Segmenting DNA sequence into ‘words’ based on statistical language model

Wang Liang

Tencent, SOSO, 100080, P.R. China

*To whom correspondence should be addressed. E-mail:wangliang.f@gmail.com

[Abstract] This paper presents a novel method to segment/decode DNA sequences based on n-gram statistical language model. Firstly, we find the length of most DNA “words” is 12 to 15 bps by analyzing the genomes of 12 model species. The bound of language entropy of DNA sequence is about 1.5674 bits. After building an n-gram biology languages model, we design an unsupervised ‘probability approach to word segmentation’ method to segment the DNA sequences. The benchmark of segmenting method is also proposed. In cross segmenting test, we find different genomes may use the similar language, but belong to different branches, just like the English and French/Latin. We present some possible applications of this method at last.

1 Introduction

Analyzing the “meaning” of DNA sequence is the key mission of bioinformatics. For gene sequence like “ATCGATGGG”, it could be divided into “ATC/ GAT/ GGG” and then decode to amino acid sequence “I/ D/ G”. But how to decode such amino acid sequence is still not very clear. Moreover, there is still no effective method to “decode” the non-gene regions. Like the gene triplet codes decoding, could we decode no-gene sequence and amino acid sequences? This paper just discusses this question.

The sequence decoding operation contains two steps. Firstly, we should segment the sequence properly. Secondly, we assign the ‘names’ for these segmented fragments. Such fragment is normally called “word”, which may represent the entity or some functions. The sequence segmentation is the key operation. We could use a simple example to describe our mission. For a sequence “Iloveapple”, we need divide it into “I/ love/ apple”.

There is also research topic ‘DNA segmentation’ in bioinformatics (1). The length of their ‘segmentation’ is normally Kbps to Mbps. They mainly concern the classification of long sequences, for example, segmenting the DNA into gene regions and non-gene regions. Here we discuss the basic meaningful segmentation of DNA, the ‘DNA word’, whose length may be only several bps.

Only after segmenting the sequence, we could detect the meaning of sentence and do other analysis. Similarly, correctly segmenting the DNA sequence is also the first key step to understand the DNA. For some language like English, the sequence is naturally segmented into words by space and punctuation. But for DNA sequence, there is no space or punctuation. In some East-Asia language like Chinese, there are also no natural delimiters. It’s also a big problem to deal with these languages by computer. In these years, this problem has been solved to great extent. Two main methods are usually used to deal with these problems. Some simple methods are normally

designed based on the vocabulary. Others use statistical features to improve the segmentation methods (1,3,4) .

In segmentation researches, some does not need the any vocabulary. These methods automatically extract the rules from large raw corpus, and then use these rules to segment the sequences. Such methods are called unsupervised or self-organized segmentation. These research normally uses the Mutual Information to identify words boundary (5,6). Some improved versions apply the EM algorithms to train the segmentation model (7,8,9,10). Although supervised approaches reach higher accuracy (>95%) than unsupervised ones (normally <85%) in many cases, they involve much more human effort. Furthermore, unsupervised approaches are more adaptive to relatively unfamiliar languages for which we do not have enough linguistic knowledge.

Still for sequence “Iloveapple”, even we have no an English vocabulary, we could still segment it into “I/ love / apple”, if we have many English documents (without any space and punctuation). Since we have little knowledge about “words” of DNA, unsupervised method seems very appropriate to segment the DNA sequence. We just apply these researches to design the DNA segmentation method.

2 Statistical language model of DNA

Most segmentation methods in natural language processing area are designed based on statistical language models. The most common model is n-gram language model (11).

N-gram are sequences of ‘n’ words in a running text. N-gram frequencies or more sophisticated statistical models of n-gram are widely used for text processing applications such as information retrieval, language identification, etc. In a biological context, n-gram can be sequences of amino acids or nucleotides. For instance, the sequence “AAACG”, its unigram are A,A,A,C， G. The 2-grams are AA, AA, AC, CG. Similarly, 3-grams are AAA, AAG, ACG.

N-gram language model uses the basic statistical properties of n-gram. An *n*-gram model predicts x_i based on $x_{i-(n-1)},\ldots,x_{i-1}$. In Probability terms, this is $P(x_i \mid x_{i-(n-1)},\ldots,x_{i-1})$. When used for language modeling, independence assumptions are made so that each word depends only on the last n-1 words.

N-gram based methods have already been successfully applied in biological domain (12). For example, the relative abundance of n-gram sequence is used as the genome signatures (13). The biological language models are applied to study the evolutionary tree (14). N-gram composition of amino acid sequences is also used to classify the protein (15).

The basic statistical feature for an n-gram model is language perplexity or entropy, which describe how well the language model predicts a new text composed of unseen sentences. Here we use the genome of 12 mode creatures to build n-gram models and calculate the language perplexity. The relation of n of “n-gram” and the perplexity of each genome is shown in Fig.1:

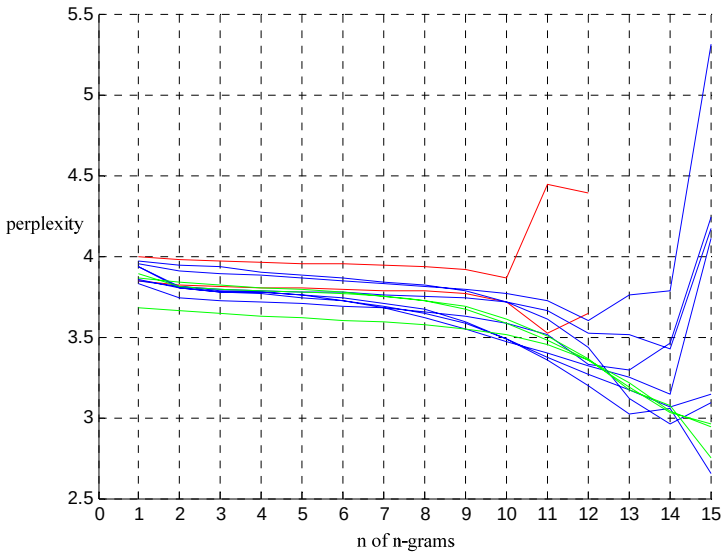

Fig 1. The relation of n-gram n and perplexity of DNA of model species. The red lines correspond to Aspergillus and Schizosaccharomyces. The green lines are Acyrthosiphon, Zebrafish, Strongylocentrotus. The other blue lines are Arabidopsis, Caenorhabditis, Fruit Fly, Human, Mouse, Oryza,Xenopus.

The Fig.1 shows the perplexities reduce with the increase of n till n<14. When n > 14 the perplexity of most genomes will increase, which means the language model will not believable for data sparse problem. The perplexities of these genomes are about 2.96 to 3.86. The corresponding language entropies are about 1.5674 to 1.9485. Correspondingly, the language entropy of Human language is about 4 to 10 bits (16). In the following sections, we will detail how to segment DNA sequence based on these DNA n-gram language models.

3 The length of DNA words

To segment a sequence, we should know the length of "word" first. The 'word' length should not too short. If the word length is 1, the DNA will have 4 words and could only represent 4 things. The length should also not too long, for example, if the word length is 1M, there will be 4^1M words in DNA vocabulary. But the length of most chromosomes is server Ms, which could only contain several words.

We could use a statistical method, "zipf's laws" to estimate the length of most words. The "zipf's laws" states, in a long enough document, about 50% words only occur once. These words are called 'Hapax legomenon'. In building the n-gram model, we have counted the number of 'n' length words. The relation of word length 'n' and the percentage of Hapax legomenon are shown in Fig.2.

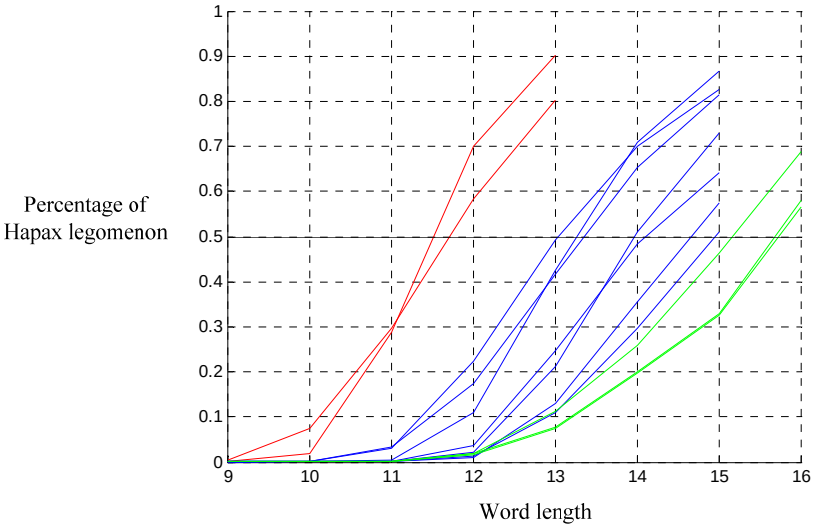

Fig.2 n-gram count results, (x axis is the word length, y axis is the percentage of hapax legomenon). The red lines correspond to Aspergillus and Schizosaccharomyces. The green lines are Human, Mouse, Xenopus. The other blue lines correspond to other genomes.

In Fig.2, we find 50% line of Hapax legomenon corresponding to word length 12 to 15 of most genomes, which show 12 to 15 bps is the word length for most DNA words.

In Fig.1, the upper bound of n of n-gram model for genomes is 15, which shows the fifteen letters almost has no relation with the prevous14 letters in a sequence. It also means the lengths of most words should be no more than 15. In our experiment, we use the 15 as the maximal length of DNA "words".

4 Segmenting the DNA sequence

Till now, we have the n-gram DNA model and know the length of DNA words. We could directly use the methods from existing research to segment DNA Sequence. One basic method is called 'probability approach to word segmentation'.

For an example, a sequence "ATAC", assume maximal word length is 3, its segmentation could be "ATA/ C", "AT/ AC", "AT/ A/ C", "A/ TAC", "A/ TA/ C", "A/ T/ AC", " A/ T/ A /C".

The product of probability of each fragment in one segmentation is the probability of this segmentation. We select the segmentation candidate which has the maximal probability as the segmentation for the sequence:

$$P^*(ATAC) = \max \begin{cases} P(ATA)P(C) \\ P(AT)P(AC) \\ P(AT)P(A)P(C) \\ P(A)P(TAC) \\ P(A)P(TA)P(C) \\ P(A)P(T)P(AC) \\ P(A)P(T)P(A)P(C) \end{cases} \quad (1)$$

Based on n-gram model, we could get the probability of each fragment like “ATA”. For example: P(‘ATA/ C’)=P(ATA)P(C). According to n-gram models, we could get: P(C), and P(ATA)=P(A)P(T|A)P(A|AT).

If the sequence length is m, there will be O(2^(m-1)) forms of segmentation. To reduce the calculation requirement, dynamic programming methods based on Viterbi algorithm are applied (17,18).

Moreover, we could maintain a ‘vocabulary’ of fragments to reduce the calculation requirement further. For example, in 4-gram counting, the frequency of “love” will be much higher than “ilov”, so “love” will be added into vocabulary. “ilov” will be disregarded. This underlines one simple rule. The DNA word should have the high frequency. We select a frequency threshold for 9-15 bps length words. All possible word of 1-9 bps are also added into vocabulary. More detailed work to construct the vocabulary from raw corpus could refer to (19,20). These are also some interesting work discussing the vocabulary in DNA (21). They all apply the similar methods.

Here is an example. A sequence in human genome is as follows:

“TGGGCGTGCGCTTGAAAAGAGCCTAAGAAGAGGGGGCGTCTGGAAGGAACCGCAACGCCAAGGGAGGGTG”

Our method will segment it into:

“TGGGCGTG/ C/ G/ CT/ TG/ AAAA/ G/ AGCCT/ AAGAA/ GAGGGGGCGTCTGGA/ AGGAA/ CC/ G/ CA/ A/ C/ GCCA/ AGGGAGGG/ TG/”

After having a DNA sequence segmentation methods, we should also set an evaluation benchmark. In natural language processing research, we normally use the precision to measure the effect of a segmentation method. It’s the ratio of number of rightly segmenting words to that of all words in the sequence. Since we didn’t know the DNA words beforehand, we design a stability indicator to evaluate the effect of DNA sequence segmentation.

For a sequence of “CCCTAAACC”, assume its segmentation is “CCC/ TAAA/ C/ C”. Then we delete the first letter of original sequence, the new sub sequence is “CCTAAACC”. If its segmentation is “CC/ TAAA/ C/ C”, it has one different “word” compared to previous sequence. But if the segmentation is “CCT/ AAA/ CC”, it will become a completely different sequence. So a good segmentation method should ensure the sub sequence is segmented into the same form with the original sequence, which explains why we set the stability as the benchmark of segmentation method.

The segmentation stability is defined as follows:

a= the number of consecutive segmentation position pairs in sub sequence which also appear in original sequence

b= the number of consecutive segmentation position pairs in sub sequence

Segmentation stability=a/b

The sub sequence is constructed by delete the first letter of original sequences.

Back to previous, sequence of “CCCTAAACC”, assume two segmenting methods botj divide it into “CCC/ TAAA/ C/ C/”. If we delete the first letter “CCTAAACC/”, its segmentation is “CC/ TAAA/ C/ C/” for first segmenting method and “CCT/AAAC/C/” for second. The first segmentation has 3 same consecutive segmentation positions pairs with original segmentation, so the stability of the first method is 1. For the second, it has 2 segmentation positions pairs, but only has one same pair with original segmentation. So its stability is 0.5. This process is illustrated in Fig.3.

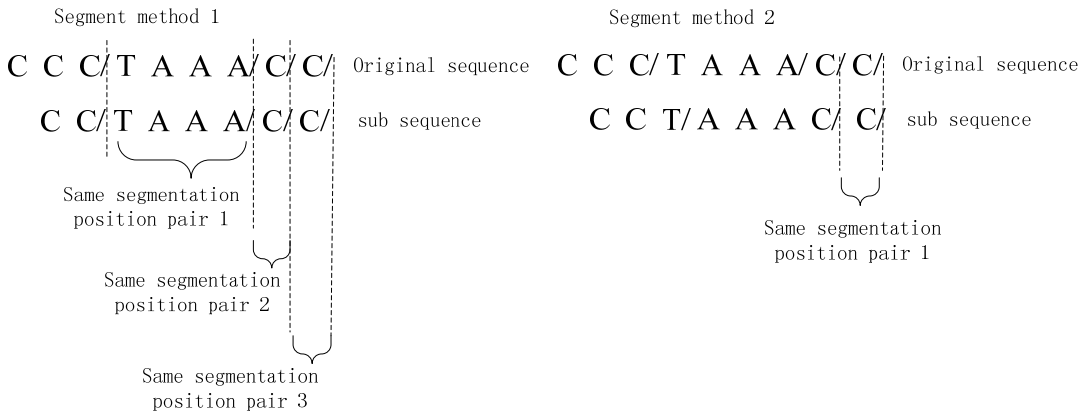

Fig 3. For segment method 1, Consecutive Segmentation pairs number in sub sequence is 3, Number of same segmentation pairs with original sequence is 3,so Stability:3/3=1. For method 2, Consecutive Segmentation pairs number in sub sequence is 2, Number of same segmentation pairs with original sequence: is ,so stability:1/2=0.5.

We randomly select a group of 70 bps length DNA sequences from each genome and run the segmentation stability test by corresponding segmentation models. The average stability ranges from 0.9 to 0.95. The stability is shown in Table.1

Table.1: segmentation stability of different genomes

genomes	Acyrthosiphon	Arabidopsis	Aspergillus	Caenorhabditis	Zebrafish	Fruit Fly
stability	0.980074	0.986467	0.973245	0.98359	0.963535	0.983323
genomes	Human	Mouse	Oryza	Schizosaccharomyces	Strongylocentrotus	Xenopus
stability	0.974546	0.965113	0.969982	0.983754	0.970433	0.973462

5 Language of different species

The previous segmentation process is based on one assumption: DNA languages vary among different species. What if different species share same languages? If so, we could build a single DNA language model for all species.

A simple feature to identify the similarity of language is the perplexity. We use the genomes of

human as experimental data and use other genome language models (15-gram) to calculate its perplexity. The results are shown in table2:

Table.2: cross perplexity of different language models

Model	Acyrthosiphon	Arabidopsis	Aspergillus	Caenorhabditis	Zebrafish	Fruit Fly
perplexity	4.32231	4.2982	5.71065	4.51787	4.08221	4.31425
Model	***Human***	Mouse	Oryza	Schizosaccharomyces	Strongylocentrotus	Xenopus
perplexity	***3.32347***	3.7709	4.15448	5.29441	4.04256	3.97019

Moreover, we could use the segment method to test the similarity of languages. We select a group of sequences from human genome and segment them by different genomes language model. Similar to the definition of segmentation stability, we could compare their number of same segmentation positions. Here we use the segmentation of human genomes models as original segmentation. Then calculate the ratio of same segmentation position pairs in other segmentation forms segmented by other genomes models. The results are shows in Table 3:

Table. 3: cross segmentation stability of different language models (15-gram model)

genomes	Acyrthosiphon	Arabidopsis	Aspergillus	Caenorhabditis	Zebrafish	Fruit Fly
stability	0.156297	0.29616	0.22841	0.248332	0.279579	0.272185
genomes	***Human***	Mouse	Oryza	Schizosaccharomyces	Strongylocentrotus	Xenopus
stability	***1***	0.43824	0.245956	0.251887	0.242873	0.322274

Since many long length words do not appear in other sequences, the stability is low for short genomes. Then we reduce the word maximal length to 9 and run this test again. The results are shown in Table.4:

Table.4 cross segmentation stability of different language models (9-gram model)

genomes	Acyrthosiphon	Arabidopsis	Aspergillus	Caenorhabditis	Zebrafish	Fruit Fly
stability	0.210119	0.395024	0.311142	0.326513	0.3877	0.363938
genomes	***Human***	Mouse	Oryza	Schizosaccharomyces	Strongylocentrotus	Xenopus
stability	***1***	0.598926	0.354348	0.345124	0.335542	0.464246

The average segmentation stability increases with lower word maximum length. If we use two completely different languages, for example, English and Chinese PinYin (represented by English letters), to segment a same sequence, such stability value will close to zero.

The cross perplexity and stability tests show the different genomes may use the similar languages, but they belong to different branches, just like the English and French/Latin in world language system. So we could build a single n-gram model and segmenting rule for all genomes. A simple way is to train a new n-gram language model by all available genomes. Here we randomly selected 100M data from 12 genomes and build a new 15-gram language model. The segmentation stability of segmenting method based on this new model is shown in Table.5:

Table.5: segmentation stability of mixed data model

genomes	Acyrthosiphon	Arabidopsis	Aspergillus	Caenorhabditis	Zebrafish	Fruit Fly
stability	0.942446	0.953038	0.949611	0.933767	0.904238	0.93521
genomes	Human	Mouse	Oryza	Schizosaccharomyces	Strongylocentrotus	Xenopus
stability	0.914045	0.898843	0.909858	0.957075	0.919044	0.92456

Although we could use the different language models for different species, a universal biology n-gram language model will bring more benefit in analyzing the DNA. More research works about the adaptation of n-gram models could refer to (22,23).

6 Some applications

After having a segmentation method, we could identify the corresponding entity or function of each “word”. This task may leave for experimental biology. Although we didn’t know the meanings of DNA words, we could still find some interesting application of DNA statistical language models. This model builds a bridge between natural language processing and DNA research. Almost all the technology in information research could be applied in DNA analyzing.

Firstly, we could build a DNA search engine like Google (24). Most current DNA search and comparing methods are similar to BLAST/FASTA algorithm, which compares one sequence with the other sequences on by one. Although many heuristic and pre-index methods could greatly reduce the search time, it’s still difficult to meet the challenge in DNA information explosion period. Many researchers agree that high performance search algorithm is demanding in the field of bioinformatics.

For mass data like Internet information, the inverted index based search systems are almost the only choice. To build a DNA search engine like Google, we only need segment the DNA sequence into ‘words’. Then using the mature search engine technology, we could easily indexing all current DNA sequences and provide the ms level search services.

The second is DNA “automatic proofreading” functions. Checking the mutant gene or mistakes in DNA sequencing is also a challenging task in bioinformatics. In search engine or word processing software, we can find correction hints when inputting the wrong words or phrase. One simple spell error check method could be designed based on vocabulary (25). Because we still have no an explicit DNA vocabulary, we could also refer to the phrase error check methods or some automatic proofreading research in East-Asia language (26,27,28). Most of these methods apply the n-gram language models. They mainly check the probability of sequence. If the probability is very low, the sequence is regarded an error sequence. Since we have built the n-gram DNA languages model, some n-gram based automatic proofreading methods could be directly applied in DNA sequence analyzing.

Methods and materials

We use the SRILM to build the language model of DNA with Good-turning as discount method (29). All genomes data are downloaded from NCBI (30). The source code of segmentation method of this paper could be found in (31).

Entropy and perplexity

The entropy is the average uncertainty of a single random variable:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \quad (1)$$

For example, the DNA sequence, $x \in \{A, T, C, G\}$, the entropy of one random variable is:

$$-(p(A)\log_2 p(A) + p(T)\log_2 p(T) + p(C)\log_2 p(C) + p(G)\log_2 p(G)) \quad (2)$$

Then for n random variables, corresponding to n length sequence, its entropy:

$$-\sum_{x_i \in X} p(x_1, x_2, \cdots x_n) \times \log_2 p(x_1, x_2, \cdots x_n) \quad (3)$$

For example, the entropy of n=2 length sequence:

$$-(p(AA)\log_2 p(AA) + p(AT)\log_2 p(AT) + p(AC)\log_2 p(AC) + p(AG)\log_2 p(AG) + p(TA)\log_2 p(TA) + \cdots + p(GG)\log_2 p(GG)) \quad (4)$$

According to Shannon-McMillan-Breiman theorem:

$$H_\infty(X) = \lim_{n \to \infty} \{-\frac{1}{n} \log_2 P(x_1, x_2, \cdots, x_n)\} \quad (5)$$

This value is defined as the language entropy. Its unit is bit. Normally, we use a very long sequence to evaluate this value.

In terms of n-gram analysis, perplexity is a measure of the average branching factor and can be used to measure how well an n-gram predicts the next juncture type in the test set. Perplexity could be calculated by entropy:

$2^{H(X)}$

Here we use the method of SRILM to calculate the perplexity. SRILM define the perplexity as:

$10^{\frac{-\log_{10} P(T)}{Word}}$, here 'T' is the sequence, 'Word' is the word number in this sequence.

Segmenting method:

The segmentation problem could be defined as:

$S = c_1 c_2 \cdots c_n$ is a sequence of DNA letters.

$W = w_1 w_2 \cdots w_m$ is a sequence of the word segmentation.

What we need is get

$$W^* = \arg\max_W P(W \mid S) \tag{6}$$

The most probable sequence of segmentation.

According to the Bayes Formula:

$$W^* = \arg\max_W P(W \mid S) \Rightarrow W^* = \arg\max_W \frac{P(W)P(S \mid W)}{P(S)} \Rightarrow \arg\max_W P(W)P(S \mid W) \tag{7}$$

Because the P(S|W) is same for all segmentations, that leaves us only maxP(W).

$$P(W) = \prod_{i=1}^{m} P(w_i) \tag{8}$$

The maximal probability segmentation method obtains a segmentation having maximal P(W). A common way to solve this problem is a dynamic programming method based Viterbi algorithm. Its main idea is as follows:

When read to letter i, we could calculate the maximal probability $P(1,i)$ till the letter i. When read to letter k, we only need calculate:

$$\begin{aligned} P(0,K) = MAX\{&P(1,k-\max Len)*p(k-\max Len+1,k),\\ &P(1,k-\max Len+1)*P(k-\max Len+2,k),\\ &\ldots\ldots P(1,k-2)*p(k-1,k)\} \end{aligned} \tag{9}$$

Here maxLen is the maximal word length. It's the dynamic programming equation of segmentation method. The time complexity of this method is O(maxLen*N), with N the sequence length.

References and Notes

1. Jerome V. Braun, Hans-Georg Muller, Statistical methods for DNA sequence segmentation. *Statistical Science*.**13(2)**,142-162(1998).
2. Chang. C.H, Chen C.D, A study on integration Chinese word segmentation and part-of-speech tagging. *Communications of COLIPS*.**3**, 69-77(1993).
3. W. J. Teahan, Y. Wen, R. McNab, I. H. Witten, A Compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, **26(3)**,375-393(2000).
4. Jin Kiat Low, Hwee Tou Ng, Wenyuan Guo. A Maximum Entropy Approach to Chinese Word Segmentation. *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 161-164(2005).
5. Sun Maosong, Shen Dayang, Benjamin K. Tsou, Chinese word segmentation without using lexicon and hand-crafted training Data. *COLING-ACL*, 1265-1271(1998).
6. Ando, R. K., Lee, L, Mostly-Unsupervised statistical segmentation of Japanese: Application to Kanji. *ANLP-NAACL*, (2000).
7. Sabine Deligne, Freddric BIMBOT, Language modeling by variable length sequences. *International Conference on Acoustics, Speech, and Signal Processing*.**1**,169-172(1995).
8. Xiaping Ge, Wanda Prat. Padhratic Smyth. Discovering Chinese words from unsegmented

text. *Proceedings on the 22 Annual International ACM SIGIR Conference On Research and Development in Information Retrieval. Berkeley CAUSA*. 217-272 (1999).

9. Peng Fuchun, Schuurmans Dale, Self-supervised Chinese word segmentation. *The 4th International Symposium on Intelligent Data Analysis, , Lisbon, Portugal.* (2001).

10. Wang, H, Zhu, J, Tang S, Fan X.A, New unsupervised approach to word segmentation. *Computational Linguistics*. **37(3)**, 421-454(2011).

11. Rosenfeld R, Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*. **88(8**), 1270-1278(2000).

12. Ganapathiraju M, Balakrishnan N, Reddy R, Klein-Seetharaman J, Computational biology and language. *Lecture Notes in Artificial Intelligence*. **3345**, 25-47(2004)..

13. Van Passel MW, Kuramae EE, Luyf AC, Bart A, Boekhout T, The reach of the genome signature in prokaryotes. *BMC Evolutionary Biology*. **6**, 84(2006).

14. Cheng BY, Carbonell JG, Klein-Seetharaman J, Protein classification based on text document classification techniques. *Proteins* . **58(4),** 955-970(2005).

15. Hatice Ulku Osmanbeyoglu, Madhavi K Ganapathiraju, N-gram analysis of 970 microbial organisms reveals presence of biological language models. *Osmanbeyoglu and Ganapathiraju BMC Bioinformatics*. **12**,12(2011).

16. Marcelo A. Montemurro, Damia´n H. Zanette, Universal entropy of word ordering across linguistic families. *PLoS ONE*. **6**, 1-9,(2011)

17. Viterbi, A.J , Error bounds for convolutional codes and an asymmetrically optimum decoding algorithm. *IEEE Transactions on Information Theory*. **13(2)**, 260-269(1967).

18. Guohong Fu,K. K. Luke, A two-stage statistical word segmentation system for Chinese. *SIGHAN 03 Proceedings of the second SIGHAN workshop on Chinese language processing*. **17**,156-159(2003).

19. Chang, J. S , Su, K. Y, An unsupervised iterative method for Chinese new lexicon extraction. *International Journal of Computational Linguistics & Chinese Language Processing*.(1997).

20. Ge, X., Pratt, W, Smyth, P. Discovering Chinese Words from unsegmented Text. *SIGIR-99*. 271-272(1999).

21. Harmen J. Bussemaker, Hao Li, Eric D. Siggia, Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis. *PNAS*. **97(18)**,10096–10100(2000).

22. Reinhard Kneser ,Volker Steinbiss, On the dynamic adaptation of stochastic language models. *In Proceedings of the IEEE conference on acoustics, speech and signal processing. Minneapolis MN*. **2**,586–589, (1993).

23. Rukmini Iyer, Mari Ostendorf, Modeling long distance dependence in language: Topic mixture vs. dynamic cache models. I*EEE Transactions on Speech and Audio Processing IEEE-SAP*.**7**, 30–39(1999).

24. Wang Liang, Fang Bo, How to build a DNA search engine like Google? *Computer Science and System Biology*. **4**, 81-86(2011).

25. K kukich, Techniques for automatically correcting words in text. *ACM Computing Survy*.**24(4)**,377-439(1992).

26. Golding A R, Schabes Yves, Combining trigram-based and feature based methods for context-sensitive spelling Correction. *Proceeding of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz*. 71- 78(1996).

27. Lei Zhang, Changning Huang, Ming Zhou, Automatic detecting/correcting errors in Chinese text by an approximate word-matching algorithm. *Proceeding ACL '2000 Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*.(2000)
28. Zhu Jinjin, Zhang Yangsen, Research and implementation on a hybrid algorithm for chinese automatic error-detecting. *International Conference on Artificial Intelligence and Computational Intelligence, Sanya*. 413 – 417(2010).
29. SRILM(SRI Language Modeling Toolkit),www.speech.sri.com/projects/srilm/
30. NCBI ftp, ftp://ftp.ncbi.nih.gov/genomes/
31. Source code of DNA sequence segmentation, http://code.google.com/p/dnasearchengine/