일관된 모티프 발견을 위한 분할 깁스 샘플링 알고리즘

히키 크리스°, 장병탁
서울대학교
{chickey, btzhang}@bi.snu.ac.kr

Split Gibbs Sampling Algorithm for Consistent Motif Discovery

Chris Hickey, Byoung-Tak Zhang
Seoul National Univeristy

요 약

Motif discovery is a widely studied problem in bioinformatics, used to discover common patterns across divergent biological sequences. Gibbs sampling is commonly used to perform motif discovery. However Gibbs sampling is often susceptible to local maxima issues, meaning that the algorithm may sometimes fail to converge on discovery of the most optimal motif in a group of biological sequences. The present study shows how by running Gibbs sampling in batches of sequences and averaging out the results of each batch, consistent motifs can be discovered across separate, independent executions of the algorithm. This method was shown to be effective, regardless of motif size.

1. Introduction

Gibbs sampling is a Markov Chain Monte Carlo method commonly used to simulate distributions that are difficult to sample from directly. This process has been widely used over the past number of decades for various use cases from data augmentation to parameter estimation [1]. However one of the most prominent use cases of the Gibbs sampling algorithm is the local multiple sequence alignment problem, aimed at finding protein motifs in Bioinformatics. Protein sequences are made up of long chains of amino acids, with each amino acid being denoted by one of the following 20 characters; *A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y.* Common patterns found across proteins allow for insight into molecular function and evolution of biological sequences. However divergence of these sequences over time means finding similarities between sequences is a challenging problem. Local multiple sequence alignment refers to the problem of finding short, common patterns of amino acids or other DNA sequences shared by otherwise dissimilar biological sequences.

The Gibbs sampling algorithm allows for local alignment motif models for N sequences to be found in N-linear time [2, 3]. However, a notable weakness, which is acknowledged by the original authors of the seminal paper on Gibbs sampling, is that the Gibbs sampling procedure is somewhat susceptible to local maxima issues [4]. This means that separate iterations of the Gibbs sampling algorithm may give different results. The present study aims to augment the Gibbs sampling method outlined in [2] to provide more consistent motif discoveries which can be consistently reproduced across the same group of N-sequences multiple times, thus providing a better estimate for the best local multiple sequence alignment motif model for those N-sequences.

2. Method

The present study examines two variants of the Gibbs algorithm[1] on three groups of protein sequences; COG1, COG160 and COG161. This results in a total of N equal to 202 sequences across these 3 protein categories. Data for these sequences can be found at [5]. Two variants of the Gibbs sampling algorithm are run against these sequences in order to find motifs of W length 3, 4 and 5. The two variants of the algorithm are as follows; the standard Gibbs sampling algorithm, and the present studies proposed split variation of the Gibbs sampling algorithm.

The standard Gibbs algorithm as outlined in [2] works as follows - first a random start position for the motif is chosen for all N sequences. Then two simultaneous data structures are updated in order to find the best motifs. The first structure maintains a probabilistic model of the frequencies of each character at each motif position. The second structure is a set of indices indicating the starting position of the most probable motif in each of the N sequences. The algorithm then updates these structures iteratively through two steps - the predictive update step and the sampling step. In the predictive step, one z sequence of the N sequences is chosen. Then, the probabilistic model data structure is updated using the following equation;

$$q_{i,j} = \frac{c_{i,j} + b_j}{N - 1 + B} \tag{1}$$

where $c_{i,j}$ is the count of character j at position i given the cur-

[1] `https://github.com/chrishickey/Split_Gibbs`

	Original Gibbs			Split Gibbs		
Executions	**3-motif**	**4-motif**	**5-motif**	**3-motif**	**4-motif**	**5-motif**
1st	GYH	YHGH	HGHTH	YHG	YHGH	GYHGH
2nd	GYH	YIDY	MSAIR	YHG	GYHG	YHGHT
3rd	YHG	ADEV	IADEV	YHG	GYHG	GYHGH
4th	EPV	HGHS	GHGHP	YHG	GYHG	GYHGR
5th	GYH	GGYH	GHSHP	YHG	YHGH	GYHGH

표 1: Results showing the most probable motifs found using both original and split Gibbs algorithms.

rent best motif starting position across all sequences, and b_j and B represent both pseudocounts and sum of pseudocounts respectively. The sampling step then considers ever possible motif starting position in sequence z, and probabilistically chooses a starting motif position based on the current $q_{i,j}$ probabilities calculated from all other sequences. In the present study's implementation of the standard Gibbs algorithm, this iterative process was repeated for 150 iterations.

The proposed split Gibbs algorithm randomly splits the N sequences into six roughly equal groups. Next, the standard algorithm outlined above is run multithreaded against each of these six mini batches for 50 iterations. Models from these six batches are then combined into one model by averaging out motif character probabilities across the 6 mini models. This entire process is then repeated for 25 iterations.

3. Results

The most probable 3, 4 and 5 character motifs found through 5 executions of both the standard and split iterations of the Gibbs sampling algorithm are summarized in Table 1. While some consistent motif patterns were discovered using the original Gibbs sampling method, specifically “HGH”, “YHG” and “GYH”, the large variation of results between different executions of the algorithm indicate that the algorithm was often converging on local maxima solutions.

Conversely, the split version of the Gibbs sampling algorithm found the exact same “YHG” motif pattern in all 15 executions of the algorithm. This consistent pattern of discovering the exact same 3-character motif or sub-motif across multiple separate executions suggests that the split version of the Gibbs sampling algorithm is better equipped to find consistent global maximums, discovering a consistent single motif pattern regardless of motif length.

4. Conclusion

Gibbs sampling is a method widely used in bioinformatics for motif discovery in partially similar but divergent biological sequences. The present study shows how just a small modification to the standard bioinformatics Gibbs Sampling algorithm can make a significant contribution to addressing the local maxima problem and make the algorithm discover consistent motifs across separate executions of the algorithm. Future studies should investigate the effectiveness of split Gibbs sanpling across different more challenging biological sequence sets.

5. Acknowledgements

This work was partly supported by the Institute for Information & Communications Technology Promotion (2015-0-00310-SW. StarLab, 2017-0-01772-VTT, 2018-0-00622-RMI, 2019-0-01367-BabyMind) and Korea Institute for Advancement Technology (P0006720-GENKO) grant funded by the Korea government.

참고 문헌

[1] E. A. Suess and B. E. Trumbo, *Introduction to probability simulation and Gibbs sampling with R*. Springer Science & Business Media, 2010.

[2] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, “Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment,” *science*, vol. 262, no. 5131, pp. 208–214, 1993.

[3] S. Kim, Z. Wang, and M. Dalkilic, “igibbs: Improving gibbs motif sampler for proteins by sequence clustering and iterative pattern sampling,” *Proteins: Structure, Function, and Bioinformatics*, vol. 66, no. 3, pp. 671–681, 2007.

[4] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[5] R. L. Tatusov, M. Y. Galperin, D. A. Natale, and E. V. Koonin, “The cog database: a tool for genome-scale analysis of protein functions and evolution,” *Nucleic acids research*, vol. 28, no. 1, pp. 33–36, 2000.