User guide to EMGU

Joey Azofeifa

University of Colorado Boulder

Contents

1 Introduction

EMGU is an open source software housed at `https://github.com/azofeifa/` written in the C/C++ programming language with OpenMP support and requires GNU version 4.9 or greater. There are four modules that EMGU: (1) a bidirectional caller, (2) formatting data given intervals, (3) running parameter estimates for many models and (4) performing model selection to pick the best model. Each module has an associated config file that the user can mutate accordingly. Thus to run EMGU the user need only navigate to the src/ directory and run

```
bash-terminal$ ./EMGU config_file.txt
```

where there are four separate config files for the four separate modules.

The purpose of this model and software package is to infer the exact position and variability of RNA Polymerase II loading, length of initiation and extent of elongation. We formulate this as a mixture model and attempt to infer the below parameters from each component. The paused form of polymerase is as actually the convolution of initiation and elongation and thus an **E**xponentially **M**odified **G**aussian; the elongation component is a **U**niform distribution. We infer these parameters via the expectation maximiation algorithm given GRO-seq read coverage data. Briefly we define the EMG component as:

$$f(z,s;\mu,\sigma,\lambda) = \lambda\phi(\frac{z-\mu}{\sigma})R(\lambda\sigma - s\cdot\frac{z-\mu}{\sigma})(\pi_p)^{\max(0,s_i)}(1-\pi_p)^{\max(0,-s_i)} \tag{1}$$

where z is the GRO-seq genomic coordinate and $s \in \{-1,1\}$, 1 if on the forward strand and -1 if on the reverse strand, $\phi(x)$ is the standard normal distribution and $R(x)$ is the *Mill's ratio* defined as $\frac{1-\Phi(x)}{\phi(x)}$ where $\Phi(x)$ is the CDF for the standard normal. We also note that Pol II may elongate 5' - 3', showing uniform read density till the transcription termination site,

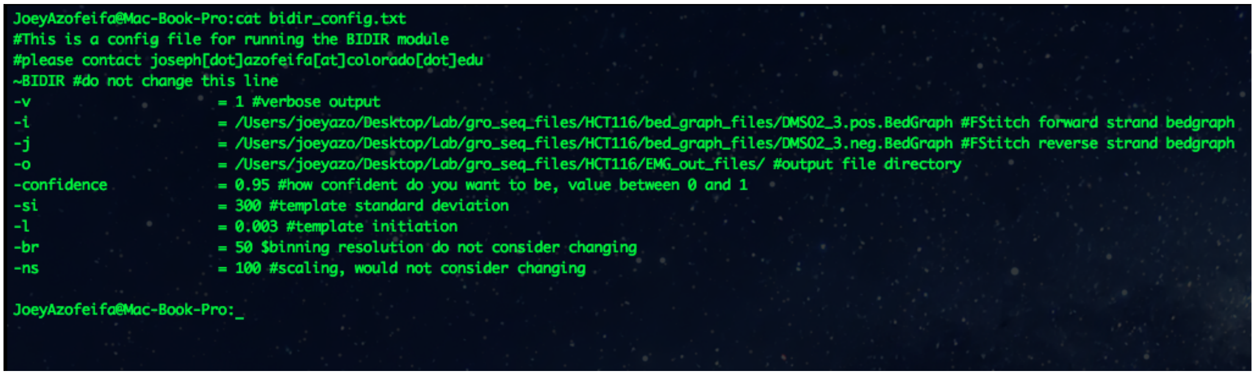
$$g(z,s;\mu,l) = \frac{1}{l-\mu}(\pi_e)^{\max(0,s_i)}(1-\pi_e)^{\max(0,-s_i)} \tag{2}$$

the final mixture,

$$p(z,s) = w\cdot f(z,s) + (1-w)\cdot g(z,s) \tag{3}$$

2 Config Files and Running EMGU

There are two ways to supply the necessary parameters to any module. One way is to insert them into a config file. This file has every parameter listed for the given module and a short description of each parameter following a pound sign. Below is an example.



```
JoeyAzofeifa@Mac-Book-Pro:cat bidir_config.txt
#This is a config file for running the BIDIR module
#please contact joseph[dot]azofeifa[at]colorado[dot]edu
~BIDIR #do not change this line
-v                      = 1 #verbose output
-i                      = /Users/joeyazo/Desktop/Lab/gro_seq_files/HCT116/bed_graph_files/DMSO2_3.pos.BedGraph #FStitch forward strand bedgraph
-j                      = /Users/joeyazo/Desktop/Lab/gro_seq_files/HCT116/bed_graph_files/DMSO2_3.neg.BedGraph #FStitch reverse strand bedgraph
-o                      = /Users/joeyazo/Desktop/Lab/gro_seq_files/HCT116/EMG_out_files/ #output file directory
-confidence             = 0.95 #how confident do you want to be, value between 0 and 1
-si                     = 300 #template standard deviation
-l                      = 0.003 #template initiation
-br                     = 50 $binning resolution do not consider changing
-ns                     = 100 #scaling, would not consider changing

JoeyAzofeifa@Mac-Book-Pro:_
```

The first line is a just a header. The second line with the tilda sign, signifies the module that this config is specific for. This line should not be changed. The rest of the lines list the parameter and their associate value. These parameters can be changed by simply replaced the number or the file path in between the equals sign and the pound sign.

Indeed, you can also overwrite any parameters in the config file by supplying them in the command line. With the above examble, $-l$ is set to 0.003 to overwrite this at run time you can do,

bash-terminal$ ./EMGU bidir_config.txt $-l$ 0.1

and 0.1 will be the parameter to the program not 0.003.

3 Bidirectional Detector

3.a Method Description

The simplest way to identify regions of the genome that have bidirectional transcription is to compare model fits between an elongation component and a bidirectional component and simply perform a template matching algorithm to identify good hits. Thus we slide the center of loading or μ in $f(z,s)$ and compare the likelihoods via Bayesian Information Criteria (BIC) to a bidirectional fit verse noise or elongation fit. Thusly we compute this score for every position in the genome:

$$\begin{gathered} ll_f = \sum_{i=j}^{k} \log f(z_i, s_i); ll_g = \sum_{i=j}^{k} \log g(z_i, s_i) \\ BIC_f = -2 \cdot ll_f + 5 \cdot \log N; BIC_g = -2 \cdot ll_g + 2 \cdot \log N \\ SCORE = BIC_g / BIC_f \end{gathered} \tag{4}$$

If this score is above one, than the likelihood of this window being in the bidirectional component is greater than that of the uniform component. To compute this score, we must fix the variance and initiating parameter of the EMG component. We let users play with this number but I have found that variance around (σ) 300 and initiating parameter (λ) at 0.003 to be the best as this is the global average parameter estimates across single isoform genes in the HCT116 DMSO GRO-seq (however this number may change from dataset to datasets...users are encouraged to figure out this number for themselves by running below modules on single isoform genes). The window is simply one standard deviation to the right and to the left of some genome position and thus is a function of the user provided σ and λ options. Lastly the user can provide a confidence threshold, anything above 0.95 would be considered highly significant, decreasing this parameter will lead to more bidirectional calls.

3.b Parameter Description

Flag	Type	Description
-i	string;/path/to/file/	forward strand bed graph file
-j	string;/path/to/file/	reverse strand bed graph file
-o	string;/path/to/directory/	output directory (bidirectional_hits.bed created)
-confidence	number(float/double)	score confidence
-si	number(float/double)	EMG template σ
-l	number(float/double)	EMG template λ
-br	number(float/double)	binning resolution***
-ns	number(float/double)	normalizing scale***
		***Would not consider changing

4 Formatting Data

4.a Method Description

4.b Parameter Description

Flag	Type	Description
-i	string;/path/to/file/	bed file of intervals
-j	string;/path/to/file/	forward strand bed graph file
-k	string;/path/to/file/	reverse strand bed graph file
-o	string;/path/to/directory/	output directory (EMGU_formatted_file.tsv created)
-pad	number, integer	additional data inserted to interval

5 Estimating Parameters

5.a Method Description

5.b Parameter Description

Flag	Type	Description
-i	string;/path/to/file/	EMGU_formatted_file.tsv, (output from FORMAT module)
-o	string;/path/to/file/	output directory (model_fits.tsv) will be created
-np	number(integer)	number of CPU cores
-chr	string; chr# or all	run on a specific chromosome or on all
-br	number(float/double)	binning resolution***
-ns	number(float/double)	normalizing scale***
-minK	number(integer)	do not consider models with less than this many component
-maxK	number(integer)	do not consider models with more than this many component
-rounds	number(integer)	number of random EM initializations
-ct	number(float/double)	EM convergence threshold difference
-mi	number(integer)	Number of EM iterations before it is aborted
-mi	number(integer)	Number of EM iterations before it is aborted
-max_noise	number(float/double)	weight of the random/uniform noise component***
-ALPHA_0	number(float/double)	hyper parameter for σ,NIG(ALPHA_0, BETA_0)***
-BETA_0	number(float/double)	hyper parameter for σ,NIG(ALPHA_0, BETA_0)***
-ALPHA_1	number(float/double)	hyper parameter for λ,Γ(ALPHA_1, BETA_1)***
-BETA_1	number(float/double)	hyper parameter for λ,Γ(ALPHA_1, BETA_1)***
-ALPHA_2	number(float/double)	beta hyper parameter for strand bias***
-ALPHA_3	number(float/double)	dirchlet hyper parameter for component weights***
		***Would not consider changing

6 Model Selection

6.a Method Description

6.b Parameter Description

Flag	Type	Description
-i	string;/path/to/directoy/	directory of model_fit.tsv files, (output of MODEL module)
-o	string;/path/to/file/	output directory (model_fits.txt and/or model_fits.bed created)
-penailty	number;(float/double)	BIC penalty***
-to_igv	0/1	output a file to view in IGV/UCSC browser viewing
-to_EMG	0/1	output a file in format as from MODEL module

7 File Type Appendix