Statistics and Its Interface Volume 8 (2015) 477–494

Detecting bacterial genomes in a metagenomic sample using NGS reads

Camilo Valdes, Meghan Brennan,
Bertrand Clarke*, and Jennifer Clarke

We use a nucleotide flipping technique on whole genome next generation sequencing (NGS) data to test for the presence of various bacterial strains in a single metagenomic sample. Our technique is novel in that we induce artificial point mutations at the nucleotide level to define a test statistic for each genome on a given reference list. After finding a suitable nucleotide flipping rate, we use a variant of the Westfall-Young procedure to correct for multiple comparisons. When we align reads to reference genomes we permit fractional reads i.e., we weight the contribution of each read by one over the number of genomes to which it aligns. In a large scale simulation we characterize our method's performance on 'clean' data with respect to accuracy, genome lengths and genome abundances. Then, we apply our technique to real data from the Human Microbiome Project (HMP). We compare our results based on adjusted p-values with the HMP findings based on abundance, as assessed by coverage. The results from the two methods have substantial overlap; discrepancies can be explained by the inherent variability of the respective processing pipelines and data.

AMS 2000 subject classifications: Primary 62G10, 62P10; secondary 62-07.
Keywords and phrases: Metagenomics, Next generation sequencing, Human Microbiome Project, Multiple comparisons, Nucleotide flipping, Artificial point mutations.

1. PROBLEM STATEMENT

We present a technique for detecting the presence of strains of bacteria in metagenomic samples generated using whole metagenome high-throughput DNA sequencing, or Next-Generation Sequencing (NGS). That is, we assume that genomic reads have been generated from a single physical sample and the goal is to do J frequentist hypothesis tests

$$\begin{aligned} &\mathcal{H}_{0,j} : C_j \text{ is not in the population} \\ &\textit{vs. } \mathcal{H}_{1,j} : C_j \text{ is in the population,} \end{aligned} \tag{1}$$

where the C_j's for $j = 1, \ldots, J$ represent a collection of reference genomes for bacterial strains. The collection of references includes consensus nucleotide sequences of different bacterial DNA (chromosomes, plasmids, and other contigs) that comprise the strain level. The efficacy of the testing will depend on J, the richness of the reference genome database across bacterial taxa, as well as the quality of the NGS data. Typically, J will be on the order of thousands, the number of chromosomes/plasmids/contigs will be on the order of hundreds of thousands, while the number of reads will be in the hundreds of thousands to millions. To accomplish the tests in (1) we use the data to test at the chromosome/plasmid/contig level, and then aggregate these results to the strain level. There may be genomes in the sample that are not among the C_j's and may not even be bacterial; if reads from such genomes are in the sample they usually end up not being used in the alignment to the references and this is not a problem for our technique. Indeed, regardless of what non-bacterial genomes may be present in a physical sample, there is interest in determining which bacterial strains are in the sample. For instance, accurately identifying which bacterial taxa (strains, species, etc.) are present in a single set of reads is important in food safety, iatrogenic diseases, and bioterrorism.

Several methods for assessing the composition of metagenomic samples using NGS data have been presented in the literature. For instance, [28] used 40 universal marker genes to define bacterial species genome references and considered a species to be present in a sample if $\geq$40% of the reference genome was covered by reads. Also, [10, 11] have reported on various approaches for metagenomic profiling. In examining the healthy human microbiome, they used taxonomic profiling with MetaPhlan ([29, 30]) to estimate species-level abundances in each sample. Separately, [24] developed a metagenome reference gene set (similar to MetaHIT ([23])) to identify genes present in metagenomic samples and infer the presence of bacterial species from these genes. In general these approaches use information at the gene level to infer which species or taxa are present in a sample and are not primarily statistical. By contrast, the methodology we present here is primarily statistical and designed to detect bacteria at the strain level using nucleotide information. It does not require extra biological information such as marker

*Corresponding author.

genes that might not be available for genomes present in metagenomic samples of unknown provenance.

There are three senses in which our work is novel. First, we use NGS data at the nucleotide level, in particular for the artificial point mutations defining the test statistics, rather than initially aggregating. Second, the 'flipping and re-aligning' approach that we take to find both raw p-values and a correction for multiple comparisons seems to be different from any standard testing procedure that has been proposed, at least in the present context. Third, we weight the contribution of each read by one over the number of reference genomes to which it aligns. That is, in the test statistics, the reads in the sample are weighted by their abilities to discriminate among bacterial strains. Thus, sequences that align to many strains contribute relatively little to each strain. Abstractly, this means we have a technique for detecting reference strings using data that is of the form of a large but finite collection of substrings drawn from an alphabet of four letters and this technique can be used whenever tests can be phrased in terms of properties of the collection of reference strings.

First, while we have used our technique on the nucleotide sequence from a specific version of the NGS technology, there are many variations; a convenient summary can be found in [18] and an overview from a statistical perspective can be found in [5]. Thus, we only use the read data from an NGS data set and we only use alignment to a set of reference genomes (with a tolerance for a few mismatched nucleotides). While there are a variety of more-or-less standard analyses for NGS data, e.g., condense NGS data and apply contingency table analyses or analysis of variance techniques to detect differences between groups, see [1], or detect whether multiple copies of a region are present in a piece of DNA, see [32] and [14], one common feature of these and other statistical approaches is that they summarize the NGS data before analyzing it. By contrast, our method of analysis focuses on the level of nucleotides. This has been done for metagenomic variation analysis; see [28]. However, we are using whole genome nucleotide information for microbial detection whereas [28] took a marker gene approach to detection.

The second point, that our testing procedure is novel, rests on an innovative construction for the null population. Essentially, we use point alterations or 'flipping' of nucleotides, to construct a null population at the nucleotide level, i.e., a collection of artificial samples that are as similar to the real data as possible and can be assumed not to contain any of the C_j's. The construction uses a fixed probability $q \in [0, 1]$ so as to flip $100q\%$ of the nucleotides. When we flip a nucleotide there are three choices and we assume all three are equally likely. Doing this repeatedly for the reads and re-aligning to the reference genomes gives us a collection of plausible read counts from the null population to serve as a baseline. We can therefore compare the number of reads aligning for each strain in the data to the number of reads aligning from the sample from the null population (for the chosen q).

This requires some explication to justify. Biologically, we want the null population and the samples we generate from it to look like a metagenomic sample but not contain any of the C_j's. If q were chosen to be too small, the artificial samples generated from the reads would be very close to the reads themselves and it would be difficult to reject the null even when it's false; the actual data would not differ meaningfully from any reasonable threshold obtained from the artificially generated reads. So, we want to put in enough noise, i.e., choose a big enough q, that it is plausible the references are not in the null population. Loosely, if the null is too biologically plausible it won't represent a null population. At the same time, we do not want to put in so much noise i.e., make q too big, that comparing the alignments of the flipped reads essentially always leads us to reject the null because the null has lost all genomic plausibility.

In statistical terms, this can be phrased in terms of a tradeoff between false rejections and false acceptances of the null. Constraining the null population by insisting the nucleotide flipping results in strings that are representative of a viable bacterium, or keeping q too small, weakens the test by making it too hard to reject the null. On the other hand, making q too large moves the null so far away from any viable bacterium that it becomes irrelevant. That is, the null population is shifted too far toward collections of nucleotide strings that do not represent viable bacteria and hence will be rejected too often.

In either case, we take artificial samples (given q) as representative of the statistical null population and re-align them to the reference genomes so that the number of reads that align to C_j in the observed sample can be compared with the distribution of the number of mutated reads that align to C_j. This can be used to generate a p-value for the test in (1). That is, it is only at the end of our procedure that we aggregate over the artificial, altered samples. (In fact, in our procedure, we do not use the p-values directly; we use the ranks of the raw counts which are equivalent to the p-values.)

In both the biological sense and the statistical sense, q controls the tradeoff between two unacceptable extremes. So, to implement our procedure we present two ways to choose q as a suitable tradeoff, i.e., as 'knees in curves'. In an example, we see that these give similar values for q and hence comparable results. It is well-known that the knee in a curve represents the point of smallest radius of curvature and corresponds to a maximal second derivative condition.

The third sense in which our work is novel is that we use fractional reads. That is, if a read aligns to, say, g reference genomes, each of the alignments is counted as $1/g$. That is, our procedure represents an averaging over the uniquenesses of the reads. Obviously, the more uniquely a read aligns the more weight it gets. For some reads that might uniquely specify a strain this may fail to make use of all the information that could be available (but usually is not). On the

other hand, our method provides an average alignment for the vast majority of reads that do not seem to identify a strain uniquely by themselves. That is, we are weighting the alignments for each read by their uniqueness and doing this for many reads. This seems to provide a good summary of the overall information in an NGS data set relative to the reference list.

The rest of this paper is as follows. In the three subsections of Sec. 2 we describe our nucleotide flipping approach, our usage of the Westfall-Young multiple comparisons correction, and the way we choose the null distribution for (1), i.e., q. In Sec. 4 we use our technique on a simple HMP data set, examine the selection of q, and compare our results to those from the HMP. In Sec. 5 we conclude by discussing some of the issues raised by our method. Some computational details are provided in an Appendix.

2. METHOD

The primary difficulty in doing the tests (1) is identifying appropriate null distributions for them. This is the case for (i) finding raw p-values for the individual tests and (ii) finding a multiple comparisons correction for the raw p-values. We solve (i) by using a flipping rate q on the nucleotides in the reads in the NGS data. For (ii) we use a variation on the Westfall-Young method (WYM). Standard WYM requires permutations, but here we use the same artificial point flips as in (i) in an analogous procedure. Both of these necessitate choosing q since it affects the number of C_j's detected.

2.1 A test using nucleotide flipping

The basic idea behind our nucleotide flipping test for the $\mathcal{H}_{0,j}$'s is to create many new data sets so that values of the test statistic under the null hypothesis can be found. These values are then used to form a histogram estimate of the sampling distribution of the statistic under a null hypothesis, and the p-value is taken to be the area to one side (for a one sided test) of the actual value of the test statistic. The null population is all values of the statistic that one might have got via flipping the nucleotides in the reads and conceptually the null distribution is the limit of the histogram as the sample size increases.

More formally, we use artificial point mutations or flipping of the nucleotides in the reads in the NGS data from which the test statistic, defined below, is calculated. This effectively creates a new data set, one that we might have observed if the null hypothesis were true. By repeating this process we can generate many such datasets, so that many values of the test statistic can be found. To do this, fix a value $q \in (0,1)$ and let $r_k,\ k = 1,\ldots,K$, denote the sample reads, each of length l_k. To get a $q*100\%$ nucleotide flipping rate, we choose ql_k nucleotides in r_k to flip to another nucleotide. We do this by drawing one number from a $Unif[0,1]$ for each nucleotide. If the number drawn for a given nucleotide is less than q we change it at random to one of the other three nucleotides, otherwise we leave it unchanged. Once we have done this to all K reads we have a collection of K flipped reads which we can align to the reference genomes C_j.

When we do this alignment of the artificial data to the reference list, we continue to use fractional reads. That is, some reads will align to a single reference, discriminate that reference from all others, and get weight $1/g = 1$. However, many reads will align to multiple, say g, references, provide less discriminative information, and be weighted $1/g,\ g > 1$.

Now, for each j let Y_j be the number of the flipped reads that align to C_j for a given q. Then $Y_j = \sum_k^K (1/g_k) 1_{r_k,j}$ where g_k is the number of references to which read r_k aligns and $1_{r_k,j}$ equals 1 if read r_k aligns to reference C_j and zero otherwise. Thus we generate a column vector $V = (y_1,\ldots,y_J)^T$ of non-negative values which may or may not be integer values depending on the uniqueness of the reads aligning to each reference. The entries y_j are the outcomes of the Y_j's. We can write $V = V_1$ and repeat the procedure $(M-1)$ more times to generate $V_2,\ldots,V_M$. These form a $J \times M$ matrix $V = (V_1,\ldots,V_M)$ in which each row contains M values that are exchangeable – in fact, independent.

There are two comments on this procedure that are important at this time. First, the number of aligned reads show little variation from V_i to V_j for $i \neq j$ for each fixed q. We do not have a good explanation for this; we merely observed it phenomenologically in the data sets we examined. We suggest it may be related to the tolerance of the aligner; for a given read and reference the aligner is sensitive to the number of mismatches but not where they occur. That is, since the rate of the flips is the same for given q, it is only the locations of the flips that can vary and within a given read/reference pair an aligner will be insensitive to this. Second, generating M artificial data sets and realigning them gives a nominal resolution of approximately $1/M$ for p-values. While sufficiently powerful computing resources would let us reduce this effectively to zero, it would be meaningless compared to the other sources of error and variability in the experiment. The sizes of p-values, at best, only indicate that the null is reasonable or not – at the resolution of information permitted by the sample and the other sources of error in the experiment. At this time, it does not seem anyone has assessed the various errors involved in NGS experiments in enough detail to permit a well-defended selection of a level for testing. Indeed, research in the general direction of examining sources of error tends to focus on quality control for data more than evaluating the relative contribution of different sources of error, see [31] and [20].

Using the flipped data as a sample from the null population is a non-standard approach to defining a hypothesis test. So, it is worthwhile examining what the construction means. First, it is important to realize that the probability of flipping a collection of reads in an NGS data set so it looks like it came from C_j when it didn't is vanishingly

small. This is so because the number of possible flipped data sets is exponential in the number of base pairs, i.e., the locations of the bases to be flipped are random and uniform over the possibilities, and the number of locations at which a flip may occur is large, namely, $\sum_{k=1}^{K} l_k$ where the l_k's are in the hundreds and K is several hundred thousand. At the same time, the set of genomes corresponding to viable organisms is likely sparse in the set of all strings of 4 letters of length $\mathcal{O}(10^8)$. Therefore it is safe to assume that essentially none of the collections of flipped reads will look like they came from C_j unless they actually did, i.e., they are valid alternatives.

Recall that Y_j is the number of the flipped reads that align to C_j for a given q. Let X_j be the random variable representing the number of reads from C_j in a sample of size K for a given j. We have found through computing various examples that typically Y_j decreases with q and in particular for any q, $Y_j \leq X_j$ holds approximately. In fact, under $\mathcal{H}_{0,j}$, $Y_j \approx X_j$ for any q. This makes sense because, as noted above, it is very unlikely to flip the nucleotides in a string so they match part of a C_j, especially as q increases. Given this, consider the J rows of V. They can be used to form histograms h_j each approximating the sampling distribution of Y_j under $\mathcal{H}_{0,j}$. Now, one minus the quantile of X_j under h_j serves as an estimate of the p-value. To do this properly, however, requires we specify a rule to break ties in the case that x_j happens to equal one or more of the y's from the M mutated data sets. Our rule is that we always assign x_j its largest rank among the y_j's, making the test slightly conservative. Thus, if we test a single hypothesis (hence ignoring multiple comparisons problems), we would reject $\mathcal{H}_{0,j}$ if $X_j = x_j$ is too high, in particular above the 95^{th} percentile of h_j because each test of the form (1) is one-sided when X_j (or Y_j) is used as the test statistic. Since this can be done for each C_j we have J distribution-free raw p-values.

2.2 Westfall-Young correction for multiple comparisons

We resolve the multiple comparisons problems with the J hypothesis tests by adjusting the p-values using the WYM, see [34]. In fact, there are several methods, the two most common being the WYM and false discovery rate (FDR). We prefer to mimic the WYM because it has optimality properties in terms of power, see [17] which do not seem to have been shown (and may not hold) for the FDR. The WYM also takes dependence of the tests into account whereas any virtues of FDR do not seem to hold outside a few specific dependence settings whose assumptions cannot be easily verified. Specifically, we use the '$\min P$' form of the WYM rather than its '$\max T$' form. The reason is that even though appropriate statistics T_j can be defined here, one cannot assume any identicality or specific form of dependence for them. In these cases, the $\min P$ and $\max T$ methods do not in general agree and the $\min P$ methods are preferred, see [6]. The reason is that the $\min P$ method is less sensitive to the lack of identicality and independence across tests than the $\max T$ method.

Let p_j be the raw p-value from the j-th test of the form (1) found in Sec. 2.1 and let P_j be the raw p-value for $\mathcal{H}_j$ as a random variable. Then the j-th step down $\min P$ adjusted p-value is

$$P(\min_{\ell=1,\ldots,J} P_\ell \leq p_j \mid \mathcal{H}), \tag{2}$$

where $\mathcal{H}$ is the complete null, $\mathcal{H} = \cap_{j=1}^{J} \mathcal{H}_j$, see [6]. Procedures based on the $\min P$ (or $\max T$) adjusted p-values control the familywise error rate weakly, i.e., under the complete null rather than an exact (correct) or 'all possibilities' (strong) null, under all conditions.

To implement the $\min P$ method, note that only the raw p-values, the p_j's from Sec. 2.1, are known. So, we must estimate (2) because the distributions of the P_j's are unknown. To do this we use the same flipping technique on the nucleotides as in Sec. 2.1. That is, our procedure for estimating the single step $\min P$ adjusted p-values fixes a q and then flips the bases in the K sample reads to generate, say, 100 flipped samples each containing K short reads. Note that the traditional double permutation algorithm for the step-down $\min P$ adjusted p-values uses the same type of permutations on the data to generate the raw p-values as it does to find the multiple comparisons correction.

Specifically, for each of the M flipped data sets, find the raw p-values for the $\mathcal{H}_{0,j}$'s, $p_{j,m}$, $m = 1, \ldots, M$ and estimate (2) by

$$\hat{p}_j = \frac{\#\{m \mid \min_{\ell=1,\ldots,J} p_{\ell,m} \leq p_j\}}{M}. \tag{3}$$

Clearly, this is just a permutation test on the level of the p-values rather than on the level of the data (where the analog of the permutations is the flipping of nucleotides rather than actually permuting the data). So, putting the null hypotheses in the same order as the $\hat{p}_{(j)}$'s, we reject $\mathcal{H}_{0,(j)}$ when $\hat{p}_{(j)} \leq j/J$.

2.3 Choice of flip rate q

The value of q influences the number of reads in a flipped sample that align to a set of C_j's. So, as discussed in Sec. 1, we want to choose q to get a reasonable null population for comparison purposes in the test. Here, we describe two ways to do this. These are exemplified in Secs. 3 and 4.

First, consider generating M flipped samples using a fixed value of q. Align these samples to the reference genomes using fractional reads and identify the reads from each sample which align successfully. Find the number of reference genomes to which each aligning read in each flipped sample aligns. If these raw numbers are converted to proportion aligning and graphed as a function of q the graph typically starts at a value near 1 for $q = .02$ for simulated essentially

'perfect' data, around .4 or .5 for $q = .02$ for clean real data but may only start at .2 for $q = .02$ for messy data. In any case, this proportion aligning graph decreases monotonically to zero, meaning that as q increases it is harder and harder to get alignment until at some point essentially no reads align to the references. This can be done on the strain or chromosome/plasmid/contig level. These graphs usually have a 'knee' that can be used to identify an optimal q as a tradeoff between specificity and sensitivity.

This tradeoff can often be seen more clearly if one produces a series of boxplots (one for each q) representing the average number of genomes to which each read aligns. Essentially, the knee in the curve formed by the medians in the box plots is the result of the sparsity of viable genomes in the set of all strings of four letters of length $\mathcal{O}(10^8)$ as discussed in Subsec 2.1. For small q, the curve decreases gradually with q because the references to which the strings are aligned form an increasingly sparse subset within the set of all strings. Past a certain value q_0 of q the curve levels off because the flipping of bases has made the reads far from the reads of any viable organism. Hence, from these boxplot charts we can again estimate q_0 by identifying the knee in the curve as a tradeoff between an excessively conservative test and an excessively liberal test, respectively.

In a second approach to estimating q, again as a tradeoff, one can plot a graph of the number of reference sequences detected as a function of q. As with the earlier two curves, these detection plots also have a knee whether one looks at the strain or chromosome/plasmid/contig level, and we can estimate q_0 by identifying the knee in the curve. In this case, the tradeoff is again between the same extremes: sensitivity vs. specificity, conservative testing vs. liberal testing, Type I error vs. Type II error, false positive vs. false negative. When the estimates of q from the two sorts of graphs are similar, it is reasonable to infer that the common value represents an appropriate tradeoff. Note that this a tradeoff between two competing factors e.g., false positive and false negative rates, rather than controlling one and ignoring the other. It will turn out that the tradeoff achieved by our choice of q tends to limit false positives much more than false negatives but this is a result of our way to choose q, not a property built into our methodology *a priori*.

Taking the knee of the curve as a good choice for q can be justified in three different ways. First, pragmatically, if it exists, and it generally does, the knee represents a tradeoff between q too small and hence a null population that is too similar to the sample (meaning it will not be rejected often enough), and q too large and hence a null population that is so far from the sample that the null will be rejected too frequently. Second, the knee in the curve represents the point of diminishing returns i.e., where the gains from the increased null are outweighed by the losses of the increased null. This interpretation for the knee in the curve is routinely invoked in some contexts, e.g., in choosing the number of principal components to use in a principal component regression analysis (called a scree plot), the number of clusters to use in a clustering (see [26]), or choosing a classifier (choose the classifier represented by the point on the ROC curve closest to (0,1)).

Third, from a more formal standpoint, the knee in the curve is the point at which the curve has the smallest radius of curvature i.e., the direction of the curve is changing most rapidly. While often viewed as a 'folk theorem', there is an extensive literature on using this point as a good way to choose auxilliary parameters such as q. For instance, [9] provided a summary of the debate surrounding the use of the knee in the curve admitting that some regard the knee in the curve as ill-defined or not meaningful. However, [27] had already proposed formalizing the concept by using the curvature function of a curve in the plane and [2] used this definition to estimate proportions in a metagenomic sample. In a Bayes version of the present analysis, [4] used the knee in the curve to identify a tuning parameter for use in the analysis of NGS data. Most recently, [3] simplified this definition to a second derivative condition and established consistency in a genomic context. Although their proof does not directly apply to the present NGS setting, the overall import of this work suggests that the knee in the curve, as used here, is a valid concept.

3. SIMULATION RESULTS

As a first test of the method outlined in Sec. 2, we set up a study to see how well it would perform on simulated data. We chose the reference genomes to be 600 randomly selected bacterial strains from the National Center for Biotechnology Information (NCBI) database GenBank. Specifically, the 600 were drawn at random from the Bacteria dataset that only contains complete genomes. In this context, 'complete' genomes are those that have been completed down to a unique DNA sequence per replicon (chromosomes). This means the reference list is very 'clean' – it has one consensus nucleotide sequence for each strain including chromosome(s), plasmids, and all other genetic scaffolds. Moreover, the nucleotide sequence is well annotated (verified from other sources, checked for other errors, etc.) and is in one file for easy use. For more details, see http://thegenomefactory.blogspot.com/2012/07/navigating-microbial-genomes-on-ncbi.html.

To generate a synthetic metagenomic NGS sample we used the MetaSim software, see [25]. It requires three inputs, namely, the genomes, their abundances, and the number of short reads to generate. Since our goal was to form 10 artificial NGS samples, we began by randomly separating the 600 strains into ten groups of size 60 each. Then, for each group of size 60 we generated 60 abundance values. The abundance values are generated to be consistent with characteristic ways that species of organisms are distributed within an ecological community. One of the classical ways is described as logarithmic, see [16]. Code to generate logarithmic abundances can

be found at http://r-sig-ecology.471788.n2.nabble.com/Fit-log-series-to-species-rank-abundance-td5729089.html#a5734514. The idea is that a rank abundance curve is formed by taking all species of organisms in a community and plotting them from lowest to highest abundance. Taking the strains as the species in this approach meant that fitting a log series to the rank abundance curve will give abundances for the 60 strains in any one of our metagenomic samples. Finally, we asked MetaSim to generate NGS samples with 150K reads. This gave ten NGS data sets, each from 60 genomes, with a common reference list of 600 genomes.

Given these ten artificial samples we can apply our method. We show the details for Group 0, but the details for groups 1–9 are similar. The first step in the method is to choose q so as to define the null population. In Sec. 2.3, three different graphs were described; they were the proportion aligning graph, the boxplot chart, and the detection plot. In each case, the idea is to use the knee in the curve as a value for q. To generate the proportion aligning graph and the boxplot chart we used 200 flipped data sets for each $q = 0, .02, .04, \ldots, .3$; we chose 200 as a reasonable value because it was large enough to see the variability but small enough to be computationally inexpensive. The proportion aligning graph and the boxplot chart for Group 0 are shown in Fig. 1.

The detection plot for Group 0 is in Fig. 2. Since each strain has one sequence individual sequences/strains are either detected or not. Hence, there is only one meaningful detection plot on the strain level as opposed to the level of individual chromosomes, plasmids, or contigs. (A detection plot could be generated on the species level but there are different numbers of strains in different species.) In Fig. 2, a significance threshold of .1 was used for the adjusted p-values from the multiple comparisons test procedure. This is an unusual choice necessitated because the adjusted p-values segregated into four ranges:

1. p-value $= 0$
2. p-value $\in (0, .1)$
3. p-value $\in [.1, .99502)$
4. p-value $= .99502$

Note that $200/201 = .99502$, the largest adjusted p-value possible derived from the fact that we generated 200 data sets by nucleotide flipping with probability $q = .16$. In fact, the majority of the adjusted p-values were zero or .9905 because (1) the variance in the number of reads aligning to the reference genomes from the 200 flipped data sets was small and (2) we have a conservative procedure to deal with ties (see Section 2.1). Effectively, the null distributions for the number of reads mapping to a given genome were highly concentrated, forcing the WYM corrected p-values toward either zero or one. Since the region $[.1, .99502)$ had few adjusted p-values in it, as seen in Fig. 3, we chose .1 as the threshold because it was the effective upper bound of the cluster of adjusted p-values that were positive but less than .99502. While unusual, this p-value structure is reflected even more starkly in the real data, see Fig. 9. In the present example there are several possible reasons why there is so little variation in the number of reads aligning. First, the simulated data is simple compared to real data, with fewer reads and fewer references. Second, the tolerance of the aligner may make alignments robust to small changes in sequences. And, third, the reference genomes are very clean and complete leading to less variability because each reference is one 'finished' file that includes plasmids, chromosomes and other contigs. This is unlike the real example

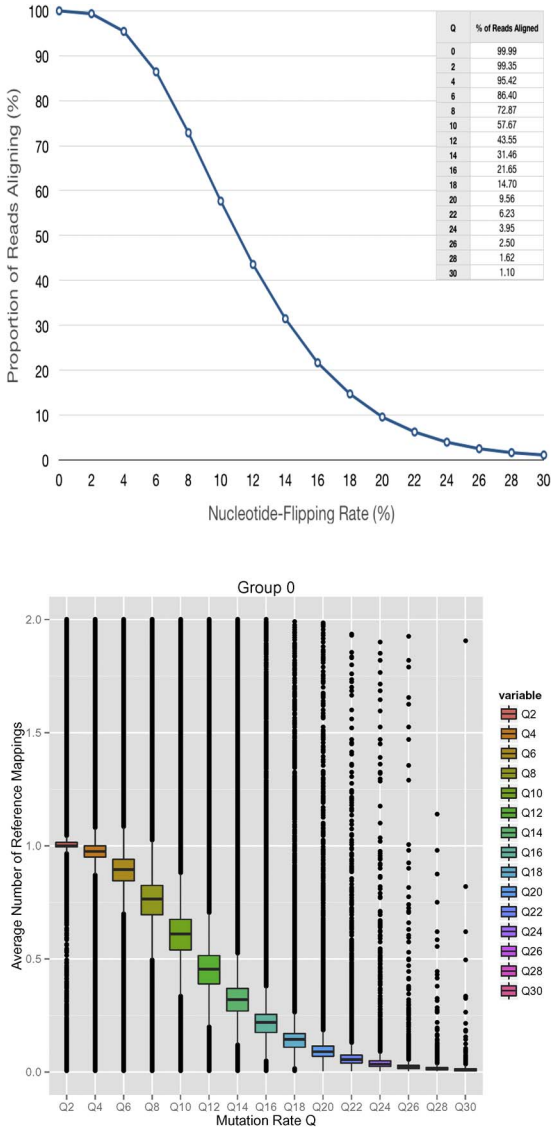

Figure 1. Top: The proportion aligning graph for Group 0 showing the average proportion of reads in the 200 flipped data sets from Group 0 that align to the first 60 reference sequences as a function of q. Bottom: The boxplot chart for Group 0 showing the median number of reference sequences to which each read aligns in the same 200 flipped data sets.

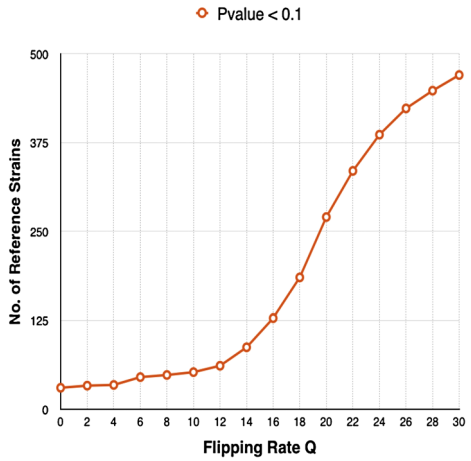

Figure 2. The number of reference sequences detected as a function of q at the strain level using a threshold of .1 for the p-values. Group 0 has 60 genomes but the reference list has 600 genomes of which the 60 from Group zero are a subset.

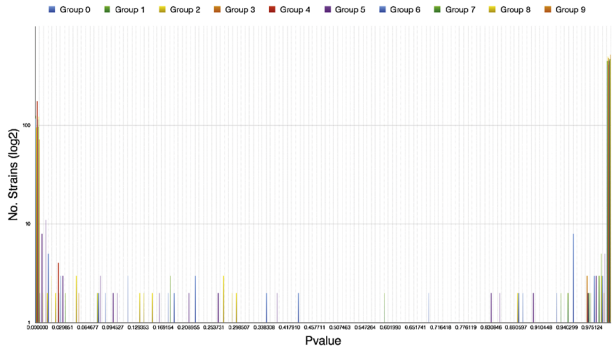

Figure 3. Graph of the WYM adjust p-values for the 10 groups using simulated NGS data. Note that the vertical axis uses the base 2 logarithm of the values. This enables the pattern to be clearly seen.

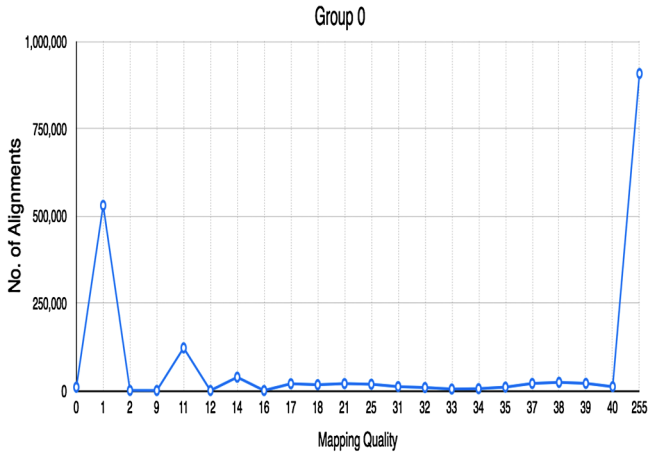

Figure 4. The number of alignments for different mapping quality values for Group 0.

to be presented in Sec. 4 where alignment of a short read to a reference genome is determined by its alignment to at least one of many files that comprise the reference genome.

As a separate issue, we also did not apply the WYM in its pure form. Rather than taking the minimum of the p-values i.e., the maximum of the number of reads aligning, we used the 95-th percentile of the number of reads aligning. The reason is that there were about 5% of genomes who had reads aligning that were unaccountably large. This may arise because some of the genomes are so large or so small that the identicality assumed by the WYM does not hold even approximately. Taking a percentile rather than the raw counts is a simple, robust correction for this. We note that in the real data analysis we also used this sort of correction but the 99.5-th percentile worked well. A smaller percentile was appropriate for the simulated data, perhaps because we analyzed the simulated data using a reference list of 600 strains whereas we analyzed the real data with a reference list of 4,389 strains.

To choose q we proceed as follows. The top panel in Fig. 1 has minimal radius of curvature at approximately .2, but values between .18 and .22 also appear reasonable. The bottom panel in Fig. 1 has minimal radius of curvature at approximately .18, but values from .16 to .2 appear reasonable. Fig. 2 has a minimal radius of curvature at approximately .14 but values between .12 and .16 appear reasonable. Considering the proportion aligning graphs and boxplot charts for Groups 2 to 10 (not shown) yields similar ranges, but they are mostly shifted slightly lower, say $.16 \pm .02$. Considering the detection plots for Groups 2 to 10 (not shown) yields a similar range but increased variability so the range of reasonable values is larger, say $[.12, .2]$.

Overall, this means if we insist on using the same q for all ten groups then one of the most reasonable choices is $q \approx .16$. Requiring a common q makes sense because randomization should ensure that the ten groups are very similar to each other in aggregate.

An interesting feature of the simulation results is that there are a lot of upper outliers in the boxplot chart in Fig. 1 compared with, say, the corresponding boxplot chart for a real data set in the bottom panel Fig. 7. This is likely due to the range in the number of alignments, with relatively few overall in the simulated data versus the real data, leading to a right skew in the simulated results but not in the real data results.

We observe Fig. 4 which shows the number of alignments as a function of mapping quality (phred score). Each read from the dataset may generate multiple alignments (see Fig. 1 and Fig. 7), and the MAPQ (mapping quality, similar in spirit to a phred score) of each alignment is a measure of how reliably a read maps to its references. In Fig. 4, it is seen that for group 0, most hits are either 1 or 255. It is generally accepted that any alignment with a MAPQ > 30

maps to only one reference. There are roughly 700K alignments below the MAPQ = 30 mark, and about 1 million above it, which includes the MAPQ value of 255.

A MAPQ of 255 is an alignment that maps to a lot of places, and is probably being generated from a read that is very common. This usually happens when a read is aligning to many places or it means that the alignment is unique. A close examination of the alignments with MAPQ 255 showed they are valid. So, what is likely is that there are a lot of very similar sequences, such as plasmids, that yielded very similar reads, and Bowtie2 flagged these alignments with 255 rather than not report them, or assigning them a low MAPQ score. One further possibility is that the process of genome finishing (taking a genome from the draft stage to the complete stage) requires that all the draft config assemblies be merged/folded into one contiguous DNA sequence. The genomic assemblers that glue these contigs together sometimes use a reference sequence of a close relative to "finish the sequence". It could be that these complete genomes share stretches of sequence that were included in the artificial NGS sample output by MetaSim. While these regions may not be large, they could generate reads that map to a lot of places.

In terms of the 150k reads generated for Group 0, we generated 1,795,305 alignments of varying degree of mapping qualities: from unique to multiple. Of these 1.8 million alignments, roughly 114k (6.4%) had unique alignments in the sense that one read mapped to one reference. However, there were a substantial number of multiple alignments: 1,681,147 (93.6%). This means that the original sample had about 35,842 reads (24%) that were very similar and mapped to a lot of places. It is these 36K reads that are causing the high number of outliers in Fig. 1. Some of these 36K reads came from plasmids, but others came from other strains that contained very similar sequences.

Having chosen a common q for the ten artificial data sets and understood how it arose, it is instructive to examine the resulting analyses. Recall that the goal of this detection problem is to minimize the number of false negatives (FN) while not forcing the number of true positives (TP) and true negatives (TN) to be too small. The effect of this is to control false positives (FP) as well, but less stringently. Since each of the ten artificial data sets has a set of 60 genomes known to be present and 540 known not to be present we can record the effect of the testing for each group in a 2×2 table. Representative examples of these are for Groups 0, 6, and 7.

$$\begin{pmatrix} TN & FN \\ FP & TP \end{pmatrix} = \begin{pmatrix} 443 & 26 \\ 97 & 34 \end{pmatrix}, \begin{pmatrix} 454 & 26 \\ 86 & 34 \end{pmatrix}, \begin{pmatrix} 473 & 25 \\ 67 & 35 \end{pmatrix}.$$

The sum down the first column is always 540; the sum down the second column is always 60.

It is desirable to aggregate over the ten analyses of the artificial data sets. Even though the analyses are testing for the presence of strains with different strains present in each case, the fact that the ten detection problems are the result of randomization means they should be comparable. Indeed, all 600 of the strains enter the problem in the same way. So, although hard to interpret outside the setting of the context of 600 strains with the specified abundances, the averages are meaningful within the limits of the problem. With these caveats, if the ten 2×2 matrices are averaged and expressed as percentages the results are

$$\begin{pmatrix} 457.5 & 24.8 \\ 82.5 & 35.2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 76.3\% & 4.1\% \\ 13.8\% & 5.9\% \end{pmatrix}.$$

If we condition on the strain being absent, i.e., look at the first column only, then the conditional probabilities of a TN and FP are, on average, $(84.7\%, 15.3\%)$. If we condition on the strain being present, i.e., look at the second column only, then the conditional probabilities of a TP and FN are, on average, $(56.7\%, 43.3\%)$. So, it is seen that while the procedure discriminates relatively well, 85% vs 15%, when the strain is absent, it discriminates poorly when the strain is present, doing about 10% better than tossing a fair coin. Moreover, these averaged results are broadly representative of the results in the ten individual cases.

The above calculations can be carried out at the species level as well. The average of the ten 2×2 matrices and its percentage form are

$$\begin{pmatrix} 309.9 & 21.5 \\ 37.9 & 32.6 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 76.9\% & 5.3\% \\ 9.4\% & 8.1\% \end{pmatrix}.$$

If we condition on the species being absent i.e., look at the first column only, then the conditional probabilities of a TN and FP are, on average, $(89\%, 11\%)$. If we condition on the species being present, i.e., look at the second column only, then the conditional probabilities of a TP and FN are, on average, $(59.7\%, 39.2\%)$. Under both conditioning events, this is a slight improvement over the strain level. However, the discrimination when a species is absent remains much better than when a species is present.

In addition we examined the relationships between accuracy of our method and the lengths or abundances of the strains present; see Figures 5 and 6. The length and abundance classes were chosen so each class contains the same number of genomes, i.e., 150 strains each, and an adequate number of genomes to assess both true positive and true negative rates. The corresponding ranges of lengths and abundances for each of the four classes are provided in the legends of Figures 5 and 6. By length, we get the best tradeoff between true positive and true negatives with the longer genomes, with true positive rates generally rising with length. It is clear from Figure 5 that there is a tradeoff between true positives and negatives, since there is no class for which both rates are at their highest. For the longest genomes our median true positive and true negative rates are 69.7% and 78.9%, respectively. By abundance, we see very different patterns of dependence, with true negative rates relatively insensitive to abundance (constant across

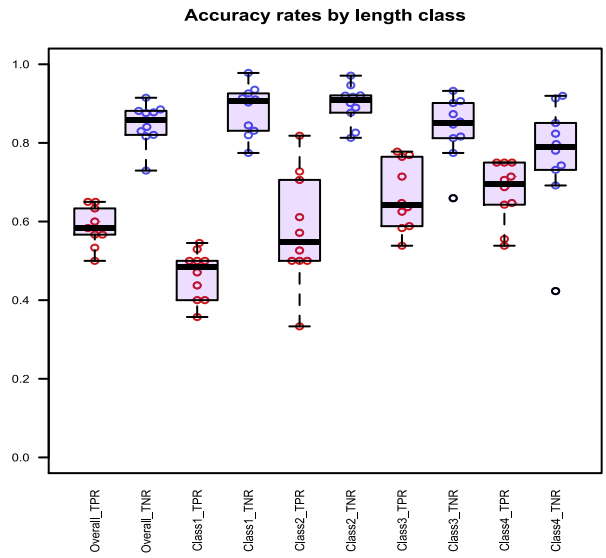

Figure 5. Boxplots by length. Length classes (in base pairs (bp)) are [1,815 – 1,660,425], [1,667,163 – 2,695,903], [2,698,137 – 4,432,590], and [4,482,059 – 10,080,619]. TPR = true positive rate, TNR = true negative rate.

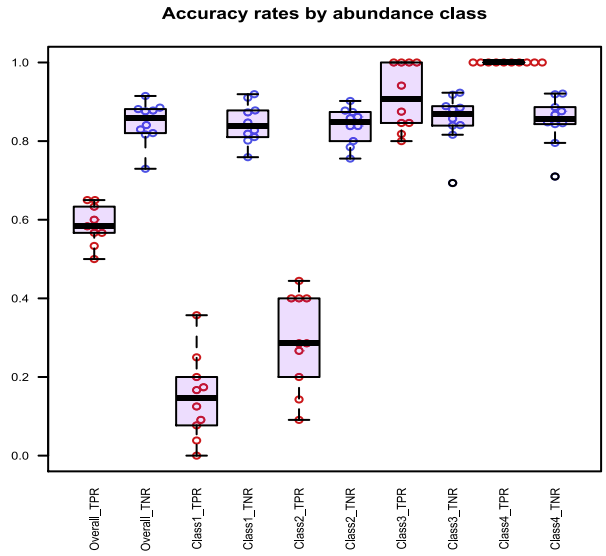

Figure 6. Boxplots by abundance. Abundance classes (in coverage) are [0 – 6], [7 – 90], [91 – 1,223], and [1,236 – 54,666]. TPR = true positive rate, TNR = true negative rate.

abundance classes) but true positive rates rising in direct relation to abundance. In fact, for the two classes of highest abundance (Class 3 and Class 4) we achieve the highest rates of true positives and true negatives, so we do not observe a tradeoff as we did for length classes. For the most abundant strains, our median true positive and true negative rates are 100.00% and 85.8%, respectively. It is not clear why for both length and abundance true negative rates initially drop, and then rise monotonically. One possible explanation is more diversity among the smallest and least abundant genomes relative to the immediately adjacent class, making detection easier at least in a relative sense.

4. ANALYSIS OF AN HMP DATA SET

In the section we apply the methodology discussed in Sec. 2 to determine which bacterial strains are in a metagenomic sample. Since our goal is to demonstrate our methodology we chose a relatively simple data set available from the NIH Human Microbiome Project (HMP), namely, data set SRS015072 ('mid-vaginal') available from http://www.hmpdacc.org/HMASM/. General descriptions of the collection and processing of the physical samples to generate the data are elaborate and are described at http://hmpdacc.org/micro_analysis/microbiome_analyses.php.

All reads aligning to the human genome have been removed; however, the remaining reads may be from a variety of biological sources including bacterial, viral, and/or archaeal. The important feature for this section is the scale of the analysis required. First, there are 495,256 'paired-end' reads in the fastq file. Paired-end means that for each fragment of DNA in the sample the two ends are sequenced leaving a gap of known distance between them that is not sequenced. Since NGS techniques can only sequence relatively short lengths; leaving a gap means one has about twice as many bp's from the fragment and they are of known distance apart. This gives more information per fragment than having one single end read from the fragment. For the SRS015072 dataset, the sample consists of paired end reads, 100 bp reads with an average mate distance of 81 base pairs (bp). To make the computations faster and the results more reliable, we preprocessed the data by filtering out low quality reads defined as having phred scores less than ten.

This data set includes abundance of DNA fragments because it is metagenomic, i.e., the physical sample from which it came includes many distinct genomes and possibly many copies of each of them. For instance, the genus *Lactobacillus* is believed to be in the sample, but the number of the different strains/species or the number from each strain/species present is not known. Also, the reads are not independent: If a read is present from one species of bacterium, it is more probable that there are other reads from that bacterium as opposed to reads from another bacterium. All software is open source and available online (see Appendix) or upon request from the authors.

4.1 Details of analysis: choosing q

To proceed with the analysis, we first used the data to generate $M = 350$ mutated data sets for 15 values of $q = .02, .04, \ldots, .3$. Second, we constructed the reference genome list by downloading 456,865 reference sequences from the IMG database v4.0 [15]. These sequences include

DNA in the form of chromosomes, plasmids, and other genetic scaffolds, representing 4,389 different bacterial strains from 2,286 bacterial species. In particular, when we pool reference sequences (i.e., chromosomes, plasmids, and other genetic scaffolds) to the strain level we say that a strain is present if any of its reference sequences is present and when we pool strains to the species level we say a species is present if any of its strains is present. Since the p-values are found at the reference sequence level, this part of the testing procedure is as sensitive as possible for detection. Third, we used a short read aligner called Bowtie2, [12], to align the reads in the HMP dataset to the reference genomes and to align the flipped data sets to the same reference genomes. Bowtie2 in local alignment mode is more sensitive and more efficient than BLAST for aligning NGS data. In our work we used seeds of length 20; the Appendix gives further computational details.

Fourth, given the Bowtie2 alignment of the flipped data to the references, we turned to finding a value for q. We generated the graph of the average of the proportion of reads from each of the 350 mutated data sets aligning to the reference list at the sequence level as a function of q. This gave us the proportion aligning graph in the top panel in Fig. 7. Note that in contrast to Fig. 1 in which the curve starts from 100%, the curve for the alignment of this real data starts from around 47%. It is seen that the knee in the curve occurs around $q \approx .16$, though any number in [.14, .18] could be argued as reasonable. This inference is seen from a different perspective in the bottom panel of Fig. 7. Here we show a boxplot chart of the average number of reference files to which each flipped read aligns as a function of q. The boxplots confirm the choice of q as the knee in the curve formed by the medians of the boxplots, which occurs around $q \approx .16$.

As a separate computation, we also chose q by generating the detection plots in which the number of reference sequences and strains detected are shown. In both cases, reference sequences and strains, we used the same 350 mutated data sets as in the proportion aligning method. Fig. 8 shows the curves for p-value thresholds 0.05 and 0.01. The two curves very nearly overlap meaning that there are few strains that get assigned p-values between .01 and .05 (see Fig. 9); this suggests that the method is giving relatively decisive answers. Again, we infer $q \approx .16$, with a reasonable range of [.14, .18], consistent with Figure 7. Thus, overall, it seems reasonable to set $q_{opt} = .16$.

Before proceeding to the inferences from the WYM-adjusted p-values, we present a graph of them parallel to Fig. 3. Specifically, Fig. 9 shows how the WYM adjusted p-values concentrate even more strongly at zero and one for this real data set than for the simulated data sets used in Sec. 3. Again, this can be attributed to reasons discussed previously; see Sec. 3. In this example, the even stronger concentration at zero and one of the WYM adjusted p-values may be attributed in part to the relatively large list of reference genomes and the relatively large number of files each of them is permitted to have. As with the simulated data, the robustness of the aligned to small changes will also tend to push WYM adjusted p-values to zero or one. Finally, the concentration at zero and one may also be the result of the fact that we took the minimum p-value over all the reference files for each strain. Since a strain is detected if any of its files is detected and not detected if none of its files are detected, there is a tendency to decisiveness.

For completeness we examine the distribution of alignments over mapping quality values; this is seen in Fig. 10.

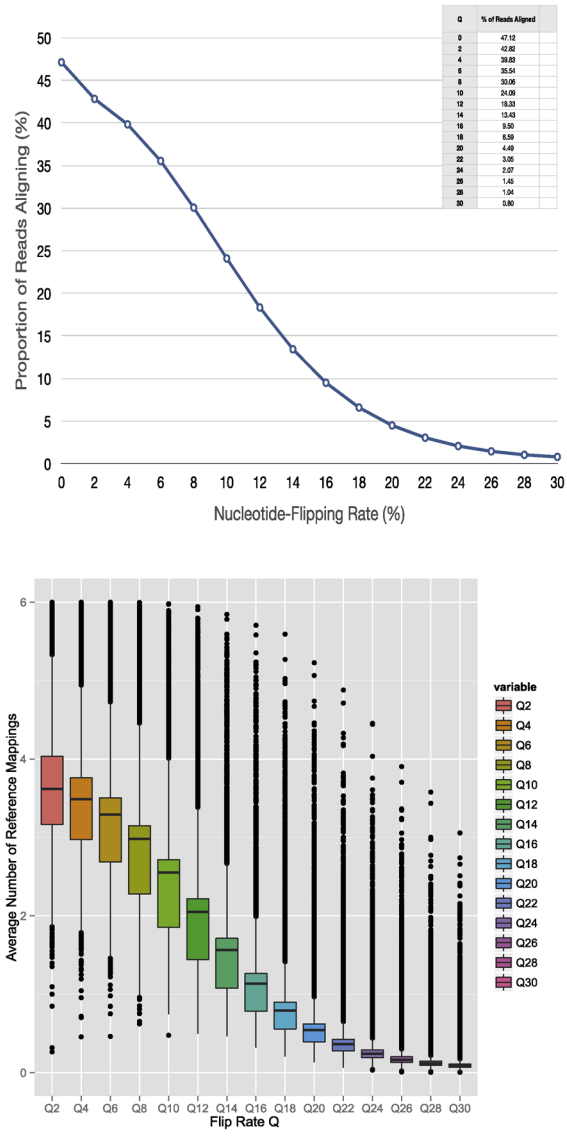

Figure 7. Top: The proportion aligning graph showing the average proportion of reads in the 350 flipped data sets that align to the reference sequences on the sequence level as a function of q. Bottom: The boxplot chart showing the median number of reference sequences to which each read aligns in the same 350 flipped data sets.

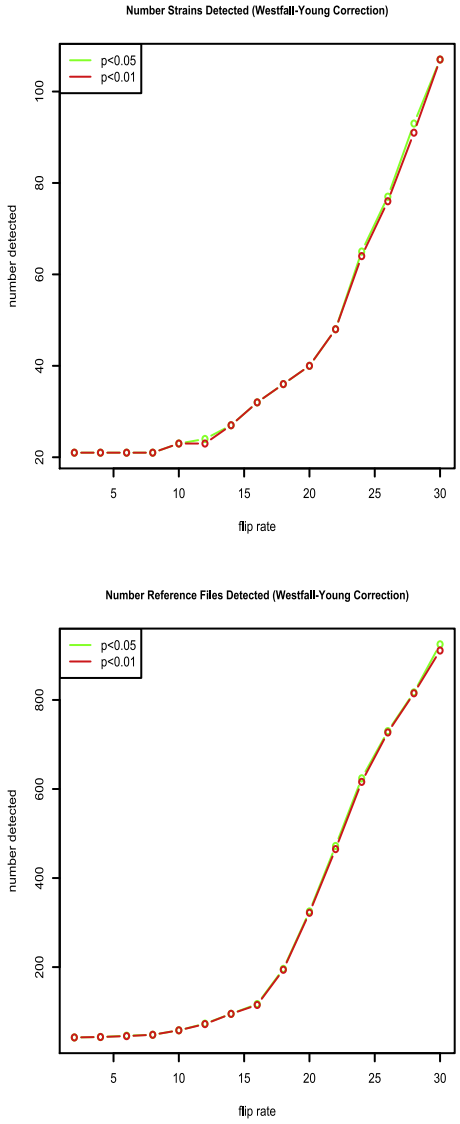

Figure 8. Detection plots showing the number of reference sequences detected as a function of q at the strain (top) and sequence level (bottom) using thresholds .05 (green) and .01 (red) for the p-values. (Color figure online)

Note that in contrast to Fig. 4 the distribution has mass over the region between one and the maximum value of 44, even though there is still a large number of poor alignments as indicated by the spike at one. This means that there are many weak, medium, and strong alignments so that the selection of significance threshold for the p-values behaves more typically as seen in Fig. 8. Also, possibly due to the size of the reference list, the quantile used instead of the minimum over p-values in the WYM (or here, the maximal number of reads per genome) is .995, i.e., just less than the minimum (or maximum number of reads aligning). The extra trimming required in the simulated data example is not necessary.

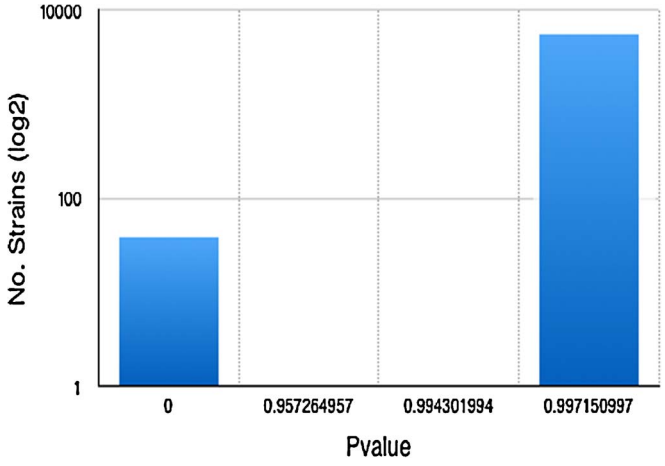

Figure 9. Graph of the WYM adjust p-values for the HMP data. Note that the vertical axis uses the base 2 logarithm of the values. This enables the pattern to be clearly seen. Although the two middle bars appear to be zero, in fact they are small positive numbers that are not visible within the resolution of the figure.

4.2 Details of analysis: running time

The method begins by mapping the NGS dataset to the IMG bacterial reference set using the Bowtie2 [12] aligner. At the time of analysis the Bowtie2 reference sequence indexer (bowtie2-build) did not have support for large genome lists, in particular those created from reference databases that contained more than 4 billion characters because its implementation did not support 64-bit integer numbers. The IMG 4.0 reference at the time contained about 16 billion nucleotide characters so using a large monolithic index was not possible. It should be noted that the latest version of Bowtie2 does support large indices but this would have no impact on the results presented here.

We resolved this limitation of Bowtie2 by splitting the reference database into 6 shards, each consisting of about 3 billion characters. For performance and scaling matters we left a little bit of room and did not set each index shard to be the full 4 billion characters. For each permutation, Bowtie2 was run against each index shard independently and its results merged and processed using SamTools [13].

At low q values (0.0, 0.02, 0.04, etc.), Bowtie2 had varying running times in each permutation depending on the complexity (length of references, amount of repeats, flipping rate) of the permutation: it took as little as 3 minutes for low complexity shards, and as high as 14 minutes for high complexity shards. For larger q values (0.10, 0.12, 0.14, etc.), Bowtie2 took slightly less at the low complexity shards (about 2 minutes) and about half the time for high complexity shards (8 minutes). At $q = 0.20$, Bowtie2 was taking about 2 minutes for low complexity shards, and about 3 minutes for high complexity shards. At $q = 0.30$,

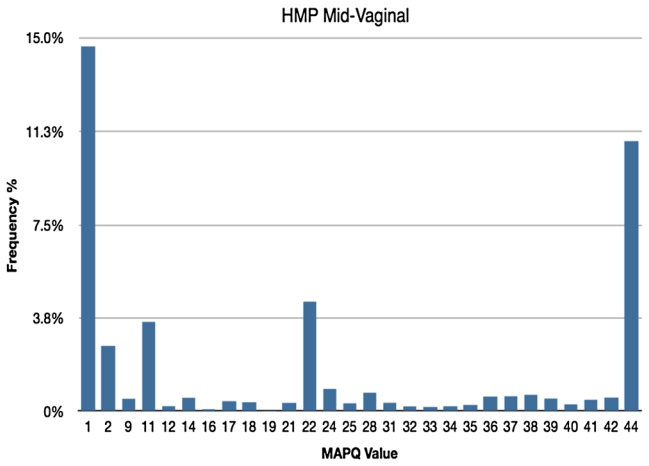

Figure 10. The number of alignments for different mapping quality values for the HMP data.

Bowtie2 took about 1.5 minutes in low complexity shards, and about 2 minutes for high complexity ones.

The overall run time depends on several input parameters such as the size of the reference database, the size of the sample data, the number of candidate flip rates, the number of compute cores available, and the number of flipped datasets. In our example, with a very large reference database and many flipped datasets, overall running time for 350 flipped data sets at each flip rate q (15 different values) was about 6 days using the Pegasus 2 compute cluster at the University of Miami. Pegasus 2 is an IBM iDataPlex cluster with Intel SandyBridge processors that support 16 cores and 32 GB of memory per compute node. The data footprint after alignment and post-processing is about 3.8 terabytes (TB). Clearly this run time can be modified by changing the input parameters, making the method more or less feasible with fixed resources.

4.3 Details of analysis: results

The p-values assigned by our method to various strains that might be represented in the data set SRS015072 using $q = .16$ are shown in Tables 1 and 2. We contrast our findings with what the HMP identified as bacterial strains present based on genome coverage. The HMP only reports strains with at least $0.01 \times$ depth across 1% of the strain's reference genome. There is no information provided regarding significance of detection or uncertainty. Despite this, their results can be taken as baseline for comparison.

The first three columns in Table 1 show the strains reported by the HMP for the SRS015072 data set with their depth and breadth of coverage. Following HMP, the strains are ordered by decreasing depth. This means that, in principle, the strains at the top of the table should be easier to find than the strains at the bottom. The fourth column in Table 1 shows the adjusted p-values our method assigns to these strains. Strains in red text are considered 'significant' at $p < 0.01$. It is seen in Table 1 that our method assigns 'significant' p-values to some, but not all, of the HMP findings. They detected 29 strains of which 8 are assigned adjusted p-values < 0.01, meaning they would be detected by our method at this alpha level. We note at lower coverage (less depth and breadth) we find fewer strains significant, but that our p-values do not increase monotonically as coverage drops.

Table 2 shows the 39 strains detected by our method; strains in red text are also included in the HMP results. If a species is listed (not a strain) then this species has no strains that have been sequenced, i.e., the species level is effectively the strain level. We see that our method detects strains/species that the HMP does not, but that there are overlaps in the results (8/29 strains; 4/16 species).

We verified that these results are independent of small changes ($\pm.02$) in the value of q.

Table 1. HMP results and WYM-adjusted p-values. Depth and breadth statistics were provided by the HMP. (See http://downloads.hmpdacc.org/data/HMSCP/SRS015072/SRS015072_abundance_table.tsv.bz2). Strains are ordered by decreasing Depth. Strains in red have p-values < 0.01

Bacterial Strain (HMP)	Depth (HMP)	Breadth (HMP)	p-value
Lact. iners AB-1	5.059	90.823	9.97E-01
Lact. iners DSM 13335	5.020	90.917	0
Lact. crispatus 214-1	4.491	81.364	0
Lact. crispatus MV-3A-US	4.321	76.964	0
Lact. crispatus MV-1A-US	3.838	72.789	0
Lact. crispatus JV-V01	3.478	73.010	0
Lact. crispatus 125-2-CHN	3.454	68.603	0
Lact. crispatus ST1	2.840	52.189	9.97E-01
Gard. vaginalis AMD	0.295	22.717	9.97E-01
Pseud. fluorescens SBW25	0.160	12.131	9.97E-01
Lact. vaginalis ATCC 49540	0.129	10.728	9.97E-01
Gard. vaginalis ATCC 14019	0.100	8.437	9.97E-01
Pseud. sp. UK4	0.078	6.331	9.97E-01
Lact. jensenii SJ-7A-US	0.077	3.779	9.97E-01
UPS 6 str. ATCC 27818	0.0620	5.652	9.97E-01
Lact. helveticus DSM 20075	0.052	1.030	0
Lact. jensenii 1153	0.051	4.036	9.97E-01
Gard. vaginalis 409-05	0.05	3.984	9.97E-01
Gard. vaginalis 5-1	0.047	3.923	9.97E-01
Lact. jensenii 269-3	0.046	3,737	9.97E-01
Lact. jensenii 208-1	0.044	2.283	0
Lact. col. 101-4-CHN	0.041	2.818	9.97E-01
Myco. abscessus	0.040	3.560	9.97E-01
Herbas. seropedicae SmR1	0.03	2.839	9.97E-01
Pseud. fluorescens Pf-5	0.02	1.857	9.97E-01
Pseud. fluorescens Pf0-1	0.02	1.848	9.97E-01
Sphing. alaskensis RB2256	0.020	1.992	9.97E-01
Prev. bivia JCVIHMP010	0.013	1.109	9.97E-01
Sten. maltophilia K279a	0.01	1.210	9.97E-01

Table 2. Significant strains using WYM-adjusted p-values < 0.01. Strains in red are also on the HMP list

Bacterial Strain	p
Acidithiobacillus sp. GGI-221	0
Acine. Sp. 6014059	0
Actino. sp. oral taxon 171 str. F0337	0
Burk. mallei PRL-20	0
Clost. bacterium 1_7_47_FAA	0
Endoriftia persephone Hot96_1+Hot96_2	0
E. coli MS 69-1	0
Flav. frigoris PS1	0
Fuso. ulcerans ATCC 49185	0
H. pylori	0
Lact. crispatus 125-2-CHN	0
Lact. crispatus CTV-05	0
Lact. crispatus MV-1A-US	0
Lact. gasseri JV-V03	0
Lact. iners DSM 13335	0
Lact. jensenii 27-2-CHN	0
Lact. sp. 7_1_47FAA	0
Mollicutes bacterium D7	0
Strep. sp. str. LCC	0
Therm. cellulosilytica TB100	0
Acine. Sp. 6013113	0
Actino. pleuropneumoniae str. 4074 Aple01_131	0
Beggiatoa sp. PS	0
Cand. Pelagibacter ubique HTCC1002	0
Crenothrix polyspora	0
Entero. faecium TX1330	0
E.coli MS21-1	0
Flavonifractor plautii ATCC 29863	0
Haemophilus parasuis	0
Lact. amylolyticus DSM	0
Lact. crispatus 214-1	0
Lact. crispatus JV-V01	0
Lact. crispatus MV-3A-US	0
Lact. helveticus DSM 20075	0
Lact. jensenii 208-1	0
Lact. malefermentans KCTC 3548	0
Lact. ultunensis DSM 16047	0
Pectobacterium carotovorum subsp. brasiliensis PBR1692	0
Strep. viridosporus T7A	0

Note that in Table 1 the best agreement between our method and the HMP method is when both depth and breadth are high (low p-value), and the worst agreement are when both depth and breadth are low (high p-values). That is, the confidence implicitly expressed by coverage in the HMP findings is broadly consistent with our findings. The discrepancies that remain have several plausible explanations. For example, there are differences in the data processing pipelines, e.g., the HMP filtered out low complexity reads and used a different alignment algorithm (see the Protocols at http://www.hmpdacc.org/HMASM/ for complete details [11]). The reference genome database used by the HMP for read mapping comprised of all archaeal, bacterial, lower eukaryote and viral organisms available in GenBank as of 11/2009. They further processed their bacterial references, removing highly redundant, non HMP-sequenced reference genomes. In contrast we used the entire IMG 4.0 bacterial database. As a result their reference database was broader than ours – by including non-bacterial genomes – but also more restricted than ours – by post-processing and limiting the bacterial reference genomes. We chose not to use the HMP reference set, or further replicate their results, because our focus is on strain detection using the most recent reference bacterial genome catalog. Nevertheless, a comparison of our results with theirs is interesting because it suggests that their true positive rate (i.e., the proportion of strains they consider 'present' which are actually present) will be high – but at the cost of a higher false negative rate. By contrast our method likely has a lower true positive rate, but a lower false negative rate. That is, if the HMP says a strain is present it is likely to be present whereas if we say a strain is not present it is likely not to be present.

5. DISCUSSION

We have developed a detection technique for bacteria on the strain level using metagenomic NGS data. Our method rests on using artificial point mutations or nucleotide flipping to generate a reasonable null population. This requires choosing an appropriate flipping rate as a tuning parameter; we have suggested several ways to do this. We performed a detailed simulation study examining the accuracy of our method in terms of both true positive and true negative rates of strain detection. Our approach leans toward controlling the rate of false positives over controlling the rate of false negatives. We have applied this method to data from the Human Microbiome Project and compared our results to theirs. The HMP findings were based on read coverage (depth and breadth) whereas ours were based on p-values, giving an assessment of significance and uncertainty as well as detection. As noted previously [24] and [28] have done related work using marker genes for species identification. By contrast our approach is purely statistical; it uses the whole genome nucleotide information for bacterial detection at the strain level. As a generality, marker genes at the strain level are not available so the [28] approach, for instance, cannot yet be applied.

The results from the two approaches (ours and HMP) have some overlap, particularly for strains with higher sequencing depth, although overall our findings are meaningfully different from those of HMP – strains they identify as present we do not. We explain this mainly by differences in the databases and tools used by the two approaches as well as the intrinsic variability in the data. The major differences in our pipeline versus the HMP pipeline include our use of a different reference set, different aligner, and different pre- and post-processing of reads. As a consequence, for instance, if there is considerable genomic similarity among

strains of the same species, different pipelines can detect different strains.

To investigate the general performance of our method under idealized circumstances we did a relatively extensive simulation study. In this simulation the reference genomes were 'clean' and the NGS data generated by Metasim was 'clean'. This is seen for instance in the fact that the proportion aligning in Fig. 1 was 100% for small q while it was only 47% for the real data in Fig. 7. Also, the reference genomes for the simulation were single 'finished' files rather than collections of files. The simulation study revealed that, roughly, if our method says a reference genome is not present, then very likely it is not present, i.e., the probability of a false positive is low. On the other hand, the method seems to sacrifice discrimination when a reference genome is present. That is, the probability of a false negative is unfortunately high. We attribute this to the tradeoff between false negatives and false positives implicit in the selection of q. While the selection of q implicitly takes false positives and false negatives into account, it was not clear a priori how the two sorts of errors would be weighted. That the probability of true negatives is generally the largest of the four probabilities merely means that our method is emphasizing specificity: Statements from our method of the form that a given genome is not present are more reliable than statement to the form that a given genome is present.

An interesting feature of the simulations that mimics the real data analysis is the similarity in the distributions of the adjusted p-values. This appears to be the result of the high level of stability of the number of reads aligning (for given q) over the flipped data sets. The implication of this is that the distribution of adjusted p-values tends to concentrate at either zero or one as seen in Fig. 3 and Fig. 9. The two solid bars in Fig. 9 are more pronounced than the concentration seen at the endpoints in Fig. 3 because a shorter, cleaner reference list was used with the simulated data.

Another way to look at the WYM adjusted p-values is to recall that, aside from the WYM adjustment, they are found relative to the null constructed by choosing q and using the flipping procedure. Hence, in a fundamental sense, all the downstream inferences depend crucially on the construction of the null. The consequence is that the probability of Type II error is inflated, i.e., the probability of failing to reject the null hypothesis that C_j is not in the population when it is false seems to be generally larger than one would want it to be. On the other hand, the probability of rejecting the null when it's true seems relatively low. This is reasonable because when searching for pathogens initially we mostly want to rule out possibilities that we are sure should be ruled out, i.e., we want high probability of true negatives. These two criteria – high true negative rate and high true positive rate – conflict in our set up because doing better with one means doing worse with the other. It is unclear how to improve on this tradeoff in a generic procedure although using more strain-specific biological information could improve performance for known strains.

There are obvious ways to refine our approach. First, we could remove strains that are not compatible with human samples and strains that have not been adequately sequenced, and add non-bacterial genomes such as viruses. Second, we have not used some information on the read and genome level. Some reads and genomic locations will be relatively unique to a given reference and very informative (see [29]), while other reads will map to many strains and species and other locations will be similar among strains and hence not be very informative. These reads and locations should be weighted differently. Our use of fractional reads is a crude way to deal with the non-uniqueness but undoubtedly better techniques will become available as read uniqueness is understood. Third, there may be higher level dependencies that exist between bacterial strains and phages. The presence of phages, like plasmids (which we include), are important in terms of biology and may aid in strain identification. Fourth, if the alignments were perturbed rather than the nucleotides or, more precisely, the perturbation of the nucleotides could be done in such a way as to have a larger effect on the variability of the number of reads aligning to each genome, then the concentration of WYM adjusted p-values at zero and one might be mitigated thereby giving better performance. Finally, we emphasize that our method applies to a single sample but the general approach could extend to multiple samples or groups of samples.

APPENDIX

5.1 Bacterial reference sequences

456,865 genomic bacterial reference sequences, in FASTA format, were procured from the Integrated Microbial Genomes and Metagenomes (IMG, version 4.0) database ([15]). The 456,865 reference sequences accounted for 5,168 bacterial genomes (at strain level) which included sequences from bacterial genomes and bacterial plasmids. The 5,168 genomic references were isolated by relying on bacterial taxon names and identifiers obtained from the Genome Browser at the IMG website (http://img.jgi.doe.gov/cgi-bin/w/main.cgi?section=TaxonList&page=taxonListAlpha&domain=Bacteria).

5.2 Metagenomic sample

A human metagenomic sample was obtained from the Human Microbiome Project ([7]). Human metagenome sample SRS015072, obtained from a female participant of the HMP Core Microbiome Sampling Protocol A (HMP-A) dbGaP study, was downloaded from the HMP FTP site. The sample consisted of 495,256 paired-end, 100 bp reads (with an average mate-distance of 81 bp) in illumina FASTQ format (Fig. 11).

5.3 Reference sequence alignment

The HMP dataset (SRS015072) was aligned to the 456,865 bacterial references using the Bowtie2 ([12]) aligner.

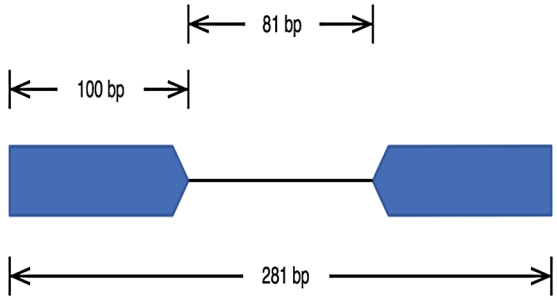

Figure 11. Paired-end reads depiction for the HMP SRS015072 sample. Each mate in the fragment is 100 bp long. There is an average or 81 bases from the end of the left-mate to the start of the right-mate. On average, the length of a fragment in the dataset was 281 bp.

Bowtie2 requires that the reference sequences be indexed so that the reads can be efficiently aligned.

The bacterial genomic references were prepared for alignment using the bowtie2-build indexer program. The indexer program was run with default values along with the -f flag (fasta sequences). The Bowtie2 indexer program uses 32-bit pointers to create the index that is used by the Bowtie2 alignment program. The 32-bit limitation creates a cap on the number of characters (A,T,G,C) in a given index (about 232-1 characters are possible), but in practice the indices created by Bowtie2 are smaller.

The bacterial references needed to be split into indices that contained less that 4 billion characters each because of the relatively large size of the single bacterial index (over 16 GB on disk). A custom PERL script was employed to recursively split the index in half until the required index size was achieved (Fig. 12). In the end, the 456,865 bacterial references were split into 6 sub-indices, each in Bowtie .bt2 format.

Once the 6 indices were created, the Bowtie2 alignment program was used to independently align the HMP metagenomic sample (SRS015072) to each of the 6 indices. Bowtie2, used in local-alignment mode, was used to align the HMP data using the following command:

```
bowtie2 --local -D 20 -R 3 -N 0 -L 20 i S,1,0.50 --time f x S
```

Samtools (0.1.18) ([13]) was then used to merge (samtools merge) the results from each of the 6 indices into one results file in standard .SAM format.

5.4 Flipped runs

Three hundred and fifty (350) flipped data sets were created for each value of the flip rate q $(.02, .04, .06, \ldots, .30)$. Each flipped run consisted of the following steps:

1. Flip the reads using a given value of q.
2. Align the flipped reads to the 6 indices using Bowtie2.
3. Post-process the alignment results using Samtools.
4. Convert SAM format to compressed BAM.

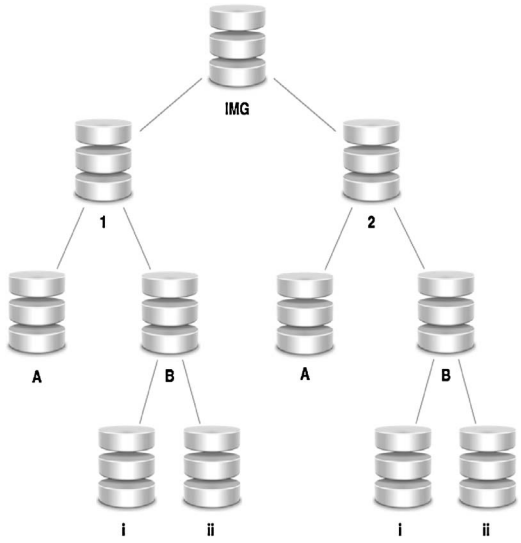

Figure 12. The IMG database had to be split into sub-indices that met the requirements of the bowtie2-build indexer software. 6 indices were created: 1.A, 1.B.i, 1.B.ii, 2.A, 2.B.i, 2.B.ii.

5. Re-head each BAM file using samtools reheader command.
6. Merge the resulting BAM files using samtools merge command.
7. Sort the BAM file using samtools sort command.
8. Index the BAM file using samtools index command.

Flipping the reads is accomplished using a custom PERL script. The software takes as input one file and one parameter: the input file is a set of reads in FASTQ format; the input parameter is the target permutation rate (q) at which the sample reads will be mutated/flipped. The software mutates the sample reads by walking along the length of each mate read, randomly drawing from a uniform distribution, and comparing the result of the draw to the target flip rate. If the result is less than the flip rate, then the nucleotide is mutated. Nucleotides are randomly mutated with other (non-self) nucleotides by randomly drawing from the remaining three nucleotides, and Chargaff's base-pairing rules are not observed in the mutation step. Once all the reads have been flipped, they are outputted to disk in FASTQ format by the software, and used in the remaining steps of the flipped run. Independent flipping steps were carried out for each mate in the paired-end fragment.

After the 350 flipped runs are completed, their results are collated using a custom PERL reporting script that takes as input two (2) files and one (1) parameter: the first file is a list of the references in the indices; the second file is a FASTA-formatted file containing the reference sequences; the parameter is the directory path in which the flipped data sets are stored. The first file is used to vet the alignment results and identify which references had reads aligned

to them; the second file is used to gather the length of the reference sequences. The reporting scripts summarize the fractional read count mappings (see below) for each reference across the flipped data sets. These fractional counts are used in the p-value calculations.

The mapping count for a given bacterial strain is calculated by taking into account the number of genomic references a read fragment maps to. The contribution of each read fragment r_k to the count of the strain is calculated as $1/g_k$ where g_k is the number of references to which read r_k aligns. If a given read fragment maps to 10 reference sequences in the reference database, then the contribution of that read to each of the references it mapped to is 0.1. These fractional counts are collected for each reference sequence, and in turn these fractional counts are aggregated into the total fractional count of the parent strain, the strain that the reference sequence belong to.

Not all reads that map to a reference sequence are used in the fractional count calculation. Reads with many mappings (high counts of reference database hits) are filtered out based on their mapping quality (MAPQ) values. A custom PERL script is used to filter out low MAPQ reads (<MAPQ of 10) and parse the alignment mappings to generate the fractional counts. These fractional counts are then used to compute the p-values.

5.5 Metagenomic simulations

Simulated metagenomic datasets were used to gauge the method. The MetaSim [25] simulator was employed to create all simulated metagenomic samples, and all samples were created using the graphical version of MetaSim, version 0.9.1, running on OS X Mavericks, 10.9.5.

Reference DNA sequences for the simulated datasets were downloaded from NCBI Genbank's Bacteria genome repository (ftp://ftp.ncbi.nih.gov//genbank/genomes/Bacteria). The Bacteria set in GenBank contains high quality complete genomes that could be used in the simulations without the worry of unassembled, incomplete 'draft-quality' genomes affecting the simulations. GenBank's bacterial genome repository contains 2,772 genome directories, representing one strain per directory. All 2,772 genomes were downloaded, and the whole genome set contained 5,179 sequences (chromosomes, plasmids) that were used in creating the simulated metagenomic datasets. Note that the 2,772 bacterial strains represented 1,484 species, and 665 genera classes. The total data footprint for the GenBank genomic sequences was about 9.72 Gigabytes (GB) on disk.

The simulation procedure consisted of four (4) steps:

1. Selection of which strains of bacterial genomes to use in the simulations.
2. Creation of taxon profiles that specify strain abundance values.
3. Execution and processing of the synthetic samples.
4. Assessment of results & benchmarking

A conditional population of 600 strains was randomly selected from the 2,772 strains using a Python script that employed the 'random.choice()' function from the 'random' library from the Python Standard Library. Once these 600 strains had been selected, their sequences (in FASTA format) were imported into MetaSim's internal genome database. The same Python script that selected the 600 random strains for the conditional population also split them into ten groups so each of the 600 strains was placed randomly into one of the 10 groups. The goal in doing this was to create ten separate metagenomic samples with a known composition of 60 strains each. As before, the 'random.choice()' function was used to randomly select the 60 strains for each group, and the conditional population list was shuffled, using Python's 'random.shuffle()', before each group of 60 was selected.

MetaSim requires a 'Taxon Profile' that specifies the composition of the metagenomic sample to be simulated. The profile defines the source genomes for creating the synthetic NGS reads, along with their abundance values (number of genome copies). The profile is text-based and tab-delimited, with one line per genome, and uses the .mprf file-system extension. A genome in the profile can be defined by either its taxonomic name, or by its GenBank Identifier (GI) number. In order to mitigate errors when matching FASTA sequences to taxonomic names in the database, the taxon profiles for MetaSim contain GI numbers. For each of the ten groups of 60 strains, one MetaSim taxon profile was created that contained the strain's GI number and an abundance value. The abundance values, defined by MetaSim as the number of genome copies, was defined by an R script that uses the **untb** [8] and **vegan** [22] libraries obtained from CRAN, and snippets that have been incorporated into the R package sads [21]. For each of the ten groups, the 'fisher.ecosystem()' function from **untb** was used to create a Fisherian ecosystem of size 150,000 the desired number of NGS reads per group. Each of the ten taxon profiles was then used in MetaSim to create ten synthetic metagenomic samples from the conditional population now residing in the MetaSim internal genomic database. Each sample consisted of 150,000, 100-nucleotide long, paired-end NGS reads in FASTA format.

The genomic DNA sequences of the 600 strains in the conditional population were uploaded to the cluster to be used as the reference database in the alignment step. They were indexed by Bowtie2 (using the 'bowtie2-build' indexer) and because the combined size of the 600 strain nucleotide sequences was about 1.83 GB, there was no need to split the reference database index in shards like with the IMG database. Each of the ten groups was then processed in the same way as the HMP dataset, with the exception that in the simulated runs only 200 permutations were created for each value of the flipping rate q. The same scripts used in the flipping and post-processing of the HMP dataset were employed in processing of the simulated datasets.

After a value of q had been selected and pvalues calculated for each strain, a confusion matrix was created to assess true-negative, false-negative, false-positive, and true-positive rates. An R script was used to get these rates which utilized the 'confusion-matrix()' function from the **SDMTools** CRAN package [33].

5.6 Software availability

All software can be obtained at the Center for Computational Science's Github repository, available at the following URL: http://camilo-v.github.io/Metagenomics.

Available are all the scripts (python, perl, shell, R) for the nucleotide flipping, permutations, counts, pvalues, and visualizations. All source code will be released under the GPL v.3 license.

ACKNOWLEDGEMENTS

We would like to thank Anthony Fodor, Ph.D., for guidance through the labyrinth of microbiology and metagenomics. The authors were supported by NSF-DTRA grant #NR66853W (DMS-1120404).

Received 22 April 2014

REFERENCES

[1] Auer, P. and Doerge, R. (2010) Statistical design and analysis of RNA sequencing data. *Genetics*, **185**, 405–416.

[2] Clarke, J., Clarke, B., and Seo, P. (2010) Statistical expression deconvolution from mixed tissue samples. *Bioinformatics*, **26**, 1043–9.

[3] Clarke, B. and Clarke, J. (2014) Estimating the proportions in a mixed sample using transcriptomics. *STAT*, **3**, 313–325.

[4] Clarke, B., Valdes, C., Dobra, A., and Clarke, J. (2014) A Bayes testing approach to metagenomic profiling in bacteria. *Stat. and Its Interface*, **8**, 173–185.

[5] Datta, S., Datta, S., Kim, S., Chakraborty, S., and Gill, R. (2010) Statistical analyses of next generation sequence data: A partial overview. *J. Proteomics Bioinform.*, **3**, 183–190.

[6] Ge, Y., Dudoit, S., and Speed, T. (2003) Resampling based multiple testing for microarray data analysis. Department of Statistics, UC Berkeley, TR #633. MR1993286

[7] Gevers, D., Knight, R., Petrosino, J., Huang, K., McGuire, A., Birren, B., Nelson, K., White, O., and Methe, B., and Huttenhower, C. (2012) The Human Microbiome Project: a community resource for the healthy human microbiome. *PLoS Biol.*, 10:e1001377.

[8] Hankin, R. K. S. (2007) Introducing unto, an R Package for simulating ecological drift under the unified neutral theory of biodiversity. *J. of Stat. Software*, **22**, 12.

[9] Horta, F., Caldera, R., Cortes, R., Carballido, J., and Alpuche, E. (2011) Mathematical model for the optimal utilization percentile in M/M/1 systems. http://arxiv.org/ftp/arxiv/papers/1106/1106.2380.pdf

[10] Human Microbiome Project Consortium. (2012a) Structure, function and diversity of the healthy human microbiome. *Nature*, **486**, 207–214.

[11] Human Microbiome Project Consortium. (2012b) A framework for human microbiome research. *Nature*, **486**, 215–221.

[12] Langmead, B. and Salzberg, S. (2012) Fast gapped-read alignment with Bowtie 2. *Nature Methods*, **9**, 357–359.

[13] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., and Abecasis, G.,and Durbin, R. (2009) 1000 genome project data processing subgroup: the sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

[14] Magi, A., Benelli, M., Gozzini, A., Girolami, F., Torricelli, F., and Brandi, M. (2010) Bioinformatics for NGS data. *Genes*, **1**, 294–307.

[15] Markowitz, V., Chen, I., Palaniappan, K., Chu, K., Szeta, E., Grechkin, Y., Ratner, A., Jacob, B., Huang, J., Williams, P., Hunteman, M., Anderson, I., Mavromatis, K., Ivanova, N., and Kyrpides, N. (2012) IMG: the integrated microbial genomes database and comparative analysis system. *Nucleic Acids*, **40**, 115–122.

[16] May, R. (1975) Patterns of species abundance and diversity. In: *Ecology and Evolution of Communities*, Cody, M. and and Diamond, J. Eds., 81–120. The Belknap Press of Harvard University Press, Cambridge.

[17] Meinshausen, N., Maathuis, M., and Bühlmann, P. (2011) Asymptotic optimality of the Westfall-Young permutation procedure for multiple testing under dependence. *Ann. Statist.*, **39**, 3369–3391. MR3012412

[18] Metzker, M. (2010) Sequencing technologies – the next generation. *Nature Reviews Genetics*, **11**, 31–46.

[19] Nagarajan, N. and Pop, M. (2012) Sequence assembly demystified. *Nature Reviews Genetics*, **14**, 157–167.

[20] Neilsen, R., Paul, J., Albrechtson, A., and Song, Y. (2011) Genotype and SNP calling from next-generation sequencing data. *Nature Reviews Genetics*, **12**, 443–451.

[21] Prado, P. and Miranda, M. (2014) sads: Maximum Likelihood Models for Species Abundance Distributions. See: http://cran.r-project.org/web/packages/sads/.

[22] Oksanen, J., Blanchet, F., Kindt, R., Legendre, P., Minchin, P., O'Hara, R., Simpson, G., Solymos, R., Stevens, M., and Wagner, H. (2013) vegan: Community Ecology Package. See: http://CRAN.R-project.org/package=vegan.

[23] Qin, J., Li, R., Raes, J., Arumugam, M., and Burgdorf, K. S., et al. (2010) A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, **464**, 59–65.

[24] Qin, J., Li, Y., Cai, Z., Li, S., and Zhu, J. et al. (2012) A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, **490**, 55–60.

[25] Richter, D., Ott, F., Auch, A., Schmid, R., and Huson, D. (2008) MetaSim – A Sequencing Simulator for Genomics and Metagenomics. PLoS ONE **3**. See: http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0003373.

[26] Salvador, S. and Chan, P. (2004) Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. 16-th IEEE International Conference on Tools with Artificial Intelligence, 576–584. See: http://cs.fit.edu/~pkc/papers/ictai04salvador.pdf.

[27] Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B. (2011) Finding a 'kneedle' in a haystack: Detecting knee points in system behavior. 166–171. 31-st International Conference on Distributed Computing Systems. See: http://www1.icsi.berkeley.edu/~barath/papers/kneedle-simplex11.pdf.

[28] Schloissnig, S., Arumugam, M., Sunagawa, S., Mitreva, M., Tap, J., Zhu, A., Waller, A., Mende, D. R., Kultima, J. R., Martin, J., Kota, K., Sunyaev, S. R., Weinstock, G. M., and Bork, P. (2013) Genomic variation landscape of the human gut microbiome. *Nature*, **493**, 45–50.

[29] Segata, N., Izard, J. Waldron, L., Gevers, D., Miropolsky, L., Garrett, W. S., and Huttenhower, C. (2011) Metagenomic biomarker discovery and explanation. *Genome Biology*, **12**, R60.

[30] Segata, N., Waldron, L., Ballarini, A., Narasimhan, V., Jousson, O., and Huttenhower, C. (2012) Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods*, **9**, 811–814.

[31] Schwartz, S., Oren, R., and Ast, G.(2011) Detection and removal of biases in the analysis of NGS reads. *PLoS ONE*, **6**, e16685. doi:10.1371/journal.pone.0016685.
[32] Teo, S., Pawitan, Y., Ku, C., Chia, K., and Salim A. (2012) Statistical challenges associated with detecting copy number variations with next-generation sequencing. *Bioinformatics*, **28**, 2711–2718.
[33] VanDerWal, J., Falconi, L., Januchowski, S., Shoo, L., and Storlie, C. (2014) SDMTools: Species Distribution Modelling Tools: Tools for processing data associated with species distribution modelling exercises. R package version 1.1-221. See http://CRAN.R-project.org/package=SDMTools.
[34] Westfall, P. and Young, S. (1993) *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment.* Wiley, New York.

Camilo Valdes
Center for Computational Sciences
University of Miami
USA
E-mail address: cvaldes3@med.miami.edu

Meghan Brennan
Department of Anesthesiology
University of Florida
USA
E-mail address: meghanmbrennan@gmail.com

Bertrand Clarke
Department of Statistics
University of Nebraska Lincoln
USA
E-mail address: bclarke3@unl.edu

Jennifer Clarke
Department of Food Science and Technology
University of Nebraska Lincoln
USA
E-mail address: jclarke3@unl.edu