Application Notes

repDNA: a Python package to generate various features of DNA sequences incorporating physicochemical properties and sequence-order effects

Bin Liu[1,2,3*], Fule Liu[1], Longyun Fang[1], Xiaolong Wang[1, 2]

[1]School of Computer Science and Technology and [2]Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong, China 518055; [3] Gordon Life Science Institute, Belmont, Massachusetts, USA 0478

Associate Editor: XXXXXXX

ABSTRACT

Summary: In order to develop powerful computational predictors for identifying the biological features or attributes of DNAs, one of the most challenging problems is to find a suitable approach to effectively represent the DNA sequences. To facilitate the studies of DNAs and nucleotides, we developed a Python package called representations of DNAs (repDNA) for generating the widely used features reflecting the physicochemical properties and sequence-order effects of DNAs and nucleotides. There are 3 feature groups composed of 15 features. The first group calculates 3 nucleic acid composition features describing the local sequence information by means of kmers; the second group calculates 6 autocorrelation features describing the level of correlation between two oligonucleotides along a DNA sequence in terms of their specific physicochemical properties; the third group calculates 6 pseudo nucleotide composition features, which can be used to represent a DNA sequence with a discrete model or vector yet still keep considerable sequence order information via the physicochemical properties of its constituent oligonucleotides. In addition, these features can be easily calculated based on both the built-in and user-defined properties via using repDNA.

Availability: The repDNA Python package is freely accessible to the public at http://bioinformatics.hitsz.edu.cn/repDNA/

Contact: bliu@insun.hit.edu.cn

Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

With the avalanche of DNA sequences generated in the post-genomic age, it is highly desirable to develop rapid and accurate computational methods for predicting their biological features or attributes. The features derived from DNA sequences have been widely used in the development of machine learning models for many computational methods, such as recombination spot identification (Chen, et al., 2013; Qiu, et al., 2014), prediction of nucleosome positioning in genomes (Guo, et al., 2014), investigation of nucleosome organization's functions (Chen, et al., 2010), promote prediction (Zhou, et al., 2013), etc.

A few web servers or programs for computing DNA features considering the sequence-order effects or structural information have been developed (Chen, et al., 2014). However, they are not comprehensive and can only be limited to a certain kind of features. Furthermore, most of them are not freely accessible, and the user-defined physicochemical properties cannot be used to compute the features.

In this study, we proposed an open source Python package called representations of DNAs (repDNA), which implemented a selection of sophisticated DNA features, including 15 different kinds of features in 3 categories. To our best knowledge, repDNA is the first Python package computing comprehensive DNA features based on the built-in and user-defined physicochemical properties. The repDNA package may hold very high potential for enhancing the power in dealing with many problems in computational genomics and genome sequence analysis.

2 PACKAGE DESCRIPTION

15 different features derived from DNA sequences can be computed by repDNA package, which can be grouped into 3 categories (**Table 1**). The first category nucleic acid composition includes three kinds of features: basic kmer, reverse compliment kmer, and increment of diversity. The nucleic acid composition features describe the local sequence information by means of kmers (subsequences of DNA sequences). The second category autocorrelation includes six kinds of features: dinucleotide-based auto covariance, dinucleotide-based cross covariance, dinucleotide-based auto-cross covariance, trinucleotide-based auto covariance, trinucleotide-based cross covariance, and trinucleotide-based auto-cross covariance. The auto-correlation features describe the level of correlation between two oligonucleotides along a DNA sequence in terms of their specific physicochemical properties. The third category pseudo nucleotide composition contains six kinds of features: pseudo dinucleotide composition, pseudo k-tupler nucleotide composition, parallel correlation pseudo dinucleotide composition, parallel correlation pseudo trinucleotide composition, series correlation pseudo dinucleotide composition, and series correlation pseudo trinucleotide composition. The pseudo nucleotide composition features can be used to represent a DNA sequence with a discrete model or vector yet still keep considerable sequence order information, particularly the global or long-range sequence order information, via the physicochemical properties of its constituent oligonucleotides. In the second and third categories, 38 dinucleotide physicochemical properties (**Table 1** in Online Supporting Information S1) and 12 trinucleotide physicochemical properties (**Table 2** in Online Supporting Information S1) have been used for calculating the corresponding features. Besides these built-in properties, the user-defined properties can also be used to calculate these features.

*To whom correspondence should be addressed.

Table 1. 15 feature vectors of DNA data calculated by repDNA.

Feature category	Features	Dimension[a]
Nucleic acid composition	Basic kmer	4^k
	Reverse compliment kmer	$\begin{cases} 2^{2k-1} (k = 1, 3, \cdots) \\ 2^{2k-1} + 2^{k-1} (k = 2, 4, \cdots) \end{cases}$
	Increment of diversity	$2k$
Autocorrelation	Dinucleotide-based auto covariance	$N * \mathrm{LAG}$
	Dinucleotide-based cross covariance	$N(N-1) * \mathrm{LAG}$
	Dinucleotide-based auto-cross covariance	$N^2 * \mathrm{LAG}$
	Trinucleotide-based auto covariance	$N * \mathrm{LAG}$
	Trinucleotide-based cross covariance	$N(N-1) * \mathrm{LAG}$
	Trinucleotide-based auto-cross covariance	$N^2 * \mathrm{LAG}$
Pseudo nucleotide composition	Pseudo dinucleotide composition	$16 + \lambda$
	Pseudo k-tupler nucleotide composition	$4^k + \lambda$
	Parallel correlation pseudo dinucleotide composition	$16 + \lambda$
	Parallel correlation pseudo trinucleotide composition	$64 + \lambda$
	Series correlation pseudo dinucleotide composition	$16 + \lambda N$
	Series correlation pseudo trinucleotide composition	$64 + \lambda N$

[a]The dimension of the feature vector depends on the parameter values of the algorithm and the number of physicochemical properties used, where k means the k value of kmer; N is the total number of physicochemical properties; LAG is the maximum value of *lag* (*lag* = 1, 2, …, LAG), where *lag* is the distance between two oligonucleotides along a DNA sequence; λ represents the highest counted rank (or tier) of the correlation along a DNA sequence. For more information of these parameters and physicochemical properties, please refer to Online Supporting Information S1.

There are four modules in the repDNA package, including util, nac, ac and psenac. The util module contains several basic functions manipulating DNA data, including reading DNA data from files or string lists (a data-structure in Python), checking the validity and normalizing the user-defined physicochemical indices, etc. The three modules nac, ac and psenac respond to the calculation of the 15 different features from three feature categories. In order to use the repDNA package to calculate these features as needed, the users need to import the appropriate class from the corresponding module, construct a responding object, and then call the corresponding methods to calculate these features. A user guide for how to use repDNA is given in Online Supporting Information S1.

As mentioned above, one of the main advantages of repDNA is that the user-defined physicochemical properties can be used to calculate the 12 features in autocorrelation category and pseudo nucleotide composition category. The user-defined properties should be normalized by normalize_index function in module util, and then the normalized properties will be stored in a dictionary (a data structure in Python). More conveniently, this dictionary can be directly used as the user-defined property to calculate the aforementioned features, which would benefit the users who want to calculate the features based on their own properties.

The repDNA was written by the pure Python language, which is a free, cross-platform language with a clean and uniform syntax. Furthermore, there are many public available Python packages of machine learning algorithms, such as scikit-learn (Pedregosa, et al., 2011). Therefore, it is convenient for users to construct their own predictors by using repDNA and these machine learning packages. Some examples of how to construct computational predictors for some specific tasks in computational genomics by using repDNA are given in Online Supporting Information S2.

3 CONCLUSION

In the field of computational genomics, more and more computational methods were based on the well-known machine learning techniques to construct their predictors, such as Support Vector Machines (SVMs), Random Forest (RF), Artificial Neural Networks (ANN), etc. All of these machine learning methods require fixed length feature vectors reflecting the characteristics of DNA sequences as inputs. Therefore, it is highly desired that a computational tool or package which can extract comprehensive features and convert the DNA sequences into fixed length vectors. To facilitate the studies of DNA and nucleotides, repDNA was proposed, which is a freely available Python package to generate various feature vectors of DNA sequences. The performance and efficiency of the various features in repDNA have been validated by a series of recent publications (Chen, et al., 2013; Chen, et al., 2014). The implementation of each algorithm in repDNA has been extensively tested by a large number of testing DNA sequences, and the output results were compared with the known values of these sequences to make sure that our implementation is correct. Furthermore, repDNA was tested with different operating systems (Mac, Linux, Unix and Windows) so as to meet the requirements of different users.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61300112).

Conflict of Interest: none declared.

REFERENCES

Chen, W., *et al.* (2013) iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition, *Nucleic Acids Res*, **41**, e68.

Chen, W., *et al.* (2014) PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition, *Anal Biochem*, **456**, 53-60.

Chen, W., Luo, L. and Zhang, L. (2010) The organization of nucleosomes around splice sites, *Nucleic Acids Res*, **38**, 2788-2798.

Guo, S.H., *et al.* (2014) iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition, *Bioinformatics*, **30**, 1522-1529.

Pedregosa, F., *et al.* (2011) Scikit-learn: Machine Learning in Python *Journal of Machine Learning Research*, **12**, 2825-2830.

Qiu, W.R., Xiao, X. and Chou, K.C. (2014) IRSpot–TNCPseAAC: identify recombination spots with trinucleotide composition and pseudo amino acid components, *Int. J. Mol. Sci.*, **15**, 1746–1766.

Zhou, X., *et al.* (2013) Predicting promoters by pseudo-trinucleotide compositions based on discrete wavelets transform, *J. Theor. Biol.*, **319**, 1-7.