DREEM: Detection of RNA folding Ensembles using Expectation-Maximization clustering

1. Overview

This manual describes how to run DREEM, a computational tool to detect alternative structures formed by RNA molecules. DREEM takes as input single molecule DNA sequencing results from an RNA chemical probing experiment (such as DMS-MaPseq) and clusters individual reads based on their mutation profiles.

The DREEM pipeline consists of two steps:

- Step 1: Conversion of aligned reads to 'bit vectors' consisting of '0's (matches) and '1's (mismatches and deletions)
- Step 2: Clustering of bit vectors based on their mutation profiles using an Expectation-Maximization (EM) algorithm

Additionally, if the sequencing data from a sample has not been mapped yet, DREEM can optionally align the reads to a reference genome as an extra step before moving on to Step 1 and creating bit vectors.

2. Running DREEM on Code Ocean

DREEM has been made available to run on Code Ocean. The DREEM capsule comes with all the dependencies pre-installed. To execute the DREEM pipeline:

- Click on *Capsule -> Duplicate*.
- After the capsule has been duplicated, click on '*Reproducible Run*'

After DREEM finishes running (∼1.5 hours), all the output files show up on the right hand side of the browser window under **'Run'**.

A demo dataset has been provided in '*/data/DREEM_Input/*' that consists of three paired-end samples (one *in vitro*, one *in vivo* and one untreated) with DMS-MaPseq data corresponding to the RRE region (∼150 base-pairs) in the HIV genome.

3. Output of DREEM

The output files from DREEM are gathered in five folders:

- *Mapping_Files* and *Mapping_Plots*: created at the mapping step
- *BitVector_Files* and *BitVector_Plots*: created at the bit vector step
- *EM_Clustering*: created at the clustering step

The contents of the five folders are described below.

3.1 Mapping_Plots

This folder comprises of output files from FASTQC (containing information regarding the quality of the reads) and from Picard (containing information regarding the quality of the alignment process).

3.2 Mapping_Files

This folder comprises of a SAM file and a BAM file that contain reads that aligned to the reference genome and the position of their alignment.

3.3 BitVector_Plots

This folder comprises of four graphs that inform about various parameters associated with the 'bit vector' step:

- *read_coverage.html*: Coverage across the region of interest. We analyzed regions in the HIV whole genome that had at least 100,000 reads covering them, satisfied by approximately 70% of the regions.
- *pop_avg.html*: Mutational fraction of the bases. Fraction of reads in which a base is mutated, for e.g. if a base is mutated in 10% of reads, its mutational fraction is 0.1. In the HIV whole genome data (Fig 4), the signal on A's and C's was around 0.02 and the noise on T's and G's was around 0.002, giving a signal-noise ratio of approximately 10.
- *mutation_histogram.html*: Histogram of the number of mutations per read. Approximately 70% of the regions in the HIV whole genome sample had at least 10% of their reads containing at least 2 mutations.
- *DMS_mutations.html*: DMS-induced base conversions

3.4 BitVector_Files

This folder comprises of a text file containing a list of all the bit vectors corresponding to the reads in the SAM file.

3.5 EM_Clustering

This folder comprises of the clustering results. The contents of the various output files and their hierarchical location are described below.

- **SampleName_RefName_Start_End**:
  Folder containing the clustering results for the sample.
  - *BitVectors_Filter.txt*:
    Number of bit vectors that passed the filtering steps and were used for clustering.

– **K_1, K_2 and (if the BIC test passes) K_3**:
Folders containing clustering results for the various numbers of clusters (*'K'*)

  * *log_likelihoods.txt*:
  Log likelihood from all the runs and their corresponding BIC score.

  * **run_1, run_2, etc.**:
  Folders corresponding to the independent EM runs. One run is carried out for *K=1* and ten runs each for *K=2* and *K=3*. After all the runs have finished, in a post-processing step the folder containing the run with the largest log likelihood is renamed with *'−best'* at the end. For example if '*run_7'* had the largest log likelihood for '*K_2'*, the folder is renamed to '$run\_7 - best$'.

    · *Clusters_Mu.txt*: Mutational fraction of the bases in the K clusters

    · *Responsibilities.txt*: Probability of the reads belonging to the K clusters

    · *Proportions.txt*: Proportions of the K clusters

    · *Log_Likelihoods.html* and *Log_Likelihoods.txt*: Log likelihood after each iteration of the EM algorithm

    · *DMSModRate_Clusters.html* and *DMSModRate.html*: Plots of the mutational fraction of the bases in the K clusters

    · *Cluster*1_*Cluster*2_*mus.html*, *Cluster*1_*Cluster*2_*normmus.html*, etc.: Plot of normalized and non-normalized mutational fraction of bases in Cluster 1 vs Cluster 2, etc.

    · '*.ps'* files: Secondary structure models by RNAstructure of the sequence expanded by 0, 50, 100, 150 and 200 bases upstream and downstream of the region of interest. This is done only in the *'best run'* folder, i.e. the run with the largest log likelihood.

4. Running DREEM on a local machine

As an alternative option to running the pipeline on Code Ocean, users can clone the Code Ocean capsule (by clicking on *Capsule -> Clone via Git*) and run DREEM on their computer. The Git repository does not come with the demo dataset; in order to download the demo dataset as well, click on *Capsule -> Export* and check '*Include data*'.

If running DREEM on a local machine, there are a number of dependencies that need to be manually installed. To run steps 1 and 2 of the DREEM pipeline, install Python version 3.6 or higher and the Java Runtime Environment version 8 or higher. The Python modules needed to run DREEM are 'plotly', 'scipy', 'pandas' and 'biopython'. These modules can be installed by invoking a package manager (such as 'conda' or 'pip'). For example:

$conda install plotly

Additionally, if you want to run mapping and/or secondary structure modeling, the following software packages are used by DREEM:
- If you need to align reads to a reference genome before running Step 1, make sure that you have installed FASTQC, TrimGalore and Bowtie2.
- If you are interested in secondary structure models of the clusters (generated using the reactivities of the bases as constraints), DREEM can run RNAstructure if it has been installed, although theoretically any thermodynamics-based secondary structure prediction tool can be used.

After installing these packages, add them to your computer's *'PATH'* environment variable. This can be done, for example, by running:

$sudo nano /etc/paths

and adding the directories containing these software packages.

Before running DREEM, the input files (such as the ones provided in the demo dataset) need to be placed in a single location. Users can process any since cell chemical probing data using the DREEM pipeline. Navigate to a directory of your choice in your local machine and place the following files in the directory:

- A BAM file containing aligned reads from the sample of interest. Ensure that the name of the BAM file follows the following format: *SampleName_RefName.bam*, where *'SampleName'* is the name of the sample and *'RefName'* is the name of the reference genome. If there is no BAM file available, i.e. mapping has not been carried out, place the FASTQ file(s) containing the sequencing reads inside the input folder. If the sample has been sequenced using a paired-end protocol, ensure that the names of mate 1 and mate 2 reads end with *'*$\_mate1.fastq$*'* and *'*$\_mate2.fastq$*'*, respectively.
- A FASTA file containing the sequence of the reference genome. Bowtie2, the aligner used by DREEM for mapping, requires building indexes **prior to** aligning reads to the reference genome. The Bowtie2 indexes can be created by navigating to the input directory and running:

  $bowtie2-build RefName.fasta RefName

Now that all the input files are ready, the DREEM pipeline can be run by using the following command, after navigating to the folder that contains the Python scripts:

$python Run_DREEM.py InputDir OutputDir SampleName RefName Start End [--single] [--fastq] [--struct]

- *InputDir*: **Full path** to the directory where the input files exist
- *OutputDir*: **Full path** to the directory where the output files will be saved
- *SampleName*: Name of the sample (**up to** *'*$\_RefName.bam$*'* if starting from a BAM file or **up to** *'*$\_mate1/2.fastq$*'*, if mapping paired-end reads)
- *RefName*: Name of the reference genome
- *Start*: Start position of the region of interest in the genome (1-based)
- *End*: End position of the region of interest in the genome (1-based)
- *--single*: Sample comes from single-end sequencing [DEFAULT: OFF]
- *--fastq*: A BAM file is not available and mapping needs to be done [DEFAULT: OFF]
- *--struct*: Run secondary structure prediction using RNAstructure with DMS reactivity as constrains [DEFAULT: OFF]

All the output that is generated by DREEM is saved to the OutputDir folder.

Note: DREEM has been tested on an Ubuntu 18.04 system and typically takes ∼1.5 hours on 12 cores of computing power to process a sample with ∼500,000 reads covering a ∼200 base-pair region.

Code written by Harish Swaminathan. Earlier versions of this code were written by Vincent Corbin with contributions from Margalit Glasgow and Matthew Edwards. Lachlan Mcintosh contributed to the likelihood model.