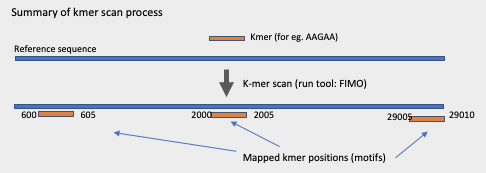
**Motif Search Document for covid Beacon**

**Aim:** To search for short sequences (called kmers or motifs) in the virus reference genome sequence using the tool ‘FIMO’.

Note: in this document kmers and motifs will mean the same thing.

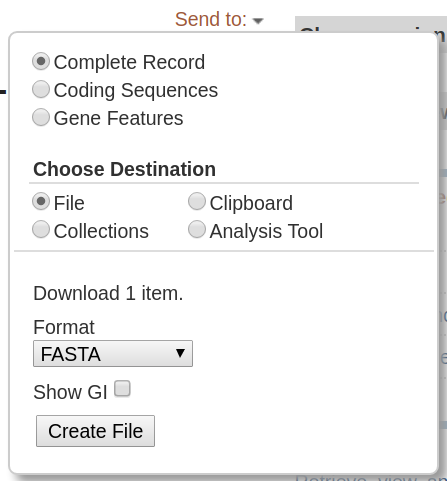


**Prerequisite:**

A. In order to run FIMO we will need to install the Meme suite software. It can be downloaded from [here](http://meme-suite.org/meme-software/5.1.1/meme-5.1.1.tar.gz). Instructions to get the suite installed (and prerequisites needed) can be found [here](http://meme-suite.org/doc/install.html?man_type=web).

B. Reference genome sequence in the server. It should be downloaded from here:

1. [NC\_045512](https://www.ncbi.nlm.nih.gov/nuccore/NC_045512.2?report=fasta)



By default, the web page will use [NC\_045512](https://www.ncbi.nlm.nih.gov/nuccore/NC_045512.2?report=fasta) as the genome reference but other genomes may be uploaded by the user in fasta format.

**User Input:**

1. **Mandatory**: Short Kmer sequence(s) (string, for eg. AAGAA from now on “motif”). Multiple kmer sequences can be provided separated by comma Eg: AAGAA, AGAGAG

2. **Optional:** P-value threshold (numeric, eg. 0.01) #decimal separator has to be “.” (default: 0.01)

3. **Optional**:Reference file. A fasta file for the reference genome may be uploaded by user (default:[NC\_045512](https://www.ncbi.nlm.nih.gov/nuccore/NC_045512.2?report=fasta)). If new interesting genomes appear, the web page could have more genomes uploaded (selected from dropdown list).

**Steps to perform Motif scan and Backend preprocessing:**

Once the input parameters are uploaded, before running FIMO, the backend should transform the motif(s) into a meme format file with following steps:

Step 1: Create a folder (referred here as **sites**) and write each motif in a new file (one motif per line per file ex: motif.txt, motif2.txt). In case user only entered only one motif, create just one file in a folder.

Example, user kmer input: AAGAA, AGAGAG

Backend :

```

cat ‘AAGAA’ >sites/motif.txt

cat ‘AGAGAG’ >sites/motif2.txt

```

Step 2: Convert these motif files into MEME format using following command:

```{bash}

**>sites2meme sites >./path/motif.meme**

```{bash}

#where sites is the name of the input folder from step 1 in which the motif is located.

#In this case

```

less ~/sites/motif.txt

AAGAA

```

An example output from Step 2.

(The motif.meme output file should look like the following for AAGAA motif):

```

cat ./path/motif.meme

```

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MEME version 4

ALPHABET= ACGT

strands: + -

Background letter frequencies (from uniform background):

A 0.25000 C 0.25000 G 0.25000 T 0.25000

MOTIF motif

letter-probability matrix: alength= 4 w= 5 nsites= 1 E= 0

1.000000 0.000000 0.000000 0.000000

1.000000 0.000000 0.000000 0.000000

0.000000 0.000000 1.000000 0.000000

1.000000 0.000000 0.000000 0.000000

1.000000 0.000000 0.000000 0.000000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In case more than one motif was written by the user, the .meme generated file will have all the motifs.

Step 3: This motif.meme file from step 2 will we used as input for fimo:

```{bash}

**fimo --o results --nocr --thresh 0.01 motif.meme NC\_045512.fasta**

#--o name of the folder in which the results will be written.

#--nocr Do not score the reverse complement strand.

#--thresh The output threshold for displaying search results. Only search results with a p-value less than the threshold will be output. May be given by the user.

# motif.meme is the file generated in the previous step.

#NC\_045512.fasta is the reference file. Maybe be given by the user

#when fimo is running the screen will show the following output:

Using motif +motif of width 5.

Computing q-values.

Estimating pi\_0 from a uniformly sampled set of 10000 p-values.

Estimating pi\_0.

Estimated pi\_0=0.99023

```

**Fimo tool output:**

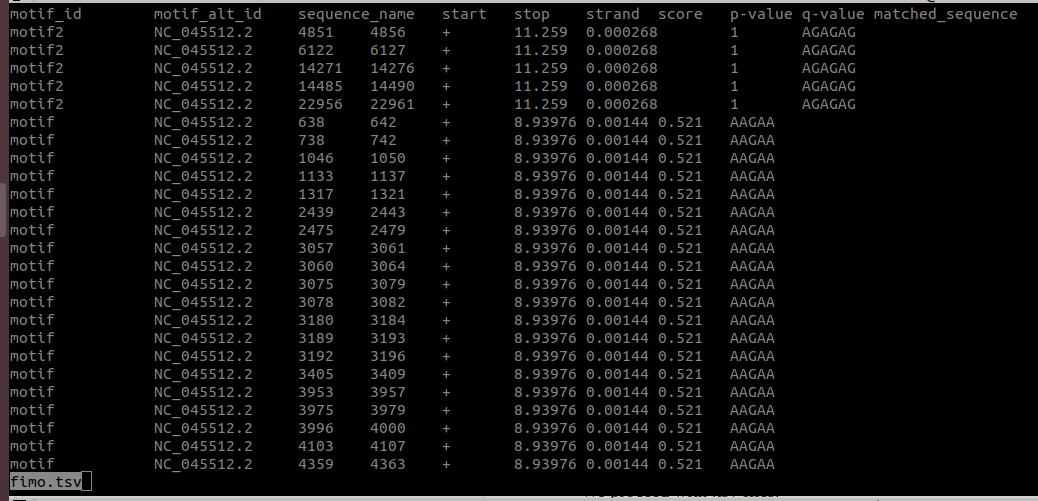
A folder called “results” will be created with 5 files for each step 3 run:

cisml.xml fimo.gff fimo.html fimo.tsv fimo.xml

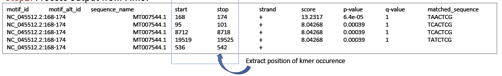
We proceed with **.tsv** files.

Example output of two fimo runs motifs:

1. Two motifs were searched



1. One motif was searched.



**Summary of Motif search output:**

User input (kmer(s) + pvalue + reference sequence) is sent to Fimo which generates 5 output files (.tsv, .html) of matched motif occurrences in genome reference. Each row in the output file is the information of mapped motif instance ie. where in the reference sequence this motif was found. From .tsv output files 4th and 5th column is extracted to know about positions of the motif instances, these positions will be then searched again in Beacon database as ‘region query’ to extract all variants present within these intervals. This information along with metadata will be displayed to the user as Beacon motif search output.

**Steps to display Motif output**

**Information to display and mapped to Beacon Schema:** Proposed in two parts

**Display Part 1: Motif instances (graphic representation)**

****

Fields info:

- Total no. of counts for motif: Comes from fimo output file. No. of rows in \*.tsv file

- whole genome

- info for statistics and response

**Display Part 2: Variant data + metadata in table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Motif Instance** | **Genomic Position** | **SNP Position** | **Kmer on reference** | **Variants on kmer** | **Kmer Frequency** | **Alt Frequency** | **Variant Type** | **Variant effect** |
| 1 | 1251-1255 | 1251A>T | AAGAA | TAGAA | 8/20 | 0.001 | SNP | Silent |
|  |  | 1254A>T | AAGAA | AAGTA |  |  |  |  |
| 2 | 2000-2005 |  | AAGAA | TAGGA | 2/20 |  |  |  |
|  |  | 2000A>T | AAGAA | TAGGA |  | 0.001 | SNP | Silent |
|  |  | 2004A>G | AAGAA | TAGGA |  | 0.002 | SNP | Silent |
| 3 | 3200-3204 |  | AAGGA | AAGTA | 5/10 | 0.01 | SNP | Non-coding |

Output table info: Different scenarios of motif instances found. For example:

motif instance 1: Two variants were found at two different kmers (5th column, first 2 rows)

motif instance 2: Two variants were found in same kmer (5th column, 3-5 rows)

motif instance 3: One variant was found on reference kmer and one on variant kmer (Column 4th and 5th, row 6th)

Output table headers & fields info:

(for each row of tsv file ie. each instance of the kmer)

a) Motif Instances: No. of the 1st, 2nd. Nth row of .tsv file

b) Genomic position: Extracted from column 4th and 5th (‘start’ and ‘stop’ of tsv file)

c) Kmer on reference: Extracted from column 10th ‘matched\_sequence’ of tsv file

d) Variation on kmer: Sequence construction from Beacon variant output for each region.

(This Information is taken from 3rd column ‘SNP position’. The reference kmer alphabet is replaced by the variant alphabet by changing the colour and underlining it. For example in the first row, reference kmer ‘**A**AGAA’ is replaced by variant kmer ‘TAGAA’ because ‘SNP position’ is 1251**A**>T.)

e) Kmer frequency : Total no. of kmer found with variant at this position on all sequences / total no. of kmers at this position on all sequences

f) Alternative allele frequency: From Beacon schema field ‘**Alt**’ in Variant Basic end-point, split into multiple rows if more than one variant found at the same kmer.

g) Variant type: From Beacon schema field ‘**variantType**’ in Variant Basic end-point, split into multiple rows if more than one variant is found at the same kmer.

h) Variant effect: From Beacon schema field ‘**molecularEffect**’ in Variant Annotation end-point. split into multiple rows if more than one variant is found at the same kmer.

[Click here](https://github.com/clauw87/virusbeacon/blob/raw_ideas/beacon_v2_endpoints.pdf) for beacon schema endpoint document.

**Additional metadata to pull:**

- Host

- Age

- Country

- Biosample

- Source