Manual and installing settings for:

Novel taxonomy-independent deep learning microbiome approach allows for accurate classification of different forensically relevant human epithelial materials

1. Summary

The procedure of identification three forensically-relevant human epithelial materials (skin, oral and vaginal secretion) from 16S ribosomal RNA (rRNA) gene massively parallel sequencing (MPS) data can be summarized in:

1) Generate a matrix with the number of reads of each nucleotide adenine (A), thymine (T), cytosine (C), guanine (G), insertion (+) or deletion (-) observed at selected single position of the 16S rRNA gene region aligned to *Escherichia coli* str. K 12 substr. MG1655 (NC_000913.3) 16S rRNA gene sequence.
2) Predict the tissue category (skin, oral, vagina) using the estimated proportions in step 1. For this, 50 previously trained taxonomy-independent deep learning (DL) networks are used. The training dataset consists of 1,636 skin, oral and vaginal samples from the Human Microbiome Project (HMP) 16S production phase I (HMP-16S-PP1) [1].
3) Average the output from the 50 DL networks.

2. Requirements and installation settings

Operating system: UNIX based. Tested on Ubuntu 16.04 LTS (should also work on different distros).

Important:

1) Please be sure to have installed the following python version and libraries:

python	3.6.8
tensorflow	1.12.0
scikit-learn	0.20.0
pandas	0.23.4
numpy	1.15.4

2) Following dependencies need to be installed on "/usr/local/bin/" (you can skip this step if already installed on your system):

SAMtools [2] (version: samtools-1.4.1) with default parameters.

1. wget https://github.com/samtools/samtools/releases/download/1.4.1/samtools-1.4.1.tar.bz2 -O \ samtools.tar.bz2
2. tar -xjvf samtools.tar.bz2
    a. cd samtools-1.4.1/
    b. ./configure
    c. make
    d. make install

BWA-MEM [3] (version: bwa-0.7.12) with parameters: -B1, -O1, -E1 and -L1.

Installation settings:

1. git clone https://github.com/lh3/bwa.git
2. cd bwa; make
3. sudo cp bwa /usr/local/bin/

Note: You can find the reference genome already aligned. However, if preferred, you can align it yourself. You must create an index from your reference genome beforehand using the following command:

bwa index ref.fa (e.g. *Ecoli_K12_ref.fa*).

This will create additional files with different extensions, but you only need to refer to the original name of the reference file.

3. Usage and examples

tissueid accepts both fasta and fastq files as input. Examples of how to run it are as follows:

python tissueid.py -fastq <path folder> -out <path folder> -model <path folder> -pos pos_file.bed -ref ref/Ecoli_K12_ref.fasta -t 16

python tissueid.py -fasta <path folder> -out <path folder> -model <path folder> -pos pos_file.bed -ref ref/Ecoli_K12_ref.fasta -t 16

4. Command line options

Provide an input file (fasta or fastq) or folder path

-h, --help show help message and exit

-fasta DIR, --FASTA DIR

Single sample file or path folder with fasta files

-fastq DIR, --FASTQ DIR

Single sample file or path folder with fastq files

-out DIR, --OUTPUT DIR

Output folder path

-model DIR, --MODEL DIR

Model folder path

-pos BED_file, --POS_FILE BED_file

Position in bed format

-ref REF_GENOME, --REF REF_GENOME

path/Ecoli_K12_ref.fasta

-t THREADS, --Threads THREADS

Set a number of additional threads to use [CPUs]

5. Output file formats

Within the path folder given by the -out command you will encounter three folders named alignments, pileup and frequency, respectively.

1) Alignments: as the name suggests, it contains all the files produced during alignment with BWA-MEM, such as sam, bam and index files.

2) Pileup: this folder stores all the base calling files performed after alignment. It uses samtools mpileup command and the pos_file.bed file containing the 240 informative positions. These pileup files are used in order to calculate the frequency for each base position match in the reference genome.

3) Frequencies: each file contains, for each position of the 16r RNA gene sequence, the number of reads that have been matched to (A), (T), (C), (G), (+) and (-) with regards to the *Escherichia coli* str. K 12 substr. MG1655 *E coli* (NC_00913.3) 16S rRNA gene sequence. For example:

Position	A	T	G	C	+	-
239	1	0	0	0	0	0
240	1708	0	1	0	0	0
241	0	0	1771	0	24	0
242	1894	1	0	0	0	0
243	0	0	1923	0	51	16
244	0	1921	0	0	0	16
245	0	1940	0	0	0	0
246	0	1940	0	0	3	0
247	0	0	2232	0	1	0
248	2402	0	1	0	26	0
249	0	2525	0	1	66	0
250	0	2	0	2610	0	2

This matrix is the one that defines the features that are going to be used as input in the DL prediction.

6. Deep learning training

A supervised neural network with a training dataset was generated with a four-layer topology. The first layer known as the input layer contains 1,440 features, the two following hidden layers containing 10 neurons with hyperbolic tangent (TANH) as the activation function [4] and Adam [5] as the optimizer with a learning rate of 0.01.

Since we consider a very limited number of samples with a large number of features as the training dataset, overfitting must be considered as the main issue. Therefore, we proposed to use algorithms to prevent overfitting such as Dropout (0.1) [6] and to average the output over a set of 50 neural networks.

In order to train each neural network, we collected a set of samples obtained from "estimation the number of reads of each sample" as training dataset. We considered 1,636 samples with oral, skin and vaginal origin of the Human Microbiome Project as the training dataset.

7. Deep learning prediction

DL based tissue prediction of a new sample requires the existence of 50 neural networks previously trained with a training dataset. The features of the new sample will be generated by using 240*6 positions from the generated matrix as input file and scaling each cell by the values observed in the same cell of the training dataset. Next, the generated features will be used in each of the 50 neural networks. The final prediction is computed averaging the predictions of each neural network over all networks.

References

1. Human Microbiome Project Data Analysis and Coordination Center. https://www.hmpdacc.org/hmp/(2017). Accessed 24 May 2018.
2. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25:2078-9. doi: 10.1093/bioinformatics/btp352.
3. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:13033997. 2013.
4. Karlik B, Olgac AV. Performance analysis of various activation functions in generalized MLP architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems. 2011;1:111-22. http://www.cscjournals.org/library/manuscriptinfo.php?mc=IJAE-26.
5. Kingma DP, Ba JL. Adam: A method for stochastic optimization. In Proc. 3rd Int. Conf. Learn. Representations. 2014. http://hdl.handle.net/11245/1.505367.
6. Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press. 2016.