino acid.

To retrieve ORFs, we used the GETORF tool [Rice et al. 2000] within the INTERPROSCAN software [Jones et al. 2014], which receives as input a collection of DNA sequences in fasta format. The tool returns, for each sequence, the ORFs found for that sequence. Having the ORFs for each TE sequence, we can scan them using a specific tool to search for PROSITE signatures.

We are using TE ORFs and searching them against PROSITE signatures to try to answer the following question: can PROSITE signatures be used to discriminate different classes of TEs? We believe the answer is yes, based on the fact that the signatures we are using as dataset attributes were found in ORFs present in TE sequences. Thus, the same patterns and profiles present in the PROSITE protein families may be present in the TE sequences.

The search for PROSITE signatures was performed using the PS-SCAN tool [Castro et al. 2006], which allows scanning of amino acid sequences against motifs of the PROSITE collection. For each ORF sequence, a number of annotated PROSITE signatures was retrieved, which were then associated to the original DNA TEs sequences. For each TE sequence, the presence or absence of such signatures were used as attributes for ML methods. Figure 4 illustrates the pipeline previously described. For illustration purposes, a small part of the final proposed dataset is shown in Table 1.

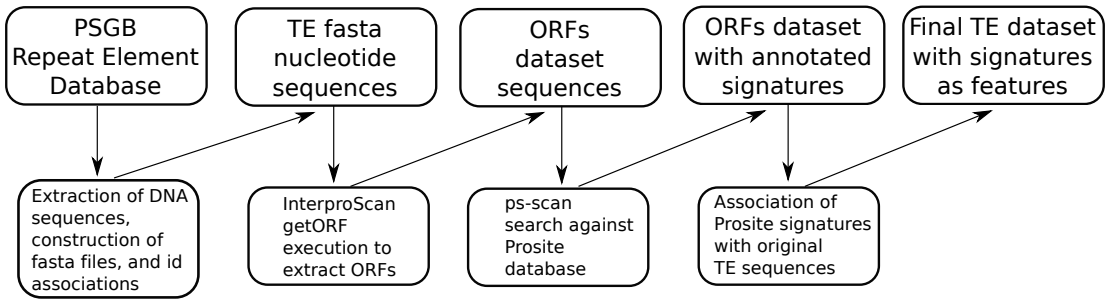

Figure 4. Pipeline for the proposed ML dataset for hierarchical classification.

Table 1. Illustration of the proposed dataset.

Sequence ID	PS00001	PS00004	...	PS60014	TE Class
Sequence ID1	1	1	...	0	1.4
Sequence ID2	1	1	...	0	1.1.2
Sequence ID3	0	1	...	0	1.4
Sequence ID1	0	0	...	0	1.5
⋮	⋮	⋮	⋮	⋮	⋮
Sequence IDn	1	1	...	1	1.1.2

From the 18678 sequences downloaded from the PSGB dataset, our pipeline es ishas up to now processed 4568 of them. This is because scanning the whole PROSITE database against signatur a very time consuming task. Until now, PS-SCAN has found 407 distinct PROSITE signatures annotated in three TE orders (LTR, LINE and SINE) and two TE superfamilies (COPIA and GYPSY). The number of sequences retrieved from each of the five TE classes is described in Table 2.

The five TE classes shown in Table 2 are Class I Retrotransposons. These TEs copy themselves in two stages, first from DNA to RNA by transcription, then from RNA

Table 2. Number of sequences for each TE class.

TE class	LTR	COPIA	GYPSY	LINE	SINE
# sequences	1053	1093	1764	470	188

back to DNA by reverse transcription. The DNA copy is then inserted into the genome in a new position. Reverse transcription is catalyzed by a reverse transcriptase enzyme, which is often encoded by the TE itself. This class of TEs behaves very similarly to retroviruses, such as HIV. It is composed of five orders, among which the retrotransposons with Long Terminal Repeats (LTRs), which are identical sequences (up to minor variations) of a few hundred nucleotides, the long interspersed nuclear elements (LINEs), and the short interspersed nuclear elements (SINEs). As Class I TEs as very abundant in genomes of many species, these are ideal candidates for evaluating how well discriminative our proposed features are.

4. Experiments and Discussion

In order to verify the discriminative power of the PROSITE signatures when used as TE dataset features, we performed experiments using two popular classifiers, the J48 decision tree induction algorithm [Quinlan 1993], and the JRip rule induction algorithm [Cohen 1995]. Both algorithms are implemented within the RWeka package [Hornik et al. 2009].

As can be seen in Table 2, our current dataset is very unbalanced. Thus, we performed two sets of experiments: (i) considering the complete unbalanced dataset, and (ii) a balanced dataset considering only the classes 1.1, 1.1.1 and 1.1.2. For each of these datasets, we executed the J48 and JRip algorithms using the 10-fold cross-validation strategy. To evaluate the results, we averaged the overall accuracy considering all classes over the 10 executions. We also averaged the per-class precision, recall and fmeasure results. Per-class precision, recall and and Fmeasure are particularly useful when class instances are not evenly distributed. The next subsections detail the experiments performed.

4.1. Complete Dataset

Considering the J48 algorithm, an average accuracy of 64.34% was obtained, with a standard deviation value of 2.9. For illustration purposes, Figure 5 shows part of the decision tree induced using the first training fold. The complete decision tree has 221 leaves. For the JRip classifier, the average accuracy obtained was 58.60%, with standard deviation value of 3.6. Analogously to J48, average per-class precision, recall, and fmeasure results were calculated. Part of the rules induced using the first fold is shown in Figure 6. The total number of rules obtained was 21. As can be seen in both figures, the numbers in parenthesis mean the coverage/errors in the training data, which is a convention in tree/rule induction. Thus, in Figure 6, for example, the rule `(PS00007 = 0) and (PS00001 = 0) and (PS00004 = 0) => TE Class 1.5 (60.0/19.0)` means that there were 60 examples with absence (0) of these three signatures that were correctly classified as belonging to superfamily 1.5, and 19 examples misclassified.

The bar graphs in Figures 7, 8 and 9 show the averaged the per-class precision, recall and fmeasure results for the J48 and JRip algorithms. By Figure 7, we can conclude that J48 obtained the highest precisions for classes 1.1.1 and 1.1.2 (fewer false positive

```
PS00006 = 0
|   PS00294 = 0: 1.5 (52.0/13.0)
|   PS00294 = 1: 1.1 (3.0/1.0)
PS00006 = 1
|   PS00001 = 0
|   |   PS50099 = 0
|   |   |   PS50079 = 0: 1.5 (80.0/35.0)
|   |   |   PS50079 = 1
|   |   |   |   PS00342 = 0: 1.4 (3.0/1.0)
|   |   |   |   PS00342 = 1: 1.1.1 (2.0/1.0)
|   |   PS50099 = 1: 1.1.1 (2.0/1.0)
|   PS00001 = 1
|   |   PS00008 = 0
|   |   |   PS00016 = 0
|   |   |   |   PS00009 = 0
|   |   |   |   |   PS00004 = 0: 1.5 (12.0/3.0)
|   |   |   |   |   PS00004 = 1: 1.4 (9.0/2.0)
|   |   |   |   PS00009 = 1: 1.4 (10.0/2.0)
|   |   |   PS00016 = 1: 1.1.2 (2.0/1.0)
|   |   PS00008 = 1
|   |   |   PS00123 = 0
|   |   |   |   PS00004 = 0
|   |   |   |   |   PS00142 = 0
|   |   |   |   |   |   PS00016 = 0
|   |   |   |   |   |   |   PS50079 = 0
|   |   |   |   |   |   |   |   PS50994 = 0
|   |   |   |   |   |   |   |   |   PS00009 = 0: 1.1.2 (128.0/71.0)
|   |   |   |   |   |   |   |   |   PS00009 = 1: 1.4 (57.0/26.0)
|   |   |   |   |   |   |   |   PS50994 = 1: 1.1.2 (3.0)
|   |   |   |   |   |   |   PS50079 = 1: 1.1.1 (4.0/2.0)
```

Figure 5. Part of the J48 decision tree for the complete dataset

```
(PS00007 = 0) and (PS00001 = 0) and (PS00004 = 0)
    => TE Class=1.5 (60.0/19.0)
(PS00007 = 0) and (PS00006 = 0)
    => TE Class=1.5 (30.0/8.0)
(PS00294 = 0) and (PS00342 = 0) and (PS00142 = 1)
    => TE Class=1.4 (23.0/0.0)
(PS00294 = 0) and (PS50994 = 0) and (PS50878 = 1) and (PS00009 = 1)
    => TE Class=1.4 (42.0/8.0)
(PS50079 = 1) and (PS50994 = 0) and (PS50319 = 1) and (PS00017 = 0)
    => TE Class=1.1 (78.0/7.0)
(PS50079 = 1) and (PS50994 = 0) and (PS50321 = 1) and (PS50317 = 0)
    => TE Class=1.1 (39.0/1.0)
(PS50994 = 1) and (PS50879 = 0) and (PS50878 = 0) and (PS50158 = 1)
    => TE Class=1.1.1 (418.0/78.0)
(PS50878 = 0) and (PS50994 = 1) and (PS50879 = 0) and (PS50099 = 1)
    => TE Class=1.1.1 (133.0/48.0)
```

Figure 6. JRip Rules - Complete data set

results in comparison to JRip). However, for classes 1.1, 1.4 and 1.5, JRip obtained higher precision values.

Considering the recall results, Figure 8, J48 obtained better results in the majority of the classes. Thus, although more precise in some classes, *e.g.* 1.4 and 1.5, the JRip algorithm obtained a lower coverage considering these classes (lower recall).

When we combine precision and recall through fmeasure (Figure 9), we conclude that J48 performed better than JRip in the complete dataset. Still, JRip obtained a higher standard deviation, indicating that the J48 algorithm was the more stable classifier.

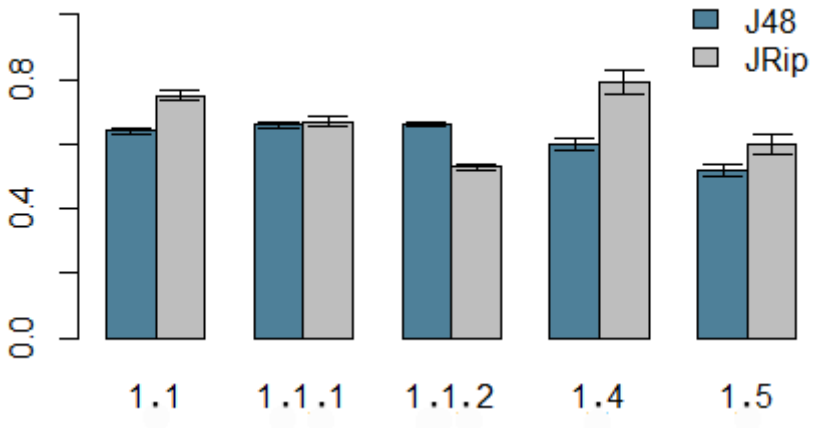

Figure 7. Precision values for the complete dataset

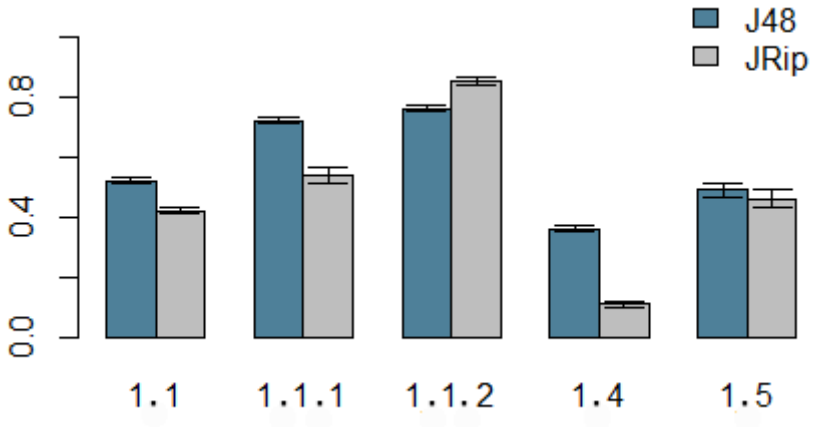

Figure 8. Recall values for the complete dataset

Regarding the discriminative power of our features, we consider these first experiments very promising, since good results were obtained. Also, we consider that with more instances, even better results can be obtained. In addition, instead of considering only the presence or absence of the PROSITE signatures, we could consider also the frequency of such signatures. Similar strategy was adopted in works such as [Wan et al. 2012] where Gene Ontology terms were used as features for protein function prediction datasets.

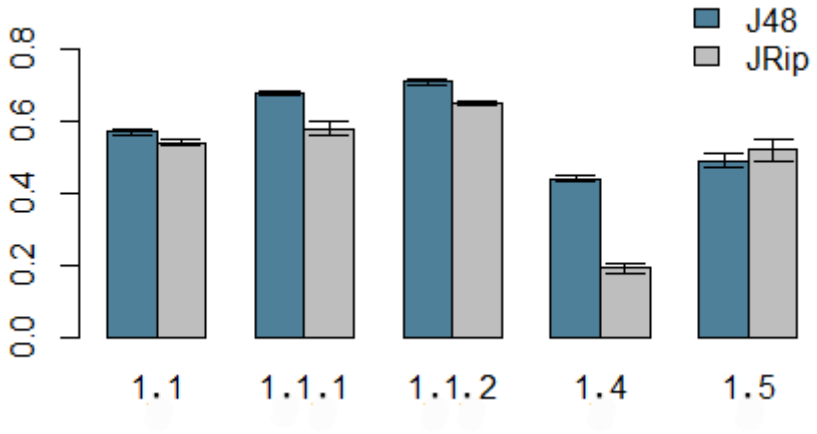

Figure 9. Fmeasure values for the complete dataset

4.2. Partial Dataset

In order to perform experiments with a balanced dataset, we removed instances belonging to the classes 1.4 e 1.5. As shown in Table 2, the number of instances belonging to classes 1.1, 1.1.1 and 1.1.2 is considerably larger than the other two remaining classes. Thus, by removing instances from classes 1.4 and 1.5, and balancing the number of instances for the remaining classes, better classification results are expected. Balancing was performed in order to have the same number of instances for each class. Thus, because the original dataset has 1053 instances belonging to class 1.1, we removed instances from classes 1.1.1 and 1.1.2, letting all three classes with 1053 instances.

After balancing the dataset, the average accuracy obtained by J48 was 68.03%, with standard deviation value of 2.2. For the JRip classifier, the average number of correct classifications was 66.09%, with standard deviation value of 2.2. For both classifier, per-class precision, recall, and Fmeasure results were also calculated.

The graphs shown in Figures 10, 11 and 12 refer to the balanced partial dataset. It is possible to conclude that the results for J48 and JRip were very close, and even the same in some cases. Thus, if we compare the results obtained in the complete and partial dataset, J48 can be considered a more robust classifier, performing better in the unbalanced dataset.

5. Conclusions and Future Works

In this paper, we proposed a new Machine Learning (ML) dataset for Hierarchical Classification of Transposable Elements (TEs). The dataset was hierarchically structured according to the taxonomy proposed by [Wicker et al. 2007], and the presence/absence of PROSITE signatures was used as features.

To construct our dataset, a pipeline connecting different Bioinformatic tools was developed. First, nucleotide sequences were collected from the PSGB public dataset, and tools such as GETORF and PS-SCAN were used to obtain Open Reading Frames (ORFs) and PROSITE signatures. To validate the discriminative power of our features, we performed experiments using two popular interpretable classifiers, the J48 decision tree induction algorithm, and the JRip rule induction algorithm.

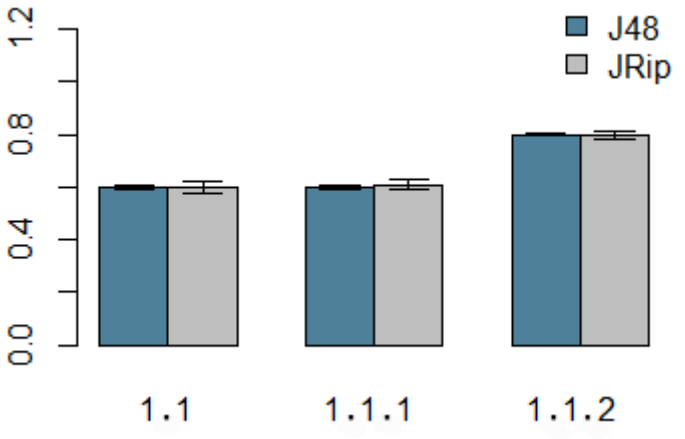

Figure 10. Precision values for the partial dataset

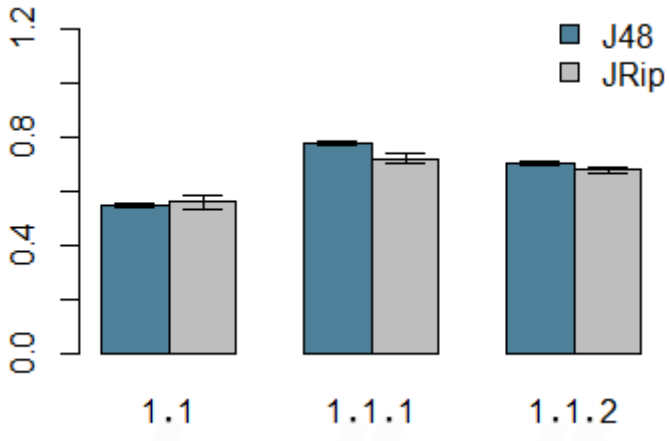

Figure 11. Recall values for the partial dataset

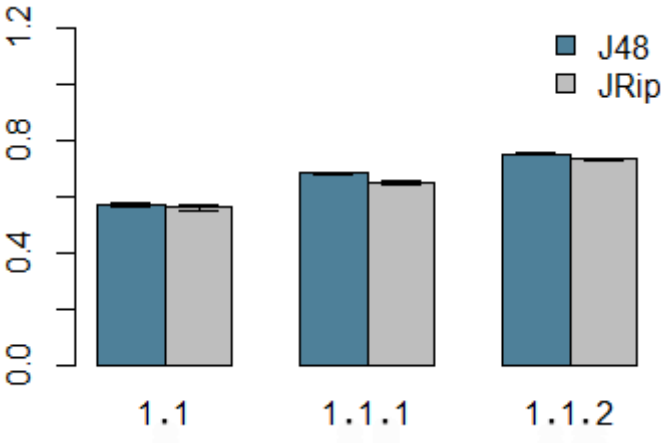

Figure 12. Fmeasure values for the partial dataset

By the time of writing this paper, our dataset had 4568 instances divided into five TE classes. Because of the dataset unbalance, we perfomed two sets of experiments: considering the original and the rebalanced partial dataset. The experiments showed that J48 is more robust to unbalanced data if compared to JRip, being more stable and more accurate. For the balanced partial dataset, both algorithms presented similar results.

The construction of the dataset presented in this work is still in progress. The final dataset should contain approximately 18600 sequences. Thus, improved results are expected with more instances in the dataset.

As future work, we plan to add sequences from other databases than PSGB. We are currently extracting sequences from Repbase [Bao et al. 2015], which contain a great number of repetitive sequences from many different genomes. We are also going perform experiments with other ML algorithms, to determine which one provides the best results. The use of ML algorithms for TEs classification according to [Wicker et al. 2007] taxonomy can be advantageous, allowing interpretable knowledge extraction in the form of classification rules. Homology based methods should also be used in our experiments, such as Blast [Altschul et al. 1997] and RepeatMasker [Smit et al. 2010].

Acknowledgment

The authors would like to thank CAPES, CNPq and FAPESP for their financial support, specially the grant #2015/14300-1 - São Paulo Research Foundation (FAPESP) to R.C. and grant PADRD-UFSCar to B.S.

References

Abrusán, G., Grundmann, N., DeMester, L., and Makalowski, W. (2009). Teclass - a tool for automated classification of unknown eukaryotic transposable elements. *Bioinformatics*, 25(10):1329–1330.

Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.*, 25(17):3389–3402.

Bao, W., Kojima, K. K., and Kohany, O. (2015). Repbase update, a database of repetitive elements in eukaryotic genomes. *Mob DNA*.

Bergman, C. M. and Quesneville, H. (2007). Discovering and detecting transposable elements in genome sequences. *Briefings in Bioinformatics*, 8(6):382–392.

Castro, E., Sigrist, C. J., Gattiker, A., Bulliard, V., Langendijk-Genevaux, P. S., Gasteiger, E., Bairoch, A., and Hulo, N. (2006). Scanprosite: detection of prosite signature matches and prorule-associated functional and structural residues in proteins. *Nucleic Acids Res.*

Cohen, W. W. (1995). Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123.

Costa, E., Schietgat, L., Cerri, R., Vens, C., Fischer, C., Carareto, C., Ramon, J., and Blockeel, H. (2013). Annotating transposable elements in the genome using relational decision tree ensembles. In *International Conference on Inductive Logic Programming*.

Feschotte, C., Keswani, U., Ranganathan, N., Guibotsy, M. L., and Levine, D. (2009). Exploring repetitive dna landscapes using repclass, a tool that automates the classification of transposable elements in eukaryotic genomes. *Genome Biology and Evolution*, 1:205–220.

Hoede, C., Arnoux, S., Moisset, M., Chaumier, T., Inizan, O., Jamilloux, V., and Quesneville, H. (2014). Pastec: An automatic transposable element classification tool. *PLoS ONE*, 9(5):e91929.

Hornik, K., Buchta, C., and Zeileis, A. (2009). Open-source machine learning: R meets Weka. *Computational Statistics*, 24(2):225–232.

Jones, P., Binns, D., Chang, H.-Y., Fraser, M., Li, W., McAnulla, C., McWilliam, H., Maslen, J., Mitchell, A., Nuka, G., Pesseat, S., Quinn, A. F., Sangrador-Vegas, A., Scheremetjew, M., Yong, S.-Y., Lopez, R., and Hunter, S. (2014). Interproscan 5: genome-scale protein function classification. *Bioinformatics (Oxford, England)*, 30(9):1236–1240.

Loureiro, T., Camacho, R., Vieira, J., and Fonseca, N. (2013). Boosting the detection of transposable elements using machine learning. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, volume 222, pages 85–91. Springer International Publishing.

Nussbaumer, T., Martis, M. M., Roessner, S. K., Pfeifer, M., Bader, K. C., Sharma, S., Gundlach, H., and Spannagl, M. (2013). Mips plantsdb: a database framework for comparative plant genome research. *Nucleic Acids Research*, 41(D1):D1144–D1151.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Rice, P., Longden, I., and Bleasby, A. (2000). Emboss: the european molecular biology open software suite. *Trends Genet.*

Sigrist, C. J. A., Castro, E., Cerutti, L., Cuche, B. A., Hulo, N., Bridge, A., Bougueleret, L., and Xenarios, I. (2012). New and continuing developments at prosite. *Nucleic Acids Res.*

Smit, A. F. A., Hubley, R., and Green, P. (1996-2010). RepeatMasker open-3.0.

Steinbiss, S., Willhoeft, U., Gremme, G., and Kurtz, S. (2009). Fine-grained annotation and classification of de novo predicted ltr retrotransposons. *Nucleic Acids Research*, 37(21):7002–7013.

Wan, S., Mak, M.-W., and Kung, S.-Y. (2012). mGOASVM: Multi-label protein subcellular localization based on gene ontology and support vector machines. *BMC Bioinformatics*, 13(1):290.

Wicker, T., Sabot, F., Hua-Van, A., Bennetzen, J. L., Capy, P., Chalhoub, B., Flavell, A., Leroy, P., Morgante, M., Panaud, O., Paux, E., SanMiguel, P., and Schulman, A. H. (2007). A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics*, 8(12):973–982.