Journal of Bioinformatics and Computational Biology
Vol. 14, No. 3 (2016) 1642004 (20 pages)
© World Scientific Publishing Europe Ltd.
DOI: 10.1142/S021972001642004X

Normalization of transposon-mutant library sequencing datasets to improve identification of conditionally essential genes

Michael A. DeJesus* and Thomas R. Ioerger†

Department of Computer Science, Texas A&M University
College Station, Texas 77843, USA
*mad@cs.tamu.edu
†ioerger@cs.tamu.edu

Accepted 16 December 2015
Published 29 February 2016

Sequencing of transposon-mutant libraries using next-generation sequencing (TnSeq) has become a popular method for determining which genes and non-coding regions are essential for growth under various conditions in bacteria. For methods that rely on quantitative comparison of counts of reads at transposon insertion sites, proper normalization of TnSeq datasets is vitally important. Real TnSeq datasets are often noisy and exhibit a significant skew that can be dominated by high counts at a small number of sites (often for non-biological reasons). If two datasets that are not appropriately normalized are compared, it might cause the artifactual appearance of Differentially Essential (DE) genes in a statistical test, constituting type I errors (false positives). In this paper, we propose a novel method for normalization of TnSeq datasets that corrects for the skew of read-count distributions by fitting them to a Beta-Geometric distribution. We show that this read-count correction procedure reduces the number of false positives when comparing replicate datasets grown under the same conditions (for which no genuine differences in essentiality are expected). We compare these results to results obtained with other normalization procedures, and show that it results in greater reduction in the number of false positives. In addition we investigate the effects of normalization on the detection of DE genes.

Keywords: Normalization; TnSeq; essentiality.

1. Introduction

Sequencing of transposon-mutant libraries using next-generation sequencing (TnSeq) has become a popular method for determining which genes and non-coding regions are essential for growth under various conditions in bacteria.[1] Briefly, a transposon-mutant library is made by transfecting in a vector carrying a transposable element, such as the Himar1 transposon,[2,3] which can insert at random locations throughout the genome (Himar1 can insert randomly at any TA dinucleotide). Each mutant in the library has an insertion at a single location, but the goal is to construct a saturating library where nearly all of the potential insertion sites are represented.

When grown under selective conditions, mutants with transposon insertions in essential regions will fail to survive. The abundance of the remaining insertion sites can be determined by using PCR to amplify the junctions between the transposon and the surrounding genome,[4] and the position of each insertion can be efficiently determined using a next-generation sequencer such as an Ilumuna HiSeq. This experiment typically yields several million reads, and the number of reads associated with each TA site is tabulated. While TA sites in non-essential regions have stochastically varying read counts, essential genes and non-coding regions (such as tRNAs, rRNAs, and sRNAs) can be identified as regions where the TA sites are uniformly devoid of insertions (i.e. read counts are 0).[5–8]

Determining which genes in an organism are essential is a difficult problem. The primary challenge is in lower-density datasets, where the fraction of TA sites represented in the library could be in the 20–30% range. The lower the density of the dataset, the more difficult it is to determine whether a region lacks insertions due to essentiality, or just due to random statistical fluctuations. In addition, not all TA sites in an essential gene must lack insertions, as insertions can sometimes be tolerated in the N- or C-terminus of an essential gene, or in non-essential domains or linkers between domains.[9,10] For methods that rely on comparing read counts, the variability of the data poses an additional problem.[11]

To address these challenges, several statistical methods have been proposed for quantifying the significance of essential genes. One method fits a Negative Binomial (NB) distribution to the insertion counts in each gene, and uses this to determine a p-value for significance of sparse regions.[12] The length of “gaps” or consecutive TA sites lacking insertions has also be used to quantify the significance of essential regions using the Extreme Value distribution.[13] Hidden Markov Models have also been developed for analyzing TnSeq data.[14,15] For comparison between growth conditions, the sum of read counts in a gene has been compared between conditions using a non-parametric test to identify regions with statistically significantly depressed insertions.[11]

For methods that rely on comparison of read counts, proper normalization of TnSeq datasets is vitally important. If two datasets that are not appropriately scaled are compared, it might cause the appearance of Differentially Essential (DE) genes in a statistical test, constituting type I errors (false positives). Several methods for normalizing TnSeq datasets have been proposed. Most of these methods rely on a linear transformation of the data, whereby the read counts in a dataset are scaled by a constant factor. The simplest of these is to normalize datasets such that their read counts have the same mean (e.g. by dividing by the total read count). Other methods like Relative Log Expression (RLE)[16] and Trimmed Mean of M-values (TMM)[17] have been proposed, both of which were initially developed for normalizing RNA-Seq datasets. These methods as well as others mentioned here are described in more detail in Sec. 2.2. Another approach is to fit a NB distribution (or a Zero-Inflated Negative Binomial (ZI-NB) to help account for an abundance of empty sites) and scaling by the estimated means of the model. While scaling read counts linearly is the most common procedure, other methods which use a nonlinear transformation have been proposed. These include Quantile Normalization (QNM)[18] which estimates empirical quantiles and then fits the datasets to match, and simulation-based normalization like the one used by ARTIST[15] which simulates a "control" dataset with similar statistical properties to an experimental dataset by sampling from a multinomial distribution.

One significant limitation of methods that linearly transform datasets is that they are susceptible to large spikes in read counts. Because these methods multiply read counts by a constant scalar value, they cannot reduce large outliers without also affecting small read counts which are more common. Even if the datasets share the same mean, for instance, any skew in distribution of read counts itself would still be present.

The distribution of read counts in most TnSeq datasets resembles a Geometric-like distribution, in that read counts at most sites are small (i.e. 1–50), with a (rapidly) decreasing probability of sites with large counts. Ideally, a normalization method would improve detection of conditionally essential genes between conditions by eliminating any skew and making the datasets more closely fit this Geometric-like distribution.

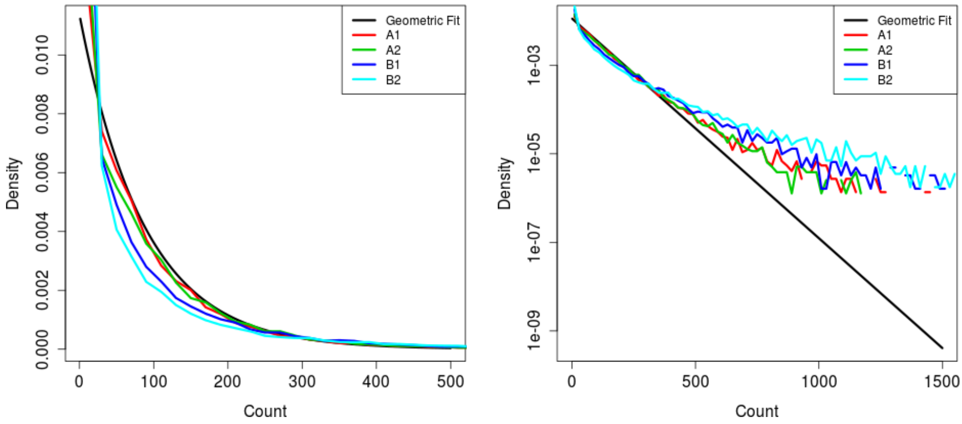
In this paper, we propose a novel method that corrects for the skew of read-count distributions observed in many TnSeq datasets by fitting them to Geometric distribution with a variable probability parameter modeled by a Beta distribution (which we call a Beta-Geometric distribution). We show that the Beta-Geometric Correction (BGC) procedure reduces the number of false positives when comparing replicate datasets grown under the same conditions (for which no genuine differences in essentiality are expected). These results are comparable to those results obtained with other normalization methods, and we show the BGC procedure produces the largest reduction in false positives. In addition we explore the effects of BGC on the detection of DE genes.

2. Beta-Geometric Correction Normalization Method

The most common method for normalization is to divide the read counts at each TA site by the overall number of reads in a dataset, which will factor out gross differences due to the amount of data collected, analogous to the calculation of RPKMs in RNA-Seq.[20] A refinement of this approach that is specific to TnSeq is to scale the read counts to have the same mean over non-zero sites (which we call "Non-Zero Mean Normalization" or NZMean), since different datasets can have widely varying levels of saturation, and distributing the same number of reads over fewer TA sites will naturally inflate the mean read count among them.

Despite these attempts at normalization, TnSeq datasets can still display quite different statistical patterns. In practice, some datasets appear "well behaved", where the distribution of read counts tends to resemble a Geometric distribution (where small read counts are most abundant, while sites with high counts are much rarer), while other datasets are skewed, with a few highly over-represented sites dominating the read count distribution. One justification why the distribution of read counts in (well-behaved) datasets might be expected to appear Geometrically distributed could be due to competition between the mutants in the population of clones in the library. The abundance of the different clones in the population will vary, reflecting differences in growth rates. In the Motomura model of species abundance,[?] competition leads to a geometric series that describes the abundance of the species in the population, where the most fit individual has the highest abundance, and less fit individuals have exponentially decreasing abundances, with the majority of the population having very low abundance. TnSeq, by sequencing reads from this culture, is in essence obtaining a sample of read counts in roughly the same proportion as the underlying population. Some models of abundance of populations use a NB distribution instead. However, because the Geometric distribution is a limiting case of the NB distribution, standardizing to a Geometric distribution can be seen as standardizing to an equivalent NB, with size parameter $r = 1$.

The resemblance to Geometric distribution can be observed in four representative datasets shown in Fig. 1(a). The skew away from an ideal Geometric, especially at high counts, can be seen better on a log scale (Fig. 1(b)). These datasets are from a Himar1 Tn-mutant library in *M. tuberculosis*, where A1 and A2 are two replicates grown *in vitro*, and B1 and B2 representing *in vivo* datasets (where the library has been passaged through a mouse). Each dataset has 2 to 5 million reads distributed over 74,602 TA sites in the H37Rv genome. Datasets A1 and A2 appear to fit a Geometric distribution more closely than B1 and B2, which show greater skew. This can also be seen on a QQ-plot (quantile-quantile), where B1 and B2 veer farther away from the 1:1 diagonal than the *in vitro* datasets. Indeed, B1 and B2 have



(a)

(b)

Fig. 1. (a) Histogram of non-zero read counts obtained from *M. tuberculosis* Tn-mutant libraries. A1, A2 are replicates grown *in vitro*, and B1 and B2 are replicates grown *in vivo*. The black line represents a Geometric fit. (b) Histogram of read counts on a log scale.

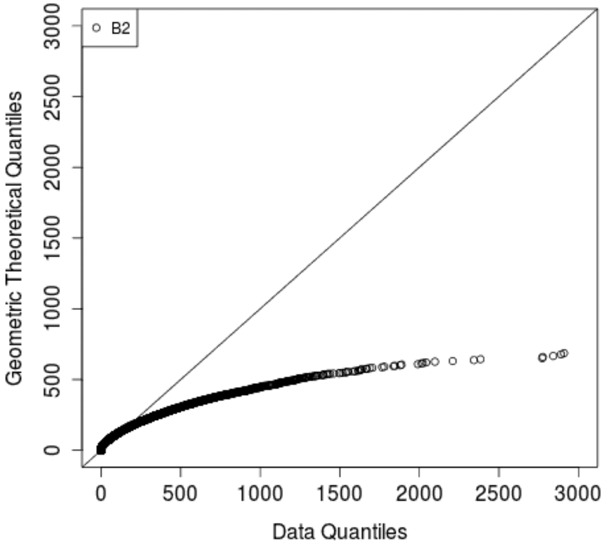


Fig. 2. QQ-plot of the raw read counts for dataset B2, and the theoretical Geometric quantiles.

extremely high counts at a few individual sites (with maximum read counts of 6009 and 16,146 respectively), compared to maximum counts of 1693 and 1175 in the A1 and A2 datasets.

The effect of the skew observed in datasets like B1 and B2 (which is a common phenomenon in TnSeq) is that it can bias the statistical analysis of essential regions, especially for methods that depend quantitatively on the read counts. Certainly, for genes containing TA sites with high spikes in read counts, they will appear exceedingly non-essential, and it could make the gene appear DE in other conditions. Simultaneously, the spikes in read counts at some TA sites will suppress the apparent level of reads at other sites, potentially making them appear relatively more essential. Figure 3 illustrates how the insertion patterns of a skewed dataset might look, before

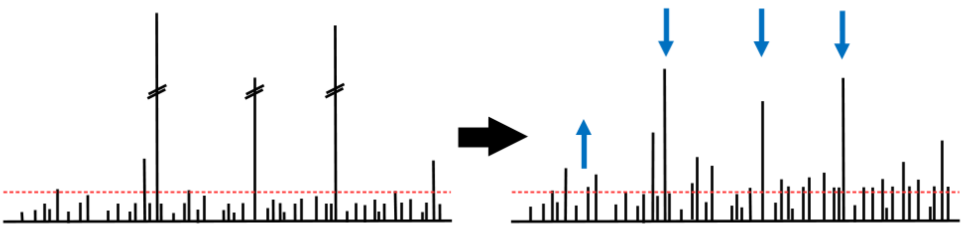
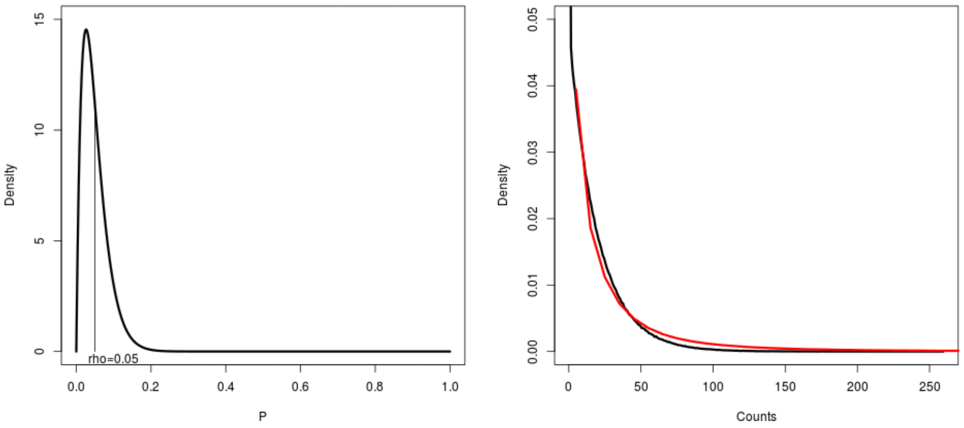


Fig. 3. Example of insertion pattern in before and after adjusting spikes in read counts. Unusually large read counts can cause regions to appear to be DE, artificially deflating counts at other sites below the mean (dashed line). Using a nonlinear transformation, large spikes are decreased while low counts are increased, adjusting them to be more in line with each other.

and after adjusting for the skew using the method proposed in this paper. Note that due to the nonlinear nature of this transformation, high counts are significantly reduced, while sufficiently small read-counts increase.

We propose a novel method for correcting for this skew in read-count distributions by fitting each dataset to a modified distribution called a Beta-Geometric distribution (Eq. (1)), and using this to adjust the observed read counts so they more closely fit a Geometric. The Beta-Geometric distribution is like a Geometric distribution but with a variable, instead of constant, parameter p, where the variation in p is modeled by a Beta distribution. This approach is based on the observation that skewed TnSeq datasets actually appear to fit not a single Geometric with a single Bernoulli parameter, p, but the weighted sum of multiple Geometric distributions with different values of p. As weights on p, we choose the Beta distribution, with parameters ρ and κ set so that the peak is around p. The Beta distribution has an extra degree of freedom representing dispersion around p (see Fig. 4). This reflects a generative model in which individual clones in the Tn-mutant library have different growth rates, some growing slightly faster and some slightly slower than wild-type cells, depending on the location of the transposon insertion in their genome. This variability in growth rates will smear out the apparent abundance of read counts after selection (i.e. several rounds of doubling in selective conditions). In this model, the spikes in read counts would come from clones that had higher-than-average growth rates, for whatever reason (biological or random):

$$\text{pdf }(c;\rho,\kappa)=\int_0^1 \text{Beta }(p|\rho,\kappa)\times\text{Geometric }(c|p)\ dp. \tag{1}$$



(a)

(b)

Fig. 4. (a) Example of a Beta distribution with $\rho = 0.05$ and $\kappa = 40$. (b) Histogram of counts from a regular Geometric distribution ($p = 0.05$, black curve), and a Beta-Geometric distribution ($\rho = 0.05$, $\kappa = 40$, red).

2.1. *Parameter estimation*

Given a set of read counts, Y_i, at n TA sites for $i \in 1, 2, 3, \ldots, n$, we assume read-counts are Geometrically distributed, with a variable parameter, p, governed by the Beta distribution:

$$\begin{aligned} &Y_i \sim \text{Geometric}(p), \\ &p \sim \text{Beta}(\kappa\rho, \kappa(1-\rho)), \end{aligned}$$

where the Beta distribution is parameterized using ρ and κ, such that ρ represents the mean of the parameter p, and κ can be thought of as analogous to a "sample size", effectively proportional to the inverse of the variance.

We seek to estimate the parameters ρ and κ that minimize the sum of squared errors (ϵ) between the observed read counts and the quantiles of the distribution:

$$\begin{aligned} \epsilon(X;\rho,\kappa) &= \sum_i^N \left(X_i' - F^{-1}(q_i;p_i)\right)^2 \\ &= \sum_i^N \left(X_i' - \frac{\log(-q_i+1)}{\log(1-p_i)}\right)^2 \\ &= \sum_i^N \left(X_i' - \frac{\log(-q_i+1)}{\log(1-\frac{\kappa\rho-1}{\kappa-2})}\right)^2. \end{aligned} \tag{2}$$

Here, X' represents the read counts in ascending order, F^{-1} represents the quantile function of the Geometric distribution, and $q_i \in [0,1]$ represents the quantiles.

To facilitate the parameter estimation, the parameter ρ is estimated as $\rho = (\sum_i^N X_i)^{-1}$, which is the maximum likelihood estimate of the Geometric distribution. The remaining parameter, κ is found by determining the root of the gradient. The gradient with respect to κ is defined as follows (derivation is included in Appendix A):

$$\frac{\partial\epsilon}{\partial\kappa} = \frac{\sum_i^N 2(2\rho-1)\log(1-q_i)\left(\log(1-q_i) - X_i \log\left(\frac{-\rho\kappa+\kappa-1}{\kappa-2}\right)\right)}{(\kappa-2)((\rho-1)\kappa+1)\log^3\left(\frac{-\rho\kappa+\kappa-1}{\kappa-2}\right)}. \tag{3}$$

The root of this gradient has an analytical solution:

$$\kappa = \frac{2 \times \exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] - 1}{\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] + \rho - 1}.$$

Once parameters ρ and κ have been estimated, capturing the skew in the dataset, the original read counts are corrected by mapping each of them to the equivalent quantile in an ideal Geometric distribution as follows:

$$c' = F^{-1}(Q(c;\rho,\kappa);p), \tag{4}$$

where $Q(c; \rho, \kappa)$ is the quantile function (CDF, obtained by sampling) for the Beta-Geometric, and $F^{-1}(q; p)$ is the inverse of the quantile function for the Geometric distribution.

2.2. *Other normalization methods*

In Sec. 3, we compare BGC to five other normalizations methods that have been proposed in the TnSeq and RNA-Seq literature.[16,19] Because of the similarities between RNA-Seq and TnSeq procedures, as well as their dependence on normalizing count-data obtained from sequencing reads, methods used for normalizing RNA-Seq data serve as a good starting point for comparison. We include two of the most popular methods from the RNA-Seq, and as well as other methods more specific to TnSeq analysis. We briefly describe each method before presenting results.

2.2.1. *Relative log expression*

One of the more popular normalization methods used in the RNA-Seq literature is RLE. This normalization was proposed by Anders and Hubers and used in their DESeq method for detection of differential expression.[16] For each sample being normalized, RLE calculates a size-factor meant to make datasets comparable regardless of their sequence depth. The factors are calculated as follows:

$$\hat{s}_j = \underset{i}{\text{median}} \frac{k_{ij}}{\left(\prod_{v=1}^{m} k_{iv}\right)^{1/m}}, \tag{5}$$

where s_j represents the scaling factor for the jth sample, and k_{ij} represents the counts at the ith position of the jth sample. The denominator is the geometric mean across all m replicates, and the median over all sites (which is more robust to outliers than the mean) is taken as a scale factor for each dataset. Read counts are then normalized by dividing them by this size-factor, rendering them comparable.

2.2.2. *Trimmed mean of M-values*

Another normalization method used in RNA-Seq is the TMM method. This method was developed by Robinson and Oshlack,[19] and estimates log-fold changes in expression and absolute expression:

$$M_g = \log_2 \frac{Y_{gk}/N_k}{Y_{gk'}/N_{k'}}, \tag{6}$$

$$A_g = \frac{1}{2}\log_2(Y_{gk}/N_k \times Y_{gk'}/N_{k'}), \tag{7}$$

where Y_{gk} represents counts at the gth counts in the kth sample, and N_k represents the total reads in that sample. The values of M_g are trimmed by 30% while the samples of A_g are trimmed by 5%. Finally the normalization factors are calculated by taking a weighted mean of the remaining M_g (after trimming) as follows:

$$\log_2(TMM_k^{(r)}) = \frac{\sum_{g\in G^*} w_{gk}^r M_{gk}^r}{\sum_{g\in G^*} w_{gk}^r}, \tag{8}$$

where

$$M_{gk}^r = \log_2 \frac{Y_{gk}/N_k}{Y_{gr}/N_r} \tag{9}$$

and

$$w_{gk}^r = \frac{N_k - Y_{gk}}{N_k Y_{gk}} + \frac{N_r - Y_{gr}}{N_r Y_{gr}}. \tag{10}$$

2.2.3. *Negative binomial*

The NB distribution is frequently used to model count data,[12,16] particularly for data that may exhibit over-dispersion. TnSeq datasets, however, contain an overabundance of sites with read counts of zero, representing either locations which are essential for growth or which were not sampled in the construction of the mutant library. Those libraries with a low saturation might make the mean read count look artificially low. Ideally, the mean read count would be calculated for all non-essential sites, however it is difficult to separate those sites which are essential from those sites that are non-essential but missing from the library. One way to account for an excessive number of zeros, and thus attempt to separate essential sites from non-essential ones, is to use a zero-inflated model. In order to examine the influence of zeros in normalizing datasets, we compared against a (ZI-NB) model, which is a 2-component mixture model. The parameters were estimated by minimizing the log-likelihood of the following model:

$$P(X_i) = \pi + \text{NB }(X_i; r, p), \quad X_i = 0, \tag{11}$$

$$P(X_i) = (1-\pi) \times \text{NB }(X_i; r, p), \quad X_i > 0, \tag{12}$$

where π represents the probability of observing a zero outside of the NB distribution, and r and p are the shape parameters of the NB distribution. For each sample, the estimated mean of the NB distribution (i.e. $\frac{pr}{1-p}$) is used as its scaling factor.

2.2.4. *Multinomial simulation normalization*

Recently, Pritchard *et al.* proposed using simulation-based normalization to effectively simulate a control sample with a multinomial distribution in order to mimic the saturation (loss of library diversity) observed in the given experimental samples. This simulation method was used as part of the ARTIST pipeline for analyzing TnSeq datasets.[15]

Because this method is based on simulating samples from a multinomial distribution, it is capable of generating an arbitrary number of control samples.

To compare with the other normalization methods, we took the expected value of the simulation as the normalized dataset. In addition, we simulated the dataset with the highest density to match the dataset with the lowest density. The method used in our comparison can be summarized briefly as follows:

$$\bar{C}' = \mathrm{E}\left[\text{Multinomial}\left(N_x, \frac{\bar{C}}{N_c}\right)\right], \tag{13}$$

where $\bar{X}$ is the vector of read counts for the input experimental sample, and $\bar{C}$ is the vector of read counts for the input control sample, and $N_x = \sum_i X_i$ and $N_c = \sum_j C_i$, which are the total number reads in the experimental and control datasets.

2.2.5. *Quantile normalization*

Another nonlinear normalization method we compare against is the (QNM) method. This method was proposed as a way to normalize DNA micro-array data by Bolstad *et al.*[18] QNM normalizes datasets so that they share the same empirical distribution of values. For a given $p \times n$ matrix of counts, $X_{i,j}$:

(1) Sort each column of X, individually, to get matrix S.
(2) Take the means across the rows of S and assign it to each element in the row to get S'.
(3) Get the normalized matrix, X', by rearranging each column of S' to have the same ordering as X.

This method can be seen as a special case of the transformation $x'_i = F^{-1}(G(Y_i))$, where the functions F and G are calculated empirically from the datasets being normalized.

3. Empirical Comparison of Normalization Methods

A set of 32 pairs of TnSeq datasets was obtained from various libraries of *M. tuberculosis* Tn-mutants grown under different conditions, with each condition being tested in duplicate. The raw read counts were reduced to unique template counts using sequencing barcodes,[4] though we will continue to refer to them generically as "read counts" throughout this paper. Each dataset had an average of 2.4M total counts, with a range of 1.1–5.4M. Densities (i.e. fraction of TA sites represented in each dataset) were in the range of 38% to 69%.

The BGC was applied to each of the 64 datasets (followed by NZMean normalization). As an example, Table 1 contains statistics for the original datasets A1, A2, B1 and B2 (corresponding to the "*in vitro*" and "Trans02c" datasets among the 32 pairs), as well as the values of ρ and κ estimated by the BGC method. The dispersion parameter κ is lower for the B1 and B2 datasets, consistent with the greater variability that is observed in those datasets. A QQ-plot of the corrected values for dataset B2 is shown in Fig. 5, displaying a much better fit to the Geometric distribution, with the skew removed (compare to Fig. 2).

Table 1. Fitting of parameters for example datasets.

Data set	Total reads	Insertion density	Mean count	Max count	ρ	κ
A1	3.12M	49.3%	84.7	1,693	0.0118	911.1
A2	1.93M	52.6%	49.2	1,175	0.0203	493.9
B1	2.78M	41.1%	89.8	6,009	0.0111	422.0
B2	3.65M	38.1%	128.4	16,146	0.0078	434.7

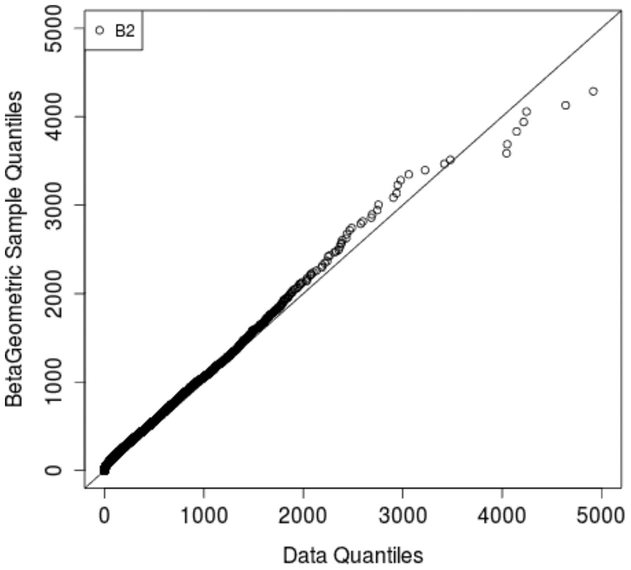


Fig. 5. QQ-plot of the raw read counts for dataset B2, and the Beta-Geometric variables obtained by sampling the parameter p from a Beta distribution with estimated parameters $\rho = 0.0078$ and $\kappa = 434.7$.

One empirical metric we can use to evaluate whether our correction method helps is to compare replicate datasets. In two datasets selected from the same Tn-mutant library under the same growth conditions, one would ideally expect no differences in essentiality of genes. However, in practice, there is usually high variability observed in TnSeq datasets, even between biological replicates. Any method for statistical analysis of TnSeq data has to be conservative enough not to detect many DE genes between replicates. Yet, when using a permutation test (described below) on multiple pairs of replicates, we often observe DE genes, in some cases far beyond what would be expected from random statistical sampling differences. We attribute many of these false positives to the skew inherent in individual datasets. Our goal in this paper is to show that, by fitting each dataset to a Beta-Geometric distribution, we can correct for the skew in read counts, and thereby reduce many of these false positives. This enhanced normalization method could be applied to other TnSeq analysis methods to improve the detection of statistically significant DE genes.

3.1. *Permutation test to identify conditionally essential genes*

In order to evaluate the differential essentiality of a gene between two conditions, possibly with multiple replicates of each, we use a non-parametric permutation test on the corrected and normalized counts at TA sites within the gene. Briefly, the counts are summed over all sites in a gene and replicate to determine the mean in each condition and then the difference is compared to background distribution of means from 10,000 random permutations of the sites. The p-value is calculated from the number of times the observed mean is greater than one of the samples.

Suppose we have m_1 replicates (datasets) in condition A, and m_2 replicates in condition B. Let $\mathbf{C}_{ij}$ be a $(m_1 + m_2) \times n$ matrix of counts at each of n TA sites i within the gene, for each dataset j:

$$\Delta = \frac{1}{n|A|} \sum_{j \in A} \sum_{i}^{n} \mathbf{C}_{ij} - \frac{1}{n|B|} \sum_{j \in B} \sum_{i}^{n} \mathbf{C}_{ij}. \tag{14}$$

Ten thousand random permutations of the counts in matrix $\mathbf{C}_{ij}$ are generated, and Δ' is calculated for each permutation. The p-value is estimated as the number of times $\Delta > \Delta'$ (or $\Delta < \Delta'$ for negative differences).

3.2. *Reduction in type I errors*

To assess the impact of the different normalization procedures when performing a comparative analysis of TnSeq datasets, we compared replicate datasets against each other. Because the datasets in each pair of replicates are selected under the same condition, the expectation is that there should be no DE genes between them. A *false positive* was defined as a gene that had a $p < 0.05$, since no statistically significant differences in essentiality are expected between replicates of the same growth condition. Note that because of the large number of genes in the *M. tuberculosis* genome (i.e. 3,989), the permutation test is expected to incorrectly reject the null hypothesis on as many as 5% of the genes through chance alone.

Table 3 presents the number of false positives obtained by using the permutation test after normalizing with the different methods. Using NZMean normalization as a reference, an average of 71.4 false positives is detected over the 32 pairs of datasets. BGC reduces false positives in 22 out of 32 cases. In comparison to other methods, BGC reduces the most false positives in 14.8 out of 32 conditions (fraction due to ties), which is more than any other normalization method. The next best normalization method was RLE, achieving the greatest reduction of false positives in 7.7 datasets. On average, BGC reduces the number of false positives the most, achieving a mean reduction of 21.7 type I errors overall.

No method achieves a consistent reduction in the number of false positives on all datasets. However, even though false positives are increased in some datasets, the amount of false positives increased by BGC is generally small (i.e. average of 6.4). In addition, most normalization methods tend to increase false-discoveries on the same conditions, suggesting these conditions are problematic for most of the methods. For

Table 2. Effect of skew on the change in the number of false positives (relative to NZMean) after applying BGC, for three representative conditions.

Dataset	Density	NZMean	Skew	Kurtosis	ΔFalse Positives
BXD04 replicate 1	43.8	51.5	44.8	3997.8	−287
BXD04 replicate 2	54.0	86.2	7.9	183.8	
In vitro replicate 1	49.3	84.7	3.2	19.6	2
In vitro replicate 2	52.6	49.2	2.9	16.6	
Trans01c replicate 1	58.3	37.8	7.3	164.4	23
Trans01c replicate 2	65.6	51.4	5.3	61.9	

instance, on condition Trans01c, which was the condition that proved toughest for BGC (increasing false positives by 23), most other methods increased false positives as well. RLE increased false positives by 11, and TMM by 141. Only ZI-NB reduced false positives by two.

Because of the way BGC corrects for the skew in datasets, it is most likely to have a more substantial effect on those cases where there is a large skew between datasets. Table 2 contains some statistics for the datasets for which applying BGC resulted in the largest reduction in read counts (BXD04), and *in vitro* (where the false positives were nearly unchanged). As can be seen, the condition on which BGC performed the best showed a very high skew and kurtosis (third and fourth moments of read counts) between its replicates, whereas the skew and kurtosis in the *in vitro* datasets were much smaller by comparison. For comparison, the skew of a dataset fitting an ideal Geometric distribution will be approximately 2.0 (depends slightly on the mean). The skew in the *in vitro* datasets is quite close to this value, implying they are not very skewed. By correcting the skew in datasets and adjusting them to a Geometric distribution (with a variable parameter), the BGC will have more success in those datasets that are more highly skewed. On those datasets where the read counts are not skewed, BGC is expected to have less of an effect, but these are likely the datasets that would benefit the least from normalization (as is the case for the *in vitro* datasets).

3.3. *Effect on detection of differential essentiality*

So far, the previous sections have focused on the effects of BGC on reducing the number of false positives when comparing replicates of the same condition (where no true positives are expected). It is important, however, to study the effects of BGC on detecting genes when the datasets are grown on different conditions (and thus at least some DE genes, or true positives, are expected). Determining the effects of normalization on detecting true positives is complicated by the fact that it is difficult to determine a (complete) set of genes which are known *a priori* to be DE in the conditions studied. This renders a proper analysis of the true-positive rate between normalization methods prohibitively difficult.

Instead, to study the effects of the normalization method on the comparative analysis between conditions, each pair of replicates for all the *in vivo* conditions was compared against the pair of replicates grown *in vitro*. This way we can get an idea of

Table 3. Change in the number of type I errors relative to the Non-Zero Mean (NZMean) method. False positives are defined as genes with $p < 0.05$ under the permutation test between replicates of the same condition. Methods marked with † have been normalized with the NZMean method after performing the corresponding normalization. The normalization methods compared were Beta-Geometric Correction (BGC), Relative Log Expression (RLE), Trimmed Mean of M-Values (TMM), Zero-Inflated Negative Binomial (ZI-NB), multinomial simulation normalization (MSN), and Quantile Normalization (QNM). Values which show the largest reduction in false positives for each condition are in bold. Mean reduction and the number of times each method achieves the best correction are shown at the bottom (ties are weighted by total number of methods with sharing score).

Condition (pair of replicates)	NZMean	ΔRLE	ΔTMM	ΔZI-NB	ΔMSN†	ΔQNM†	ΔBGC†
AJ	13	−2	1	2	0	1	−3
BL6	74	−20	−21	−12	1	−12	**−25**
BXD01	2	0	1	−1	1	1	0
BXD03	0	**0**	**0**	**0**	**0**	**0**	2
BXD04	535	−328	364	−142	2	−73	−287
BXD05	33	1	99	1	0	−1	13
BXD06	91	−4	82	−6	0	8	−10
BXD07	78	−17	−21	−7	2	−10	−42
BXD08	241	−75	−105	−42	−9	−52	−154
BXD09	6	0	11	0	0	0	−3
CAST	17	−3	−2	1	2	0	−3
CCcont	2	**0**	98	**0**	**0**	**0**	3
DS01	12	−2	14	0	−1	−2	−4
DS02	22	2	13	−1	1	−1	−1
DS04	49	3	79	2	3	2	−17
DS0c	42	−9	−1	−5	−1	−2	−15
GP01	74	−44	−41	−1	−4	−17	−46
In vitro	2	**0**	77	**0**	**0**	**0**	2
PWK	100	3	4	0	2	−3	−3
Trans01	32	−13	−1	−5	−6	−2	−13
Trans01c	62	11	141	−2	1	2	23
Trans02c	84	−37	−33	−11	−1	−12	−28
Trans03	46	1	101	−4	−4	−3	3
Trans03c	52	**0**	121	2	1	2	5
Trans05	142	−7	1	−5	−1	2	−19
Trans05c	30	3	64	4	**1**	2	12
Trans07	158	−11	43	−10	−1	−1	−61
Trans07c	70	−8	−4	−2	1	0	1
Trans09	32	2	27	−4	0	0	−1
Trans09c	78	−27	341	7	1	−1	−3
Trans11	22	−9	3	−6	1	0	−4
Trans11c	85	21	446	−5	4	−2	−18
Mean	71.4	−17.8	59.4	−7.9	−0.13	−5.4	−21.7
No. of Best	N/A	7.7	0.2	4.5	2.2	2.5	14.8

how the normalization methods would affect the overall number of significant hits (though we cannot say for certain whether this leads to more true positives or not). Table 4 contains the total number of genes labeled as DE (relative to *in vitro*) after normalizing with each of the procedures. DE genes were those which were assigned an adjusted p-value of $q < 0.05$ (using the Benjamini–Hochberg correction for multiple comparisons). On average, the TMM method tended to predict more genes as

Table 4. Number of genes classified as Differentially Essential (DE) by the permutation test after applying the different normalization methods. DE genes are defined as genes with $q < 0.05$ under the permutation test between a pair replicates of the given condition and a pair replicates grown *in vitro*. Methods marked with † have been normalized with the Non-Zero Mean (NZMean) method after performing the corresponding normalization. The normalization methods compared were Beta-Geometric Correction (BGC), Relative Log Expression (RLE), Trimmed Mean of M-Values (TMM), Quantile Normalization (QNM), Zero-Inflated Negative Binomial (ZI-NB), and multinomial simulation normalization (MSN).

	# DE GENES						
Condition (versus *in vitro*)	NZMean	RLE	TMM	ZI-NB	MSN	QNM	BGC
AJ	441	486	652	436	37	436	431
BL6	383	323	249	303	282	367	280
BXD01	366	372	421	355	37	347	301
BXD03	330	432	355	321	11	330	281
BXD04	315	273	266	302	13	307	254
BXD05	308	381	315	301	38	306	248
BXD06	356	338	412	346	36	337	299
BXD07	301	388	315	298	38	299	281
BXD08	299	416	258	288	37	289	247
BXD09	329	535	320	338	43	326	317
CAST	460	478	819	461	36	461	466
DS01	387	553	400	381	43	384	363
DS02	379	654	371	367	37	382	338
DS04	336	334	511	315	35	334	235
DS0c	323	431	326	324	37	329	299
GP01	844	481	852	628	545	763	507
PWK	453	478	559	428	33	436	408
Trans01	307	268	423	286	43	308	109
Trans01c	36	61	44	38	37	35	64
Trans02c	398	290	253	306	284	380	287
Trans03	266	257	425	255	37	259	202
Trans03c	91	84	223	77	35	87	124
Trans05	283	841	351	277	35	274	200
Trans05c	149	1,208	833	86	37	152	272
Trans07	282	734	409	272	39	277	226
Trans07c	39	142	35	42	42	45	100
Trans09	524	278	477	446	3	497	137
Trans09c	24	34	72	27	40	25	74
Trans11	695	307	1,164	563	2	535	154
Trans11c	43	76	86	33	34	35	83
Mean No. of DE	325	398	407	297	67	312	253

DE, with a mean of 406 DE genes, followed by RLE with a mean of 398. On the other hand, Multinomial Simulation Normalization (MSN) showed a tendency to consistently predict the least number of DE genes, predicting an average of 67 genes as DE. The BGC method falls in between, predicting an average of 253 genes as DE.

To further explore the effect of normalizing with the BGC method, we plotted the number of DE genes detected before and after applying BGC normalization (see Fig. 6). A slight reduction in the number of DE genes identified is seen in most conditions (possibly representing a decrease in the number of false positives obtained

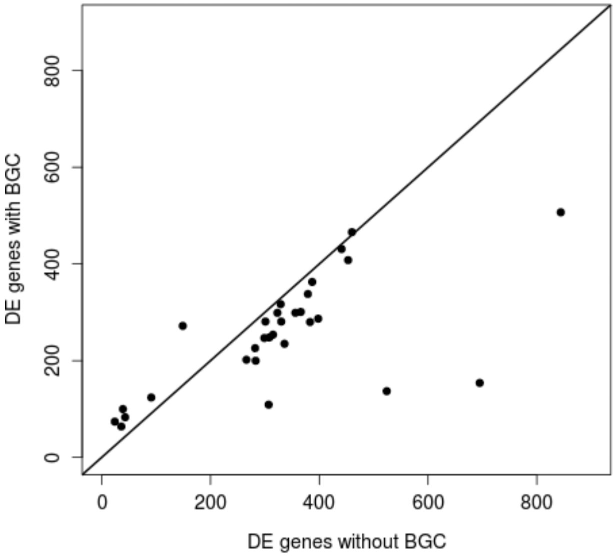


Fig. 6. Scatter plot of the number of the DE obtained with and without applying BGC. The solid black line represents the identity line. Applying BGC results in a reduction in the number of DE genes identified in most conditions, possibly representing a reduction in false positives. In addition, BGC produces results which are less extreme, increasing the number of DE genes identified when this number is low and decreasing the number of DE genes identified when it is very high.

by correcting for the skew). This shows that the reduction in false positives between replicates is not achieved at the cost of a dramatic reduction in overall DE genes detected between conditions. However, when the number of genes classified as DE is low (due to possible under-detection of true positives), the BGC procedure tends to increase the number of DE genes predicted. On the other hand, when the number of DE genes predicted is exceedingly high (> 500), BGC normalization significantly decreases the number of DE genes predicted. This phenomenon suggests that applying BGC adjusts datasets so that they produce results that are less extreme in terms of number of DE genes detected.

4. Discussion

Analysis of TnSeq data has become a valuable tool for determining DE genes. However, the large amount of intrinsic variability that is observed in these datasets (e.g. read counts) makes direct comparison between datasets problematic. Common ways of normalizing the datasets have focused primarily on a linear transformation of read counts between datasets,[16,17] usually by making their mean read counts comparable. While important, normalization of the means alone is not enough to correct for the large skew that is observed in some datasets.

Other nonlinear normalization methods have been proposed in the past to overcome the limitations of scaling datasets by a constant factor.[15,18] Indeed, the BGC method is similar to QNM,[18] except traditional QNM scales datasets together based on an empirical distribution function, without making assumptions about the form of the distribution. On the other hand, the simulation-based approach taken by ARTIST is fundamentally different.[15] It attempts to simulate the effects of selection on the control dataset, by sampling read counts from a multinomial distribution to obtain a new, simulated, control sample that has approximately the same number of reads and saturation.

We proposed the BGC method for adjusting datasets for comparative analysis. This method showed the largest overall reduction in false positives out of all the normalization methods studied. What sets BGC apart from most of the other methods evaluated is the fact that it is a nonlinear transformation of the data that is based on adjusting observed reads to an ideal distribution. It assumes that the skew in read counts comes from dispersion in the parameter p underlying a Geometric distribution. The skew is captured by fitting the data to a Beta-Geometric distribution, which allows the parameter p of the Geometric distribution to vary according to a Beta distribution. The original read counts are then adjusted back to an ideal Geometric distribution by matching quantiles. This approach is non-linear, with high-counts (spikes) being reduced and unusually suppressed counts increased. We choose to correct read counts back to a Geometric distribution (with a variable parameter), since such a profile of abundances at different TA sites (i.e. high proportion of low counts, low proportion of high counts) would be expected from sampling from a population of competing cells with a range of growth rates.

In addition to reducing false positives in replicate datasets from the same condition, we examined the effects of applying BGC when comparing datasets of different conditions (where at least some true positives are expected). While it is difficult to say with certainty how the BGC method affects the detection of true DE genes, we showed that in most cases it tends to decrease the number of DE genes slightly, likely due to reducing false positives. As the overall reduction was relatively small, this suggests that the reduction of type I errors that is seen when comparing replicates of the same condition does not come at the expense of a large reduction in the overall number of positives detected.

One potential limitation of BGC, along with most of the normalization methods examined here (except ZI-NB and MSN), is that they do not take the saturation (or density) of the data into account when adjusting reads. Accounting for different saturation levels is especially important when comparing datasets from different libraries, where saturation levels can be significantly imbalanced due to differences in biological selection. ZI-NB and MSN take into consideration the differences in saturation of the libraries in their own ways (ZI-NB by using a mixture model to allow the NB distribution to include some, but not all, empty sites; and MSN by adjusting the saturation of the control dataset). Despite this limitation, BGC actually produces a larger reduction in false positives compared to ZI-NB and MSN. This suggests that correcting for the skew in datasets may be more important for reducing false positives than accounting for the difference in saturation, particularly for the well-saturated datasets such as those examined here (with insertion densities in the range of 38% to 69%). A future direction for this work could be to modify BGC so that it takes into consideration the differences in saturation levels between datasets.

Acknowledgments

This work was supported by NIH grant U19 AI107774 (TRI). We thank Chris Sassetti for providing the TnSeq datasets used in this study.

Appendix A. Derivation

To minimize the SSE, we find the root of the derivative of SSE with respect to κ:

$$\frac{\partial SSE}{\partial \kappa} = \frac{\sum_i^N 2(2\rho-1)\log(1-q_i)\big(\log(1-q_i) - X_i \log\left(\frac{-\rho\kappa+\kappa-1}{\kappa-2}\right)\big)}{(\kappa-2)((\rho-1)\kappa+1)\log^3\left(\frac{-\rho\kappa+\kappa-1}{\kappa-2}\right)} = 0.$$

To facilitate finding the root we ignore the denominator and remove constant terms from the numerator as these do not affect the final result:

$$\sum_i^N \log(1-q_i)\left(\log(1-q_i) - X_i \log\left(\frac{-\rho\kappa+\kappa-1}{\kappa-2}\right)\right) = 0,$$

$$\sum_i^N \log^2(1-q_i) = \sum_i^N X_i \log(1-q_i)\log\left(\frac{-\rho\kappa+\kappa-1}{\kappa-2}\right),$$

$$\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)} = \log\left(\frac{-\rho\kappa+\kappa-1}{\kappa-2}\right),$$

$$\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] = \frac{-\rho\kappa+\kappa-1}{\kappa-2},$$

$$(-\kappa+2)\times\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] = \rho\kappa-\kappa+1,$$

$$-\ \kappa\times\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] + 2\times\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] = \rho\kappa-\kappa+1,$$

$$2\times\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] - 1 = \kappa\left(\rho-1+\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right]\right),$$

$$\kappa = \frac{2\times\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] - 1}{\exp\left[\frac{\sum_i^N \log^2(1-q_i)}{\sum_i^N X_i \log(1-q_i)}\right] + \rho - 1}.$$

References

1. van Opijnen T, Camilli A, Transposon insertion sequencing: A new tool for systems-level analysis of microorganisms, *Nat Rev Microbiol* **11**(7):435–442, 2013.
2. Lampe DJ, Churchill ME, Robertson HM, A purified mariner transposase is sufficient to mediate transposition *in vitro*, *Eur Mol Biol Organ J* **15**(19):5470–5479, 1996.
3. Rubin EJ, Akerley BJ, Novik VN, Lampe DJ, Husson RN, Mekalanos JJ, *In vivo* transposition of mariner-based elements in enteric bacteria and mycobacteria, *Proc Natl Acad Sci* **96**(4):1645–1650, 1999.
4. Long JE, DeJesus M, Ward D, Baker RE, Ioerger TR, Sassetti CM, Identifying essential genes in Mycobacterium tuberculosis by global phenotypic profiling, *Methods Mol Biol* **1279**: 79–95.
5. Sassetti CM, Boyd DH, Rubin EJ, Genes required for mycobacterial growth defined by high density mutagenesis, *Mol Microbiol* **48**(1):77–84, 2003.
6. Sassetti CM, Rubin EJ, Genetic requirements for mycobacterial survival during infection, *Proc Natl Acad Sci* **100**(22):12989–12994, 2003.
7. Gawronski JD, Wong SMS, Giannoukos G, Ward DV, Akerley BJ, Tracking insertion mutants within libraries by deep sequencing and a genome-wide screen for haemophilus genes required in the lung, *Proc Natl Acad Sci* **106**(38):16422–16427, 2009.
8. Griffin JE, Gawronski JD, DeJesus MA, Ioerger TR, Akerley BJ, Sassetti CM, High-resolution phenotypic profiling defines genes essential for mycobacterial growth and cholesterol catabolism, *PLoS Pathog* **7**(9):e1002251, 2011.
9. Smith V, Chou KN, Lashkari D, Botstein D, Brown PO, Functional analysis of the genes of yeast chromosome V by genetic footprinting, *Science* **274**:2069–2074, 1996.
10. Akerley BJ, Rubin EJ, Camilli A, Lampe DJ, Robertson HM, Mekalanos JJ, Systematic identification of essential genes by *in vitro* mariner mutagenesis, *Proc Natl Acad Sci USA* **95**:8927–8932, 1998.
11. Zhang YJ, Ioerger TR, Huttenhower C, Long JE, Sassetti CM, Sacchettini JC, Rubin EJ, Global assessment of genomic regions required for growth in *Mycobacterium tuberculosis*, *PLoS Pathog* **8**(9):e1002946, 2012.
12. Zomer A, Burghout P, Bootsma HJ, Hermans PW, van Hijum SA, ESSENTIALS: software for rapid analysis of high throughput transposon insertion sequencing data, *PLoS ONE* **7**(8):e43012, 2012.
13. DeJesus MA, Zhang YJ, Sassetti CM, Rubin EJ, Sacchettini JC, Ioerger TR, Bayesian analysis of gene essentiality based on sequencing of transposon insertion libraries, *Bioinformatics* **29**(6):695–703, 2013.
14. DeJesus MA, Ioerger TR, A hidden Markov model for identifying essential and growth-defect regions in bacterial genomes from transposon insertion sequencing data, *BMC Bioinform.* **14**:303, 2013.
15. Pritchard JR, Chao MC, Abel S, Davis BM, Baranowski C, Zhang YJ, Rubin EJ, Waldor MK, ARTIST: High-resolution genome-wide assessment of fitness using transposon-insertion sequencing, *PLoS Genet* **10**(11):e1004782, 2014.
16. Anders S, Huber W, Differential expression analysis for sequence count data, *Genome Biol.* **11**(10):R106, 2010.
17. Robinson MD, Oshlack A, A scaling normalization method for differential expression analysis of RNA-seq data, *Genome Biol* **11**(3):R25, 2010.
18. Bolstad BM, Irizarry RA, Astrand M, Speed TP, A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics* **19**(2):185–193, 2003.

19. Robinson MD, McCarthy DJ, Smyth GK, edgeR: A Bioconductor package for differential expression analysis of digital gene expression data, *Bioinformatics* **26** (1):139–140, 2010.
20. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B, Mapping and quantifying mammalian transcriptomes by RNA-Seq, *Nat. Methods* **5**(7):621–628 (2008).

Michael A. DeJesus received his Bachelor's degree in computer science in 2008 from the University of Puerto Rico, Mayaguez. He received a Master's degree in Computer Science from Texas A&M University in 2012. Currently, he is a Ph.D. student in Computer Science at Texas A&M. His research focuses on using machine learning and statistical pattern recognition techniques to analyze sequence data.

Thomas R. Ioerger graduated with honors from The Pennsylvania State University in 1989, securing a B.S. in Molecular and Cell Biology. He received an M.S. and Ph.D. in Computer Science from the University of Illinois in Urbana-Champaign, the latter in 1996. He is an associate professor in the Department of Computer Science and Engineering at Texas A&M University. His primary research interests are in the areas of bioinformatics and machine learning.