# DNA Statistics and the Null Hypothesis

Arete Associates RIPS 2016 Project Description

## 1 INTRODUCTION

Machine learning algorithms are being developed and refined at a rapid pace, and have shown tremendous utility for analyzing “big data”. Big data typically involves high-dimensional data, but where the number of samples (data points) is much larger than the number of dimensions. Recently, machine learning techniques have been applied to DNA data. See [1] for an overview of machine learning in bioinformatics. DNA data is very high dimensional (millions, hundreds of millions, or even more potential features), but the number of samples is typically much smaller than the dimensionality. Two problems arise from this type of “big data”. The first concerns how to efficiently search such a large potential feature space. The second concerns evaluating the statistical significance of a discovered feature. The latter problem is the focus of this project.

## 2 STATISTICAL SIGNIFICANCE

Suppose, by hook or crook, we discover a DNA feature which seems to hold predictive power. For example, imagine we search a collection of bacterial DNA samples and find that bacteria with the sequence "ATCTCTGTTCCTATCATATATATACCCCG" are resistant to a particular antibiotic, while ones without that sequence are susceptible. Given that we discovered this feature using a finite set of genomes, does this rule generalize to all genomes? Or is this pattern just a coincidence?

Ultimately, we wish to estimate the statistical significance of a DNA classification feature. A feature could be said to be statistically significant if it is unlikely to occur by chance. In other words, we seek to compute the probability of the feature occurring randomly. We refer to this as the probability of the null hypothesis. The *null hypothesis*, in this case, is that the discovered DNA feature is just random noise that happens to partition our data correctly. The *alternate hypothesis* is that the discovered DNA feature is "real", in the sense that it has some true correlation with the bacterial properties that will extrapolate to new samples. If we can show that the probability of the null hypothesis is small (i.e. that the feature was unlikely to have occurred randomly), then we lend confidence to our discovery.

One way to estimate the probability of the null hypothesis is to use *cross validation*, which works as follows. Instead of searching our entire dataset for classifying features, we randomly partition our data into "training" and "testing" sets. Then we search for features in the training set, and check to see how well the discovered features predict the properties in the testing set. We perform this partition-train-test process repeatedly, and accumulate the success statistics.

Cross validation is widely employed in machine learning, and with great success. However, a serious problem arises in our bioinformatics application, due to the limited amount of data we have available. DNA sequence and biological testing are time-consuming and expensive, which means we often work with small datasets. Suppose we have 20 positive samples and 20 negative samples, and that there

exists a feature (call it feature 1) that perfectly classifies the samples. If we employed cross validation, we would choose (for example) 15 positive and 15 negative samples for our training set. Since feature 1 exists in all 20 positive samples, it must exist in our 15 positive test samples. Assuming our search algorithm is good, we will find it. Then we will test feature 1 against our 5 positive and 5 negative test samples, and the classification will work perfectly! No matter how we partition the data, we will get the same feature and the same perfect classification. Does this mean our feature is perfect? No, it just means our dataset is too small for cross validation.

This project explores an alternate way to estimate the probability of the null hypothesis.

# 3 THE APPROACH

DNA sequences are very complex and not well understood. We will start with some very strong simplifying assumptions and then gradually increase complexity. You should start with task 3.1, but the remaining tasks may be completed out-of-order.

## 3.1 SINGLE-SEQUENCE FEATURES AND RANDOM DNA

DNA is often represented as a sequence of the letters A, T, C, and G. These letters each represent one of the four nucleotides in the DNA sequence. Thus we can think of a DNA sequence as a sequence of base-4 integers. Sometimes it useful to consider nucleotide triples (e.g. ATP) as the DNA base, which would make for a sequence of base-64 integers. To cover all possibilities, we will consider DNA to be a sequence of base-b integers.

Suppose we have a set of *n* base-*b* random sequences of length *m*. Suppose also that we partition the sequences into *k* 'positive' and *n-k* 'negative' sequences. What is the probability of finding a subsequence of length *s* which partitions the set into the two populations*?* That is, the subsequence is present in all *k* of the 'positive' samples and absent in the other *n-k* 'negative' samples.

This can be done on paper or computationally, but the end result should be a function of m, b, s, n, and k. This is a relatively straight-forward (but still challenging) statistical problem, which will help introduce the concept and challenges of the overall problem.

## 3.2 IMPERFECT CLASSIFICATION

The above task assumes that the feature perfectly divides the n positive and n-k negative samples. Suppose we have a feature which misclassifies q samples. Repeat the above calculation in this case.

## 3.3 FEATURE COMBINATIONS

Sometimes it is not a single DNA snippet which is predictive, but a logical combination of snippets. For example, a trait might require the presence of two snippets, or the presence of one and the absence of another. Repeat the above calculation, but for logical combinations of subsequences. The combinations of interest are: 'A and B', 'A or B', 'A and not B'.

## 3.4 Data Errors

The above calculations assume that all of our data is correct. Unfortunately, this is not true. DNA reading systems produce occasional errors. These errors can arise in multiple ways. Here we will assume that we can represent the errors with a simple error rate (probability of erroneous sequence element).

Repeat the first calculation, but allowing for an error rate 'r'. You may assume that r is small.

## 3.5 Markov Chain

So far, we have assumed that DNA sequences are random, but DNA sequences are not truly random. One way to model a DNA sequence is with a Markov chain. You may refer to reference [2] for a discussion of Markov models in bioinformatics. A Markov chain of order *m* is a random process where the probability distribution for the next state depends only on the *m* previous states. So in a zero-order Markov chain, the probability of each element is independent of the previous elements. A zero-order Markov chain model for DNA needs to only specify the probability of the occurrence of each possible element. For example, we can specify that: P(A) = P(T) = P(C) = P(G) = 0.25.

In a first order Markov chain, the probability of each element depends on the previous element. We can represent the Markov chain as a transition probability matrix:

$$P = \begin{pmatrix} P_{A\to A} & \cdots & P_{A\to G} \\ \vdots & \ddots & \vdots \\ P_{G\to A} & \cdots & P_{G\to G} \end{pmatrix}$$

Repeat the first calculation, but using a first-order Markov model for the probabilities.

## 3.6 Real-World DNA (stretch goal)

We have made a lot of simplifying assumptions about DNA in order to make these calculations tractable. How accurate are the assumptions we have made? There are freely available DNA sequences online for a variety of organisms. Picking one organism (e.g. bacteria), study the statistics. Are the elements distributed randomly? Can you create a simple Markov model which performs better than random?

See reference [3] for access to DNA data. If you need help deciding on an organism, you can start with the bacterium Staphylococcus aureus. It's in the news a lot and there is lots of data.

# 4 References


[1] P. Larranaga, et al., "Machine Learning in Bioinformatics," Briefings in Bioinformatics, vol. 7 no. 1, 86-112, 2006.

[2] E. Birney, "Hidden Markov Models in Biological Sequence Analysis," IBM J. Res. & Dev., vol. 45 no. 3/4, May/July 2001

[3] http://www.ncbi.nlm.nih.gov/genome/browse/.