A new approach to Base calling.

Project progress report

Introduction :

Nanopore based DNA sequencing involves capturing the ionic current levels produced by the DNA fragment when it passes through a nanopore and making use of base calling models to predict the corresponding sequence of nucleotides. The base calling mechanisms available today achieves a very low accuracy rate in this regard and the reads obtained are not very useful. The goal of our project is to come up with a more accurate base-calling mechanism. To achieve this, we are using a supervised learning algorithm based on support vector machines. To train the model, we use the raw signals available from the nanopore (in the form of events) as the input and the actual bases they correspond to as the output. Our approach and progress in this regard are mentioned in detail in the next section. We would also train our data using the naive bayes classifier and compare the accuracy rates obtained for base calling. Finally if time permits we will also explore other learning approaches like Neural Networks.

Approach and Progress :

Our approach is divided into 3 categories as described below :

1) **Data extraction and processing**

   We have downloaded two sets of genome data from GIGAdb for EColi, which is available publicly on their website. One of them is “Ecoli_R7_ONI”, which is ~21 GB in size. The other is “Ecoli_R73”, which is larger dataset, ~37 GB in size. Both the dataset are in FAST 5 format.

   Poretools software can be used directly on the native fast5 file format given by Oxford Nanopore. It has a set of utility functions that can be on FAST5 files to get the event data(i.e. time-series of nanopore translocation and get FASTA format reads to facilitate sequence alignment).

   To get the events data from the FAST5 files,we have developed a perl script which goes through the list of the given FAST5 files and runs the poretools events utility for the individual files to give the time-series data along with the suggested model k-mers(calculated using their HMM model).

   Similarly, for generating the FASTA files, we have developed a script which will go through all the given FAST5 files and give the reads corresponding to every event file in FASTA format.

2) **Alignment**

For training the classifier, we must have the aligned sequence from which we extract the k-mers to train the classifier corresponding to an event. We have used BLAST as our alignment tool and have coded a module that goes through a set of FASTA files, aligns each of them with the original genome sequence and outputs it to a file(one for each fasta aligned). This new fasta file(which contains the aligned sequence), along with the corresponding fast5 file, is then used to generate another file that stores the data in the format required for $SVM^{multiclass}$.

3) **Classification**

Given the dataset consisting of the event data and the aligned reads corresponding to the events different kinds of classifiers can be used over the given dataset. Classifiers in general (like Support Vector Machine or Naive Bayes) are sensitive to parameter optimization(i.e different parameter selection can significantly change the output). So, if we observe that the chosen classifier performs better than the other classifier, it's only true for the selected parameters.

We have used the following classifiers for classification module :

3.1 Support Vector Machine: According to our original plan, we used multi-class Support Vector Machines to predict a fragment's read based on the events data. Support Vector Machine is a discriminative classifier that is defined by a separating hyperplane. Given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which classifies the new examples, depending on which side of the hyperplane it lies. After getting the aligned string from the BLAST tool, for a particular fasta file, we run a program that generates the corresponding feature vectors and the class label based on the fast5 file that was aligned in $SVM^{multiclass}$ format. We use the following idea to get the feature vectors. Let the event vector(α_t) for the $t_{th}\, k-mer$ in the aligned read, be the **mean**, **standard deviation** and **the length** of the event that corresponds to that $k-mer$. The feature vector(f_t) is the concatenation of the vectors α_t, α_{t-1} and α_{t-2}. We normalized the vectors giving maximum weight to α_t and lesser weights to α_{t-1} and α_{t-2}.The class label, on the other hand is a unique numeric value in the range of $0 - 1023$ depending on the 1024 different possible 5-mers. We have ignored gaps for the time being.

Unfortunately, we could not train and test our data using $SVM^{multiclass}$ because the extracted Ecoli_R73_MG1655_ONI files(the ones that aligned the best using BLAST)

were corrupted. We have run $SVM^{multiclass}$ on a very small subset of the data. Even though the classifier worked properly, the classification was poor with a very low accuracy(because the number of training sets were very small and the gaps were completely ignored).

3.2 Naive Bayes Classifier : It's a simple probabilistic classifier based on the Bayes theorem and supports strong independent assumptions i.e. a particular feature of a class is unrelated to the presence (or absence) of any other feature. Given the nature of our classification problem, the naive bayes classifier can be efficiently trained to output the appropriate sequence of nucleotides. Currently, we are also developing(coding) a model based on Naive Bayes Algorithm. We will supply the feature vectors for training it and finally compare the results with the SVM classifier(which is our central approach).

Road blocks so far :

For sequence alignment, we were able to successfully align the Ecoli K12_R7 FASTA reads with the original genome sequence. However, we are facing difficulties in extracting the event data from the FAST5 files of Ecoli K12_R7. We are still trying to fix the issue. We have another set of Ecoli MG1655 FAST5 files but the reads are not getting aligned to the actual genome reference.

The second roadblock we face is that a significant portion of the alignment is composed of gaps. This not only increases the number of classes, it also reduces our training set significantly.