Software Setup Beginner Guide for KmerVC

Requirements:

Dependencies:

- Python 3.6 (or 2.7.13)
- Pandas 0.20.3
- Numpy 1.13.1
- Scipy 0.19.1
- Bedtools 2.25.0
- Jellyfish 2.3.0

Data Files:

- BED format **OR** VCF format sequence variant file
- FASTA format reference sequence file
- FASTA format control and tumor sequence files **OR** JELLYFISH format control and tumor sequence files.

Command Line Directions for Installing Dependencies:

- Clone or download the KmerVC project from: https://github.com/compbio/kmerVC.git

- Navigate into the kmerVC directory with and install the dependencies using the requirements file with **pip**:
  ```
  cd kmerVC
  pip install -r requirements.txt
  ```

- (Optional) Instead of using pip, you can install the requirements using Anaconda in a virtual environment. The Anaconda Distribution for Python 2.7 can be downloaded from: https://www.anaconda.com/distribution/. Call these commands to create and activate your virtual environment:
  ```
  cd kmerVC
  conda create --name kmervc_env --file requirements.txt
  conda activate kmervc_env
  ```

- Download and install Bedtools from: https://bedtools.readthedocs.io/en/latest/content/installation.html. If using conda, this can be done with the following commands:
  ```
  conda install -c bioconda bedtools=2.25.0
  ```

- Download, extract, and install Jellyfish from: https://github.com/gmarcais/Jellyfish/releases. This can be done using the following commands:
  ```
  wget https://github.com/gmarcais/Jellyfish/releases/download/v2.3.0/jellyfish-2.3.0.tar.gz
  ```

tar xvzf jellyfish-2.3.0.tar.gz
cd jellyfish-2.3.0
make -j 4
make install

- Navigate to the directory containing the folder of Jellyfish binaries and add the Jellyfish executable binaries to a directory in your PATH (/usr/local, /usr/local/bin, etc.)
cp bin/* /usr/local/bin

```
(kmervc_test) ashuaibi@tensorflow-1-vm:~$ ls
bin  include  jellyfish-2.3.0  jellyfish-2.3.0.tar.gz  kmerVC  lib  share  workspace
(kmervc_test) ashuaibi@tensorflow-1-vm:~$ sudo cp bin/* /usr/local/bin
```

- To test that you have successfully downloaded Bedtools and Jellyfish, solely run the commands *bedtools* and *jellyfish* and observe that the software information and usage is displayed as below:

```
(kmervc_test) ashuaibi@tensorflow-1-vm:~/kmerVC$ bedtools
bedtools is a powerful toolset for genome arithmetic.

Version:   v2.29.2
About:     developed in the quinlanlab.org and by many contributors worldwide.
Docs:      http://bedtools.readthedocs.io/
Code:      https://github.com/arq5x/bedtools2
Mail:      https://groups.google.com/forum/#!forum/bedtools-discuss

Usage:     bedtools <subcommand> [options]

The bedtools sub-commands include:
```

```
(kmervc_test) ashuaibi@tensorflow-1-vm:~/kmerVC$ jellyfish
Too few arguments
Usage: jellyfish <cmd> [options] arg...
Where <cmd> is one of: count, bc, info, stats, histo, dump, merge, query, cite, mem, jf.
Options:
  --version       Display version
  --help          Display this message
```

Directions for Installing Data Files:

This will carry you through the installation of the data files required for execution of the example runs in the kmerVC/examples directory.

All data files can be downloaded from https://dna-discovery.stanford.edu/publicmaterial/software/kmervc/. To carry out the examples, download all the files in the **example** and **reference** directories and place them in the **kmerVC/examples/resources** directory on your machine. This can be done through the command line with curl:
curl -LOk https://dna-discovery.stanford.edu/publicmaterial/software/kmervc/example/normal-1.fq

This can be done for every file in the specified download directories where the argument passed to curl is the link address of the file.

You are now ready to proceed with running the script.

Running KmerVC:

The software is run fully through the command line with specified command line arguments. All possible arguments are enumerated below with short descriptions:

required positional arguments: {compare}

required keyword arguments:

- -k, --kmer_size KMER_SIZE: Size of kmer to use for analysis
- -o, --output_name OUTPUT_NAME: Output file directory name
- -v, --vcf VCF_INPUT: Input vcf file **OR** -b, --bed BED_INPUT: Input bed file

fastq_group arguments: fastq input files

- -t1, --test1 TEST_FASTQ1: Fastq file 1 from test sample
- -t2, --test2 TEST_FASTQ2: Fastq file 2 from test sample
- -c1, --control1 CONTROL_FASTQ1: Fastq file 1 from control sample
- -c2, --control2 CONTROL_FASTQ2: Fastq file 2 from control sample

jellyfish_group arguments: jellyfish input files

- -j1, --jellyfish_test JELLYFISH_TEST: Jellyfish file of test input
- -j2, --jellyfish_control JELLYFISH_CONTROL: Jellyfish file of control input

optional arguments:

- -h, --help : show usage help message and exit
- -fi, --reference_genome_fasta REFERENCE_GENOME_FASTA: Reference genome fasta file to use, if different than default
- -m, --microsatellite Flag: indicating if doing microsequence analysis with respective vcf file
- -r, --rna Flag: indicating if doing RNA analysis
- -poi, --poisson Flag: indicating if using doing poisson distribution: for variant analysis
- -a, --alpha ALPHA: Alpha value used in hypothesis testing

Test :

In this example, we will perform an analysis of the differences between normal and tumor sequence files using variant information available in the **variants.bed** file in the **resources** directory starting with fasta file sample input.

- First, navigate to the **examples/fastq_start_example** directory on your machine.

- Create the jellyfish count file for the reference genome **chrT.fa** located in the resources directory. Do so with the following command:
  jellyfish count -m 30 -s 100M -t 24 -C -o chrT_30mer.jf ../resources/chrT.fa

- Call the program specifying the required positional argument, the required keyword arguments, and the fastq group arguments:
  python ../../kmervc.py compare -k 30 -t1 ../resources/tumor-1.fq -t2 ../resources/tumor-2.fq -c1 ../resources/normal-1.fq -c2 ../resources/normal-2.fq -b ../resources/variants.bed -o example_1 -fi ../resources/chrT.fa

- Your output will be available in your current directory and named **example_1_variant_summary_table.txt**. All intermediate files are located in the **example_1** directory.