Meeting 10/9

- BLAST is a web server for doing sequencing via a web interface on huge data sets
    - Underlying algorithm is optimal sequence alignment
- HMMER also does alignment it uses a HMM
    - Same underlying dynamic programming algorithm as BLAST; prefiltering, then calculate on 1% of the data
- SHIFT from graph algorithms to a focus on the HMMER project - creating a web server
- hmmbuild - not implemented - finds consensus between sequences and builds a HMM, like multiple sequence alignment problem
- hmmsearch - implemented -

Codename **HBaaS**

Heterogeneous BaaS: Bioinformatics-as-a-Service

To create an online BaaS platform, accelerated by GPUs and the Accumulo DB.

We aim to create a web service providing sequence and motif alignment and search for proteins. Our platform will run on a cluster of machchines equipped with GPUs. Each machine will host data as part of the Accumulo distributed database and leverage their attached GPUs to accelerate parallel computation and data retrieval.

Our target users range among physicians working in personalized medicine, forensic scientists working in crime scene identification, epidemiologists working in identification, genealogy consultants working in kinship analysis, biologists working in research generally, and other groups interested in protein model-to-sequence scoring.

We aim to provide two service classes: *hmmsearch*and *hmmbuild*.

In *hmmsearch*, a user provides HMMs (Hidden Markov Models) representing protein sequence motifs, and requests the top-scoring sequences for each model, from a database of protein sequences stored on our Accumulo platform.

In *hmmbuild*, a user specifies a range of sequences and requests the HMM motif that best represents the consensus between the sequences.

We envision the following outline of our service:

1. A user submits a query via our web interface.
2. Our web server backend creates a customized query-and-compute request for the Accumulo database and launches it.
3. In parallel, each machine on the Accumulo database searches through its locally stored protein sequences. The top scoring results are returned to the web server backend.
4. The web server backend forwards the results to display on the web interface.

We will compare our performance to the existing HMMER search application online at http://hmmer.janelia.org/. We aim to show great speedups by leveraging GPUs and the Accumulo DB.

To recap, we chose to work on the HMMER project because it is a concrete application-- we know what we will create and that it is a substantial contribution to the world. It is new.

We're still up for other Accumulo or GPU-enhanced graph algorithms if you find a direction to take them. We shifted our direction because after one month of thinking, we're still not sure what we can do with graph algorithms that is not already done.

Here are some todos for the close future:

[Xuelian, Yao] Master HMMER. Become experts in its use and understand how we use GPUs to accelerate the *hmmsearch* algorithm. Start thinking about *hmmbuild*, which is not yet implemented.

[Eric, Xuelian, Yao, Jaroor] Read about JNI; this SO answer and this Wiki article are good places to start.

[Xin, Hefei, Di] Propose a web server architecture--language and framework-- ideally in the form of a diagram. Explain why you feel your choice is best. The web server backend is critical-- it will

1. receive requests from the front end in a separate threads (so it must be multithreaded),
2. create the appropriate Iterator representing our filtering and scoring computation,
3. launch the scan against the Accumulo DB,
4. receive results and forward them to the front end web interface, say as an asynchronous Server Side Event. You can brainstorm other approaches, such as creating a file containing the result of a query and serving that file.

You can look at the current HMMER project for an idea. Possibilities include Apache Tomcat 8, some Python frameworks and others. Again, Java will make life easier since you can call the Accumulo API through the Java interface, but you're welcome to use Apache Thrift to call Accumulo methods via another language.

[Eric, Xin] Learn all you can about Accumulo iterators. Dylan can't be the only one knowledgeable of Accumulo since it is now a prime component.

[Dylan] Get Accumulo running natively on Ganesan's server so that we can start developing on Accumulo.