QuicK-mer v1.0 User Manual

Commands in 4 - 6 assume you start in the root directory of QuicK-mer.

1. Prerequisites
   Before using the QuicK-mer CNV pipeline, here is a list of programs required:
   1) Jellyfish 2
   2) Python 2.7
   3) matplotlib 1.1.0 or later
   4) samtools (only necessary if input file is in BAM format)

2. What is QuicK-mer?
   QuicK-mer is an efficient, paralog-sensitive CNV estimation pipeline based around Jellyfish-2. It counts the occurrences of each predefined k-mer inside Illumina sequencing data and normalizes to correct copy number based on pre-defined control regions. QuicK-mer supports both FASTQ and BAM format as input.

3. Download QuicK-mer
   QuicK-mer is distributed as a source package on github. Grab QuicK-mer using the following command:

   ```
   git clone https://github.com/KiddLab/QuicK-mer.git
   ```

4. Compile
   There are 3 required executables written in a compiled language to increase pipeline efficiency. Pre-compiled binaries are included in the distribution. If an OS/CPU not supported by the existing distributed binary is used, the user should compile the programs.

   1) KmerCor
   This is the core program for GC bias estimation and depth normalization in QuicK-mer. To compile use the below command:

   ```
   cd kmer/
   fpc -O KmerCor.lpr
   ```

   2) kmer2window
   This program is used to convert depth data into copy number in a bedGraph format based on predefined window sizes and control regions. Each window contains a fixed number of k-mers. Note that the last window at the end of each chromosome may contain fewer.

   ```
   cd kmer/
   g++ -O -o kmer2window kmer2window.cpp
   ```

   3) CorDepthCombine
   The CorDepthCombine program is used to merge each GC-corrected sequencing library (or sequencing lane) from the same sample together. Each sequencing

library (or lane) usually contains distinctive GC bias patterns and should be run through QuicK-mer separately.

```
cd kmer/
fpc -O CorDepthCombine.lpr
```

5. Installation

QuicK-mer does not need to be installed; all you need to do is add the application folders to your path directory.

```
QuicK-mer/
QuicK-mer/kmer/
```

To do so in unix-like systems, open your `.bashrc` file in the home directory using a text editor or with vi. Add the following line:

```
PATH=$PATH: path_before_Quick-mer/QuicK-mer/:path_before_Quick-mer
/QuicK-mer/kmer/
```

Then execute using:

```
source .bashrc.
```

6. Premade 30-mer lists available for download

The following genomes have unique 30-mer catalogs ready for download (http://kiddlabshare.umms.med.umich.edu/public-data/QuicK-mer/Ref/):

1) mm10
2) hg19
3) panTro4
4) canFam3.1

7. Description of supporting files

Once extracted, each folder contains six files to support the QuicK-mer pipeline. Using hg19 as an example, below is a list of the six files.

```
hg19_kmer.bed
k30_hg19_GC.bin
k30_hg19_CN2.bin
hg19_50_window.bed
hg19_500_window.bed
hg19_uniq.bc
```

`hg19_kmer.bed` is the predefined 30-mer list in bed format. It contains the location of each 30-mer and its sequence in the last column. `k30_hg19_GC.bin` is the GC content of the surrounding 400bp with the 30-mer in the center. `k30_hg19_CN2.bin` records a true/false flag with each byte per 30-mer indicating if the 30-mer is **excluded** from control region. Hence, 0x00 30-mers are used for building the GC bias curve. `hg19_50_window.bed` and `hg19_500_window.bed` are the window files in 50 or 500 30-mers per bin used for track displaying and smoothing. User can easily redefine the window in section 9. Finally, `hg19_uniq.bc` is the bloom counter for Jellyfish-2 which will speed up the QuicK-mer counting process and reduce I/O load.

8. Working Example

Here we use an example using public data from the NCBI short read archive to demonstrate QuicK-mer usage. Here we assume you are in your working directory.

1) Download NA19240 sequencing file from SRA.

```
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR136/SRR1364052/SRR1364052.sra
fastq-dump -O SRR1364052 --split-files --gzip SRR1364052.sra
```

2) Download hg19 30-mer reference

We premade the 30-mer list for hg19 reference genome.

```
wget http://kiddlabshare.umms.med.umich.edu/public-data/QuicK-mer/Ref/hg19.tar.gz

tar xzfv hg19.tar.gz
```

3) Running the QuicK-mer

Add the following command to a job submission script and request 2 CPU cores with 35GB of total memory.

```
cd SRR1364052/
start_kmer_pipeline.py 19485_ATGTCA_L007\*.fastq.gz -o 19485_ATGTCA_L007 hg19/
```

This process usually takes 6 hours. Once done, QuicK-mer will generate 3 files under the `SRR1364052/` directory:

```
19485_ATGTCA_L007_result.bin
19485_ATGTCA_L007.txt
19485_ATGTCA_L007.PNG
```

The text file and PNG image record the GC-depth bias in the control region. The binary file `19485_ATGTCA_L007_result.bin` contains all the GC-corrected depths for all 30-mers.

4) Merge data

To merge multiple GC-corrected depth files, move all the `*_result.bin` files into a directory and execute the following command from that directory:

```
ls *_result.bin > sample_name.txt
CorDepthCombine -l sample_name.txt
```

The result `sample_name_merged.bin` will contain the merged depth data for the files specified in `sample_name.txt` text file.

5) Integrate browser track

Finally, the user needs to convert the depth file into the bedGraph format based on predefined or user-defined windows.

```
kmer2window 19485_ATGTCA_L007_result.bin ../hg19/k30_hg19_CN2.bin
../hg19/hg19_500_window.bed > 19485_copy_number.bedGraph
```

The `hg19_500_window.bed` is a file specifying the genome location and number of k-mers in each window. The file `19485_ATGTCA_L007_result.bin` can be substituted with the merged binary file `sample_name_merged.bin` when dealing with samples from multiple libraries.

This file can be further indexed and compressed into UCSC bigwig format and displayed using the UCSC genome browser.

9. Custom k-mer list

The user can define the k-mer list for any genome besides the premade ones listed in Step 6. The list of k-mers should have the following tab-delimited format:

```
chr1    10454   10484   chr1-10455      CTAACCCTAACCCTCGCGGTACCCTCAGCC
chr1    10455   10485   chr1-10456      CGGCTGAGGGTACCGCGAGGGTTAGGGTTA
chr1    10456   10486   chr1-10457      AACCCTAACCCTCGCGGTACCCTCAGCCGG
chr1    10457   10487   chr1-10458      ACCCTAACCCTCGCGGTACCCTCAGCCGGC
chr1    10458   10488   chr1-10459      CCCTAACCCTCGCGGTACCCTCAGCCGGCC
chr1    10459   10489   chr1-10460      CCTAACCCTCGCGGTACCCTCAGCCGGCCC
```

The first three columns define the genomic location of k-mer with the fifth column defines the k-mer sequence. The file must be in tab-delimited format and sorted based on genomic location.

10. Generate supporting files for custom k-mer list approach

Once a custom k-mer list is given, user could easily create the 3 essential axillary files using the built in command line tools. Below, we use the hg19 30-mer list as a starting point to create the axillary files.

1) Bloom Counter

The bloom counter double counts each k-mer in the list and then feeds it into the Jellyfish-2 for bloom counter generation. Essentially, this step marks predefined k-mers as "high frequency" during the actual counting process. This will reduce I/O when building the k-mer database.

```
cd hg19/
make-fasta-from-kmer.py hg19_kmer.bed | jellyfish-2 bc -C -m 30 -s 3G -t
16 -o kmer/ hg19_uniq.bc /dev/fd/0
```

2) GC content

To generate GC content binary file, you'll need the reference genome files in FASTA format with sequence layout as 50bp per line.

```
cd hg19/
generate_GC_bin.py hg19_kmer.bed genomes/hg19/fasta/ k30_hg19_GC.bin
```

3) Window segments

Use the following command to make the window file for an existing k-mer list. The first argument "50" indicates 50 k-mers per window. The user can increase

this value in order to trade finer resolution for minimization of the signal-to-noise ratio.

```
make_window_kmer.py 50 hg19_kmer.bed > hg19_50_window.bed
```