Statistical Modeling of Genomic Words and Motifs

Guozhu Zhang[1*], Stephen Sauchi Lee[2]

[1] Bioinformatics Research Center, North Carolina State University, Raleigh, NC, USA

[2] Department of Statistical Science, University of Idaho, Moscow, ID, USA

* Correspondence to:
Guozhu Zhang (gzhang6@ncsu.edu)

Abstract

The arrangement of the four nucleotides A, C, G, and T along the genome is known to be non-random. Vast amount of information are built into the complex arrangements and compositions of genomic nucleotides. It can be viewed as a book of nucleotide text of instructions at the cellular level. Genome is decoded as a continuous stream of nucleotide alphabets message as one read the genomic text. We approach the reading of genomic text by segmentation – dividing the continuous stream into chunks according to some statistical measures of homogeneity. The goal would be to segment the genome into the most probable dictionary of motifs or words. Words are defined by our segmentation method as more homogeneous units within the boundaries than without. The core idea of this paper is to introduce the method of setting word boundaries. We applied the method to compare the yeast and worm genomes, to distinguish ordered and disordered protein sequences, and to characterize different English texts.

Key word: Segmentation, genomic words, motifs.

Introduction

In this article, we are going to seek the usefulness of word segmentation algorithm in large scale protein and genomic research. The core idea is that the genome has its own structure of DNA patterns (called words) and relationships between these patterns in word counts, locations, and distributions. The ordered and disordered protein also follows the same concept; they also have their own structure of amino acids patterns. We do not believe that the linguistic structures of grammar and syntax in any written language will resemble patterns in any genome or protein. However, the statistical principles used in the computational linguistic segmentation which involves abundant counts, over and under representations, entropy, and homogeneity versus heterogeneity in substrings, will be promising approaches to explore the manifold genomic and protein word landscapes.

Word segmentation, i.e., identifying word boundaries in continuous speech or text, plays an important role in Natural Language Processing (NLP). In recent years, people have developed some unsupervised approaches to word segmentation and have made excellent progress. Those algorithms have been applied to English, Chinese and some other languages which all contain no spaces and punctuations. The segmentation accuracy is typically estimated by the F score [3] as well as the precision and recall rate [3]. A high F-score indicates better overall accuracy.

	True	
Predicted	TP (true positive)	FP (false positive)
	FN (false negative)	TN (true negative)

Table I: Expect versus Predicted table

The table above is the prediction table and the F score is:

$$\mathrm{F} = \frac{2 \times \text{Recall rate} \times \text{Hit rate}}{\text{Recall rate} + \text{Hit rate}}$$

Where,

$$\text{recall rate} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}\text{, and hit rate} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$

In 2006, Cohen *et al.* developed a word segmentation algorithm which called voting experts and applied this algorithm to George Orwell's *1984* and got a score of $\mathrm{F} = .76$. In 2010, Wang *et al.* also developed a new unsupervised approach to word segmentation. This algorithm was focused on Chinese corpora with the highest score of $\mathrm{F} = .81$. In 2011, Chen *et al.* developed a simple and effective unsupervised word segmentation approach and applied this method to English phonetic transcripts and achieved a score of $\mathrm{F} = .78$. In this article, we employed the modified voting experts algorithm to English test as well as to genomic and protein data.

Homogeneous segments of genomic DNA are frequently associated with important units of biological data [7]. People have developed lots of methods in large genomic segmentation. Hidden Markov Models (HMM) were first developed for speech recognition and now are widely used in computational biology, especially in large genomic and protein segmentation. Additionally, higher-order- Markov-Model based segmentation procedures are helpful in the classification of biologically meaningful regions [10]. Recently the Bayesian approach has been widely used in DNA sequence segmentation [1,8]. Now we used the computational linguistic method to see if we can find some meaningful regions.

Interestingly, intrinsically disordered proteins do not inhabit a reliable three dimensional structure, as they exist as interchanging conformations in solution [6]. Each protein is defined by its sequence of amino acids, corresponding to an alphabet of twenty symbols. Vucetic *et al.* have developed different methods for improving protein disorder prediction and got an accuracy of 82.6% [11]. These methods include ordinary least squares (OLS) [11], logistic regression (LR) [11], and neural networks (NN) [11]. In our research, we are going to find significant sequence patterns in the hope to improve ordered versus disordered protein prediction.

Materials and Methods

Before we introduce the segmentation algorithm, we would first like to introduce several statistics. The first one is conditional entropy. The entropy or unpredictability of elements within a chunk is relatively low, whereas the entropy or unpredictability of elements between chunks is relatively high.

The conditional entropy is defined as:

$$H(x_n|x_1,\dots,x_{n-1}) = -\sum_{x_n \in X} \Pr(x_n|x_1,\dots,x_{n-1}) \log \Pr(x_n|x_1,\dots,x_{n-1}),$$

where $\Pr(x_n|x_1,\dots,x_{n-1})$ is the conditional probability and is illustrated in the following example:

Consider a DNA sequence "TTGATTC". The frequency of "A", "C", "G" and "T" is1, 1, 1 and 4 respectively. And the frequency of "AT" (2-mer subsequence which includes A) is 1. Thus,

$$\Pr(T|A) = \frac{\text{frequency of (AT)}}{\text{frequency of (A)}} = 1$$

However, the conditional entropy of "A" is zero, because each occurrence of "A" is always followed by "T".

The other variable used in this research is rank. It is just the group of the significant chunks with same length. This is because we found that the significant conditional entropy prefers the shorter chunks which might be biased by excluding the longer ones. Thus, by using the rank we can keep some longer significant chunks as well as reduce the bias. We use the conditional entropy as an example to show how to find the rank. The data set used here is the yeast chromosome II (UCSC Genomic Research Center, Saccharomyces cerevisiae version one, Oct 2003) [15].

Word	*Entropy*	*Rank*
A	1.343	3
C	1.338	4
G	1.366	1
T	1.359	2

Table II: Rank

For all the chunks of size k, we rank them based on the entropy from the maximum to minimum with 1 assigned to the maximum chunk and 2 assigned to the chunk which has the next largest value, etc.

We applied this algorithm to an English text (King James Bible) first and found that the conditional entropy and the rank combined to give the best prediction accuracy, which is 96.28% estimated by the **ROC** curve [12]. Other statistics such as differential entropy [9], log ratio of observe count over expected count [4], frequency and their ranks didn't help increase the accuracy.

Our segmentation algorithm was designed based on the voting experts method [3], but it performed better than the voting experts. The voting expert is a simple segmentation method, which is used as an unsupervised algorithm for segmenting sequences. However, this method highly depends on the window length. We applied this method to the same English text (King James Bible) and found that the best prediction accuracy was found at a window length threshold equal to 7. That helps explain why the

voting expert tends to miss the longer words. Considering those advantages and drawbacks, our segmentation algorithm is:

- Define significant substrings based on conditional entropy and its rank

We calculated the conditional entropy and its rank from the chunks of size 2 up to size 40. The chunk is just all the possible combinations. For example, in the English text, the total combination of substring of size 2 is $26^2 = 676$. We deleted the chunks that their entropies are zero, because those chunks are not that frequent and would not help to increase the prediction accuracy and find the true words. We used the English text as a supervised learning and found that the top 5% conditional entropies and their ranks gave the best accuracy. Thus, we treated the chunks that have the top 5% conditional entropy and rank as statistically significant chunks.

- Forward run through the whole text, if the chunk is significant, and then give a vote to the end of that chunk. We used the significant chunks of size two as an example to show how the voting procedure works.

Before voting, each individual gap has zero votes:

i	0	n	0	t	0	h	0	e	0	b	0	e	0	g	0	i	0	n	0	n	0	i	0	n	0	g	0

For the first two letters “in” of this text, we found that it is significant, so the second gap receives a vote.

i	0	n	1	t	0	h	0	e	0	b	0	e	0	g	0	i	0	n	0	n	0	i	0	n	0	g	0

Then we moved on to the next two letters, if it is significant, then we gave a vote to that gap. After we run through the whole text based on all the significant chunks, each individual gap would probably receive some votes or none. Then we chose a minimum vote threshold to segment the whole string into words. Because we had two groups of significant chunks, one was found by the conditional entropy and the other one was found by its rank. After voting, each gap should receive two different numbers of votes, we added them together to get the total votes.

- Segmentation

Here is an example shows how we segment the whole string. After voting, each individual gap has some votes or none. We choose the minimum vote threshold at 8 to segment the whole string. Thus, this string was segmented to “inthe” and “beginning”. This text contains the word “the”, however, after segmentation; this algorithm might not find that word. It would either be separated or combined with other letters.

i	0	n	1	t	1	h	0	e	8	b	3	e	7	g	1	i	2	n	2	n	0	i	1	n	2	g	22

Results and Discussion

We first applied the method to four different English texts, which are Alice (Alice in Wonderland), King (King James Bible), Koran (Koran) and Tale (A Tale of Two Cities). The significant words are listed in a decreasing order in p-value in Table III. Due to limited space we only list part of the entire table. The others can be obtained from the author. The p-value was computed from the chi-square test, which tests for the homogeneity across different groups. After segmentation, we counted the frequency of each single word and applied a chi-square test. As we can see from the table, most of the

significant words are true English words. This result gave us a bright sight in finding true protein and genome words.

word	alice	king	koran	tale	chisquare	pvalue
lord	0	7247	311	0	2592	0
they	0	27	269	0	1219	6.6E-264
king	54	2356	1	22	926.3	1.8E-200
unto	0	3131	754	0	675.7	3.9E-146
land	0	1669	34	1	668.7	1.3E-144
been	5	225	189	290	657.2	4E-142
same	0	156	267	41	590.7	1.1E-127
thou	0	2034	183	0	580.2	2E-125
lves	0	0	104	0	538.8	1.8E-116
door	29	35	0	93	527.7	4.8E-114
jury	23	0	0	32	511.2	1.8E-110
dodo	12	0	0	0	508.4	7.3E-110
said	0	2254	241	76	449.7	3.76E-97
baby	10	0	0	0	423.7	1.67E-91
game	10	0	0	0	423.7	1.67E-91
some	9	13	29	91	388	8.84E-84
will	0	1932	694	96	385.6	2.87E-83
hall	1	1400	126	8	379.3	6.81E-82
fore	2	2273	540	61	363.9	1.46E-78
went	0	874	0	22	343.5	3.8E-74
bank	0	0	0	49	343.3	4.17E-74
have	14	2254	924	310	329	5.18E-71
came	3	951	18	26	322.8	1.18E-69
eous	0	0	61	0	316	3.36E-68
hath	0	1427	364	0	313.6	1.15E-67
hand	0	605	41	269	309.1	1.05E-66
come	2	1458	138	49	306.2	4.45E-66
dark	0	0	0	43	301.3	5.26E-65
face	5	169	0	135	300.7	7.15E-65
arth	0	861	40	4	292.9	3.51E-63
head	14	138	0	122	292.7	3.88E-63
ywho	0	0	55	0	285	1.8E-61
also	0	839	46	0	282.2	7.25E-61
pass	0	619	3	0	270.6	2.32E-58
room	0	0	0	38	266.2	2E-57
road	0	0	0	37	259.2	6.56E-56
made	0	1081	80	40	252.3	2.1E-54
lady	0	0	0	36	252.2	2.15E-54
very	45	225	6	88	249.7	7.58E-54

Table III: Highly 40 significant words after segmentation

In genomic research, we found that after segmentation, some words only occur in all the chromosomes that belong to the same species. For example,

> "CGTT, ACCTG, ACCTT, CGTTT, CTATT, CTCAA, …" these words occur in all the 16 yeast chromosomes but not in C.elegans.
> "TCC, AACC, CCCC, CTCC, CTGC, GACC, …" these words occur in all the 6 C.elegans chromosomes but not in yeast.

All the words above are found after segmentation.

In the ordered and disordered protein research, after applying our method, each single protein was segmented into many chunks which are treated as true protein words. For a specific word, we count how many times it occurs across all the ordered proteins, and then the frequency is just the observed count over the total proteins. This is the same in the disordered proteins. Finally, we did a two sample proportion test of all the words from 1-mer up to 4-mer. And we got only one significant word of all the possible combinations of four different amino acids. Thus, when the word length is greater than four, there will be just a few significant words. The following are the list of part of the significant words,

"PP, KK, TAE, TSS, DDE, DGE, ILA, MEE, PVE, PKE, RPR, RSK, KEE, EE, QEE, QKK, QPP, STS, TAT, VAE, VSS, AP, GEG, TAP", these are all the highly significant protein words that have a higher frequency in disordered protein.

"C, K, L, D, G, W, R, V, H, T, M , N, F, I, Y, GT, E, Q, A, P, SV, YG, IG, LG, DG, IV, S, DR, IA, VL, FV, RV, VV, NL, ST, LV, TL, TA…" these are part of the highly significant words that have a higher frequency in ordered protein.

Conclusion and Comments on future work

In this article, we employed a computational linguistic method to large scale DNA sequence and protein sequence data. We applied the algorithm to the King James Bible and it worked very well. It performed better than the Voting Experts method. Meanwhile, it gave a fairly high F score and it was also useful in finding the true English words. Thus, theoretically this algorithm should also be useful in other sequence data sets, because the statistical principles used here, such as conditional entropy, differential entropy, frequency, are the same. Statistically, we found some significant words in both the protein and the yeast DNA data set. Statistical analysis of the genome and protein data is exploratory but illuminating. However, the biological significance of these words still needs to be further investigated.

For the ordered and disordered proteins, in the future, we have to test if these significant words can help to build the protein predictor or not. Such as in logistic regression, should these words be given more attentions? For genomic research, we have to see if those segments are biologically useful. Also, are these unique patterns useful in species determination? This method also needs more comparison to other existing methods like Hidden Markov Models in genomic segmentation.

References

1. R.J. Boys and D.A. Henderson (2004) **A Bayesian Approach to DNA Sequence Segmentation,** *Biometrics*, **60**, 573-581

2. S.J Chen, Y.H. Xu and H.Y. Chang (2011) **A simple and Effective Unsupervised Word Segmentation Approach. Conference on artificial intelligence**, 866-871

3. P. Cohen, N. Adams and B. Heerings (2007) **Voting Experts: An Unsupervised Algorithm for Segmenting Sequences**. *Intelligent Data Analysis,***11,**607-625

4. S. Lee (2008) **Seeking Significant Oligomers via Set Partitions Expected Count**, *International Journal of Computational Science*

5. M. Logsdon (2011) **The Voting Experts Segmentation Algorithm**, unpublished paper from University of Idaho

6. C.J. Oldfield, Y. Cheng, M.S. Cortese, C.J. Brown, V.N. Uversky and A.K. Dunker (2005) **Comparing and Combining Predictors of Mostly Disordered Proteins**, *Biochemistry*, **44**

7. L. Peshkin and M. S. Gelfand (1999) **Segmentation of yeast DNA using hidden Markov models**, *Bioinformatics*, **15**, 980-986

8. V.E. Ramensky, V.JU. Makeev, M.A. Roytberg and V.G. Tumanyan (2000) **DNA Segmentation Through the Bayesian Approach**, *Journal of Computational Biology*, **7**

9. T. Schurmann and P. Grassberger (2002) **Entropy estimation of symbol sequences**. *CHAOS*, **6**, 414-427

10. V. Thakur, R. K. Azad and R. Ramaswamy, (2007) **Markov models of genome segmentation**, *American Physical Society*, **75**

11. S. Vucetic, C.J. Brown, A.K. Dunker and Z. Obradovic (2003) **Flavors of Protein Disorder**, *Protein*, **52**, 573-584

12. S. Vucetic, P. Radivojac, Z. Obradovic, C.J. Brown and A.K. Dunker (2001) **Methods for Improving Protein Disorder Prediction**, *Conference,***4**,2718-2723

13 H.S. Wang, J. Zhu, S.P. Tang and X.Z. Fan (2011) **A New Unsupervised Approach to Word Segmentation**. *Computational Linguistics*, **37**, 421-454

14. Bioconductor, *http://www.bioconductor.org/*

15. *UCSC Genomic Research Center http://genome.ucsc.edu/*