Gibby: A Gibbs Sampling Motif Finding Package

Joseph Hwang Kairi Tanaka

Introduction

Although Gibbs Sampling is often discussed as a valid method for discovering motifs, it is quite difficult to find popular tools today that utilize this algorithm. As a result, we decided to develop our own tool. Gibby is a python package that utilizes Gibbs Sampling in order to identify the motif of a given transcription factor de novo based on its ChIP-seq data. The main goal was to create a tool that successfully identifies the motifs of a given transcription factor based on its ChIP-seq data, and explore the feasibility of Gibbs Sampling when applied to motif discovery.

Methods

General Implementation

The Gibby pipeline begins with the user providing a genome assembly file and peak file as input for the tool. Note Gibby allows HOMER peak files and BED files. The peak file is filtered to contain peaks above a certain score threshold to only allow high quality peaks; for HOMER files, the default threshold is 60, and 999 for BED files. In addition, the tool requires a genome assembly FASTA file which is read using a Biopython SeqIO package. Gibby then extracts the peak sequences of the transcription factor using the genome assembly and peak file. The extracted peak sequences are output as a list which is then input into our Gibbs Sampler whose default parameters are 500 iterations of kmer length 20.

Running the Gibbs Sampling algorithm to completion, a list of kmers representing the most conserved subsequences among the peak sequences are then output as a list. Gibby then constructs a position weight matrix using these kmers. Finally, the position weight matrix is visualized using the seqLogo package to identify the most common pattern that occurred, ultimately representing the consensus motif.

Gibby outputs four files after completing its task: “PFM.txt”, “PWM.txt”, “gibbs_potential_bind.txt”, and “motif.png”. These files include the position frequency matrix, position weight matrix, and potential motifs that were generated by Gibbs Sampling, and a png file visualizing which motif was most strongly conserved among the peak regions.

Benchmarking Methods

To benchmark Gibby, we compared its runtime with three other tools: HOMER[3] (v4.11), MEME[1] (v5.5.5), and RSAT[2] (info-gibbs: 20140213). We used a ChIP-seq dataset for ZNF24 from ENCODE[4] which used the GRCh38 genome assembly.

For Homer (findMotifsGenome.pl), all default parameters were used with the addition of the “-mask” option. For MEME (https://meme-suite.org/meme/tools/meme-chip), all default parameters were used. For RSAT (http://rsat.sb-roscoff.fr/info-gibbs_form.cgi), we had to convert our BED file to fasta format; coincidentally, MEME automatically generates this file when running its process, so the same file was used. All default parameters were used except matrix length set to 20 and maximum number of iterations set to 500. All default parameters were used for Gibby.

To compare accuracy, we compared Gibby’s motif with HOMER’s motif using the lab 5 ChIP-seq datasets for the transcription factors OCT4, KLF4, and SOX2. For both tools, the GRCm38 genome assembly and the same HOMER “peaks.txt” files (that had been generated in lab 5) were used. For HOMER, the default parameters were used with the addition of the “-mask” option and “-size 100”. For Gibby, all default parameters were used.

Overview of Gibbs Recursive Sampler

Gibbs Sampling is a statistical method used to estimate the distribution of variables when direct sampling is difficult. It is particularly useful in motif finding where we want to identify common patterns in a set of sequences.

For example, imagine you are trying to perfect a secret recipe, but you don't have all the ingredients at once. You start by randomly choosing some ingredients and proportions, then you taste the result. Based on how good or bad it tastes, you keep some ingredients, change others, and try again. Each iteration helps you understand what works and what doesn't—gradually leading you to the best recipe. In the context of genomic sequences, Gibbs Sampling helps us identify common patterns by iteratively refining guesses based on the resulting sequences (and their scores). Each iteration, which involves some randomness, helps to gradually reveal the underlying motifs more accurately.

Gibbs Sampling Implementation

Given S sequences, we aim to find the most mutually similar subsequences of length k from each sequence. The process involves iterative optimization using a probabilistic approach to find a local minimum that represents the most mutually similar subsequences.

Initially, we randomly choose a starting position for a length k subsequence in all S sequences. The randomness allows for a diverse selection of subsequences, ensuring that the search space is independent and unbiased.

In each iteration, one sequence *s'* is left out, and a position weight matrix (PWM) is constructed by using the counts of the remaining position *k* subsequences in the *S-1* subsequences. The PWM is a *4*x*k* matrix representing the frequency of each nucleotide at each position of the *k*-mer. To avoid zero frequencies, pseudocounts *p* are added to each count.

For the left out sequence *s'*, the probability of every possible subsequence of length *k* is calculated using the PWM. These likelihoods are then normalized to get a probability distribution over all the starting positions.

A new starting position for the subsequence in *s'* is sampled according to the probability distribution from above. Then this new position replaces the old position in this sequence which has allowed us to sample the entire landscape without bias from this sequence.

After updating the positions, the new set of motifs is scored. If the new score is better than the score of the previous iteration the new set of motifs are kept.

The iteration process is repeated, typically 500 iterations in Gibby, until the positions of the subsequences even out. Through this iterative refinement, the algorithm converges to the most mutually similar subsequences, revealing conserved motifs across the sequences.

Results

Benchmarking Results

To benchmark the accuracy of Gibby, we compared its motif results with HOMER's results for the ChIP-seq datasets of OCT4, KLF4, and SOX2 from lab 5. In general, Gibby was able to successfully discover motifs that were very similar to the motifs found by HOMER and also to the published motifs on HOCOMOCO as shown in figure 1.

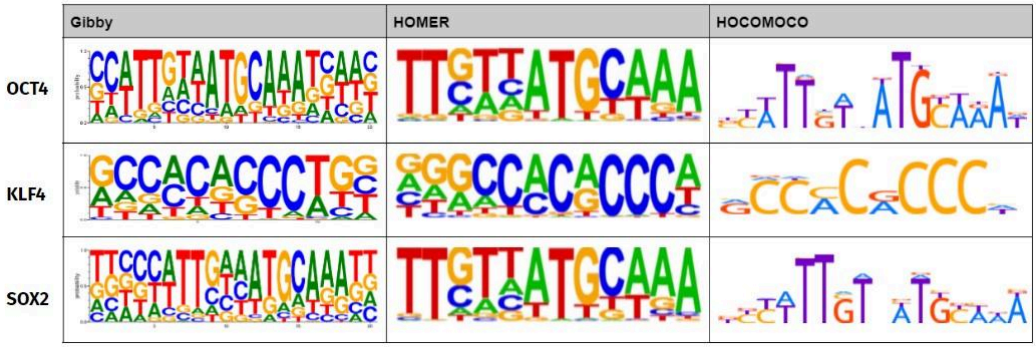

Figure 1. *Motif by Gibby (left), HOMER (center), and published motifs on HOCOMOCO (right)*

Note that since Gibbs Sampling is a stochastic process, the logos will look different for every run of the tool. What *should* be similar for each run are the large letters stacked at the top and their relative positioning to one another. The large sets of letters represent the strongly conserved motifs that Gibby discovered; the other smaller letters represent "noise" or

nucleotides that were not as strongly conserved among peak regions. Gibby may output the forward or reverse complement of the conserved motif since we do not pre-process the peak sequences to only contain a certain direction of strand.

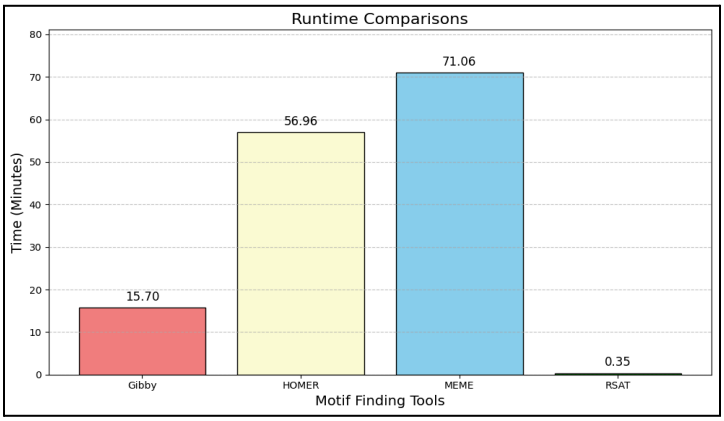

Figure 2. *Runtime comparisons using a ChIP-seq dataset for ZNF24 from ENCODE.*

Based on Figure 2, it initially appears as if HOMER and MEME are substantially slower than tool and RSAT; however, we hypothesize that the longer runtime is due to how HOMER and MEME discover multiple conserved motifs, in contrast to Gibbs Sampling which only finds the single most conserved motif. As a result, the most fair comparison was between Gibby and RSAT. RSAT was found to be about 50 times faster than our tool for the same dataset.

Other Dataset Analysis

ZNF24, also known as Zinc Finger Protein 24, is a transcription factor that plays a crucial role in the regulation of gene expression. Recent studies have revealed the important functions of ZNF24 in regulating cell proliferation, differentiation, migration, and invasion, as well as tumor angiogenesis. To investigate the binding motifs of ZNF24, we utilized a ZNF24 Chip-Seq BED file sourced from the ENCODE project. Figure 3 shows how Gibby was able to discover the correct motif.

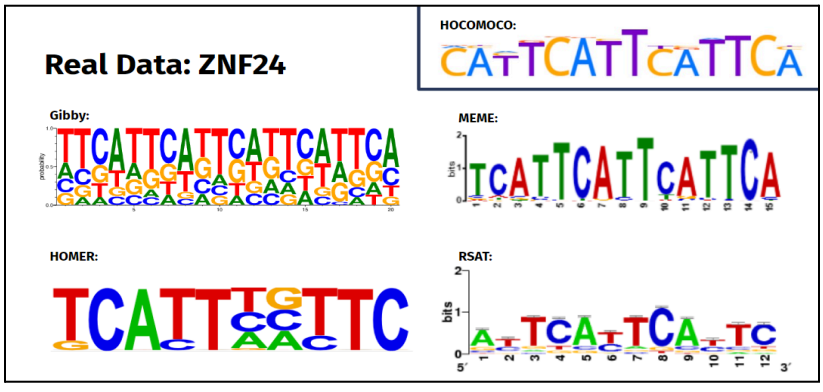

Figure 3. *Motifs found by Gibby, HOMER, RSAT, and MEME; published motif from HOCOMOCO is on the top right. Motifs are similar across the board.*

Discussion

There were many challenges we faced throughout development. One significant challenge we encountered during our analysis was when running Gibby on ChIP-seq datasets for transcription factors that had multiple possible motifs. For example, STAT1 is a transcription factor known to bind to six distinct motifs. We realized that the presence of multiple motifs often diluted the signal for each individual motif, making it difficult for Gibbs Sampling. In multiple runs of Gibby to discover the motif for STAT1, we observed how the algorithm converged on an incorrect motif, as seen in Figure 4. We suspect this issue may be a limitation of Gibbs Sampling because RSAT was also unable to identify the correct motif.

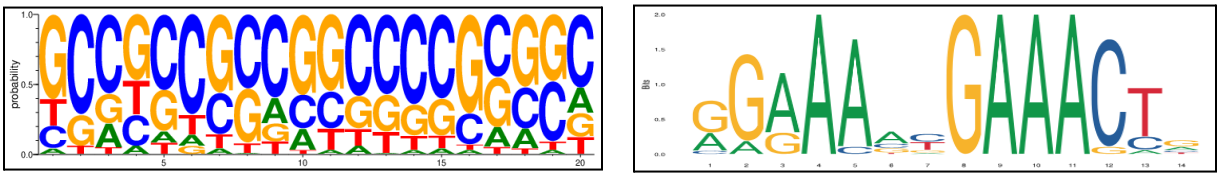

Figure 4. *Gibby motif (left) and one of multiple JASPAR published motif (right) for STAT1*

Another challenge we faced was selecting the appropriate parameters for Gibby. Parameters, including the length of the motifs (k), number of iterations, and score thresholds, are important components which affect the runtime and accuracy of Gibby. Too few iterations negatively affect the accuracy, too many iterations substantially increase the runtime; too low score thresholds introduces noise, while too high thresholds result in too little peaks; too short kmer lengths result in missing the motif, too long–again–increases the runtime. A series of trial and errors were made to find a set of parameters that would give Gibby a good balance between runtime and accuracy.

In conclusion, we developed Gibby, a Python package that uses Gibbs Sampling to identify transcription factor motifs de novo from ChIP-seq data. We validated Gibby using multiple datasets and found that the discovered motifs closely matched published motifs, confirming the tool's accuracy. In addition, we identified the strengths and weaknesses of Gibbs Sampling when applied to motif finding. Gibbs Sampling–although fast–has difficulty identifying motifs for transcription factors with multiple equally possible motifs. Ultimately, in terms of future directions, it is important that we look into refining or modifying our implementation of Gibbs Sampling in order to avoid or solve these issues.

Code Availability: https://github.com/kairitanaka/gibby

References:

1. Philip Machanick and Timothy L. Bailey, "MEME-ChIP: motif analysis of large DNA datasets", *Bioinformatics* **27**(12):1696-1697, 2011. [full text]
2. Defrance M, van Helden J. (2009) *Info-gibbs*: a motif discovery algorithm that directly optimizes information content during sampling. Bioinformatics 25(20):2715-22. [Pubmed 19689955] [Open access]
3. Heinz S, Benner C, Spann N, Bertolino E et al. Simple Combinations of Lineage-Determining Transcription Factors Prime cis-Regulatory Elements Required for Macrophage and B Cell Identities. Mol Cell 2010 May 28;38(4):576-589. PMID: 20513432
4. ENCODE:

Epitope	PEAK accession
ZNF24	ENCSR072PWP
STAT1	ENCFF973ACG

5. HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis