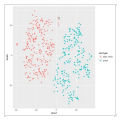


Jun 26, 2019

Instructional tutorial for using *demuxlet*

Hyun Min Kang[1], Meena Subramaniam[2], Sasha Targ[2], Michelle Nguyen[2], Lenka Maliskova[2], Elizabeth McCarthy[2], Eunice Wan[2], Simon Wong[2], Lauren Byrnes[2], Cristina M Lanata[2], Rachel E Gate[2], Sara Mostafavi[3], Alexander Marson[2], Noah Zaitlen[2], Lindsey A Criswell[2], Chun Jimmie Ye[2]

[1]Department of Biostatistics and Center for Statistical Genetics, University of Michigan, [2]University of California, San Francisco, [3]Department of Statistics, University of British Columbia

Working dx.doi.org/10.17504/protocols.io.233ggqn

Human Cell Atlas Method Development Community

Anton Ogorodnikov

ABSTRACT

Genetic multiplexing of barcoded single cell RNA-seq

Introduction

demuxlet is a software tool to deconvolute sample identity and identify multiplets when multiple samples are pooled by barcoded single cell sequencing.***demuxlet*** takes (1) a SAM/BAM/CRAM file produced by the standard 10x sequencing platform, or any other barcoded single cell RNA-seq (with proper --tag-UMI and --tag-group) options (2) a VCF/BCF file containing the genotype (GT), posterior probability (GP), or genotype likelihood (GL) to assign each barcode to a specific sample (or a pair of samples) in the VCF file.

Tutorial for demuxlet

This tutorial provides streamlined instructions for using the tool **demuxlet**. For a more detailed description of all of the options available to use with **demuxlet**, please refer to the README (attached in .zip file). **demuxlet** has several dependencies and this tutorial provides steps to use a docker instance to minimize compatibility issues and an alternative of installing demuxlet from source.

demuxlet is a software tool to deconvolute sample identity and identify multiplets when multiple samples are pooled by barcoded single cell sequencing. **demuxlet** requires the following input files:

1. a SAM/BAM/CRAM file produced by the standard 10x sequencing platform, or any other barcoded single cell RNA-seq (with proper **--tag-UMI** and **--tag-group**) options.
2. a VCF/BCF file containing the genotype (GT), posterior probability (GP), or genotype likelihood (GL) to assign each barcode to a specific sample (or a pair of samples) in the VCF file.

EXTERNAL LINK

https://github.com/statgen/demuxlet/tree/master/tutorial

tutorial.zip

GUIDELINES

Additional resources

The README for demuxlet is available here.
If you have questions about using demuxlet or suggestions for future releases, please contact jimmie.ye@ucsf.edu.

Source of 293T and Jurkat VCF file.

1. 293T VCF Source website: http://hek293genome.org/v2/data.php Source file:http://bioinformatics.psb.ugent.be/downloads/genomeview/hek293/SNP/293T_RTG.vcf.gz
2. Jurkat VCF Source website: https://zenodo.org/record/400615#.WYIh7IQrLIV Source file: https://zenodo.org/record/400615/files/jurkat_final_variant_calls.tar.gz
3. 293T:jurkat VCF file generation We used the CrossMap tool to liftover the 293T vcf file from hg18 to hg19. The tetraploid genotype for the Jurkat vcf was collapsed to a diploid genotype before being merged with the 293T vcf file and the resulting file was filtered to contain only the exon positions.

Installing demuxlet

1. MAC NOTES

You will need to install some additional packages on a Mac.

First, you will need XCode CommandLineTools

Second, it's good to install homebrew:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Then, we need to install autoconf, automake, and libtool

```
$ brew install autoconf automake libtool
```

We will also need a new clang since Apple's clang doesn't support OpenMPI

```
$ brew install llvm $ export CC=/usr/local/Cellar/llvm/4.0.0_1/bin/clang $ export
CXX=/usr/local/Cellar/llvm/4.0.0_1/bin/clang++  $ export LDFLAGS=-L/usr/local/Cellar/llvm/4.0.0_1/lib/
```

The CRAM format may use LZMA2 compression, which is implemented in HTSlib by using compression routines from liblzma http://tukaani.org/xz/.

Building HTSlib requires liblzma development files to be installed on the build machine; you may need to ensure a package such as liblzma-dev (on Debian or Ubuntu Linux), xz-devel (on RPM-based Linux distributions or Cygwin), or xz (via Homebrew on macOS) is installed; or build XZ Utils from source. Assuming we are on a Mac:

```
$ brew install xz
```

2. Install htslib

```
$ git clone https://github.com/samtools/htslib.git
$ cd path/to/htslib
$ autoheader
$ autoconf
$ ./configure # optional, --prefix=/path/file
$ make
$ make install # optional, DESTDIR=/path/file
```

3. Install demuxlet

demuxlet and htslib should be installed in same directory

```
$ git clone https://github.com/statgen/demuxlet.git
```

```
$ cd /path/to/demuxlet
$ mkdir m4
$ autoreconf -vfi
$ ./configure # optional, --prefix=/path/file
$ make
$ make install # optional, DESTDIR=/path/file
```

Tips for running

- Set **--alpha 0 --alpha 0.5**, which assumes the expected proportion of 50% genetic mixture from two individuals, to get better estimates of doublets.
- Set **--group-list** to a list of barcodes (i.e. barcodes.tsv from 10X) to speed things up and only get demultiplexing for cells called by other methods
- To reproduce the results presented in Figure 2 of the demuxlet paper, please go to: https://github.com/yelabucsf/demuxlet_paper_code/tree/master/fig2 to download the vcf and the outputs of demuxlet.

Using demuxlet

demuxlet uses a self-documentation utility. You can run each utility with -man or -help option to see the command line usages.

```
$ ./demuxlet          (for short usage)
$ ./demuxlet -help    (for detailed usage)
```

The detailed usage is also pasted below.

```
Options for input SAM/BAM/CRAM
   --sam           [STR: ]             : Input SAM/BAM/CRAM file. Must be sorted by coordinates and
indexed
   --tag-group     [STR: CB]           : Tag representing readgroup or cell barcodes, in the case to
partition the BAM file into multiple groups. For 10x genomics, use CB
   --tag-UMI       [STR: UB]           : Tag representing UMIs. For 10x genomiucs, use UB

Options for input VCF/BCF
   --vcf           [STR: ]             : Input VCF/BCF file, containing the individual genotypes (GT),
posterior probability (GP), or genotype likelihood (PL)
   --field         [STR: GP]           : FORMAT field to extract the genotype, likelihood, or
posterior from
   --geno-error    [FLT: 0.01]         : Genotype error rate (must be used with --field GT)
   --min-mac       [INT: 1]            : Minimum minor allele frequency   --min-callrate  [FLT: 0.50]
: Minimum call rate
   --sm            [V_STR: ]           : List of sample IDs to compare to (default: use all)
   --sm-list       [STR: ]             : File containing the list of sample IDs to compare

Output Options
  --out           [STR: ]             : Output file prefix
  --alpha         [V_FLT: ]           : Grid of alpha to search for (default is 0.1, 0.2, 0.3, 0.4,
0.5)
  --write-pair    [FLG: OFF]          : Writing the (HUGE) pair file
  --doublet-prior [FLT: 0.50]         : Prior of doublet
  --sam-verbose   [INT: 1000000]      : Verbose message frequency for SAM/BAM/CRAM
  --vcf-verbose   [INT: 10000]        : Verbose message frequency for VCF/BCF

Read filtering Options
   --cap-BQ        [INT: 40]           : Maximum base quality (higher BQ will be capped)
   --min-BQ        [INT: 13]           : Minimum base quality to consider (lower BQ will be skipped)
   --min-MQ        [INT: 20]           : Minimum mapping quality to consider (lower MQ will be
ignored)
   --min-TD        [INT: 0]            : Minimum distance to the tail (lower will be ignored)
```

```
   --excl-flag     [INT: 3844]         : SAM/BAM FLAGs to be excluded

Cell/droplet filtering options
   --group-list    [STR: ]             : List of tag readgroup/cell barcode to consider in this run.
All other barcodes will be ignored. This is useful for parallelized run
   --min-total     [INT: 0]            : Minimum number of total reads for a droplet/cell to be
considered
   --min-uniq      [INT: 0]            : Minimum number of unique reads (determined by UMI/SNP pair)
for a droplet/cell to be considered
   --min-snp       [INT: 0]            : Minimum number of SNPs with coverage for a droplet/cell to be
considered
```

Interpretation of output files

demuxlet generates multiple output file, such as **[prefix].best**, **[prefix].sing**, **[prefix].sing2**, and optionally **[prefix].pair** (with **--write-pair** argument). Each file contains the following information

- The **[prefix].best** file contains the best guess of the sample identity, with detailed statistics to reach to the best guess
- The **[prefix].sing** file contains the statistics for matching each cell with each possible sample.
- The **[prefix].sing2** file contains the statistics similar information to the previous one, but generated for sanity checking of the **[prefix].pair** results.
- The **[prefix].pair** file contains the statistics for matching each cell with each possible configuration of doublet.

The **[prefix].best** file contains the following 22 columns.

1. BARCODE - Cell barcode for the cell that is being assigned in this row
2. RD.TOTL - The total number of reads overlapping with variant sites for each droplet.
3. RD.PASS - The total number of reads that passed the quality threshold, such as mapping quality, base quality.
4. RD.UNIQ - The total number of UMIs that passed the quality threshold. If a UMI is observed in a single variant multiple times, it won't be counted more. If a UMI is observed across multiple variants, it will be counted as different.
5. N.SNP - The total number of variants overlapping with any read in the droplet.
6. BEST - The best assignment for sample ID.
   - For singlets, SNG-
   - For doublets, DBL---
   - For ambiguous droplets, , AMB---<doublet ID1/ID2>)
7. SNG.1ST - The best singlet assignment for sample ID
8. SNG.LLK1 - The log(likelihood that the ID from SNG.1ST is the correct assignment)
9. SNG.2ND - The next best singlet assignment for sample ID
10. SNG.LLK2 - The log(likelihood that the ID from SNG.2ND is the correct assignment)
11. SNG.LLK0 - The log-likelihood from allele frequencies only
12. DBL.1ST - The sample ID that is most likely included if the assignment is a doublet
13. DBL.2ND - The sample ID that is next most likely included ifthe assignment is a doublet
14. ALPHA - % Mixture Proportion
15. LLK12 - The log(likelihood that the ID is a doublet)
16. LLK1 - The log(likelihood that the ID from DBL.1ST is the correct singlet assignment)
17. LLK2 - The log(likelihood that the ID from DBL.2ND is the correct singlet assignment)
18. LLK10 - The log(likelihood that the ID from DBL.1ST is one of the doublet, and the other doublet identity is calculated from allele frequencies only)
19. LLK20 - The log(likelihood that the ID from DBL.2ND is one of the doublet, and the other doublet identity is calculated from allele frequencies only)
20. LLK00 - The log(likelihood that the droplet is doublet, but both identities are calculated from allele frequencies only)
21. PRB.DBL - Posterior probability of the doublet assignment
22. PRB.SNG1 - Posterior probability of the singlet assignment when excluding all possible doublets

BEFORE STARTING

Add bam file name for **$bam** and vcf file name for **$vcf**. Use **<(zcat $vcf)** or **<(gzcat $vcf)** if vcf file is compressed.

The options for **--field** are individual genotypes (GT), posterior probability (GP), or genotype likelihood (PL). If using GT **--field** option for, you must include **--geno-error**, which is the genotype error rate.

demuxlet output

The demuxlet software produces 3 output files. Among those, most important are:

1. **[prefix].best** The .best file contains the assignments of the best sample identity (singlet: **SNG-**; doublet: **DBL-**; ambiguous: **AMB-< >**) in the **BEST** column for each cell barcode identified in the **BARCODE** column along with details of the statistics used to determine the best identity.
2. The **[prefix].single** file contains the statistics for matching each cell with each possible sample

For complete descriptions of the generated files and columns in the output, please see the Guidelines.

Getting Started

1 Please select between the two following options:

1. Installating from Docker
2. Installing from source

step case

Installating from Docker

Select this option for docker installation.

2 **Installing docker**

First, get docker for whatever platform you feel comfortable with: https://www.docker.com/get-started

3 **Running demuxlet from docker**

We have created a docker container at **yimmieg/demuxlet** to run demuxlet through docker.

You can run it with

```
$ docker run -v path/to/tutorial:/data yimmieg/demuxlet --sam /data/$bam --vcf /data/$vcf --
field $(GT or GP or PL) --out /data/$filename
```

Analyzing the sample dataset

4 **Create directory and download datasets**

Now, let's first download the data we need. We are now providing a one-stop-shop to download all of the data you need for the tutorial: https://ucsf.box.com/s/vg1bycvsjgyg63gkqsputprq5rxzjl6k. After downloading and unzipping (it's BIG), you should have a directory called **demuxlet.tutorial**.

5 **Run demuxlet**

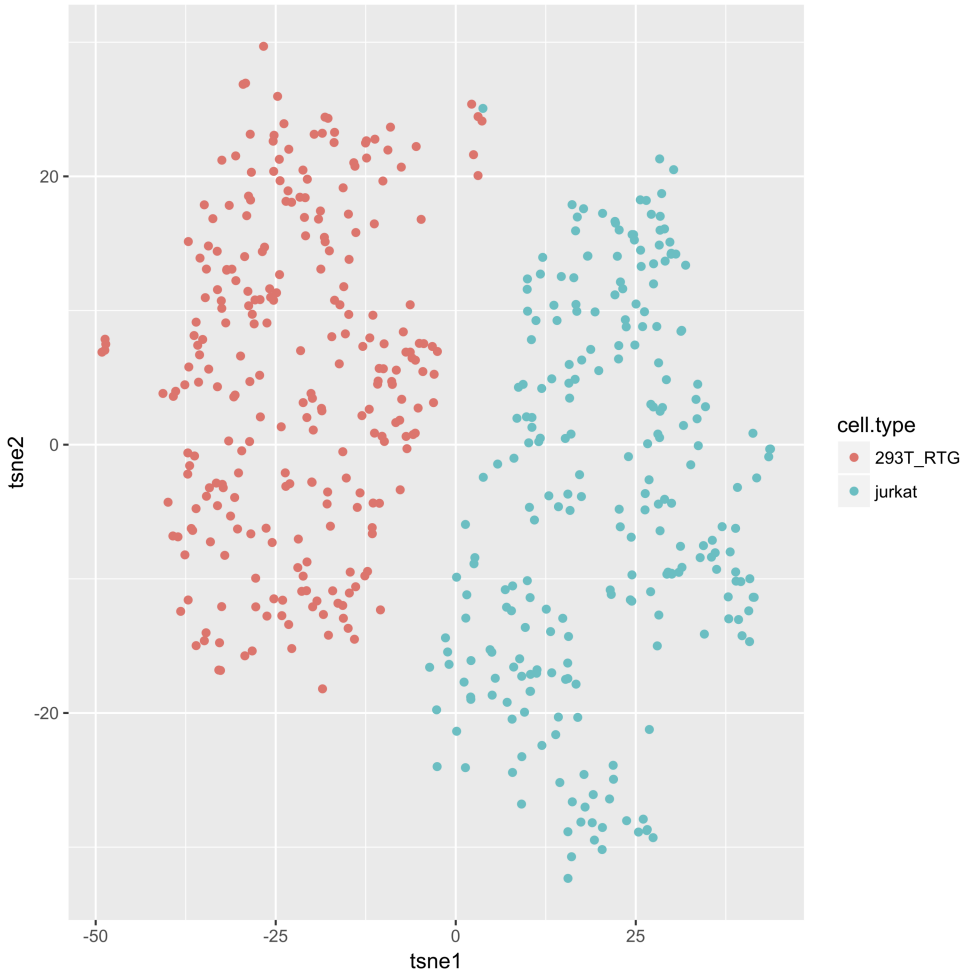
```
$ cd demuxlet.tutorial
$ docker run -v ./:/data yimmieg/demuxlet --sam /data/jurkat_293t_downsampled_n500_full_bam.bam
--vcf /data/jurkat_293t_exons_only.vcf --field GT --out data/jurkat_293t_demuxlet
```

6 **Compare the called genotypes vs transcriptome data**

In this analysis, we will use R to produce a t-SNE (t-Distributed Stochastic Neighbor Embedding) plot of the cells from the 293T:Jurkat 10x experiment with the cells colored by the assignments from the demuxlet pipeline.

```
library(ggplot2);
library(data.table);

## let's read in the barcodes
tsne <- fread("analysis_csv/tsne/projection.csv");
demuxlet <- fread("jurkat_293t_demuxlet.best");

## let's filter for the barcodes that we sampled

df <- data.frame(tsne1=tsne$"TSNE-1"[na.omit(match(demuxlet$BARCODE,tsne$Barcode))],
tsne2=tsne$"TSNE-2"[na.omit(match(demuxlet$BARCODE,tsne$Barcode))],
doublet=sapply(demuxlet$BEST,function(x){strsplit(x,"-")[[1]][[1]]}),
cell.type=sapply(demuxlet$BEST,function(x){strsplit(x,"-")[[1]][[2]]}))

ggplot(aes(tsne1,tsne2,color=cell.type),data=df)+geom_point()
```

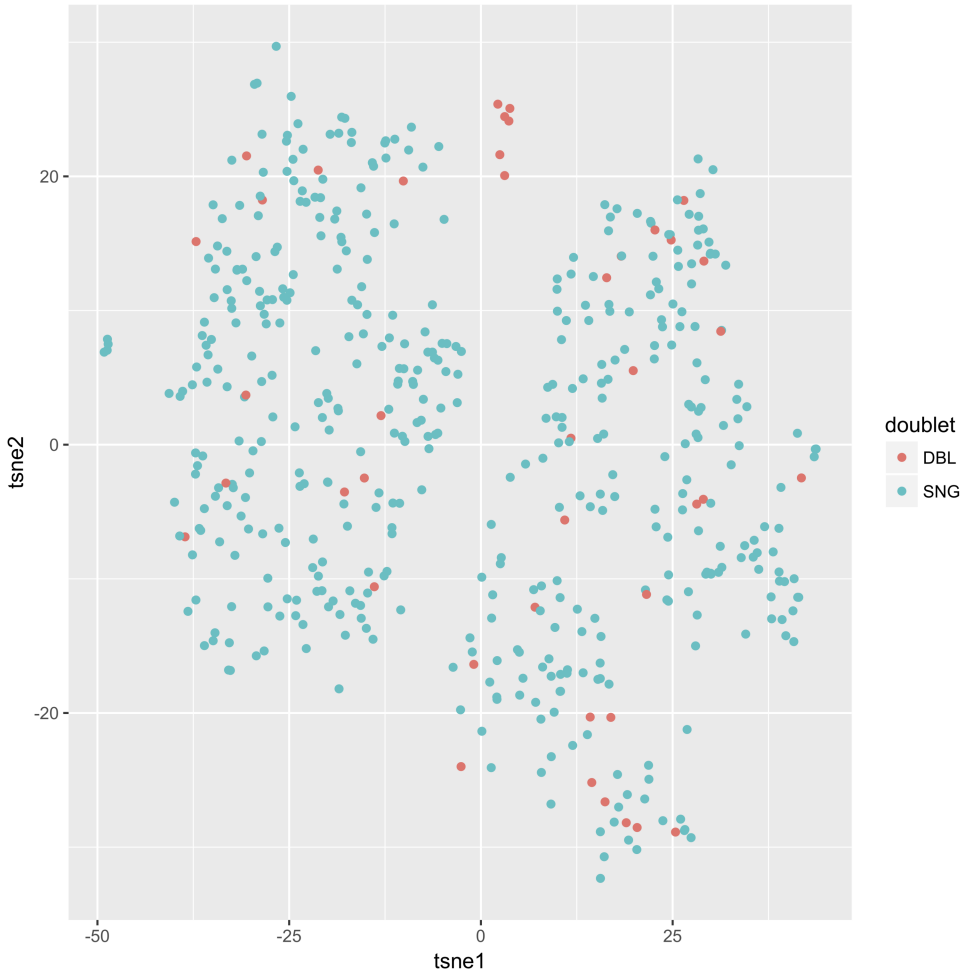
After this, you should get the following image:



7 Let's also take a look at the doublets.

```
ggplot(aes(tsne1,tsne2,color=doublet),data=df)+geom_point()
```

After that, you should get the following image:



Getting Started

step case

Installing from source

Select this option for installing demulex from source.

2 Before installing demuxlet, you need to install htslib in the same directory you want to install demuxlet (i.e. demuxlet and htslib should be siblings).

After installing htslib, you can clone the current snapshot of this repository to install as well

```
$ git clone https://github.com/statgen/demuxlet.git
$ cd demuxlet
$ autoreconf -vfi
$ ./configure  (with additional options such as --prefix)
$ make
$ make install (may require root privilege)
```

Remember to add demuxlet to $PATH

Analyzing the sample dataset

3 Create directory and download datasets

Now, let's first download the data we need. We are now providing a one-stop-shop to download all of the data you need for the tutorial: https://ucsf.box.com/s/vg1bycvsjgyg63gkqsputprq5rxzjl6k. After downloading and unzipping (it's BIG), you should have a directory called **demuxlet.tutorial**.