**Multi-Genome Alignment Pipeline Manual**

1. **Introduction**

Here I will discuss the general use of the pipeline as well as background information that is recommended before use. This section also will talk about limitations and uncertainty of the output. I will also include a citation.

1. **User Support**

This section will contain information on potential issues and how to get support(contact info/ designated email?). I will also link to 3rd party manuals of software incorporated in the pipeline (RAxML, MUMmer, etc.). Potentially also reference to which .py files can be modified if need be.

1. **Download and System Requirements**

This section will discuss the recommended systems that this software can be run on(linux clusters). Similarly, I will try to extrapolate computing times and suggested cores to be used per job based on # and size of genomes.

Sub section will talk about download links and versions of the pipeline as well as incorporated 3rd party tools. I predict this will be a very difficult section to write based on compatibility of software. This will also include how to specify download directories for all necessary software.

1. **Input Files**

This section will discuss the file types required by the software. This includes the annotated genome(.gff), the genomes to be compared(.fasta, .fna or .fa) as well as the necessary elements of the userConfig.txt file.

There will also be a subsection on file locations and directories.

1. **Config File and Optional Settings**

This section will discuss all forms of customizability of the pipeline.

* 1. How to use the config file to designate project parameters(name, output dir, threads)
  2. How to “jump in half way” - Some users may have their own alignments or trim fastas in a specific way.
  3. Custom Commands – Users have the option to input entirely custom commands for blast search, blast db, or RAxML tree. Similarly, users can customize single parameters for these commands or the alignment.
  4. Graphic Settings – Users will have the ability to adjust the color scheme and dimensions of the output graphic.

1. **Output Files**

This section will discuss primarily the log file and how to interpret it. The log file provides information about each project/ pipeline run as well as the software versions that were used at the time. This is useful for users in terms of organization as well as for crash reports. I will discuss how to interpret common error messages and link to how to overcome certain failures.

A subsection here will also discuss directory locations and how to find/change output files.

1. **Frequently Asked Questions**