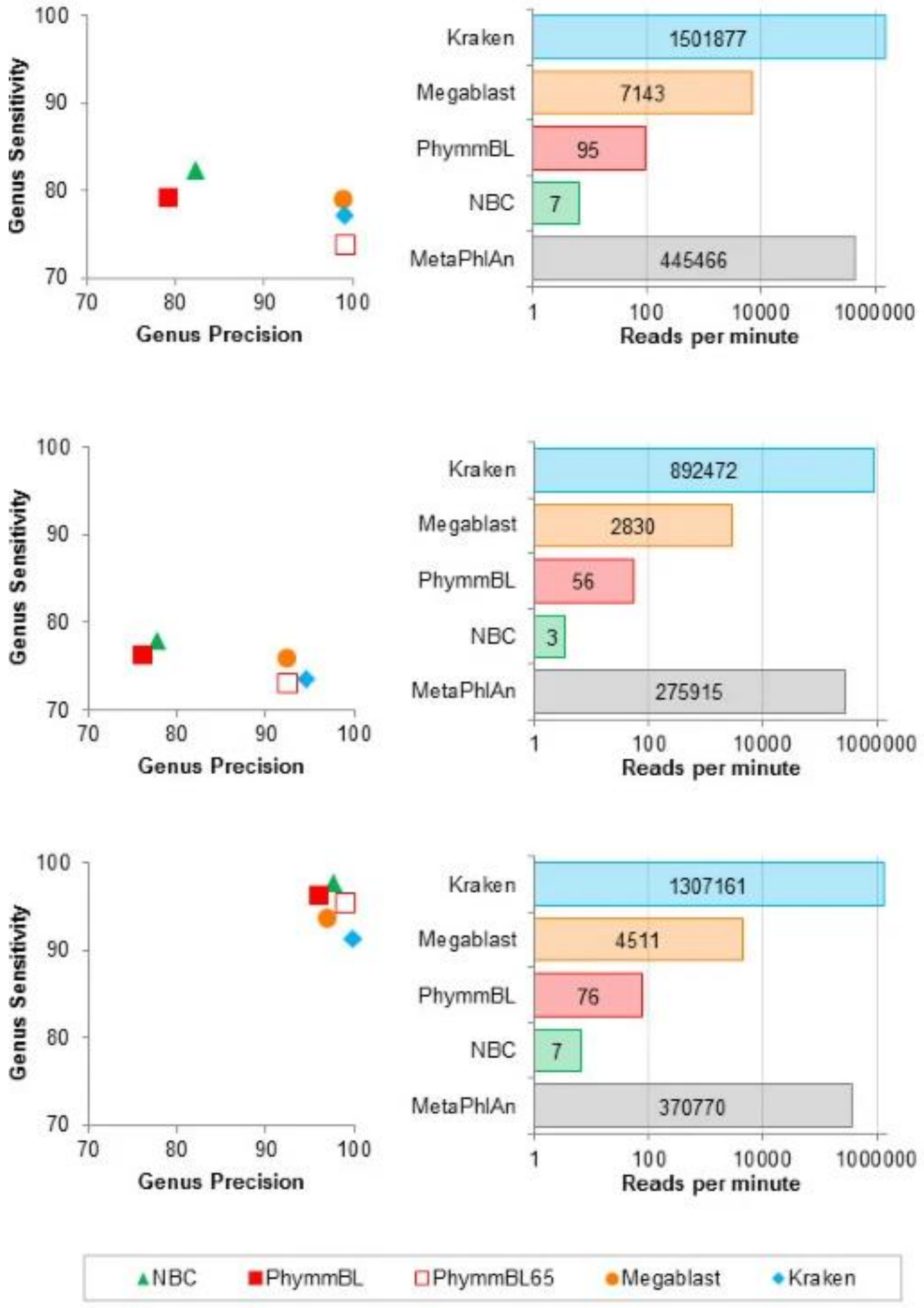
TUTORIAL 8 KRAKEN

BY CODY SCULLY & CHRISTOPHER UZOKWE

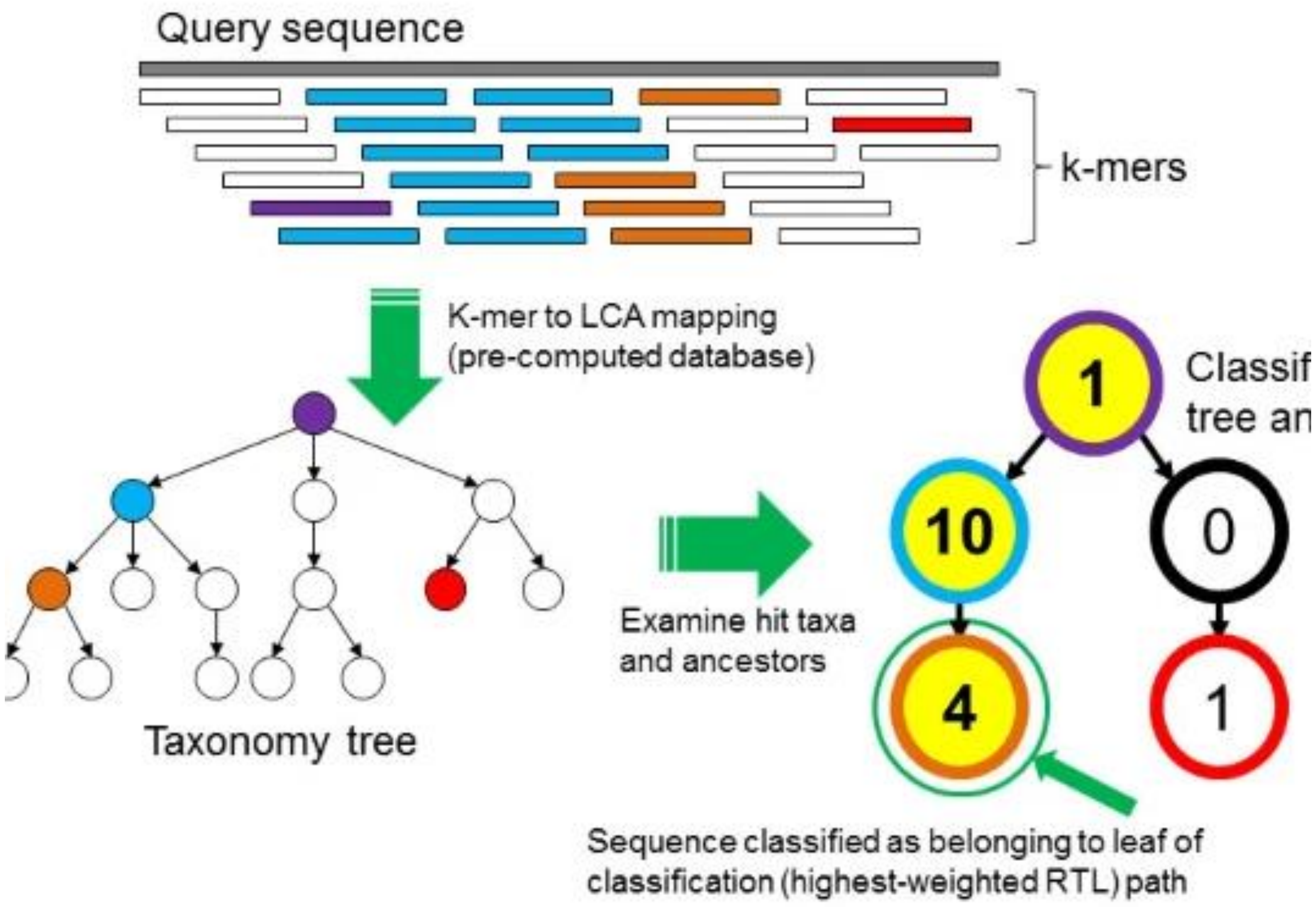
TAXONOMIC CLASSIFICATION

- Blast (Magablast)
- The Naïve Bayes Classifier (NBC)
- PhymmBL
- MatePhlAn
- Kraken

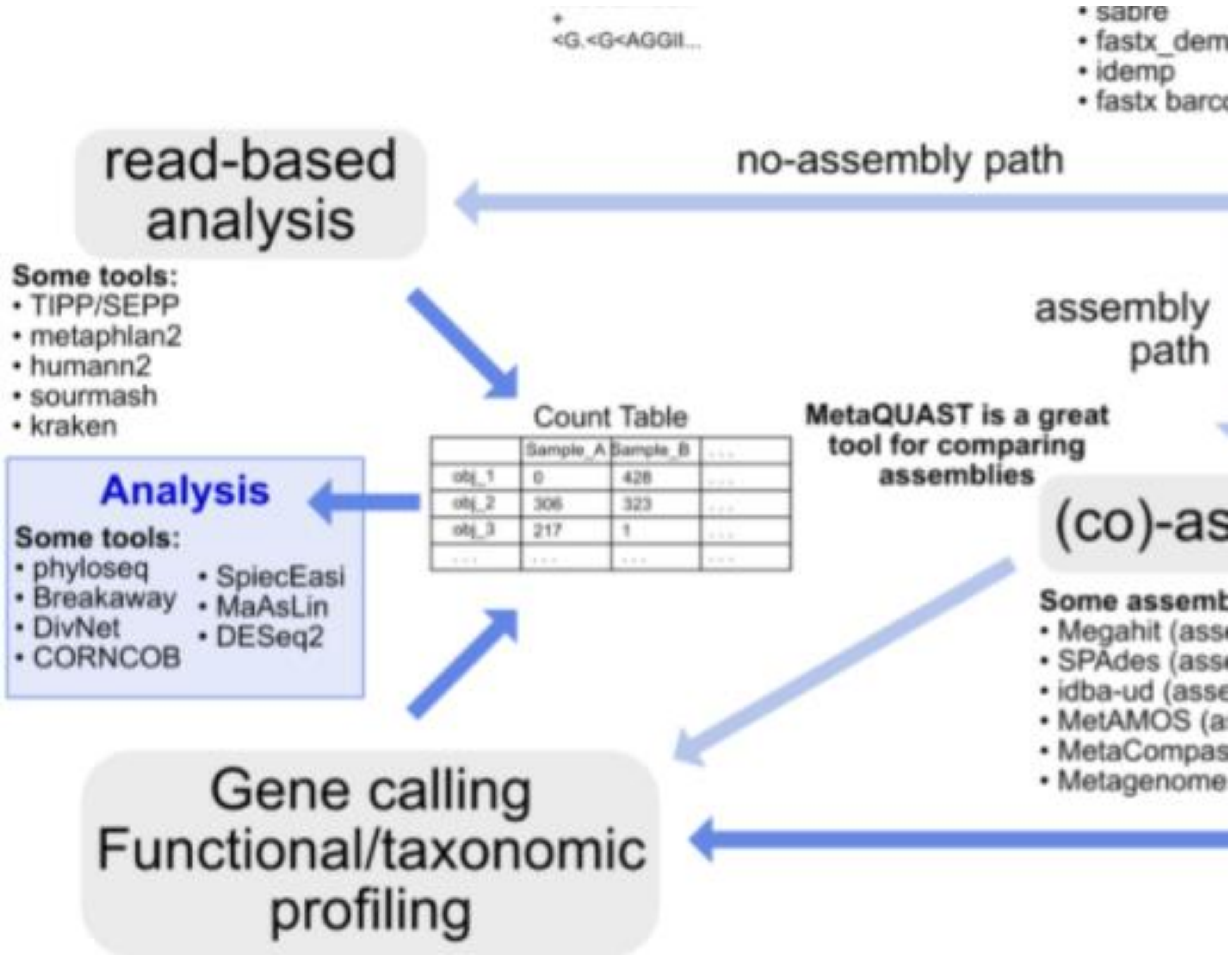


OVERVIEW OF KRAKEN

- This is a Program that reads in Group of unmapped data
- Classifies the data using k-mers to map the LCA
- Creates a Taxonmy tree with the Classifications
- Removes any unclassified leaves from the tree



OVERVIEW OF METAGENOMICS WORKFLOW



+
<G.<G<AGGII...

- sabre
- fastx_dem
- idemp
- fastx barc

no-assembly path

read-based analysis

Some tools:
- TIPP/SEPP
- metaphlan2
- humann2
- sourmash
- kraken

assembly path

Count Table

	Sample_A	Sample_B	...
obj_1	0	428	...
obj_2	306	323	...
obj_3	217	1	...
...	...	...	...

MetaQUAST is a great tool for comparing assemblies

Analysis

Some tools:
- phyloseq
- Breakaway
- DivNet
- CORNCOB
- SpiecEasi
- MaAsLin
- DESeq2

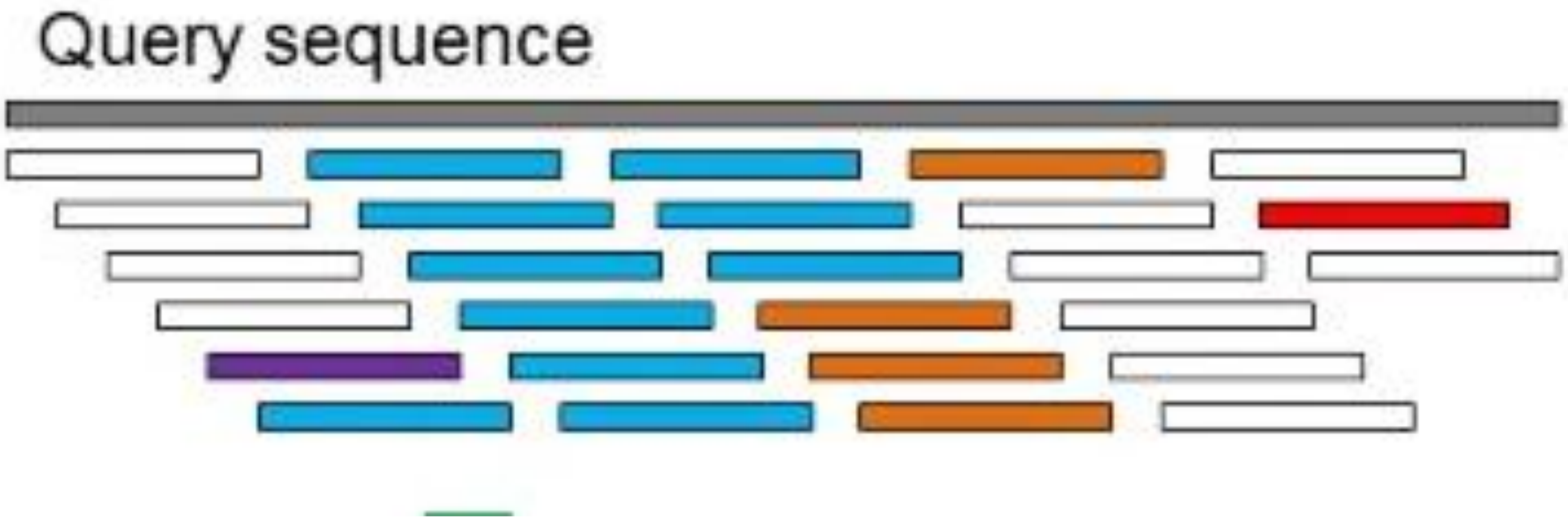
(co)-as

Some assemb
- Megahit (ass
- SPAdes (ass
- idba-ud (asse
- MetAMOS (a
- MetaCompas
- Metagenome

Gene calling Functional/taxonomic profiling

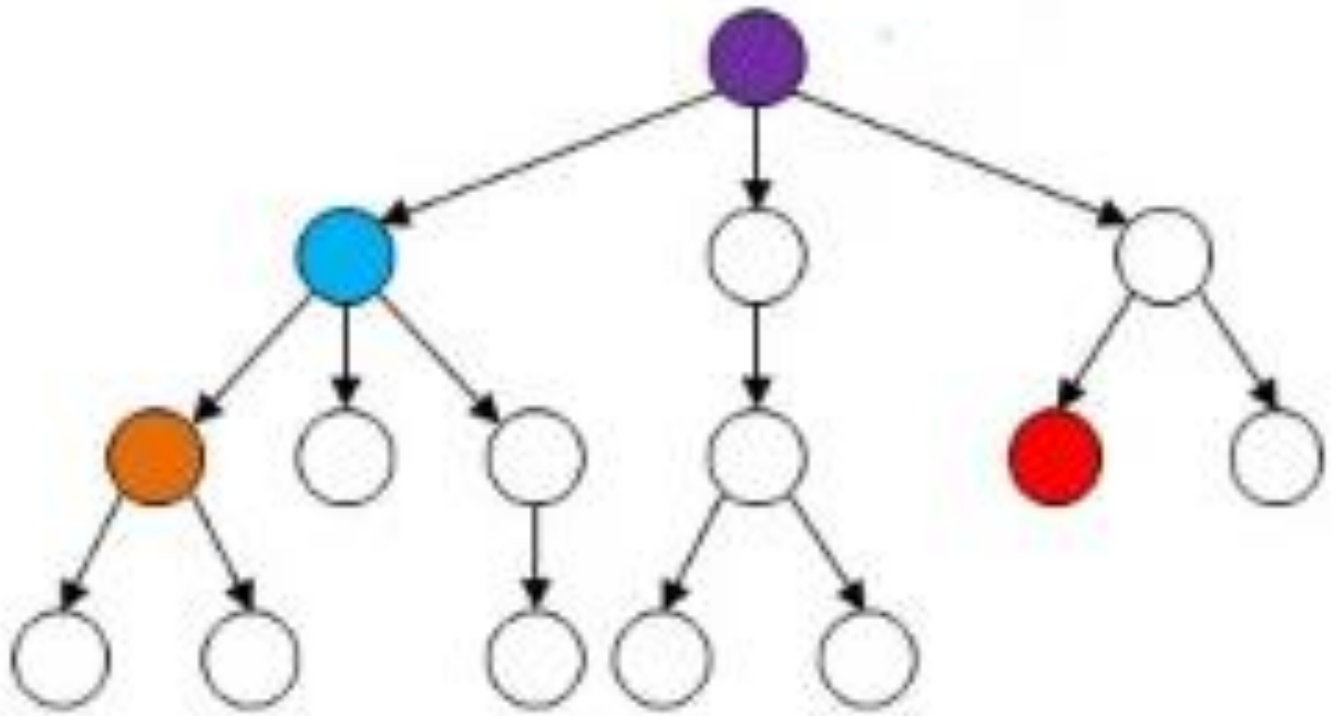
QUERY SEQUENCE

- The First step that Kraken takes is to sequence the data set
  - This is done by a library of metagenome data
- Every k-Mer is either
  - **Classified** with the data from the library
  - **Unclassified** if nothing is found for them.



TAXONOMY TREE

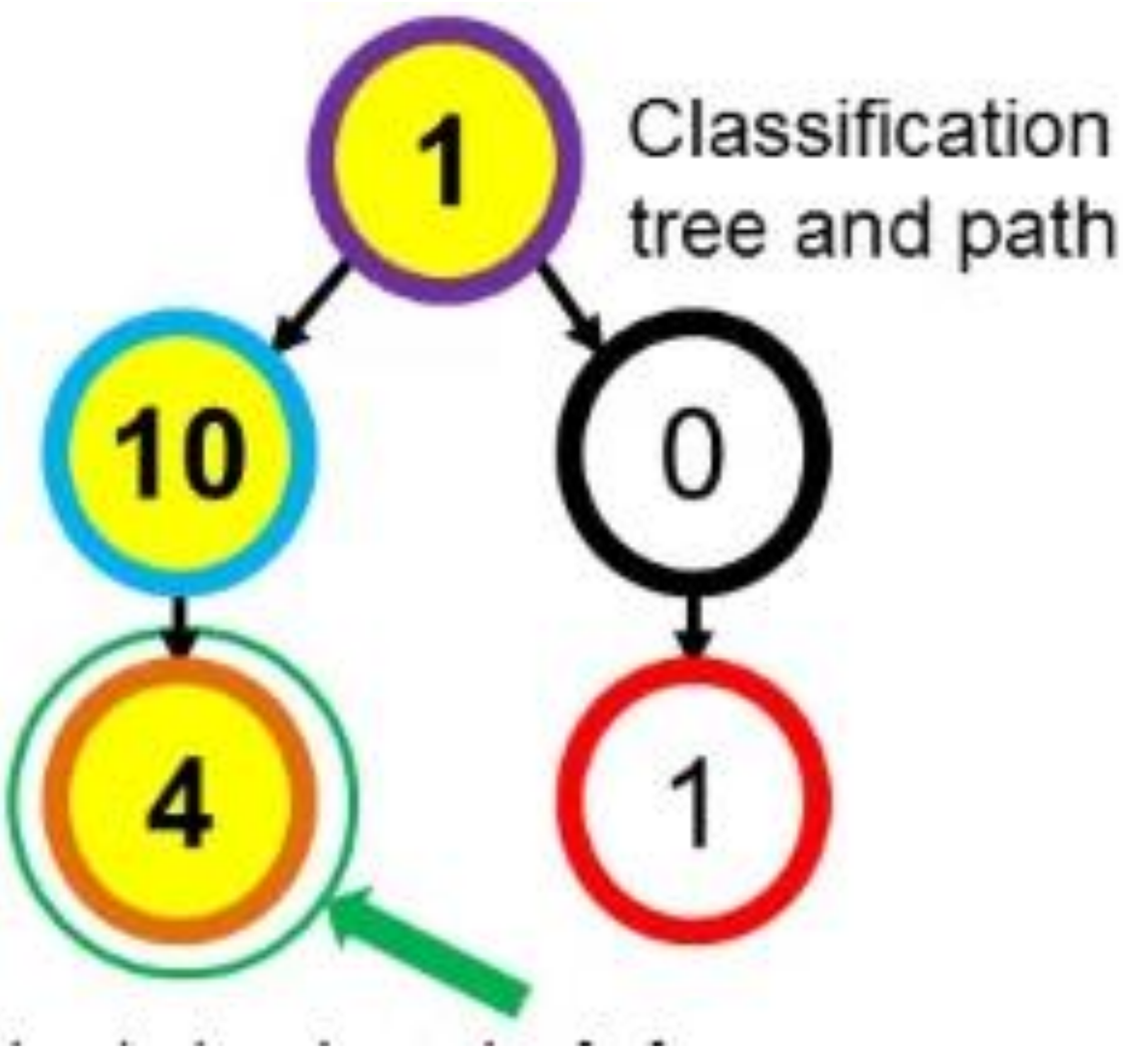
- Each K-mer is mapped to the lowest common ancestor (LCA) of the genomes that contains that k-mer



Taxonomy tree

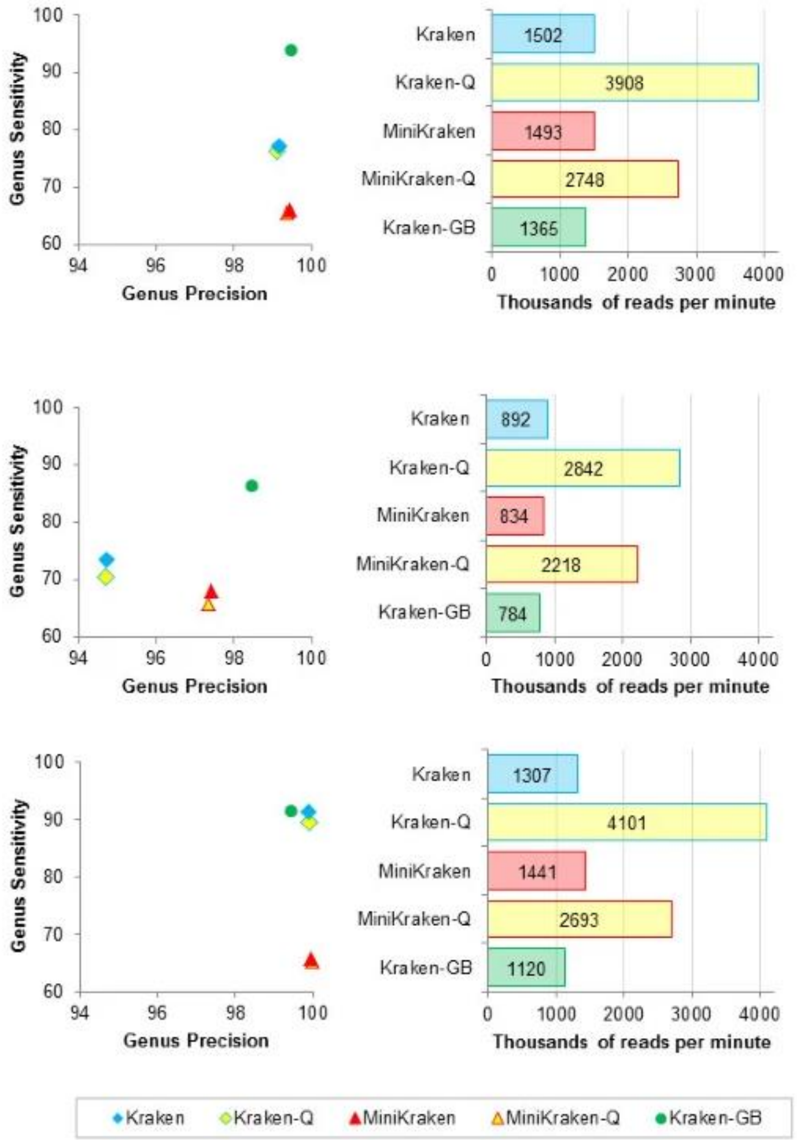
CLASSIFICATION TREE AND PATH

- Finally the unclassified K-mers are removed and the tree is reduced and weighted/scored



KRAKEN VARIANCE

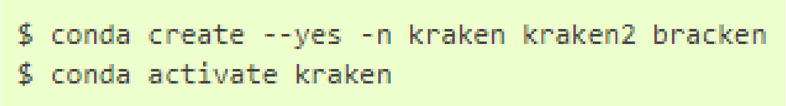
- Kraken
  - Needs 70 GB of RAM
- Kraken-Q/MiniKraken-Q
  - Faster then Kraken /MiniKraken
  - Decreased Accuracy
- MiniKraken
  - Reduced database
    - Needs 4 GB of RAM
  - Less overall Sensitivity (~11-25%)
- Kraken-GB
  - Uses GenBank Database
    - Very high sensitivity



KRAKEN2 TUTORIAL

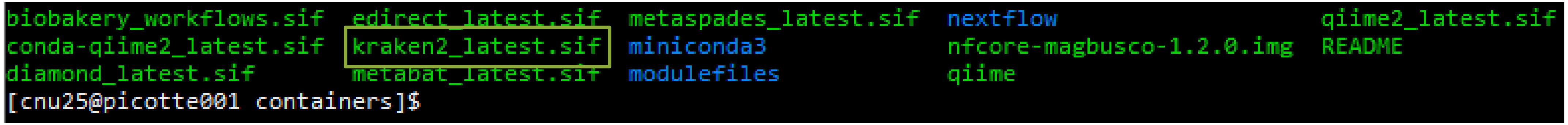
INSTALLATION & DOWNLOADS

- Download Kraken2 Docker:



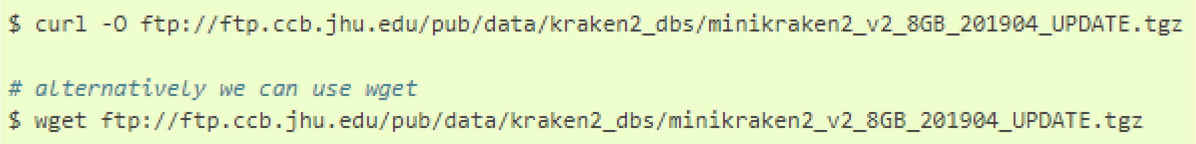
```
$ conda create --yes -n kraken kraken2 bracken
$ conda activate kraken
```

- (Or) find it on Picotte:



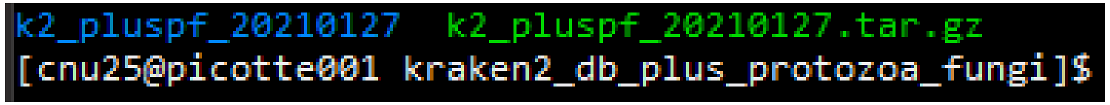
```
biobakery_workflows.sif  edirect_latest.sif  metaspades_latest.sif  nextflow                   qiime2_latest.sif
conda-qiime2_latest.sif  kraken2_latest.sif  miniconda3             nfcore-magbusco-1.2.0.img  README
diamond_latest.sif       metabat_latest.sif  modulefiles            qiime
[cnu25@picotte001 containers]$
```

- Must also download taxo database:



```
$ curl -O ftp://ftp.ccb.jhu.edu/pub/data/kraken2_dbs/minikraken2_v2_8GB_201904_UPDATE.tgz

# alternatively we can use wget
$ wget ftp://ftp.ccb.jhu.edu/pub/data/kraken2_dbs/minikraken2_v2_8GB_201904_UPDATE.tgz
```

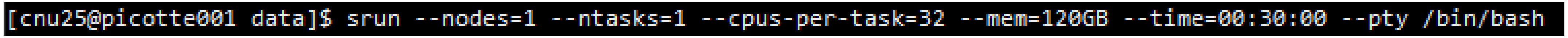
- Also available on picotte



```
k2_pluspf_20210127  k2_pluspf_20210127.tar.gz
[cnu25@picotte001 kraken2_db_plus_protozoa_fungi]$
```

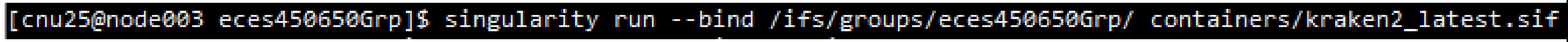
RUN KRAKEN2 CONTAINER IN SINGULARITY

- Request node to run on:



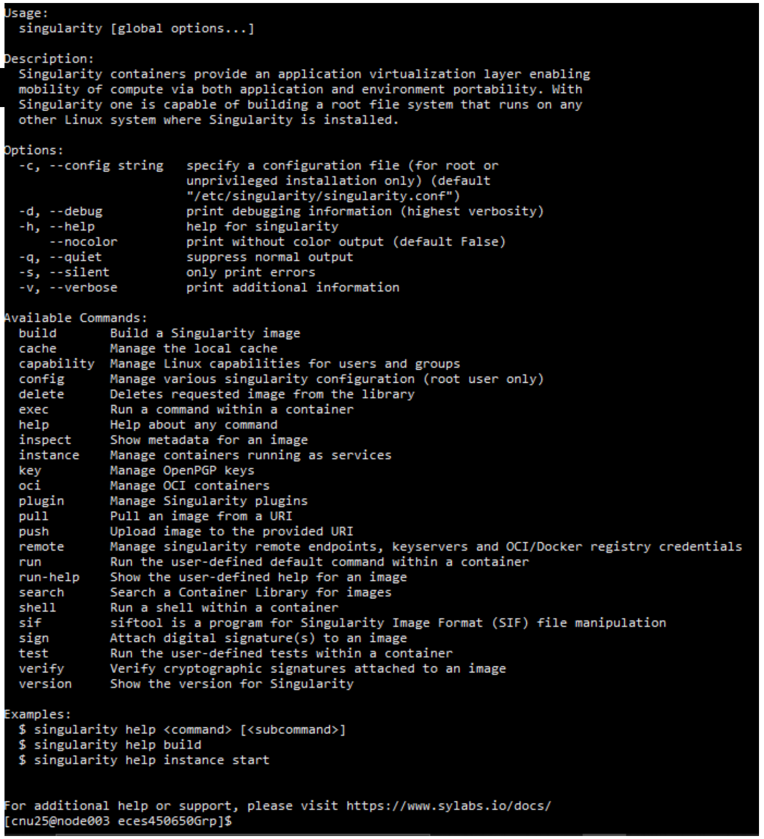
```
[cnu25@picotte001 data]$ srun --nodes=1 --ntasks=1 --cpus-per-task=32 --mem=120GB --time=00:30:00 --pty /bin/bash
```

- Run the docker using singularity, bind to our groups folder



```
[cnu25@node003 eces450650Grp]$ singularity run --bind /ifs/groups/eces450650Grp/ containers/kraken2_latest.sif
```

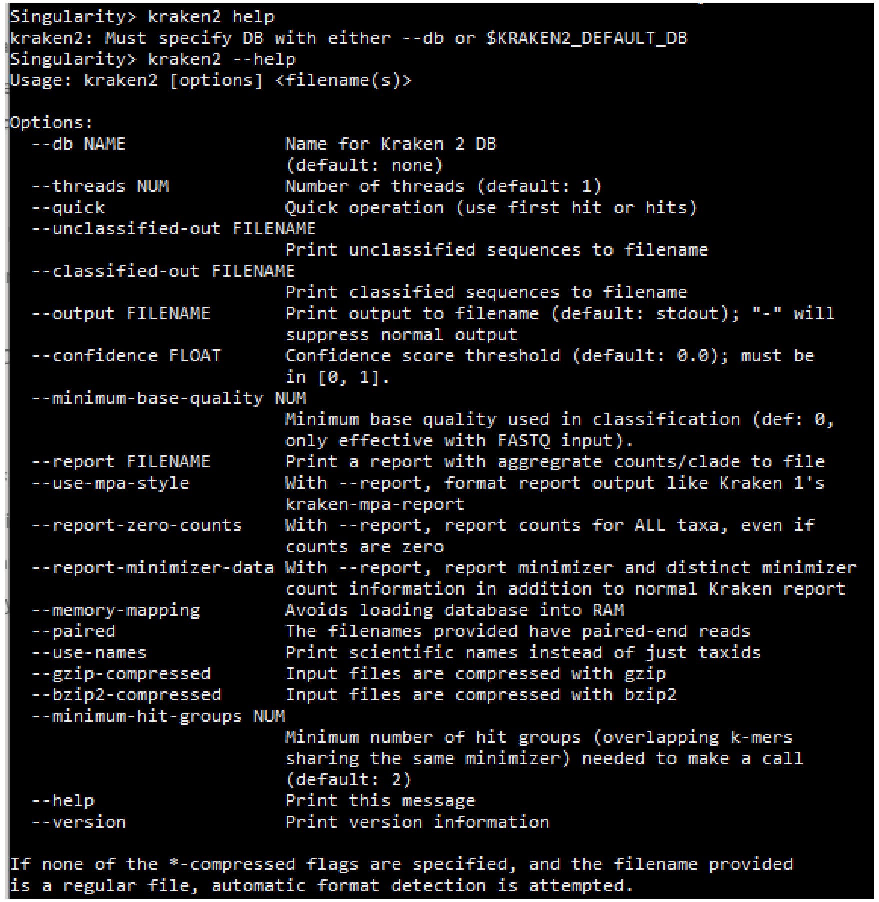
- We are using the run command to execute through the kraken container



```
Usage:
  singularity [global options...]

Description:
  Singularity containers provide an application virtualization layer enabling
  mobility of compute via both application and environment portability. With
  Singularity one is capable of building a root file system that runs on any
  other Linux system where Singularity is installed.

Options:
  -c, --config string   specify a configuration file (for root or
                        unprivileged installation only) (default
                        "/etc/singularity/singularity.conf")
  -d, --debug           print debugging information (highest verbosity)
  -h, --help            help for singularity
      --nocolor         print without color output (default False)
  -q, --quiet           suppress normal output
  -s, --silent          only print errors
  -v, --verbose         print additional information

Available Commands:
  build       Build a Singularity image
  cache       Manage the local cache
  capability  Manage Linux capabilities for users and groups
  config      Manage various singularity configuration (root user only)
  delete      Deletes requested image from the library
  exec        Run a command within a container
  help        Help about any command
  inspect     Show metadata for an image
  instance    Manage containers running as services
  key         Manage OpenPGP keys
  oci         Manage OCI containers
  plugin      Manage Singularity plugins
  pull        Pull an image from a URI
  push        Upload image to the provided URI
  remote      Manage singularity remote endpoints, keyservers and OCI/Docker registry credentials
  run         Run the user-defined default command within a container
  run-help    Show the user-defined help for an image
  search      Search a Container Library for images
  shell       Run a shell within a container
  sif         siftool is a program for Singularity Image Format (SIF) file manipulation
  sign        Attach digital signature(s) to an image
  test        Run the user-defined tests within a container
  verify      Verify cryptographic signatures attached to an image
  version     Show the version for Singularity

Examples:
  $ singularity help <command> [<subcommand>]
  $ singularity help build
  $ singularity help instance start


For additional help or support, please visit https://www.sylabs.io/docs/
[cnu25@node003 eces450650Grp]$
```

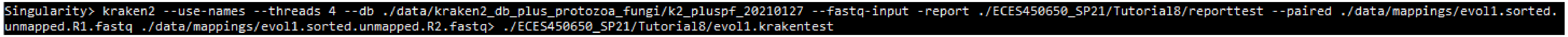
USAGE NOTES (KRAKEN)

`--use-names` : print scientific names
`--db` : Database reference file path
`--fastq-input` [deprecated] : We are using fastq formatted files
`--paired` : We are dealing with paired end data
`--report FILE` : This provides us with a sample-wide report



```
Singularity> kraken2 help
kraken2: Must specify DB with either --db or $KRAKEN2_DEFAULT_DB
Singularity> kraken2 --help
Usage: kraken2 [options] <filename(s)>

Options:
  --db NAME               Name for Kraken 2 DB
                          (default: none)
  --threads NUM           Number of threads (default: 1)
  --quick                 Quick operation (use first hit or hits)
  --unclassified-out FILENAME
                          Print unclassified sequences to filename
  --classified-out FILENAME
                          Print classified sequences to filename
  --output FILENAME       Print output to filename (default: stdout); "-" will
                          suppress normal output
  --confidence FLOAT      Confidence score threshold (default: 0.0); must be
                          in [0, 1].
  --minimum-base-quality NUM
                          Minimum base quality used in classification (def: 0,
                          only effective with FASTQ input).
  --report FILENAME       Print a report with aggregrate counts/clade to file
  --use-mpa-style         With --report, format report output like Kraken 1's
                          kraken-mpa-report
  --report-zero-counts    With --report, report counts for ALL taxa, even if
                          counts are zero
  --report-minimizer-data With --report, report minimizer and distinct minimizer
                          count information in addition to normal Kraken report
  --memory-mapping        Avoids loading database into RAM
  --paired                The filenames provided have paired-end reads
  --use-names             Print scientific names instead of just taxids
  --gzip-compressed       Input files are compressed with gzip
  --bzip2-compressed      Input files are compressed with bzip2
  --minimum-hit-groups NUM
                          Minimum number of hit groups (overlapping k-mers
                          sharing the same minimizer) needed to make a call
                          (default: 2)
  --help                  Print this message
  --version               Print version information

If none of the *-compressed flags are specified, and the filename provided
is a regular file, automatic format detection is attempted.
```

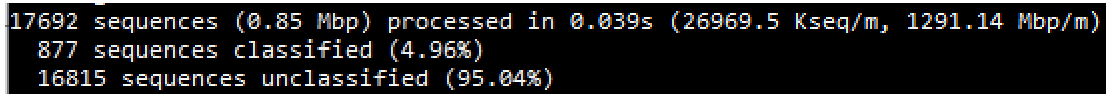
KRAKEN2 COMMAND, I/O, RUNTIME



```
Singularity> kraken2 --use-names --threads 4 --db ./data/kraken2_db_plus_protozoa_fungi/k2_pluspf_20210127 --fastq-input -report ./ECES450650_SP21/Tutorial8/reporttest --paired ./data/mappings/evol1.sorted.
unmapped.R1.fastq ./data/mappings/evol1.sorted.unmapped.R2.fastq> ./ECES450650_SP21/Tutorial8/evol1.krakentest
```

- Input files: /ifs/groups/eces450650Grp/data/mappings/evol1.sorted.unmapped.R1.fastq
  /ifs/groups/eces450650Grp/data/mappings/evol1.sorted.unmapped.R1.fastq
- Output files: ./ECES450650_SP21/Tutorial8/evol1.kraken ← classifications
  ./ECES450650_SP21/Tutorial8/report ← report

Execution time:



```
17692 sequences (0.85 Mbp) processed in 0.039s (26969.5 Kseq/m, 1291.14 Mbp/m)
  877 sequences classified (4.96%)
  16815 sequences unclassified (95.04%)
```

RESULTS & UNDERSTANDING - CLASSIFICATIONS

1.C/U: one letter code indicating that the sequence was either classified or unclassified.

2.The sequence ID, obtained from the FASTA/FASTQ header.

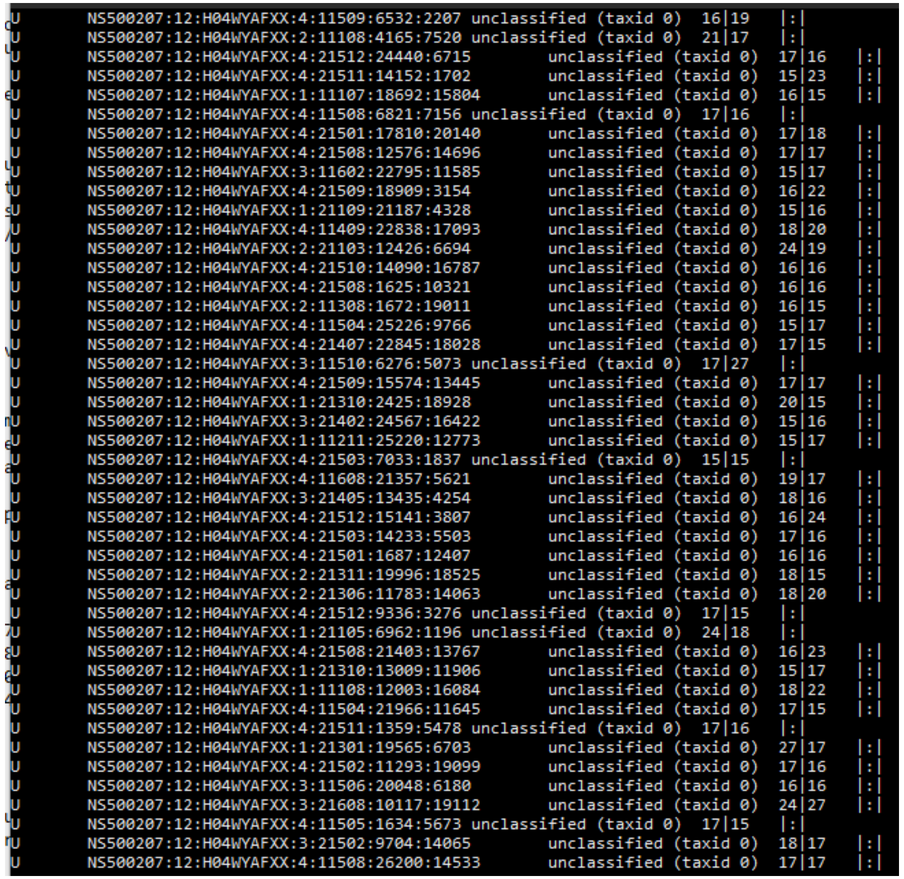
3.The taxonomy ID Kraken2 used to label the sequence; this is **0** if the sequence is unclassified and otherwise should be the NCBI Taxonomy identifier.

4.The length of the sequence in bp.

5.A space-delimited list indicating the lowest common ancestor (in the taxonomic tree) mapping of each k-mer in the sequence. For example, 562:13 561:4 A:31 0:1 562:3 would indicate that:

- the first 13 k-mers mapped to taxonomy ID #562
- the next 4 k-mers mapped to taxonomy ID #561
- the next 31 k-mers contained an ambiguous nucleotide
- the next k-mer was not in the database
- the last 3 k-mers mapped to taxonomy ID #562

Source: https://genomics.readthedocs.io/en/latest/ngs-taxonomic-investigation/index.html#wood2014

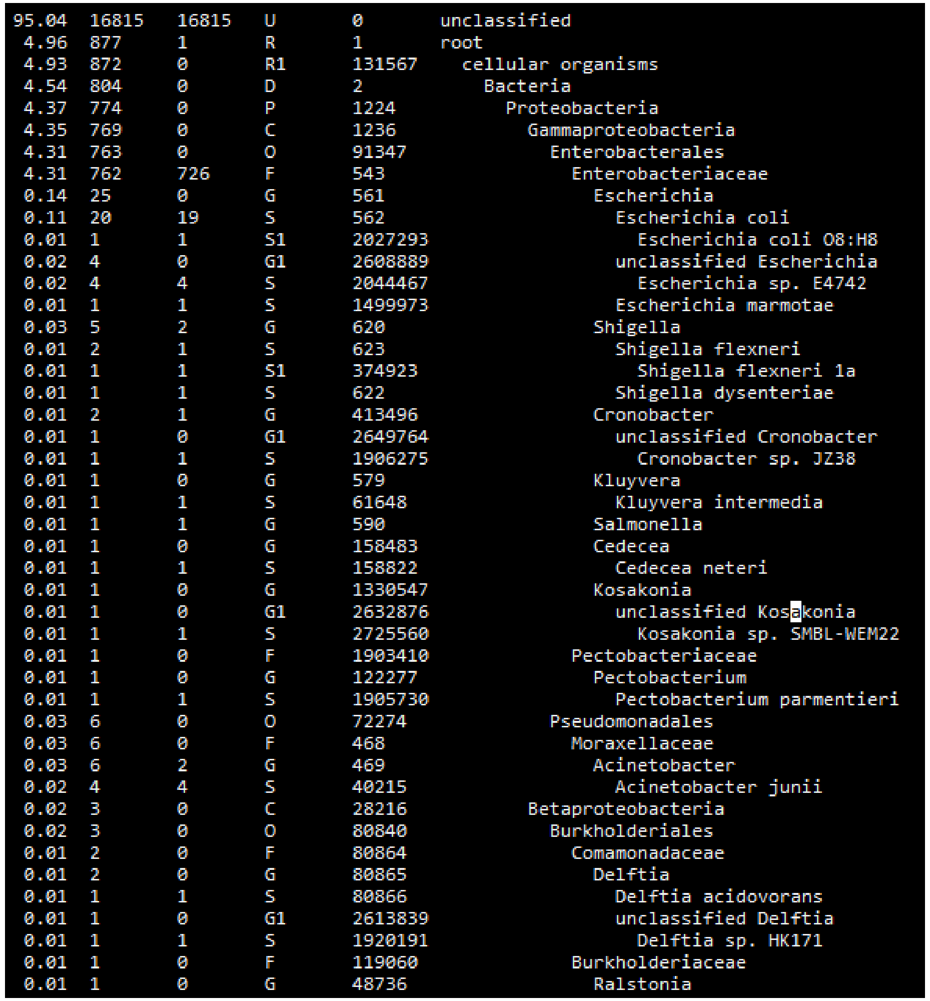


```
U       NS500207:12:H04WYAFXX:4:11509:6532:2207 unclassified (taxid 0)  16|19   |:|
U       NS500207:12:H04WYAFXX:2:11108:4165:7520 unclassified (taxid 0)  21|17   |:|
U       NS500207:12:H04WYAFXX:4:21512:24440:6715        unclassified (taxid 0)  17|16   |:|
U       NS500207:12:H04WYAFXX:4:21511:14152:1702        unclassified (taxid 0)  15|23   |:|
U       NS500207:12:H04WYAFXX:1:11107:18692:15804       unclassified (taxid 0)  16|15   |:|
U       NS500207:12:H04WYAFXX:4:11508:6821:7156 unclassified (taxid 0)  17|16   |:|
U       NS500207:12:H04WYAFXX:4:21501:17810:20140       unclassified (taxid 0)  17|18   |:|
U       NS500207:12:H04WYAFXX:4:21508:12576:14696       unclassified (taxid 0)  17|17   |:|
U       NS500207:12:H04WYAFXX:3:11602:22795:11585       unclassified (taxid 0)  15|17   |:|
U       NS500207:12:H04WYAFXX:4:21509:18909:3154        unclassified (taxid 0)  16|22   |:|
U       NS500207:12:H04WYAFXX:1:21109:21187:4328        unclassified (taxid 0)  15|16   |:|
U       NS500207:12:H04WYAFXX:4:11409:22838:17093       unclassified (taxid 0)  18|20   |:|
U       NS500207:12:H04WYAFXX:2:21103:12426:6694        unclassified (taxid 0)  24|19   |:|
U       NS500207:12:H04WYAFXX:4:21510:14090:16787       unclassified (taxid 0)  16|16   |:|
U       NS500207:12:H04WYAFXX:4:21508:1625:10321        unclassified (taxid 0)  16|16   |:|
U       NS500207:12:H04WYAFXX:2:11308:1672:19011        unclassified (taxid 0)  16|15   |:|
U       NS500207:12:H04WYAFXX:4:11504:25226:9766        unclassified (taxid 0)  15|17   |:|
U       NS500207:12:H04WYAFXX:4:21407:22845:18028       unclassified (taxid 0)  17|15   |:|
U       NS500207:12:H04WYAFXX:3:11510:6276:5073 unclassified (taxid 0)  17|27   |:|
U       NS500207:12:H04WYAFXX:4:21509:15574:13445       unclassified (taxid 0)  17|17   |:|
U       NS500207:12:H04WYAFXX:1:21310:2425:18928        unclassified (taxid 0)  20|15   |:|
U       NS500207:12:H04WYAFXX:3:21402:24567:16422       unclassified (taxid 0)  15|16   |:|
U       NS500207:12:H04WYAFXX:1:11211:25220:12773       unclassified (taxid 0)  15|17   |:|
U       NS500207:12:H04WYAFXX:4:21503:7033:1837 unclassified (taxid 0)  15|15   |:|
U       NS500207:12:H04WYAFXX:4:11608:21357:5621        unclassified (taxid 0)  19|17   |:|
U       NS500207:12:H04WYAFXX:3:21405:13435:4254        unclassified (taxid 0)  18|16   |:|
U       NS500207:12:H04WYAFXX:4:21512:15141:3807        unclassified (taxid 0)  16|24   |:|
U       NS500207:12:H04WYAFXX:4:21503:14233:5503        unclassified (taxid 0)  17|16   |:|
U       NS500207:12:H04WYAFXX:4:21501:1687:12407        unclassified (taxid 0)  16|16   |:|
U       NS500207:12:H04WYAFXX:2:21311:19996:18525       unclassified (taxid 0)  18|15   |:|
U       NS500207:12:H04WYAFXX:2:21306:11783:14063       unclassified (taxid 0)  18|20   |:|
U       NS500207:12:H04WYAFXX:4:21512:9336:3276 unclassified (taxid 0)  17|15   |:|
U       NS500207:12:H04WYAFXX:1:21105:6962:1196 unclassified (taxid 0)  24|18   |:|
U       NS500207:12:H04WYAFXX:4:21508:21403:13767       unclassified (taxid 0)  16|23   |:|
U       NS500207:12:H04WYAFXX:1:21310:13009:11906       unclassified (taxid 0)  15|17   |:|
U       NS500207:12:H04WYAFXX:1:11108:12003:16084       unclassified (taxid 0)  18|22   |:|
U       NS500207:12:H04WYAFXX:4:11504:21966:11645       unclassified (taxid 0)  17|15   |:|
U       NS500207:12:H04WYAFXX:4:21511:1359:5478 unclassified (taxid 0)  17|16   |:|
U       NS500207:12:H04WYAFXX:1:21301:19565:6703        unclassified (taxid 0)  27|17   |:|
U       NS500207:12:H04WYAFXX:4:21502:11293:19099       unclassified (taxid 0)  17|16   |:|
U       NS500207:12:H04WYAFXX:3:11506:20048:6180        unclassified (taxid 0)  16|16   |:|
U       NS500207:12:H04WYAFXX:3:21608:10117:19112       unclassified (taxid 0)  24|27   |:|
U       NS500207:12:H04WYAFXX:4:11505:1634:5673 unclassified (taxid 0)  17|15   |:|
U       NS500207:12:H04WYAFXX:3:21502:9704:14065        unclassified (taxid 0)  18|17   |:|
U       NS500207:12:H04WYAFXX:4:11508:26200:14533       unclassified (taxid 0)  17|17   |:|
```

RESULTS & UNDERSTANDING – REPORT SUMMARY

1. **Percentage** of reads covered by the clade rooted at this taxon
2. **Number of reads** covered by the clade rooted at this taxon
3. **Number of reads** assigned directly to this taxon
4. A rank code, indicating **(U)nclassified, (D)omain, (K)ingdom, (P)hylum, (C)lass, (O)rder, (F)amily, (G)enus, or (S)pecies**. All other ranks are simply **"-"**.
5. NCBI Taxonomy ID
6. The indented scientific name

Source: https://genomics.readthedocs.io/en/latest/ngs-taxonomic-investigation/index.html#wood2014



```
95.04  16815   16815   U       0       unclassified
 4.96  877     1       R       1       root
 4.93  872     0       R1      131567    cellular organisms
 4.54  804     0       D       2           Bacteria
 4.37  774     0       P       1224          Proteobacteria
 4.35  769     0       C       1236            Gammaproteobacteria
 4.31  763     0       O       91347             Enterobacterales
 4.31  762     726     F       543                 Enterobacteriaceae
 0.14  25      0       G       561                   Escherichia
 0.11  20      19      S       562                     Escherichia coli
 0.01  1       1       S1      2027293                   Escherichia coli O8:H8
 0.02  4       0       G1      2608889                 unclassified Escherichia
 0.02  4       4       S       2044467                   Escherichia sp. E4742
 0.01  1       1       S       1499973                 Escherichia marmotae
 0.03  5       2       G       620                   Shigella
 0.01  2       1       S       623                     Shigella flexneri
 0.01  1       1       S1      374923                    Shigella flexneri 1a
 0.01  1       1       S       622                     Shigella dysenteriae
 0.01  2       1       G       413496                Cronobacter
 0.01  1       0       G1      2649764                 unclassified Cronobacter
 0.01  1       1       S       1906275                   Cronobacter sp. JZ38
 0.01  1       0       G       579                   Kluyvera
 0.01  1       1       S       61648                   Kluyvera intermedia
 0.01  1       1       G       590                   Salmonella
 0.01  1       0       G       158483                Cedecea
 0.01  1       1       S       158822                  Cedecea neteri
 0.01  1       0       G       1330547               Kosakonia
 0.01  1       0       G1      2632876                 unclassified Kosakonia
 0.01  1       1       S       2725560                   Kosakonia sp. SMBL-WEM22
 0.01  1       0       F       1903410             Pectobacteriaceae
 0.01  1       0       G       122277                Pectobacterium
 0.01  1       1       S       1905730                 Pectobacterium parmentieri
 0.03  6       0       O       72274             Pseudomonadales
 0.03  6       0       F       468                 Moraxellaceae
 0.03  6       2       G       469                   Acinetobacter
 0.02  4       4       S       40215                   Acinetobacter junii
 0.02  3       0       C       28216           Betaproteobacteria
 0.02  3       0       O       80840             Burkholderiales
 0.01  2       0       F       80864               Comamonadaceae
 0.01  2       0       G       80865                 Delftia
 0.01  1       1       S       80866                   Delftia acidovorans
 0.01  1       0       G1      2613839                 unclassified Delftia
 0.01  1       1       S       1920191                   Delftia sp. HK171
 0.01  1       0       F       119060              Burkholderiaceae
 0.01  1       0       G       48736                 Ralstonia
```

SOURCES

- “6. Taxonomic investigation¶,” *6. Taxonomic investigation - Genomics Tutorial 2020.2.0 documentation*. [Online]. Available: https://genomics.readthedocs.io/en/latest/ngs-taxonomic-investigation/index.html#kraken2. [Accessed: 20-May-2021].
- D. E. Wood and S. L. Salzberg, “Kraken: ultrafast metagenomic sequence classification using exact alignments,” *Genome Biology*, 03-Mar-2014. [Online]. Available: https://genomebiology.biomedcentral.com/articles/10.1186/gb-2014-15-3-r46. [Accessed: 20-May-2021].
- D. Wood, “DerrickWood/kraken2,” *GitHub*. [Online]. Available: https://github.com/DerrickWood/kraken2. [Accessed: 20-May-2021].
- M. Lee, “Genomics,” *Happy Belly Bioinformatics*, 2021. [Online]. Available: https://astrobiomike.github.io/genomics/. [Accessed: 20-May-2021].