Bioinformatics I
WS 15/16
Tutor: Alexander Seitz

1	2	$\sum$ (6)

Jonas Ditz
& Benjamin Schroeder

Assignment 7

(Handed in 30. November 2015)

Practical Assignment - *Planted Motif Problem Generator*

A executable .jar-File of the written program for this task is attached to the our hand-in. The .jar can be executed via a commandline with the command "java -jar SeqGen.jar". The help for usage can be called with the option "-h", "-help", "–help" or "-?". Generally we tried to be as user-friendly as possible, via using options as input markers and generating detailed and clear error messages, if a input is invalid or missing. General needed input values are the options "-k <int>" ammount of sequences in the final set, "-n <int>" the sequence length, "-d <int>" the amount of deviation positions and "-l <int>" the length of the motif or "-M <String>" a motif itself. If "-l <int>" is entered, a random motif is generated. With "-M <String>" the String is used as motif and "-l" is of course not needed, and vice versa. The "-o <String>" option is optional and stand for the output name. If no option is added the test run is started with the default values from the assignment sheet.
The motifs are generated with d deviation positions, which are not independent from each other. So if d is 2, there are two variable positions in the motif. These can not fall onto the same position. But each position can deviate as one of all four basic nucleotides, so also as the one which was already contained.
Another feature of our program is, that it also generates sequences with based on an amino acid alphabet. Switching to the amino acids the option "-E p" is needed. For the basicDNA bases no option is needed, although there is the "-E n" option. Also a matter of user-friendliness is the fact of never overwriting the output, but instead counting up an number. This causes the comfort of never overwriting accidently a file.

Practical Assignment - *MEME and Gibbs sampler*

Method

We used two different software solution of Motif finder to perform a Motif search on our test sets. The first tool was MEME-suite [1] a set of Motif-based sequence analysis tools. For this task we used the online implementation of MEME[1] with the site distribution parameter set to “One occurrence per sequence” and the number of Motifs set to one. On the other hand we used the Gibbs Motif Sampler [2]. All parameters were set to their default values except we set the -n flag to use nucleic acid alphabet.

All test sets of sequences used for this task were generated by SeqGen. This is a self-programmed Java tool, which uses four parameters to build a set of sequences. A overview of important

[1] http://meme-suite.org/tools/meme

parameters of SeqGen can be found in table 1. We generated five different test sets (see table 2), which we used for comparison of MEME and Gibbs sampler.

Table 1: Parameters of SeqGen

Parameter	Description
k	number of sequences
n	length of each sequence
l	length of Motif
d	number of deviating positions
M	specify a Motif (optional)

Table 2: parameter, which was used to generated test sets

	k	n	l	d	M
test set 1	20	100	10	2	AGTGGAACAG
test set 2	15	150	15	3	CTTTGAGCAAATAAT
test set 3	20	100	5	1	ATATC
test set 4	25	50	5	2	CTGCA
test set 5	25	100	10	1	ACAGGGGTGC

Result

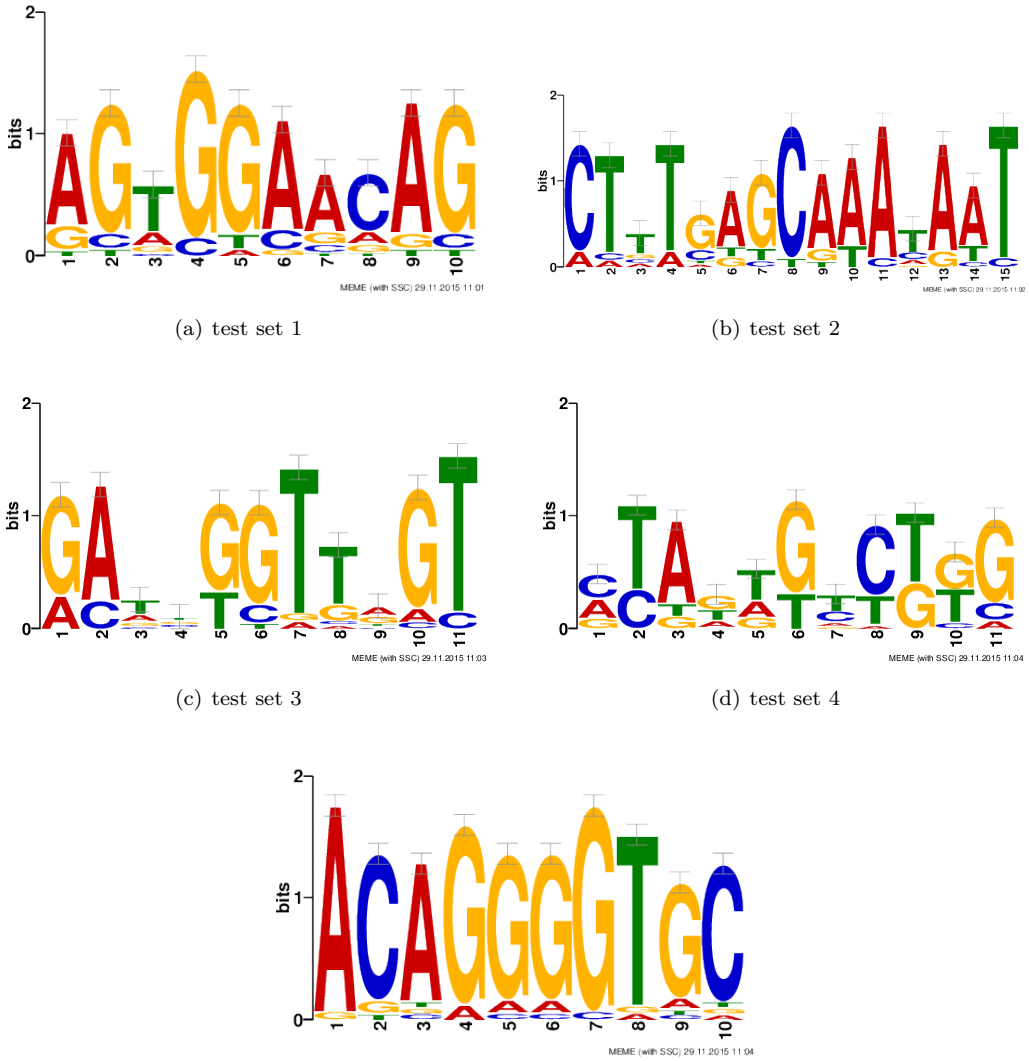
MEME was able to find a Motif for all of the five test sets (see Figure 1). However, one can see easily that it overestimated the length for short Motifs.

Even worst was the result of Gibbs sampler. This program was only able to find the Motif in test set two and five. For all of the three other sets, Gibbs sample could not find any Motif in the sequences (see table 3).

Table 3: Detected Motifs by Gibbs sampler. The letter N means that Gibbs sampler did not give a suggestion for this position.

	Motif
test set 1	*No Motifs detected*
test set 2	ANTNTTNGCTNANAG
test set 3	*No Motifs detected*
test set 4	*No Motifs detected*
test set 5	TACAGGGGTNC

If we compare the runtime of both programs (see table 4) one can see that MEME is in general a little bit faster. But since we used the webservice of MEME and run Gibbs sampler on a local machine with a single core CPU and just one Gb of RAM, the comparability of theses runtimes may not be given.



(a) test set 1

(b) test set 2

(c) test set 3

(d) test set 4

(e) test set 5

Figure 1: Sequence logo of calculated Motifs by MEME

Discussion

With the results given above one can see that both, MEME and Gibbs sampler, have a problem with Motifs of too small size. Since the Motif finding problem is not statistically solvable for small Motifs [3], this observation makes sense. Also MEME seems to have a higher sensitivity, since it still find a Motif for test set three and four but a wrong one. Obviously for these wrong Motifs the position found by MEME were also wrong but for all other Motifs MEME found the exact positions. On the other hand Gibbs sampler has a lower sensitivity than MEME and it also did not output the positions of the founded Motifs. So it is difficult to proof that Gibbs sampler found the right Motifs.

Under the circumstances discussed above and with respect to the much more user-friendly interface of MEME-Suite we would recommend to use that website.

Table 4: Runtime of used programs in Seconds

	MEME	Gibbs sampler
test set 1	1.66	1.04
test set 2	3.33	1.13
test set 3	1.66	1.05
test set 4	1.66	0.78
test set 5	3.33	1.19

References

[1] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, pp. 28–36, 1994.

[2] T. W, R. EC, and L. CE, "Gibbs recursive sampler: finding transcription factor binding sites," Nucleic Acids Res., vol. 31, no. 13, pp. 3580–3585, 2003.

[3] K. Nieselt and D. Huson, "Bioinformatics 1: On sequences, genes, proteinsand genomes," University of Tübingen, 2015.