Bioinformatics, 31(8), 2015, 1307–1309
doi: 10.1093/bioinformatics/btu820
Advance Access Publication Date: 10 December 2014
Applications Note

Sequence analysis

repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects

Bin Liu[1,2,3,*], Fule Liu[1], Longyun Fang[1], Xiaolong Wang[1,2] and Kuo-Chen Chou[3,4,*]

[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 518055, China, [3]Gordon Life Science Institute, Belmont, MA 02478, USA and [4]Center of Excellence in Genomic Medicine Research (CEGMR), King Abdulaziz University, Jeddah, 21589, Saudi Arabia

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on October 1, 2014; revised on November 5, 2014; accepted on December 5, 2014

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 three feature groups composed of 15 features. The first group calculates three nucleic acid composition features describing the local sequence information by means of kmers; the second group calculates six 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 six 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 and implementation: The repDNA Python package is freely accessible to the public at http://bioinformatics.hitsz.edu.cn/repDNA/.

Contact: bliu@insun.hit.edu.cn or kcchou@gordonlifescience.org

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

1 Introduction

With the avalanche of biological sequences generated in the post-genomic age, one of the most challenging problems in computational biology is how to formulate a biological sequence with a discrete model or vector, yet still keep considerable sequence-order information. This is because almost all the existing machine-learning algorithms were developed to handle vector but not sequence samples. However, a vector defined in a discrete model may completely lose all the sequence-order information. To avoid completely losing the sequence-order information for proteins, the pseudo amino acid composition or PseAAC (Chou, 2001, 2005) was propose. Ever since the concept of PseAAC was proposed in 2001, it has been widely used in almost all the areas of computational proteomics [see e.g. (Cao *et al.*, 2013) as well as a long list of references cited in a recent article (Du *et al.*, 2014)]. Encouraged by the successes of using PseAAC to deal with protein/peptide sequences, a natural question has occurred: how to use the similar approach to deal with DNA sequences? Actually, this problem had been encountered in various genome analysis studies, such as DNA 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) and promoter prediction (Zhou *et al.*, 2013; Lin *et al.*, 2014).

Since various features derived from DNA sequences have been increasingly used for developing different models to analyze many genome analysis problems, recently a web server called pseudo k-tupler nucleotide composition (PseKNC) (Chen *et al.*, 2014) was established to generate PseKNC. However, PseKNC is limited to a certain kind of features without the function of user-defined physicochemical properties.

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 three 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

Fifteen different features derived from DNA sequences can be computed by repDNA package, which can be grouped into three categories (Table 1). The first category nucleic acid composition includes three kinds of features: basic kmer, reverse compliment kmer and increment of diversity (ID). 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 (DAC), dinucleotide-based cross covariance (DCC), dinucleotide-based auto-cross covariance (DACC), trinucleotide-based auto covariance (TAC), trinucleotide-based cross covariance (TCC) and trinucleotide-based auto-cross covariance (TACC). The autocorrelation 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 (PseDNC), PseKNC, parallel correlation PseDNC (PC-PseDNC), PC-pseudo trinucleotide composition (PC-PseTNC), series correlation PseDNC (SC-PseDNC) and series correlation PseTNC (SC-PseTNC). 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,

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

Category	Feature	Dimension[a]	Description
Nucleic acid composition	Basic kmer	4^k	*k*-tuple nucleotide composition
	Reverse compliment kmer	$\begin{cases} 2^{2k-1}(k=1,3,\ldots) \\ 2^{2k-1}+2^{k-1}(k=2,4,\ldots) \end{cases}$	A variant of the basic kmer, in which the kmers are not expected to be strand-specific, so reverse complementary are collapsed into a single feature
	ID	$2k$	Measuring the relation between target sequence and standard source based on kmers
Autocorrelation	DAC	N*LAG	Incorporating the correlation of the same property between two dinucleotides
	DCC	$N(N-1)$*LAG	Incorporating the correlation of the different properties between two dinucleotides
	DACC	N^2*LAG	Combination of DAC and DCC
	TAC	N*LAG	Incorporating the correlation of the same property between two trinucleotides
	TCC	$N(N-1)$*LAG	Incorporating the correlation of the different properties between two trinucleotides
	TACC	N^2*LAG	Combination of TAC and TCC
Pseudo nucletide composition	PseDNC	$16+\lambda$	Combining dinucleotide composition and global sequence-order effects
	PseKNC	$4^k+\lambda$	Improving PseDNC by incorporating *k*-tuple nucleotide composition
	PC-PseDNC	$16+\lambda$	Improving PseDNC by incorporating 38 built-in properties and user-defined properties
	PC-PseTNC	$64+\lambda$	Combining trinucleotide composition and global sequence-order effects by parallel correlation
	SC-PseDNC	$16+\lambda N$	Combining dinucleotide composition and global sequence-order effects by series correlation
	SC-PseTNC	$64+\lambda N$	Combining trinucleotide composition and global sequence-order effects by series correlation

[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 algorithms, parameters and physicochemical properties, please refer to online Supplementary information S1.

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 and 12 trinucleotide physicochemical properties 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.

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 list (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 Supplementary information S1.

As mentioned earlier, 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), which can be directly used as the user-defined property to calculate the aforementioned features.

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. 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 by using repDNA are given in online Supplementary information S2.

3 Conclusion

To facilitate the studies of DNA and nucleotides, repDNA was proposed, which is able to generate various feature vectors for 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, 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.

Funding

This work was supported by the National Natural Science Foundation of China [grant no 61300112, 61370010 and 61272383] and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

Conflict of Interest: none declared.

References

Cao,D.S. *et al.* (2013) Propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics*, **29**, 960–962.

Chen,W. *et al.* (2010) The organization of nucleosomes around splice sites. *Nucleic acids Res.*, **38**, 2788–2798.

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.

Chou,K.C. (2001) Prediction of protein cellular attributes using pseudo amino acid composition. *Proteins*, **43**, 246–255. See Erratum: ibid., (2001). *Proteins*, **44**, 60.

Chou,K.C. (2005) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics*, **21**, 10–19.

Du,P. *et al.* (2014) PseAAC-General: fast building various modes of general form of Chou's pseudo-amino acid composition for large-scale protein datasets. *Int. J. Mol. Sci.*, **15**, 3495–3506.

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.

Lin,H. *et al.* (2014) iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acids Res.*, **42**, 12961–12972.

Qiu,W.R. *et al.* (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.