Multilenkmer Overview

Ramon Schöndorf

1 Introduction

ChipSeq data can be used to determine transcription factor binding site motifs (Johnson et al., 2007). Many algorithms rely on peak calling to identify the genome position, that most likely contains the binding Motif (Wilbanks and Facciotti, 2010). Peak calling approaches have been found to be susceptible to noisy data which impairs reproducibility (Nakato and Shirahige, 2016). Menzel et al. (2020) proposed NoPeak as an alternative to peak calling. Instead of focusing on peaks of accumulated reads, which may or may not be distinguishable from background noise, depending on the dataset, NoPeak uses the spatial distribution of reads around each kmer to build and score kmer-specific profiles. High scoring of a kmer profile might have biological meaning (Menzel et al., 2020). NoPeak generates sequence logos based on clustered, similar kmers. The position count matrix required for this step is derived from the alignment of kmers of one cluster to their highest scoring member. So far only kmers of the same length had been used for the sequence logo generation. Multilenkmer re-implements the logo generation step from NoPeak with the possibility to use datasets containing kmers of different lengths. The visualization of the generated sequence logos is extended by a color coded alignment matrix placed underneath the logo.

2 Objective and Implementation

Multilenkmer is intended to expand on the previously implemented sequence logo generation step of NoPeak. Kmers of multiple lengths from different NoPeak-runs are used to generate sequence logos. The script starts with the overall highest scoring kmer and aligns successively lower scoring kmers to the consensus of all previously aligned kmers. Alignments are done using the local alignment algorithm implementation from Biopython (Cock et al., 2009).

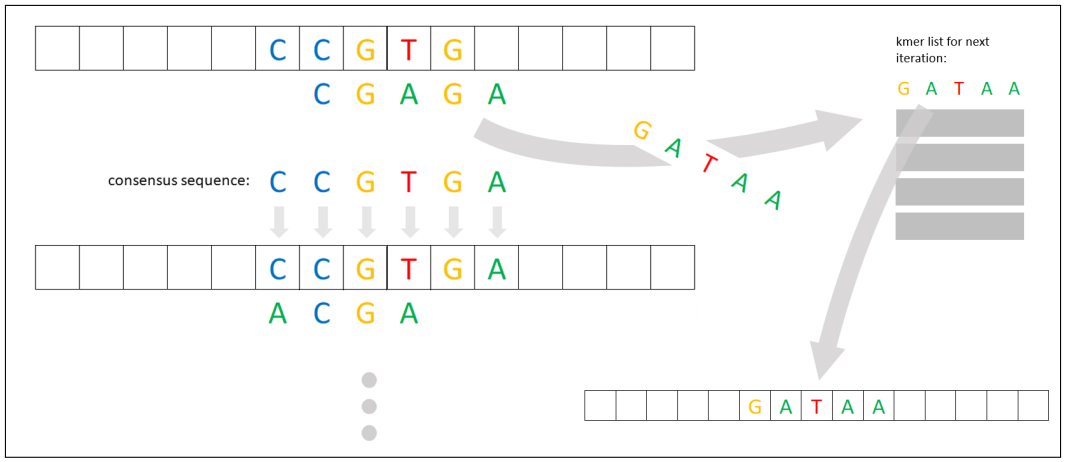

Figure 1: Alignment of kmers to, and generation of consensus sequence. The multiple sequence alignment (MSA) for the generation of a position count matrix, required for sequence logos, is derived from this alignment process. When a kmer is added, its position in the MSA is given by the position relative to the array holding the consensus sequence, that it aligned best to.

The alignment position of a kmer to the current consensus dictates the position of the kmer in the multiple sequence alignment which generates the maturing consensus sequence. The alteration of the consensus sequence by an added kmer is proportional to its NoPeak score. Kmers that do not

align to the consensus sequence with at least *length(consensus)/scorecutoff* (see section: Usage and Parameters) are not added to the alignment and saved for a later iteration, again, starting with the highest scoring of these kmers. This is done to account for proteins that bind to binding sites, differing in their base sequence, and thus have more than one binding motif.

3 Usage and Parameters

Multilenkmer is called from the command line:

```
python3 multilenkmer.py -i <input dataset> -o <output directory>
```

The input dataset is the output of NoPeaks *–export kmers* option (see NoPeak usage for further reference). Generated plots and a latex file listing the results of all combinations of supplied parameters, are saved in the directory specified in the *-o* argument. Optional parameters are:

- [–MI <maximum num. of iterations>]
- [–SC <scorecutoff> ...]
- [–m <match bonus> ...]
- [–mm <mismatch penalty> ...]
- [–go <gap opening cost> ...]
- [–ge <gap extension cost> ...]

Multilenkmer starts a new multiple alignment run, with kmers that did not meet the minimum alignment score criterion (which is modulated by the scorecutoff parameter as described in section Objective and Implementation). The maximum number of such iterations Multilenkmer will perform, is specifiable with the *-MI* option. The parameters match bonus and penalty and gap opening and extension cost refer to the alignment parameters of Biopythons bio.pairwise2 module (see the Biopython docs for further reference).

4 Output, Visualization and Results

Multilenkmer produces a special type of sequence logo plot, that includes a color coded representation of the alignment matrix containing the information incorporated in the sequence logo. Every row in the matrix resembles one kmer, the colors green, blue, yellow, red of the cells represent the 4 bases, analog to the color coding commonly used for sequence logos.

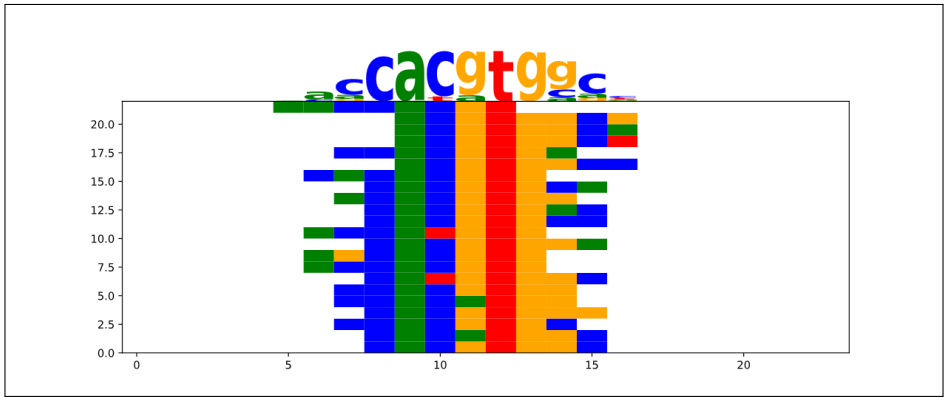

Figure 2: Sequence logo and alignment matrix. Transcription factor MAX.

4.1 Results

Multilenkmers results for tested data sets are largely consistent with the transcription factor binding profiles presented on databases like Jaspar (Fornes et al., 2019).

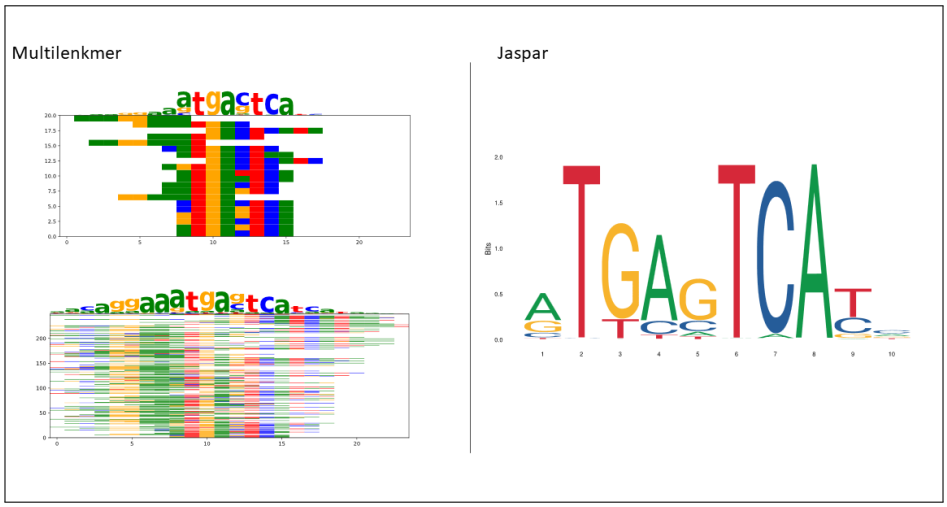

Figure 3: Comparison TF JUN sequence logos. Multilenkmer (left) with scorecutoffs 2.5 (upper image) and 4 (lower image) and Jaspar (right) (Matrix ID MA0099.3)

Different parameter combinations lead to variances in logo clearness but do not did not produce conflicting results. The Jaspar database supplies multiple sequence logos for some transcription factors, which may represent various binding motifs or different parts of the same binding motif. In the case of the transcription factor GABPA it seems, that the different sequences of the logos from Jaspar appear at different positions in the logo from Multilenkmer.

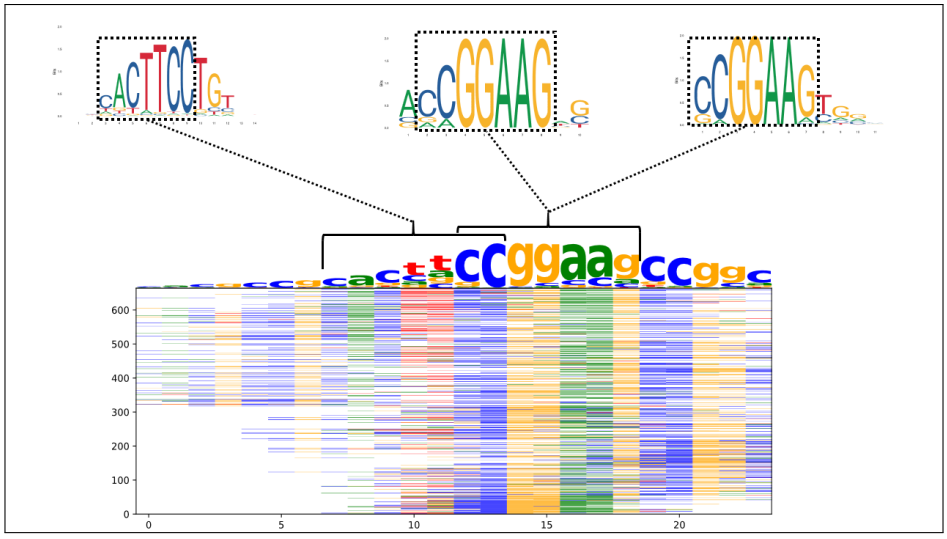

Figure 4: Comparison of sequence logos (TF GABPA) from Jaspar (Jaspar matrix IDs left to right: MA0062.3 , MA0062.1 , MA0062.2) and Multilenkmer (parameters: matchbonus=2, mismatchpenalty=2, gapopening cost=3, gapextension cost=1, scorecutoff=2.4 .

References

Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 03 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp163. URL `https://doi.org/10.1093/bioinformatics/btp163`.

Oriol Fornes, Jaime A Castro-Mondragon, Aziz Khan, Robin van der Lee, Xi Zhang, Phillip A Richmond, Bhavi P Modi, Solenne Correard, Marius Gheorghe, Damir Baranašić, Walter Santana-Garcia, Ge Tan, Jeanne Chèneby, Benoit Ballester, François Parcy, Albin Sandelin, Boris Lenhard, Wyeth W Wasserman, and Anthony Mathelier. JASPAR 2020: update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 48(D1):D87–D92, 11 2019. ISSN 0305-1048. doi: 10.1093/nar/gkz1001. URL `https://doi.org/10.1093/nar/gkz1001`.

David S. Johnson, Ali Mortazavi, Richard M. Myers, and Barbara Wold. Genome-wide mapping of in vivo protein-dna interactions. *Science*, 316(5830):1497–1502, 2007. ISSN 0036-8075. doi: 10.1126/science.1141319. URL `https://science.sciencemag.org/content/316/5830/1497`.

Michael Menzel, Sabine Hurka, Stefan Glasenhardt, and Andreas Gogol-Döring. NoPeak: k-mer-based motif discovery in ChIP-Seq data without peak calling. *Bioinformatics*, 09 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa845. URL `https://doi.org/10.1093/bioinformatics/btaa845`. btaa845.

Ryuichiro Nakato and Katsuhiko Shirahige. Recent advances in ChIP-seq analysis: from quality management to whole-genome annotation. *Briefings in Bioinformatics*, 18(2):279–290, 03 2016. ISSN 1467-5463. doi: 10.1093/bib/bbw023. URL `https://doi.org/10.1093/bib/bbw023`.

Elizabeth G. Wilbanks and Marc T. Facciotti. Evaluation of algorithm performance in chip-seq peak detection. *PLOS ONE*, 5(7):1–12, 07 2010. doi: 10.1371/journal.pone.0011471. URL `https://doi.org/10.1371/journal.pone.0011471`.