# Quantitative Cellular and Molecular Biology Laboratory
# Computational Biology Department
# Comp Bio 02-261

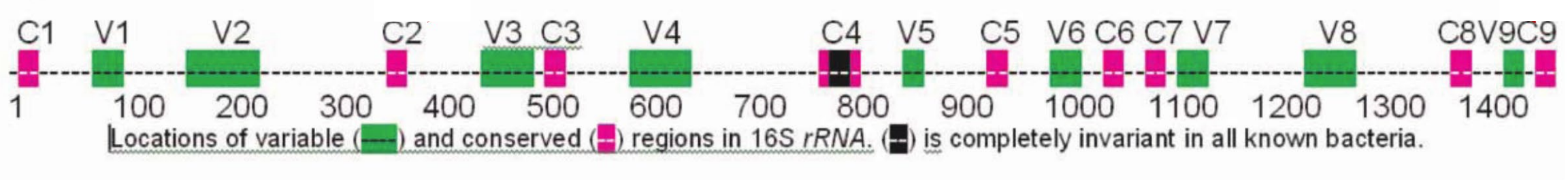
## Bacteria Analysis Lab

# Microbiome Sequencing

Microbiome – microbiological population of an environment

- Extract DNA
- Use PCR to amplify 16S gene.  How?

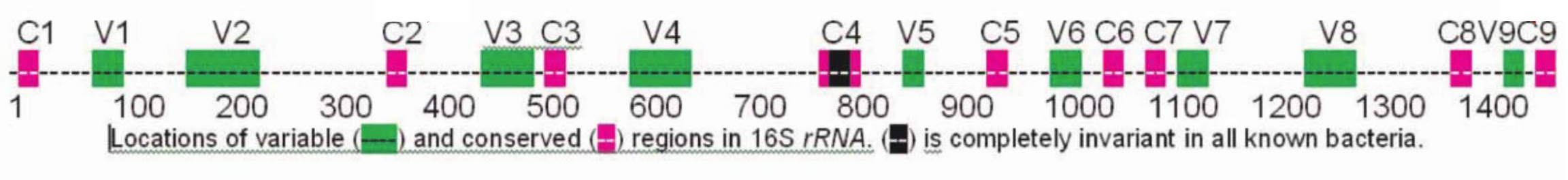
# From Lab 2



Locations of variable (——) and conserved (--) regions in 16S *rRNA*. (--) is completely invariant in all known bacteria.

- What are variable regions?
- What are conserved regions?
- Why would some regions be conserved?
- If want to amplify DNA from a single species specifically, where should we put our primers?
- If want to amplify DNA from all species non-specifically, where should we put our primers?

# From Lab 2




- What are variable regions?
- What are conserved regions?
- Why would some regions be conserved?
- If want to amplify DNA from a single species specifically, where should we put our primers?
- If want to amplify DNA from all species non-specifically, where should we put our primers?
- **If I want to identify species in a mixture of species with sequencing, where should we put our primers?**

# Microbiome Sequencing

Microbiome – microbiological population of an environment

- Extract DNA
- Amplify 16S ribosomal RNA gene (found in all bacteria)
- Sequence copies of amplified 16S gene DNA from across all species present.
- How do we identify the species of millions of reads?
  - We'll compare a couple methods in this lab.

# How will we identify your species of bacteria?

- Search a database of known 16S genes for a match to your sequence (example.fa).
- BLAST (Basic Local Alignment Search Tool)

- **How long did that take to run?**

# Tasks

1. Split the dataset into query set (50 sequences) and database set (200 sequences)
2. Implement function for matching of experimental read to known 16S gene using local alignment
3. Implement alignment free sequence matching
4. Plot agreement curve as a function of k-mer size (k=1,3,5,7,9,11,13,15,17,19)
5. (12 unit only) Implement minimizers for sequence matching
6. (12 unit only) Plot agreement curve as a function of window size with the best k-mer size from step 4. (windows = 20,25,30,35,40,45,50,55,60,65)
7. Run the analysis above with two mutated datasets.
8. Benchmark each method you implemented (seconds per query sequence)
9. BLAST your 16s sequence
10. Analyze your 16s sequence with your favorite method

# What are you provided with?

- Library of most known 16S genes
  (n=20,486; average bp = 1350)
- Code with some helper functions implemented for you already

# Task 1: split available data randomly

- Select two random subsets of sequences from the 16S gene list.
  - Knowns (200 sequences, database) and Unknowns (50 sequences, query set)

# 2. Implement function for matching of experimental read to known 16S gene

- For a given sample sequence:
  - Determine 16S sequence with greatest local alignment to your query sequence

# 3. Implement alignment free sequence matching

- Alignment Free Sequence Matching
  - Matching two sequences based on the relative presence or absence of *k*-mers
  - *k*-mer = substring of length k
  - Example:
    - ACTGA -> 1-mer -> [A,C,T,G]
    - ACTGA -> 2-mer -> [AC, CT, TG, GA]
    - ACTGA -> 3-mer -> [ACT,CTG,TGA]
    - ACTGA -> 4-mer -> [ACTG,CTGA]

# 3. Implement alignment free sequence matching

- Simple Alignment Free Sequence Matching Algorithm
  1. Convert each sequence to k-mer sets
  2. Calculate Jaccard index of the pair of sets

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

$$\frac{\text{Count k-mers in both A and B}}{\text{Count k-mers in A and/or B}}$$

$$0 \leq J(A,B) \leq 1$$

- Determine similarity of Sequences A and B
- A = ACTGGA
- B = CGTGAG

# 3. Implement alignment free sequence matching

- Example: Determine similarity of Sequences A and B (k=2)
  - A = ACTGGA
  - B = CGTGAG

$$\frac{\text{Count k-mers in both A and B}}{\text{Count k-mers in A and/or B}}$$

# 3. Implement alignment free sequence matching

- Example: Determine similarity of Sequences A and B (k=2)
  - A = ACTGGA -> [AC, CT, TG, GG, GA]
  - B = CGTGAG -> [CG, GT, TG, GA, AG]

$$\frac{\text{Count k-mers in both A and B}}{\text{Count k-mers in A and/or B}}$$

# 3. Implement alignment free sequence matching

- Example: Determine similarity of Sequences A and B (k=2)
  - A = ACTGGA -> [AC, CT, TG, GG, GA]
  - B = CGTGAG -> [CG, GT, TG, GA, AG]

$$\frac{[TG, GA]}{[AC, CT, TG, GG, GA, CG, GT, AG]} = \frac{2}{8}$$

$$\frac{\text{Count k-mers in both A and B}}{\text{Count k-mers in A and/or B}}$$

**Issues with this approach?**

# 3. Implement alignment free sequence matching

- Example: Determine similarity of Sequences A and B (k=2)
  - A = ACTGGA -> [AC, CT, TG, GG, GA]
  - B = CGTGAG -> [CG, GT, TG, GA, AG]

$$\frac{[TG, GA]}{[AC, CT, TG, GG, GA, CG, GT, AG]} = \frac{2}{8}$$

$$\frac{\text{Count k-mers in both A and B}}{\text{Count k-mers in A and/or B}}$$

**Issues with this approach?**
What if sequences are different lengths?

# 4. Plot agreement curves for different size k-mers across different thresholds (k=1,3,5,7,9,11,…,19)

- Calculate the pairwise alignment score and sequence free alignment scores for both subsets.

Alignment Scores

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 251 | 19 | 32 |
| Seq 2 | 22 | 150 | 50 |
| Seq 3 | 153 | 141 | 92 |

Alignment Free Scores (k=3)

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 0.95 | 0.92 | 0.11 |
| Seq 2 | 0.21 | 0.52 | 0.42 |
| Seq 3 | 0.32 | 0.11 | 0.89 |

# 4. Plot agreement curves for different size k-mers across different thresholds (k=1,3,5,7,9,11)

- Calculate the pairwise alignment score and sequence free alignment scores for both subsets.

Alignment Scores

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 251 | 19 | 32 |
| Seq 2 | 22 | 150 | 50 |
| Seq 3 | 153 | 141 | 92 |

Alignment Free Scores (k=3)

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 0.95 | 0.92 | 0.11 |
| Seq 2 | 0.21 | 0.52 | 0.42 |
| Seq 3 | 0.32 | 0.11 | 0.89 |

Do our predictions for sequence A match between the two methods?

# 4. Plot agreement curves for different size k-mers across different thresholds (k=1,3,5,7,9,11)

- Calculate the pairwise alignment score and sequence free alignment scores for both subsets.

Alignment Scores

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 251 | 19 | 32 |
| Seq 2 | 22 | 150 | 50 |
| Seq 3 | 153 | 141 | 92 |

Alignment Free Scores (k=3)

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 0.95 | 0.92 | 0.11 |
| Seq 2 | 0.21 | 0.52 | 0.42 |
| Seq 3 | 0.32 | 0.11 | 0.89 |

# 4. Plot agreement curves for different size k-mers across different thresholds (k=1,3,5,7,9,11)

- Calculate the pairwise alignment score and sequence free alignment scores for both subsets.

Alignment Scores

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 251 | 19 | 32 |
| Seq 2 | 22 | 150 | 50 |
| Seq 3 | 153 | 141 | 92 |

Alignment Free Scores (k=3)

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 0.95 | 0.92 | 0.11 |
| Seq 2 | 0.21 | 0.52 | 0.42 |
| Seq 3 | 0.32 | 0.11 | 0.89 |

# 4. Plot agreement curves for different size k-mers across different thresholds (k=1,3,5,7,9,11)

- Calculate the pairwise alignment score and sequence free alignment scores for both subsets.

Alignment Scores

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 251 | 19 | 32 |
| Seq 2 | 22 | 150 | 50 |
| Seq 3 | 153 | 141 | 92 |

Alignment Free Scores (k=3)

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 0.95 | 0.92 | 0.11 |
| Seq 2 | 0.21 | 0.52 | 0.42 |
| Seq 3 | 0.32 | 0.11 | 0.89 |

# 4. Plot agreement curves for different size k-mers across different thresholds (k=1,3,5,7,9,11)

- Calculate the pairwise alignment score and sequence free alignment scores for both subsets.

Alignment Scores

Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 251 | 19 | 32 |
| Seq 2 | 22 | 150 | 50 |
| Seq 3 | 153 | 141 | 92 |

Alignment Free Scores (k=3)

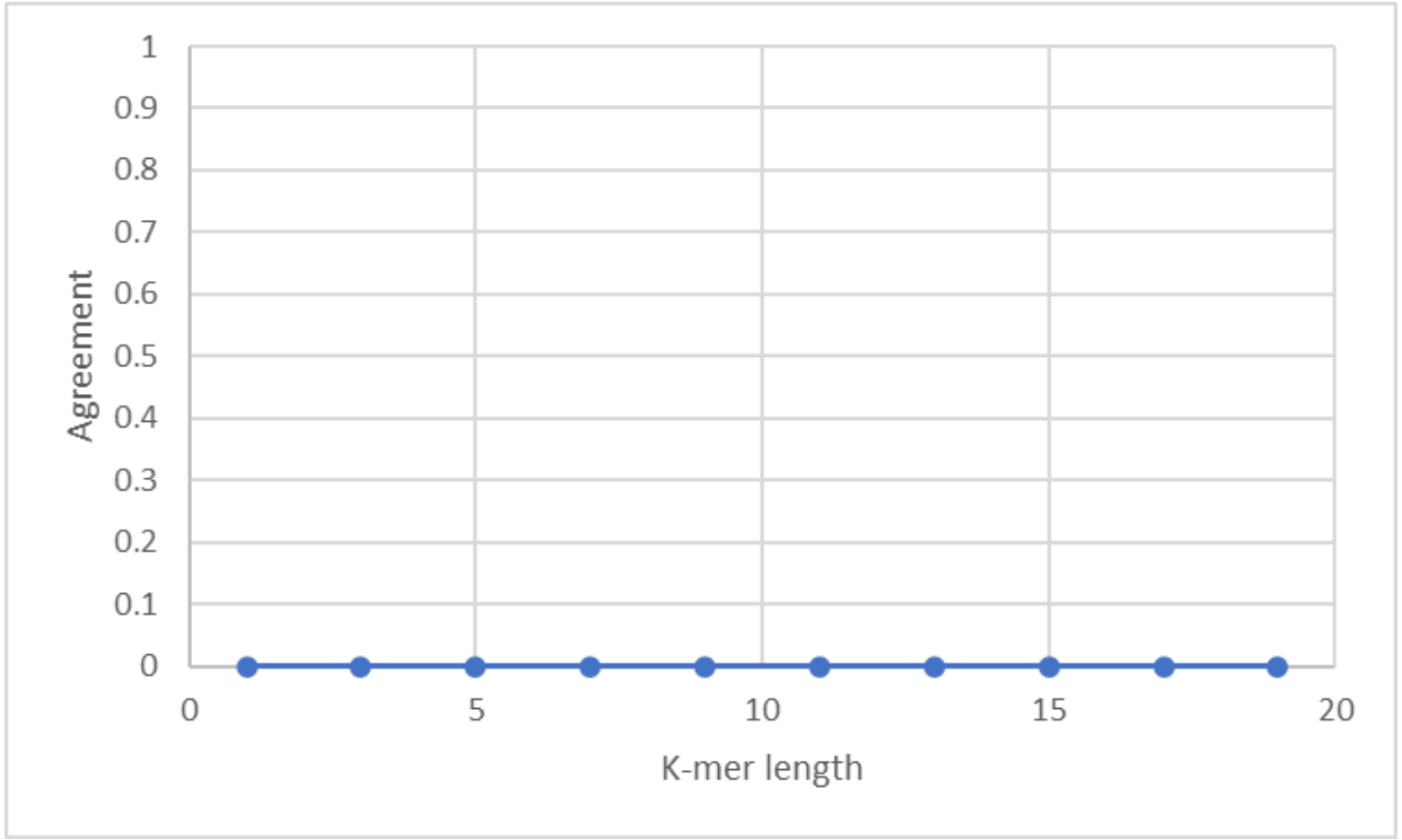
Unknowns (query set)

Knowns (Database)

| | Seq A | Seq B | Seq C |
|---|---|---|---|
| Seq 1 | 0.95 | 0.92 | 0.11 |
| Seq 2 | 0.21 | 0.52 | 0.42 |
| Seq 3 | 0.32 | 0.11 | 0.89 |

Is agreement the same as accuracy?
Why or why not?

# 4. Plot agreement curves for different size k-mers across different thresholds (k=1,3,5,7,9,11,13,15,17,19)




What agreement would you expect at k=1? k=5?

# 5. Implement Minimizers

Minimizer is the “smallest” k-mer within a window of length m.

Example: Window of length 20, k=4

`ACTGGACTACGCGAATGACC -> Minimizer = ?`

What is a reasonable meaning for “smaller”?

# 5. Implement Minimizers

Minimizer is the “smallest” k-mer within a window of length m.

Example: Window of length 20, k=4

`ACTGGACTACGCGAATGACC -> Minimizer = ?`

What is a reasonable meaning for “smaller”? Alphabetical order!

# 5. Implement Minimizers

Minimizer is the “smallest” k-mer within a window of length m.

Example: Window of length 20, k=4

`ACTGGACTACGCGAATGACC -> Minimizer = AATG`

What is a reasonable meaning for “smaller”? Alphabetical order!

# 5. Minimizer Algorithm

Pre-calculate the minimizers for your database set and store them.

Number of windows per sequence = (length of sequence/(m-k))

*equal length with slight overlaps between windows*

# 5. Minimizer Algorithm

Pre-calculate the minimizers for your database set and store them.

Number of windows per sequence = (length of sequence/(m-k))

*equal length with slight overlaps between windows*

*m = 13, k=4*

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT (50 bp)

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT

# 5. Minimizer Algorithm

Pre-calculate the minimizers for your database set and store them.

Number of windows per sequence = (length of sequence/(m-k))

*equal length with slight overlaps between windows*

*m = 13, k=4*

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT (50 bp)

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT > ACGT

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT > ACCG

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT > ACGG

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT > ATAT

ACTGCTACGTACGACCGACTAGCGGTACGGCTATATATTCGGCGCTAGCT > AGCT

# 5. Minimizer Algorithm

Pre-calculate the minimizers for your database set and store them.

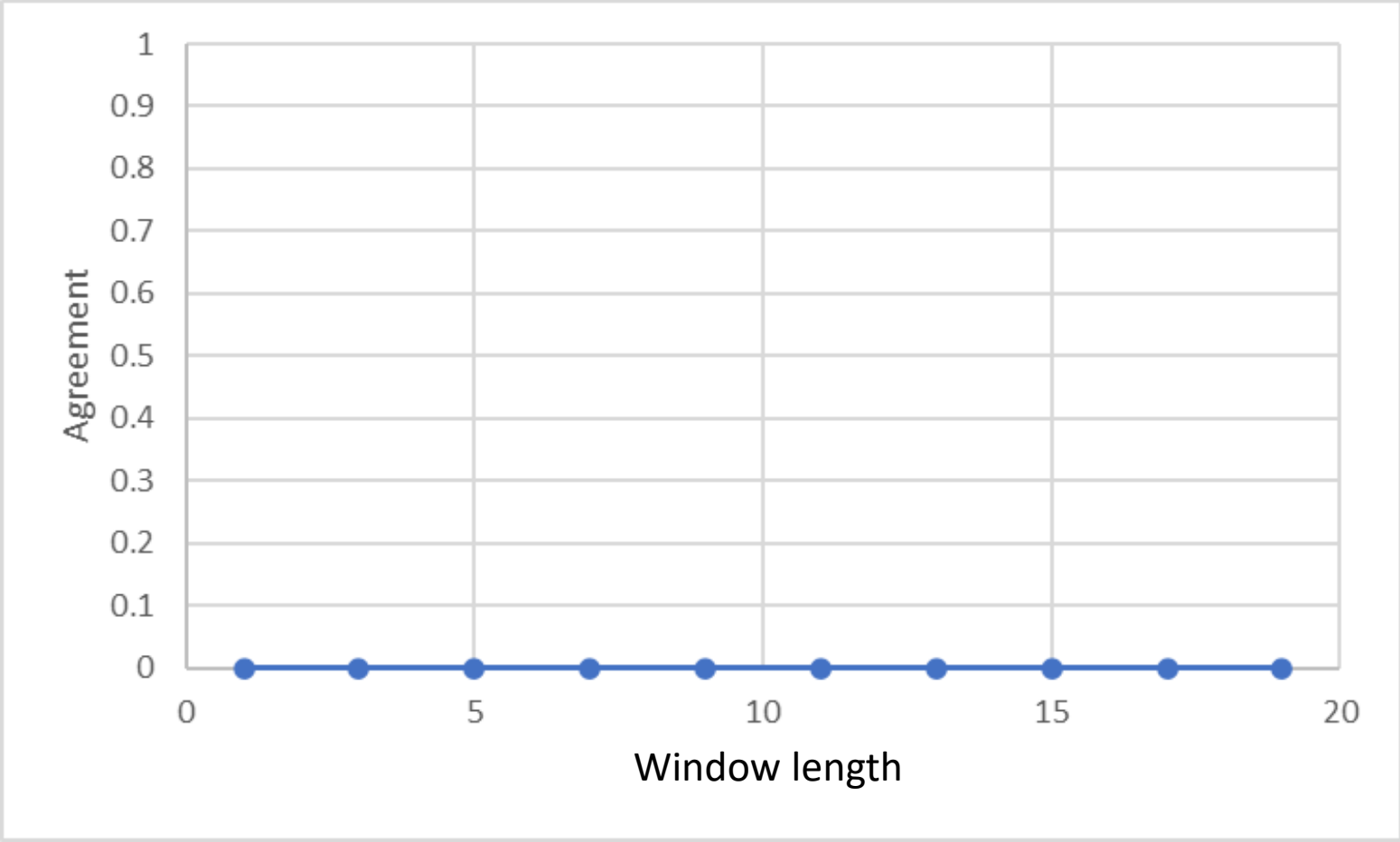
Number of windows per sequence = (length of sequence/(m-k))

*equal length with slight overlaps between windows*

Query:

1. Calculate minimizers for query sequence.
2. Run alignment free sequence matching on the query sequence and any database sequences which match at least one minimizer.
3. Of those database sequences for which you ran alignment free sequencing, accept the highest scoring sequence as the match to your query.

6. Plot agreement curve as a function of window size with the best k-mer size from step 4. (windows = 20,25,30,35,40,45,50,55,60,65)



# 7. Mutated Datasets

Recall from lecture, there are different error rates and sequence read lengths for different sequencing methods.

Illumina – 99% accurate (1/100 bases are incorrect in any given read), reads <= 250 bp

Oxford Nanopore – 90% accurate (~1/10 bases are incorrect in any read), reads > 10kb

Modify your query dataset sequences based on these two sequencing strategies. How would we do that?

# 7. Mutated Datasets

- Rerun your code for previous steps with mutated datasets.
- For agreement, compare against best match from alignment to unmutated sequence.

# 8. Benchmark

- Ignoring any precomputation on the database dataset, compare the runtime to find the best match for a query sequence using the best performing parameters for each method.

- Your plot should include times and max agreement for the following:
  - Local Alignment
  - Alignment free sequence matching (with best k)
  - Minimizers (with best k and m) (required for 12 unit only)
  - *Minimizers (with best k and m while only running AFSM on sequences with at least 2 minimizer matches)*
  - *Minimizers (with best k and m while only running AFSM on sequences with at least 4 minimizer matches)*
  - *Minimizers (with best k and m while only running AFSM on sequences with at least 6 minimizer matches)*

Extra Credit! (more for 9 unit than 12 unit)

# What to turn in?

- Task 1-8: code and plots
- Task 9: screenshot of BLAST result, paragraph describing top hit of the sequence from your isolated bacteria. (Where is it commonly found? What are some interesting characteristics of this bacteria? Do they match what you observed in the colony you isolated?) If you don't have a hit on your sequence, discuss why you might not have a match.
- Task 10: Trim any N's off of the ends of your bacterial sequences and run your sequence against the full database using the method of your choice (alignment, alignment-free sequence matching, minimizers followed by alignment free sequence matching).

  ***Note: If you didn't end up with a good sequencing run, pick a classmate's sequence for 9 and 10.***

# Important Hint:

- Any time you calculate a score, save it to disk and reuse the data later. Avoid recalculating alignments or scores.
- Don’t melt your computers!
- Today focus on writing code for a smaller library and query set (ex. 10 x 5) then run later with full set (200 x 50).