Genome Skimming Workshop:

Using Low Coverage WGS to Estimate Distances and Sequencing Parameters.

by Eduardo Charvel and Siavash Mirarab

UCSD

ALFRED P. SLOAN FOUNDATION

MINDEROO

FIELD MUSEUM Grainger Bioinformatics Center

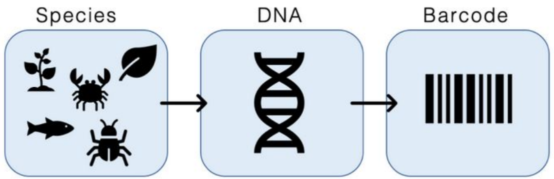
Cloning GitHub and Installing Tools:

```
git clone https://github.com/echarvel3/skimming_scripts-echarvel.git
```

```
cd skimming_scripts-echarvel
bash ./install.sh
```

Intro: Evolutionary Ecology Approaches

Genetic Markers



marker genes, mtDNA, microsatellites

PROS:

- cost-effective (single genes)
- no recombination (mtDNA)

CONS:

- expensive (larger bait sets)
- limited resolution (single genes)
- reference dependency
- homoplasy

SNP based

RAD-seq, ANGSD, GATK

PROS:

- genome-wide resolution (WGS)

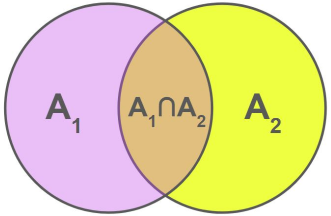
CONS:

- reference-based (WGS)
- higher coverage (>10x) (WGS)
- restriction-enzymes limit resolution (RAD-seq)

SNP

Genotype	1	2	3	4	5	6
1	0	1	1	1	0	1
2	0	1	0	0	0	0
3	1	0	0	1	1	1

k-mer based



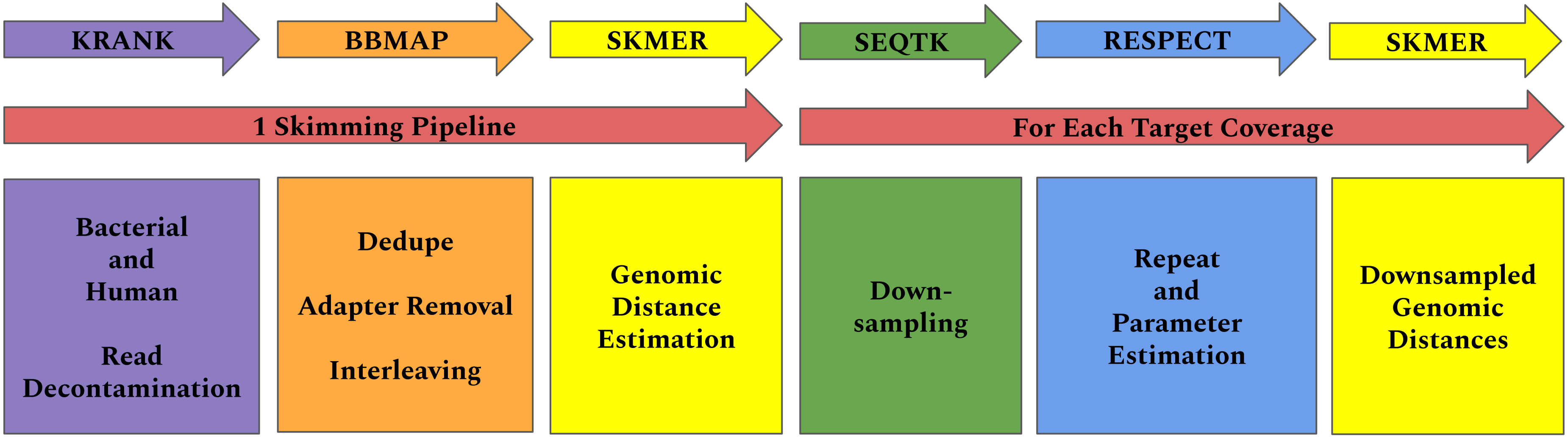
PROS:

- genome-wide resolution
- low coverage (< 1x .. 8x)
- alignment-free

CONS:

- sensitive to sequencing quality:
  - contamination *(addressed in this workshop)*
  - repetitiveness *(active area of research)*
  - hybridization

Skimming Pipeline Overview



KRANK

BBMAP

SKMER

SEQTK

RESPECT

SKMER

1 Skimming Pipeline

For Each Target Coverage

Bacterial and Human Read Decontamination

Dedupe

Adapter Removal

Interleaving

Genomic Distance Estimation

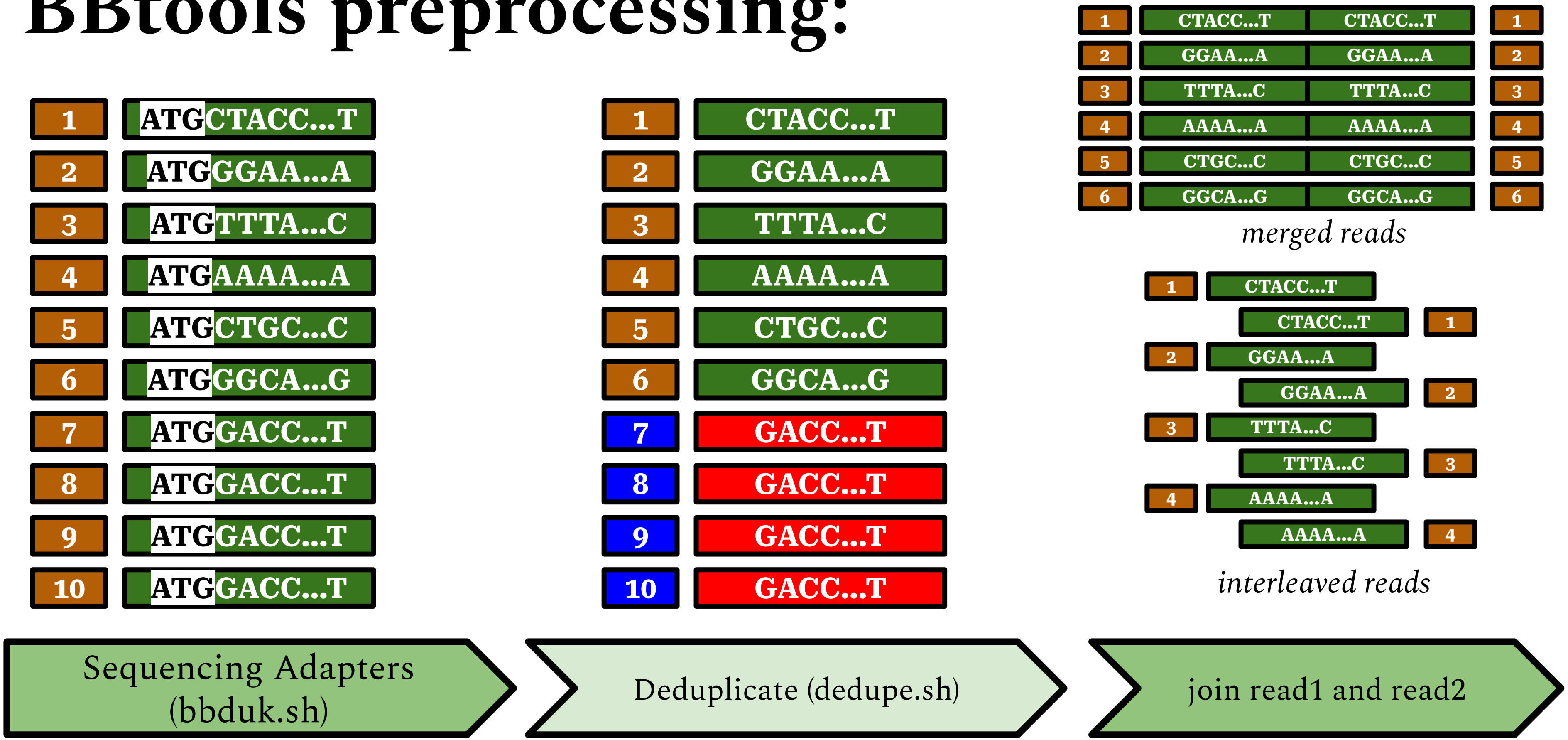
Down-sampling

Repeat and Parameter Estimation

Downsampled Genomic Distances

- Meant to easily perform all operations necessary for accurate genomic distance and sequencing parameter estimation.

BBtools preprocessing:



1	ATGCTACC...T
2	ATGGGAA...A
3	ATGTTTA...C
4	ATGAAAA...A
5	ATGCTGC...C
6	ATGGGCA...G
7	ATGGACC...T
8	ATGGACC...T
9	ATGGACC...T
10	ATGGACC...T

1	CTACC...T
2	GGAA...A
3	TTTA...C
4	AAAA...A
5	CTGC...C
6	GGCA...G
7	GACC...T
8	GACC...T
9	GACC...T
10	GACC...T

1	CTACC...T	CTACC...T	1
2	GGAA...A	GGAA...A	2
3	TTTA...C	TTTA...C	3
4	AAAA...A	AAAA...A	4
5	CTGC...C	CTGC...C	5
6	GGCA...G	GGCA...G	6

merged reads

1	CTACC...T	
	CTACC...T	1
2	GGAA...A	
	GGAA...A	2
3	TTTA...C	
	TTTA...C	3
4	AAAA...A	
	AAAA...A	4

interleaved reads

Sequencing Adapters (bbduk.sh)

Deduplicate (dedupe.sh)

join read1 and read2

Running BBMAP

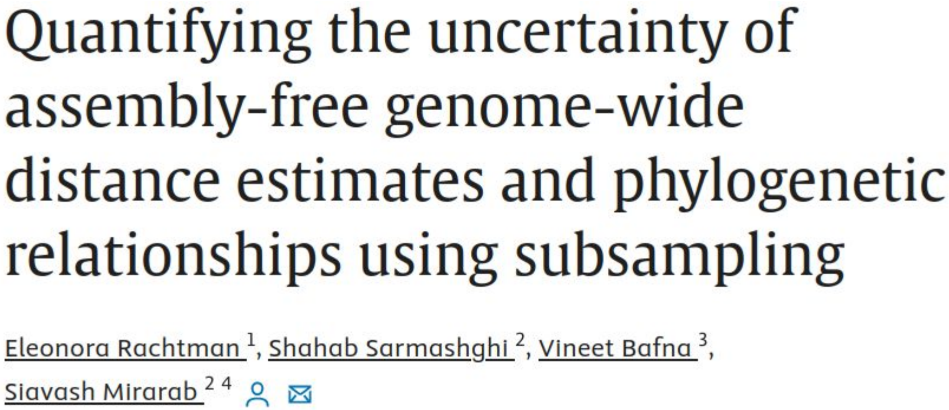
```
cd skimming_scripts-echarvel
conda activate ${CONDA_ENV} #replace with env name
cd ./test/
gunzip ./skims/*gz
mkdir bbmap_reads
bash ../bbmap_pipeline.sh \
       ./skims/read1.fq \
       ./skims/read2.fq \
       ./bbmap_reads/read_out.fq
```

Main Skmer Paper

Skmer: assembly-free and alignment-free sample identification using genome skims

Shahab Sarmashghi, Kristine Bohmann, M. Thomas P. Gilbert, Vineet Bafna & Siavash Mirarab

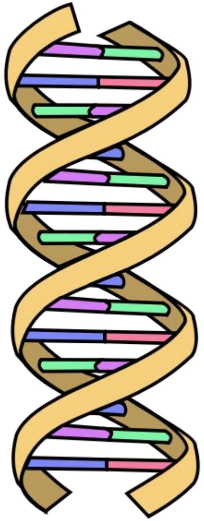
Skmer Bootstrapping Approach



Quantifying the uncertainty of assembly-free genome-wide distance estimates and phylogenetic relationships using subsampling

Eleonora Rachtman[1], Shahab Sarmashghi[2], Vineet Bafna[3], Siavash Mirarab[2 4]

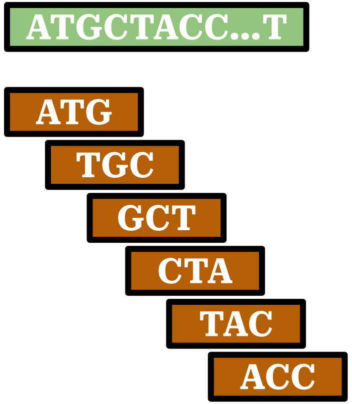
SKMER Theory:

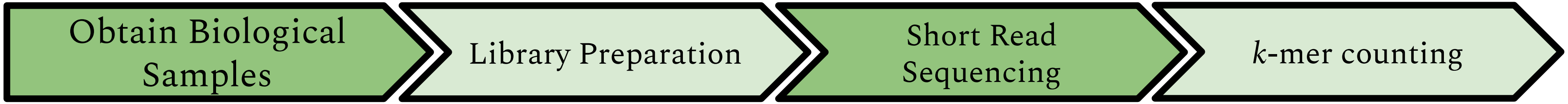


collection of
reads ~ 150 bp

1	ATGCTACC...T
2	CTAGGAA...A
3	CCCTTTA...C
4	GATAAAA...A
5	TATACTGC...C
6	GTCGGCA...G
7	GGACTGC...C
8	CTTATCC...G
9	TAAGTGG...C
10	CAGGACC...T

each can be decomposed
into a set of *k*-mers





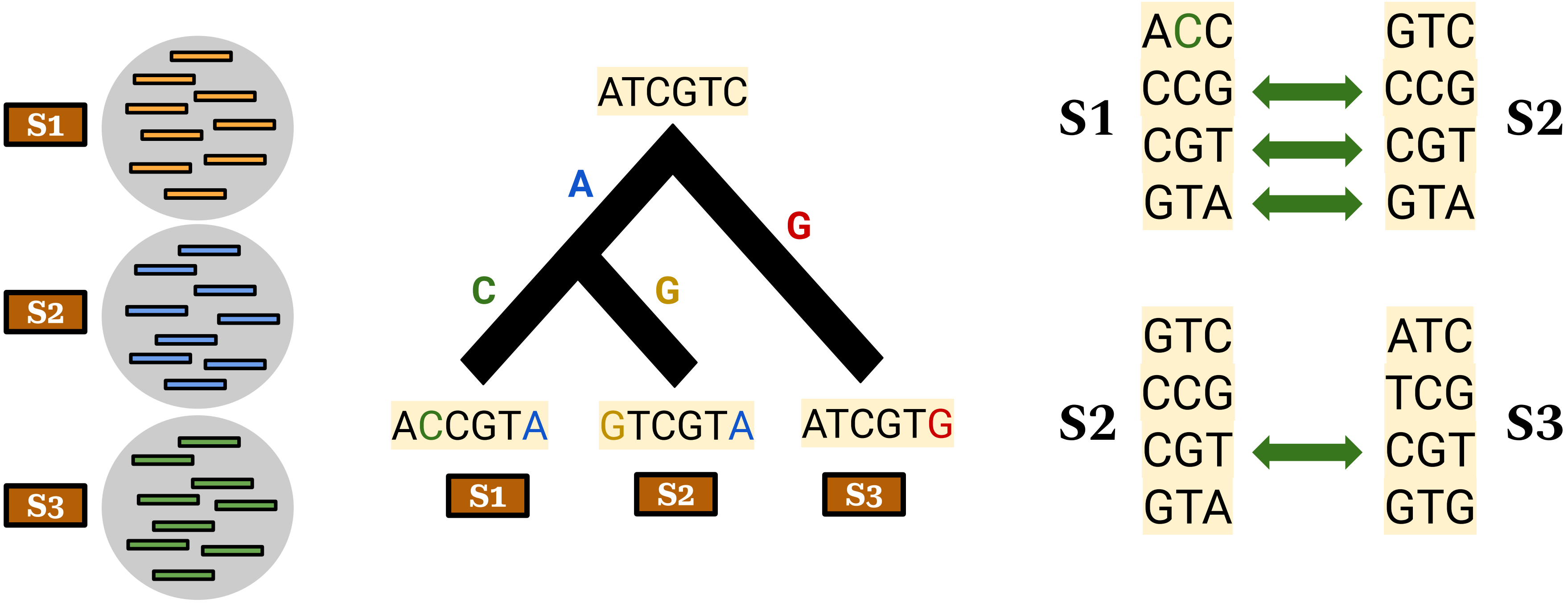
Obtain Biological Samples

Library Preparation

Short Read Sequencing

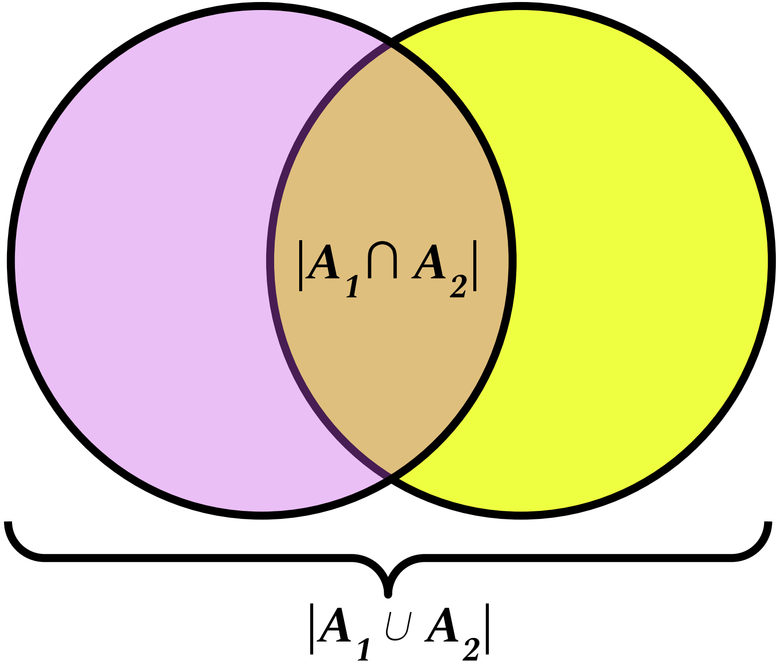
k-mer counting

SKMER Theory:



k-mer set for each sample → Mutations accumulate → Compare *k*-mer similarities

Mash (2016): *k*-mer Based Distances



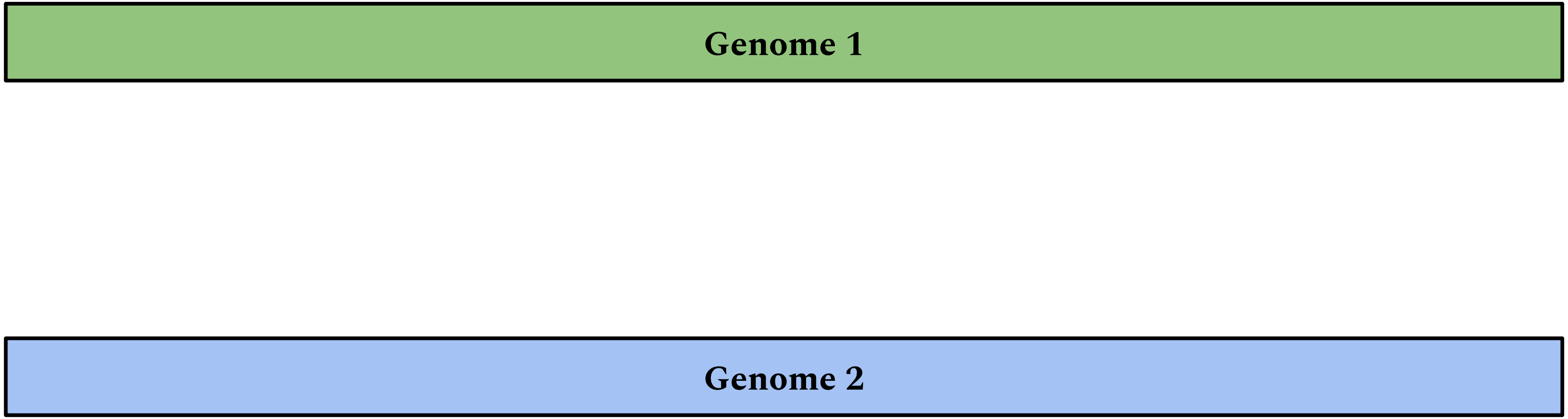
shared *k*-mers = intersection = $|A_1 \cap A_2|$

set of all *k*-mers = union = $|A_1 \cup A_2|$

Jaccard Similarity Index = $J = \dfrac{|A_1 \cap A_2|}{|A_1 \cup A_2|}$

$$\boxed{D = 1 - (\frac{2J}{1+J})^{1/k}}$$

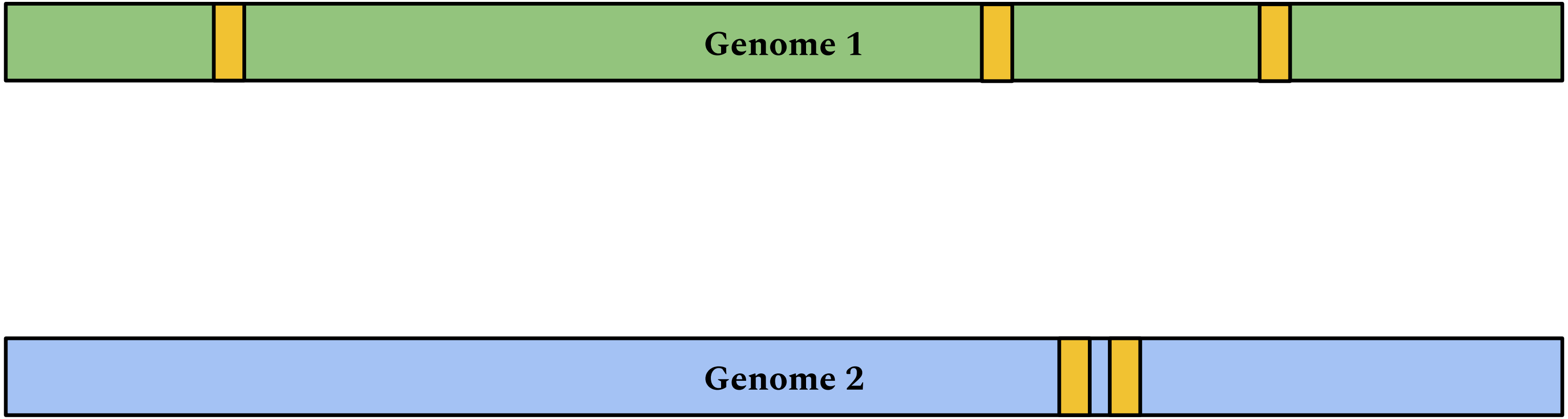
SKMER: Using Low Coverage Skims



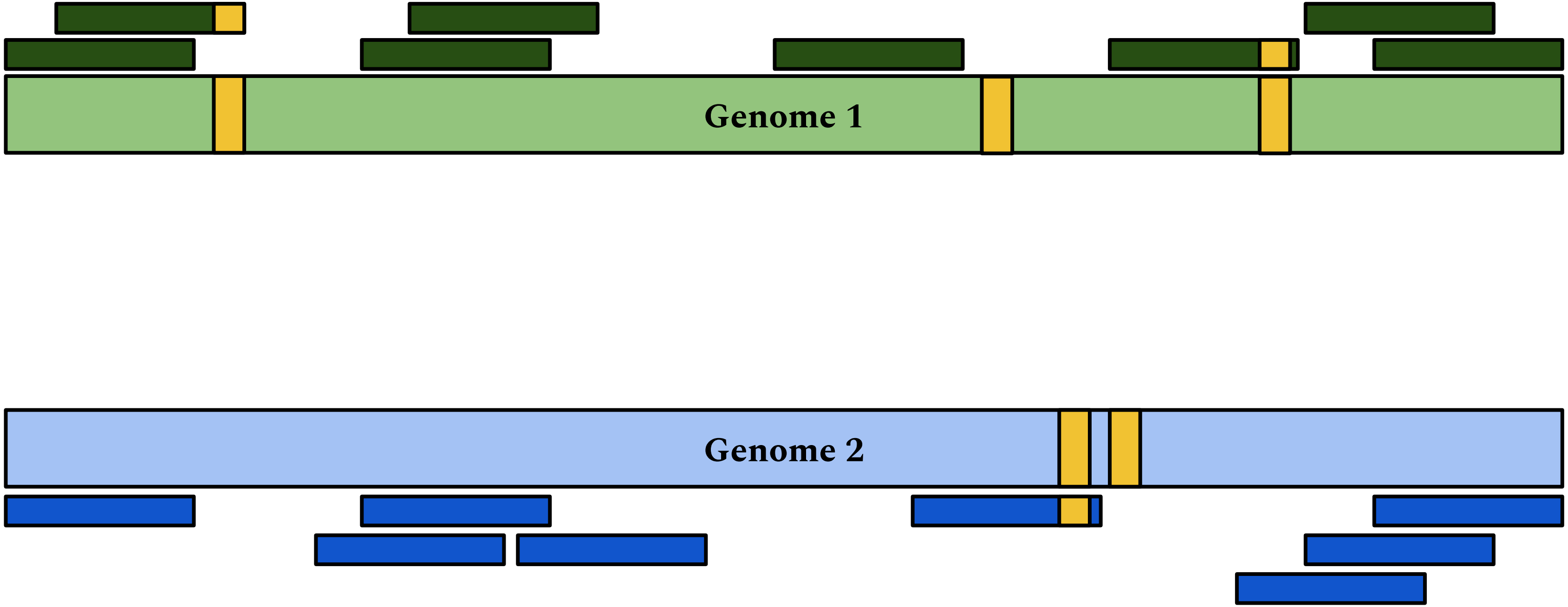
Genome 1

Genome 2

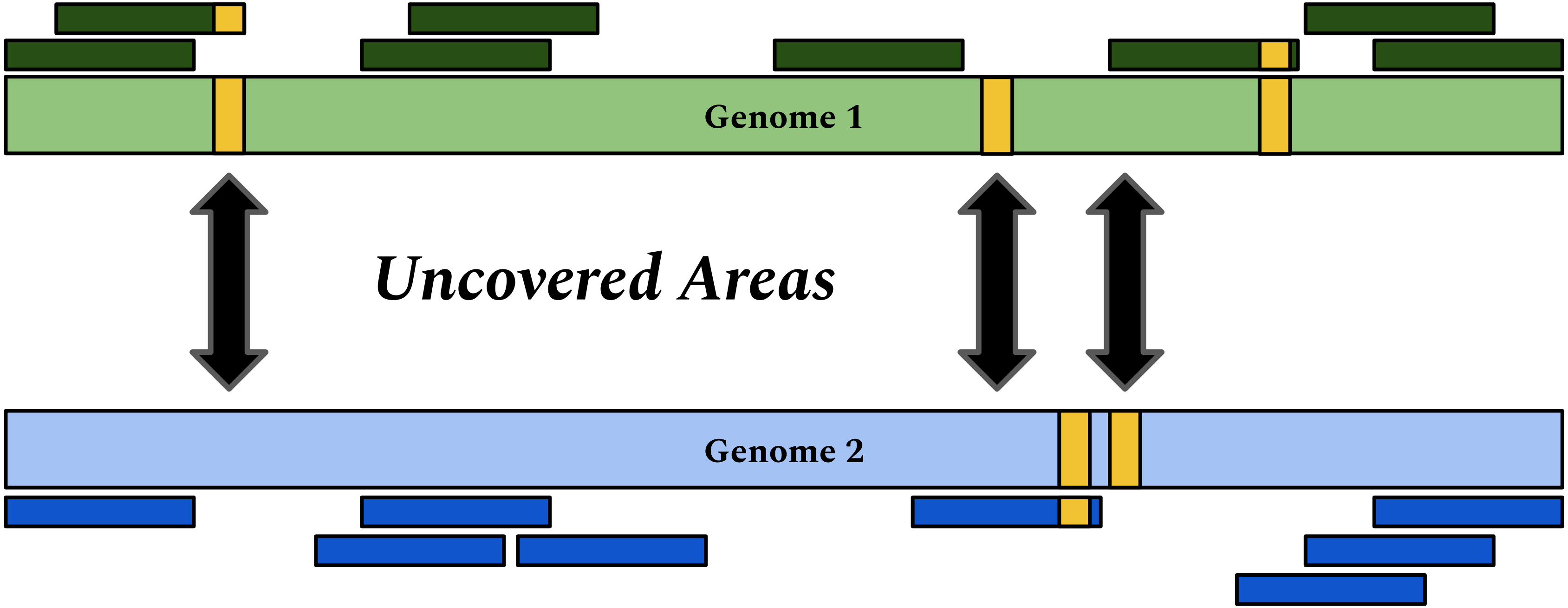
SKMER: Using Low Coverage Skims



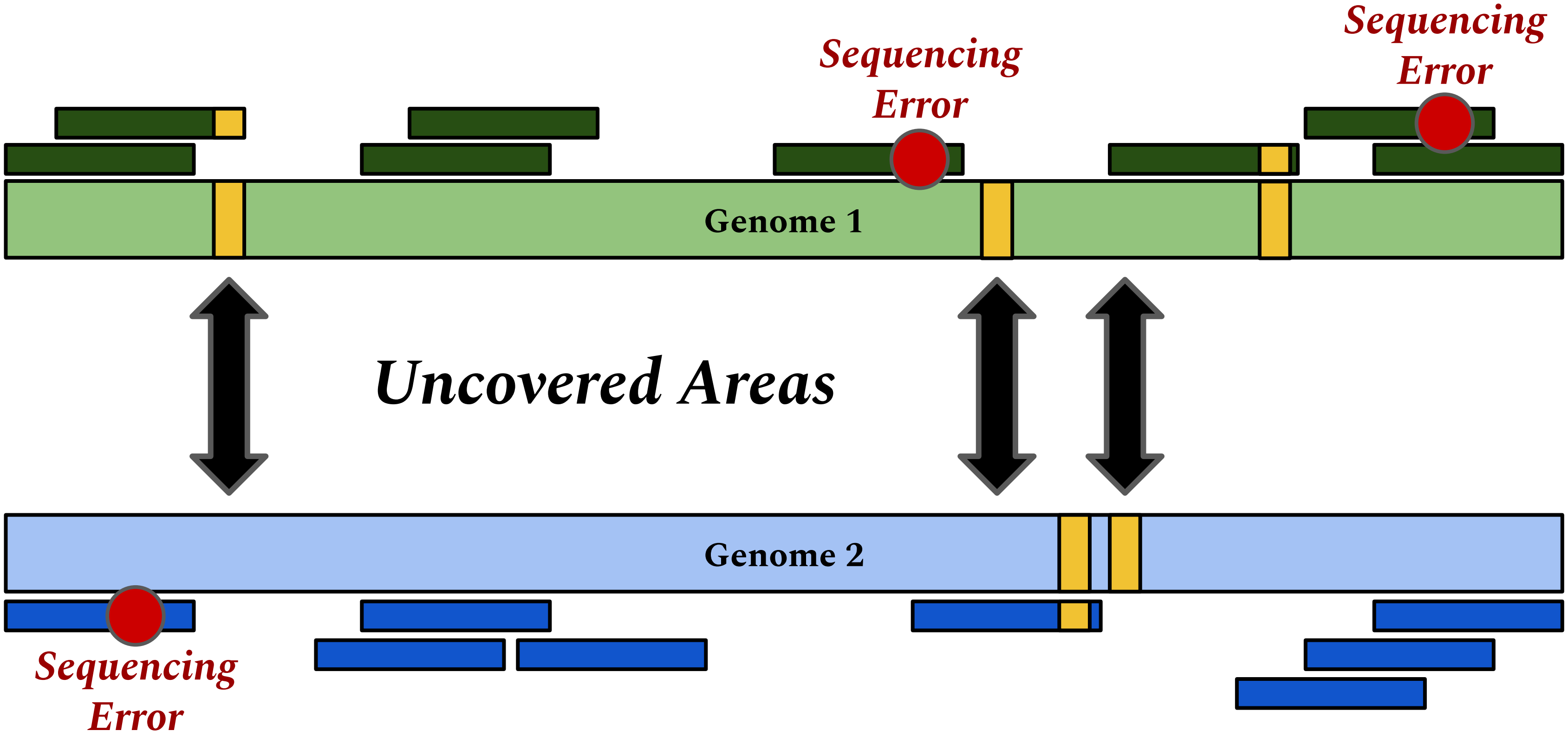
SKMER: Using Low Coverage Skims



SKMER: Using Low Coverage Skims



SKMER: Using Low Coverage Skims



SKMER: Using Low Coverage Skims

$$D = 1 - (\frac{2J}{1+J})^{1/k}$$

→

$$D = 1 - \left(\frac{2(\zeta_1 L_1 + \zeta_2 L_2)J}{\eta_1 \eta_2 (L_1 + L_2)(1+J)}\right)^{1/k}$$

η_i (eta) = probability a k-mer is covered at least once without error.

ζ_i (zeta) = the total number of k-mers observed from both genomes.

L_i = genome size.

→ Both **eta** and **zeta** are functions of **coverage** and **error**.

SKMER: Estimating Parameters

$\boldsymbol{\lambda}$ (lambda) = how many times a *k*-mer is covered

$\boldsymbol{\xi}$ (xi) = number of error free reads covering a *k*-mer

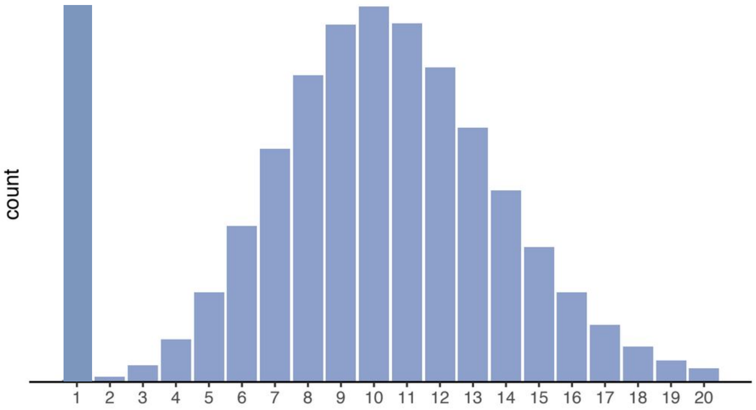
$\boldsymbol{\varepsilon}$ (epsilon) = sequencing error rate

$\mathbf{|A \cap B|}$ (obs. intersection) = shared *k*-mers between samples.

- Use *k*-mer frequency spectrum to estimate of parameters.

- Use *MASH* to compute observed intersection.

Estimating Error and Coverage:



$$P(X) = \frac{\lambda^x e^{-\lambda}}{X!}$$

WGS follows a Poisson Distribution

$$h = \operatorname{argmax}_{i \geq 2} M_i$$ = ***mode***

$$\xi = \frac{M_{h+1}}{M_h}(h+1)$$ = ***number of error-free reads covering a k-mer***

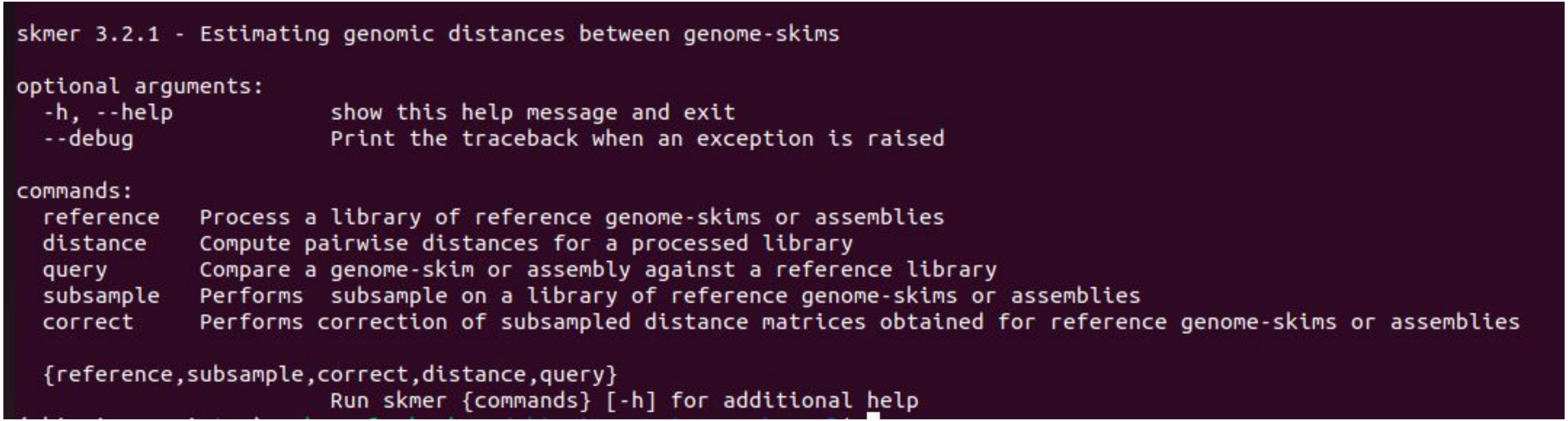
$$\lambda \quad = \quad \frac{M_1}{M_h}\frac{\xi^h}{h!}e^{-\xi} + \xi(1 - e^{-\xi})$$ = ***k-mer coverage***

$$\epsilon \quad = \quad 1 - (\xi/\lambda)^{1/k}$$ = ***sequencing error rate***

Running Skmer Reference

```
cd skimming_scripts-echarvel
conda activate ${CONDA_ENV} #replace with env name
gunzip ./test/skims/*gz
skmer reference ./skims/bbmap_reads -p1
```

Other Skmer Utilities:



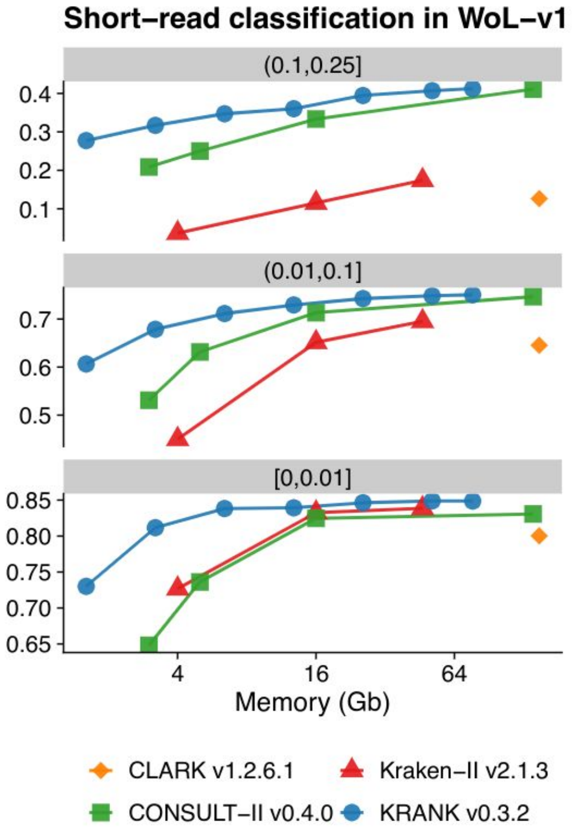
```
skmer 3.2.1 - Estimating genomic distances between genome-skims

optional arguments:
  -h, --help            show this help message and exit
  --debug               Print the traceback when an exception is raised

commands:
  reference   Process a library of reference genome-skims or assemblies
  distance    Compute pairwise distances for a processed library
  query       Compare a genome-skim or assembly against a reference library
  subsample   Performs  subsample on a library of reference genome-skims or assemblies
  correct     Performs correction of subsampled distance matrices obtained for reference genome-skims or assemblies

  {reference,subsample,correct,distance,query}
                        Run skmer {commands} [-h] for additional help
```

distance *== generate distance matrix (automatic for reference)*
query *== add new skim/assembly to library*
subsample *== perform bootstrap approach*

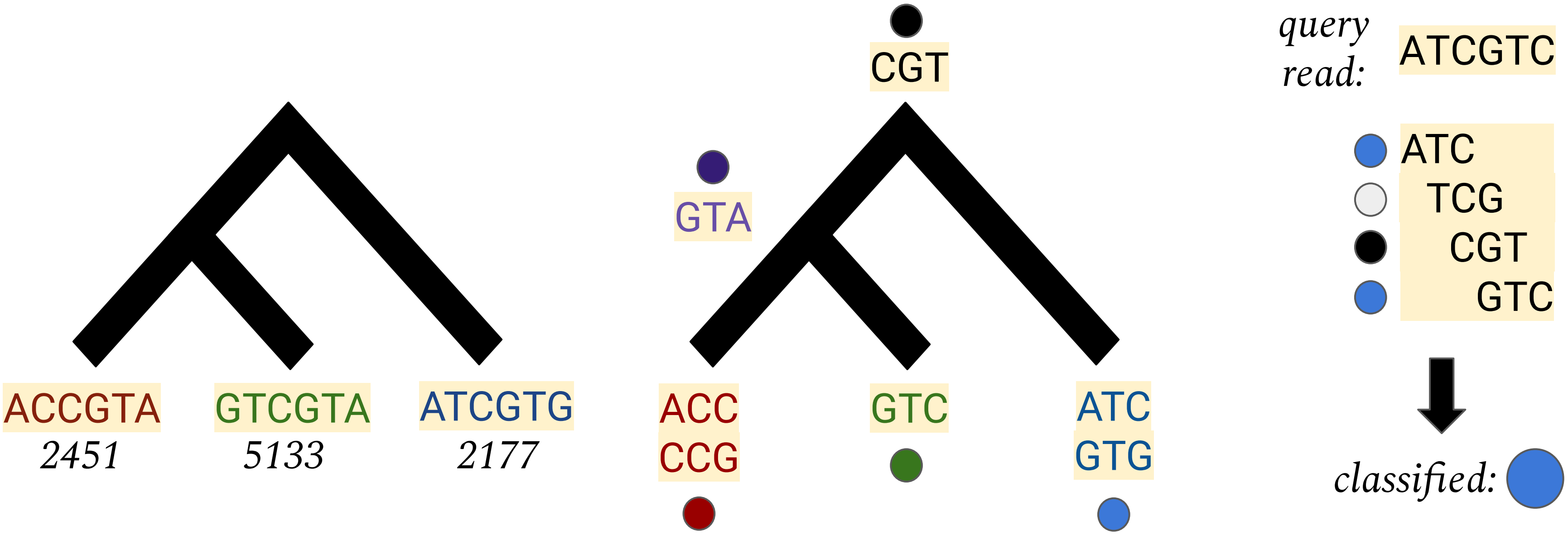
Memory-bound *k*-mer selection for large and evolutionary diverse reference libraries

Ali Osman Berk Şapcı, Siavash Mirarab

doi: https://doi.org/10.1101/2024.02.12.580015



KRANK: Decontamination



genomes and taxonomy IDs

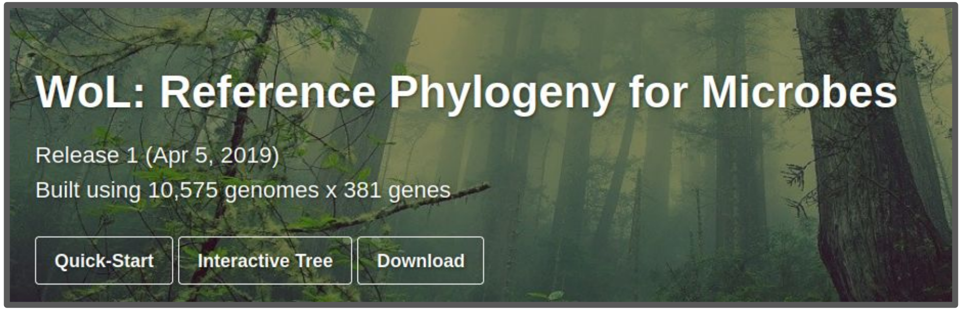
chooses a subset of representative *k*-mers

classifying reads

KRANK: Decontamination

Most Common Sources of Contamination:

- **Bacterial** → WoL-v1 dataset - archaeal and bacterial genomes



WoL: Reference Phylogeny for Microbes

Release 1 (Apr 5, 2019)

Built using 10,575 genomes x 381 genes

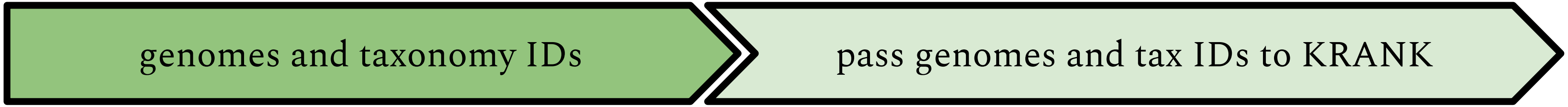
- **Human** → T2T Pangenome [05-2024]

KRANK: Decontamination

	genomes		tax IDs
1	ATGCTACC...T		52631
2	CTAGGAA...A		23100
3	CCCTTTA...C		23100
4	GATAAAA...A	×	48019
5	TATACTGC...C		5001
6	GTCGGCA...G		662
7	GGACTGC...C		80919
8	CTTATCC...G		80919

```
krank build
        -l $LIBRARY_DIRECTORY
        -t $TAXONOMY_DIRECTORY
        -i $MAPPING_FILE
        --from-library
        --batch-size 8
        --target-batch 0
        --num-threads $NUM_THREADS
```

https://github.com/bo1929/KRANK



Running KRANK build

```
cd ./skimming_scripts-echarvel/test/KRANK_test
```

```
krank build \
  -l ${LIBDIR} -t ./taxonomy/ \
  -i ./input_map.tsv  \
  -k 27 -w 35 -h 12 -b 8 -s 2\
  --from-scratch --input-sequences \
  --kmer-ranking representative --adaptive-size --lca soft \
  --num-threads ${NTHREADS} \
```

```
krank build \
  -l ${LIBDIR} -t ./taxonomy/ \
  -i ./input_map.tsv  \
  -k 27 -w 35 -h 12 -b 8 -s 2\
  --target-batch 0 --fast-mode --from-library --input-sequences --keep-intermediate \
  --kmer-ranking representative --adaptive-size --lca soft \
  --num-threads ${NTHREADS}
```

Running KRANK query

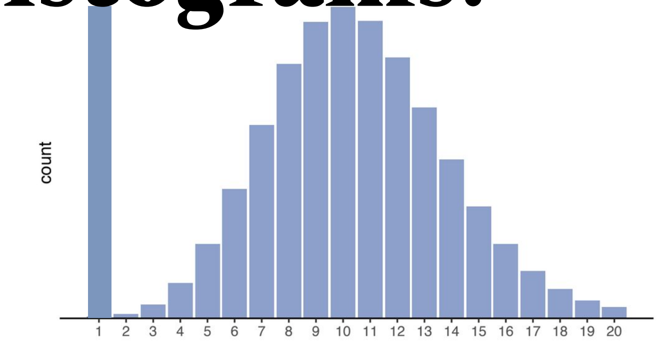
```
cd ./skimming_scripts-echarvel/test/
```

```
${SCRIPT_DIR}/KRANK/krank query \
        --library-dir ${LIBRARIES} \
        --query-file ./bbmap_read.fq  \
        --max-match-distance 5 \
        --total-vote-threshold 0.03 \
        --num-threads ${NUM_THREADS} \
        --output-dir "${OUTPUT_DIRECTORY}/krank_output/krank_reports/"
```

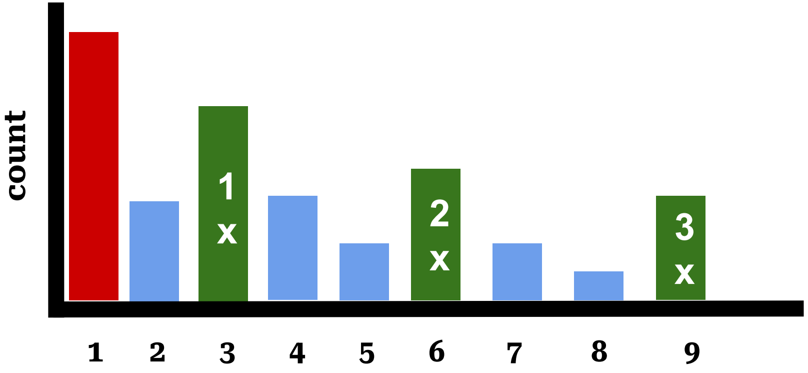
Estimating repeat spectra and genome length from low-coverage genome skims with RESPECT

Shahab Sarmashghi, Metin Balaban, Eleonora Rachtman, Behrouz Touri, Siavash Mirarab, Vineet Bafna ✉

How Repeats Affect *k*-mer Histograms:



simple genome skim

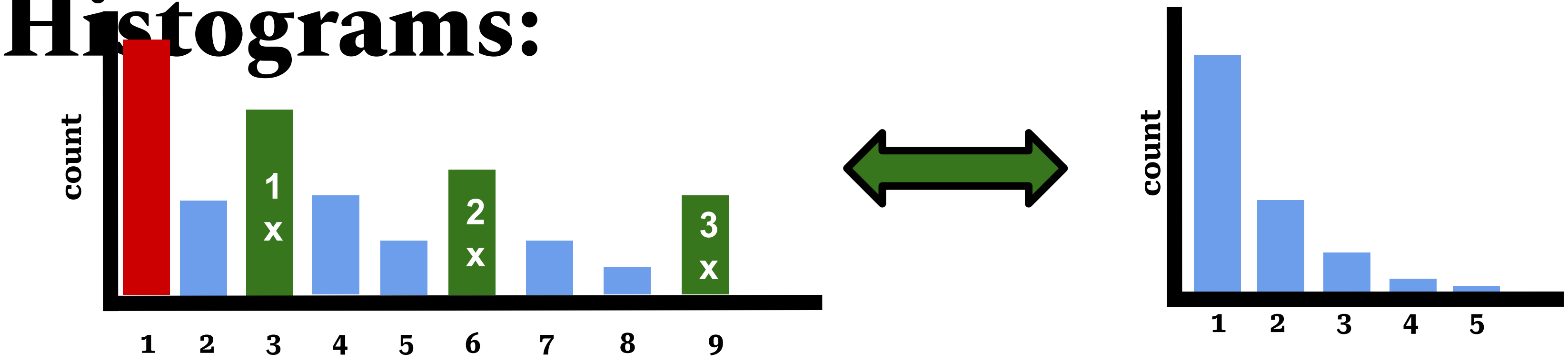


repetitive genome skim

Multiple peaks and coverages.

Can we get measures of repetitiveness using this?

How Repeats Affect *k*-mer Histograms:



repetitive genome

Repeat spectra of an assembly

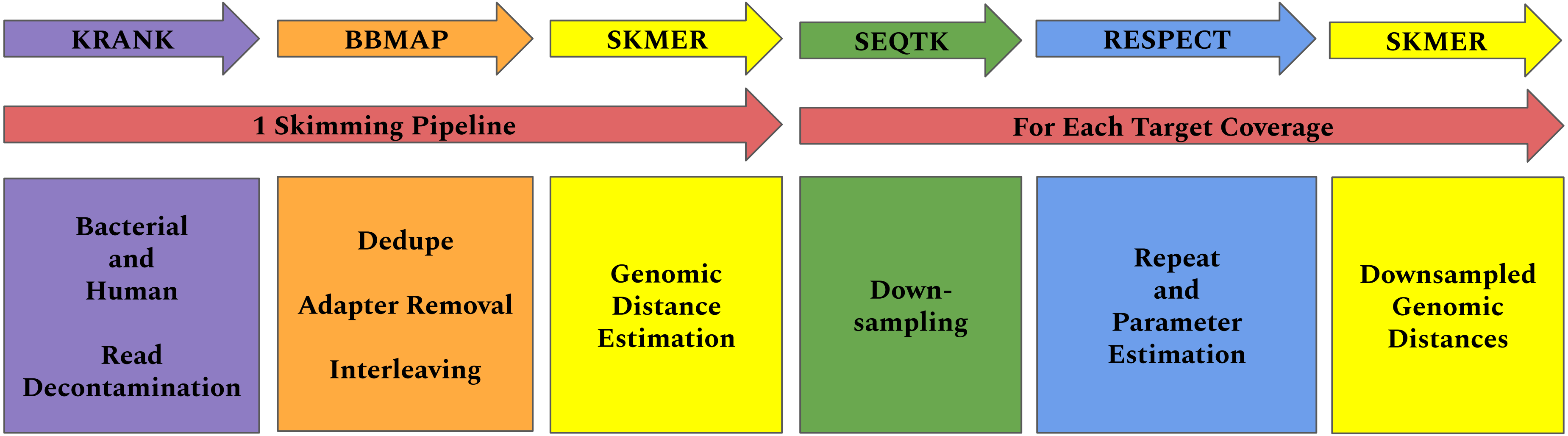
RESPECT output:

- repeat-informed coverage, error, and genome length estimates
- repeat spectra of a genome
- repetitiveness metrics:
  - Uniqueness Ratio, High Copy Repeats per Million

Running RESPECT

```
cd ./skimming_scripts-echarvel/test/
respect -d ./bbmap_reads/ -N 2 --threads 1
```

Using the Skimming Pipeline



KRANK

BBMAP

SKMER

SEQTK

RESPECT

SKMER

1 Skimming Pipeline

For Each Target Coverage

Bacterial and Human

Read Decontamination

Dedupe

Adapter Removal

Interleaving

Genomic Distance Estimation

Down-sampling

Repeat and Parameter Estimation

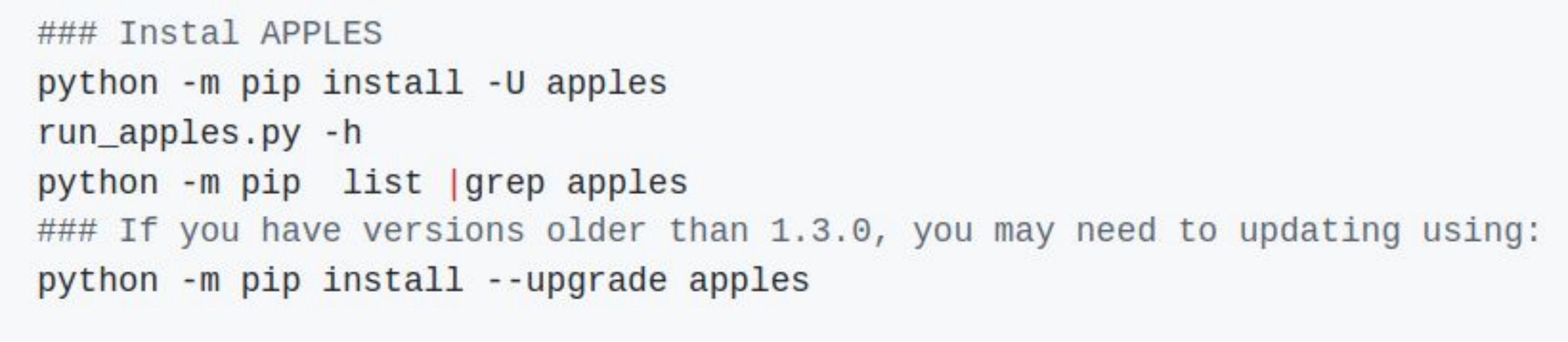
Downsampled Genomic Distances

```
bash ./skimming_pipeline.sh -h
```

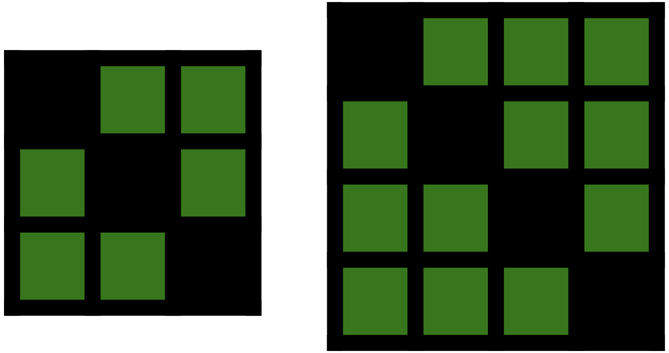
APPLES: Scalable Distance-Based Phylogenetic Placement with or without Alignments FREE

Metin Balaban, Shahab Sarmashghi, Siavash Mirarab ✉

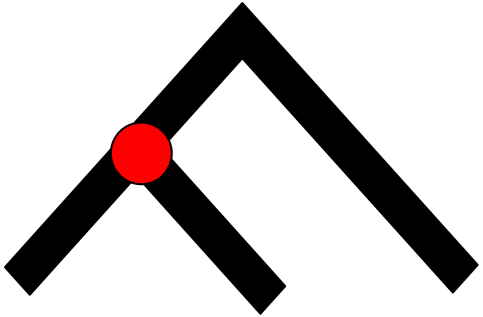
Installing Apples:



```
### Instal APPLES
python -m pip install -U apples
run_apples.py -h
python -m pip  list |grep apples
### If you have versions older than 1.3.0, you may need to updating using:
python -m pip install --upgrade apples
```



skmer query



Apples placement

Thank you!

FIELD MUSEUM Grainger
Bioinformatics
Center

MINDEROO

Siavash Mirarab and Vineet Bafna

Mirarab Lab:
Isaac, Daira, Homere, Ali, Shayasteh, Nora, and Yueyu

Field Museum: Grainger Bioinformatics Center
Minderoo Foundation