Metastats 2.0: An improved method and software for analyzing metagenomic data

Joseph N Paulson* Mihai Pop† Héctor Corrada Bravo‡

October 20, 2011

Abstract

This document outlines the project proposal for the 2011-2012 AMSC 663/664 course series. The project is to develop Metastats 2.0, a software package analyzing metagenomic data. We propose two major improvements to the Metastats software and the underlying statistical methods. The first extension of Metastats is a mixed-model zero-inflated Gaussian distribution that allows Metastats to account for a common characteristic of metagenomic data: the presence of many features with zero counts due to under sampling of the community. The number of 'missing' features (zero counts) is correlated to the amount of sequencing performed, thereby biasing abundance measurements and the differential abundance statistics derived from them. In the second extension we describe new approaches for data normalization that enable a more accurate assessment of differential abundance by reducing the covariance between individual features implicitly introduced by the traditionally used ratio-based normalization. We provide an introduction to the project and then provide an outline for the implementation, validation, and deliverable. A timeline for major milestones is provided.

[1]Applied Mathematics and Scientific Computing (AMSC), Center for Bioinformatics and Computational Biology (CBCB), University of Maryland - College Park, jpaulson@umiacs.umd.edu

[2]Department of Computer Science, AMSC, CBCB, University of Maryland - College Park, mpop@umiacs.umd.edu

[3]Department of Computer Science, AMSC, CBCB, University of Maryland - College Park, hcorrada@umiacs.umd.edu

1 Problem Introduction

Metagenomics is the study of the genetic material recovered from an environmental sample. The DNA from a particular environmental sample is amplified through a process known as polymerase chain reaction (PCR). This process essentially doubles the DNA with each cycle of the process. Final DNA material is approximately $DNA_b \cdot 2^k$ where, DNA_b is the initial DNA quantity supplied, and k are the number of cycles [4].

This process is required for the next steps in the analysis pipeline. Following amplification, the DNA is sequenced, a process to determine the order of the nucleotides of a particular DNA strand. The end result are thousands of nucleotide sequences in a text file. For second generation sequencing technologies, each line in the text file consists of 30 - 400 base pairs representing a replication of a fragment of DNA. Each of these are known as a read. These reads are then annotated, a process of assigning the read to a particular organism based on a biological database. The number of reads assigned to a particular organism is an approximation of the abundance of that organism in the community. Typically the reads are first clustered according to similarity, given an arbitrary name and these clusters are annotated by their representative sequence. These clusters are known as OTUs.

In many studies, there is a goal to compare samples, as in to determine whether or not the abundance of one or more organisms is correlated with some characteristic of the sample, including health/disease status. In metagenomic data, there are many issues trying to compare samples as there is a large variation in the number of reads output by the sequencer for unknown reasons.

As there are an arbitrary number of reads output determined by the sequencing instrument, and one's ability to sample from potentially millions of bacteria in a particular environment, we are dealing with relative abundances (to a true population) where lower abundant bacteria are missed due to the sampling process. We hypothesize that many bacteria are also preferentially sampled at varying degrees. It should be noted that in many metagenomic studies, and the datasets we will use, a certain conserved / hypervariable region of a bacteria's genome is specifically sought out during the amplification and sequencing stage and used for annotation. The common region used is called 16S ribosomal DNA and refers to the $\approx$ 1,500 nucleotides that encode that region of the RNA. The 16S region is itself a subregion of the 30s subunit of a prokaryotic ribosome (unit of cells that help assemble proteins).

1.1 Previous approaches

Metagenomic studies originally focused on exploratory and validation projects, but are rapidly being applied in a clinical setting. In this setting, researchers are interested in finding characteristics of the microbiome that correlate with the clinical status of the corresponding sample [3]. Comparatively few computational/statistical tools have been developed that can assist in this process, rather most developments in the metagenomics community have focused on methods

that compare samples as a whole. Specifically, the focus has been on developing robust methods for determining the level of similarity or difference between samples, rather than identifying the specific characteristics that distinguish different samples from each other.

Metastats [7] was the first statistical method developed specifically to address the questions asked in clinical studies. Metastats allows a comparison of metagenomic samples (represented as counts of individual features such as organisms, genes, functional groups, etc.) from two treatment populations (e.g., healthy vs. disease) and identifies those features that statistically distinguish the two populations.

The underlying algorithm used by Metastats was to compute a t-statistic from the two groups for each particular feature/bacteria i: $t_i = \frac{\bar{X}_{i1} - \bar{X}_{i2}}{(s_{i1}^2/n_1 + s_{i2}^2/n_2)^{.5}}$. Following that initial observed t-statistic, an empiraclly obtained p-value would be obtained by permuting the samples B times, recalculating a t-statistic for each feature each time and taking the proportion of t-statistics greater than the originally observed value, ie. $p_i = \frac{\{|t_i^{ob}| \geq |t_i| b \in 1 \ldots B\}}{B}$.

There are a few other approaches [6] that prefer a comparison between groups of samples using other non-parametric tests (including Kruskal-Wallis), but I will delve into those later.

2 Approach I

As mentioned before, many low abundant features are not "found" in a particular sample, simply because of the large sample size and low total number of reads, ie. depth of coverage.

Here we propose two major improvements to the Metastats software and the underlying statistical methods. The first extension of Metastats is a mixed-model zero-inflated Gaussian distribution that allows Metastats to account for a common characteristic of metagenomic data: the presence of many features with zero counts due to under sampling of the community. The number of 'missing' features (zero counts) correlates with the amount of sequencing performed, thereby biasing abundance measurements and the differential abundance statistics derived from them.

The zero-inflated model is defined for the continuity-corrected log of the count data:

$$y_{ij} = \log_2(c_{ij} + 1)$$

as a mixture of point mass at zero $I_{\{0\}}(y)$ and a count distribution $f_{count}(y; \mu, \sigma^2) \sim N(\mu, \sigma^2)$. Given mixture parameters π_j, we have that the density of the zero-inflated gaussian distribution for feature i, in sample j with S_j total counts and values $\theta_{ij} = \{S_j, \beta_0, \beta_1, \mu_i, \sigma_i^2\}$:

$$f_{zig}(y_{ij}; \theta_{ij}) = \pi_j(S_j) \cdot f_{\{0\}}(y_{ij}) + (1 - \pi_j(S_j)) \cdot f_{count}(y_{ij}; \mu_i, \sigma_i^2)$$

The mean is specified as, given class membership k_j:

$$E(y_{ij}|k_j) = \pi_j \cdot 0 + (1 - \pi_j) \cdot (b_{i0} + b_{i1}k_j)\,.$$

.

Based on the observation that the number of zero-valued features on a sample depend on its' total number of count s, using a binomial model, we model the mixture parameters $\pi_j(S_j)$,

$$\log \frac{\pi_j}{1 - \pi_j} = \beta_0 + \beta_1 \cdot \log(S_j)$$

To estimate the parameters we will make use of the E-M algorithm.

The input data will be a matrix of normalized count values, samples along the columns and features (organisms) along the rows, total raw counts (ie. number of reads for a particular sample) S_j, and class indicator k_j.

2.1 Expectation-Maximization algorithm:

We can get maximum-likelihood estimates using the expectation-maximization algorithm, where we treat mixture membership $\Delta_{ij} = 1$ if y_{ij} come s from the zero point mass as latent indicator variables. Denote the full set of estimates as $\theta_{ij} = \{\beta_0, \beta_1, b_{i0}, b_{i1}\}$. The log-likelihood in this extended model is then

$$l(\theta_{ij}; y_{ij}, S_j) = (1-\Delta_{ij}) \log f_{count}(y; \mu_i, \sigma_i^2) + \Delta_{ij} \log \pi_{ij}(s_j) + (1-\Delta_{ij}) \log\{1-\pi_{ij}(s_j)\}.$$

E-Step: Estimates responsabilities $z_{ij} = Pr(\Delta_{ij} = 1|\hat{\theta}_{ij}, y_{ij}) = E(\Delta_{ij}|\hat{\theta}_{ij}, y_{ij})$ as:

$$\hat{z}_{ij} = \frac{\hat{\pi}_j \cdot I_{\{0\}}(y_{ij})}{\hat{\pi}_j \cdot I_{\{0\}}(y_{ij}) + (1 - \hat{\pi}_j) f_{count}(y_{ij}; \hat{\theta}_{ij})}$$

ie. the responsibility, or proportion of counts coming from the spike-mass distribution. Notice $\hat{z}_{ij} = 0 \;\forall\; y_{ij} > 0$.

M-Step: Estimates parameters $\hat{\theta}_{ij} = \{\hat{\beta}_0, \hat{\beta}_1, \hat{b}_{0i}, \hat{b}_{1i}\}$ given current estimates $\hat{z}_{ij}$:

Current mixture parameters are estimated as: $\hat{\pi}_j = \sum_{i=1}^{M} \frac{1}{M} \hat{z}_{ij}$ from which we estimate β, using least squares on the logit model as

$$\log \frac{\hat{\pi}_j}{1 - \hat{\pi}_j} = \beta_0 + \beta_1 \log(s_j)$$

.

Parameters for the count distribution are estimated using weighted least squares where the weights are $1-\hat{z}_{ij}$. Note only samples with $y_{ij} = 0$ potentially have weights less than 1.

For up to ten iterations, at each iteration we will calculate the negative log-likelihood for each feature and determine if the estimates reached convergence for a particular feature.

2.2 P-values

Following the E-M algorithm we will calculate a t-statistic, $t_i^{ob} = \frac{b_{1i}}{(\sigma_i^2/\Sigma(1-z_{ij}))^{.5}}$, permute class membership, k_j, for B times and calculate a p-value for each particular feature, $p_i = \frac{\{|t_i^{ob}| \geq |t_i| b \in 1 \ldots B\}}{B}$.

The ultimate goal is to get a spreadsheet of values for each particular feature (b_{0i}, b_{1i}, p).

I am considering to also include a presence-absence Fisher's test making use of the weights.

3 Approach II

In the second extension we describe new approaches for data normalization that enable a more accurate assessment of differential abundance by reducing the covariance between individual features implicitly introduced by the traditionally used ratio-based normalization. These normalization techniques are also of interest for time-series analyses or in the estimation of microbial networks.

When dealing with sequencing data, there is a need to normalize count data due to the extreme variance in sample coverage and remove the arbitrariness of the sampling process from the equation []. The hope is to cleary identify the biological differences, in particular differential abundance of the particular feature, whether it be gene, 16S, or read count. Unfortunately, obscuring variation can be induced due to sample preparation, sample site, etc. In short, there is interesting variation and obscuring variation, normalization hopes to diminish the effect of obscuring variation.

The usual normalization procedure for bacterial counts is dividing each count by the sample's total counts. This introduces false correlations between taxa resulting from dividing the numerator (count, c_{ij}), for a specific taxa by a denominator derived in part by the numerator, ie. $y_{ij} = c_{ij}/N_j$ where $N_j = \sum_i c_{ij}$ [5]. However, the need to normalize across samples with different sequencing yields certainly exists when analyzing metagenomic data.

In both of the following algorithms the input will be a matrix counts of size M x N for the M features and N samples. The output will be a matrix of normalized counts, ie. some sort of scaling of the original counts.

3.1 Cumulative Sum Normalization

In 2002, quantile normalization for micro-array data was shown to be the ideal method for normalization [1]. The technique is a method meant to make two distributions identical in statistical properties and remove the variation of non-biological origin. The motivation is coupled by the fact that certain measurements are sampled preferentially.

Similarly to quantile normalization, the assumption follows that the rate of sampling a particular measurement is similar for those with similar proportions

of identified taxa. We too show a technique for making two distributions identical in statistical properties with the additional metagenomic assumption that there is a finite capacity in a metagenomic community. As such, the cumulative summation of a samples' 16s or metagenomic count should follow a similar rate to that of other samples with similar proportions of zeros at an OTU level.

Our algorithm follows (wording is similar to [1]):

- bin samples into groups, G_m, of similar zeros proportions at the OTU level, with a finer mesh at the higher proportions;
  - given n_i samples $\in G_m$ all of length p, form X_m of dimension p x n_i;
  - sort each column of X_i to obtain $X_{m,sort}$;
  - replace each column of $X_{m,sort}$ with the cumulative sum of that column;
  - take the means across rows of $X_{m,sort}$ and assign the mean to each element in the row to get $X'_{m,sort}$ and take the inverse of the cumulative norm;
  - get $X_{m,normalized}$ by rearranging each column of $X'_{m,sort}$ to have the same ordering of the original X_m
- scale each group's normalized counts to the median groups.

3.2 Scaling normalization

A recent proposal for normalization of RNA-seq data is to scale counts by the 75th quantile of each samples non-zero count distribution q_{75} ie. $y_{ij} = c_{ij}/q_{75j}$ [2]. This type of normalization is motivated by the observation that a few measurements, e.g., taxa or genes, are sampled preferentially as sequencing yield increases, and have an undue influence on normalized counts derived by the usual normalization procedure. In that case, the 75th quantile was a chosen as it behaved consistently across samples. In our data, we have analyzed the distribution of non-zero counts and have determined that the 95th quantile is more appropriate [1]. Nonetheless, the usual normalization procedure in metagenomic data does assume there is a finite capacity in metagenomic communities, which is not necessarily true in RNA-seq samples. To address this we introduce another, simpler novel normalization method denoted S_{95}, which scales counts by dividing the sum of each samples counts up to and including the 95th quantile, ie. for all samples x_j, $S_{95j} = \sum_i c_{ij} \leq q_{95j}$. This procedure addresses both issues identified above, namely, it constraints communities with respect to a total capacity, but does not place undue influence on features that are preferentially sampled.

[1] Picture to be included later on

3.3 Possible issues

There are several issues that one could potentially encounter. The biological data needs to be processed and as we have very large datasets we need to preprocess the data and remove select features.

4 Implementation

4.1 Software

Code will be developed using the R language. R is useful for the various statistical R functions and R packages available. Given time, C code that will be wrapped to in R will be implemented. The C code would include use of the OpenMP library (a parallel programming C library).

The bottleneck in the algorithm will be to calculate the p-values empiracly by bootstrapping. This step is trivially parallelizable and given time steps will be made to parellize the p-value calculations.

4.2 Hardware

Development on my Macbook Air, 1.6 core duo, 4 GB of ram.
Code will be run on UMIACS's computer Ginkgo
8 x Quad-core AMD Opteron Processor 8365 (2300MHz) (32 cores), 256 GB Ram, RHEL5 x86_64

5 Database

There are various biological databases that the National Institutes of Health maintains. National Center for Biotechnology Information (NCBI) maintains the NR - nucleotide database, one database I may use to blast raw sequences again and annotate raw sequences. However, the majority of datasets I use will be 16S DNA datasets, and as such I will obtain published datasets in the second semester from Genbank / EMBL / DDBJ. To annotate 16S datasets I will make use of the RDP - ribosomal database project classifier to annotate our sequences. The datasets I will be using currently are unpublished and following analysis will be placed in Genbank for the world. There are two main datasets I will analyze using my algorithm(s). The first one is a diseased and healthy dysentery metagenomic 16S gut datasets consisting of 1007 samples. The second one is 139 samples of approximately 12 gnotobiotic mice that have been placed on two separate diets in a longitudinal study.

6 Validation

The first method of validation of the code will be to ensure that when I compare my results to a matrix of non-zero counts, my model's results should coincide

with the log model fit, $E(y_{ij}|k_j) = (b_{i0}+b_{i1}\cdot k_j)$. The results should be identical.

The second approach that I intend to validate the model with is to generate data using the model. I will generate OTU level datasets with 1000 features. Each feature will get .1% of the total sample counts. For the two groups, for each feature, there will be a base mean and one group will have a siginificant mean difference. That particular group will have a very low variance. Sparsity will be randomly introduced. The resulting data will be plugged into the algorithm and should show that π_j for the first group (no large k_j effect) will be approximately 1. However, for the second group, we would expect π_j to be closer to zero. The fits should show this. This is in development and we will expand on this later on.

For the second part, normalization, the codes will be validated by running a few trivially simplified datasets by hand and comparing the results.

7 Testing

I will generate OTU level datasets with 1000 features, 50 will be "significant". One of the groups will be different from the second group for those 50 and the model will be run on this data, as well as Metastats 1.0.

For the normalization, testing will be addressed later on.

8 Project Schedule + Milestones

- November 30th
  Finish code that will preprocess data, including the normalization of a dataset. I will implement as a function routine in R that will load a tab-delimited file of counts, annotation, OTU, and sample names in a convenient manner and quickly. The original Metastats has a version that runs very slowly.

- December 15
  I will have finished the E-M algorithm, I will have a script function to pass data that was loaded in the previous step and normalized/processed and send it to the E-M code. and present a mid-year report claiming I have finished all of the zero-inflated Gaussian model and normalization codes and beginning ruminations on validation and testing.

- January 15-February 15
  I will continue reading and work on validation of the methods, I also hope to compare the normalization methods on real datasets. I will also throughout this time begin packaging the code, commenting, etc to make it ready for submission to bioconductor (an R package).

- March 15
  I will finish analyzing various datasets (those mentioned previously) and if the schedule is not delayed, I will parallelize the code (if necessary and

time permitting). I also will find datasets other than those, whether from NCBI or other sources to analyze.

- May 15
  I plan to deliver the final report.

9 Deliverables

The deliverables include submission of the the R code package for Bioconductor, a final-year report, and submission of the datasets into the NCBI databases if collaborations accept. Also, an archive of results for datasets made public and published will be included if there are any at that time.

References

[1] B M Bolstad, R A Irizarry, M Astrand, and T P Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[2] James H Bullard, Elizabeth Purdom, Kasper D Hansen, and Sandrine Dudoit. Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC Bioinformatics*, 11:94, 2010.

[3] National Academy of Science Committee on Metagenomics. The new science of metagenomics: Revealing the secrets of our microbial planet. *National Academy of Sciences*, 2007.

[4] O. Paliy and Foy B. Mathematical modeling of 16s ribosomal dna amplification reveals optimal conditions for the interrogation of complex microbial communities with phylogenetic microarrays. *Bioinformatics*, 2011.

[5] Karl Pearson. Mathematical contributions to the theory of evolution.– on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Society*, 60:489–498, 1896.

[6] Nicola Segata, Jacques Izard, Levi Waldron, Dirk Gevers, Larisa Miropolsky, Wendy S Garrett, and Curtis Huttenhower. Metagenomic biomarker discovery and explanation. *Genome biology*, 12(6):R60, June 2011.

[7] James White, Niranjan Nagaranjan, and Mihai Pop. Statistical methods for detecting differentially abundant features in clinical metagenomic samples. *PLOS Comp Bio*, 11, 2009.