First, I`m glad you’ve asked these questions (and future ones). It helps me justify why we do it like this—or if not, how to improve it. It’s always good to have fresh eyes looking at the problem and the goals.

Questions:

How to BLAST-align things? Is there an existing software I should download?

There are several levels to this, and in the end I am flexible as to how we get there.

First, I am presuming (hoping?) that the idea behind this is that a user with a reasonable computer/laptop will run this remotely while connected to the internet.

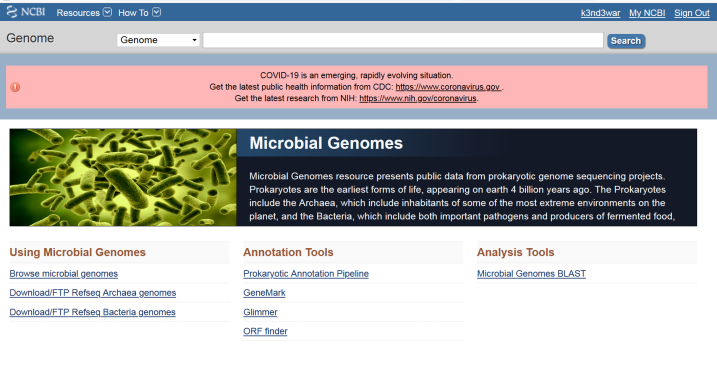
Second, like before (Genome Synchronizer), the user’s computer downloads the “instructions” from us, but the computations are done and files saved on their machine.

Third, we do this in a two-step process (find the best species, then find the best strain) and this might be partially or fully redundant. As we work through it, you might wish to convince me that a one-step process is easier or better.

Fourth, we use BLAST because of its generally universal familiarity. However, it has a weakness in how it calculates its final alignment score (taking an objective alignment score and adjusting it with a subjective penalty due to database sizes). Ideally, I would prefer something like BLAT, where a perfect alignment score can be accurately calculated based on solely sequence length.

Module One

The idea for Module One is to scan all species with finished genomes to identify which species are present in our sequence data. It becomes a choice of which NCBI database to use. NR/NT is not optimal since it is the largest database (compute effort) and the results are a mixture of finished genomes, unfinished genomes, genes, fragments, etc. I am not sure which database Matt uses by command line, but I use Microbial Genomes BLAST (<https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch&BLAST_SPEC=MicrobialGenomes> accessed from (<https://www.ncbi.nlm.nih.gov/genome/microbes/>) (shown below).

I don’t particularly like it, since the database is restricted to microbes. For instance, it wouldn’t indicate high levels of host contamination (i.e. human). Nonetheless, it does answer the question of whether microbes are present, and which ones.

There is also the question of which flavor of BLAST to use, but since we are concerned with finding the most similar species, I use a stringent BLAST (i.e. MegaBLAST).

In the first stage, I am not interested in the alignment per se, but moreso the informativity of the best hits in identifying a species.

For example, with the three fasta reads

>1

AACTCCTCTACCAAGTACGAAGGCGGCAACAAAGGCCATCGTCCGGGCGTGAAAGGCGGTTATTTCCCGGTTCCGCCAGTTGACTCCGCGCAGGATATCCGTTCTGAAATGTGTCTGGTCATGGAGCAGATGGGCCTGGTCGTTGAAGCGCACCACCACGAAGTAGCGACCGCAGGTCAGAACGAAGTGGCGACCCGCTTTAACACCATGACCAAAGATCGGAAGAGCACACGTCTGAACTCCAGTCACT

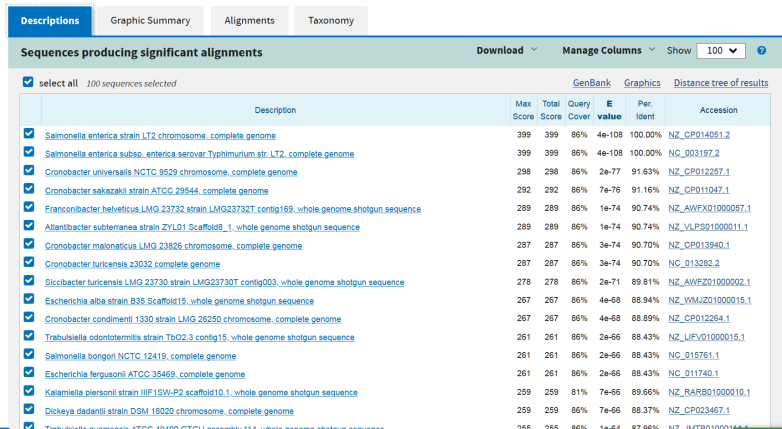
>2

TTCAGAACGTAACGTAGCGAATCTTTATATGACTGAAAGGGACTTATTTATGAAAGTAAAACTGCTTGCTGCCGGTATTTTGTTCACGCTGCCGTTCTGGGCCTGCGCCAAAGATGTCACCATTATTTACACCAACGATCTACACGCCCATGTGGAGCCTTATAAAGTACCGTGGATTGCTGACGGTAAACGCGATATTGGCGGTTGGGCCCATATCACCACGCTGGTGGAGCAGATCGGAAAAGGACAC

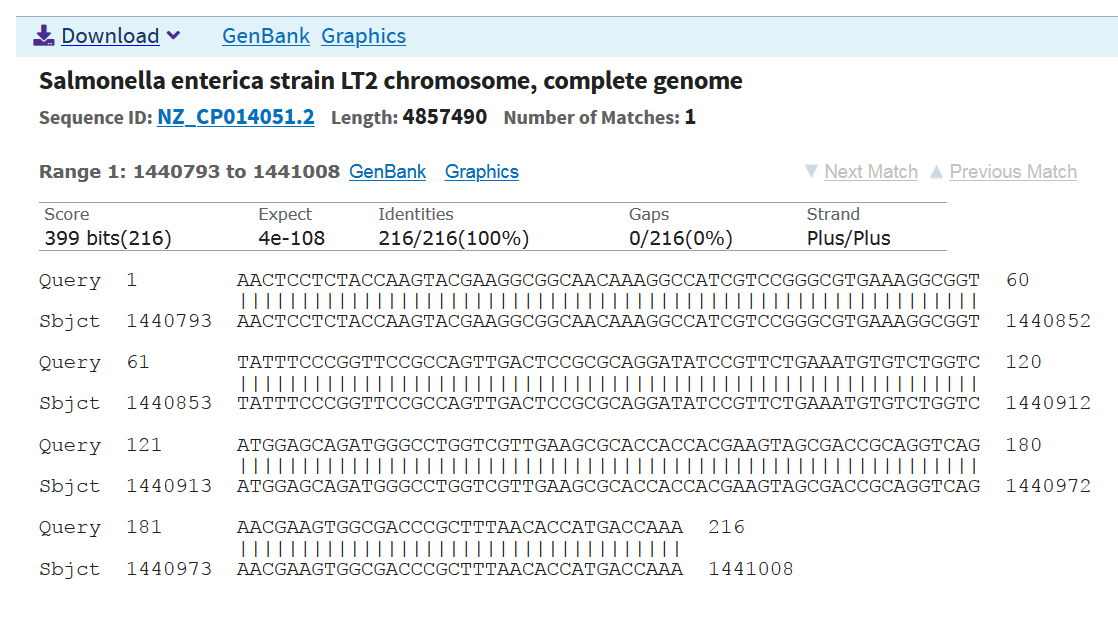
>3

AAAATCACCATTATGCTGCTACCTGTTTTGTGCAGCGATGTCAGCTTCATCAGCCGCGCTGGCTGCTACCGCGCCGGTCTCCGCAGGCGTTATTCATTTCAAAGGCCAGATTGTCGAATATGGTTGTAATCTTGCGCCTCACGATCGCAATATCGAGGTATCATGTTTGCGGAATAATATCCCCCATTTGCAGACAGTCGCCACACCGTCTGGAAACGCTATTTCGTTGATGAATGGAATCGCCACGGTA

Read >1:

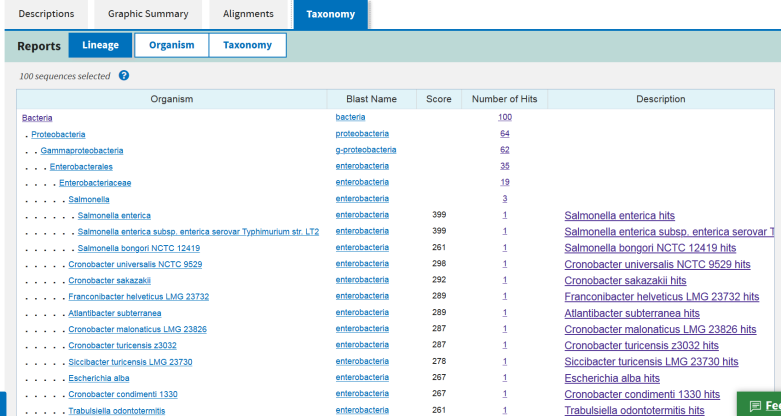


The “Max Score” in the description of results tells us most of what we need to know. There are two top hits (*Salmonella enterica* and *Salmonella enterica* serovar Typhimurium), both tied at 399, and then a set of descending scores to other things.



Looking at the alignment, we see that the sequence was quality trimmed (beforehand) to 216 nt, but it matches perfectly to *Salmonella*.

(My pet peeve is that the score of 399 doesn’t indicate a trimmed sequence, nor a perfect match.)



Looking at the phylogenetic display, we see that the other hits are to species related to *Salmonella*. Overall, we see we have a shortened sequence that matches *Salmonella enterica* perfectly, and is for a locus that has evolutionary conservation beyond *Salmonella*.

We don’t really need all this information at this stage, and can work from the first result.

Read >2: Has a highest Max Score of 424 to the same as Read >1, and declining scores to other things. It is a longer read (235 nt) but is not perfectly matching (2 mis-matches near the end).

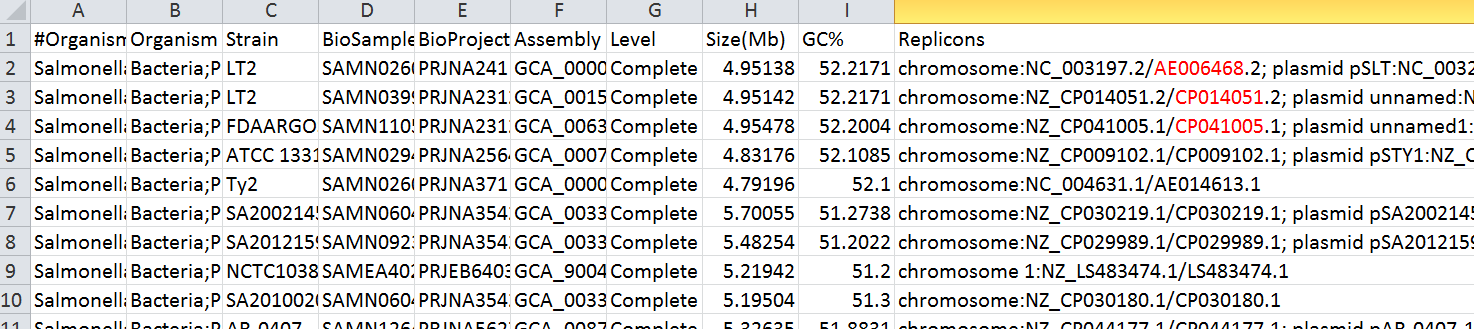
Read >3: Highest Max Scores (462) to the same, and no other hits (i.e. a locus specific to *Salmonella*). It’s a perfectly matching full-length read (250 nt). (Pet peeve again, it is not obvious knowing beforehand that a perfectly matching full-length read should have this score. The corresponding BLAT score (250) is much more interpretable.)

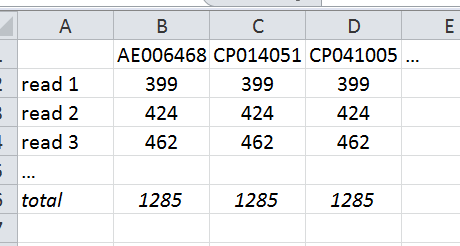
Cycling through 1000 sample reads would tell us that we have a high proportion of *Salmonella enterica* in our data. The outputs of Module One would be (i) the 1000 read fasta .txt file; and (ii) a summary report estimating whether we have enough reads and coverage to move forward. One part of the report would include the total number of reads, total number of nucleotides, and a histogram of read lengths. The other part would include the species identified (i.e. *Salmonella enterica*); its genome size (based on the representative genome, i.e. 4.856 Mb); the number of reads identifying it (3 of 3 here), and their aggregate length (216 + 235 +250 = 701 nt here); then an extrapolation to the number of total reads and their aggregate length; and an estimation of total coverage available (extrapolated aggregate length / genome size).

Module Two

Since we have tie for the best species, we would have to take the least narrow choice (i.e. *Salmonella enterica* and not *Salmonella enterica* serovar Typhimurium). Returning to the NCBI Microbial Genome resource (<https://www.ncbi.nlm.nih.gov/genome/microbes/>) the user would search for *Salmonella enterica* and realize that there are currently 11,132 sequenced genomes, of which 874 are complete. The user would decide whether to use all or some of the 874 for the next step. (My suggestion is to limit this module to 100 or less, and the user can run multiple lists if they want to do more than 100.)

The user can download the table and create a list of accession numbers (such as those shown in red).



After entering the list and specifying the read set (the same 1000 fasta as previously), each read would be aligned directly to each accession (blast2seq) and the Max Score recorded. While many reads will have identical scores across many accessions, the aggregate total for the 1000 reads will be different and the highest aggregate score will identify the genetically closest individual genome.

The outputs of this module would be a summary table (accession and aggregrate score) and a spreadsheet of the full results.

I'll need an example that I could use to run each step and make sure I've understood everything right. Can you provide me such an example? I will need any necessary data to complete the process (eg input files, which species it maps to, which genomes to compare it to, which genome it maps to, user-provided fixed length for module 3)

I'm not sure I understand Module 3, Step 3. Maybe an example will help.

For Module 3, the overall idea is that Illumina sequence errors increase with read length, and every sequencing run may have slightly different error rates due to reagent batches, run conditions, etc. While we could trim each read individually, this would lead to a population of reads with variable lengths, which makes some aspects of downstream analysis more difficult. Here we try to find the “sweet spot” for each data set, where a fixed read length is chosen that eliminates the most spurious errors. (The idea is that polymorphisms at the ends of reads are more likely to be errors—and if they are not, getting rid of them anyways won’t hurt because other reads will also carry the polymorphisms at more interior portions of the read.)

I am using the first 10 random reads from your file as an example, and for this purpose, pretending accession CP045956 had been previously identified as the genetically closest genome reference.

When all 10 reads (all at 250 nt) are aligned to CP045956, only 2 match perfectly.

When all 10 are trimmed to 245, still only 2 match perfectly.

240: 2 product = 280

235: 2 product = 270

230: 2 product = 260

225: 2 product = 250

220: 2 product = 240

215: 2 product = 230

(Okay, this is not a good example. But if there were enough reads, the product of [(length)\*(number perfect matches)] would be maximized at a certain read length.

The iterative part could be something like trying 250-240-230-220-210 first, and if for example, 220 gave the best results, a further round of refinement could be done with 218-219-220-221-222.

How do you see the interface for this?

First, I like what you did with Genome Synchronizer, so am perfectly willing to profit from your creativity and sense of aesthetics. I’ve made a very simple mock with ppt (attached). The different modules are all independent, in the sense that at whatever stage a set of suitable data files and parameters are uploaded/entered.