Protein secondary structure detection based on unsupervised word segmentation

Wang Liang[1], Zhao KaiYong[2]

Unsupervised word segmentation methods were applied to analyze protein secondary structures. Protein sequences, such as “MTMDKSELVQKA…..”, were used as input to these methods. Segmented ‘protein word’ sequences, such as ‘MTM DKSE LVQKA’, were then obtained. The protein sequence can also be ‘divided’ into segments, such as ‘MTMD KSE LVQKA’, according to its secondary structure. The boundaries of the “protein words” produced by unsupervised segmentation were clearly determined in accordance with the boundary of the secondary structure.

Word segmentation mainly refers to the process dividing a string of written language into its component words. For some East-Asian languages, such as Chinese, no spaces or punctuations are placed between letters. The ‘letter’ sequences must be segmented into word sequences to process the text of these languages. For example, ‘我爱苹果’ is segmented into ‘我 I 爱 love 苹果 apple’.

For protein sequences, such as “MTMDKSELVQKA”, the corresponding consecutive amino acids of the same secondary structure can also be regarded as a ‘word’. The segmentation, such as ‘MTM DKSEL VQKA’, is called secondary structure segmentation (**Fig. 1**). The segment in this segmentation is the ‘secondary structure word’.

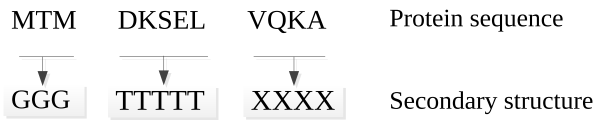

Figure 1. Secondary structure segmentation. The protein sequence is segmented by its secondary structure. This sequence contains three secondary structure words, ‘MTM’, ’DKSEL’, and ’VQKA’.

The secondary structure of a protein depends on its primary structure [1]. Thus, secondary structure segmentation can be determined using only protein sequences. Similar segmentation phenomena are also found in natural language processing. The first is how infants learn language [2]. Infants have no preliminary knowledge of languages, but can still identify word boundaries in continuous speech. The second is about the evolution of human languages [3]. A language begins with few simple ‘words’. New terms with complex semantics are eventually invented to convey more information. Finding “new” words or phrases becomes an important research topic. These methods are called unsupervised segmentation methods because they only use letter sequences.

Supervised segmentation methods can be correspondingly designed using segmented word sequences. Supervised methods are better than unsupervised methods; however, unsupervised approaches are highly adaptive to relatively unfamiliar ‘languages’ for which we do not have enough linguistic knowledge.

[1] Sogou Tech, Beijing, 100080, P.R. China. [2] Department of Computer Science, Hong Kong Baptist University, HK, 999077, P.R. China. Correspondence to Wang Liang:wangliang.f@gmail.com

This study aims to discover how to obtain segmented ‘protein word’ sequences using only amino acid letter sequences; thus, unsupervised segmentation was selected. Rules from amino acid letter sequences can be determined, and the sequences can be segmented into ‘protein word’ sequences using unsupervised segmentation. The relationship of these ‘protein words’ and their corresponding secondary structure segmentations were analyzed.

Unsupervised word segmentation can be considered as a ‘chunking’ operation, which simply is the process of identifying letter sequences that “go together” [4, 5]. The key to chunking letter sequences is to specify how a sequence of letters goes together. ‘Chunks’ are sequences with low internal entropy and high boundary entropy, indicating that letters within a chunk can be used to predict one another, but not the letters outside the chunk. The main idea of most unsupervised word segment methods depends on certain predefined criterion, e.g., mutual information, to recognize a substring in the text as a word. Candidate substrings have three basic types of goodness measurements [6, 7]:

1. Frequency [8] is the occurrence of candidate substrings in the text. Only the substrings that occur more than once are considered as qualified word candidates. Frequency mainly describes the certainty in a substring.
2. Boundary entropy [9] is the measure of the uncertainty before or after the current substring.
3. Description length gain [10] performs segmentation to maximize the compression effect, which is a global effect throughout the text.

The goodness score of a candidate correlates to the probability that it is a true word. After constructing a word list with goodness, word segmentation methods can be designed to segment the letter sequence into a ‘word’ sequence. A Viterbi-style algorithm is usually applied to search for the segmentation with the highest goodness (**Supplementary Note 2**).

Three representative unsupervised segmentation methods were selected in this study. The first only applies frequency as goodness to segment the letter sequence, which can be regarded as a baseline. The second is the Soft-counting method, which is an EM-based algorithm [11]. The last is HDP, which is based on Bayes theory [12–14].

Segmentation performance is evaluated using the F-score measure, $F = 2RP/(R + P)$. The recall R and precision P are the proportions of the correctly recognized word boundaries to all boundaries in the gold-standard and an output for word segmentation of a segmenter, respectively.

The F-scores of unsupervised segmentation methods for Chinese texts reach about 75%. The F-scores for deleting spaces between words of English text reach about 70%.

Unsupervised segmenting methods require only raw letter sequences. Data are selected from the PDB structure database (**Supplementary Note 1**). The PDB dataset contains about 100,000 protein sequences with secondary structures. These protein sequences are used as input to the unsupervised segmentation method. The corresponding secondary structure segmentations are treated as the gold-standard segmentation.

The results of the three unsupervised segmentation methods are shown in **Table 1**.

Table 1. Word segmentation evaluation

	Precision	Recall	F-score
Frequency	0.23	0.09	0.13
Soft counting	0.18	0.35	0.23
HDP	0.16	0.57	0.25

The boundary F-score of the protein corpus was much lower than that of the English corpus (normally >70%), which is mainly caused by sparse data. About 90% of the secondary structure segmentation words appear only once. The statistical features of very infrequent segments are not reliable. These 1-frequency words account for about 60% of the amino acids of all protein sequences, making the segmentation precision very low. About 50% of the words appear only once for the same amount of English corpus, but account for only about 5% of all letters.

This problem of sparse data is caused by words that are too long. The segmentation process is an encoding process, replacing the letters of a word with related word symbol. The 'Description Length (DL)' can be used to describe segmentation efficiency[15]. A codebook in which each word is represented by a unique string can be used to encode a corpus of words as efficiently as possible. The total number of letters required to encode the corpus (sum of the lengths of the codebook and encoded corpus) using a well-designed codebook would be less than the original corpus. Smaller units, such as morphemes or phonemes, which require fewer code words and thus a shorter codebook, can be encoded further. However, efficiently encoding the corpus becomes more difficult using fewer code words. Meanwhile, some words may never be used when too many words are in the codebook. Thus the length of the codebook and the length of the encoded corpus must be balanced. The DL principle states that a codebook that leads to the shortest total combined length must be chosen. This value mainly depends on the selection of maximal word length for segmentation.

The lengths of 10% of the words in secondary structure segmentation are more than 15. The 15 letters can represent 20^15, which is about 3.3e+19 distinct items. Only 4.3e+5 secondary structure words are distinct. The description length value of secondary structure segmentation is much more than the segmentations of the three unsupervised segmentation methods. Thus, using the secondary structure may not be a good encoding method.

Another explanation is that the long secondary structure words are 'compound words', which are combinations of short words. These long compound words can be divided into short words. This operation would reduce the maximal word length of secondary structure segmentation. The segmented word sequences constructed through structure segmentation enable the use of the supervised method to segment the long secondary structure words into short sub-words (**Supplementary Note 3**). Secondary structure information is used to re-segment the secondary structure segmentation. Unsupervised segmentations did not change.

The lengths of most secondary structures are less than 12. The maximal length of protein words were set as 6, 7, …, 11, and 12. The related gold-standard segmentation was rebuilt by segmenting the long words into short words. The percentage of 1 frequency words decreased with the reduction of maximal word length. The 1 frequency words accounted for about 50% of all words when the maximal word length was set at 6. Only about 8% of the letters in the corpus was covered. The description length was also reduced (**Supplementary Fig 1**).

The new gold-standard segmentations were again compared with the segmentations of the three unsupervised segmentation methods. The F-score of the unsupervised segmentation increased with the reduction of maximal word length. The F-score of most methods exceeded 0.4 when the maximal word length was set at 9 (**Supplementary Fig 2**). The best F-scores of the three methods are shown in **Table 2**.

Table 2. Evaluation of word segmentation (set maximal word length)

	Precision	Recall	F-score
Frequency	0.44	0.22	0.29
soft counting	0.43	0.40	0.41
HDP	0.42	0.56	0.48

Table 2 clearly shows the consistency of the distinction between the ‘protein word’ and the secondary structure. The word segmentation methods are ‘learned’ only from raw protein sequences. This statement may explain how protein primary structure governs its secondary structure.

References and Notes

1. Anfinsen C B. Principles that govern the folding of the protein chains. Science. **181**, 223-227(1973).
2. Batchelder. Bootstrapping the lexicon: A computational model of infant speech segmentation. Cognition. **83**,167–206(2002).
3. Ruey-Cheng Chen, Chiung-Min Tsai, Jieh Hsiang. Regularized compression method to unsupervised word segmentation. Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology. **Canada**, 26–34(2012).
4. Paul Cohen, Niall Adams, Brent Heeringa. Voting Experts: An unsupervised algorithm for segmenting sequences. Intell. Data Anal. **11**,607-625(2007).
5. Daniel Hewlett, Paul Cohen. Word segmentation as general Chunking. Proceedings of the Fifteenth Conference on Computational Natural Language Learning. **USA** , 39–47(2011).
6. Hai Zhao, Chunyu Kit. Exploiting unlabeled text with different unsupervised segmentation criteria for Chinese word segmentation. Proceedings of the Conference on Empirical Methods in Natural Language Processing. **Israel**, 17-23(2008).

7. Hai Zhao,Chunyu Kit. An empirical comparison of goodness measures for unsupervised Chinese word segmentation with a unified framework. The Third International Joint Conference on Natural Language Processing. **India 1**, 9-16(2008).
8. Xueqiang Lv, Le Zhang, Junfeng Hu. Statistical substring reduction in linear time. Proceeding of the 1st International Joint Conference on Natural Language. **China**, 320–327(2004).
9. Zhihui Jin, Kumiko Tanaka-Ishii.Unsupervised segmentation of Chinese text by use of branching entropy. COLING/ACL. **Australia**, 428–435(2006).
10. Chunyu Kit, Yorick Wilks. Unsupervised learning of word boundary with description length gain. CoNLL. **Norway**, 1-6(1999).
11. 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. **USA**, 217-272(1999).
12. Sharon Goldwater, Thomas L Griffiths, Mark Johnson. Contextual dependencies in unsupervised word segmentation. ACL-44. **Australia**, 673–680(2005).
13. Sharon Goldwater, Thomas L Griffiths, Mark Johnson. A bayesian framework for word segmentation: Exploring the effects of context. Cognition. **112**,21–54(2009).
14. Mochihashi D, Yamada T, Ueda N. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. ACL-47. **Singapore**,100–108(2009).
15. Sharon J Goldwater. Nonparametric bayesian models of lexical acquisition. Thesis: Brown University, (2007).

ONLINE METHODS

Data source. Unsupervised segmenting methods only need raw letter sequence. We mainly use the data of PDB (http://www.rcsb.org/pdb/) as our experiment data. This dataset contain about 100,000 pieces of protein sequences. We also use protein sequence data of website “uniprot.org” (http://www.uniprot.org/downloads), which is a central repository of amino acid sequence. We select the UniProtKB/Swiss-Prot and UniProtKB/TrEMBL. We use CD-HIT algorithms to delete the similar protein sequence. Its codes could be found in : http://weizhong-lab.ucsd.edu/cd-hit/download.php

Unsupervised segmentation methods. Most unsupervised methods are designed based on Expectation Maximization (EM) method, an iterative method for finding maximum likelihood estimates of parameters in statistical models. By this methods, we could evaluate the probabilities or other goodness of all possible words.

n-multigram language model is a typical model for EM based methods, here the n is the maximal word length. This model assumes the likelihood of a sequence is the sum of likelihoods of its all forms of segmentations. For example, a sequence ‘AATD’, assume the maximal word length is 3. Then its likelihood:

$$L(\mathrm{AATD}) = \mathrm{sum}\begin{bmatrix} p(\mathrm{A})\,p(\mathrm{A})\,p(\mathrm{T})\,p(\mathrm{D}) \\ p(\mathrm{A})\,p(\mathrm{A})\,p(\mathrm{TD}) \\ p(\mathrm{A})\,p(\mathrm{AT})\,p(\mathrm{D}) \\ p(\mathrm{AA})\,p(\mathrm{T})\,p(\mathrm{D}) \\ p(\mathrm{AA})\,p(\mathrm{TD}) \\ p(\mathrm{AAT})\,p(\mathrm{D}) \\ p(\mathrm{A})\,p(\mathrm{ATD}) \end{bmatrix}$$

, here, p(AT) is the probability of word 'AT'

Then for EM method:

In initial step, it give the random or uniform initial probability for all words firstly. Here we set probability as 0.1 for all the words.

For E-step, we calculate the probability of all forms of segmentation. For example above, in the first round, the probability of every word is 0.1, so:

$$p(\mathrm{A})\,p(\mathrm{A})\,p(\mathrm{T})\,p(\mathrm{D}) = 0.1*0.1*0.1*0.1 = 0.0001$$

$$p(\mathrm{AA})\,p(\mathrm{T})\,p(\mathrm{D}) = 0.1*0.1*0.1 = 0.001$$

……

The sum of all probabilities is 0.0331, which is the sequence probability. We should normalize this value to 1. So the probability of each form of segmentation becomes:

$$p(\mathrm{A})\,p(\mathrm{A})\,p(\mathrm{T})\,p(\mathrm{D}) = 0.0001/0.0331 = 0.003$$

$$p(\mathrm{AA})\,p(\mathrm{T})\,p(\mathrm{D}) = 0.001/0.0331 = 0.03$$

……

For M-step, we evaluate the words probability according to segmentation forms above.

For $p(\mathrm{A})\,p(\mathrm{A})\,p(\mathrm{T})\,p(\mathrm{D}) = 0.003$, Every word 'A','A','T','D' in this segmentation get the probability of 0.0001.

For $p(\mathrm{AA})\,p(\mathrm{T})\,p(\mathrm{D}) = 0.03$, every word 'AA', 'T', 'D' gets probability of 0.001.

Every segmentation forms all assign its probability to its every words. Then for each word, we add its probability together. For word 'A', its probabilities sum is 0.3988, the 'AA' is 0.3323, etc.

We do the same thing for every sequence in test data. Then for each word, we add its probabilities together as the probability of this word. We should also normalize the probabilities of all words to ensure the sum of all words probabilities is 1.

We repeat the E-step and M-step until the probability of words don't change anymore. Finally, we could get a vocabulary with word probability.

After obtaining a list of word candidates each associated with a goodness like probability:

$W=\{\{\omega_i,\mathrm{g}(\omega_i)\}_{i=1,\ldots,\mathrm{n}}\}$,here ω_i is a word candidate and $\mathrm{g}(\omega_i)$ is a goodness function.

To find the optimal segmentation of a given letter sequence, a Viterbi-style one to search for the best segmentation S^* for a text T:

$$S^*=\underset{\omega_1\cdots\omega_i\cdots\omega_n=T}{\arg\max}\sum_{i=1}^{n}g(\omega_i),\{\omega_i,\mathrm{g}(\omega_i)\}\in\mathrm{W}$$

Description length. Description length is the empirical description length of a corpus in bits that can be estimated by the Shannon-Fano code or Huffman code as below, following classic information theory. It can be formulated in terms of token counts in the corpus as below for empirical calculation:

The description length of a corpus $X=x_1x_2\cdots x_n$ a sequence of tokens(e.g., letters, words), its description length:

$$DL(X)=-n\sum_{x\in V}p(x)\log\mathrm{p(x)}=-\sum_{x\in V}c(\mathrm{x})\log\frac{c(x)}{|\mathrm{X}|}$$

Where V is the set of distinct tokens in X and $\mathrm{c}(x)$ is the count of x in X .

For segmentation method, its description is:

$DL(X[\mathrm{r}\rightarrow\mathrm{s}]\oplus\mathrm{s})$, where r is an index, $X[\mathrm{r}\rightarrow\mathrm{s}]$ represents the resultant corpus by the operation of replacing all occurrences of s with r through out X and $\oplus$ represents the concatenation of two things.

Supervised segmentation methods. Because we have obtained the segmented protein word sequence, we could simply count the occurrence of each word and use frequency as goodness to design the Viterbi-style segmentation method.

Experiment results:

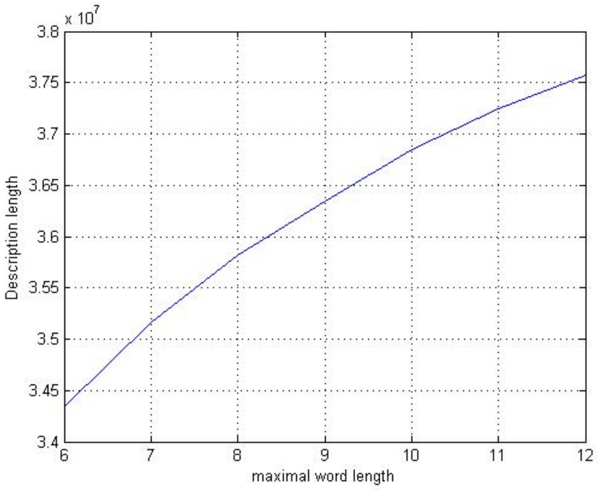

Figure 1. Relation of maximal word length and description length

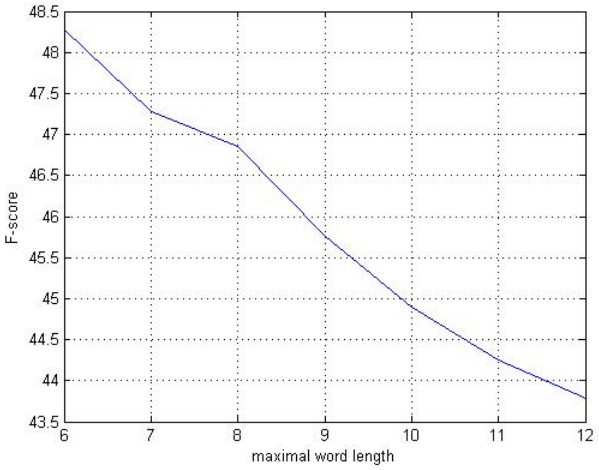

Figure 2. Relation of maximal word length and segmentation F-score (HDP segmentation method)

Source code. The source codes of this paper could be found in: https://code.google.com/p/dnasearchengine/.