Discovery Motifs by Evolutionary Computation

Jader C. Garbelini
Federal Technological University of Paraná
Department of Bioinformatics
Cornélio Procópio, Brazil – 80230–901
jadermcg@gmail.com

André Y. Kashiwabara
Federal Technological University of Paraná
Department of Bioinformatics
Cornélio Procópio, Brazil – 80230–901
kashiwabara@utfpr.edu.br

Danilo S. Sanches
Federal Technological University of Paraná
Department of Bioinformatics
Cornélio Procópio, Brazil – 80230–901
danilosanches@utfpr.edu.br

Keywords

Motifs; evolutionary computation; transcription factors

1. INTRODUCTION

Within the context of biological sequences analysis, regulatory motifs are short and recurring patterns of nucleotides residues that are presumed to have some common biological meaning. The overall gene expression is primarily controlled by protein factors, which bind to specific regions of the gene called transcription factors binding sites (TBFS).

The main objective of regulatory motifs discovery is to identify the ones responsible for the start of gene expression within a particular regulatory context. This problem consists in identifying which are overrepresented regulatory motifs, in a set of genes that are expressed by a specific transcription factor, within their respective promoter regions relative to other genes in the set.

In the past, TBFS were found using experimental techniques such as DNase footprinting, gel-shift or reporter construct assays. With the growing number of sequenced genomes, it was necessary to emerge a fast and reliable way to analyze all the data generated. Likewise, the computational techniques have been gaining prominence in the analysis of biological sequences. There are several motifs discovery approaches in the literature. They can be roughly divided into three main approaches: exhaustive techniques, probabilistic techniques and machine learning techniques.

In this paper, we present DMEC (Discovery Motifs by Evolutionary Computation), a method that uses evolutionary computation to find similar motifs in upstream regions of co-expressed genes. We particularly show that evolutionary algorithms can be very effective, and they achieve results equal to, or better than, other traditional approaches such as expectation maximization and Gibbs sampling.

2. MATERIALS AND METHODS

The main algorithm was developed using the Java programming language, release 8u65 (64 bits). The central idea of DMEC is to evolve a population of position specific scoring matrix (PSSM) to find a solution that maximizes the relative entropy or Kullback-Leibler divergence.

GECCO'16, July 20-24, 2016, Denver, Colorado, USA.

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2931640

2.1 Position specific scoring matrix

A PSSM, also known as PWM, is a weight matrix created from a multiple local sequences alignment. It is commonly used to represent a probabilistic model of a motif.

The first step in creating a PWM process starts with the count of the amount of residue in each alignment column. To prevent miscalculations pseudocounters may be used. There are several approaches to calculate pseudocounters in the literature, the best known being the Laplace's Rule. This approach adds 1 to count each residue, avoiding calculation errors when residues is not found in certain columns. This is particularly important because the 0 does not have a defined value log. This restriction will be better explained later. At the end of this stage we have the position frequency matrix (PFM) created.

The next step is the development of the position probability matrix (PPM). Its creation is quite simple and consists in dividing each PFM value by the total number of sequences plus the pseudocounters.

The following two steps are the last to create the PWM. Once created the PPM, we can use Bernoulli's model which assumes that the background model are independent successions of residues. Thus, we can calculate the a priori probability of each residue in the dataset or take the value 0.25 to nucleotides residues or 0.05 for amino acids residues. After obtaining the a priori values, each row of the PPM are divided by their respective residue prior values.

The last step is the logarithm calculation for each value resulting from the previous step. The logarithm calculation allows add up instead of multiplication. This is a security measure aimed at not overflow values when very small numbers are multiplied.

2.2 Initial population and structure of chromosomes

The initial population is randomly chosen, and its representation is given by an array of integers, which chromosomes are lines. Each chromosome can be seen as a multiple local alignment, where their genes store the start positions of each possible motif. A previous value called w, which refers to the size of the motif to be found, must be passed as a parameter to the algorithm. The value of each gene must respect the Equation 1, because, otherwise, it may contain values that exceed the total size of each sequence.

$$T = L - w + 1 \quad (1)$$

Where:

- T represents the maximum value that may receive each gene;
- L represents the total size of each sequence;
- w represents the size of the sought motif.

2.3 Evaluation and selection

The evaluation of each individual was calculated using the relative entropy, or Kullback-Leibler information. The relative entropy (see Equation 2) is a measurement that indicates how much a motif deviates from the background distribution of nucleotides.

$$F = \sum_{i=1}^{w} \sum_{j=1}^{\lambda} c_{i,j} \ log \frac{q_{i,j}}{p_j} \quad (2)$$

Where F is the value of score to be calculated, w is the size of motif, λ is the number of letters of the alphabet ($\lambda = 4$ for nucleotides and $\lambda = 20$ for amino acids), $c_{i,j}$ is the number of times a particular base λ_j appears in column w_i, $q_{i,j}$ is the conditional probability $Pr(\lambda|w_i)$ and p_j representing the background probability of residues distribution.

After the random initialization of chromosomes, each individual is transformed into a subsequence of size w. They are positioned one below the other, forming an $n \times w$ matrix, where n is the number of sequences available in the dataset and w is the size of the motif to be discovered.

The goal is to have a score that indicates the degree of confidence from motifs evaluated. The larger the F value, the greater the chance the motif has been generated by the model. In other words, the F value corresponds to maximum likelihood $Pr(motif|model)$, regarding the probability background $Pr(motif|background)$.

After the fitness calculation, the selection is performed using the tournament technique. Two individuals are randomly selected, and their scores are compared. The chromosome that has greater fitness wins, and consequently it is selected to compose the next generation of the population.

2.4 Matrix recombination

Recombination involves mixing two or more matrices of the population. In general, two individuals are randomly selected to exchange genes between them. The recombination ends when the intermediate population reaches the same size as the initial population. The purpose of this operator is to keep the genetic diversity among the elements and avoid premature convergences.

2.5 Matrix mutation

The matrix mutation changes a set of individuals randomly chosen according to a previously established mutation rate. Also, a random number of genes change to a value that respects the limit imposed by Equation 1. The mutation is only effective if the individual's fitness value improves compared to the last value. Otherwise, the mutation is not performed. Because the mutation can only be carried out in cases of fitness improvement, elitism was not enabled.

2.6 Slide window

This operator add a heuristic based on opt operators. Its slides a window to the right or left according to a parameter p previously established. If $p < 0.5$, then the window slides to the left, otherwise, it slides right.

2.7 PSSM to find out more motifs

In many cases, the nucleotide sequences have more than one motif in their structure. Thereby, it was possible to use the PSSM matrix generated by running the DMEC to find these regions. To do so, we divide the target dataset into "pieces" of size of size w, or w-mers, and calculate a p-value to each one. The p-value is the probability to find a score greater than or equal to that observed randomly, i.e. generated by the background model. The regions that possess a p-value smaller than a cutoff entered by the user, are classified as possible motifs.

3. RESULTS

We tested our method in several real and synthetic datasets, however, for lack of space we just put the results in the real dataset CRP (cAMP protein receptor). This dataset has 18 sequences with 105 residues each. In addition, it has 23 motifs experimentally identified.

Table 1: Real positions vs discovered positions.

Sequence	Real Position	Discovered Position
cole1	17, 61	61
ecoarabop	17, 55	55
ecobglr-1	76	76
ecocrp	63	63
ecocya	50	50
ecodeop	7, 60	7, 60
ecogale	42	42
ecoilvbpr	39	20
ecolac	9, 81	9
ecomale	14	14
ecomalk	29	61
ecomalt	41	41
ecoompa	48	48
ecotnaa	71	71
ecouxu-1	17	17
pbr-p4	53	53
trn9cat	1, 85	84
tdc	78	76

4. CONCLUSION

We presented the DMEC, a novel algorithm based on evolutionary computation to efficiently solve the search of motifs. We have also made a comparison of our algorithm to other well-known approaches in literature, and concluded that the evolutionary computation was able to achieve equal or better results. For future work, we will implement an efficient and automatic way to find the size of w. We will also work on a method to penalize sequences that do not have motifs. Our approach works on the premise that each sequence has at least one motif copy. In Datasets that have sequences without motifs, the results may not be satisfactory.