Metastats 2.0

An improved method and software for analyzing metagenomic data

Joseph N. Paulson
jpaulson@umiacs.umd.edu

Mihai Pop
mpop@umiacs.umd.edu

Héctor Corrada Bravo
hcorrada@umiacs.umd.edu

Abstract:
Here we present major improvements to Metastats software and underlying statistical methods.

1) A mixed-model zero-inflated Gaussian distribution.
2) A novel normalization method.
Application Background

- What is metagenomics?
- Why is it important?
- What do I hope to do?
Detection of differential abundance!

Definition: A count, $c_{ij}$ is the number of reads annotated as a particular taxa $i$ for the $j$th sample.
Statistical Methods for Detecting Differentially Abundant Features in Clinical Metagenomic Samples

James Robert White¹, Niranjan Nagarajan², Mihai Pop³*

\[
\bar{X}_{it} = \frac{1}{n_t} \sum_{j \in \text{treatment } t} f_{ij}
\]

\[
s_{it}^2 = \frac{1}{n_t - 1} \sum_{j \in \text{treatment } t} (f_{ij} - \bar{X}_{it})^2
\]

\[
t_i = \frac{\bar{X}_{i1} - \bar{X}_{i2}}{\left( \frac{s_{i1}^2}{n_1} + \frac{s_{i2}^2}{n_2} \right)^{\frac{1}{2}}}
\]

\[
p_i = \frac{\{ |t_i^b| \geq |t_i| \text{ for } b \in 1...B \}}{B}
\]
Too slow! Can’t handle large datasets
  • More and more data coming daily!

 Doesn’t account for depth of coverage

 Normalization induces spurious correlations
\[ f_{total}(y_{ij}; \theta) = \pi \cdot f_0(y_{ij}) + (1 - \pi) \cdot f_1(y_{ij}) \]
Approach: Zero-inflated Gaussian

- Counts are log transformed as: \( y_{ij} = \log_2(c_{ij} + 1) \)
- Mixture of point mass, \( f\{0\} \), at zero and a count distribution \( f_{\text{count}}(y; \mu, \sigma^2) \sim N(\mu, \sigma^2) \)
- Mixture parameter \( \pi_j \)
- Values \( \theta = \{S_j, \beta_0, \beta_1, \mu_i, \sigma_i^2\} \)
- Density is:

\[
 f_{\text{zig}}(y_{ij}; \theta) = \pi_j(S_j) \cdot f\{0\}(y_{ij}) + \\
 (1 - \pi_j(S_j)) \cdot f_{\text{count}}(y_{ij}; \mu_i, \sigma_i^2)
\]
Zero-inflated Gaussian

- And a mean specified as:

\[ E(y_{ij} | k(j)) = \pi_j \cdot 0 + (1 - \pi_j) \cdot (b_{i0} + b_{i1} \cdot k(j)) \]

- Where \( k_j \) is our class label
\[ f_{\text{count}}(y; \mu, \sigma^2) \]

\[ y_{ij} = \log_2(c_{ij} + 1) \]
Mixture parameters

Zero-valued features depend on a sample’s total number of counts, $S_j$. They follow a binomial distribution.

We model the linear effect with our mixture parameter $\pi_j$ via linear regression with a transformation function:

$$\log \frac{\pi_j}{1 - \pi_j} = \beta_0 + \beta_1 \cdot \log(S_j)$$
\[ f_{\{0\}}(\Delta_{ij}) \]

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

\[ f_{\text{count}}(y; \mu, \sigma^2) \]
Log-likelihood

We can get the maximum-likelihood estimates using the Expectation-Maximization algorithm, where we treat mixture membership $\Delta_{ij} = 1$ if $y_{ij}$ comes from the zero point mass as a latent indicator variable.

Denote the full set of estimates as $\theta_{ij} = \{\beta_0, \beta_1, b_{0i}, b_{1i}\}$

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

$$+ (1 - \Delta_{ij}) \log(1 - \pi_j(S_j))$$
Algorithm:

1. Preprocess Data
2. Take initial guesses for the expected value of the latent indicator variables.
   - $ij$ positions with counts $> 0$, the value is 0, else .5

\textit{For i in 1.....M:}

3. Expectation
4. Maximize
5. Calculate negative log-likelihoods for each feature

\textit{Repeat}

7. Permute class membership (labels)
8. Calculate new t-statistic, permute and calculate p-values
Expectation-Maximization

E-step:
Estimates responsibilities,

\[ z_{ij} = Pr(\Delta_{ij} = 1|\hat{\theta}, y_{ij}) = E(\Delta_{ij}|\hat{\theta}, y_{ij}) \]

as:

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

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

Current mixture parameters are estimated as:

$$\hat{\pi}_j = \sum_{i=1}^{M} \frac{1}{M} \hat{z}_{ij}$$

Parameters for the count distribution are estimated using weighted least squares where the weights are $\hat{z}_{ij}$.
Algorithm continued

- Permute the labels \( K_j \)
- Compute \( t_{i}^{ob} = \frac{b_{1i}^{K_j}}{(\sigma_{i}^{2}/\Sigma(1 - z_{ij}))^{.5}} \)
- Divided by the newly weighted standard error.
- Calculate \( p_i = \frac{\{|t_{i}^{ob}| \geq |t_i|^b \in 1...B\}}{B} \)
- Plan to add a few other tests.
Algorithm 2

• Ratio Normalization:
  – What are the issues with it??

\[ y_{Aj} = \frac{c_{Aj}}{c_{1j} + \ldots + c_{Aj} + c_{Bj} + \ldots c_{Mj}} \]

  – Spurious correlation [1]
  – False negatives [2]
  – False positives [2]

---

1 Pearson, Mathematical Contributions to the Theory of Evolution. On a Form of Spurious Correlation Which May Arise WhenIndices Are Used in the Measurement of Organs
2 Bullard et. al., Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments, BMC Bioinformatics, 2010
Genes are sampled preferentially as sequencing yield increases (# PCR cycles biases as well).

Unlike RNA-seq data\textsuperscript{c}, we assume finite capacity in metagenomic communities:

\[ S_{95j} = \sum_i c_{ij} \leq q_{95j} \]

This procedure addresses the issues:

- constraints communities with respect to a total capacity
- No undue influence on features that are preferentially sampled.

\textsuperscript{c}RNA-seq data normalization: \( y_j = c_j / q_{75j} \)
Implementation

- **Software:**
  - R and possibly C
    - Make use of R and various R package functions
    - Make use of open MP (time permitting)
- **Numerically, the bottleneck is the bootstrapping measure (fitting the weighted least squares).**
  - Thankfully that step is trivially parallelizable.
- **Hardware:**
  - Develop on my Macbook Air
    - 1.6 core duo
    - 4 gigs of ram
  - Run on Ginkgo
    - 8 x Quad-core AMD Opteron™ Processor 8365 (2300MHz) (32 cores)
    - 256 GB Ram
    - RHEL5 x86_64
Databases

- Diseased and healthy dysentery data
- Oral microbiome
- Two diet groups of gnotobiotic mice
- Access others with more time from Genbank database.
Validation

• Compare non-zero matrix results with another method, the log model fit, to ensure exact same results.

\[ E(y_{ij}|k(j)) = (b_{i0} + b_{i1} \cdot k(j)) \]

• Simulate data for known quantities (known difference, small variance) and see how model reacts.
Testing

• Ensure that preprocessing of the data is handled correctly – biologically

• Compare to Metastats, Kruskal-Wallis (non-parametric test), etc.
Project Schedule

- November 30:
  - Preprocessing data
  - Finish normalization codes
- December 15:
  - Continue reading
  - Finish Zig model
  - Midyear report
- January 15:
  - Continue reading
  - Validation of methods
- February 15:
  - Finish a comparison of normalization methods
  - Package, comment, etc.
- March 15:
  - Analyze various datasets
- April 15:
  - Parallelize
- May 15:
  - Deliver all
  - Final report
Bibliography


