Program File Tree Structure and Auxiliary Files Introduction

```
├── res # dir to store result, and corresponding results will be explained
later
│   ├── ablation
│   ├── alignment_analysis
│   ├── true_reads
│   └── contigs
├── matrices
│   ├── pam250.m
│   └── unitary.m
├── src
│   ├── assembly.py # to run test on SampleReads
│   ├── hiv_assembly.py # to run test on HIV
│   ├── debruijn.py # debruijn graph assembly
│   ├── local_alignment.py # apply local alignment
│   ├── correction.py
│   ├── contamination.py
│   └── preprocessing.py
├── test_cases
│   ├── assembly
│   └── debruijn
│
├── Assembly.pdf # report
├── debruijn.sh
├── assembly.sh
└── structure.tree # use to generate and keep track file tree structure
```

Shell Command

When run any command, make sure you are under the corresponding directory.

Specifically, here, you need to under the root dir.

Local Alignment

```
sh debruijn.sh reads.txt
-----------------------------------------------------------
exm: sh debruijn.sh ./test_cases/debruijn/reads1.txt
```

Assembly for SampleReads

```
sh assembly.sh <vector.txt> <sampleReads.txt> contamination_k correction_k
correction_d correction_t correction_mode
-----------------------------------------------------------
exm: sh assembly.sh vector.txt sampleReads.txt 8 17 2 4 20 stack
```

Assembly for HIV

```
go to the hiv_assembly.py dir, the parameters can be adjusted in the program
-----------------------------------------------------------
python3 hiv_assembly.py
```

Part1: Debruijn Graph

```
sh debruijn.sh reads1.txt
```

Part 2

The relative order of HIV and SampleRead Analysis is inversed, since the writer conducted experiements on HIV first, and also referred some conclusion in HIV for the SampleRead

HIV Reads Assembly

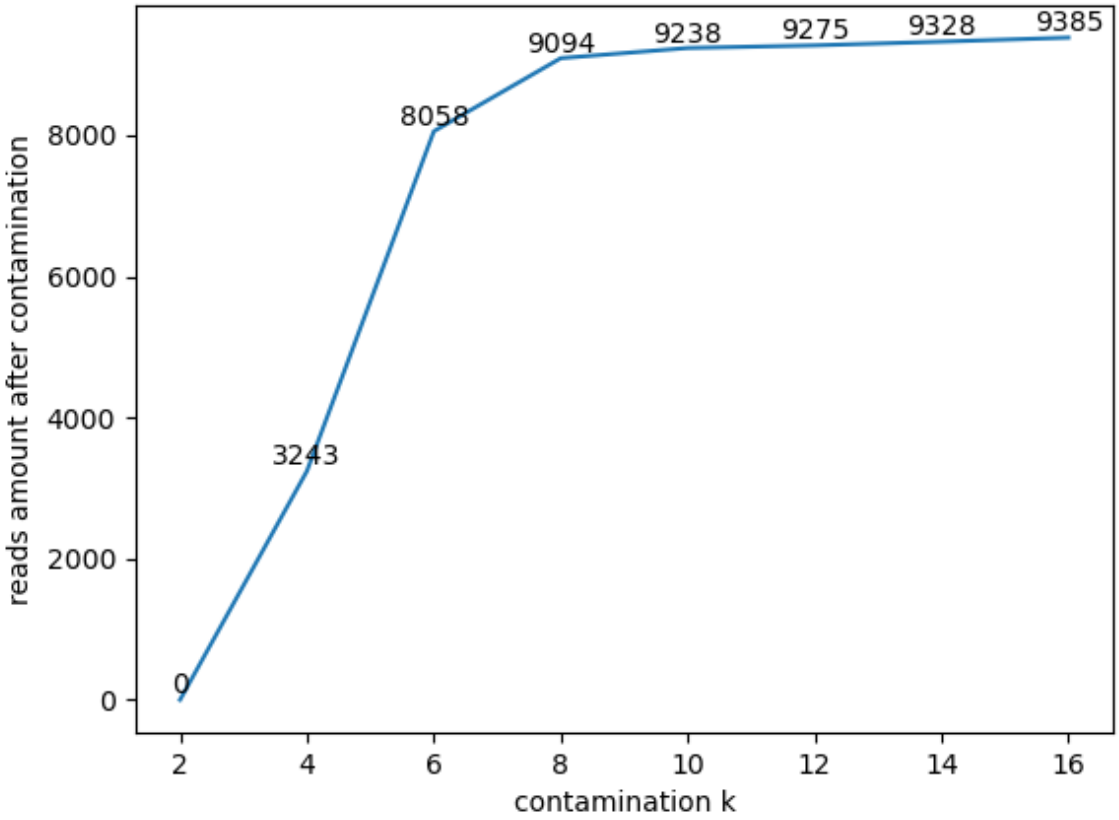
Ablation Test

How contamination k affects the result?

Tests have been conducted to different value of contamination k, and the left reads number is recorded.

Intuively, we know that the contamination k is used to define a whether a read is contaminated: a contiguous subsequence of the contamination source of length greater than or equal to k located at the beginning or end of a read. Therefore, when k is larger, there will be more reads left since it is a more strict threshold.

The writer tested contamination k in range [2,4,6,8,10,12,14,16], and the remained reads number is as following:



Now, as implies in the instruction, we know about 10% of total **9681** reads are contaminated, so the contaminated number would be near **968**, and the reads amount of contamination process would be around 9681-918 = 8763, which is abundant amount of reads for high coverage. So here, we will use contamination k as 6 and get 8058 reads as our later input.

Correction hyper-parameters

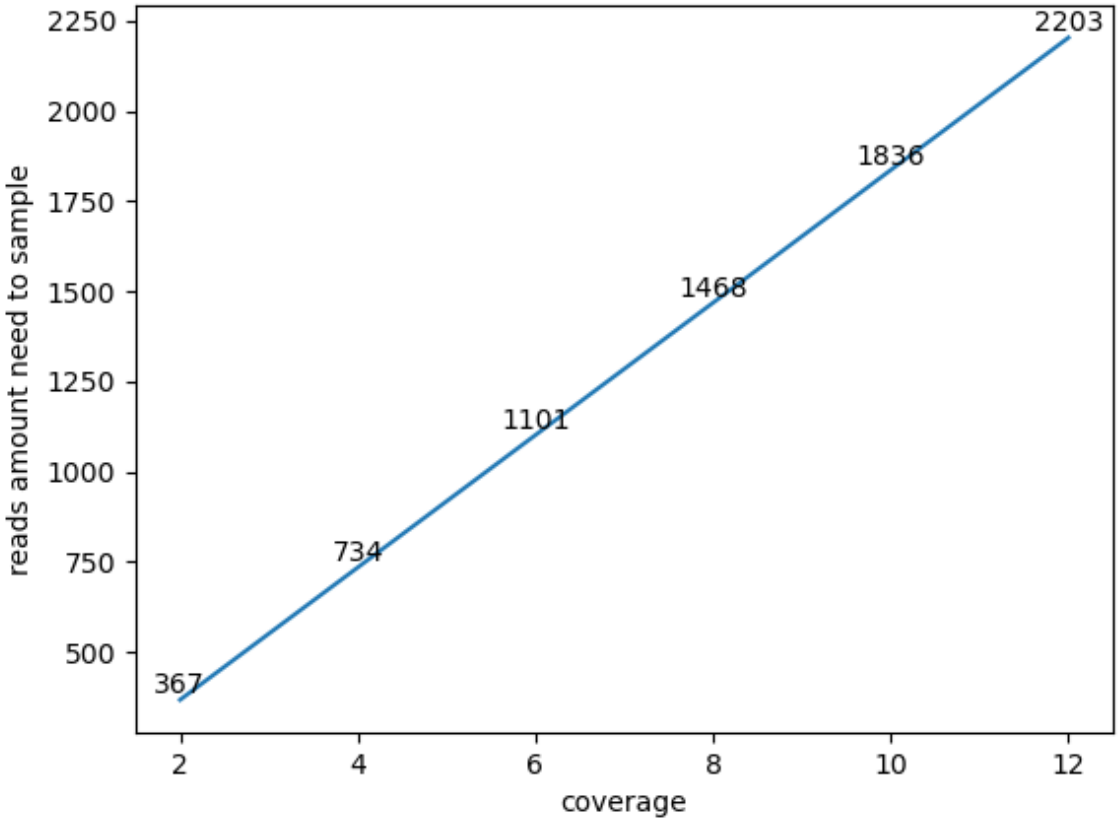
Since we are using the same dataset as Project2, therefore, we can use the conclusion in Project2 and use

- CORRECTION_k = 17
- CORRECTION_t = 4
- CORRECTION_d = 2

What is the corresponding sampled reads number for different coverage?

With $a = \frac{NL}{G}$, where a is the coverage, N is the sampled number, L is the reads length, G is the orginal sequence length, we can calculate N for different coverage a.

The writer explored the a in range [2, 4, 6, 8, 10, 12], and the result is as following:



Hyper-parameters Grid Search

What is the coverage range?

The writer applied grid search on the coverage space: [4,6,8,10]

What is the k range for graph k-mer?

The writer applied grid search on the graph k space: [10,20,30,40,50]

How many contigs need to be reserved?

Now, instead of choosing the longest path inferred in De Bruijn Graph, we want to produce several inferred sequences (or contigs). So it is important to determine how many contigs we need to choose. There are three methods we can consider with corresponding pros and cons.

Method 1: Mean Contigs Number from Lander-Waterman Statistics

Here, we use the Lander-Waterman statistics. Remember when we answering the mean number of contigs, we have formulate the problem as the following:

> Each contig has a unique rightmost read by definition. The probability that no reads cover an arbitrary region of length L is e^{-a} as shown above. This quantity is therefore the probability that no other read has a left-hand end within the length L defined by a given read; i.e., the probability of "success" (p) for a Bernoulli trial. Modeling the number of unique contigs as a binomial distribution (where "success" indicates that a given read is the rightmost read of a unique contig), it follows that the mean number of contigs is

$$Np = \boldsymbol{Ne^{-a}}$$

Therefore, if we consider the problem reversely, the mean number of contigs can be seen as the number we want to preserved from the result of the De Bruijn Graph. So for each given hyper-parameter set: N, L, G (also coverage), we can calculate the corresponding mean number of contigs, and choose that number as the contigs we reserve.

- Pros: this is kind of "cheating" since we know the Lander-Waterman statisitcs should give a relatively correct answer
- Cons: it only gives the mean number, therefore, the number of true contig number might be larger or smaller than the mean number. If mean number we use is larger than the true number, it is ok since what we probably will get is the some repeated or redundant contigs; however, is the mean number is smaller than the true number, then it can cause problems since we cannot evaluate the quality of the inferred genome.

Method 2: Threshold

Previously, we only choose the longest path as output in part 1 by seaching the contigs with length equal to the stored max length , now we can simply set a threshold h, and includes the contigs with $length \geq max - h$.

- Pros: easy to realize
- Cons: different hyper-parameters can generate contigs with different length, and the distribution of the lengths are different with different mean and variance. However, threshold is fixed, which can lead to the situation: for example, our threshold is 5, for certain hyperparameter setting, we get average mean contig length be 1000 with relatively larger variance; for another hyperparameter setting, we get average mean contig length be 10 with relatively small variance, this will result we obtain much less contigs from the 1000 one than the 10 one.

Method 3: Percentage

This method is very intuitive, we get the number of all paths in certain De Bruijn with specific hyper-parameters setting, then we take top $x\%$ of the longest paths as contigs.

The percentage can be chosen by Lander-Waterman statistics with average length of the contigs. Set the average length of the contigs as threshold, and see how many contigs are larger than the threshold, calculate the percetage, and apply fine-tune finally.

In this experiment, the writer chooses 7%. You are encouraged to explore by your own.

- Pros: make sure no matter for what kind of settings, we can always have abundant contigs
- Cons: the number of contigs chosen by percentage might be redundant for certain hyper-parameter settings, thus we will spend extra calculation on unnecessary contigs (but it is always better to have more than less)

In conclusion, the writer tested all of the three methods, and finally decided to use the method3 with percentage be 7%.

Metrics

Single Alignment Score

The single alignment score is calculated for each contig we find, it applies the local alignment and get the score s. With the length of the contig be l, the single alignment score is:
$single\ alignment\ score = \frac{s}{l}$.

This can be considered as a normalization on the length of the contigs, to avoid the situation: the alignment score for each contig tends to be higher if the length is longer.

MAFFT

Refers to **Multiple alignment program for amino acid or nucleotide sequences**, in order to provide a more intuitive GUI for alignment results: like whether the contigs are connected or can be merged, what are the distribution of contigs on the whole gene sequence.

- Katoh, Rozewicki, Yamada 2019 (*Briefings in Bioinformatics* **20**:1160-1166)
  

  MAFFT online service: multiple sequence alignment, interactive sequence choice and visualization

Global Alignment

The global alignment can be used to identify the covered region of the orginal sequence by the inferred genome. The writer concatenated the contigs and apply the global alignment to concatenated contigs and true genome.

- How to interpret the result of global alignment?
  - we use dynamic programming to realize global alignment, and we can obtain the highest score of the global alignment by reading the right-down-most entry information.
  - we define each hit by +1, each gap be 0, therefore, the highest score can be seen as the coverage (or hit amount) of the inferred sequence to the original sequence. For example, if score is 1000, and the true hiv genome length is 9181, then it means 1000 nucleic acids are covered in the original sequence. But notice this might not be able to represent the single contig quality, since there might have the situation of high score but large gaps, and this cannot be considered as the good inferred contig. That's why we need **Single Alignment Score** mentioned above.
- Why contigs can be concatenated?
  - since the gap penalty for global alignment is 0, therefore, the algorithm will "automatically" find the optimal solution
  - for example, if we have two contigs: ABCD, AB, and the true genome is ABCDE. When we concatenated the contigs together, it will be ABCDAB, we want to verify under this situation, whether optimal alignment ABCD-ABCDE can be identified. And intuitively, it can! Since we set gap penalty be zero, therefore, the algorithm will choose to insert gaps for the subordinary contigs.
- How to deal with large time complexity?
  - as we know, global alignment implemented by dynamic programming is quadratic time complexity, so if the sequences we applied global alignment is long, the time will be large.
  - unfortunately, we will have really long sequence, if we directly concatenate all the contigs: for example, if calculate the contigs with coverage be 4 and graph k be 10, we will get have over 300 contigs with length around 3,000 for each. If we run global alignment on this one, it will take large amount of time.
  - so what we will do is the concatenation process starts on the longest path, and we keep

an eye on the concatenated length, if the concatenated length is larger than 10,000 (remember the true hiv genome is 9181), then we will end the concatenation.

Results

From the above, the writer applied the "grid search" on the mentioned parameters, and the results are reported as the following:

Generated Contigs Files

Can be found in $res/contigs/hiv\_contig\_\{coverage\}\_\{graph\ k\}.txt$

In order to run analysis on MAFFT, the contig files are in the format of "kind of" FASTA format, each ">" identifies a new line for the contig.

Generated Analysis Files

In order to see the performance for different parameters, the data is stored in res/alignment_analysis

Average Contig Length

Coverage	4	4	4	4	4	6	6	6	6	6	8	8	8	8	8	10	10	10	10	10
Graph K	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
Avg Length	3265	2368	407	99	51	3160	8687	1290	132	51	3268	9014	2414	202	51	3360	9110	5101	230	52
Global Score	9181	7106	2442	1898	2298	9181	9181	3872	2253	3187	9181	9014	7244	2893	4051	9181	9181	9181	2766	4805
Local Score	1	0.77	0.26	0.26	0.25	1	1	0.42	0.24	0.34	1	1	0.78	0.31	0.44	1	1	1	0.31	0.53

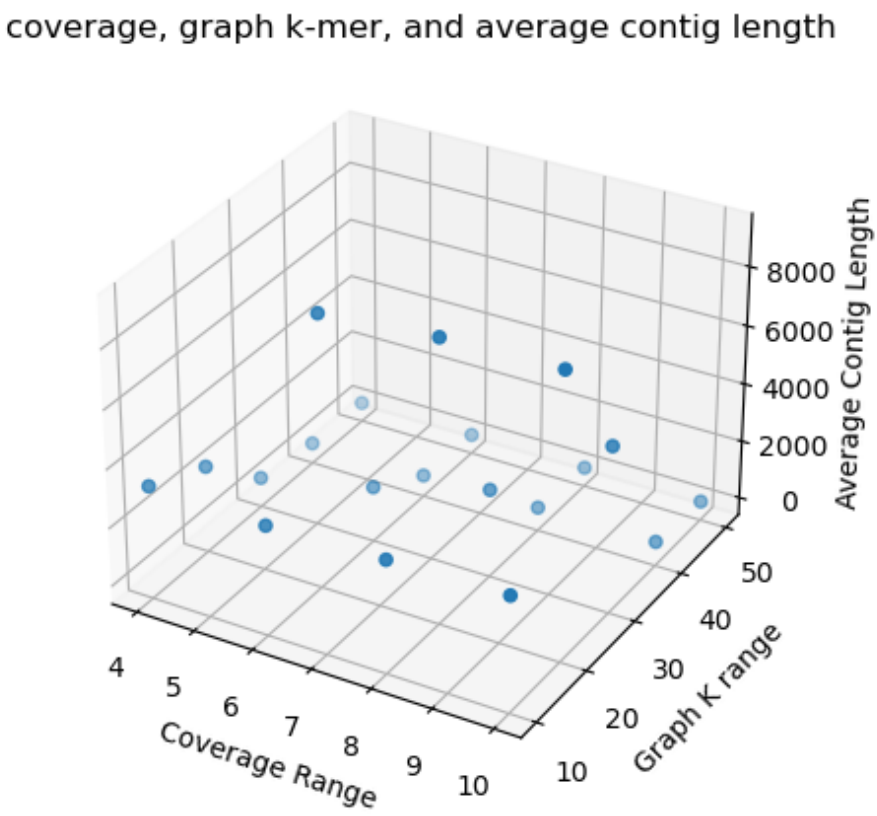
Consider the Lander-Waterman Statistics, we cannot observe the very typical trend for average contig length, which indicates when when the coverage increases, the average contig length should also increase.

The reason why we cannot capture this characteristic here might be related to:

- **definition of contigs**: as we aware, there are many paths in the debruijn graph, and the way we choose the "contig" is longest path in certain percentage + concatenated length smaller than original genome length to reduce time complexity. This process might not be able to select all of the contigs, therefore, the calcualted average length might not be accurate.
- **contig overlap**: in Lander-Waterman Statistics, notice the average contig length is defined by "the last, non-overlapped read", however, here, we include overlapped contigs.
- **extra parameter graph k**: graph k denotes the k-mer length to construct the De Bruijn graph, what the writer did is first sampling corresponding amount of reads by calculating the amount under certain coverage, and then the writer explored the performance of dividing reads into smaller k-mers to construct the graph. This extra parameter might influence the result since the coverage might be changed for the further division.

However, if we observe the data we obtain, we can also observe some interesting trend.

- Average contig length obtains highest when the graph k is equal to 30, and the length is barely influenced by the coverage.
- Graph k can bring larger influence on the average contig length than coverage, this might because of the third reason we mentioned above.
- With fixed k, when coverage increases, the average contigs length will also increase. The increasment value is large at first (from coverage 4-8), smaller later (from coverage 8-10).



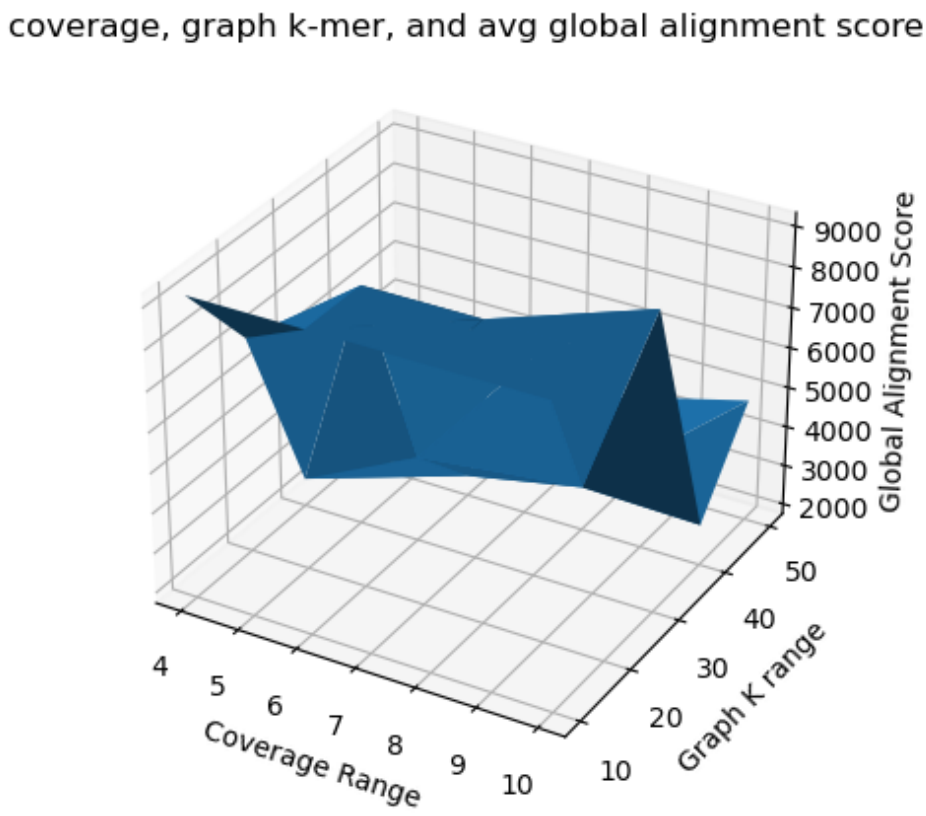
Global Alignment Score

Coverage	4	4	4	4	4	6	6	6	6	6	8	8	8	8	8	10	10	10	10	10
Graph K	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
Avg Length	3265	2368	407	99	51	3160	8687	1290	132	51	3268	9014	2414	202	51	3360	9110	5101	230	52
Global Score	9181	7106	2442	1898	2298	9181	9181	3872	2253	3187	9181	9014	7244	2893	4051	9181	9181	9181	2766	4805
Local Score	1	0.77	0.26	0.26	0.25	1	1	0.42	0.24	0.34	1	1	0.78	0.31	0.44	1	1	1	0.31	0.53

Global alignment score, as we mentioned above, can be used to observe how many "hits" of the inferred sequence to original sequence.

The interesting trend we can observe is:

- when coverage is fixed, when the graph k is increasing, the score will be decreased
- There exist full scores (i.e. 9181), but this might not mean we inferred the correct one, it might also because concatenated contigs are much longer thant the original one.

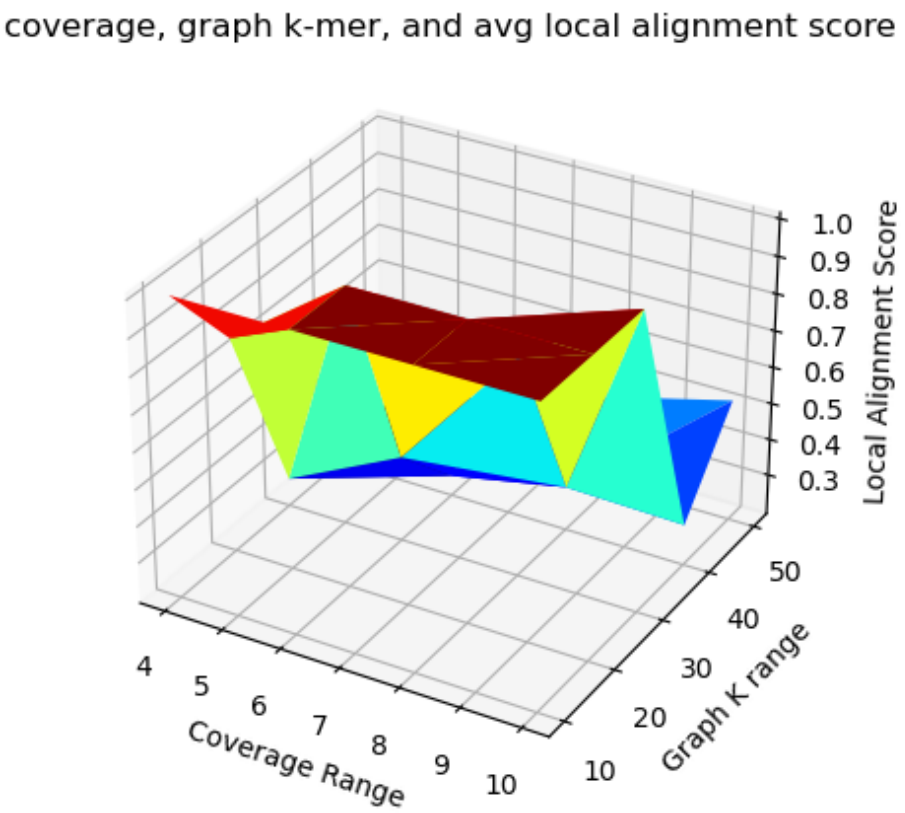


Single Alignment Score

Coverage	4	4	4	4	4	6	6	6	6	6	8	8	8	8	8	10	10	10	10	10
Graph K	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
Avg Length	3265	2368	407	99	51	3160	8687	1290	132	51	3268	9014	2414	202	51	3360	9110	5101	230	52
Global Score	9181	7106	2442	1898	2298	9181	9181	3872	2253	3187	9181	9014	7244	2893	4051	9181	9181	9181	2766	4805
Local Score	1	0.77	0.26	0.26	0.25	1	1	0.42	0.24	0.34	1	1	0.78	0.31	0.44	1	1	1	0.31	0.53

As we mentioned above, single alignment score can be used to observe each contig quality: whether they can align with the original sequence; whether there are a lot of gaps in the alignment. It can fix the limitation of global alignment metrics.

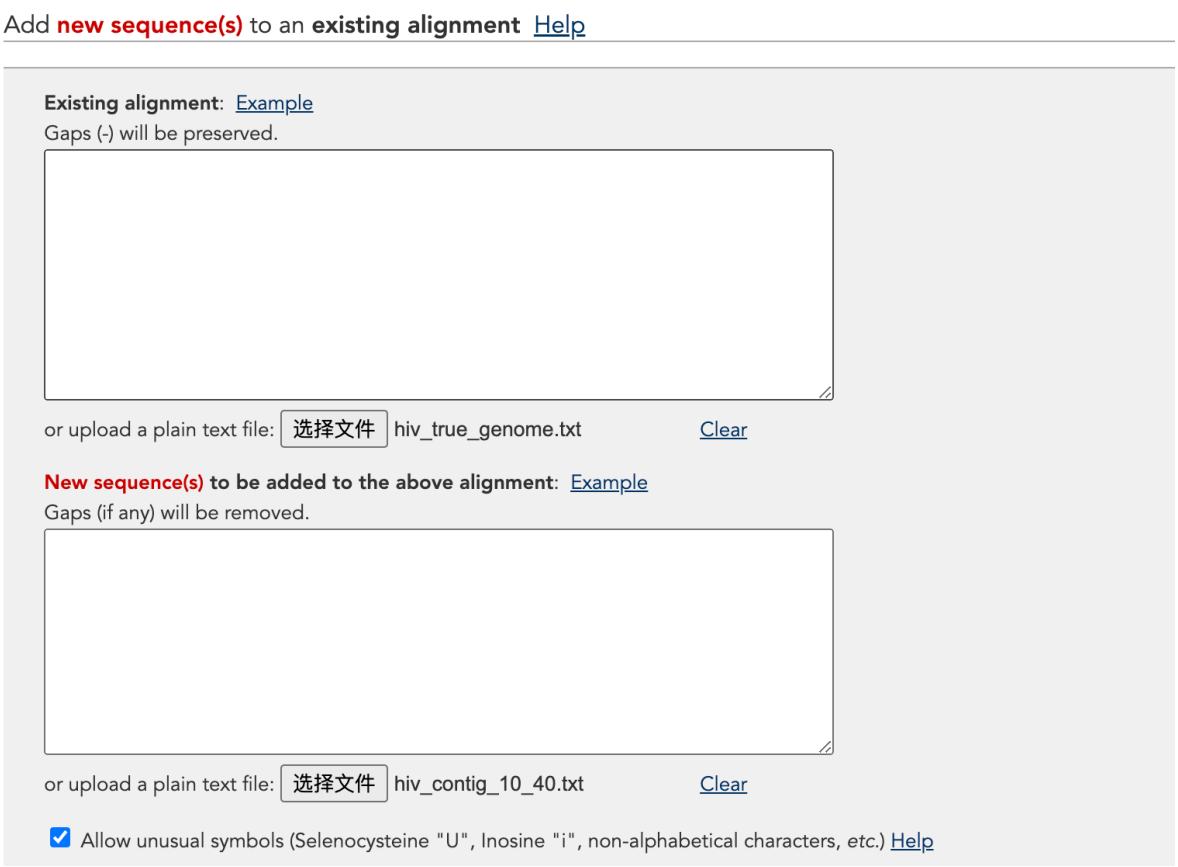
- for example, when coverage = 8, graph k = 30, we notice the average local score for contigs is 0.78, that means the contigs have a lot of "holes" in the "7244" alignment



MAFFT

It is hard to visualize all the multiple sequence alignment result due to space limitation, but you are encouraged to try by yourself with the website: https://mafft.cbrc.jp/alignment/server/add_sequences.html.

- in the existing alignment part upload the "hiv_true_genome.txt" file at $res/true\_reads/hiv\_true\_reads.txt$
- in the new sequences part upload the inferred contig file at $res/contigs/hiv\_contig\_\{coverage\}\_\{graph\ k\}.txt$
- after analysis, click view buttom



Here, we choose some representative result and try to visualize them, click the URL and check the results (**let me know if the URL is not working cause I do not know how long will the server keep the result**):

- Relatively Success Exm: Coverage = 8, Graph K = 20, Avg = 9014, Global Score = 9014, Local Score = 1
    - the result can be found in link https://mafft.cbrc.jp/msaviewer/?LoadURL=/alignment/server/spool/_out.2204060558257905ebBJxmowLvqCNHin1Is6lsfnormal.pir&Pos=997,0,39&ColorScheme=MAFFT+Nucleotide#
    - this result seems quite successful, the inferred contigs merged into one (or several) long contigs. All the contigs have relatively long length, and they covered basically all of the region (that's to say they are overlapped) but with certain contigs have more "hits" than others.
- Relatively Failure Exm: Coverage = 4, Graph K = 30, Avg = 407, Global Score = 2442, Local Score = 0.26
    - the result can be found in link https://mafft.cbrc.jp/msaviewer/?LoadURL=/alignment/server/spool/_out.220406060914146dPMpKShNERg0avWOq4o2ulsfnormal.pir&Pos=3513,0,96&ColorScheme=MAFFT+Nucleotide
    - as you can tell from the result, the contigs only cover a small range of the original sequence (which corresponds to the low global alignment score), and even the covered part, there are a lot of gaps (which corresponds to the low local alignment score).

(57.14% Consensus)

Sample Reads Assembly

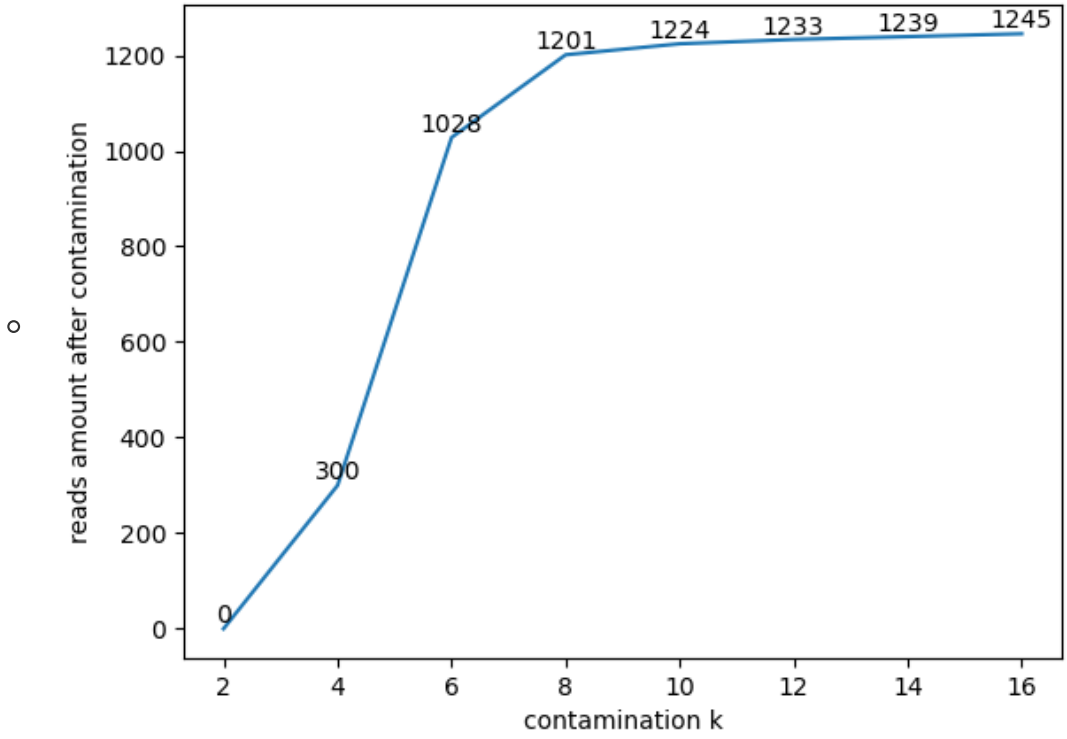
Work Flow

Hyper-parameters

Since the setup for HIV and SampleRead is similar: same contamination vector and sequencing error rate, therefore, in this part, the writer directly use the hyper-parameters from the HIV part. However, notice the experiements are focused on different algorithms implemented in PR2.

The used hyper-parameters are:

- CONTAMINATION_k = 8: Consider 10% contaminated reads: 1274 - 127 = 1147



- CORRECTION_k = 17
- CORRECTION_t = 4

- CORRECTION_d = 2
- GRAPH_k = [10, 20, 30, 40, 50]

Correction Algorithms

Refer to the file $correction\_algorithm.pdf$ to refresh the implemented algorithms.

In this experiment, we tested four correction algorithms: Stack Correction, Naive Correction, Simple Correction, Merge Correction.

Process

This part is more like a combination of the previous implementation.

1. Contamination: according to the given vector and reads file, delete the contaminated reads with given contamination threshold k.
2. Correction: experiements applied on two methods, finally choose method 1
    1. **Original correction with four algorithms**: stack correction, merge correction, simple correction naive correction
    2. **Consider frequency of vector with four algorithms**: with the four correction methods metioned above, add weights to the obtained k-mers. The weights comes from the vector k-mer frequency. Specifically, for the k-mers obtained in correction, if the k-mer is also in the vector k-mers, then it is more likely to be the contanminated, use certain weight multiplies the frequency to decrease the frequency make the k-mer more infrequent. The weight can be obtained by applying **softmax** to the vector k-mer frequency, which returns the weights in the range of 0 to 1 and also $P(weight\ for\ all\ k-mer) = 1$.
    3. Notice though the latter one seems to be a more considerate algorithm, the test results do not show large differences between these two algorithms. This might because the second algorithm aims to fix the case when contamination and sequencing errors to overlap. However, this kind of situation is rare, thus makes little difference.
3. De Bruijn Graph Assembly
    1. Choose graph k, further dividing the reads to graph k length k-mers. Mentioned by the article *How to apply de Bruijn graphs to genome assembly*, graph k choice is related to the k-mer representation coverage, gaps number and repeats number.
    2. Obtain contigs with the algorithm described in Section **How many contigs need to be reserved**.
4. Result Analysis with metrics defined following.

Metrics

Connectivity

Unlike the HIV genome, we have the true reads, for sample reads, since we do not have the groundtruth, what we can see as a metric is connectivity. If there are less isolated nodes, then the quality might be higher.

Graph K		10	20	30	40	50
Stack Correction	# of isolates	0	0	0	3	11
	# of nodes	63	47	47	44	36
Merge Correction	# of isolates	0	0	7	39	139
	# of nodes	1505	1336	1341	1294	998
Simple Correction	# of isolates	0	0	3	4	11
	# of nodes	63	47	44	44	36
Naive Correction	# of isolates	0	0	0	3	11
	# of nodes	77	61	61	58	50

"Cheating" Metrics: BLAST

The writer has the assumption the sample reads are obtained from the existed sequence, therefore, as a cheating method, the writer input the inferred sequence in BLAST and try to find whether there exists high score alignment.

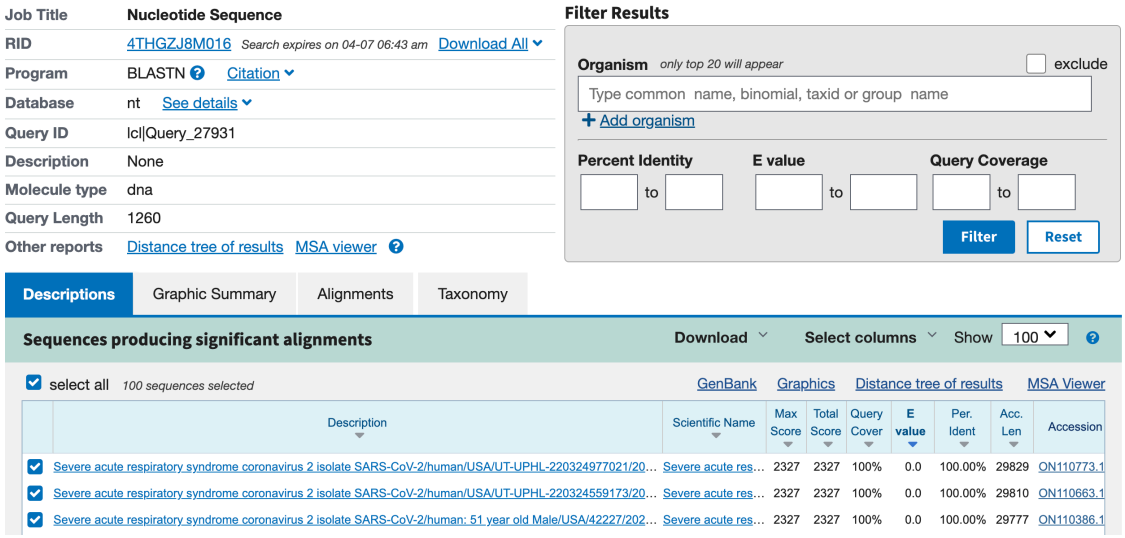
Specifically, we choose the graph K gives us the least number of isolates, here we choose graph K be 20 (cause it is the optimal parameter from HIV and also it gives us 0 isolates).

The inferred sequence results of different correction algorithm are tested on the BLAST and the outputs are as following:

- Stack Correction
  - Result:
    ATGTCTGATAATGGACCCCAAAATCAGCGAAATGCACCCCGCATTACGTTTGGTGGACCCTCAG
    ATTCAACTGGCAGTAACCAGAATGGAGAACGCAGTGGGGCGCGATCAAAACAACGTCGGCCC
    CAAGGTTTACCCAATAATACTGCGTCTTGGTTCACCGCTCTCACTCAACATGGCAAGGAAGACC
    TTAAATTCCCTCGAGGACAAGGCGTTCCAATTAACACCAATAGCAGTCCAGATGACCAAATTGG
    CTACTACCGAAGAGCTACCAGACGAATTCGTGGTGGTGACGGTAAAATGAAAGATCTCAGTCC
    AAGATGGTATTTCTACTACCTAGGAACTGGGCCAGAAGCTGGACTTCCCTATGGTGCTAACAAA
    GACGGCATCATATGGGTTGCAACTGAGGGAGCCTTGAATACACCAAAAGATCACATTGGCACC
    CGCAATCCTGCTAACAATGCTGCAATCGTGCTACAACTTCCTCAAGGAACAACATTGCCAAAAG
    GCTTCTACGCAGAAGGGAGCAGAGGCGGCAGTCAAGCCTCTTCTCGTTCCTCATCACGTAGTC
    GCAACAGTTCAAGAAATTCAACTCCAGGCAGCAGTAGGGGAACTTCTCCTGCTAGAATGGCTG
    GCAATGGCGGTGATGCTGCTCTTGCTTTGCTGCTGCTTGACAGATTGAACCAGCTTGAGAGCA
    AAATGTCTGGTAAAGGCCAACAACAACAAGGCCAAACTGTCACTAAGAAATCTGCTGCTGAGG
    CTTCTAAGAAGCCTCGGCAAAAACGTACTGCCACTAAAGCATACAATGTAACACAAGCTTTCGG

CAGACGTGGTCCAGAACAAACCCAAGGAAATTTTGGGGACCAGGAACTAATCAGACAAGGAAC
TGATTACAAACATTGGCCGCAAATTGCACAATTTGCCCCCAGCGCTTCAGCGTTCTTCGGAATG
TCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTGGTTGACCTACACAGGTGCCATCAAA
TTGGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGCTGAATAAGCATATTGACGCATA
CAAAACATTCCCACCAACAGAGCCTAAAAAGGACAAAAAGAAGAAGGCTGATGAAACTCAAGC
CTTACCGCAGAGACAGAAGAAACAGCAAACTGTGACTCTTCTTCCTGCTGCAGATTTGGATGAT
TTCTCCAAACAATTGCAACAATCCATGAGCAGTGCTGACTCAACTCAGGCCTAA

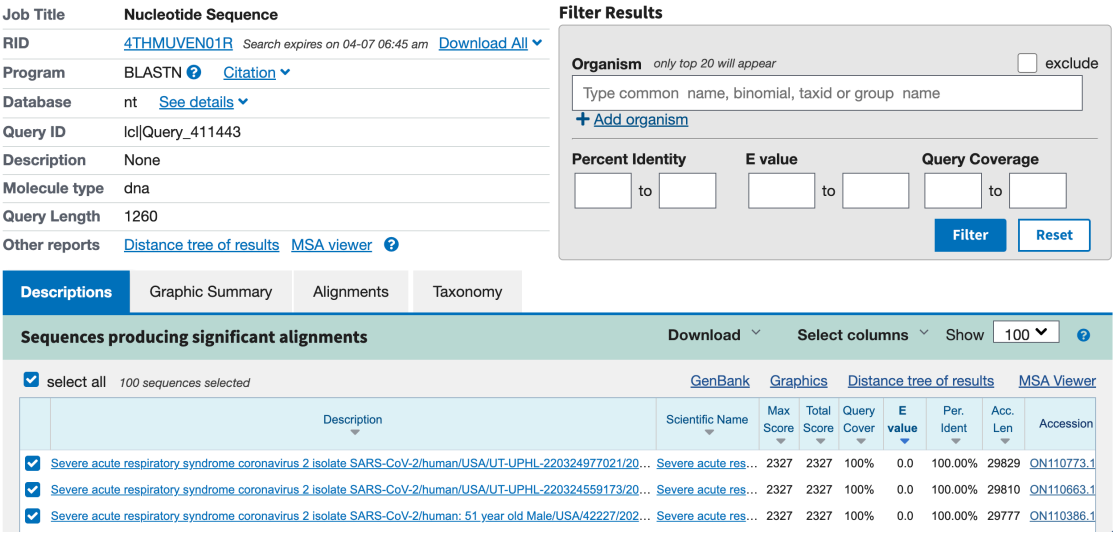
- BLAST result:



- Naive Correction
  - Result:

ATGTCTGATAATGGACCCCAAAATCAGCGAAATGCACCCCGCATTACGTTTGGTGGACCCTCAG
ATTCAACTGGCAGTAACCAGAATGGAGAACGCAGTGGGGCGCGATCAAAACAACGTCGGCCC
CAAGGTTTACCCAATAATACTGCGTCTTGGTTCACCGCTCTCACTCAACATGGCAAGGAAGACC
TTAAATTCCCTCGAGGACAAGGCGTTCCAATTAACACCAATAGCAGTCCAGATGACCAAATTGG
CTACTACCGAAGAGCTACCAGACGAATTCGTGGTGGTGACGGTAAAATGAAAGATCTCAGTCC
AAGATGGTATTTCTACTACCTAGGAACTGGGCCAGAAGCTGGACTTCCCTATGGTGCTAACAAA
GACGGCATCATATGGGTTGCAACTGAGGGAGCCTTGAATACACCAAAAGATCACATTGGCACC
CGCAATCCTGCTAACAATGCTGCAATCGTGCTACAACTTCCTCAAGGAACAACATTGCCAAAAG
GCTTCTACGCAGAAGGGAGCAGAGGCGGCAGTCAAGCCTCTTCTCGTTCCTCATCACGTAGTC
GCAACAGTTCAAGAAATTCAACTCCAGGCAGCAGTAGGGGAACTTCTCCTGCTAGAATGGCTG
GCAATGGCGGTGATGCTGCTCTTGCTTTGCTGCTGCTTGACAGATTGAACCAGCTTGAGAGCA
AAATGTCTGGTAAAGGCCAACAACAACAAGGCCAAACTGTCACTAAGAAATCTGCTGCTGAGG
CTTCTAAGAAGCCTCGGCAAAAACGTACTGCCACTAAAGCATACAATGTAACACAAGCTTTCGG
CAGACGTGGTCCAGAACAAACCCAAGGAAATTTTGGGGACCAGGAACTAATCAGACAAGGAAC
TGATTACAAACATTGGCCGCAAATTGCACAATTTGCCCCCAGCGCTTCAGCGTTCTTCGGAATG
TCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTGGTTGACCTACACAGGTGCCATCAAA
TTGGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGCTGAATAAGCATATTGACGCATA
CAAAACATTCCCACCAACAGAGCCTAAAAAGGACAAAAAGAAGAAGGCTGATGAAACTCAAGC
CTTACCGCAGAGACAGAAGAAACAGCAAACTGTGACTCTTCTTCCTGCTGCAGATTTGGATGAT
TTCTCCAAACAATTGCAACAATCCATGAGCAGTGCTGACTCAACTCAGGCCTAA

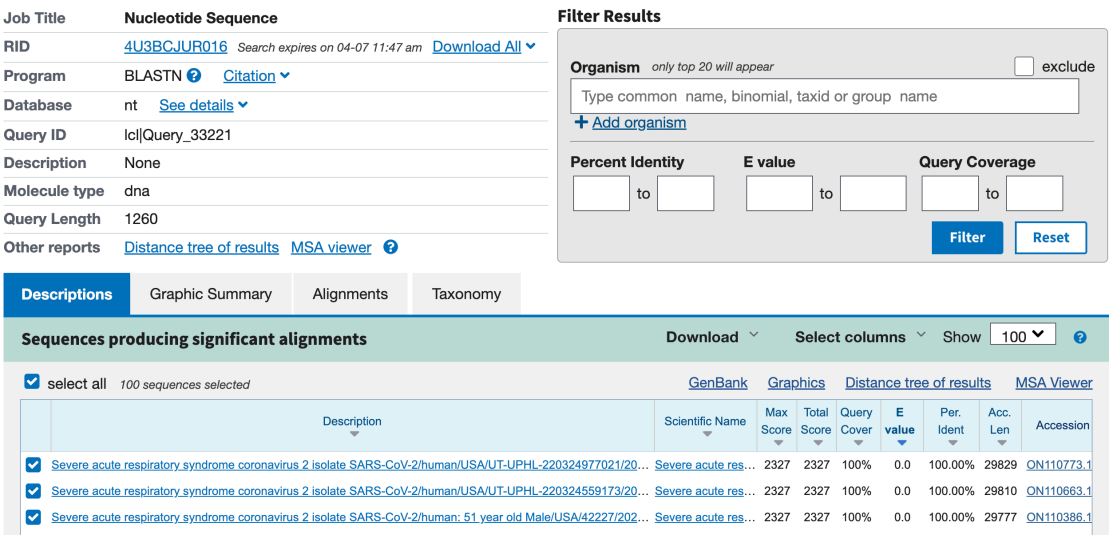
  - BLAST Result



- Simple Correction
  - Result:

    ATGTCTGATAATGGACCCCAAAATCAGCGAAATGCACCCCGCATTACGTTTGGTGGACCCTCAG
    ATTCAACTGGCAGTAACCAGAATGGAGAACGCAGTGGGGCGCGATCAAAACAACGTCGGCCC
    CAAGGTTTACCCAATAATACTGCGTCTTGGTTCACCGCTCTCACTCAACATGGCAAGGAAGACC
    TTAAATTCCCTCGAGGACAAGGCGTTCCAATTAACACCAATAGCAGTCCAGATGACCAAATTGG
    CTACTACCGAAGAGCTACCAGACGAATTCGTGGTGGTGACGGTAAAATGAAAGATCTCAGTCC
    AAGATGGTATTTCTACTACCTAGGAACTGGGCCAGAAGCTGGACTTCCCTATGGTGCTAACAAA
    GACGGCATCATATGGGTTGCAACTGAGGGAGCCTTGAATACACCAAAAGATCACATTGGCACC
    CGCAATCCTGCTAACAATGCTGCAATCGTGCTACAACTTCCTCAAGGAACAACATTGCCAAAAG
    GCTTCTACGCAGAAGGGAGCAGAGGCGGCAGTCAAGCCTCTTCTCGTTCCTCATCACGTAGTC
    GCAACAGTTCAAGAAATTCAACTCCAGGCAGCAGTAGGGGAACTTCTCCTGCTAGAATGGCTG
    GCAATGGCGGTGATGCTGCTCTTGCTTTGCTGCTGCTTGACAGATTGAACCAGCTTGAGAGCA
    AAATGTCTGGTAAAGGCCAACAACAACAAGGCCAAACTGTCACTAAGAAATCTGCTGCTGAGG
    CTTCTAAGAAGCCTCGGCAAAAACGTACTGCCACTAAAGCATACAATGTAACACAAGCTTTCGG
    CAGACGTGGTCCAGAACAAACCCAAGGAAATTTTGGGGACCAGGAACTAATCAGACAAGGAAC
    TGATTACAAACATTGGCCGCAAATTGCACAATTTGCCCCCAGCGCTTCAGCGTTCTTCGGAATG
    TCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTGGTTGACCTACACAGGTGCCATCAAA
    TTGGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGCTGAATAAGCATATTGACGCATA
    CAAAACATTCCCACCAACAGAGCCTAAAAAGGACAAAAAGAAGAAGGCTGATGAAACTCAAGC
    CTTACCGCAGAGACAGAAGAAACAGCAAACTGTGACTCTTCTTCCTGCTGCAGATTTGGATGAT
    TTCTCCAAACAATTGCAACAATCCATGAGCAGTGCTGACTCAACTCAGGCCTAA

  - BLAST Result



Job Title	Nucleotide Sequence
RID	4U3BCJUR016 *Search expires on 04-07 11:47 am* Download All
Program	BLASTN Citation
Database	nt See details
Query ID	lcl|Query_33221
Description	None
Molecule type	dna
Query Length	1260
Other reports	Distance tree of results MSA viewer

Filter Results

Organism *only top 20 will appear* exclude

Type common name, binomial, taxid or group name

Add organism

Percent Identity to

E value to

Query Coverage to

Filter Reset

Descriptions | Graphic Summary | Alignments | Taxonomy

Sequences producing significant alignments Download Select columns Show 100

select all *100 sequences selected* GenBank Graphics Distance tree of results MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/USA/UT-UPHL-220324977021/20…	Severe acute res…	2327	2327	100%	0.0	100.00%	29829	ON110773.1
Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/USA/UT-UPHL-220324559173/20…	Severe acute res…	2327	2327	100%	0.0	100.00%	29810	ON110663.1
Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human: 51 year old Male/USA/42227/202…	Severe acute res…	2327	2327	100%	0.0	100.00%	29777	ON110386.1

- Merge Correction
  - ATGTCTGATAATGGACCCCAAAATCAGCGAAGTGCACCCCGCATTACGTTTGGTGGACGCCCA GATTCAACTGGCAGTAACCAGAATGGAGAACGCAGTGGGGCGCGATCAAAACAACGTCGGCC CCAAGGTGTACCCAATAATACTGCGTCTTGGTTCACCGTTCTCACTCAACATGGCAAGGAACAC CTTAAATTCCCTCGAGGACAAGGCGTTCCAATTAATACCAATAGCAGTCCAGATGACCGAATTG GCTACTACCGAAGAGCTACCAGACGAATTCCTGGTGGTGACGGTAAAATGAAAGATCTCAGTC CAAGAGGGTATTTCTACTACCTAGGAACTGGGCCAGAAGCTGCACTTCCCTATGGTGCTAACAA AGACGGCATCATATGGTTTGCAACTGAGGGAGCCTTGAATACACCAAAAGATCACATTGGCAC CCGCAAGCCTGCTAACAATGCTGCAATCGTGCTACAACTTCCTCAAGGAACAACATTGCCAAAA TGCTTCTACGCAGAAGGGAGCAGAAGCGACAGTCAAGCCTCTTCTCGTTCTTCATCACGTAGTC GCAACAGTTCTAGAAATTCAACTCCAGGCATCAGTAGGGGAACTTCTCCTGATAGAATGGCTG GCAATGGCGGTGATGCTGCTGTTGCTTTGCTGCTGCTTGACAGATAGAACCAGCTTGAGAGCA AAATGTCTCGTAAAGGCCAACAACAACAAGGACAAACTGTCACTAAGAAATCTCCTGCTGAGGC TTCTAAGAAGCCTCGGCAAAAACGTACTGACACTAAAGCATACAATGTAACGCAAGCTTTCGGC AGACGTGGTCCAGAACAAACCCATGGAAATTTTGGGGACCAGGAACTAATCAGACAAGGAACT GATTACAAACATTGGCCGCAAATTGCAAAATTTGCCCCCAGCGCTTCAGCGTTCTTGGGAATGT CGCGCATTGGCATGCAAGTCACACCTTCGGGAACGTTGTTGACCTACACAGGTGCCATCCAATT GGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGCTCAATAAGCATATTGACGCATACA AAACATTTCCACCAACAGAGCCTAAAAAGGACAAAAAGAAGAAGGCTGATGAAACTCAAGCCT TACCGCAGAGACAGAAGAAACAGCAAACTGTGACTCTTCTTACTGCTGCAGATTTGGATGATTT CTCCAAACAATTGCAAGAATCCATGAGCAGTGCTGACTCAACTCAGGCCTAA
  - BLAST Result

Job Title	**Nucleotide Sequence**
RID	4WBE3E56016 *Search expires on 04-08 08:18 am* Download All
Program	BLASTN Citation
Database	nt See details
Query ID	lcl|Query_28165
Description	None
Molecule type	dna
Query Length	1260
Other reports	Distance tree of results MSA viewer

Filter Results

Organism *only top 20 will appear* exclude

Type common name, binomial, taxid or group name

+ Add organism

Percent Identity to **E value** to **Query Coverage** to

Filter Reset

Descriptions | Graphic Summary | Alignments | Taxonomy

Sequences producing significant alignments Download Select columns Show 100

select all *100 sequences selected* GenBank Graphics Distance tree of results MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/USA/PA-UW21020343623/2021 O…	Severe acute res…	2128	2128	100%	0.0	97.14%	29540	OM897851.1
Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/USA/FL-CDC-ASC210822244/2022…	Severe acute res…	2128	2128	100%	0.0	97.14%	29801	OM890605.1
Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/SouthAfrica/NHLS-UCT-GS-0058/2…	Severe acute res…	2128	2128	100%	0.0	97.14%	29815	OM725346.1