Friday, February 21, 2014

Determining an Architecture

- Profiled NCBI software
  - made benchmark tests
  - use different databases and different length of queries
- Need to determine best benchmark set
  - Might ask NCBI to determine standard benchmark tests
  - They are not able to provide exact statistics
  - longest database is 4 million+ length
  - Can handle 500000 queries daily
  - between one second to several minutes
- Must implement algorithms that take the longest
  - Some strings more and less similar
  - some strings longer and shorter
  - Smaller the word size greater potential for more hits
- Mitrionic method
  - Bloom filter on query
  - Creates some false positives but doesn't remove hits
  - Open-source
- Tasks for next week
  - Fully develop benchmark system
    - add more variance into queries
    - keep same database(s)
    - need much more than 100 queries
      - One perfect hit
      - One hit that varies by a base

  - What is in the NCBI benchmark?
    - sort the queries by difference/similarity, length
- Beef up the benchmark
  - add longer/shorter queries that aren't there yet
  - Add parts to the benchmark that the standard system doesn't do
  - Throw cout statements into it to see when it gets executed

———————————————

- Understand functions that take the longest
  - start with Taylor's original profiling run
  - NCBI Software code review
  - determine stride length/hit length preferences
  - See if software has different options for specifying stride length/word length
  - How do these functions get called

———————————————

- Review papers methods
  - treeblast
  - megablast
  - mercuryblast
- Identify connection between blast methods and functions in profiling
  - information in treeblast paper

———————————————

- Come up with initial architecture for our implementation