

Bacterial phylogenetics: polymerase chain reaction, molecular clocks and the tree of life

Seppe Kuehn

February 27, 2020

1 Introduction

Since Darwin's seminal text *On the Origin of Species*, it was thought that all life was connected through evolutionary descent. It wasn't until more than a century later that Carl R. Woese, working at the University of Illinois at Urbana-Champaign, that we got an unmistakable picture of that tree. The picture that Woese generated in the late 1970s revealed a previously unknown domain of life the *Archaea*. It is the belief of the instructor that Woese's biggest contribution was not to expose the "true" tree of life (Fig. 1), but instead was figuring out the right way to look at evolutionary history by mining the fossil record held in DNA. In this module you will use Woese's method to study the evolutionary history of bacteria. We'll amplify the same gene in four or five different bacterial species and send the amplified DNA for sequencing. You will then analyze the resulting sequence data using Matlab to answer a simple question – which of the five species are most closely related? Mathematically we'll need to learn how to compute distances between DNA sequences and construct trees from those distances using clustering methods. Let's start with some biological background.

1.1 Background: biology

One remarkable feature of evolution is the presence of conservation. Conservation simply means that biological systems reuse the same parts over and over again, across even very diverse species. One such example is the triplet-code. Every three letters in our DNA alphabet encode one amino acid in a protein (e.g. a protein with 100 amino acids is encoded by a gene with 300 nucleotides (A,C,T,G)). It turns out that all life on Earth uses the same encoding for mapping DNA to protein. We say the code is conserved.

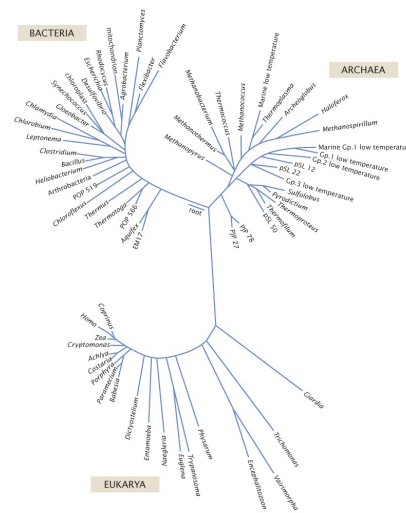


Figure 3.6 Physical Biology of the Cell, 2ed. (© Garland Science 2013)

Figure 1: The tree of life. A tree constructed with methods similar to those developed by Woese. Note the three large clusters constituting the three domains of life. (borrowed from Phillips et al.)

which of the five species are most closely related? Mathematically we'll need to learn how to compute distances between DNA sequences and construct trees from those distances using clustering methods. Let's start with some biological background.

A related conserved feature is the molecular machine inside all cells that actually makes proteins from DNA (see Fig. 2). The ribosome, which actually makes proteins from ‘messenger’ RNA or “mRNA” which is itself a short-lived copy of DNA, is the evolved machine that gets this job done. Since all life is built on proteins that are encoded in DNA with the same triplet code, all life also relies on ribosomes to build proteins. The ribosome is highly conserved, meaning we all use a very similar machinery for constructing protein from mRNA. The ribosome is itself a big complex molecular machine built out of both protein and RNA that is structural. The ribosome has many subunits that come together to make a functional ribosome. The detailed study of the assembly and function of this remarkable machine is the subject for another time (or course!).

Woese’s insight was to realize that the ribosomal subunits contained regions that were both highly conserved and quite variable. In essence, some parts of these molecules must be the same for the ribosome to work and some parts can freely vary due to mutations without breaking the ribosome. By realizing this Woese recognized that the sequences of the genes that encoded the ribosomal subunits could act as molecular clocks! In essence, some parts of the genes that encode the ribosome (and all organisms have these genes) evolve relatively quickly. We call these regions ‘hypervariable’ regions of a gene (Fig. 3). What that means is that if you look at two sequences, make an assumption about the rate of mutations occurring in the gene, you can use these regions as clocks to measure the evolutionary time that elapsed between any two species. To be clear, the idea of a molecular clock had been around for some time when Woese did his work, but he applied the idea to a ‘universally’ conserved gene and that led to his broader insights and new tool development.

1.2 Goals for the experiment

Your goal is to construct a phylogenetic (the evolutionary relationships between organisms) tree for a set of different bacterial species that we supply for you. The questions we would like you to answer are: of the six bacterial species we have given you: which two are most closely related? Which two are most distantly related? Using a larger library of 16S sequences supplied to you, can you find a ‘hypervariable’ region of the 16S gene? How phenotypically different are strains that are closely related? To accomplish this you will need to:

- Perform polymerase-chain reaction (PCR) on the 16S ribosomal RNA gene on each of a set of six bacterial species we give you. PCR acts to amplify a specific target sequence in the genome using ‘primers’ and gives you a test tube with a large number of DNA fragments (short segments) that can then be used for further analysis.
- Send your PCR product for Sanger sequencing by a local facility. (The actual sending will be done by your TA.) Sanger sequencing reads the letters on the 16S gene for each of your six species.

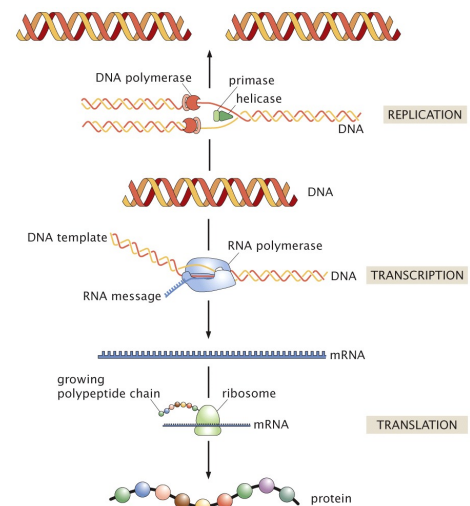


Figure 3.12 Physical Biology of the Cell, 2ed. (© Garland Science 2013)

Figure 2: The central dogma of molecular biology. DNA is replicated during cell division. RNA polymerase transcribes DNA into mRNA which is then translated by the ribosome into protein.(borrowed from Phillips et al.)

- Analyze the sequences by:

(1) Assemble forward and reverse Sanger reads (we'll explain what this is) to get a consensus sequence for each species' 16S gene. You will use existing software for this and we will not discuss the inner workings of this process. The outcome will be six 16S gene sequences, one for each of the species we gave you. (Ugene or Matlab).

(2) Align the sequences – essentially line them up so that they can be compared. Alignment is an important bioinformatic process, but in the interest of tractability we will not discuss the details of how this is done. We can discuss this process or you can refer to Phillips p. 966 for technical details.

(3) Compute a distance between each pair of 16S genes. The point is to ask: how many mutations do you expect were required for the 16S gene of one species to evolve into the 16S gene of another species? This number of mutations is a 'genetic distance' which measures how far apart these two species are in evolutionary terms. For this we will define and compute a quantity called the Jukes-Cantor distance which is derived below. The result will be a symmetric 6 by 6 matrix with entries that are the genetic distance between each pair of strains. (Matlab)

(4) Use the Jukes-Cantor distance to build a phylogenetic tree. To accomplish this we will use a technique called unweighted pair group method with arithmetic mean (UPGMA). The technical details of this method are discussed below. The outcome will be a tree depicting the evolutionary relationships between species. The length of branches on this tree will correspond to the (expected) number of mutations separating each species! You will then inspect this tree to answer the questions above.

- Characterize your library of strains phenotypically by testing for resistance to antibiotics and the ability to grow on different carbon sources. What do you find? Are strains that are "close" in terms of their genetic distance phenotypically identical in terms of the carbon sources they use and the antibiotics they can resist? Can you speculate why?
- Use Matlab to perform an identical analysis to the one you performed above on a larger library of 67 bacterial strains. These can be found in the google document folder link from the website. You will use these data to locate the hypervariable regions in the 16S gene (see **Extended experimental investigation** below).

1.3 Experimental details

1.3.1 The basics of DNA

You need to have some understanding of how DNA is structured to do this project. Here I sketch what you need to understand, but it will be necessary for you to follow-up reading this section with some of the supporting materials supplied in the google doc which give you a deeper look at DNA, PCR and Sanger sequencing. DNA is present in nearly all organisms (neglecting some viruses) as a double stranded helix that you are probably familiar with. DNA is made up of deoxynucleotides of which there are four: A,T,G, and C. Each base on one strand pairs with a base on the other strand with A pairing with T and G with C. The chemical structure of these bases gives each strand of DNA a directionality which is termed 5' ("five prime") to 3' ("three-prime") direction or the converse. As mentioned above within a gene sets of three base pairs in a row form what is called a "codon" which encodes the amino

acid of the protein product of that gene (Fig. 2). For example, the codon CAG encodes the amino acid glutamine. Genes are typically 1000-5000 base pairs and they live on chromosomes which are between a few million (bacteria) and a few billion (human) base pairs long. As an aside, bacteria have on the order of a few thousand genes and humans about 20,000 genes.

As discussed above, one of the shared genes across all of life is a gene that encodes a structural subunit of the ribosome. This gene (16S rRNA gene) is about 1500 base pairs long. You are going to sequence this gene in each of the five different bacterial species supplied to you. To sequence a gene, you first need multiple copies. The trick we will use to accomplish multiplying the rRNA is called polymerase chain reaction or PCR.

1.3.2 Polymerase chain reaction:

PCR revolutionized biology in the 1980s by making it possible to amplify (create many copies) of any part of a chromosome in the genome in a sequence specific manner. These could then be used for further analysis or genetic engineering. Modern biological experiments would not be possible without PCR, which is typically credited to Kary Mullis. The key insight was the finding of a thermally stable DNA polymerase that could function at high temperatures ($>60^{\circ}\text{C}$). PCR involves the following steps (see Fig. 5). First the DNA of interest is heated to high temperatures and the double helix melts (comes apart). Two primers, short segments of single stranded DNA which are complementary to a specific section of the larger melted DNA molecule, are included in the reaction vessel. After melting, the temperature is dropped to $\sim 50^{\circ}\text{C}$ or so and these primers anneal (bind to the larger DNA molecule) in the specific place where they are complementary. These primers are typically 10-20 bases long specifying a unique location on the genome. Once these primers bind then the temperature is raised to the temperature where the polymerase can synthesize new DNA strands. The process is repeated through many cycles driving a chain reaction which yields exponential growth in the concentration of the amplified fragment. This gives you a tube full of many double stranded fragments of the sequences specified by the primers. I suggest you watch the YouTube video linked to in the supplementary google doc for an animation.

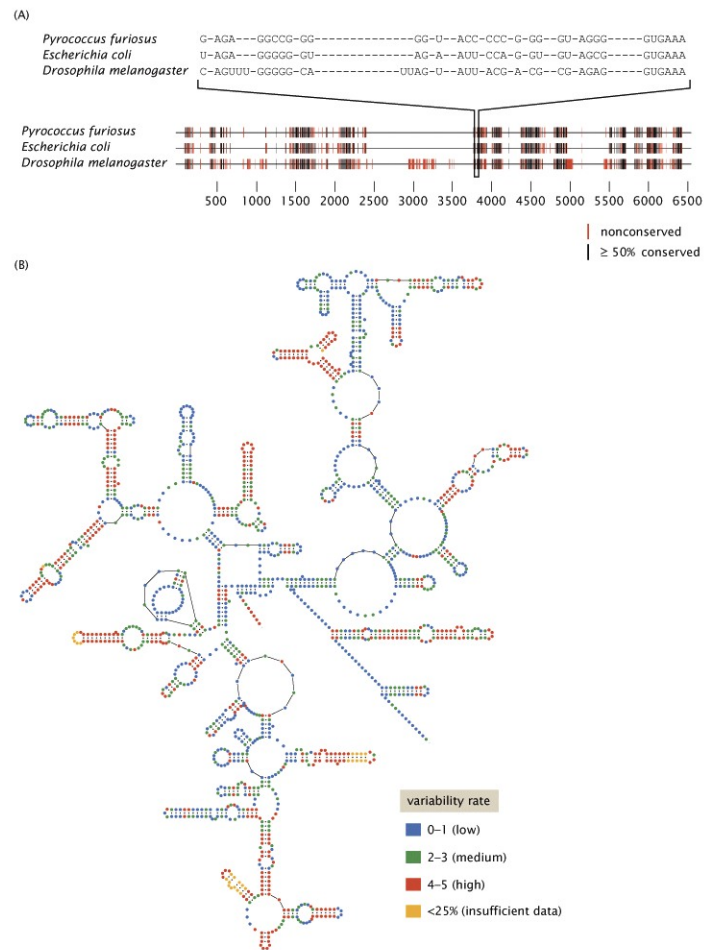


Figure 21.32 Physical Biology of the Cell, 2ed. (© Garland Science 2013)

Figure 3: The secondary structure of the 16S rRNA ribosomal subunit. Note the conserved and variable regions of the gene as noted in the legend. The loops arise due to RNA (which is single stranded) pairing with itself. (borrowed from Phillips et al.)

Polymerase chain reaction - PCR

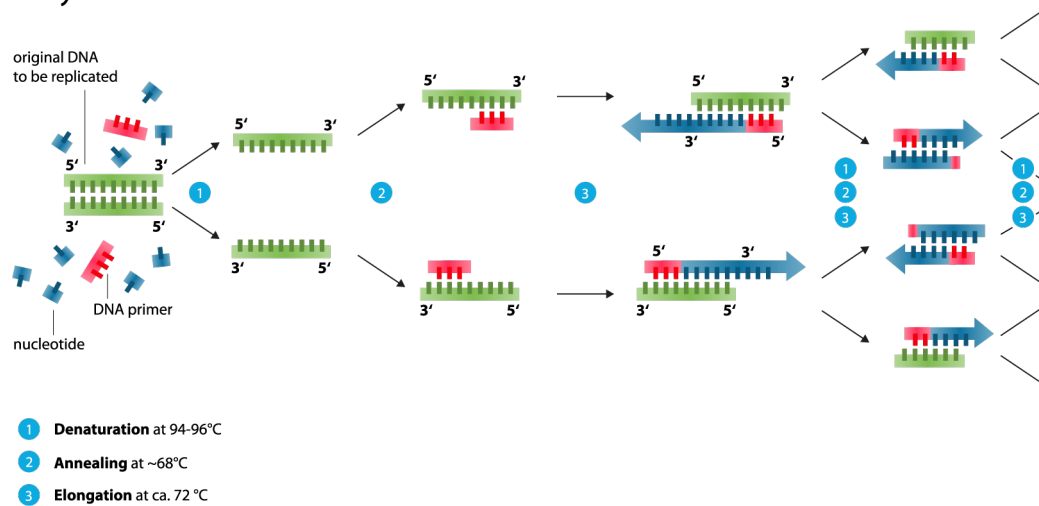


Figure 4: Schematic of PCR (taken from wikipedia). An overview of the PCR process.

We will give you primers that can be used to amplify only the 16S gene in any bacteria – these are universal primers which means they work on any species or strain. When you run the PCR reaction using these primers, you will get a tube full of DNA fragments of ONLY the 16S gene. The next step is to send these off for sequencing.

1.3.3 Sanger sequencing

The next step is to read the sequence of A, T, C, G on each 16S gene that you have amplified for each species (see top of Fig. 3 for an example). We will use the gold standard method for this which is called Sanger sequencing. I will not discuss this process in detail, but I again ask that you watch the supplied YouTube video to understand the fundamentals of how it works. Sanger sequencing uses primers to create forward and reverse reads of the DNA strand of interest.

Be sure to watch the supplied YouTube videos. There you will learn that the Sanger sequencing process actually gives you TWO ‘reads’ (sequences) for a given run. The ‘forward’ and ‘reverse’ reads are complementary (e.g. the pair A with T and G with C). The forward read is the one with the non-template (coding) sequence and the reverse with the non-template sequence. The Ugene software takes these two reads and assembles them into a single consensus sequence for that gene. The result is a measured sequence for each 16S gene of the bacteria we gave you.

Once you receive your data from the Sanger facility you will need to assemble the forward and reverse reads into a single consensus sequence. This can be accomplished using the Ugene platform and the instructions in the associated google doc. Export these sequences as .txt files for subsequent analysis in Matlab.

1.4 Phenotypic characterization

Your next experimental goal is to ask: how phenotypically similar are your bacterial strains? To do this we will perform a simple experiment where we test each strain for its ability to grow in the presence of antibiotics and their ability to utilize different sources of carbon for growth. In brief, you will create an array of wells in a 96-well microtiter plate where columns are media containing either an antibiotic or different sources of carbon. Each row of the plate will be inoculated with one of your isolates. You will

then test for growth by measuring optical absorbance. **Please present your findings in your writeup.** What antibiotics can be tolerated by which strains? Is there a pair of strains that are genetically "similar" in terms of 16S (low Jukes-Cantor) but phenotypically quite distinct? How could this be?

Please consult the protocol provided for this experiment.

1.5 Data analysis

Once the Sanger sequencing results are compiled you will need to analyze these sequences to build a phylogenetic tree. There are three problems that must be solved to accomplish this – aligning sequences, measuring distances between sequences, and building a tree. We will discuss the last two only. Sequence alignment, while important, is a course in itself, so you will need to take it on faith that this can be done reliably. It is far from a trivial problem, and there are resources in the associated google doc for you to explore this process. Here is an overview of the Matlab code you will use:

- You will need to import your sequences from the .txt files you created using the Ugene assembly tool.
- For alignment you will need to use the Matlab function 'multialign'. Be sure to read the documentation carefully.
- The next task is to compute a distance between your aligned sequences in terms of expected number of mutations. See the detailed discussion below regarding estimates of this, and use the Matlab function 'seqpdist' to compute the Jukes-Cantor distance between all of your sequences.
- Build a tree. You will use the function 'seqlinkage' to perform the UPGMA method for building phylogenetic trees. See below to understand how this works. View the output using the 'view' function.
- **Extended computational investigation.** You are supplied with a .mat data structure which contains the 16S sequences for 67 bacterial isolates. Perform the same analysis above again and answer the following questions:
 - Each bacterial strain is given a genus and class level name. Genus is one level higher than species, class a few levels higher yet in taxonomic rank. How many classes are represented in your sample of 67 strains? Do different classes separate on the tree you build for these strains or not?
 - Detecting hypervariable regions. For the sequence alignment you performed on these 67 strains find the hypervariable regions of the 16S gene. To do this determine the frequency of each of the 5 possible letters (ATCG + gap) at each position. At each position then compute the frequency of the most common letter (f_c). If this frequency is high the variation at that site is low and the converse (check!). Plot, along the 16S sequence, $(1 - f_c)$, it looks noisy right? Smooth these data with a moving average, can you see the hypervariable regions? How many are there? How many did you expect?

1.6 Computing distances between sequences

Having successfully aligned sequences our next task is to ask the following question: given two 16S gene sequences, how many mutations (changes in the A,T,C,G string of letters) were necessary to change on sequence into the other? In other words, what was the sequence of the gene in the ancestor of these two species? The question is not an easy one! We will make a several approximations to get at an answer.

You might think it is a good idea just to go down the list of letters and simply count every time one sequence differs from the other sequence. This approach underestimates the true number of mutations in cases where multiple changes happened at the same site (location on the string). For example, imagine the two sequences have an 'A' at a given position. It is possible that in one species the letter changed from an A to a C and then back to an A again! Meaning that there would have been two mutations at that position even though the sequences match!

The way to deal with this problem is to have a model for the evolutionary process and then to use that model to estimate the number of changes between two sequences. I will walk you through the simplest such model which is implemented in the Matlab function 'seqpdist'.

1.6.1 Jukes-Cantor distance

Consider just one site on two aligned sequences. Suppose that the two sequences have the same letter at that site. Did the ancestor which both sequences share have that letter? Or did they both mutant from a different ancestral letter. For example, say both sequences have a G. Was the ancestor also a G or instead a C which was later mutated to a "G"?

First assume that the mutation rate is λ and is stable in time. Secod, assume that the two sequences of interested are separated by some time Δt . Neither λ or Δt are known! So what we need to do is propose a model for the rates of mutation and then use that model to infer λ and Δt and therefore an estimate for the actual number of mutations that occurred during the evolutionary process separating the two sequences of interest.

Assume all mutations (e.g. $A \rightarrow C$, $G \rightarrow G$etc) are equally likely and happen with a rate α per unit time. This means that a site has probability $1 - 3\alpha$ of not mutating per unit time. This assumption is demonstrably false, but makes the calculation straightforward. Define a probability that a given site has a given letter at a given time as $\phi_i^{(t)}$. Given the assumptions above we can write down the following equation

$$\phi_A^{(t+1)} = (1 - 3\alpha)\phi_A^{(t)} + \alpha\phi_G^{(t)} + \alpha\phi_T^{(t)} + \alpha\phi_C^{(t)}. \quad (1)$$

This simplifies to

$$\phi_A^{(t+1)} = (1 - 3\alpha)\phi_A^{(t)} + \alpha(1 - \phi_A^{(t)}). \quad (2)$$

and this applies to any letter (A,C,T,G) due to the symmetry in the mutation rates. So say $y \in \{A, C, T, G\}$ then

$$\phi_y^{(t+1)} = (1 - 3\alpha)\phi_y^{(t)} + \alpha(1 - \phi_y^{(t)}). \quad (3)$$

Which further simplifies to:

$$\phi_y^{(t+1)} = (1 - 4\alpha)\phi_y^{(t)} + \alpha \quad (4)$$

Now, subtract $\phi_y^{(t)}$ from both sides and simplify to get

$$\phi_y^{(t+1)} - \phi_y^{(t)} \approx \frac{d\phi_y^{(t)}}{dt} = \alpha(1 - 4\phi_y^{(t)}) \quad (5)$$

This differential equation has the following solution (check it!)

$$\phi_y^{(t)} = \frac{1}{4} + (\phi_y^{(0)} - \frac{1}{4})e^{-4\alpha t} \quad (6)$$

What does this mean? This is an equation which gives us a probability of a given site being y at some time t and that probability depends on the probability that the site was y at time $t = 0$ or not! (e.g. whether $\phi_y^{(0)}$ is zero or one).

So the probability that the site did NOT change between time 0 and time t is just (setting $\phi_y^{(0)} = 1$)

$$p_{yy}(\Delta t) = \frac{1}{4} + \frac{3}{4}e^{-4\alpha\Delta t} \quad (7)$$

and the probability that the site did change is just

$$p_{yx}(\Delta t) = \frac{1}{4} - \frac{1}{4}e^{-4\alpha\Delta t} \quad (8)$$

Note that the equation above has a limiting value of $1/4$ as t goes to infinity. This reflects the fact that the symmetry in mutation rates means that all four letters are equally likely, so the chance that the site ends up being y if it started as $x \neq y$ is just 25%.

So, now we must ask ourselves – under the assumptions of this model, what are the expected (average) number of mismatches between any two sequences? What is the probability of a match (P_M) after some time Δt ?

$$P_M = (p_{AA}^{(\Delta t)})^2 + (p_{TA}^{(\Delta t)})^2 + (p_{CA}^{(\Delta t)})^2 + (p_{GA}^{(\Delta t)})^2 \quad (9)$$

How do you think about these terms? Well, for one sequences there is probability $p_{AA}^{(\Delta t)}$ that it was A a time Δt ago and is *still* A . Same probability for both sequences! (hence the square). But we just computed precisely these probabilities so we know that:

$$P_M = p_{yy}^2 + 3p_{yx}^2 \quad (10)$$

plugging and chugging on this expression gives us a probability of getting a match at a given site of:

$$P_M = \frac{1}{4} + \frac{3}{4}e^{-8\alpha\Delta t} \quad (11)$$

and we know then that $P_{mis} = 1 - P_M$ (probability of a mismatch). Now, we can estimate P_{mis} by just counting the number of differences between the two sequences and dividing by the length of the sequences! Say there are m mismatches in sequences of length n , then $P_{mis} = m/n$.

But what we wanted at the beginning was an estimate for the number of mutations that happened as these two sequences diverged over evolutionary time. We wrote down that this should be $2\lambda\Delta t$. Note now that the probability of each site mutating per unit time is $\lambda = 3\alpha$ so then the **expected number of mutations in a time $\Delta t = 6\alpha t$** . So now we need an estimate of $6\alpha t$! This can be acquired with some algebra as follows. First realize that P_{mis} can be simplified to

$$P_{mis} = \frac{3}{4}(1 - e^{-8\alpha\Delta t}) \quad (12)$$

which means that

$$e^{-8\alpha\Delta t} = (1 - \frac{4}{3}P_{mis}) \quad (13)$$

and then

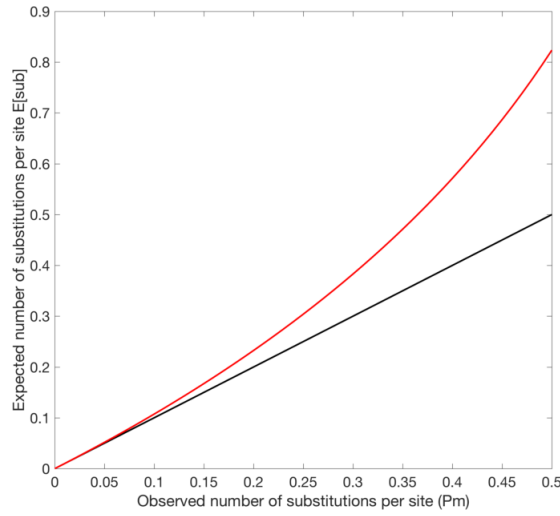


Figure 5: Jukes-Cantor estimate for number of mutations as a function of the observed number of mismatches. Red curve is $E[sub]$ and black line is m/n . So as the observed number of differences between sequences increases so does our expectation for the number of mutations that occurred during the evolutionary process separating those two sequences.

$$-8\alpha\Delta t = \log_e\left(1 - \frac{4}{3}P_{mis}\right) \quad (14)$$

which can be re-written as

$$6\alpha\Delta t = -\frac{3}{4}\log_e\left(1 - \frac{4}{3}P_{mis}\right) \quad (15)$$

So this means that the expected number of mutations that occurred between the two sequences is just:

$$E[sub] = -\frac{3}{4}\log_e\left(1 - \frac{4}{3}\frac{m}{n}\right) \quad (16)$$

Wow! We made many assumptions, but we have a closed form solution to the question posed at the outset. What does this answer mean? The best way to answer that question is with a simple plot where we compute the expected number of mutations as a function of the ratio m/n . This means that if I am given two aligned sequences and I simply count up the number of mismatches m and the number of sites then I can plug those numbers into the equation for $E[sub]$ to get my estimate. Note that $E[sub]$ is bounded by 0 and 1, and the correct interpretation of this object is the average number of substitutions per site in the gene of interest. This is crucial so that the estimate does not depend on the length of the gene being compared (n).

Use the 'seqpdist' function in Matlab to compute this for all pairs of sequences that you have performed Sanger sequencing on. The result will be a symmetric matrix which has as entries the expected number of mutations per site. Prof. Kuehn or your TA Laura can assist you with putting this matrix together – take a look! Can you find the two sequences with the greatest distance in terms of the per site number of mutations.

1.7 Building trees by UPGMA

We now arrive at the final step of the data analysis which is to build a phylogenetic tree. I suggest you watch the youtube video supplied the ancillary materials before using Matlab to build a tree by UPGMA. The algorithm is quite simple and works by just joining neighbors together in a tree based on their Jukes-Cantor distance. Each time two sequences are joined, we recompute their distance to all other nodes by taking the average distance each joined sequence has with all other sequences.

Use the 'seqlinkage' function to build a tree using the sequences you aligned and measured distances between. **It is imperative that you carefully read the online documentation Matlab provides for these functions.** If you have trouble implementing the analysis please consult us, it is not complex and should take no more than about 20-30 lines of code.

View the tree using the 'view' function. Answer the questions above.