CHAPTER 01

BLAST

The inventors of the BLAST (Basic Local Alignment Search Tool) computer program should win the Nobel Prize in Physiology or Medicine. I'm serious! Actually, they should probably share it with the inventors of the FASTA algorithm, which preceded the more efficient and flexible BLAST algorithm.

Why should the makers of a computer program that finds the best database match to a biological sequence win the most prestigious award in biomedical science? The Nobel Prize is usually reserved for discoveries made in the laboratory, such as the discovery of novel disease organisms, how to genetically engineer mice, proteins that glow in the dark, the mechanism of cell death, that sort of thing. A computer program seems a paltry gimmick in comparison, and biologists often chuckle or even scoff when I suggest that it deserves the Nobel. Determining the best match of a biological sequence appears rather uninspiring compared with the discovery of antibiotics or proteins that act like viruses. Yet in an era of rapidly expanding DNA sequencing capability and big biological data, BLAST's simplicity, elegance, and efficiency make this algorithm the most powerful biomedical discovery tool in the history of science.

BLAST It

Researchers use BLAST so often that it has been turned into a verb. (Like "to google" or "to chill.") "Hey, did you BLAST that sequence I gave you yet? No? Well, what are you waiting for? BLAST it!" To understand why BLAST is so useful and popular, imagine that you work in a molecular biology lab studying a novel infectious disease and you just received the results of your very first sequencing run, perhaps the first genetic information in the history of this unknown organism. Unfortunately, the sequence looks like this[1]:

```
>FX093345
TGCAGTCGATCATCAGCATACCTAGGTTTCGTCCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTTGTTC
TTGGTGTCAACGAGAAAACACACGTCCAACTCAGTTTGCCTGTCCTTCAGGTTAGAGACGTGCTAGTGCG
TGGCTTCGGGGACTCTGTGGAAGAGGCCCTATCGGAGGCACGTGAACACCTCAAAAATGGCACTTGTGGT
CTAGTAGAGCTGGAAAAAGGCGTACTGCCCCAGCTTGAACAGCCCTATGTGTTCATTAAACGTTCTGATG
CCTTAAGCACCAATCACGGCCACAAGGTCGTTGAGCTGGTTGCAGAAATGGACGGCATTCAGTACGGTCG
TAGCGGTATAACACTGGGAGTACTCGTGCCACATGTGGGCGAAACCCCAATTGCATACCGCAATGTTCTT
CTTCGTAAGAACGGTAATAAGGGAGCCGGTGGTCATAGCTATGGCATCGATCTAAAGTCTTATGACTTAG
```

That's some exciting research you have there! Well, maybe, but what exactly is it? Clearly, it is a DNA sequence of some sort, but is it the secret to uncovering a deadly new pathogen, or is it human DNA contamination from the technician at the sequencing facility? Is it part of a protein toxin that allows a deadly microbe to destroy epithelial cells in the lungs, or is it a regulatory gene sequence from a housefly that landed on your pipette tip when you weren't looking?

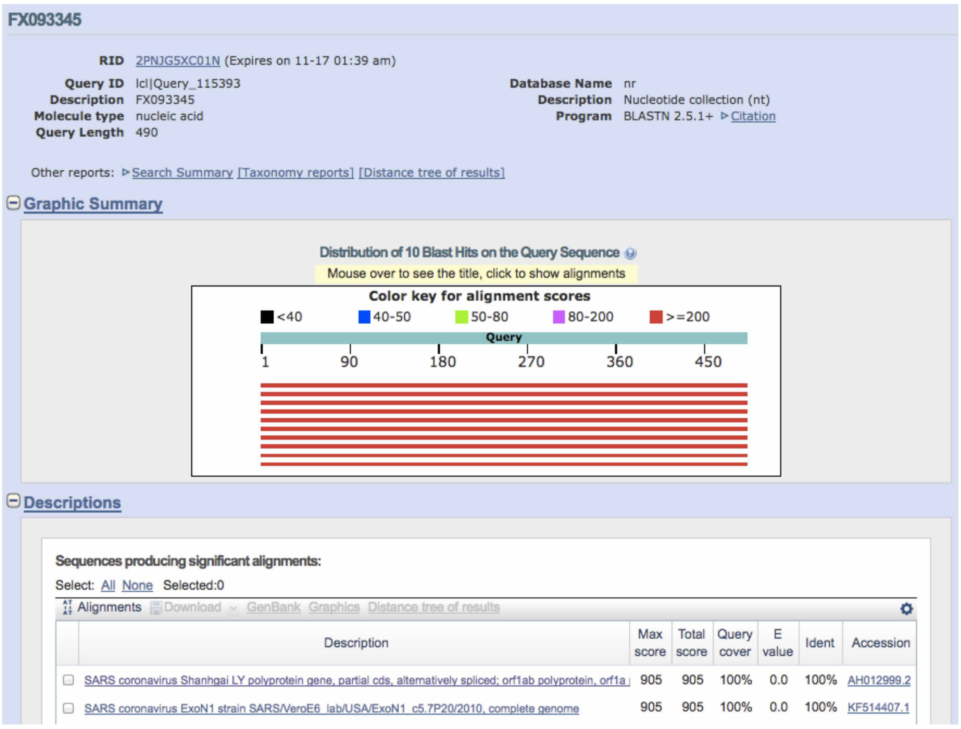
Eyeballing a bunch of letters, the same four letters (A, G, T, and C) repeated in different orders, will get you nowhere. Wouldn't it be great if, without pilfering a pipette or pouring a petri plate, you could answer the following questions?

- Is the genetic material from a bacterium or a virus?
- Did you accidentally sequence a contaminating organism?
- Does the sequence code for a protein?
- What is the biological function of the sequence?
- Is the genetic material related to other, possibly well-characterized, pathogens?
- Is the sequence new to science?

Using BLAST, you can answer most of these questions in milliseconds. That's how long it takes for BLAST and a bunch of remote supercomputers at the National Center for Biotechnology Information (NCBI) to match your sequence to every DNA sequence ever catalogued. Figure 1.1 shows the BLAST results for your hypothetical new pathogen. In under a second, you discovered (i) the nature of the organism, namely, a known deadly virus of the coronavirus (a common cold virus) family, (ii) that your sequence encodes a protein, and (iii) that the protein is part of the virus's outer shell, a critical part of the infectious process. Not bad for a few seconds of effort. (Your tax dollars at work!)

Scaling Up: Massive Parallelization of BLAST

Imagine that instead of just one mystery sequence you had 10, 100, or 1,000 sequences and you had to match them against databases containing millions of sequences. Furthermore, these sequences encompass multiple different organisms, are part of different molecular processes, and are involved in a variety of cellular functions. Now you begin to see the scale of the problem and why bioinformatics can be so darn useful. By running BLAST searches in parallel on a suite of high-performance computers, it is possible to analyze thousands of mystery sequences against millions of known sequences generated by researchers all over the world (Fig. 1.2).



FX093345

RID 2PNJG5XC01N (Expires on 11-17 01:39 am)

Query ID	lcl|Query_115393	Database Name	nr
Description	FX093345	Description	Nucleotide collection (nt)
Molecule type	nucleic acid	Program	BLASTN 2.5.1+ ▷ Citation
Query Length	490		

Other reports: ▷ Search Summary [Taxonomy reports] [Distance tree of results]

Graphic Summary

Distribution of 10 Blast Hits on the Query Sequence

Mouse over to see the title, click to show alignments

Color key for alignment scores

■ <40 ■ 40-50 ■ 50-80 ■ 80-200 ■ >=200

Descriptions

Sequences producing significant alignments:

Select: All None Selected:0

	Description	Max score	Total score	Query cover	E value	Ident	Accession
☐	SARS coronavirus Shanhgai LY polyprotein gene, partial cds, alternatively spliced; orf1ab polyprotein, orf1a	905	905	100%	0.0	100%	AH012999.2
☐	SARS coronavirus ExoN1 strain SARS/VeroE6_lab/USA/ExoN1_c5.7P20/2010, complete genome	905	905	100%	0.0	100%	KF514407.1

FIGURE 1.1. Results of a BLAST nucleotide search with the FX093345 nucleotide sequence. The results, returned in under a second, reveal that the sequence is an exact match to the genome of a severe acute respiratory syndrome (SARS) coronavirus protein first isolated in Shanghai (apparently misspelled "Shanhgai" in the database), China. The first 50 matches were to SARS coronaviruses isolated at different times, which provides confidence that we have indeed isolated a SARS virus. At the time this BLAST analysis was performed, the sequence was a perfect 100% match (bottom right under the "Ident" column) to more than one sequence, so we cannot say with certainty that it is the Shanghai strain. The fact that so many were identical also means that the results appear in a random order. That, and the fact that databases are constantly updated with new sequences, means that another search with the sequence may return different results from those seen in the figure. One of the most important considerations for interpreting BLAST results is the E-value. The E-value is the Expectation value, which tells the user how likely the search similarity result is due to random chance. An E-value of 1e-4 is the same as 0.0001, which indicates this similarity would occur 1 in 10,000 times by chance. Note, an E-value of 0.0 signifies a likelihood below 1e-250.

Why a Nobel Prize?

The real power of bioinformatics, especially BLAST, is twofold. First, bioinformatics methods dramatically amplify experimental results. Second, bioinformatics methods generate new testable hypotheses and targets for further discovery. To understand how BLAST amplifies biological knowledge, one must first have an appreciation of just how much in terms of time, energy, and resources it takes to make discoveries using the tools of molecular biology. Without basic information on the genetics and biochemical function of the genes or gene regions gleaned from laboratory experimentation, BLAST matches would not be particularly

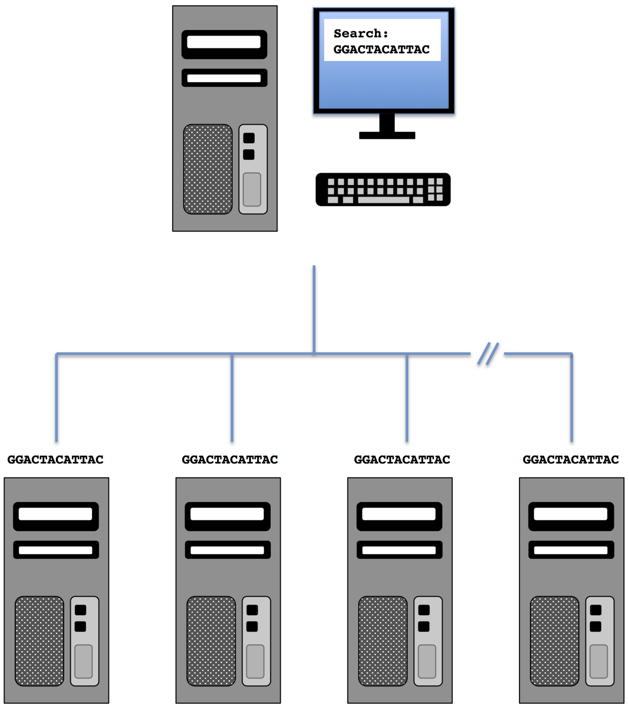


FIGURE 1.2. Parallel BLAST searches. Each "Query" sequence is simultaneously matched to hundreds or thousands of sequence databases using the BLAST algorithm. The scores for the best matches are retrieved and ranked.

useful. As an example, Fig. 1.3 shows some of the standard experimental approaches needed to determine the function of a protein-coding gene found in a bacterial genome.

While the cost of characterizing even a single protein-coding gene can be significant in terms of both time and money, once we have this characterization and this sequence in a database, algorithms like BLAST become especially powerful. Figure 1.3B illustrates how BLAST and DNA sequencing technology can amplify biological knowledge.

In the example, BLAST is really amplifying the experimental knowledge by allowing us to infer the function of 20 different previously unknown DNA sequences through a rapid sequence alignment. The process can also go in the other direction. For instance, we could use BLAST to discover if a DNA sequence that codes for a gene of unknown function is present in every living organism ever sequenced. However, should an experimentalist decide to target the protein product of this gene of unknown function for analysis and figure out its molecular function, we would suddenly know the function of this gene in millions of species.

Given the rate of sequence generation, it is safe to say that BLAST and other bioinformatics methods are the primary source of information for 99.9% of all newly sequenced DNA. This has made BLAST instrumental for discovering, among other things, the following:

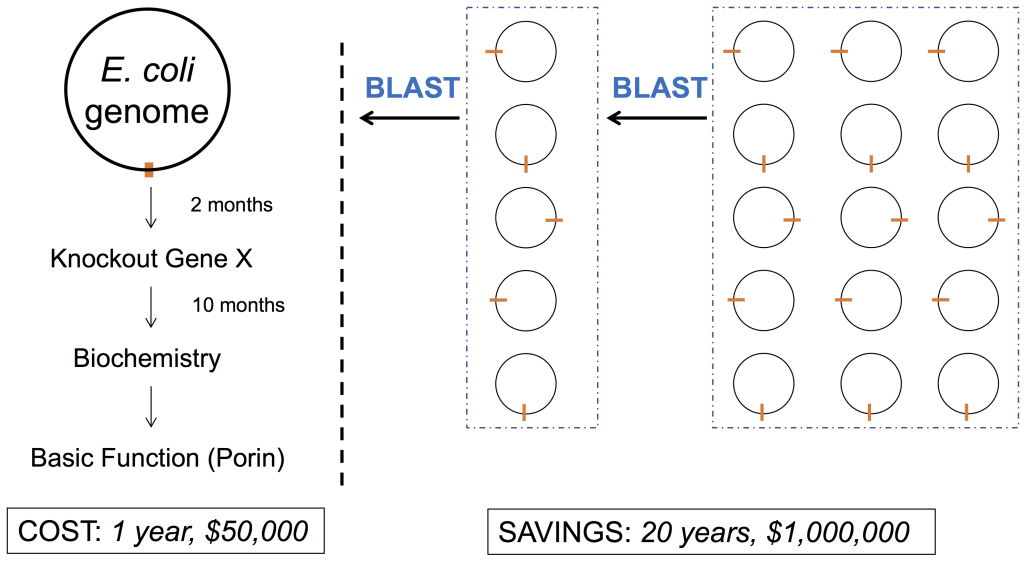


FIGURE 1.3. Characterizing the function of a novel bacterial protein encoded in the *Escherichia coli* genome and then using BLAST to find similar functions in newly sequenced genomes. (Left) Flowchart of approaches necessary to characterize a novel protein in the *E. coli* genome. Most bacteria have one copy of each gene on a single circular chromosome.[2] Each of the flowchart steps represents weeks or months of painstaking work. Typically, this process may take a year and cost on the order of $50,000. **(Right)** Once a gene has been characterized, sequences from other bacterial genomes (circles) can be quickly matched to this sequence using BLAST. The figure shows BLAST identifying porin proteins in 20 different bacterial genomes. When the matches are sufficiently strong, we can infer that these proteins have functions highly similar to that of the original, which saves the need to experimentally characterize this protein from all 20 genomes.

- Novel pathogens, both animal and human
- Life in extreme environments
- Mechanisms of genetic diseases
- Novel protein functions
- Novel cellular processes

If this doesn't warrant a Nobel Prize in Medicine, frankly, I don't know what does.

Notes

1. Note the FASTA format of the sequence, the program's enduring legacy to bioinformatics.
2. In the cell, the genomic DNA is usually tightly wound in a supercoiled state.

ACTIVITY 1.1 BLAST ALGORITHM

Motivation

The purpose of this activity is to teach the basic concepts behind the BLAST algorithm and how to use a web-based implementation of this algorithm to analyze DNA and protein sequence data. BLAST (Basic Local Alignment Search Tool) is a fast computational method for making sequence alignments. Sequence alignments are a critical part of bioinformatics. Computational methods for making pairwise alignments of biological molecules (DNA, RNA, or protein) were some of the very first bioinformatics algorithms developed. Among other things, sequence alignments allow researchers to determine the organisms from which the molecule came (human, oyster, pine tree, bacterium, etc.) and predict the cellular function of biological molecules based only on their sequence. For example, BLAST can report with high confidence that the protein sequence YNFGSGSAYGGSFGGVDGLLAGGEKATMQNL is keratin from the domestic dog hair found on your sofa.

BLAST was created to speed up the process of making sequence alignments. Full pairwise sequence alignment methods (see Chapter 03) are too computationally intensive to handle the alignment of thousands or millions of sequences. BLAST speeds up this process by "chopping up" an input sequence into smaller bits and matching these smaller bits to millions of different sequences. The algorithm then attempts to extend the sequence alignment to make a full alignment. Then the algorithm ranks the sequence alignments, and the longest alignment with the fewest mismatches wins! In bioinformatics parlance we call the good matches "hits," and the best ones are "best hits" or "top hits." BLAST is a heuristic method, meaning that it is not guaranteed to find the optimal alignment, but it is much faster than more stringent approaches.

Learning Objectives

1. Know the basic purpose and utility of the BLAST computational method (Motivation).
2. Understand the concepts behind the BLAST algorithm (Concepts and Exercises).
3. Correctly solve sequence-matching problems based on the BLAST algorithm (Concepts and Exercises).
4. Learn how to use NCBI's BLAST web-based sequence analysis website and be able to correctly interpret its output (Concepts and Exercises).

Concepts

As mentioned above, the purpose of the BLAST algorithm is to find the best hit (highest-scoring match) of an unknown DNA or protein sequence in a database. The method has been so successful because of its clever simplicity. In order to better grasp the method behind the algorithm, try the preparatory exercise. Using your brain and a pencil, try to find regions (local alignments) that best match the following DNA sequence and protein sequences. First, try to match the Query DNA sequence to the Sbjct[1] (Subject) DNA sequence. Then do the same for the protein Query and Sbjct sequences. Find the regions of best alignment between the two, and keep in mind that the match doesn't have to be perfect and may even need spaces to help it line up. Circle or draw lines between the matching letters.

DNA MATCH

```
Query:    AGCGAATATTATGTTGAAGTAGCAAAGTCCTGGAGCCT

Sbjct:    ACTACAGGGGAGTTTTGTTGAAGTTGCAAAGTCCTGGAGCCTCCAGAGGGC
```

PROTEIN MATCH

```
Query:    MEMKATTALLNDRVLRAMLYFWCKAEETCALEVCEE

Sbjct:    ETIRRAYPDANLLNDRVLRAMLYFWRKAEETCAPSVSMRKIVATWMLEVCEE
```

Reflection

- How much of the query did you try to match at one time?
- How did you find a match? Can you describe it in words?
- Were there any mismatches for the best sequence?
- Were there ever multiple matches? Would breaking up the Query (introducing a gap) help?

Below is the answer. The vertical lines indicate a perfect match between the letters of the two sequences. Note that there are some mismatches and that a big gap must be inserted for the end of the protein Query sequence to match the Sbjct sequence (LEVCEE).

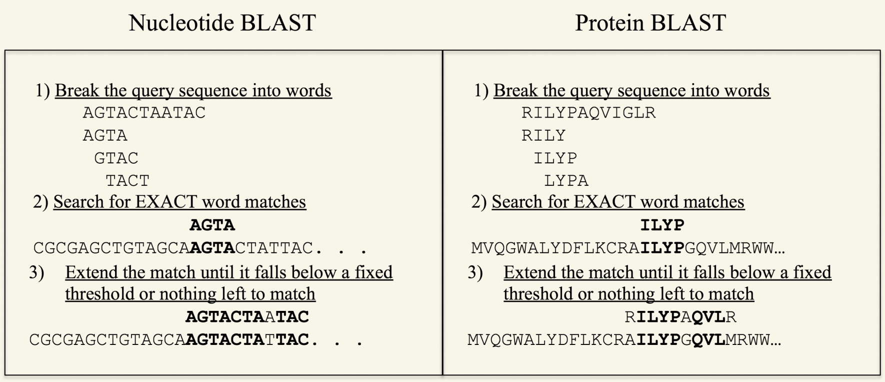
DNA MATCH

```
Query:     AGCGAATATTATGTTGAAGTAGCAAAGTCCTGGAGCCT
                   || ||||||||| |||||||||||||||||
Sbjct: ACTACAGGGGAGTTTTGTTGAAGTTGCAAAGTCCTGGAGCCTCCAGAGGGC
```

PROTEIN MATCH

```
Query:    MEMKATTALLNDRVLRAMLYFWCKAEETCA-------------LEVCEE
                  |||||||||||||| |||||||             ||||||
Sbjct: ETIRRAYPDANLLNDRVLRAMLYFWRKAEETCAPSVSMRKIVATWMLEVCEE
```

Most students solve this problem by (i) sliding the Query sequence along the Sbjct sequence, (ii) finding a short region that matches well, and then (iii) extending the match as far as possible. This is essentially how the BLAST algorithm works. Figure 1.1.1 details the basic steps of the algorithm.



```
1) Break the query sequence into words
     AGTACTAATAC
     AGTA
      GTAC
       TACT
2) Search for EXACT word matches
                AGTA
CGCGAGCTGTAGCAAGTACTATTAC. . .
3) Extend the match until it falls below a fixed
   threshold or nothing left to match
                AGTACTAATAC
CGCGAGCTGTAGCAAGTACTATTAC. . .
```

```
1) Break the query sequence into words
     RILYPAQVIGLR
     RILY
      ILYP
       LYPA
2) Search for EXACT word matches
                ILYP
MVQGWALYDFLKCRAILYPGQVLMRWW…
3) Extend the match until it falls below a fixed
   threshold or nothing left to match
               RILYPAQVLR
MVQGWALYDFLKCRAILYPGQVLMRWW…
```

FIGURE 1.1.1. Principles behind the BLAST algorithm. (1) The first step of the algorithm is to break the Query sequence into smaller pieces called "words." For DNA, the word size is usually 10 or 11 letters long, but the example uses four-letter words for simplicity. (Four-letter words. *snicker*) Protein sequence matches start with fewer letters. (2) Then the algorithm slides these smaller words across possible target sequences until it finds a perfect match with one of the small words. (3) Starting with this small alignment, BLAST then extends the alignment until it runs out of letters or the alignment becomes poor (lots of mismatches). The score of the alignment is determined by summing up the scores for matches and mismatches. For DNA, BLAST uses +5 for a match and –4 for a mismatch. The scores of protein sequence alignments are determined by using a special table of match/mismatch scores, such as the BLOSUM62 matrix (see Chapter 07).

Exercises

Interactive exercise (theory)

Use the online BLAST Interactive Link below to learn how the algorithm makes and scores BLAST sequence alignments. Click on the dark circle with the yellow letter I at the top of the page to learn how to use the BLAST Exercise teaching interactive. Once you learn how it works, solve the activity problem.

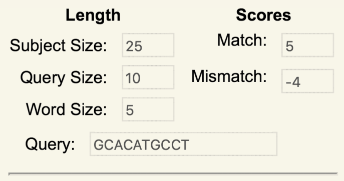
BLAST Interactive Link
Link:
http://kelleybioinfo.org/algorithms/default.php?o=1

Problem

Practice with the BLAST Exercise Link, then solve the problem below.

1. Write the best alignment of the Query to each DNA sequence in the boxes and circle the first matching word from the Query.

2. Calculate scores and rank the three alignments.



Length		Scores	
Subject Size:	25	Match:	5
Query Size:	10	Mismatch:	-4
Word Size:	5		
Query:	GCACATGCCT		

Query: **G C A C A T G C C T**

DNA 1:	C	A	G	A	G	A	C	C	T	C	G	A	T	C	C	T	T	A	C	A	T	G	C	C	A
DNA 2:	C	A	G	A	G	A	C	C	T	C	G	A	T	G	C	A	C	A	T	G	T	C	T	T	G
DNA 3:	C	A	C	A	C	A	T	G	A	C	T	T	G	G	A	C	T	C	T	G	C	C	A	T	G

	Score	Rank
DNA 1:		
DNA 2:		
DNA 3:		

Lab Exercises (Practice)

In this part of the exercise, you will learn how to analyze mystery DNA and protein sequences using the BLAST algorithm online at NCBI. You will also learn how to interpret the output from the program including what the values mean and how to find information about the best match in the database to your query sequence. You will also use a program, called ORF finder, that translates a DNA sequence into likely protein sequences.

NCBI BLAST Tutorial
Link:
http://kelleybioinfo.org/algorithms/tutorial/TAli1.pdf

Sample and lab exercise data:
http://kelleybioinfo.org/algorithms/data/DAli1.txt

Lab Exercise

Click on the sample and lab exercise data link for the sequence data used in this exercise.

Part 1

Use NCBI BLAST tools to analyze the following DNA that you just sequenced from a plasmid and answer the following questions:

```
>Part1_Plasmid_Derived_Sequence
CGTTTACGGCGTGGACTACCAGGGTATCTAATCCTGTTCGCTCCCCAACGCTTTCGCTCCTCAGCGT
CAGTTACTGCCCAGAGACCCGCCTTCGCCACCGGTGTTCCTCCTGATATCTGCGCATTCCACCGCTA
CACCAGGAATTCCAGTCTCCCCTGC
```

1. Use the NCBI BLAST tool to perform a sequence search with the above sequence.

    a. The highest-scoring BLAST hit is to what named organism? (Ignore the unknown/uncultured organism hits.)

    b. What is the gene name?

    c. What is the function of the gene, if known? (Don't know? Try asking "Professor" Google or "Dr." Wikipedia!)

    d. Who submitted the sequence?

    e. From what institution?

    f. Get the following data for this particular match.

        i. E-value:

        ii. Identities:

Part 2

```
>Part2_Protein_Sequence
MNGTEGPNFYVPFSNKTGVVRSPFEYPQYYLAEPWQFSMLAAYMFLLIVLGFPINFLTLYVTVQHKK
LRTPLNYILLNLAVANHFMVFGGFTTTLYTSLHGYFVFGSTGCNLEGFFATLGGEIALWSLVVLAIE
RYVVVCKPMSNFRFGENHAIMGVAFTWVMALACAAPPLVGWSRYIPEGMQCSCGIDYYTLKPEVNNE
SFVIYMFVVHFTIPMTIIFFCYGQLVFTVKEAAAQQQESATTQKAEKEVTRMVIIMVIAFLICWVPY
ASVAFYIFTHQGSDFGPILMTLPAFFAKSSAIYNPVIYIMMNKQFRNCMLTTICCGKNPFGEEEGST
TASKTETSQVAPA
```

1. Use the NCBI BLAST tool to perform a sequence search with the protein sequence above.

    a. The highest-scoring BLAST hit is to what organism?

    b. What is the gene?

    c. What is the function of the gene, if known?

    d. Who submitted the sequence?

    e. From what institution?

    f. Get the following data for this particular match.

        i. E-value:

        ii. Identities:

2. Use the NCBI ORF finder (**https://www.ncbi.nlm.nih.gov/orffinder/**) to translate the Mystery DNA below. This sequence came from a bacterial culture, so use the bacterial genetic code.

   a. First of all, what exactly is an "ORF" anyway?

   b. What are the nucleotide positions of the longest ORF?

   c. How many different reading frames did the program reveal that were longer than 100 amino acids (aa)?

   d. BLAST the longest putative ORF.

      i. What is the name of the gene?

      ii. What is the name of the organism?

>Part2_Mystery_DNA

```
TCCCTCCACAAAGAATGGAGCTGTGAACTACTAGCACGCAATGTGATTCCTGCAATTGAAAATGAACAAT
ATATGCTACTTATAGATAACGGTATTCCGATCGCTTATTGTAGTTGGGCAGATTTAAACCTTGAGACTGA
GGTGAAATATATTAAGGATATTAATTCGTTAACACCAGAAGAATGGCAGTCTGGTGACAGACGCTGGATT
ATTGATTGGGTAGCACCATTCGGACATTCTCAATTACTTTATAAAAAAATGTGTCAGAAATACCCTGATA
TGATCGTCAGATCTATACGCTTTTATCCAAAGCAGAAAGAATTAGGCAAAATTGCCTACTTTAAAGGAGG
TAAATTAGATAAAAAAAACAGCAAAAAAAACGTTTTGATACATATCAAGAAGAGCTGGCAACAGCACTTAAA
AATGAATTTAATTTTATTAAAAAATAGAAGGAGACATCCCTTATGGGAACTAGACTTACAACCCTATCAA
ATGGGCTAAAAAACACTTTAACGGCAACCAAAAGTGGCTTACATAAAGCCGGTCAATCATTAACCCAAGC
CGGCAGTTCTTTAAAAACTGGGGCAAAAAAAATTATCCTCTATATTCCCCAAAATTACCAATATGATACT
GAACAAGGTAATGGTTTACAGGATTTAGTCAAAGCGGCCGAAGAGTTGGGGATTGAGGTACAAAGAGAAG
AACGCAATAATATTGCAACAGCTCAAACCAGTTTAGGCACGATTCAAACCGCTATTGGCTTAACTGAGCG
TGGCATTGTGTTATCCGCTCCACAAATTGATAAATTGCTACAGAAAACTAAAGCAGGCCAAGCATTAGGT
TCTGCCGAAAGCATTGTACAAAATGCAAATAAAGCCAAAACTGTATTATCTGGCATTCAATCTATTTTAG
GCTCAGTATTGGCTGGAATGGATTTAGATGAGGCCTTACAGAATAACAGCAACCAACATGCTCTTGCTAA
AGCTGGCTTGGAGCTAACAAATTCATTAATTGAAAATATTGCTAATTCAGTAAAAACACTTGACGAATTT
GGTGAGCAAATTAGTCAATTTGGTTCAAAACTACAAAATATCAAAGGCTTAGGGACTTTAGGAGACAAAC
TCAAAAATATCGGTGGACTTGATAAAGCTGGCCTTGGTTTAGATGTTATCTCAGGGCTATTATCGGGCGC
```

```
AACAGCTGCACTTGTACTTGCAGATAAAAATGCTTCAACAGCTAAAAAAGTGGGTGCGGGTTTTGAATTG
GCAAACCAAGTTGTTGGTAATATTACCAAAGCCGTTTCTTCTTACATTTTAGCCCAACGTGTTGCAGCAG
GTTTATCTTCAACTGGGCCTGTGGCTGCTTTAATTGCTTCTACTGTTTCTCTTGCGATTAGCCCATTAGC
ATTTGCCGGTATTGCCGATAAATTTAATCATGCAAAAAGTTTAGAGAGTTATGCCGAACGCTTTAAAAAA
TTAGGCTATGACGGAGATAATTTATTAGCAGAATATCAGCGGGGAACAGGGACTATTGATGCATCGGTTA
CTGCAATTAATACCGCATTGGCCGCTATTGCTGGTGGTGTGTCTGCTGCTGCAGCCGGCTCGGTTATTGC
TTCACCGATTGCCTTATTAGTATCTGGGATTACCGGTGTAATTTCTACGATTCTGCAATATTCTAAACAA
GCAATGTTTGAGCACGTTGCAAATAAAATTCATAACAAAATTGTAGAATGGGAAAAAAATAATCACGGTA
AGAACTACTTTGAAAATGGTTACGATGCCCGTTATCTTGCGAATTTACAAGATAATATGAAATTCTTACT
GAACTTAAACAAAGAGTTACAGGCAGAACGTGTCATCGCTATTACTCAGCAGCAATGGGATAACAACATT
GGTGATTTAGCTGGTATTAGCCGTTTAGGTGAAAAAGTCCTTAGTGGTAAAGCCTATGTGGATGCGTTTG
AAGAAGGCAAACACATTAAAGCCGATAAATTAGTACAGTTGGATTCGGCAAACGGTATTATTGATGTGAG
TAATTCGGGTAAAGCGAAAACTCAGCATATCTTATTCAGAACGCCATTATTGACGCCGGGAACAGAGCAT
CGTGAACGCGTACAAACAGGTAAATATGAATATATTACCAAGCTCAATATTAACCGTGTAGATAGCTGGA
AAATTACAGATGGTGCAGCAAGTTCTACCTTTGATTTAACTAACGTTGTTCAGCGTATTGGTATTGAATT
AGACAATGCTGGAAATGTAACTAAAACCAAAGAAACAAAAATTATTGCCAAACTTGGTGAAGGTGATGAC
AACGTATTTGTTGGTTCTGGTACGACGGAAATTGATGGCGGTGAAGGTTACGACCGAGTTCACTATAGCC
GTGGAAACTATGGTGCTTTAACTATTGATGCAACCAAAGAGACCGAGCAAGGTAGTTATACCGTAAATCG
TTTCGTAGAAACCGGTAAAGCACTACACGAAGTGACTTCAACCCATACCGCATTAGTGGGCAACCGTGAAG
AAAAAATAGAATATCGTCATAGCAATAACCAGCACCATGCCGGTTATTACACCAAAGATACCTTGAAAG
CTGTTGAAGAAATTATCGGTACATCACATAACGATATCTTTAAAGGTAGTAAGTTCAATGATGCCTTTAA
CGGTGGTGATGGTGTCGATACTATTGACGGTAACGACGGCAATGACCGCTTATTTGGTGGTAAAGGCGAT
GATATTCTCGATGGTGGAAATGGTGATGATTTTATCGATGGCGGTAAAGGCAACGACCTATTACACGGTG
GCAAGGGCGATGATATTTTCGTTCACCGTAAAGGCGATGGTAATGATATTATTACCGATTCTGACGGCAA
TGATAAATTATCATTCTCTGATTCGAACTTAAAAGATTTAACATTTGAAAAAGTTAAACATAATCTTGTC
ATCACGAATAGCAAAAAAGAGAAAGTGACCATTCAAAACTGGTTCCGAGAGGCTGATTTTGCTAAAGAAG
TGCCTAATTATAAAGCAACTAAAGATGAGAAAATCGAAGAAATCATCGGTCAAAATGGCGAGCGGATCAC
CTCAAAGCAAGTTGATGATCTTATCGCAAAAGGTAACGGCAAAATTACCCAAGATGAGCTATCAAAAGTT
GTTGATAACTATGAATTGCTCAAACATAGCAAAAATGTGACAAACAGCTTAGATAAGTTAATCTCATCTG
TAAGTGCATTTACCTCGTCTAATGATTCGAGAAATGTATTAGTGGCTCCAACTTCAATGTTGGATCAAAG
TTTATCTTCTCTTCAATTTGCTAGAGCAGCTTAATTTTTAATGATTGGCAACTCTATATTGTTTCACACA
TTATAGAGTTGCCGTTTTATTTTATAAAAGGAGACAATATGGAAGCTAACCATCAAAGGAATGATCTTGG
TTTAGTTGCCCTCACTATGTTGGCACAATACCATAATATTTCGCTTAATCCGGAAGAAATAAAACATAAA
```