What exactly is “read mapping”?
A read can be seen as a substring of a genome. Given a reference genome and a read, the goal is to find the position in the reference genome where the read came from.

How do I identify mutations from the aligned reads?
You should first align all the reads to the reference genome. Then you should go through each base in the reference genome and check what are the corresponding bases among the aligned reads. If the observed bases in the reads are different from your reference genome then you have encountered either a true mutation or a random sequencing error.

You should design your own way of distinguishing between mutations and errors. One possible way of doing so would be applying a threshold for the base counts. For example, your reference genome has a base A at this position, and in your aligned reads you observed m As and n Gs. You only mark this as an A to G substitution when n > k where k is your custom threshold and otherwise call this a sequencing error and ignore it.

What are the programming language requirements?
Any language should do fine. The point is to choose a language you are most comfortable with.

What will we be submitting?
You will be submitting:
(1) A text file containing the list of mutations you found from the 10000-length reference genome. You should follow the format specified on page 2 of the project specs.
(2) The zipped source code of your project, such that we can run your code on the command line and verify your program is working. If you are using a compiled language you should include a compiled executable. If you are using an interpreted language you should include whatever scripts you have written. You should also include any necessary instructions (in the form of a readme.txt file) for a user to run your code and replicate your results.

How do I get started doing this project?
You should try applying substring searching methods from chapters 1, 2 and 9 in the textbook. The easiest way to get started is to use the brute force approach where you linearly search for each read in your reference genome, but this can be very slow when you work on a much larger genome (in the upcoming project 1b). You can also implement the three-fragments read mapping method described in the lectures, that is you cut each read in 3 and then use a hash table of k-mers to find a perfect match among the 3 reads. Luke’s discussion slides for week 2 provide some details.

Do you provide any “starter code” for this project?
We are not planning to.

Will I be penalized for identifying false mutations?
Yes. We evaluate your performance using an F1 score (https://en.wikipedia.org/wiki/F-score). You lose score when you have false positives (and also false negatives). However, you don’t

have to have a perfect F1 score in order to receive full credit for this project; in other words, we are allowing you to identify some false mutations and miss some mutations. You get full credit when your F1 score is above a certain threshold.

Should I be using the Hoffman2 server?

This project should be completely doable without using a computing cluster.