Chapter 4

Database Searching and Multiple Alignment:

Investigating Antibiotic Resistance

Chapter Overview

This chapter develops skills in two very commonly used types of Web-based bioinformatics tools: searching sequence databases for high-scoring matches to a query sequence (using BLAST) and multiple sequence alignment (using ClustalW). No programming project is provided; however, the algorithms and parameters used by these programs, both of which use heuristic methods to speed up complex tasks, are discussed in some detail. This chapter focuses on algorithms for optimal alignment of DNA sequences. This chapter is recommended for both programming and non-programming courses because these techniques and those related to them are used extensively in real-world bioinformatics applications.

Biological problem: Overuse of agricultural antibiotics and development of antibiotic resistance

Bioinformatics skills: One-to-many sequence alignments and multiple sequence alignment

Bioinformatics software: BLAST and ClustalW

Programming skills: Heuristics

Understanding the Problem:
Antibiotic Resistance

Fifty years ago, many people believed the newly discovered antibiotics—drugs that selectively kill bacteria without harming human hosts—would end infectious diseases caused by bacteria. Indeed, these "miracle drugs" have preserved the lives of millions. Today, however, tuberculosis, pneumonia, diarrheal disease, staph infections, and other bacterial diseases remain important—and in some cases increasing—causes of illness and death. One important reason is the dramatic rise of antibiotic-resistant bacteria no longer killed by commonly used antimicrobial drugs.

Resistance results from selection for mutants that can survive antibiotic treatment (see Bio-Background at the end of this chapter). As the use of an antibiotic becomes widespread, bacteria are increasingly exposed to it, escalating selective pressure and resulting in rapid evolution

Figure 4.1 The extensive use of antibiotics in agricultural animals that are not sick has sparked controversy about the role of this practice in speeding the development of antibiotic-resistant bacteria. Courtesy of Scott Bauer/USDA ARS. Inset © AbleStock.

of strains that thrive when antibiotics kill their susceptible cousins. Thus, in an effort to curb resistance, physicians today are much more cautious than in the past, prescribing antibiotics only when the need is clear and holding those least prone to resistance in reserve.

The nontherapeutic use of antibiotics in agricultural animals and even on food crops (**Figure 4.1**) is at the center of a current controversy over resistance. Routine use of antibiotics in animal feed prevents disease and promotes growth, allowing more animals to be raised more cheaply in less space. But many believe these economic benefits come at a high cost: Are the 28 million tons of agricultural antibiotics used annually in the United States and Canada (far outweighing the 3 million tons for all human uses) promoting antibiotic resistance? Most scientists believe that antibiotic overuse is a major contributor to the development and spread of resistance, leading to bans on subtherapeutic agricultural use of antibiotics in Denmark in 1999 and in the European Union in 2006. No such legislation is yet in place in the United States, and those who oppose such laws argue that no causal link has been definitively established between agricultural antibiotics and antibiotic-resistant disease bacteria in humans. We can investigate this link using some more advanced sequence alignment techniques.

Bioinformatics Solutions:
Advanced Sequence Comparison Algorithms

There is no question that intensive use of antibiotics in animals increases the prevalence of antibiotic-resistant bacteria—in animals. But how can a microbiologist determine experimentally whether these bacteria are an important source of resistance genes for bacteria that cause disease in *humans*? In 2001, Abigail Salyers and her colleagues used bioinformatics to look for evidence that bacteria inhabiting the human gut had been the recipients of antibiotic-resistance genes originating in bacteria found in domestic animals (see References and Supplemental

Reading). Taking advantage of the many sequenced bacterial genomes and the huge collection of sequenced genes in public genome databases, they looked for *un*related animal and human bacteria that have closely related resistance genes.

New or altered genes, including those that allow a bacterial cell to resist an antibiotic, arise by random mutation, which is rare. However, once these genes exist in a bacterial community, they can be readily passed from one bacterium to another (usually on plasmids), a phenomenon known as horizontal gene transfer (HGT; see BioBackground), allowing resistance to spread rapidly in a bacterial community. If a "donor" bacterium gives a resistance gene to a "recipient" organism, the two should have the *same* gene—that is, one that encodes a protein with the same amino-acid sequence. Furthermore, if human pathogens have the same antibiotic-resistance genes as bacteria from domestic animals, it would suggest that HGT occurs between them, supporting the conclusion that increased resistance among agricultural bacteria is indeed dangerous to human health. Similarity, of course, can be measured by sequence alignment, so Salyers used alignment first to retrieve genes from GenBank that were similar to a particular resistance gene and then to ask how similar the genes from unrelated species were. Two resistance genes that were ≥95% identical were assumed to have resulted from an interspecies gene transfer event.

The pairwise comparison techniques we have used thus far are of limited value when many sequences must be compared efficiently. In the sections that follow, we explore tools that build on the alignment algorithms we have already seen to allow for the rapid comparison of one sequence to many or the simultaneous alignment of multiple sequences.

BioConcept Questions

To successfully complete this chapter's projects, you need to understand a little about antibiotic resistance, HGT, and how similarity measurement can help us decide whether HGT has occurred. Use these questions to test your biological understanding; read BioBackground at the end of the chapter if you need a better foundation.

1. What is the difference between vertical and horizontal gene transfer? Why are the terms "vertical" and "horizontal" used to describe these processes?
2. Any bacterium could become antibiotic resistant by means of mutation. Why is HGT considered so much more of a threat, at least in terms of medically important resistance?
3. How does the degree of similarity between two genes help us understand whether they descended vertically from a common ancestor (recent or distant) or whether they could have moved from one species to the other by HGT?
4. Suppose you have evidence that two genes in two different bacterial species have a single, common origin. Give two possible explanations for how this might have occurred.

Understanding the Algorithm:
Database Searching and Multiple Alignment

BLAST: A Heuristic Approach to Database Searching

The Needleman-Wunsch algorithm is a relatively efficient algorithm for optimal, global pairwise sequence alignment. However, imagine that you wanted to align an antibiotic-resistance gene of interest with every other sequence in GenBank. The computational time required is the time to make one alignment (compute the matrix and alignment paths) times the number of sequences in the database—currently more than 100 million. We would say that the time required to solve this problem is $O(NS)$ or on the order of NS, where S is the number of

sequences. It gets large quickly: If one alignment took 1 second of computer time, the whole search would take more than 3 years. Yet, **BLAST** (Basic Local Alignment Search Tool; see References and Supplemental Reading) can compare a query sequence to the entire database and return all matching sequences in a matter of seconds.

BLAST and its several variations are perhaps the most widely used of all bioinformatics software. As its name suggests, BLAST implements a local alignment algorithm similar in principle to the Smith-Waterman algorithm. However, it uses a **heuristic** or "shortcut" that makes it a practical and efficient solution to this complex problem.

To understand how BLAST works, we first need to clarify what we mean by a "matching" sequence. The point of comparing an antibiotic-resistance gene to GenBank is to identify similar sequences—generally, orthologs. Thus, a sequence matches the query if it shows statistically significant and/or biologically relevant similarity when aligned to the query. But how does BLAST make 100 million alignments so quickly? In fact, it does not make a full alignment for every sequence. Its first step is to break the query sequence into short "words" called *k*-tuples: subsequences *k* characters long. The default setting for *k* is 11 for DNA alignments and 3 for protein alignments. BLAST then scores close matches between these short sequences and each database sequence; this process is known as "seeding." Where it finds a good match, a local alignment algorithm finally comes into play, and the program tries to extend the alignment in both directions, comparing the resulting score with a threshold value. An alignment that can be extended to score above the threshold is referred to as a **high-scoring pair** (**HSP**).

Figure 4.2 shows an example of the BLAST algorithm using a famous quotation for which several variations can be found instead of a sequence (remember, alignment programs can compare any two strings). The alignments in the figure are scored with a simple match = 1, mismatch = 0, gap = −1 system. If the query sequence is broken down into three-letter words

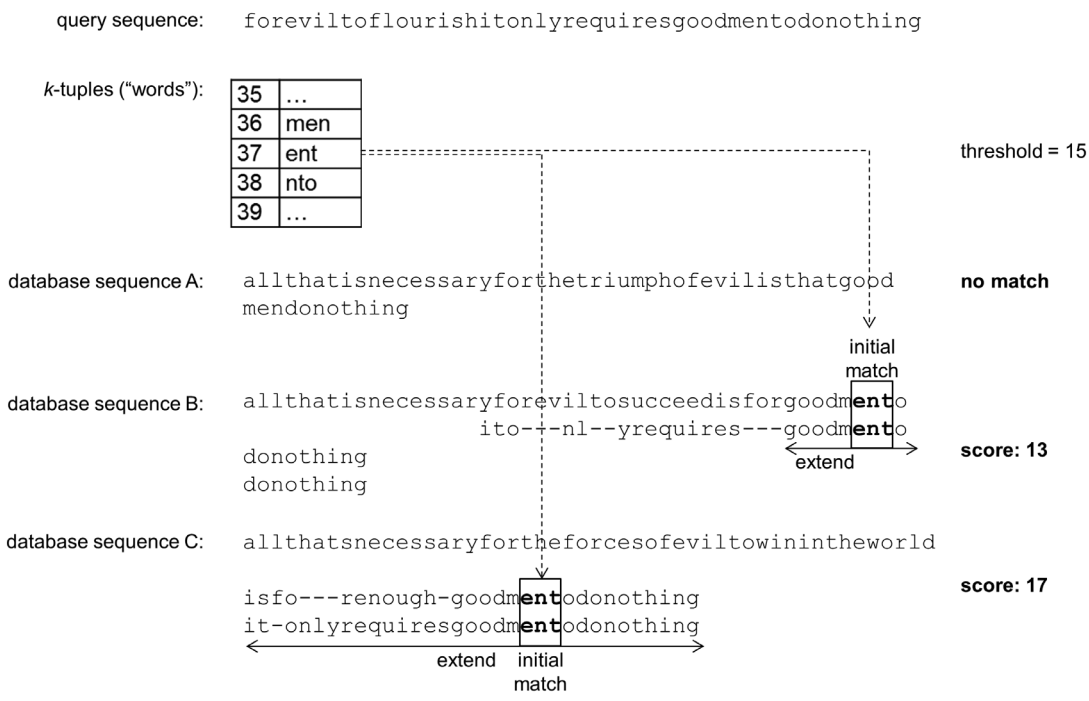


Figure 4.2 An example of how the BLAST algorithm finds an initial match between a short subsequence (*k*-tuple or "word") of the query and the target sequence, then extends the match to find a local alignment with scoring above a threshold value.

(*k*-tuples) and we focus on the 37th *k*-tuple, "ent," there is no match for any of the words in database sequence A, so this sequence can be discarded. Sequence B has an initial match, but attempting to extend the alignment does not increase the score above the threshold, so this sequence would not be reported as a significant alignment. Sequence C, however, has an alignment that exceeds the threshold score and would be reported as a match.

BLAST then calculates the statistical likelihood that a given score would occur based on mere chance alignment of unrelated sequences (the *e*-value) and orders the matching sequences according to this measure of statistical significance. As we will see in the next section, BLAST reports back to the user the name of the matching sequence, the score, the *e*-value, and the alignment itself. In addition to changing scoring parameters such as the gap penalty, BLAST allows the user to adjust the *k*-tuple value if desired. Although the default value typically works well, decreasing the word size allows the identification of sequences that match less well (useful when similarity of the query to other sequenced genes is weak) and is also needed if the sequence to be compared is very short (current implementations of BLAST do this automatically when a short query is entered).

You use heuristics all the time without realizing it. Consider, for example, how you decide which route to take when you have several alternatives. It is extremely difficult to calculate a truly optimal solution (accounting for traffic, construction, traffic lights, speed limits, school zones, and many more variables), so you apply a heuristic: You decide to take the route that is shortest in mileage or the one you believe has the least traffic. This allows you to choose rapidly but does not guarantee that you will in fact choose the fastest option. Similarly, BLAST's heuristic approach allows it to quickly discriminate possible matches from unrelated sequences. Although it may not find optimal alignments, it deals with large volumes of data extremely rapidly while finding solutions that are acceptably close to optimal.

ClustalW: Multiple Sequence Alignment

Although BLAST can quickly identify a large number of sequences similar to a query, it displays only individual alignments of the query with each matching sequence. However, we might instead want to see an alignment of a whole group of similar sequences at once (**Figure 4.3A**). For example, perhaps the sequences of genes similar to our query resistance gene fall into two or three distinctly identifiable groups. Or, we might want to identify a **consensus sequence**: the nucleotides or amino acids that appear the most frequently at each position in a given region of the sequence. Rather than a pairwise alignment, this requires a **multiple sequence alignment** algorithm.

The computational complexity problem for multiple sequence alignment is even greater than for database searching. Here, the order of adding sequences to the alignment matters. Suppose, for example, we have optimally aligned two sequences, `GTCT` and `GGT` as in **Figure 4.4A**. If we now want to align the sequence `CT` with the other two, we might get the alignment in **Figure 4.4B**. However, if we aligned `GTCT` with `CT` first, we might find the optimal alignment to be the one in **Figure 4.4C** instead. The dynamic programming approach of Needleman and Wunsch could deal with this problem by building a matrix of size $L \times M \times N$, each dimension one character longer than the length of one sequence. However, as more sequences are added, the matrix becomes four-, five-, six-dimensional, and so on and the computational time required becomes $O(N^s)$: the time required for one alignment raised to the *power* of the number of sequences, which obviously becomes impractical very fast.

Thus, multiple sequence alignment algorithms again use heuristics to manage the complexity of the problem. ClustalW (see References and Supplemental Reading) is one of the most popular multiple sequence alignment algorithms; it uses a **progressive alignment** algorithm in which the order of adding new sequences to the alignment is determined by first calculating a rough phylogenetic tree called a **guide tree** (**Figure 4.3B**). The guide tree

Figure 4.3 (A) Segment of a multiple sequence alignment for the coding region of a penicillin-resistance gene from five different species. Darker shading indicates nucleotides that are conserved among more of the five sequences. (B) Guide tree used by ClustalW to produce this multiple alignment. Data from: EBI ClustalW.

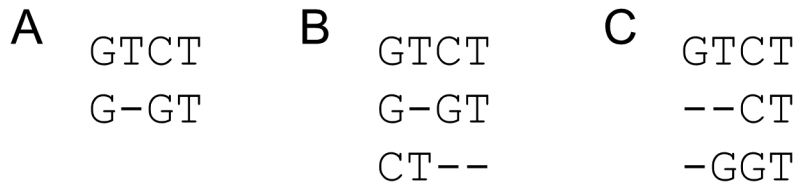


Figure 4.4 Multiple sequence alignment is complex because the order of adding sequences to the alignment can affect the alignment results.

is generated by first doing pairwise alignments and then using the score or percent similarity from those alignments to draw a tree showing which sequences are more and less closely related (we will have much more to say about the mechanics of generating a phylogenetic tree in subsequent chapters). Starting with the two most closely related sequences (in the example in Figure 4.3B, these are *Bacteroides xylanisolvens* and *B. fragilis*), ClustalW then does global, pairwise alignments to align each new sequence with those already aligned, in order of decreasing relatedness. Note that although this is an efficient way to produce a multiple alignment, the fact that it is based on global alignment means ClustalW may not correctly align sequences that share regions of similarity if the sequences are not very similar overall.

Test Your Understanding

1. Describe two features of the BLAST algorithm that enable it to complete a database search much faster than the Needleman-Wunsch algorithm would.
2. For the BLAST example in Figure 4.2, are there *k*-tuples within the query sequence that give a very different result? What might be an example of a query sequence that would yield an HSP for all three database sequences?
3. Describe briefly how the sequence differences you can see in Figure 4.3A relate to the lengths of the branches in Figure 4.3B.
4. In Figure 4.3, the sequence labeled CA_F7SDb01 is from an organism that has not yet been characterized sufficiently to give it a species name; all other sequences are from species within the genus *Bacteroides*. Based on the region of the multiple alignment shown in this figure, would you characterize CA_F7SDb01 as likely to belong to some *Bacteroides* species or likely to come from a different genus?
5. Write out a set of six short (seven or eight nucleotides) DNA sequences in which all six are related but there are two sets of three that are more closely related to each other than to the other set. Show how the guide tree might look for your sequences and then what the multiple sequence alignment might look like.

CHAPTER PROJECT:

Horizontal Gene Transfer of Antibiotic Resistance

Salyers and her colleagues (see References and Supplemental Reading) used bioinformatics methods to look for evidence of horizontal transfer of antibiotic genes between bacteria found in animals routinely fed antibiotics and bacteria that might affect human health. Because of the enormous number of bacteria residing normally in the human large intestine, they hypothesized that these bacteria serve as a reservoir for HGT (**Figure 4.5**) and could easily

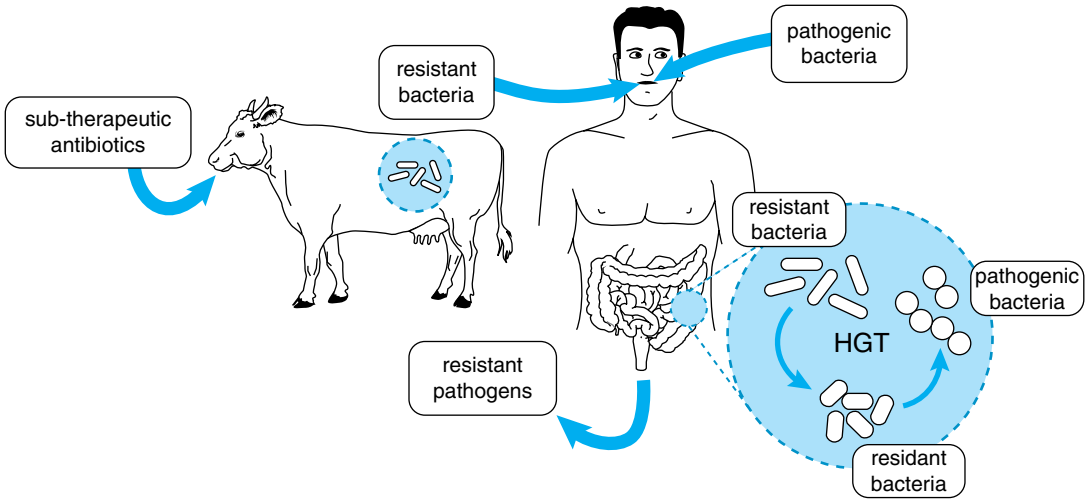


Figure 4.5 According to the "reservoir hypothesis" proposed by Salyers and others, resistant bacteria ingested in food that pass through the human large intestine have the opportunity to transfer resistance to any of the trillions of bacteria resident there, creating a reservoir of resistance, which could then lead to transfer to human pathogens.

exchange genes with ingested bacteria, including antibiotic-resistant bacteria originating in agricultural animals. Thus, using alignment methods, Salyers focused on determining whether common intestinal bacteria might carry the *same* genes for antibiotic resistance as unrelated species that are not gut residents. Related species, of course, are likely to have highly similar genes, but a high degree of similarity between genes of otherwise *dissimilar* organisms strongly suggests horizontal transfer. Salyers used the criterion of ≥95% similarity to decide whether sequences from two organisms in fact represented the same gene. In this project, we will use BLAST to identify a set of resistance genes of interest and ClustalW to examine the similarity among them, enabling us to draw some conclusions about the impacts of subtherapeutic agricultural antibiotic use. We will focus on genes enabling bacteria to resist the antibiotic erythromycin, a drug commonly used in both therapeutic and agricultural applications.

Learning Objectives

- Understand the value of searching a database for sequences matching a query
- Gain experience with the use of BLAST in database searching and understand its parameters
- Appreciate the importance of a heuristic in processing large amounts of data rapidly
- Understand the use of multiple sequence alignment and know how to use ClustalW for this purpose

Suggestions for Using the Project

This project is designed to build skills in using two very important pieces of bioinformatics software: BLAST and ClustalW. Because of their wide use, familiarity with these tools is highly recommended for students in both programming and nonprogramming courses. The BLAST and ClustalW sections that follow can be used independently; instructors can download a set of *ermB* sequences from the *Exploring Bioinformatics* website if they would like their students to do the multiple alignment without first using BLAST to identify sequences of interest. Instructors could also ask students in programming courses to implement a BLAST-like algorithm based on the earlier discussion.

Searching for Erythromycin Resistance Genes with BLAST

Obtaining the ermB *Sequence*

Erythromycin is an antibiotic that halts bacterial growth by binding to the bacterial ribosome and blocking translation. Two different mechanisms of erythromycin resistance have been observed: Some resistant bacteria have acquired a gene whose product modifies the ribosome so erythromycin can no longer bind, whereas others have acquired a gene encoding a transport protein (called an efflux pump) that rapidly removes erythromycin from the cell. You already know how to find sequences in GenBank via a text search; however, a key word such as "erythromycin" will retrieve both kinds of genes and will fail to retrieve any resistance genes that were not annotated as such. Instead, using BLAST, we can search using a *sequence* as our query and retrieve all similar sequences, regardless of how they are annotated.

As our query sequence, we use an erythromycin-resistance gene called *ermB* from *Streptococcus agalactiae*, a Gram-positive bacterial species commonly associated with the udder of cows, where it can cause mastitis. This gene produces one of several known resistance proteins of the ribosome-modification type. Erythromycin resistance due to *ermB* has commonly been seen in the human pathogen *Streptococcus pneumonia*, the most common cause of bacterial pneumonia, so it will be interesting to determine whether HGT of this gene has occurred among diverse bacteria. Start by obtaining the DNA sequence for the *S. agalactiae ermB* coding region from GenBank in FASTA format by using a text search, by searching for the accession number DQ355148.1, or by downloading the file from the *Exploring Bioinformatics* website.

Understanding BLAST Results

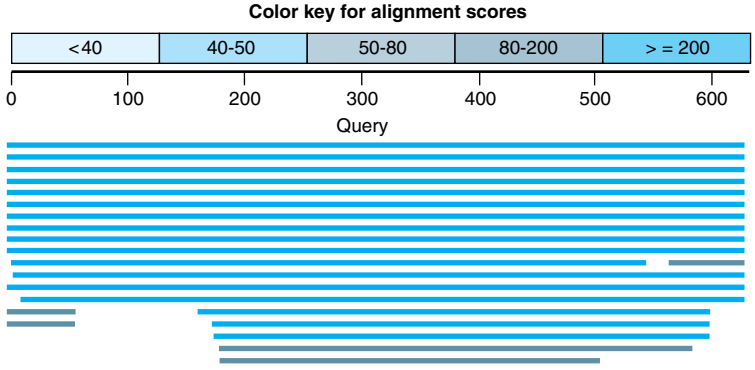
BLAST results are shown in three sections. The top section is a graphical view (see sample of some representative BLAST results in **Figure 4.6A**), with a bar for each sequence that matches the query. The length of the bar shows the length(s) of the matching region(s), and its color represents the score for each segment. The middle section (**Figure 4.6B**) gives details about each match: the accession number and description for the gene matched and five parameters related to the quality of the match:

- **Max score**: the score of the best matching segment (remember, this is a local alignment, not a global one).
- **Total score**: the total scores of all matching segments found (same as max score if there is only one matching segment).
- **Query coverage**: the percentage of the query sequence that aligned to some part of the match.
- ***e*-Value**: a statistical measure evaluating how likely it is that a match this good would occur by chance. The lower the *e*-value, the more likely it is that the two sequences are truly similar and not just chance matches. Two identical sequences would have an *e*-value of zero.
- **Max ident**: the percentage of nucleotides that are identical between the query and target sequences within the matching regions.

The third section (**Figure 4.6C**) shows the actual pairwise alignments between the query sequence and the top matching database sequences. Links in each section provide direct access to a variety of additional information about the matching sequences.

Identifying ermB *Orthologs with BLAST*

From the **NCBI BLAST home page**, you can see several ways to run BLAST, including both nucleotide and protein comparisons. For this exercise, we compare DNA sequences, so you should choose the nucleotide option. This should take you to a search form where you can either paste or upload your *S. agalactiae ermB* sequence.



(A)

Sequences producing significant alignments:
(Click headers to sort columns)

Accession	Description	Max score	Total score	Query coverage	E-value	Max ident
CP000948.1	Escherichia coli str. K12 substr. DH10B, complete genome	1132	1132	100%	0.0	100%
AP009048.1	Escherichia coli W3110 DNA, complete genome	1132	1132	100%	0.0	100%
U00096.2	Escherichia coli str. K-12 substr. MG1655, complete genome	1132	1128	100%	0.0	99%
CP000946.1	Escherichia coli ATCC 8739, complete genome	1126	1126	100%	0.0	99%
CP000802.1	Escherichia coli HS, complete genome	1126	1126	100%	0.0	99%
CP000970.1	Escherichia coli SMS-3-5, complete genome	1122	1122	100%	0.0	99%
CP000266.1	Shigella flexneri 5 str. 8401, complete genome	1113	1113	100%	0.0	99%
AE014073.1	Shigella flexneri 2a str. 2457T, complete genome	1113	1113	100%	0.0	99%
BA000007.2	Escherichia coli O127:H7 str. Sakai DNA, complete genome	1113	1113	100%	0.0	99%
AE005174.2	Escherichia coli O157:H7 EDL933, complete genome	1113	1113	100%	0.0	99%
AE005674.1	Shigella flexneri 2a str. 301, complete genome	1108	1108	100%	0.0	99%
CP000034.1	Shigella dysenteriae Sd197, complete genome	1108	1108	100%	0.0	99%
CP000038.1	Shigella sonnei Ss046, complete genome	1108	1108	100%	0.0	99%
CP000800.1	Escherichia coli E24377A, complete genome	1099	1099	100%	0.0	98%
CP000036.1	Shigella boydii Sb227, complete genome	1095	1095	100%	0.0	98%
CP001063.1	Shigella boydii CDC 3083-94, complete genome	1090	1090	100%	0.0	98%
CP000468.1	Escherichia coli APEC 01, complete genome	722	722	99%	0.0	85%

(B)

Score = 1132 bits (1254), Expect = 0.0
Identities = 627/627 (100%), Gaps = 0/627 (0%)
Strand = Plus/Plus

```
Query  1        TTAAGCCAGCTCACCCTTCACTAAAGGGACAAAGCGCACGGCCTCC
                ||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2959457  TTAAGCCAGCTCACCCTTCACTAAAGGGACAAAGCGCACGGCCTCC

Query  61       AAATTCGCCTCCCCGACGACGCACCCGTTTCAAATACTGGTGCTCC
                ||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2959517  AAATTCGCCTCCCCGACGACGCACCCGTTTCAAATACTGGTGCTCC

Query  121      GACGAGAATCCCGCCTTCGTCCAGCTGCGTCATTAGCGCAGTTGGA
                ||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2959577  GACGAGAATCCCGCCTTCGTCCAGCTGCGTCATTAGCGCAGTTGGA

Query  181      CGCCGTAACAATGATAGCGTCAAACGGCGCACGTGCCTGCCAACCT
                ||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2959637  CGCCGTAACAATGATAGCGTCAAACGGCGCACGTGCCTGCCAACCT

Query  241      ATGACGGGTTGAAACATTATGTAAATCAAGATTTTTCAGGCGGCGA
                ||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2959697  ATGACGGGTTGAAACATTATGTAAATCAAGATTTTTCAGGCGGCGA

Query  301      CAAGCCTTTAATCCGTTCAACCGAGCAAACATGCTGGACAAGATGC
                ||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2959757  CAAGCCTTTAATCCGTTCAACCGAGCAAACATGCTGGACAAGATGC
```

(C)

Figure 4.6 Sample results of a BLAST search for database sequences matching a nucleotide query sequence: (A) graphical summary of results, (B) table of scores, and (C) alignments.

Many options and parameters are available on this page. Notice the section labeled `Choose Search Set`, where you can specify the sequences to be searched. Importantly, the default set of sequences is the subset of GenBank containing human DNA sequences. This obviously will not work in our case, where we want to retrieve bacterial sequences. Change the database to `nucleotide collection (nr/nt)`, which will search all the unique ("nonredundant" or nr) sequences in GenBank. Furthermore, many sequences in GenBank are from bacteria that have been sequenced (using DNA harvested from an environmental sample) but never cultured; these are not useful to us because we do not know what species they come from, so check the box to exclude sequences from uncultured samples. To further refine the results, there is also an input box where you can limit your search to a particular organism or group of organisms; you could type `bacteria` here to exclude any nonbacterial sequences that might happen to match. Finally, there is a box where you can type an Entrez query to include or exclude specific kinds of sequences.

If you click `Algorithm parameters` near the bottom, you can set the parameters that BLAST uses for its comparison. These options should be starting to look familiar to you: For example, you can set a linear or affine gap penalty, change the match and mismatch scores, and alter the word size (*k*-tuple) for the initial match. Some of these parameters are set automatically when you make a choice from the `Program selection` section, where you choose the specific algorithm that will be used by selecting options such as `Highly similar sequences (megablast)` or `Somewhat similar sequences (blastn)`. With the parameters visible, try clicking each of these options and notice how the parameters change. For example, megablast has a default word size of 28, whereas blastn has a default of 11; how would this change the results? When you have finished exploring, choose `blastn` for now to see both very similar and less-similar sequences the program might identify. Click the `BLAST` button to start the search and compare your *ermB* sequence with the selected sequences. In a short time, you should get a page of results (see Figure 4.6 for an example of what this page would look like).

Web Exploration Questions

1. In their original survey, Salyers and colleagues used a cutoff of 95% identity for sequences considered similar enough to have been shared by HGT. You can get a quick measure of identity by using the `max ident` score in the BLAST results—however, you can also get a high max ident for a very small matched region, so also consider the query coverage. Looking at these parameters, are the matches that BLAST retrieved highly similar to your query or less similar? Do your data suggest that all or most of them represent the same gene, transferred from organism to organism by HGT?
2. You may notice in your list that a number of the sequence matches come from cloning vectors—engineered DNA molecules used for laboratory manipulations. Construct an Entrez query to exclude these from your results and run your search again—but be careful not to exclude too much. Remember that unless you limit the field, the entire text of each entry will be searched for a match. What query did you use?
3. What evidence can you find among your BLAST results to support or refute the hypothesis that resistance genes are being shared between unrelated species—especially between agricultural species and human pathogens or human gut bacteria that might come into contact with pathogens? You will have to do some detective work to answer this question: For example, find a bacterial phylogenetic tree online to help you decide how closely related the different species in your list are, and then try to find out which ones might be found in domestic animals, which are residents of the human gut, and which are human pathogens.
4. There are so many sequences in GenBank today, including many whole genome sequences, that BLAST often fills up its list of top matching sequences without ever getting down to less related but potentially more interesting matches. In your initial BLAST results, for example, it is likely that most

if not all sequences come from Gram-positive organisms, one major division of the bacteria. HGT to the more distantly related Gram-negative organisms would be very interesting but is hard to assess from this list. Construct a BLAST search that excludes Gram-positive matches. Or, another way to get interesting results might be to require matches to specific groups of Gram-negative organisms that you know live in the human gut, such as *Bacteroides* (the most common genus among human gut bacteria) or *Escherichia*. Be careful to exclude from consideration sequences that come from cloning vectors in this case—you only want sequences naturally found in these bacteria. Describe how you searched, the similarity of your results to the query, and whether the percent identity suggests that your results represent horizontally transferred genes or genes arising by mutation.

5. Based on your results thus far, would you say that you have evidence for (a) extensive HGT, (b) a mix of HGT and evolution by mutation, (c) evolution mostly by mutation with occasional HGT, or (d) a number of unrelated resistance genes? Support your answer with evidence.

Retrieving Sequences

In the next section, we will carry out a multiple alignment of some *ermB* genes from different species, which requires retrieving their sequences in FASTA format. The NCBI implementation of BLAST includes a number of useful tools for working with the sequences it finds, including a means of quickly retrieving the ones in which you are interested. Checkboxes next to the sequences BLAST aligned allow you to select interesting matches; chose some that are from different genera, from human pathogens or gut organisms, from Gram-negative organisms, and so on. Then, you should see a download link allowing you to retrieve the sequences in FASTA format. You can combine the results of several searches simply by downloading each set and then cutting and pasting in the resulting text files. Compile a file with several interesting sequences that you can go on to align with ClustalW.

Before leaving BLAST, take a look at the sequences you retrieved. In some cases, BLAST will have retrieved an entire plasmid or even genome sequence, even though only a short region of this sequence is actually of interest. You can use the accession numbers of these sequences to retrieve the GenBank entry and then obtain just the coding sequence (see Chapter 1). Or, even though BLAST aligned your query with a correctly oriented nontemplate strand of the gene from the database, it might retrieve the template strand if that is how the matching sequence was entered into GenBank; you can get the reverse complement using Sequence Manipulation Suite (Chapter 2) if this is the case. Your text file should ultimately contain correctly oriented coding sequences for all the *ermB* orthologs you intend to align. Finally, the comment lines may be long and not terribly helpful. Because the ClustalW implementation we will use does not like spaces and will truncate the comments, replace the comment lines with something more useful, such as simply the name of the species with no spaces (e.g., >Streptococcus_agalactiae).

Multiple Sequence Alignment with ClustalW

Although you were able to get some information about the similarity of many sequences to your query sequence from your BLAST results, you undoubtedly noticed that BLAST still only made pairwise comparisons: It showed alignments between your query and one other sequence at a time. When comparing many sequences, it can be much easier to analyze the results when all alignments can be visualized at once. Furthermore, some questions might be better answered by aligning a group of sequences: for example, to ask if there are particular regions of the sequences that are more or less conserved. ClustalW is an example of a multiple sequence alignment program designed for this purpose; sample output is shown in Figure 4.3A.

For this part of the project, you will need a text file containing the sequences of at least six to eight sequences similar to *ermB* in FASTA format. You should have all your sequences

in a single file, separated by their comment lines; be sure you have the coding regions only. If your class did not do the BLAST part of the project, your instructor can download a file with some interesting sequences from the *Exploring Bioinformatics* instructor website and make it available to you.

A good Web implementation of **ClustalW** is maintained by the EBI. Once you have loaded ClustalW, paste your entire list of sequences into the input box or upload your text file. Notice that two sets of parameters can be set: one for the initial pairwise alignments used to generate the guide tree and another for the subsequent multiple alignment itself. You will notice familiar ideas such as gap opening and extension penalties. Run your alignment initially with the default parameters.

When the results are returned, you will see the alignment in simple text format, with asterisks below the alignment wherever a particular nucleotide is found in all sequences. You can view the guide tree by clicking the appropriate tab, and the `Result Summary` tab shows the results of the individual pairwise alignments that were done. A more sophisticated presentation can be obtained by using Jalview, a Java-based viewer: click the `Result Summary` tab and then click `Start Jalview`. Here, you can see a consensus sequence representing the most conserved nucleotides at each position, and you can format and color the alignment in various ways. A convenient way to visualize differences among the sequences is by selecting `Percentage Identity` from the `Colour` menu; this gives a dark background for nucleotides conserved in all sequences and lighter colors for nucleotides conserved in fewer sequences.

Web Exploration Questions

6. Which *ermB*-like genes are the most similar? Which are less similar? Are there particular regions of the gene that are highly conserved or less conserved?
7. What kinds of differences can you see among these genes? Do substitutions outnumber indels or vice versa? What do you notice about the indels that occur in the alignment?
8. Try running ClustalW again with a very low gap penalty. Do the alignments change significantly? Which alignment is more biologically relevant, and what is your evidence for this view?
9. Based on the criterion of closely related genes from unrelated organisms, do your results support the HGT hypothesis?

How would you summarize your findings and conclusions regarding the likelihood that agricultural use of antibiotics can result in resistant human gut residents and/or resistant human pathogens? Your instructor may ask you to write up your findings in the form of a short report.

BioBackground: Antibiotic Resistance and Gene Transfer

Bacteria can have **natural (intrinsic) resistance** to some antibiotics because of their cell structure. For example, Gram-negative bacteria (such as the common intestinal organism *Escherichia coli*) are resistant to penicillin simply because the cell wall that penicillin attacks is protected by an outer membrane that other bacteria lack. But the resistance that is really important medically is **acquired resistance**: when bacteria that were previously sensitive to (killed by) an antibiotic

become resistant to it, making that antibiotic useless for treatment. Acquired resistance requires genetic change: Either a new gene or new variant of a gene arises by mutation or a cell acquires a preexisting gene by horizontal transfer.

Many people have the idea that using an antibiotic "makes" bacteria resistant. This is not true, however: Antibiotics do not *cause* resistance to occur (nor is it true that antibiotic use makes the *person* resistant to the antibiotic). However, antibiotic use can **select** for bacteria that have already become resistant, allowing them to become more prevalent in a population. As shown in **Figure 4.7**, if some bacteria in a population are more resistant to an antibiotic than others (due to mutation or to genes they have acquired), they will not be killed as easily when they encounter it. Thus, the antibiotic kills the most sensitive cells first and leaves the more resistant ones to pass their genes on. This can happen in your own body if you do not finish your antibiotic prescription: The most resistant cells remain alive and can then multiply and cause a relapse. The more we expose bacteria to antibiotics—whether in the body, in animals, or in the environment—the more we select for resistant organisms and thus the more prevalent the resistant bacteria become.

If a mutation gives a bacterial cell some advantage—and antibiotic resistance is just one of many possible examples—that cell's descendants inherit the altered gene. This is sometimes called **vertical gene transfer** (**Figure 4.8**, left panel) and could lead to increased resistance by selection if the population is challenged by an antibiotic. However, mutations are relatively rare, and resistance would develop slowly if bacteria had to rely on inheriting a rare mutation from their parents. A major reason for the rapid spread of resistance is that bacteria can also acquire genes by HGT. This refers to genetic material being transferred from one cell to another that is not its descendant (Figure 4.8, right panel). For example, many antibiotic resistance genes are carried on plasmids: small, circular, independent DNA molecules. A cell with a resistance plasmid can often transfer that plasmid to nonresistant cells around it, so that the

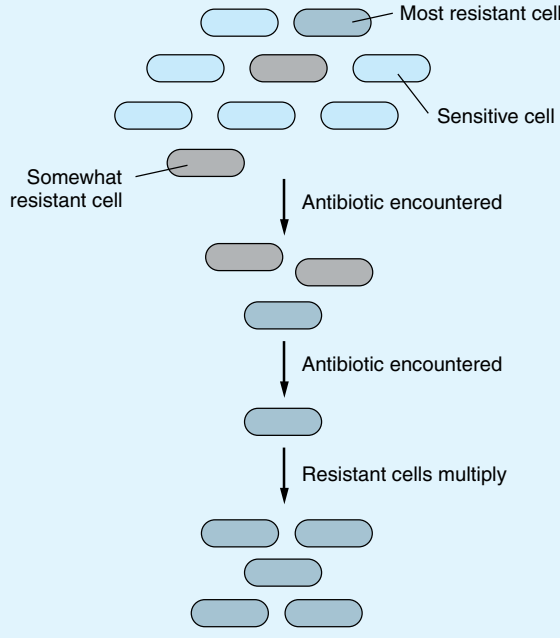


Figure 4.7 How exposure to antibiotics selects for the survival of resistant cells in a population of bacteria.

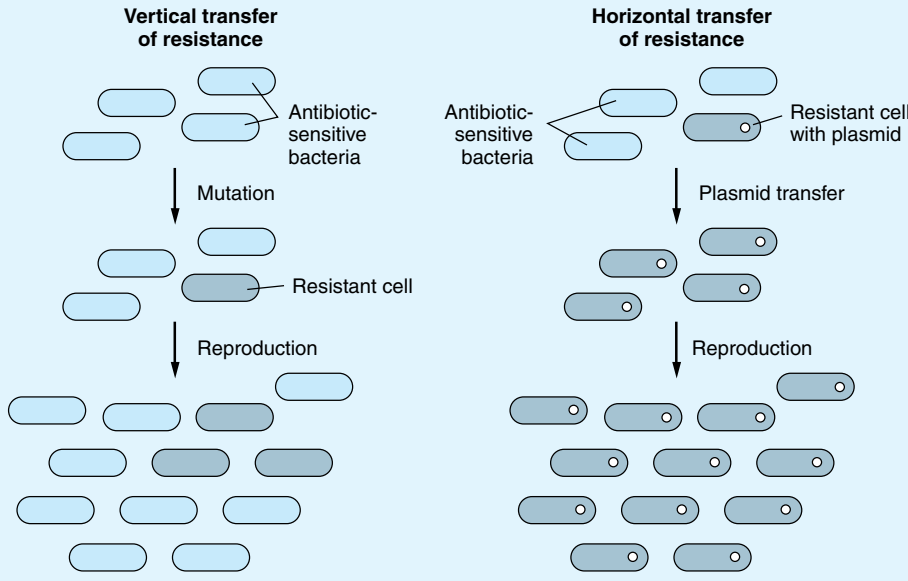


Figure 4.8 Vertical gene transfer occurs when a cell passes a resistance mutation to its offspring (left); horizontal transfer from cell to cell (right) allows much faster spread of resistance.

resistance gene is passed not only to a cell's descendants but to its peers and to their descendants. Depending on the circumstances, this transfer could occur by cell-to-cell contact (conjugation), by means of a bacterial virus (transduction), or by direct uptake of DNA released into the environment (transformation). Antibiotic resistance genes are also often found within transposons, semi-independent DNA sequences that can move within a genome, further promoting their mobility.

As discussed in the preceding chapter, when the sequences of two genes are similar, we conclude that they have a common origin; furthermore, we assume that highly similar genes diverged from that common origin only recently and have not had much time to evolve independently. Two very similar sequences found in dissimilar organisms—those that do not have a recent common ancestor—suggest that HGT has occurred: The gene evolved in one species but was then transferred intact to another relatively recently, so there has been limited opportunity for mutation.

References and Supplemental Reading

Original BLAST Algorithm

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. Basic local alignment search tool. J. Mol. Biol. **215**:403–410.

Modified BLAST Algorithms

Altschul, S. F., T. L. Madden, A. A. Schaeffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**:3389–3402.

Importance of BLAST

Harding, A. 2005. BLAST: how 90,000 lines of code helped spark the bioinformatics explosion. The Scientist **19**(16):21–25.

ClustalW

Thompson, J. D., D. G. Higgins, and T. J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. **22**:4673–4680.

HGT Between Agricultural Bacteria and Human Pathogens

Salyers, A. A., A. Gupta, and Y. Wang. 2004. Human intestinal bacteria as reservoirs for antibiotic resistance genes. Trends Microbiol. **12**:412–416.

Shoemaker, N. B., H. Vlamakis, K. Hayes, and A. A. Salyers. 2001. Evidence for extensive resistance gene transfer among *Bacteroides* spp. and among *Bacteroides* and other genera in the human colon. Appl. Environ. Microbiol. **67**:561–568.