Computational Biology Independent Project **Eliotte Garling**
Prof. Anna Ritz Lab Date: 05/02/17

Problem Statement

The development of Next-Generation sequencing (NGS) technologies has proceeded at an unprecedented pace, leading to two major bottlenecks in determining the best way to assemble and extrapolate information from genomes. The first has to do with the genome assembly process, the second is annotation. Both processes are significantly hindered by repetitive sequences, which make up large portions of most genomes. The main motivation of the project is to write a program which will filter sequences into low, medium and high complexity sequences – ideally speeding up the process of genome annotation.

Data Formatting

During the initial stages of the project I used a toy sequence from Homework 5, then implemented a 4090 bp portion of a *Daphnia magna* mitochondrial genome that I previously assembled.

Steps of the Program

The first part of this project is adapted from the frequent words problem of homework 5. A sequence is divided up into k-mers then the most and least frequent are separated. The most

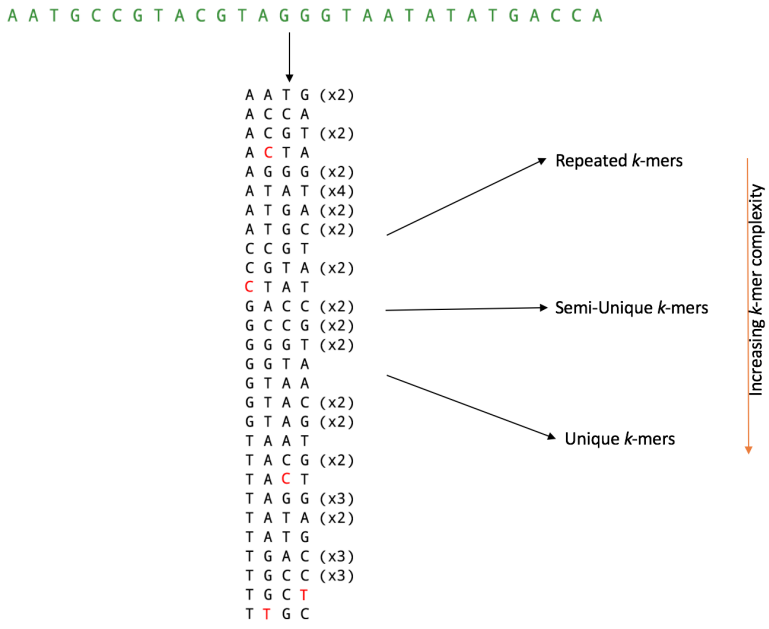

Figure 1: Diagram of the project steps used to organize genomic sequences into k-mers of relative complexity and frequency to improve the genome annotation problem. (Portions of figure adapted from Wikipedia Commons)

frequent presumably, represent retrotransposons or transposable elements (TEs), which are capable of replicating in the genome and have highly conserved sequences. The 'uncommon k-mers', or k-mers, which don't have identical sequences, are put into a frequency table to generate a consensus sequence. The hamming distance of the infrequent k-mers to the consensus sequence is measured, and k-mers are grouped as highly complex if they had little similarity with the consensus sequence, and semi-complex if they had higher similarity with the consensus sequence. The level of similarity can be adjusted, so that the threshold for uniqueness can be determined by the user by selecting a proportion between 1 and 0. Ideally, this program would allow quick access to high complexity sequences, since they should be functional sequences and give a decent idea of the unique functional aspects of the genome. This project was inspired by a paper by Anvar et al. 2014, which described the formation new open-source database called kPAL, which allows for the rapid characterization of functional and repetitive sequences in a genome by analyzing k-mer sequences.

Discussion

Overall the program was successful in it's categorization of k-mers based on the relative frequency, and complexity compared to a consensus sequence. I discovered that automatically formating to a FASTA or FASTQ format would be ideal so that multiple k-mers could be BLASTed at once. However, upon BLASTing a 200 bp k-mer length where sequences with more than 50% similarity to the consensus sequence were filtered into the 'semi-unique k-mer' category. And sequences with less than 50% similarity to the consensus sequence were filtered into the 'Unique k-mer' category. A BLAST search of a k-mer from the repeated k-mer category, was a 97% identity, and had 98% query cover with a *Daphnia magna* mitochondrion genome, and mapped to a portion of the NADH dehydrogenase subunit 5 gene, a highly conserved gene in the mitochondria. A BLAST with a unique k-mer output had a 97% identity and 89% query cover to the *Daphnia magna* mitochondrion genome, and mapped to a portion of the ATP synthase F0 subunit 6 gene.

Interpretation

The largest assumption made with this program is assuming that functional sequences will be considered 'unique' sequences, or should at least possess very low sequence identity with a large portion of genomic sequences. Another assumption is that transposable elements (TEs), if they are still retrotransposable will be highly repetitive within the genome, and TEs which have conferred large portions of mutations rendering them unable to retrotranspose will be semi-unique. However, there are other biological processes at play which may modulate TE transposition and degradation within an organism. Finally, the other difficulty is that TEs range greatly in size, so changing the k-mer length will be critical to parsing out repetitive elements, but this also reduces the efficiency of the program.

Conclusions

Overall, this program is useful in dividing up the genome by lower to higher complexity sequences to allow for some amount of efficient genome annotation. Next steps would be to adjust the format of the output so that they can be automatically BLASTed, and speeding up the efficiency of the code so that it runs faster, especially with larger strings.

Biblography

Anvar, S. Y., Khachatryan, L., Vermaat, M., van Galen, M., Pulyakhina, I., Ariyurek, Y., Laros, J. F. (2014). Determining the quality and complexity of next-generation sequencing data without a reference genome. Genome Biology, 15, 555. https://doi.org/10.1186/s13059-014-0555-3