

A 20,000 league view of Bioinformatics

Kelsey Florek, MPH, PhD
2019 AMD Symposium
May 23, 2019

Slides available at:
www.k-florek.net/talks



[https://amd-midwest.github.io
/bioinfo_course/](https://amd-midwest.github.io/bioinfo_course/)

Bioinformatics: An interdisciplinary field that develops methods and software tools for understanding biological data.

What does the data look like

@M03478:141:000000000-C5B4D:1:1101:25956:10945 1:N:0:1
TTCCGTATTCATGCAACCTATGATGAAAGTATTAGTCGGTTACTCAATGTATTTGAGCGC
+
ABBBBCFFFFFFFGGGGGGGGGG5GHHHHHGHHHHHHHGGGGFHHHHHHHHHHHHHFGHGG

```
@M03478:141:000000000-C5B4D:1:1101:25956:10945 1:N:0:1
TTCCGTATTCATGCAACCTATGATGAAAGTATTAGTCGGTACTCAATGTATTTGAGCGC
+
ABBBBCFFFFFFGGGGGGGGGG5GHHHHHGGHHHHHHGGGGFHHHHHHHHHHHHHHFHHGG
```

- **M03478** - the unique instrument name
- **141** - the run id
- **000000000-C5B4D** - flowcell id

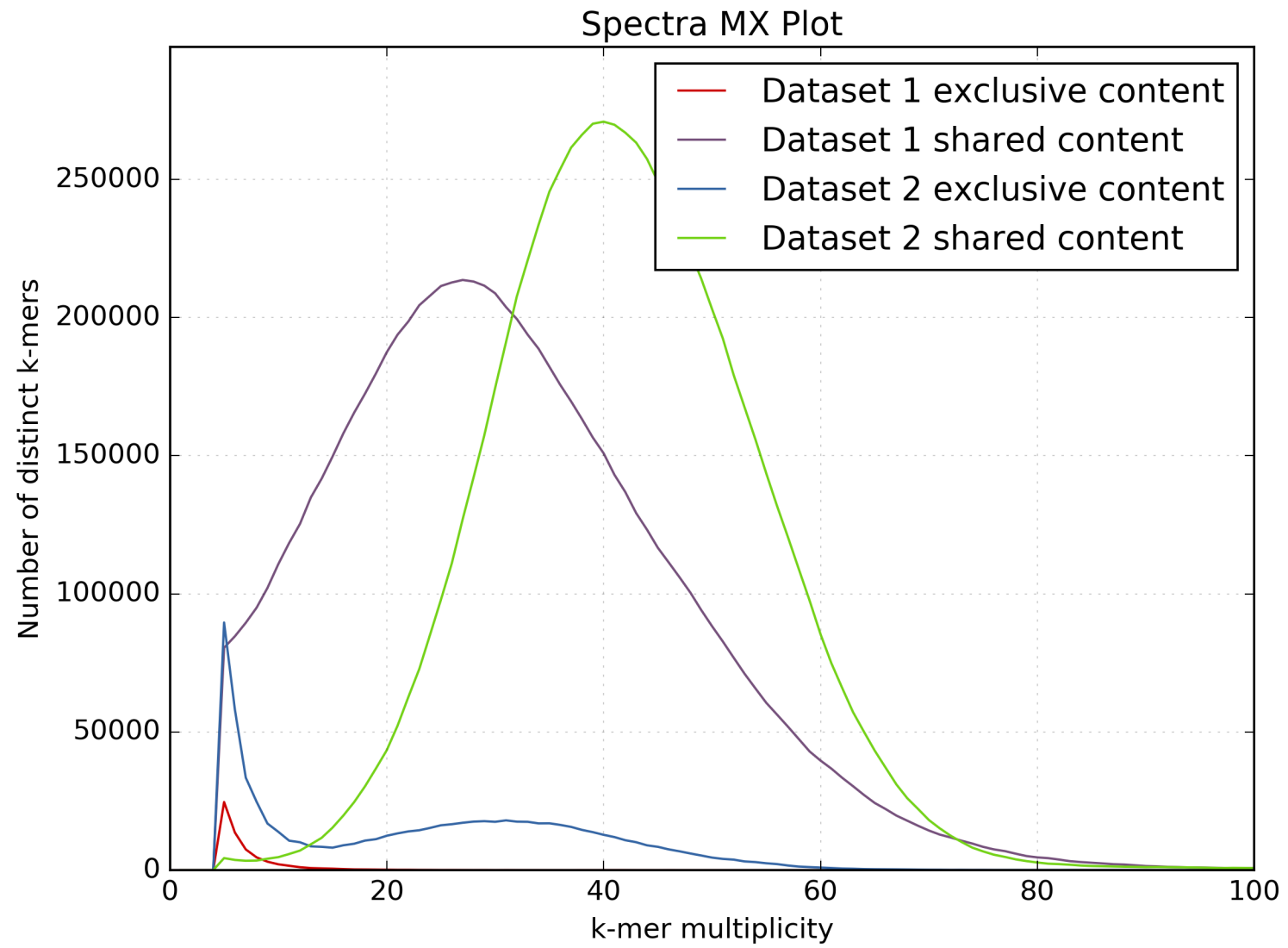
Phred Score

- 10: 1 in 10 90%
- 20: 1 in 100 99%
- 30: 1 in 1000 99.9%
- 40: 1 in 10,000 99.99%
- 50: 1 in 100,000 99.999%
- 60: 1 in 1,000,000 99.9999%

what can you do with fastq / read data

k-mers: all the possible substrings of length k

```
TTCCGTATTCATGCAACCTATGATGAAAGTATTAGTCGGTTACTCAATGTATTTGAGCGC
TTCCG  TTCAT  AACCT  GATGA  GTATT  TCGGT  CTCAA  TATTT
TCCGT  TCATG  ACCTA  ATGAA  TATTA  CGGTT  TCAAT  ATTTG
CCGTA  CATGC  CCTAT  TGAAA  ATTAG  GGTTA  CAATG  TTTGA
CGTAT  ATGCA  CTATG  GAAAG  TTAGT  GTTAC  AATGT  TTGAG
GTATT  TGCAA  TATGA  AAAGT  TAGTC  TTACT  ATGTA  TGAGC
TATTC  GCAAC  ATGAT  AAGTA  AGTCG  TACTC  TGTAT  GAGCG
ATTCA  CAACC  TGATG  AGTAT  GTCGG  ACTCA  GTATT  AGCGC
```



<https://kat.readthedocs.io>

basic analysis pipeline

- quality trimming
- assembly
 - *de novo* assembly
 - reference mapping
- antibiotic resistance detection

ensuring quality reads

```
@M03478:141:000000000-C5B4D:1:1101:25956:10945 1:N:0:1
TTCCGTATTCATGCAACCTATGATGAAAGTATTAGTCGGTTACTCAATGTATTTGAGCGC
+
ABBBBCFFFFFFFGGGGGGGGGG5GHHHHHGHHHHHHHGGGGFHHHHHHHHHHHHHHFHHGG
```

- remove sequencing adapters
- trim when quality drops
- specify a minimum length
- scan for contamination

basic analysis pipeline

- ~~quality trimming~~
- assembly
 - *de novo* assembly
 - reference mapping
- antibiotic resistance detection

de novo assembly: assembly of read data without the use of a reference sequence

de Bruijn graph: a directed graph representing overlaps between sequences of symbols

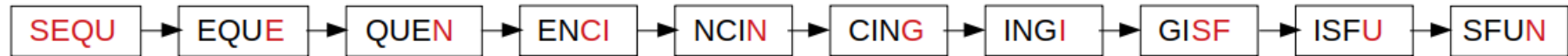
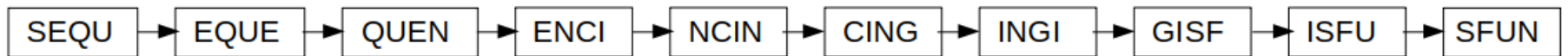
de Bruijn graphs

cingi sequen sfun encin cing isfu

all 4-mers: cing ingi sequ eque quen sfun enci ncin cing gisf isfu

unique 4-mers: cing ingi sequ eque quen sfun enci ncin gisf isfu

assembly graph:



sequencingisfun

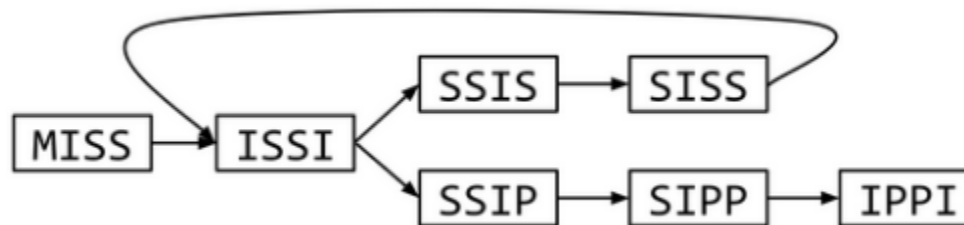
difficult de Bruijn graph

missis ssissi ssippi

all 4-mers: miss issi ssis ssis siss issi ssip sipp ippi

unique 4-mers: miss issi ssis siss ssip sipp ippi

assembly graph:



mississippi or mississississippi

choosing k

- low k
 - more connections
 - higher chance of repeats
 - higher coverage
- high k
 - less connections
 - higher chance of resolving repeats
 - lower coverage



storing genome assemblies (the .fasta file)

```
>A/Hong_Kong/4801/2014_NP  
gttaataatcactcactgagtgacatcaaagtcattggcgt  
ccaaggcaccaaacggtcttatgaacagatggaaactga  
tgagatcgccagaatgcaactgagattagggcatccgtc  
gggaagatgattgatggaattgggagattctacatccaaa
```

reference mapping: a method of mapping the reads to a
reference sequence

TATATTTATGCTATTCAGTTCTAAATATAGAAATTGAAACAGCTGTGTTTAGTGCCCTTTGTTCA-----ACCCCCTTGCAACAACCTTGAGAACCCCAGGGGAATTTG1
TATATT ATGCTATTCAGTTCTAAATATAGAAATTGAAACAG GTGTTTAGTGCCCTTTGTTCA-----ACCCCCTTGCAACAAC aaccccagggaatttgt
tatatttatgetattcagttctaaatatagaaatt acagctgtgttttagtgcctttgttca-----acccccttg aacaaccttgagaaccccagggaatttgt
TATAT TATGCTATTCAGTTCTAAATATAGAAATTGAAACA ctgtgttttagtgcctttgttca-----acccccttgcaac ACCTTGAGAACCCCAGGGGAATTTG1
TATATTTA getattcagttctaaatatagaaattgaaacagct GTTTAGTGCCCTTTGTTTACATAGACCCCCTTGCAA aaccttgagaaccccagggaatttgt
TATATTTATGCTATTCAGT GAAATTGAAACAGCTGTGTTTAGTGCCCTTTGTTCA ccccttacaacaaccttgagaaccccagggaattt
tatatttatgetattcagt GCCTTTGTTTACATAGACCCCCTTGCAACAACCTT cagggaatttgt
tatatttatgetattcagttcta AG-----ACCCCCTTGCAACAACCTTGAGAACCCCAGGGGA
TATATTTATGCTATTCAGTTCTAA A-----ACCCCCTTGCAACAACCTTGAGAACCCCAGGGGA
TATATTTATGCTATTCAGTTCTAAA A-----ACCCCCTTGCAACAACCTTGAGAACCCCAGGGGA
TATATTTATGCTATTCAGTTCTAAA TGCAACAACCTTGAGAACCCCAGGGGAATTTG1
TATATTTATGCTATTCAGTTCTAAAT TGCAACAACCTTGAGAACCCCAGGGGAATTTG1
TATATTTATGCTATTCAGTTCTAAAT TGCAACAACCTTGAGAACCCCAGGGGAATTTG1
tatatttatgetattcagttctaaatatagaaatt tgcaacaaccttgagaaccccagggaatttgt
tatatttatgetattcagttctaaatatagaaatt CAACCTTGAGAACCCCAGGGGAATTTG1
TATTTATGCTATTCAGTTATAAATATAGAAATTGAAACAG CTTGAGAACCCCAGGGGAATTTG1
atatttatgetattcagttctaaatatagaaattgaa CTTGAGAACCCCAGGGGAATTTG1
tttaegetattcagtaactaaatatagaaattgaaa CTTGAGAACCCCAGGGGAATTTG1
ttatgetattcagttctaaatatagaaattgaaac gggaatttgt

storing read mapping (the .sam file)

- read name / reference name
- position read maps to on the reference sequence
- sequence read and quality information
- many others..

storing the read mappings in a binary format
(the .bam file)

provides a faster access to data and tends to use less memory

```
000b8e54: 11011110 10001110 01010011 11111110 10001110 11001110 ..S...
000b8e5a: 10101101 00100100 11100111 01000001 10100000 11000000 .$A..
000b8e60: 11000111 01011011 11000101 11110010 01011011 00011100 .[...[.
000b8e66: 01001110 01110000 10101001 11000001 01011011 10011011 Np...[.
000b8e6c: 11100000 01011010 00000111 00010110 10001100 01011000 .Z...X
000b8e72: 11001110 11001101 01100101 01110100 01010111 00000001 ..etW.
000b8e78: 10101011 01000111 00011101 10000000 01101000 11110011 .G..h.
000b8e7e: 10110110 11001010 00110100 10111000 11110011 00011111 ..4...
000b8e84: 10111110 11101111 01000110 00110011 01110111 00011010 ..F3w.
000b8e8a: 11111111 10011000 01010001 01010011 01010110 00011100 ..QSV.
000b8e90: 01011011 01001111 00010010 01011011 10110100 00101010 [0.[.*
000b8e96: 01100111 01010001 11110011 00111010 11101100 01010101 gQ...U
000b8e9c: 10010001 11010001 01010100 11011011 00110001 11110110 ..T.1.
000b8ea2: 10110110 11011011 10111100 01000000 11101010 11110100 ...@..
000b8ea8: 10111000 00010110 11100100 01110110 10011000 01111110 ...v.~
000b8eae: 00111111 11101011 10110100 01001110 10010001 01100101 ?..N.e
000b8eb4: 01110101 10101000 01110111 10000111 00001111 00110111 u.w..7
000b8eba: 01101100 10010101 01111010 11111001 10010100 00000011 l.z...
000b8ec0: 11100011 01001111 10011111 10011110 00110111 00100111 .0..7'
000b8ec6: 00010101 10110110 00111111 10001010 11110010 01011110 ..?...^
000b8ecc: 00010100 01111111 01011111 01000100 01111001 10111111 .._Dy.
000b8ed2: 11100111 01010110 10101001 11110010 11100101 00110100 .V...4
```

compression

- gzip
- repetitions in the data are replaced by references to the data
- repetitions in the data are replaced by references to <7,8>
- replaces more frequent characters with variable length encoding
- T : 01010100 ----> T : 11

compression matters

- Uncompressed:
 - **E coli** both set of reads ~900MB
 - **E coli** sequencing run (16 isolates) ~20GB
- Compressed:
 - **E coli** both set of reads ~200MB
 - **E coli** sequencing run (16 isolates) ~4GB

moving data

data moves across the internet in 1,500 byte packets

- ftp
- http
- sftp
- https

basic analysis pipeline

- ~~quality trimming~~
- assembly
 - ~~*de novo* assembly~~
 - ~~reference mapping~~
- antibiotic resistance detection

using the data to find resistance mechanisms

- database
 - multifasta
 - SQL
- search for patterns

NCBI BLAST (basic local alignment search tool)

BLAST finds similar sequences by locating short matches between sequences

after the first match BLAST begins to make local alignments

location: 4377811 - 4378944

gene name: Escherichia_coli_ampC

coverage: 100

identity: 100

database: card

description: A class C ampC beta-lactamase (cephalosporinase) enzyme described in Escherichia coli shown clinically to confer resistance to penicillin-like and cephalosporin-class antibiotics.

review

- quality control / trimming of reads
- assembly
 - *de novo*
 - reference mapping
- AR detection using BLAST

review

- sequencing data storage
- data compression
- transferring data across networks

applied Linux virtual course

<https://forms.gle/oKSB5KFKcv5DX4k57>

Course Dates: June 10th - June 14th, 2019

Length: 2hr sessions Monday, Wednesday and Friday; Office hours on Tuesday and Thursday

kelsey.florek@slh.wisc.edu