*Bioinformatics*, 33(1), 2017, 35–41
doi: 10.1093/bioinformatics/btw539
Original Paper

OXFORD

Sequence analysis

# iRSpot-EL: identify recombination spots with an ensemble learning approach

**Bin Liu[1,2,3,*], Shanyi Wang[1], Ren Long[1] and Kuo-Chen Chou[3,4]**

[1]School of Computer Science and Technology, [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 April 1, 2016; revised on August 1, 2016; accepted on August 11, 2016

## Abstract

**Motivation:** Coexisting in a DNA system, meiosis and recombination are two indispensible aspects for cell reproduction and growth. With the avalanche of genome sequences emerging in the post-genomic age, it is an urgent challenge to acquire the information of DNA recombination spots because it can timely provide very useful insights into the mechanism of meiotic recombination and the process of genome evolution.

**Results:** To address such a challenge, we have developed a predictor, called **iRSpot-EL**, by fusing different modes of pseudo K-tuple nucleotide composition and mode of dinucleotide-based auto-cross covariance into an ensemble classifier of clustering approach. Five-fold cross tests on a widely used benchmark dataset have indicated that the new predictor remarkably outperforms its existing counterparts. Particularly, far beyond their reach, the new predictor can be easily used to conduct the genome-wide analysis and the results obtained are quite consistent with the experimental map.

**Availability and Implementation:** For the convenience of most experimental scientists, a user-friendly web-server for **iRSpot-EL** has been established at http://bioinformatics.hitsz.edu.cn/iRSpot-EL/, by which users can easily obtain their desired results without the need to go through the complicated mathematical equations involved.

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

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

## 1 Introduction

Recombination plays an important role in genetic evolution, which describes the exchange of genetic information during the period of each generation in diploid organisms. Recombination provides many new combinations of genetic variations and is an important source for biodiversity, which can accelerate the procedure of biological evolution. Knowledge of recombination spots may also provide very useful information for in-depth understanding the reproduction and growth of cells. Therefore, it is highly demanded to develop computational methods for predicting the recombination spots.

Actually, many efforts have been made in this regard. For instance, based on the gapped dinucleotide composition features, Jiang *et al.* (2007) developed a predictor called RF-DYMHC to do the job. Liu *et al.* (2012), using the kmer approach and the increment of diversity combined with quadratic discriminant analysis, developed the IDQD predictor for the same purpose. In the above two predictors, however, only the local DNA sequence information was utilized, and hence their prediction quality may be limited. To improve this situation, recently two new predictors, iRSpot-PseDNC (Chen *et al.*, 2013) and iRSpot-TNCPseAAC (Qiu *et al.*, 2014) were developed. The former was based on the DNA local structural

properties (Chen *et al.*, 2012) and pseudo dinucleotide composition (Chen *et al.*, 2014); while the latter based on the DNA trinucleotide composition (Chen *et al.*, 2014) as well as the corresponding pseudo amino acid components (Chou, 2001).

Each of the aforementioned methods has its own advantage, and did play a role in stimulating the development of this important area. Meanwhile, they also have some disadvantages, as reflected by the following facts. (i) Although powerful predictors have been proposed, there is no efficient approach to combine them to further improve the predictive performance. (ii) None of these methods allows users to set the desired parameters for prediction, and hence it is difficult for them to optimize the predictor system according to the need of their focus. (iii) Except the RF-DYMHC (Jiang *et al.*, 2007), all the other predictors cannot be directly used for genome-wide analysis. Even for the RF-DYMHC predictor, its approach is not accurate because the window size therein is arbitrary.

This study was initiated in an attempt to address these shortcomings by developing a more powerful predictor for identifying DNA recombination spots. The proposed predictor is called **iRSpot-EL**, where 'i' stands for 'identify', 'RSpot' for 'recombination spot' and 'EL' for 'ensemble learning'.

To develop a new predictor usually consists of two purposes. One is to stimulate theoretical studies in the relevant areas, and the other is to make experimental scientists easier to get their desired information. To realize these, the rest of this article is presented according to the following five guidelines (Chou, 2011): (i) benchmark dataset, (ii) sample representation, (iii) operation algorithm, (iv) validation, and (v) web-server.

## 2 Materials and methods

### 2.1 Benchmark dataset

A reliable and stringent benchmark is pivotal to the development of an accurate prediction method. In literature, the benchmark dataset usually consists of a training dataset and a testing dataset: the former is for the purpose of training a proposed model, while the latter for the purpose of testing it. As pointed out by a comprehensive review (Chou and Shen, 2007b), however, there is no need to separate a benchmark dataset into a training dataset and a testing dataset for validating a prediction method if it is tested by the jackknife or subsampling (K-fold) cross-validation because the outcome thus obtained is actually from a combination of many different independent dataset tests. In this study, for facilitating the comparison of the proposed predictor with the existing ones, we adopted the widely used benchmark dataset (Chen *et al.*, 2013; Jiang *et al.*, 2007; Liu *et al.*, 2012; Qiu *et al.*, 2014) that can be formulated as

$$\mathbb{S} = \mathbb{S}^+ \cup \mathbb{S}^- \tag{1}$$

where $\mathbb{S}$ is the benchmark dataset, $\mathbb{S}^+$ the positive subset containing 490 DNA segments (hotspot samples) with the relative hybridization ratios (Gerton *et al.*, 2000) higher than 1.5 (Jiang *et al.*, 2007), $\mathbb{S}^-$ the negative subset containing 591 DNA segments (coldspot samples) with the relative hybridization ratios (Gerton *et al.*, 2000) lower than 0.82 (Jiang *et al.*, 2007), and the symbol $\cup$ denotes the union in the set theory. In order to reduce redundancy and homology bias, the CD-HIT software (Li *et al.*, 2001) was used to remove sequences whose similarity is >75%. Finally, 478 hotspots (positive samples) and 572 coldspots (negative samples) were obtained. For readers' convenience, the 478 hotspot samples and 572 coldspot samples as well as their detailed sequences are given in Supplementary Materials S1.

### 2.2 Pseudo k-tuple nucleotide composition

With the avalanche of biological sequences emerging in the post-genomic age, one of the most challenging problems in computational biology is how to formulate a biological sequence with a vector, yet essentially still keep its key pattern or characteristics. This is because nearly all the existing machine-learning algorithms were developed to handle vector but not sequence samples, as elaborated in a recent review (Chou, 2015). Unfortunately, a vector defined in a discrete model may completely lose all the sequence-order information or sequence pattern characteristics. To overcome such a problem for protein/peptide sequences, the pseudo amino acid composition (PseAAC) (Chou, 2001) was introduced, and has become an important tool (Cao *et al.*, 2013; Du *et al.*, 2012, 2014) widely used in nearly all the areas of computational proteomics [see a long list of references cited in Chou (2011)]. Encouraged by the successes of PseAAC, the pseudo nucleotide composition (PseKNC) (Chen *et al.*, 2014, 2015b; Liu *et al.*, 2015a, 2016b) was introduced to formulate DNA/RNA sequences, and it has been increasingly used in computational genetics and genomics (see, e.g. a recent review (Chen *et al.*, 2015a) as well as a long list of references cited therein). Recently, a web-server called 'Pse-in-One' was developed for generating various modes of pseudo components for DNA/RNA and protein/peptide sequences (Liu *et al.*, 2015b).

Here the concept of PseKNC was used to define the feature vectors for identifying recombination spots via 15 indices (Table 1) of local DNA structural properties, which were selected from (Friedel *et al.*, 2009). Note that PseKNC model contains three uncertain parameters: $k$ is the number of neighboring nucleic acid residues; $\lambda$ is the highest ranks or tiers (Chou, 2005); $w$ is the weight factor. These three parameters will be discussed in the Ensemble Learning Section.

### 2.3 Dinucleotide-based auto-cross covariance

In this study, the DNA sequences were generated by a very special mode of PseKNC (Liu *et al.*, 2015b), the so-called dinucleotide-based auto-cross covariance (DACC) approach, which is a combination of dinucleotide-based auto covariance (DAC) and dinucleotide-based cross covariance (DCC). The former is based on a same physicochemical property listed in Table 1; while the latter, based on two different ones. Note that there is one shift parameter *lag* in the DACC, as will be discussed later.

### 2.4 Support vector machine

Support vector machine (SVM) (Suykens and Vandewalle, 1999) is an efficient supervised learning approach in the field of machine learning, and has been widely used for classification and regress analysis. The basic idea of SVM is to transform the input data into a high dimensional feature space and then determine the optimal separating hyperplane. For more details about SVM, see Cristianini and Shawe-Taylor (2000) and Vapnik (1999).

In this study, the LIBSVM package (Chang and Lin, 2001) with RBF kernel was used to implement SVM, in which there are two parameters: one is the regularization parameter $C$, and the other is the kernel width parameter $\gamma$. Thus, there are a total of five uncertain parameters when using SVM on the PseKNC model, while three uncertain parameters on the DACC model. All these parameters were optimized on the validation sets

### 2.5 Ensemble learning

As demonstrated by a series of previous studies, such as protein fold pattern recognition (Shen and Chou, 2006), membrane protein type

**Table 1.** The values of the fifteen DNA dinucleotide properties

| Structural index | AA/TT | AC/GT | AG/CT | AT | CA/TG | CC/GG | CG | GA/TC | GC | TA |
|---|---|---|---|---|---|---|---|---|---|---|
| F-roll | 0.04 | 0.06 | 0.04 | 0.05 | 0.04 | 0.04 | 0.04 | 0.05 | 0.05 | 0.03 |
| F-tilt | 0.08 | 0.07 | 0.06 | 0.10 | 0.06 | 0.06 | 0.06 | 0.07 | 0.07 | 0.07 |
| F-twist | 0.07 | 0.06 | 0.05 | 0.07 | 0.05 | 0.06 | 0.05 | 0.06 | 0.06 | 0.05 |
| F-slide | 6.69 | 6.80 | 3.47 | 9.61 | 2.00 | 2.99 | 2.71 | 4.27 | 4.21 | 1.85 |
| F-shift | 6.24 | 2.91 | 2.80 | 4.66 | 2.88 | 2.67 | 3.02 | 3.58 | 2.66 | 4.11 |
| F-rise | 21.34 | 21.98 | 17.48 | 24.79 | 14.51 | 14.25 | 14.66 | 18.41 | 17.31 | 14.24 |
| Roll | 1.05 | 2.01 | 3.60 | 0.61 | 5.60 | 4.68 | 6.02 | 2.44 | 1.70 | 3.50 |
| Tilt | −1.26 | 0.33 | −1.66 | 0.00 | 0.14 | −0.77 | 0.00 | 1.44 | 0.00 | 0.00 |
| twist | 35.02 | 31.53 | 32.29 | 30.72 | 35.43 | 33.54 | 33.67 | 35.67 | 34.07 | 36.94 |
| Slide | −0.18 | −0.59 | −0.22 | −0.68 | 0.48 | −0.17 | 0.44 | −0.05 | −0.19 | 0.04 |
| Shift | 0.01 | −0.02 | −0.02 | 0.00 | 0.01 | 0.03 | 0.00 | −0.01 | 0.00 | 0.00 |
| Rise | 3.25 | 3.24 | 3.32 | 3.21 | 3.37 | 3.36 | 3.29 | 3.30 | 3.27 | 3.39 |
| Energy | −1.00 | −1.44 | −1.28 | −0.88 | −1.45 | −1.84 | −2.17 | −1.30 | −2.24 | −0.58 |
| Enthalpy | −7.60 | −8.40 | −7.80 | −7.20 | −8.50 | −8.00 | −10.60 | −8.20 | −9.80 | −7.20 |
| Entropy | −21.30 | −22.40 | −21.00 | −20.40 | −22.70 | −19.90 | −27.20 | −2.20 | −24.40 | −21.30 |

classification (Chou and Shen, 2007a), signal peptide prediction (Shen and Chou, 2007a), protein subcellular location prediction (Chou and Shen, 2008), enzyme functional classification (Shen and Chou, 2007b), identifying phosphorylation sites (Qiu *et al.*, 2016b) and multiple lysine PTM sites in proteins (Qiu *et al.*, 2016a), the ensemble predictor formed by fusing an array of individual predictors via a voting system can yield much better prediction quality.

There are two main components in the ensemble learning framework: (i) How to select the basic classifiers? (ii) How to ensemble the basic classifiers so as to make the final prediction? In order to select the representative basic classifiers, the distance between any two classifiers $\mathbb{C}(i)$ and $\mathbb{C}(j)$ was measured by the following equation considering both the diversity and complementarity of the classifiers:

$$\text{Distance}(\mathbb{C}(i),\ \mathbb{C}(j)) = 1 - \frac{1}{2m}\sum_{k=1}^{m}(d_{ik}\Delta d_{jk}) \tag{2}$$

where $m$ represents the number of training samples, $d_{ik}$ represents the misclassification probability of classifier $\mathbb{C}(i)$ on the $k$th sample, and $d_{ik}\Delta d_{jk}$ can be calculated by:

$$d_{ik}\Delta d_{jk} = \begin{cases} d_{ik} + d_{jk}, & \text{if } \mathbb{C}(i) \text{ and } \mathbb{C}(j) \text{ incorrectly predicts the } k\text{th sample} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The range of the distance defined in Equation (2) is from 0 to 1, where a distance of 1 indicates the predictive results of two classifiers are completely complementary, and 0 means that their results are identical. Based on the distance, the affinity propagation clustering algorithm (Frey and Dueck, 2007) was employed, which is quite suitable for the current task since the center clusters are not required in this algorithm.

For the PseKNC (Chen *et al.*, 2014), different values of $\lambda$, $k$ and $w$ will correspond to different input types. In the present study, 500 different PseKNC classifiers were constructed by using the following parameter combinations:

$$\begin{cases} 2 \le k \le 6 & \text{with step } \Delta = 1 \\ 0 \le w \le 1 & \text{with step } \Delta = 0.1 \\ 1 \le \lambda \le 10 & \text{with step } \Delta = 1 \end{cases} \tag{4}$$

Likewise, 10 different DACC classifiers were generated with different values of *lag* ($lag = 1, 2, \ldots, 10$). By using the aforementioned methods, 510 different classifiers were obtained, which were then clustered into seven clusters by using the affinity propagation clustering (Frey and Dueck, 2007). For each cluster, the top performing one was selected. For this study, the ensemble classifier can be formulated by (see Table 2)

$$\mathbb{C}^{\mathrm{E}} = \mathbb{C}(1)\forall\mathbb{C}(2)\mathbb{C}(3)\forall\mathbb{C}(4)\forall\mathbb{C}(5)\forall\mathbb{C}(6)\forall\mathbb{C}(7) = \forall_{i=1}^{7}\mathbb{C}(i) \tag{5}$$

where $\mathbb{C}^{\mathrm{E}}$ denotes the ensemble classifier, the symbol $\forall$ denotes the fusing operator (Chou and Shen, 2007b), and the fusion was operated via the following fractional votes

$$Y = \frac{1}{7}\sum_{i=1}^{7} F_i P_i \tag{6}$$

where $P_i$ denotes the probability from the classifier $\mathbb{C}(i)$, and $F_i$ its fraction used, which was optimized on the validation sets (see Table 2). If $Y > 0.5$, the sample is predicted as a hotspot; otherwise, coldspot.

For more detailed about the process of fusing individual basic classifiers into an ensemble classifier, see a comprehensive review (Chou and Shen, 2007b) where a crystal clear elucidation with a set of elegant equations are given and hence there is no need to repeat here.

The flowchart of ensemble strategy on different clustering is given in Figure 1.

## 2.6 Cross-validation

Three cross-validation methods are often used in literature; they are independent dataset test, K-fold cross-validation test, and jackknife test (Chou and Zhang, 1995).

In this study, the 5-fold cross-validation was used. The benchmark dataset was randomly divided into five subsets with an approximately equal number of samples. Each predictor runs five times with five different training and test sets. For each run, three sets were used to train the

predictor, one set was used as the validation set to optimize the parameters, and the remaining one was used as the test set to give the final results.

## 2.7 Metrics used to reflect the success rates

For a binary classification system such as the one in this study, the following set of four metrics are often used to quantitatively

**Table 2.** List of the seven basic classifiers selected by using affinity ropagation clustering algorithm

| Basic classifier | Feature | Dimension | Fraction |
|---|---|---|---|
| $\mathbb{C}(1)$ | PseKNC[a] | 20 | 0.25 |
| $\mathbb{C}(2)$ | PseKNC[b] | 22 | 0.05 |
| $\mathbb{C}(3)$ | PseKNC[c] | 26 | 0.10 |
| $\mathbb{C}(4)$ | PseKNC[d] | 26 | 0.00 |
| $\mathbb{C}(5)$ | PseKNC[e] | 67 | 0.05 |
| $\mathbb{C}(6)$ | PseKNC[f] | 72 | 0.05 |
| $\mathbb{C}(7)$ | DACC[g] | 1125 | 0.50 |

[a]The optimal parameters were $k=2$, $\lambda=4$, $w=0.5$.
[b]The optimal parameters were $k=2$, $\lambda=6$, $w=0.8$.
[c]The optimal parameters were $k=2$, $\lambda=10$, $w=0.9$.
[d]The optimal parameters were $k=2$, $\lambda=10$, $w=1.0$.
[e]The optimal parameters were $k=3$, $\lambda=3$, $w=0.8$.
[f]The optimal parameters were $k=3$, $\lambda=8$, $w=0.9$.
[g]The optimal parameter was $lag=5$.

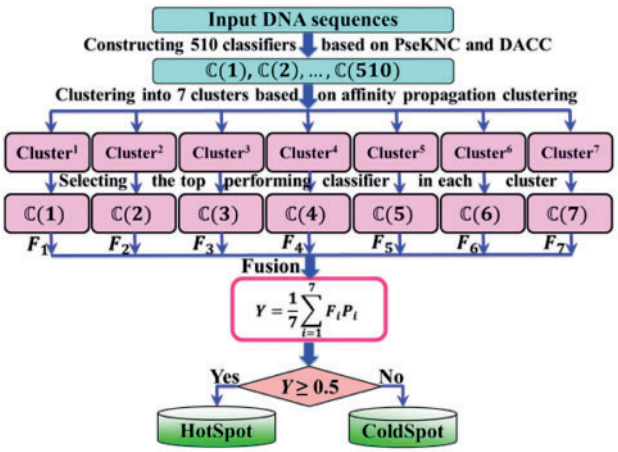


**Fig. 1.** A flowchart to show how the iRSpot-EL predictor works

measure the quality of a predictor (see, e.g. Guo *et al.*, 2014; Jia *et al.*, 2016; Liu *et al.*, 2016c; Qiu *et al.*, 2016a)

$$\begin{cases} \mathrm{Sn}=1-\dfrac{N_-^+}{N^+} & 0 \le \mathrm{Sn} \le 1 \\ \mathrm{Sn}=1-\dfrac{N_+^-}{N^-} & 0 \le \mathrm{Sp} \le 1 \\ \mathrm{Acc}=1-\dfrac{N_-^+ + N_+^-}{N^+ + N^-} & 0 \le \mathrm{Acc} \le 1 \\ \mathrm{MCC}=\dfrac{1-\left(\dfrac{N_-^+}{N^+}+\dfrac{N_+^-}{N^-}\right)}{\sqrt{\left(1+\dfrac{N_+^- - N_-^+}{N^+}\right)\left(1+\dfrac{N_-^+ - N_+^-}{N^-}\right)}} & -1 \le \mathrm{MCC} \le 1 \end{cases} \quad (7)$$

where Sn, Sp, Acc and MCC represent sensitivity, specificity, overall accuracy, and Mathew's correlation coefficient, respectively (Chen *et al.*, 2007). The total numbers of recombination hotspots and coldspots are denoted by $N^+$ and $N^-$, respectively. The number of hotspot samples incorrectly predicted to be of coldspot is denoted by $N_-^+$, while the number of coldspot samples incorrectly predicted to be of hotspot is by $N_+^-$. As for the meanings of the four metrics in Equation (7) along with their score regions, see Lin *et al.* (2014) where a clear and incisive analysis has been elaborated and hence there is no need to repeat here.

### 2.8 *F*-score

The *F*-score can be calculated by using the following equation:

$$F_i = \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n^+-1}\sum_{k=1}^{n^+}(x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n^--1}\sum_{k=1}^{n^-}(x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2} \quad (8)$$

where $n^+$ stands for the total number of the positive samples, $n^-$ for the total number of the negative samples, $\bar{x}_i^{(+)}$ for the mean value of the $i$th feature of entire positive samples, $\bar{x}_i^{(-)}$ for the mean value of $i$th feature of entire negative samples, $\bar{x}_i$ for the mean value of the $i$th feature of the total samples. $x_{k,i}^{(+)}$ for the value of the $i$th feature of the $k$th sample in the positive data set, and $x_{k,i}^{(-)}$ for the value of the $i$th feature of the $k$th sample in the negative data set. The larger the *F*-score is, the more important the feature is (Akay, 2009).

## 3 Results and discussion

### 3.1 Comparison with basic methods and existing methods

Listed in Table 3 are the 5-fold cross-validation results by **iRSpot-EL** on the benchmark dataset of Equation (1) (see Supplementary Material S1). For facilitating comparison, listed in that table and Figure 2 are also the corresponding results obtained by the RF-DYMHC predictor (Jiang *et al.*, 2007), IDQD predictor (Liu *et al.*, 2012), iRSpot-PseDNC predictor (Chen *et al.*, 2013) and iRSpot-TNCPseAAC (Qiu *et al.*, 2014).

From the table, we can see the following. (i) Among the five predictors the newly proposed one achieved the highest success rates in both Acc and MCC, the two most important metrics used to measure the quality of a predictor as elucidated in the follow-up text to Equation (7). (ii) Although the Sn rate by the proposed predictor was about 4% lower than that by **IDQD**, its Sp rate was about 7% higher than that by **IDQD**. As mentioned in Section 2.7, the two metrics are used to measure a predictor from two opposite angles, and they are constrained with each other. Therefore, it is meaningless to use only one of the two for comparing the quality of two predictors. In other words, a meaningful comparison in this regard should count the rates of both Sn and Sp, or even better, the rate of their combination that is none but MCC. As shown in Table 3, the MCC rate achieved by the proposed predictor **iRSpot-EL** is higher than other existing predictors by about 3.5–17.7%.

### 3.2 Feature analysis

In order to further investigate the discriminant power of different features and basic classifiers, the *F*-score method (Lin *et al.*, 2014) was adopted to analyze the seven basic classifiers listed in Table 2.

The top 10 most important features for each basic classifier are listed in Table 4, from which we can see that the important features between PseKNC and DACC classifiers are different, indicating that these classifiers are mutually complementary. Therefore, performance improvement can be observed by combining these classifiers via an ensemble learning approach. Some common patterns can also be observed, for examples, CG, AT, TA, GC are very important for all the six PseKNC classifiers, which is fully consistent with Jiang *et al* (2011) study.

### 3.3 Performance on analysis of the whole genome

To further demonstrate its practical application, the genome-wide analysis by **iRSpot-EL** was performed on the yeast chromosome III.

**Table 3.** List of the metrics scores (cf. Eq.7) obtained by various methods via 5-fold cross-validation on the same benchmark dataset of Supporting Information S1

| Methods | Sn(%)[f] | Sp(%)[f] | Acc(%)[f] | MCC[f] | AUC[g] |
|---|---|---|---|---|---|
| RF-DYMHC[a] | 73.01 | 86.56 | 80.40 | 0.6049 | 0.8777 |
| IDQD[b] | 79.52 | 81.82 | 80.77 | 0.6160 | 0.8822 |
| iRSpot-PseDNC[c] | 71.75 | 85.84 | 79.33 | 0.5830 | 0.8631 |
| iRSpot-TNCPseAAC[d] | 76.56 | 70.99 | 73.52 | 0.4737 | 0.8138 |
| iRSpot-EL[e] | 75.29 | 88.81 | 82.65 | 0.6510 | 0.8922 |

[a]The predictor reported in (Jiang et al., 2007).
[b]The predictor reported in (Liu et al., 2012).
[c]The predictor reported in (Chen et al., 2013).
[d]The predictor reported in (Qiu et al., 2014).
[e]The proposed predictor in this article.
[f]See Equation (7) for the metrics definition.
[g]See Figure 2 and its legend.

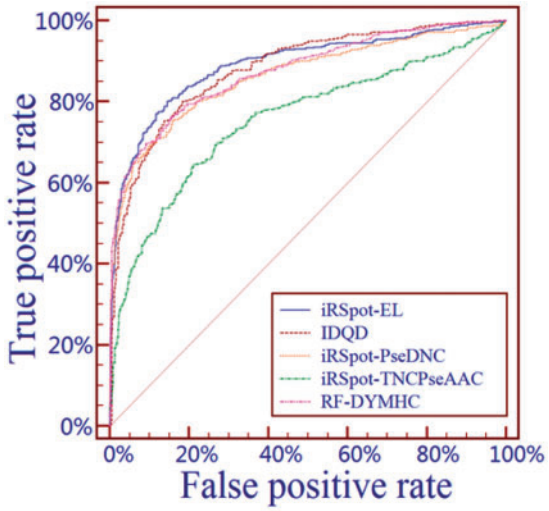


**Fig. 2.** The ROC (receiver operating characteristic) curves obtained by different methods. The area under the ROC curves is called AUC. They are 0.8922, 0.8138, 0.8822, 0.8631 and 0.8777 for iRSpot-EL, iRSpot-TNCPseAAC, IDQD, iRSpot-PseDNC and RF-DYMHC, respectively. The larger the AUC, the better the corresponding predictor is (Davis and Goadrich, 2006; Fawcett, 2005)

In order to avoid the homology redundancy bias, the CD-HIT software (version 4.6) (Li *et al.*, 2001) was used to remove those DNA sequences from the benchmark dataset that have >75% sequence identity to the 1 kb length DNA fragments in chromosome III. Trained with such a reduced benchmark dataset, the **iRSpot-EL** predictor was used to identify the hotspots in chromosome III with reliability index value set as 6 as suggested by (Jiang *et al.*, 2007). For investigation into the effects of different parameters on the predictive performance, the genome-wide prediction was conducted with different sliding windows and step sizes. The predicted results of the center position were smoothed by using the average value of 200-bp in a sliding window. The results predicted by **iRSpot-EL** on yeast chromosome III are given in Figure 3, where for facilitating the comparison the corresponding recombination profile by experiments (Mancera *et al.*, 2008) is also given. It can be clearly seen that the recombination profile predicted by **iRSpot-EL** is highly consistent with that of experimental observations (Mancera *et al.*, 2008), further demonstrating that **iRSpot-EL** is indeed a very useful high-throughput tool for genome-wide analysis of recombination spots. Interestingly, we have also observed that the cases with lager sliding window sizes tend to show better results. The reason is that larger window sizes can incorporate more global sequence information, which is critical for improving the performance (Liu *et al.*, 2016a). Another important observation is that the step size has little impact on the predictive performance. Based on the aforementioned experimental outcomes, we suggest the users to set the parameters of sliding window size and its step size as 2000 and 200 bp, respectively, for the genome-wide analysis when using **iRSpot-EL**.

### 3.4 Web server and user guide

As pointed out in two recent review papers (Chen *et al.*, 2015b; Chou, 2015), a prediction method with its web-server available will attract more users. In view of this, the web-server for **iRSpot-EL** has been established. Moreover, to maximize the convenience for users, a step-by-step guide is provided below.

**Step 1.** Open the web server by clicking the link at http://bioinformatics.hitsz.edu.cn/iRSpot-EL/and you will see the home page of **iRSpot-EL**. Click on the ReadMe button to see a brief introduction about the server.

**Step 2.** Click on the Server button. Either type or copy/paste the query DNA sequence into the input box. You can also upload your

**Table 4.** List of the top 10 important features in the basic classifiers

| Nos. | PseKNC[a] | PseKNC[b] | PseKNC[c] | PseKNC[d] | PseKNC[e] | PseKNC[f] | DACC[g] |
|---|---|---|---|---|---|---|---|
| 1 | CG | CG | CG | CG | GCC | GCC | DAC($lag = 2$, F-tilt) |
| 2 | AT | AT | AT | AT | AAT | AAT | DCC($lag = 1$, F-shift, Shift) |
| 3 | TA | TA | TA | TA | TTA | TTA | DCC($lag = 1$, Energy, Shift) |
| 4 | GC | GC | GC | GC | CGC | CGC | DCC($lag = 1$, F-tilt, Shift) |
| 5 | CC | CC | CC | CC | TAA | TAA | DAC($lag = 1$, F-shift) |
| 6 | AA | AA | AA | AA | ATT | ATT | DCC($lag = 1$, Shift, F-shift) |
| 7 | AC | AC | AC | AC | CGG | CGG | DCC($lag = 1$, Shift, Energy) |
| 8 | CA | CA | CA | CA | CCG | CCG | DCC($lag = 1$, Roll, F-tilt) |
| 9 | TT | $\lambda = 6$ | TT | TT | ACG | ACG | DCC($lag = 1$, F-shift, F-tilt) |
| 10 | GG | TT | GG | GG | GGC | GGC | DCC($lag = 1$, F-tilt, F-shift) |

[a]Parameters were $k = 2$, $\lambda = 4$, $w = 0.5$, $C = 2^{15}$, and $\gamma = 2$.
[b]Parameters were $k = 2$, $\lambda = 6$, $w = 0.8$, $C = 2^{15}$, and $\gamma = 2^3$.
[c]Parameters were $k = 2$, $\lambda = 10$, $w = 0.9$, $C = 2^{15}$, and $\gamma = 2^3$.
[d]Parameters were $k = 2$, $\lambda = 10$, $w = 1.0$, $C = 2^{15}$ and $\gamma = 2^3$.
[e]Parameters were $k = 3$, $\lambda = 3$, $w = 0.8$, $C = 2^{13}$, and $\gamma = 2^3$.
[f]Parameters were $k = 3$, $\lambda = 8$, $w = 0.9$, $C = 2^{13}$, and $\gamma = 2^3$.
[g]Parameters were $lag = 5$, $C = 2^5$, and $\gamma = 2^{-5}$. The values of DNA dinucleotide properties are given in Table 1.

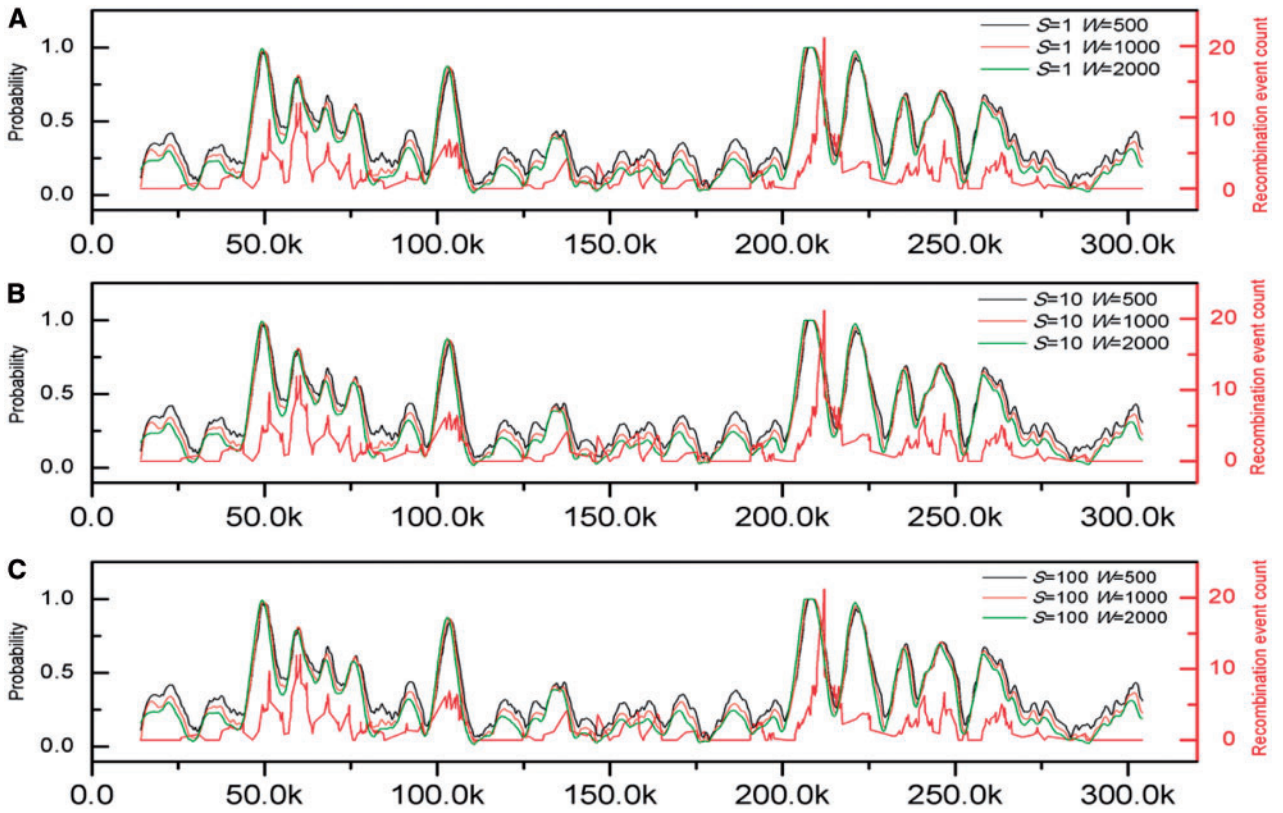


**Fig. 3.** Comparison between prediction results of iRSpot-EL and experimental map along yeast chromosome III. The red line represents the recombination event rate determined experimentally by Mancera *et al.* (2008). The other curves represent the probability values calculated by iRSpot-EL with different sliding window sizes and step sizes

input data via the Browse button. The input sequence should be in the FASTA format. For the examples of sequences in FASTA format, click the Example button right above the input box.

**Step 3.** Users are able to set three parameters for **iRSpot-EL**, including the size of sliding windows and step size. For more information of these parameters, please click the '?' symbol nearby.

**Step 4.** Click on the Submit button to see the predicted results. For example, if you use the query DNA sequence in the Example window as the input with '2' for the size of sliding windows and '200' for the step size, you will see the following results on the screen: (i) The query sequence contains one hotspot (sub-sequences: 3601–4200), and one coldspot (sub-sequence: 1–2400). (ii) By clicking Sequence Information, you will see the sequence information of the corresponding sub-sequence. (iii) By clicking Detailed results, you will see the detailed prediction results for each sliding window in the sub-sequence.

**Step 5**. The distributions of the hotspots and coldspots along the input sequence can be visualized by clicking the Result visualization button near the query sequence name.

## 4 Conclusion

The **iRSpot-EL** predictor is a new bioinformatics tool for predicting DNA recombination spots. When compared with the existing state-of-the-art predictors in this area, the new predictor yielded remarkably better prediction quality as demonstrated by rigorous cross-validation and genome-wide analysis. We anticipate that the web-server of **iRSpot-EL** will become a very useful high-throughput tool for conducting genome analysis.

## Funding

This work was supported by the National High Technology Research and Development Program of China (863 Program) (2015AA015405), the National Natural Science Foundation of China (Nos. 61672184, 61300112, 61573118 and 61272383), the Natural Science Foundation of Guangdong Province (2014A030313695), Guangdong Natural Science Funds for Distinguished Yong Scholars (2016A030306008), and Scientific Research Foundation in Shenzhen (Grant No. JCYJ20150626110425228).

*Conflict of Interest*: none declared.

## References

Akay,M.F. (2009) Support vector machines combined with feature selection for breast cancer diagnosis. *Exp. Syst. Appl.*, **36**, 3240–3247.

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

Chang,C. and Lin,C.J. (2001) LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, 1–27.

Chen,J. *et al.* (2007) Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. *Amino Acids*, **33**, 423–428.

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. *et al.* (2015a) Pseudo nucleotide composition or PseKNC: an effective formulation for analyzing genomic sequences. *Mol. BioSyst.*, **11**, 2620–2634.

Chen,W. *et al.* (2012) iNuc-PhysChem: a sequence-based predictor for identifying nucleosomes via physicochemical properties. *PLoS One*, **7**, e47843.

Chen,W. *et al.* (2015b) PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics*, **31**, 119–120.

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

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

Chou,K.C. (2011) Some remarks on protein attribute prediction and pseudo amino acid composition (50th Anniversary Year Review). *J. Theor. Biol.*, **273**, 236–247.

Chou,K.C. (2015) Impacts of bioinformatics to medicinal chemistry. *Med. Chem.*, **11**, 218–234.

Chou,K.C. and Shen,H.B. (2007a) MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochem. Biophys. Res. Comm.*, **360**, 339–345.

Chou,K.C. and Shen,H.B. (2007b) Review: Recent progresses in protein subcellular location prediction. *Anal. Biochem.*, **370**, 1–16.

Chou,K.C. and Shen,H.B. (2008) Cell-PLoc: A package of Web servers for predicting subcellular localization of proteins in various organisms. *Nat. Protoc.*, **3**, 153–162.

Chou,K.C. and Zhang,C.T. (1995) Review: prediction of protein structural classes. *Crit. Rev. Biochem. Mol. Biol.*, **30**, 275–349.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction of Support Vector Machines and Other Kernel-Based Learning Methodds*. Cambridge University Press, Cambridge, UK.

Davis,J. and Goadrich,M. (2006) The relationship between Precision-Recall and ROC curves. In: *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 233–240.

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.

Du,P. *et al.* (2012) PseAAC-Builder: across-platform stand-alone program for generating various special Chou's pseudo amino acid compositions. *Anal. Biochem.*, **425**, 117–119.

Fawcett,J.A. (2005) An Introduction to ROC Analysis. *Patt. Recog. Lett.*, **27**, 861–874.

Frey,B.J. and Dueck,D. (2007) Clustering by passing messages between data points. *Science*, **315**, 972–976.

Friedel,M. *et al.* (2009) DiProDB: a database for dinucleotide properties. *Nucleic Acids Res.*, **37**, D37–D40.

Gerton,J.L. *et al.* (2000) Global mapping of meiotic recombination hotspots and coldspots in the yeast Saccharomyces cerevisiae. *Proc. Natl. Acad. Sci. U S A*, **97**, 11383–11390.

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.

Jia,J. *et al.* (2016) pSumo-CD: Predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general PseAAC. *Bioinformatics*, doi: 10.1093/bioinformatics/btw387.

Jiang,H. *et al.* (2011) High recombination rates and hotspots in a Plasmodium falciparum genetic cross. *Genome Biol.*, **12**, R33.

Jiang,P. *et al.* (2007) RF-DYMHC: detecting the yeast meiotic recombination hotspots and coldspots by random forest model using gapped dinucleotide composition features. *Nucleic Acids Res.*, **35**, W47–W51.

Li,W. *et al.* (2001) Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, **17**, 282–283.

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.

Liu,B. *et al.* (2016a) iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. *Bioinformatics*, **32**, 362–389.

Liu,B. *et al.* (2015a) repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics*, **31**, 1307–1309.

Liu,B. *et al.* (2016b) repRNA: a web server for generating various feature vectors of RNA sequences. *Mol. Genet. Genomics*, **291**, 473–481.

Liu,B. *et al.* (2015b) Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res.*, **43**, W65–W71.

Liu,B. *et al.* (2016c) iDHS-EL: identifying DNase I hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. *Bioinformaitcs*, **32**, 2411–2418.

Liu,G. *et al.* (2012) Sequence-dependent prediction of recombination hotspots in Saccharomyces cerevisiae. *J. Theor. Biol.*, **293**, 49–54.

Mancera,E. *et al.* (2008) High-resolution mapping of meiotic crossovers and non-crossovers in yeast. *Nature*, **454**, 479–485.

Qiu,W.R. *et al.* (2016a) iPTM-mLys: identifying multiple lysine PTM sites and their different types. *Bioinformatics*, doi: 10.1093/bioinformatics/btw380.

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.

Qiu,W.R. *et al.* (2016b) iPhos-PseEn: identifying phosphorylation sites in proteins by fusing different pseudo components into an ensemble classifier. *Oncotarget.*, **7**, 51270–51283.

Shen,H.B. and Chou,K.C. (2006) Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, **22**, 1717–1722.

Shen,H.B. and Chou,K.C. (2007a) Signal-3L: a 3-layer approach for predicting signal peptide. *Biochem. Biophys. Res. Comm.*, **363**, 297–303.

Shen,H.B. and Chou,K.C. (2007b) EzyPred: A top-down approach for predicting enzyme functional classes and subclasses. *Biochem. Biophys. Res. Comm.*, **364**, 53–59.

Suykens,J.A. and Vandewalle,J. (1999) Least squares support vector machine classifiers. *Neural Process. Lett.*, **9**, 293–300.

Vapnik,V.N. (1999) An overview of statistical learning theory. *IEEE Trans. Neural Netw.*, **10**, 988–999.