Bucknell High Performance Research Computing Network
Kraken2 Documentation

Lourenco Martins, Go Agata & Ken Field, Ph.D.

Program Location:
BisonNet

Version:
2.1.1

BisonNet Module:

Program Description:
Kraken is a taxonomic sequence classifier that assigns taxonomic labels to short DNA reads. It does this by examining *k*-mers within a read and querying a database with those *k*-mers. *K*-mers are substrings of length contained within a biological sequence.

Authors/Background:
Kraken performs an ultrafast metagenomic sequence classification using exact alignments. Kraken was written by Derrick E Wood and Steven L Salzberg. Here is the link to the Kraken paper for citation:
https://genomebiology.biomedcentral.com/articles/10.1186/gb-2014-15-3-r46

Input Document Files:
-FASTA file typically, though there are flags for other input files

Note: there has been issues with the -fastq_input flag and so gunzipped fq files seem to work (example below)

Standard Output Information:
Kraken2 has two standard outputs:

1. A kraken report output file which indicates which taxa were identified in the sequences and the percentage and number of fragments that mapped back to each taxon.
2. A kraken text file that indicates how each input sequence was classified as well the the taxon that each *k*-mer mapped back to, if the *k*-mer was able to be classified.

Troubleshooting Common Errors:

If you receive an error message when running Kraken2 start with the following:

- Check for punctuation and spelling errors in the script
- Check the path to the files (specifically the database and the input files)
- Check you that have asked for enough memory (in the example that follows 180 GB was required)

If these do not resolve the issue, additional help can be found at the Kraken2 GitHub Issue Board: https://github.com/DerrickWood/kraken2/issues

Running Kraken2 on BisonNet:

Mandatory Arguments:

kraken2 –db [DATABASE] [INPUT FILES] [OTHER OPTIONS]

Optional Arguments (obtained directly from author's documentation)
Multithreading: Use the `--threads NUM` switch to use multiple threads.
Quick operation: Rather than searching all *k*-mers in a sequence, stop classification after the first database hit; use `--quick` to enable this mode. Note that `--min-hits` will allow you to require multiple hits before declaring a sequence classified, which can be especially useful with custom databases when testing to see if sequences either do or do not belong to a particular genome.
Sequence filtering: Classified or unclassified sequences can be sent to a file for later processing, using the `--classified-out` and `--unclassified-out` switches, respectively.
Output redirection: Output can be directed using standard shell redirection (`|` or `>`), or using the `--output` switch.
FASTQ input: Input is normally expected to be in FASTA format, but you can classify FASTQ data using the `--fastq-input` switch.
Compressed input: Kraken can handle gzip and bzip2 compressed files as input by specifying the proper switch of `--gzip-compressed` or `--bzip2-compressed`.
Input format auto-detection: If regular files are specified on the command line as input, Kraken will attempt to determine the format of your input prior to classification. You can disable this by explicitly specifying `--fasta-input`, `--fastq-input`, `--gzip-compressed`, and/or `--bzip2-compressed` as appropriate. Note that use of the character device file `/dev/fd/0` to read from standard input (aka `stdin`) will **not** allow auto-detection.
Paired reads: Kraken does not query *k*-mers containing ambiguous nucleotides (non-ACGT). If you have paired reads, you can use this fact to your advantage and increase Kraken's accuracy by concatenating the pairs together with a single `N` between the sequences. Using the `--paired` option when running `kraken` will automatically do this for you; simply specify the two mate pair files on the command line. We have found this to raise sensitivity by about 3 percentage points over classifying the sequences as single-end reads. For more information about paired reads input/output, see Paired Reads

On BisonNet:

(I highly suggest writing your script in a text editor like BBedit and then copying & pasting the script into a shell script onto the command line)

1. Enter an interactive session
    a. srun -n 1 -p short --pty /bin/bash
2. Create Script ***Note: This script has been optimized for paired read fq files***

Script Components:

1. Hash-bang (Tells the computer you are writing in bash)
2. Enter the required arguments to use the queue
3. Load the Kraken2 module
4. Specify the absolute path to the kraken database
5. Run the script using the command “sbatch {insert_script_name_here}.sh

Template Script:

```
#!/bin/bash
#SBATCH -p medium # partition (queue)
#SBATCH -N 1 # (leave at 1 unless using multi-node specific code)
#SBATCH -n 8 # number of cores
#SBATCH --mem-per-cpu=32G # memory per core
#SBATCH --job-name="Kraken2_Template_Script" # job name
#SBATCH -o slurm.%N.%j.stdout.txt # STDOUT
#SBATCH -e slurm.%N.%j.stderr.txt # STDERR
#SBATCH --mail-user=yourbucknellid#@bucknell.edu # address to email
#SBATCH --mail-type=END # mail events (NONE, BEGIN, END, FAIL, ALL)

#Description:This script will run kraken2 on input files
#Usage:sbatch run_template_Kraken2_script.sh
#print "start" to log file

echo "start"
#copy the input data files
#the file needs to be in gzip format
cp ~/path/to/input_file.fq.gz

#load the module for Kraken2
module load kraken2
#run Kraken2 with input
for i in *.fq do echo "Running Kraken on" $i
kraken2 --threads 16 --db /home/kfield/kraken-nt-CoV2-db \
 --paired --classified-out classified-out.R#.fq \
--unclassified-out unclassified-out.R#.fq \
 --confidence 0.5 \
 --output ${i/.fq.gz/.kraken.txt} \
 --report ${i/.fq.gz/.kreport2}  \
$i ${i/_1/_2}

#print "end" when done
echo "done kraken"
#remove Kraken2 module
module unload kraken2
#delete the data copies
rm input_file.fq.gz
```

Note: The script lines with backslashes “\” should all be on a continuous line of code

After running Kraken2 here are things you can do:

1. Reading the Kraken2 Report file:

```
82.53  35718678        35718678        U       0       unclassified
17.47  7562035 248279  R       1       root
16.90  7313267 85907   R1      131567    cellular organisms
16.61  7190253 533342  D       2759        Eukaryota
15.37  6652790 156814  D1      33154         Opisthokonta
14.72  6372393 13427   K       33208           Metazoa
14.69  6358966 42670   K1      6072              Eumetazoa
14.59  6316296 438209  K2      33213               Bilateria
13.43  5814188 279487  K3      33317                 Protostomia
12.79  5534688 20149   K4      1206794                 Ecdysozoa
12.74  5514536 1023    K5      88770                     Panarthropoda
12.74  5513513 210402  P       6656                        Arthropoda
12.25  5303099 28019   P1      197563                        Mandibulata
12.19  5275080 125998  P2      197562                          Pancrustacea
11.90  5149013 5745    P3      6960                              Hexapoda
11.88  5143268 3338    C       50557                               Insecta
11.88  5139930 3941    C1      85512                                 Dicondylia
11.87  5135989 31957   C2      7496                                    Pterygota
11.79  5104032 680591  C3      33340                                     Neoptera
10.22  4422960 633199  C4      33392                                       Holometabola
 8.76  3789692 30605   O       7041                                          Coleoptera
 8.69  3759087 765878  O1      41084                                           Polyphaga
 6.92  2993116 17935   O2      41087                                             Elateriformia
 6.87  2975181 277472  O3      71193                                               Elateroidea
 6.23  2697709 308597  F       7049                                                  Lampyridae
 5.52  2389112 307977  F1      433514                                                  Lampyrinae
 4.37  1892508 16      G       7053                                                      Photinus
 4.37  1892492 1892492 S       7054                                                        Photinus pyralis
 0.44  188627  489     G       94998                                                     Ellychnia
 0.42  181834  181834  S       94999                                                       Ellychnia corrusca
 0.01  6304    6304    S       454433                                                      Ellychnia californica
 0.00  88      0       O2      41088                                             Cucujiformia
 0.00  72      0       O3      71528                                               Chrysomeloidea
 0.00  72      0       F       27439                                                 Chrysomelidae
 0.00  61      0       F1      63710                                                   Galerucinae
```

The kraken report tells the user to which taxon sequence fragments mapped back to and is tab-delimited with each column meaning the following:

1) Percentage of fragments covered by the clade rooted at this taxon
2) Number of fragments covered by the clade rooted at this taxon
3) Number of fragments assigned directly to this taxon
4) A rank code indicating (U)nclassified, (R)oot, (D)omain, (K)ingdom, (P)hylum, (C)lass, (O)rder, (F)amily, (G)enus, (S)pecies. Taxa that are not any of these 10 ranks have a rank code that is formed by using the rank code of the closest ancestor rank with a number indicating the distance from that rank. Ex. Lampyrinae has a rank code of “F1” because it is a subfamily one step below Lampyridae (the firefly family).
5) The Taxonomic ID number from NCBI
6) Indented Scientific Name

2. The Kraken output text files look like this:

```
U       A01050:44:HWNGNDSXX:2:1101:3269:1016    0       150|150 7054:45 0:71 |:| 7054:6 0:3 7054:2 0:5 7054:2 0:98
U       A01050:44:HWNGNDSXX:2:1101:3992:1016    0       150|150 7054:2 0:82 7054:4 33340:5 0:5 7054:9 10001:5 7054:
U       A01050:44:HWNGNDSXX:2:1101:4173:1016    0       150|150 0:91 7054:7 0:18 |:| 0:40 7054:26 0:4 7054:2 0:37 7
U       A01050:44:HWNGNDSXX:2:1101:5873:1016    0       150|150 0:24 7054:5 0:1 7054:4 0:5 7054:2 0:8 72781:3 7054:
U       A01050:44:HWNGNDSXX:2:1101:6777:1016    0       150|150 0:13 7054:1 0:5 7054:11 0:5 7054:5 0:36 7054:8 0:32
U       A01050:44:HWNGNDSXX:2:1101:7337:1016    0       150|150 7054:7 0:15 7054:5 0:16 7054:6 0:9 7054:3 0:55 |:|
U       A01050:44:HWNGNDSXX:2:1101:9706:1016    0       150|150 0:24 7054:7 33213:5 2759:6 0:4 8673:5 0:7 64144:1 0
U       A01050:44:HWNGNDSXX:2:1101:10050:1016   0       150|150 0:32 7054:30 0:54 |:| 0:91 7054:25
```

Each sequence classified by Kraken has a single line of output that has 5 tab-delimited fields:

1) One letter code indicating whether the sequence was classified or unclassified
2) The sequence ID obtained from the input FASTA/FASTQ file
3) The taxonomy ID Kraken2 used to label the sequence (0 if the sequence is unclassified)
4) The length of the sequence in bp. For paired read data (which the above is) this will be a string containing the lengths of the two sequences in bp separated by a pipe character. For the above it is 150|150 (which is good because the input FASTQ files were 150 bp PE reads).
5) A space-delimited list indicating the LCA mapping of each *k*-mer in the sequence(s).
    a. For example the first line is read as
        i. 45 *k*-mers mapped to taxonomy ID 7054
        ii. The next 71 *k*-mers are not in the database
        iii. The next 6 mapped to taxonomy ID 7054
        iv. The following 3 were not in the database

    And so on and so forth…

3. If creating a *De Novo* assembly, you can use the unclassified output fq files to create an assembly that doesn't have any contaminants (Note: check the Kraken2 reports to make sure sequences from your study species weren't allocated into the classified output file.
    a. Example, I am creating a *de novo E. corrusca* transcriptome but as you can see below 0.42% or 217939 fragments were classified for this input file. I want to make sure I include those fragments when I'm assembling my transcriptome

```
5.62  2917159 2       G       7053                    Photinus
5.62  2917157 2917157 S       7054                      Photinus pyralis
0.43  222974  651     G       94998                   Ellychnia
0.42  217939  217939  S       94999                     Ellychnia corrusca
0.01  4383    4383    S       454433                    Ellychnia californica
0.00  1       0       G1      2632667                   unclassified Ellychnia
0.00  1       1       S       1932580                     Ellychnia sp. CO1197
```

Sources:

Wood, Lu and Langmead. Improved metagenomic analysis with Kraken 2. *Genome Biology*. (2020). Retrieved on April 2021 from https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1891-0

Bucknell University. (2020). BisonNet. Retrieved November 10, 2020, from http://bisonnet.bucknell.edu/