POSTER ABSTRACT

# Evolutionary Segmentation of Yeast Genome

Daniel Mateos
Computer Science Depmt. U. of Seville
Avda. Reina Mercedes s/n 41012
Seville Spain
+34 954 553 866
mateos@lsi.us.es

José C. Riquelme
Computer Science Depmt. U. of Seville
Avda. Reina Mercedes s/n 41012
Seville Spain
+34 954 552 775
riquelme@lsi.us.es

Jesús S. Aguilar-Ruiz
Computer Science Depmt. U. of Seville
Avda. Reina Mercedes s/n 41012
Seville Spain
+34 954 553 871
aguilar@lsi.us.es

## ABSTRACT

Segmentation algorithms differ from clustering algorithms with regard to how to deal with the physical location of genes throughout the sequence. Therefore, segments have to keep the original positions of consecutive genes, which is not a constraint for clustering algorithms. It has been proven that exist functional relations among neighbour-genes, so the localization of the boundaries between these functionally similar groups of genes has turned out an important challenge. In this paper, we present an evolutionary algorithm to segment the yeast genome.

## 1. INTRODUCTION

Chromosomes are organized in gene sequences. Each chromosome has a variable number of genes that physically are located in consecutive positions. Genome study tries to find the functionality of every gene. Recent researches in Genetics try to discover the existence of functional relations among one gene and its “neighbours” within a chromosome. This process is known as DNA segmentation, and it exists little scientific literature about it.

The commonly used techniques work with DNA sequences instead of numerical values associated to each gene. Nowadays, the microarray techniques are generating great amounts of data, which might be very useful to analyze the functional properties of genes, as they collect a numerical value for every gene. This fact clears the way for new algorithms that can handle this sort of data.

In this work, we present an Evolutionary Algorithm (EA) to find valid segments from the yeast genome. For the yeast genome study, we have a file with the sixteen chromosomes (*NREG*). Each gene is a row of the file.

The file has three columns, and each column represents a genomic characteristic under specific conditions. The object is either clustering consecutive genes with similar properties with regard to the three variables, or clustering consecutive genes properly differentiated from adjacent clusters. Each cluster will be a segment of genes, as it will maintain the physical location within the genome.

## 2. EVOLUTIONARY ALGORITHM

Each individual of population is a static array of natural numbers with size *NCOR*, and it represents a cutoffs collection into yeast genome. Fifteen of these cutoffs correspond to the sixteen chromosomes of yeast genome, and they are permanents. The sixteen cutoffs corresponding to centromeres also are permanents.

These cutoffs (*NCORFIJ*=31) although they can’t be moved, they have been included in all individuals, making easier the computing process. For example, if a cutoffs array includes among others, the values 34, 57, 7, 25 and 80, it means that there’s a cutoff between the 34th and the 35th entry of file, between the 57th and the 58th entry, between the 7th and the 8th entry, etc. Therefore, the segments comprise from first to 7th gene, from 8th to 25th gene, from 26th to 34th gene, from 35th to 56th gene, etc.

In order to verify the quality of the fitness functions, we execute the algorithm with the original data, and with randomized versions. We can understand that a fitness function is correct if the results obtained with the random data are inferior to the obtained with the original data. In another case, we can say that we have an “artifact” (An apparent experimental result that is not actually real but is due to the experimental methods).

The fitness function of the first experiments, calculated the median of each variable for each segment, and it maximized the correlations between these medians (Eq. 1).

$$F_1 = \max\left(r_{12}^2 + r_{13}^2 + r_{23}^2\right)$$
$$r_{ij} = correl\left(Med_i, Med_j\right)$$
$$Med_i = \bigcup_{k=1}^{NCOR+1} med_k^i$$

$$med_k^i = median\ of\ the\ k^{th}\ segment\ for\ the\ i^{th}\ variable$$

**Eq. 1.** Fitness function (inter-median correlations)

This fitness function turned out to be an artifact, because the results with random data were similar to the results with the original data.

Another possibility for the fitness function is to maximize the difference between the values of the variables of two consecutive segments, for each variable separately and for all. We can use as statistical the median (robust against outliers) or the classic

*SAC ’04*, March 14-17, 2004, Nicosia, Cyprus.

arithmetic average for representing the value of a variable. We represent these functions in the equations 2 and 3 respectively.

$$\textit{for each variable: } f_j = \max\left(\sum_{i=1}^{NCOR+1}\left|me_i^j - me_{i+1}^j\right|\right)$$

$$me_i^j = \textit{median of the } i^{th} \textit{ segment for the } j^{th} \textit{ variable}$$

$$\textit{for all variables: } F_2 = f_1 + f_2 + f_3$$

**Eq. 2.** Fitness function (inter-median differences)

The fitness function which maximizes the inter-average differences (Eq. 3) has the advantage of the computational cost of average, which it is smaller than the one of median.

$$f_j = \max\left(\sum_{i=1}^{NCOR+1}\left|\overline{x_i^j} - \overline{x_{i+1}^j}\right|\right)$$

$$\overline{x_i^j} = \textit{average of } i^{th} \textit{ segment for the } j^{th} \textit{ variable}.$$

**Eq. 3.** Fitness function (inter-average differences)

We have used uniform crossover owing to the easy implementation and to the adaptation to the problem. That is, we build a new individual choosing randomly cutoffs from one of both parents. As well, we applied other well-know methods (fixed length one point and fixed length two points) but they made worse results.

The mutation operator alters each cutoff according to two probabilities: p1 and p2. The probability p1 controls if a cutoff must be modified; and the probability p2 controls if the mutation has resulted in a replacement by a random cutoff in the range allowed, or in a light change of the existent cutoff.

# 3. EXPERIMENTS

We show the EA parameters and their values for each experiment. As well, we used the uniform crossover as crossover operator, and 0.4 and 0.2 as p1 and p2 probabilities respectively.

| Fitness | Population | Generations | #Cutoffs |
|---|---|---|---|
| Correlation | 300 | 50 | 50 |
| Median | 400 | 200 | 50 |
| Average | 400 | 200 | 50 |

**Table 1.** EA parameters

In the tests, we executed 20 times the EA with the original and random data on Pentium IV to 2.4 GH, with 512 MB of RAM.

| Fitness | Original data | Random data |
|---|---|---|
| $F_1$ | 2.541378498 | 2.463235855 |

**Table. 2.** Example of artifact for fitness function based on correlations

The fitness function based on correlation isn't valid for to measure the fitness of segmentation, because we can find very high inter-variable correlations with any distribution of genes in chromosomes. The time computation for an execution is approximately of 2 minutes and half.

| Variable | Original data | Random data |
|---|---|---|
| gc3s | 2.982499838 | 1.934499979 |
| expH | 19.699998856 | 22.200000763 |
| rec_ | 13.464997292 | 6.829999447 |
| All ($F_2$) | 42.144832511 | 40.086082458 |

**Table 3.** Fitness function for inter-median differences

It exists significant differences between the original and random data for two of the three studied variables (*gc3s* and *rec _*).

These results demonstrate that for the two mentioned variables in the first place, it is possible to find a segmentation where the results depend of the order of the genes, that is to say, a segmentation of the chromosome with own sense.

The time of computation is approximately of 4 minutes by execution.

| Variable | Original data | Random data |
|---|---|---|
| gc3s | 1.828593254 | 0.804380655 |
| expH | 88.142761230 | 86.721817017 |
| Rec_ | 13.985332489 | 3.723937988 |

**Table 4.** Fitness function based on averages

The main virtue that we can emphasize of the fitness function based on averages, is that its time of computation is approximately the fourth part (a minute) of the previous case. We can see the best results in the table 4.

# 4. ADDITIONAL AUTHORS

Antonio Marín, Department of Genetics of University of Seville, anmarin@us.es.

# 5. REFERENCES

[1] Bradnam KR, Seoighe C, Sharp PM, Wolfe KH. 1999. G+C content variation along and among *Saccharomyces cerevisiae* chromosomes. *Mol Biol Evol* **16**: 666-675

[2] Kruglyak S, Tang H. 2000. Regulation of adjacent yeast genes. *Trends Genet* **16**: 109-111.

[3] Li W, Bernaola-Galva P, Haghighi F, Grosse I. 2002. Applications of recursive segmentation to the analysis of DNA sequences. *Computers & Chemistry (genome and informatics special issue)*, **26(5)**: 491-510.

[4] Li W. 2001. New stopping criteria for segmenting DNA sequences. *Physical Review Letters*, **86(25)**: 5815-5818.

[5] Li W. 2001. DNA segmentation as a model selection process. *RECOMB01: Proceedings of the Fifth Annual International Conference on Computational Biology*, pp. 204-210. ACM Press.