BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 22 2014, pages 3257–3263
doi:10.1093/bioinformatics/btu528

Data and text mining

Advance Access publication August 5, 2014

Thresher: an improved algorithm for peak height thresholding of microbial community profiles

Verena Starke* and Andrew Steele

Carnegie Institution of Washington, Geophysical Laboratory, Washington DC 20015, USA

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: This article presents Thresher, an improved technique for finding peak height thresholds for automated rRNA intergenic spacer analysis (ARISA) profiles. We argue that thresholds must be sample dependent, taking community richness into account. In most previous fragment analyses, a common threshold is applied to all samples simultaneously, ignoring richness variations among samples and thereby compromising cross-sample comparison. Our technique solves this problem, and at the same time provides a robust method for outlier rejection, selecting for removal any replicate pairs that are not valid replicates.

Results: Thresholds are calculated individually for each replicate in a pair, and separately for each sample. The thresholds are selected to be the ones that minimize the dissimilarity between the replicates after thresholding. If a choice of threshold results in the two replicates in a pair failing a quantitative test of similarity, either that threshold or that sample must be rejected. We compare thresholded ARISA results with sequencing results, and demonstrate that the Thresher algorithm outperforms conventional thresholding techniques.

Availability and Implementation: The software is implemented in R, and the code is available at http://verenastarke.wordpress.com or by contacting the author.

Contact: vstarke@ciw.edu or http://verenastarke.wordpress.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on December 10, 2013; revised on July 30, 2014; accepted on July 31, 2014

1 INTRODUCTION

This article presents a new peak thresholding algorithm called 'Thresher' that treats samples and replicates independently, taking their richness into account. We will demonstrate that the Thresher algorithm (and the software of the same name that implements it) offers improved performance over previous approaches to thresholding.

DNA profiling or fingerprinting techniques are essential to microbial ecology. For example, automated rRNA intergenic spacer analysis, or ARISA (Fisher and Triplett, 1999), and terminal restriction fragment length polymorphism, or T-RFLP (Liu *et al*., 1997), are widely used. These techniques depend on fluorescent-labeled DNA fragments, discriminating the sizes of fragments present in a sample. Analysis profiles are typically run in replicates.

When analyzing ARISA or T-RFLP electropherogram profiles, instrument noise and sample noise are typically present. Instrument noise refers to artifacts produced by the instrument and is usually removed by applying a conservative minimum peak height threshold to the raw electropherograms (Boeer *et al*., 2009; Fuhrman *et al*., 2006, 2008; Kent *et al*., 2007; Luna et al., 2006; Popa *et al*., 2009; Wood *et al*., 2008; Yannarell and Triplett, 2004).

Sample noise, in contrast, refers to false peaks that can result from Polymerase Chain Reaction (PCR) artifacts or sample contaminants, and is normally dealt with by applying an additional threshold based on the percentage of total amplified DNA. It is particularly important to set such thresholds carefully when techniques based on the presence or absence of peaks are to be used in the subsequent analyses (Hewson and Fuhrman, 2006; Lee *et al*., 2012; Yannarell and Triplett, 2005, 2004), as presence/absence analyses disproportionately weight weak peaks.

We argue that the threshold used to eliminate sample noise must be sample dependent. In past works, such thresholds have typically been applied across a whole sample set, treating all samples identically (Buchan *et al*., 2003; Caravati *et al*., 2010; Carson *et al*., 2009; Danovaro *et al*., 2006; Fuhrman *et al*., 2006; Hewson and Fuhrman, 2004, 2006; Luna *et al*., 2006; Popa *et al*., 2009; Ramette, 2009).

The normal procedure in ARISA data reduction is to standardize data, allowing for more direct comparison of samples with one another. Each standardized peak height gives the percentage of the total counts of peaks for that replicate or sample. This procedure is particularly important when ARISA runs differ in their peak intensity. In most previous ARISA processing procedures, a common threshold is applied to all samples after standardization. However, a problem can arise when cross-comparing standardized samples to which a common threshold has been applied, because such a comparison ignores richness variations among samples. Samples with similar peak heights but different richness can be standardized to dramatically different percentages because of variations in the amount of total counts. Therefore, two samples with different richness but similar peak heights can have different percentages for equivalent peaks, reducing their effective height in the richer sample. This effect can put true data peaks below a threshold for a rich sample, resulting in unwanted data loss. So, instead of using a common threshold across all samples, it is important to select thresholds that are sample dependent and take richness into account.

A similar problem can arise when comparing two replicates in a pair. For example, a replicate pair can be identical in data peak distribution and sample noise level, except that the peaks in one

*To whom correspondence should be addressed.

© The Author 2014. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

replicate have lower intensities than equivalent peaks in the other. Standardization will equalize the sample, causing noise peaks to gain weight in the lower intensity replicate. Some noise peaks could then be counted as data peaks if not thresholded properly. Again, it is more appropriate to apply a properly chosen threshold to each replicate separately.

This article presents an improved technique for finding peak height thresholds for ARISA profiles and similar data. Our technique also provides a robust method for outlier rejection, selecting for removal any replicate pairs that are so different from one another that they cannot be considered useful replicates.

2 APPROACH

2.1 Threshold selection

Our approach to setting thresholds that eliminate sample noise is based on comparing two replicate profiles for each sample in a set. Thresholds are calculated individually for each replicate in a pair, and separately for each sample. The thresholds are selected to be the ones that minimize the dissimilarity between the replicates after thresholding. Figure 1 illustrates the important point that thresholds must be calculated separately for each replicate in a pair to correctly find the minimum dissimilarity. A single threshold for both replicates would limit investigation to the diagonal of the diagram (Fig. 1A). In contrast, Thresher investigates all threshold combinations within the full two-dimensional solution space (Fig. 1B). If a choice of threshold results in the two replicates in a pair failing some quantitative test of similarity, either that threshold or that sample must be rejected. As we show below, an appropriate test of similarity can be used both to set thresholds and to recognize 'outlier' samples that should not be used in the analysis.

The test of similarity must be chosen with care. If peak heights were to be considered, then standard statistical measures of similarity like Pearson's product-moment correlation coefficient (Pearson's r) could be used. However, in methods like ARISA, presence or absence of peaks is usually analyzed rather than using their heights. Popular measures of similarity that are based on presence/absence alone include Sørensen's index [which is equal to one minus Bray-Curtis dissimilarity (Bray and Curtis, 1957)] and Jaccard's index (Jaccard, 1901). The measure of similarity used to select the thresholds should be compatible with the analysis techniques that will be used subsequently. Because ARISA analysis typically involves presence/absence, we use Bray–Curtis dissimilarity.

2.2 Outlier rejection

The measure of dissimilarity between the replicate pairs can also be used for outlier rejection. One quantitative approach to this problem for ARISA data was presented by Ruan *et al.* (2006). In our approach, the Bray–Curtis dissimilarities for each thresholded replicate pair are compared with one another, and a pair is declared to be an outlier if their dissimilarity is above some number of standard deviations chosen number by the user.

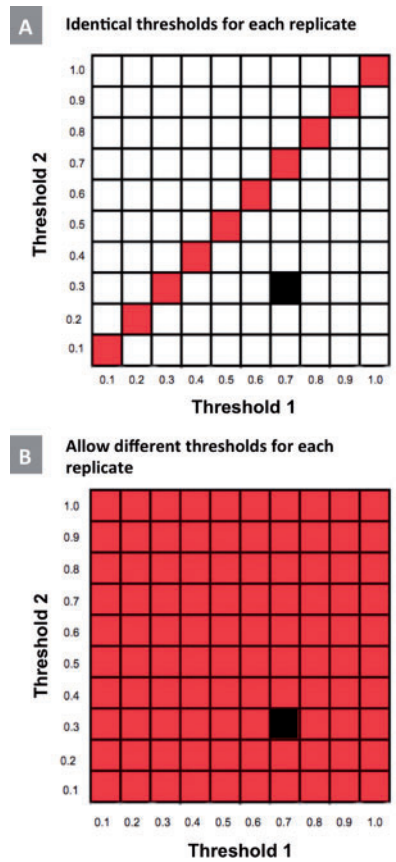

Fig. 1. Schematic representation of threshold selection, illustrating why separate thresholds should be selected for each replicate. Red squares represent the combinations of thresholds for which Bray–Curtis dissimilarities between the replicates are calculated. (**A**) Single thresholds for both replicates, ranging from 0.1 to 1%, in steps of 0.1%. Only thresholds along the diagonal are investigated in this approach. (**B**) Separate thresholds for each replicate. Black box represents the threshold combination with the lowest dissimilarity between both replicates. If the lowest dissimilarity lies off the diagonal, separate thresholds for each replicate are necessary

3 IMPLEMENTATION

In implementing the approach described above, steps are performed in the following order:

Step 1: Application of a minimal threshold, in fluorescent units, to all profiles, to eliminate instrument noise.

Step 2: Standardization of all profiles to 100%.

Step 3: Application of a range of possible thresholds to both replicates in each pair.

Step 4: Fixed-width binning of both replicates in each pair for all possible threshold combinations.

Step 5: Presence/absence transformation.

Step 6: Computation of Bray–Curtis dissimilarity for all possible threshold combinations.

Step 7: Selection of the optimal thresholds for each replicate in each pair.

Step 8: Identification of outlier samples.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

Steps 1 and 2 were discussed above in Section 1. The additional steps are described below.

Step 3: The goal of Thresher is to assess all possible combinations of reasonable thresholds, selecting the two best thresholds for each pair. Threshold values used in the literature span the range from 0.09 to 1.0% of total standardized fluorescence (Buchan *et al.*, 2003; Carson *et al.*, 2009; Hewson and Fuhrman, 2004; Popa *et al.*, 2009; Ramette, 2009). Therefore, we allow the choice of threshold for each replicate to vary from 0.0 to 1.0% in steps of 0.01%, resulting in a total of $101 \times 101 = 10\,201$ possible combinations. For all combinations, the thresholds are applied to each pair by eliminating all peaks with a standardized peak height lower than the threshold.

Step 4: The standardized ARISA data are binned to compensate for uncertainty in the estimates of ARISA fragment length (Fuhrman *et al.*, 2006). For this step, we use the fixed binning capability of dpbin (Ruan *et al.*, 2006). We chose fixed binning over dynamic binning because of its computational efficiency and also because we find that both binning processes lead to similar results for this purpose.

Thresholding is performed before binning because unwanted peaks could otherwise influence binning results.

Step 5: Each profile is transformed to presence and absence, equalizing all peak heights as is typically performed for ARISA data.

Step 6: Bray–Curtis dissimilarity is calculated for each replicate pair for all combinations of the pair's two thresholds. The result is a 101×101 matrix of Bray–Curtis dissimilarity values.

Step 7: Two thresholds are selected that yield the best result for a replicate pair. Thresher finds the matrix location with lowest dissimilarity, $\mathrm{Dis}_{\mathrm{min}}$.

Step 8: Once thresholds have been selected for all replicates in all pairs, the final step is to identify any replicate pairs that should be rejected as outliers. This step is performed as follows:

8.1. Determine the Bray–Curtis dissimilarity value (replicate 1 versus replicate 2) for the chosen thresholds.

8.2. Determine the median and standard deviation of all dissimilarity values for all replicate pairs. The median is less influenced by outliers and therefore more appropriate than the mean.

8.3. Reject any replicate pairs that have a dissimilarity value more than N standard deviations above the median.

8.4. Remove these replicate pairs and recalculate the median and standard deviation without the influence of outliers.

The choice of N is important, as it represents the investigator's definition of what constitutes an outlier. We recommend use of $N = 2$, so that all dissimilarities of non-outlier replicate pairs are within two standard deviations of the median.

4 RESULTS

4.1 Environmental data from troll springs, Svalbard

To demonstrate application of Thresher, we use a dataset for microbial communities from Troll springs on Svalbard in the high Arctic (Starke *et al.*, 2013). Microorganisms there exist under an unusually wide range of environmental conditions: in warm water as biofilms, in moist granular materials, and in cold, dry rock as endoliths. These samples vary in their richness and community composition and were analyzed using ARISA as well as 454 pyrosequencing. The 454 results were presented in the study by Starke *et al.* (2013); we present the ARISA results for the first time here. Preparation of the samples used for ARISA processing is described in Supplementary Material. To show how the Thresher algorithm performs, we assume that the sequencing dataset represents the true community composition, and compare it with the thresholded ARISA data.

4.2 Dissimilarities of evenly and unevenly distributed replicates

Figure 2 shows two examples of the 101×101 dissimilarity matrix for two endolithic samples, displaying dissimilarity as a function of the thresholds for each replicate in the pair as color contour maps (Fig. 2a) and the corresponding standardized ARISA profiles for both replicates (Fig. 2b).

Ideally, the lowest values of dissimilarity should lie along the diagonal, drawn as a dotted line on the contour maps in Figure 2a. The sample on the left has two replicates with similar peak amounts and distributions. When thresholds are applied to this sample, the lowest values of dissimilarity are found close to the diagonal.

However, if two replicates differ significantly in their distribution or number of peaks (e.g. because of noise highlighted by arrows in Fig. 2b, right), then the lowest dissimilarity may lie off the diagonal, and each replicate in the pair requires its own threshold. The greater richness in one replicate means that peak heights in that replicate are reduced by standardization. A common threshold applied to both replicates would result in unbalanced treatment of the standardized peak values. The algorithm will select threshold values that lie off the diagonal, accounting for the different initial richness in each replicate of the pair.

4.3 Outlier rejection

Figure 3 illustrates samples with thresholds chosen by the Thresher algorithm, and by using three single thresholds for all samples. Sample 3 has a Bray–Curtis dissimilarity substantially greater than all the others. Using the outlier rejection criterion of two standard deviations above the median (i.e., $N = 2$), it is rejected as an outlier in the single and separate threshold selection scenario. Additionally, when choosing a single threshold, samples 12 or 5 should be rejected under the same outlier rejection criterion for a single threshold of 0.1 or 0.9%, respectively. However, when choosing thresholds independently using Thresher, both of these samples can be retained.

4.4 Determination of the best thresholding method

As noted above, the purpose of ARISA is 'community fingerprinting', allowing an investigator to quickly profile the makeup of a microbial community. Our goal in creating Thresher is to provide a thresholding technique that allows ARISA community fingerprinting to be as biologically accurate as possible. Thresher's performance can be evaluated both on a

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

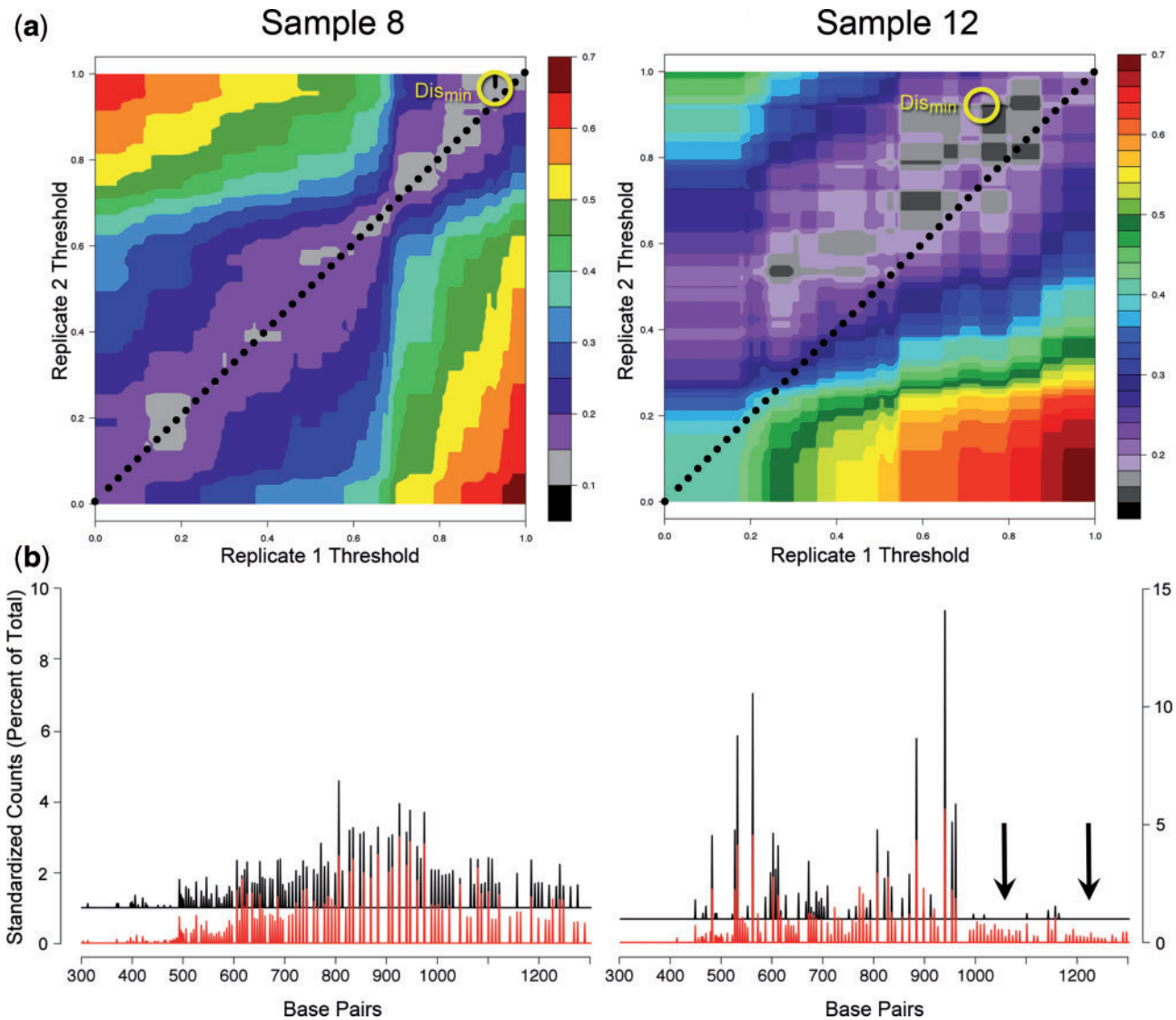

Fig. 2. Visual display of dissimilarity between two replicates in a sample. (**a**) Calculated 101 × 101 dissimilarity matrices for two samples, displayed as contour maps. (**b**) Standardized ARISA profiles of replicate profiles (one red and one black) for two samples. Similar profiles lead to a relatively symmetric dissimilarity matrix (left sample), while major differences (highlighted by arrows) can lead to an asymmetric dissimilarity matrix (right sample). The *y*-axis for the second replicate in each pair is shifted upward by 1% for better visualization

sample-by-sample basis and, more comprehensively, on a suite of related samples.

To evaluate the performance of Thresher sample by sample, we compared Bray–Curtis dissimilarities between replicates after thresholding both by Thresher and by using single thresholds (Fig. 4). Thresher consistently finds the lowest dissimilarity between replicates, outperforming traditional single thresholding by this metric. No single threshold performs better than all others for all samples. For example, sample 6 has a lower dissimilarity when using a common threshold of 0.1%, whereas sample 1 is treated better with a threshold of 0.9%.

To evaluate the performance of Thresher in determining community structure across a suite of samples, we compared both Thresher and the traditional single thresholding technique with results obtained from the pyrosequencing dataset from Troll springs (Starke *et al.*, 2013). The DNA extracted from these samples was used for sequencing as well as the ARISA analysis.

For this purpose, we assumed that the pyrosequencing dataset represents the true community composition that ARISA is attempting to approximate. Whole-community fingerprinting techniques like ARISA reveal only a description of microbial community structure rather than identifying specific named taxonomic levels. Therefore, we judged the quality of the thresholding algorithm simply by noting the closeness of community structure of the post-thresholding ARISA-derived dissimilarity matrix to the equivalent matrix derived from sequencing data.

We tested Dis_{min} as chosen by Thresher for each replicate and each sample, and compared it with results for single thresholds of 0.1 (least stringent allowing more peaks), 0.5 and 0.9% (most stringent) that were applied across the whole dataset.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

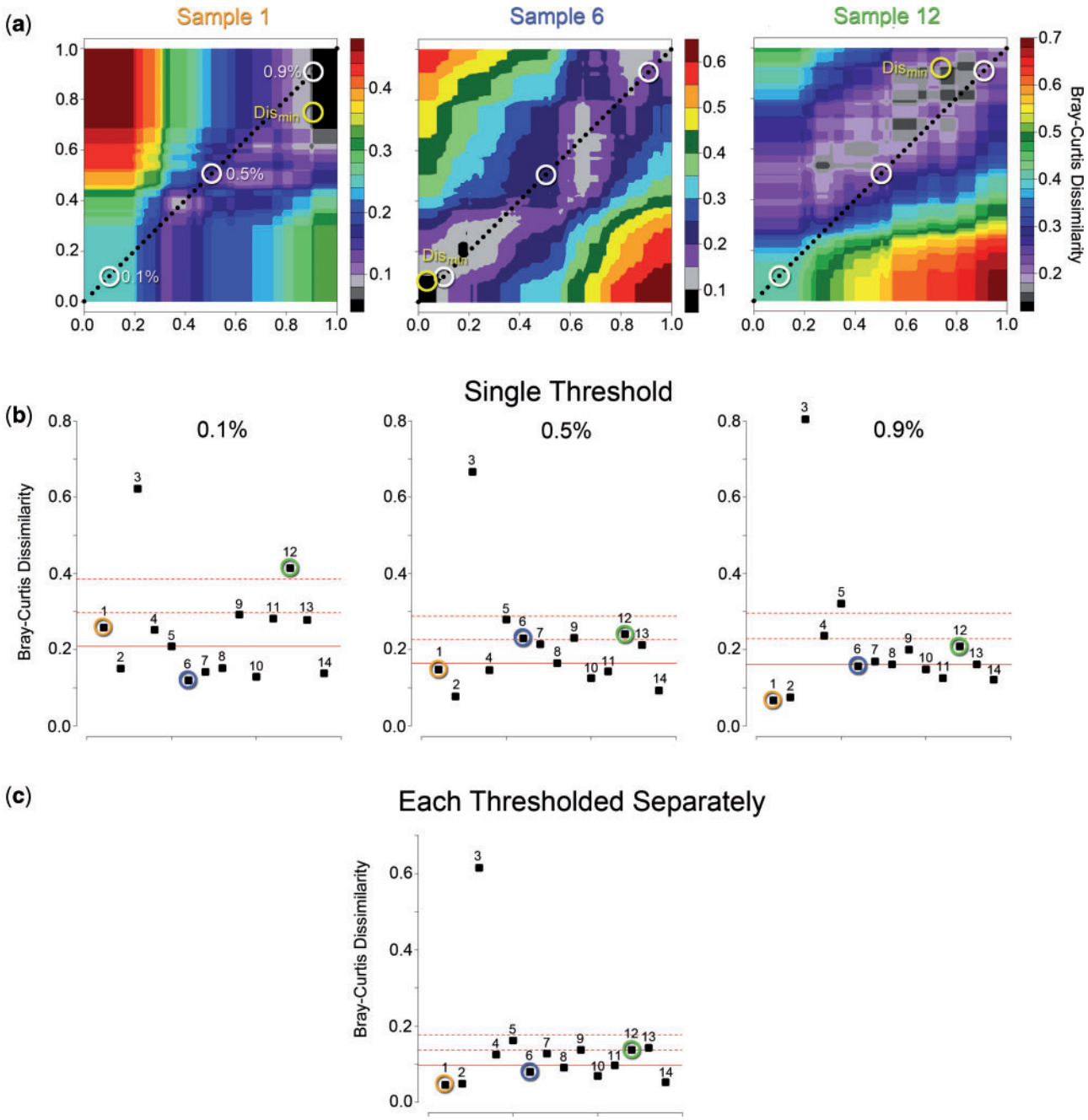

Fig. 3. ARISA thresholding and outlier rejection. (**a**) Contour maps of calculated 101 × 101 dissimilarity matrices for three samples. Yellow circles show thresholds selected by the algorithm by finding the minimum dissimilarity Dis_{min}. White circles show three possible common thresholds. (**b**) Bray–Curtis dissimilarity for all samples using the three different common thresholds. Colored circles correspond to the samples shown in the top row. (**c**) Bray–Curtis dissimilarity for all samples using independently selected thresholds and Dis_{min}. Solid red lines in (b) and (c) show the median value, and dashed lines lie one and two standard deviations above the median after outlier removal has been performed

After applying the thresholds, binning is necessary to combine the two replicates in a pair into a single profile, and to allow profiles to be compared with one another. The standardized and thresholded ARISA data were binned using the dynamic binning capability of dpbin (Ruan *et al.*, 2006).

The 454 sequencing reaches greater community depths and, consequently, contains higher Operational Taxonomic Unit (OTU) counts compared with ARISA. Therefore, to adjust the sequencing OTU count as closely as possible to the corresponding sample in the ARISA dataset, we applied a threshold to the sequencing data to eliminate minor OTUs. This resulted in similar OTU counts for each sample in the sequencing and the ARISA datasets.

To evaluate the performance of the two competing thresholding techniques, we adopted a statistical approach widely used by

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

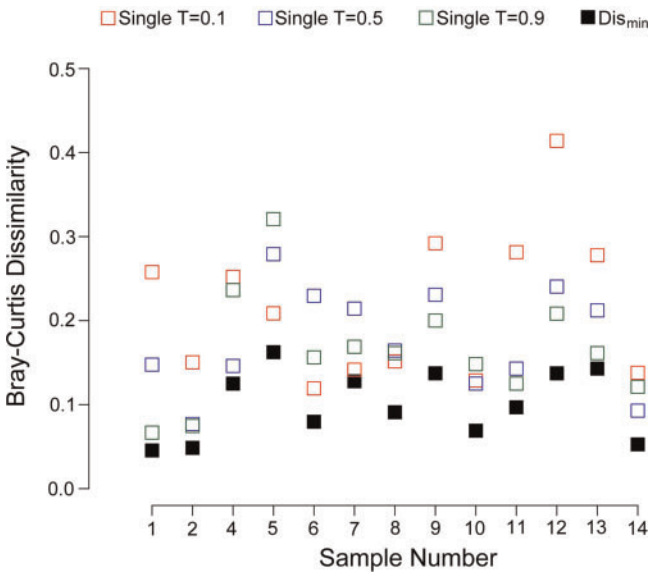

Fig. 4. Comparison of Bray–Curtis dissimilarity between replicates in a pair for all samples except the rejected outlier (sample 3) using three different common thresholds and separate thresholds. Open squares correspond to common thresholds of 0.1% (red), 0.5% (blue) and 0.9% (green). Solid black squares represent independently selected thresholds determined by Thresher

ecologists to describe microbial communities. Specifically, we performed a presence-absence transformation on the data, and then calculated Bray–Curtis dissimilarities among all sample pairs. A total of 10 samples were used, yielding 10×10 dissimilarity matrices. We obtained such matrices for the 454 sequencing data, for ARISA data that were thresholded using single thresholds of 0.1, 0.5 and 0.9% and for the same ARISA data thresholded using Dis_{min} as chosen by Thresher.

A variety of approaches exist for drawing biological inferences from large dissimilarity matrices. Some of the techniques most familiar to biologists (e.g. non-metric multidimensional scaling) suffer from the shortcoming that they sacrifice quantitative information to allow for easy visualization (Borg and Groenen, 2005). Because we wished to perform a robust comparison of thresholding techniques, we instead used a simpler, but rigorously, quantitative approach. First, we normalized all of the dissimilarity matrices to one another, allowing direct intercomparison. We then calculated the difference between each normalized ARISA-derived matrix and the 454-derived matrix that we assume represents 'biological truth'. This approach is functionally equivalent to visual comparison of results obtained from common ordination techniques, but it reveals the best match between dissimilarity matrices in a straightforward and quantitative manner. The thresholding technique that provides the best match, as determined by the sum of the squares of the differences between normalized matrices, represents the best result.

The results are summarized in Table 1. As shown, Dis_{min} as chosen by Thresher yielded a better match to the sequencing dataset (i.e. a smaller sum of squares) than when the whole dataset was treated with any single threshold. Stated in biological terms, this means that the ARISA-based microbial community fingerprint obtained using Thresher was closer to sequencing results—and, we therefore assume, closer to biological truth—than the fingerprint obtained using any single threshold.

Table 1. Comparison of the ARISA dataset to the sequencing dataset

Threshold choices	Sum of squares between the ARISA and sequencing matrix
Dis_{min}	0.471
Single T = 0.5	0.576
Single T = 0.1	0.588
Single T = 0.9	0.621

Note. The differences are calculated as sum of squares where the lowest sum represents the highest similarity. Values are ordered according to their sum of squares. ARISA matrices are thresholded separately or one single threshold ('Single T') for all replicates. Threshold values are in percentage.

5 CONCLUSIONS

This article presents Thresher, a sample-by-sample peak thresholding method that works by maximizing the similarity of replicates by using their best individual thresholds. Thresher outperforms single thresholding methods applied across the whole dataset. It also identifies and eliminates outlier samples.

Despite the availability of high-throughput sequencing techniques, ARISA and T-RFLP remain among the fastest and easiest means of assessing variations in microbial community structure. They are particularly well suited to comparing large numbers of samples, recognizing differences in community structure rather than identifying specific taxa. Microbial samples can exhibit large variations in richness, requiring that the essential thresholding be performed on a sample-by-sample basis.

The technique can be readily applied to ARISA data from a range of environmental samples. In principle, our thresholding approach could also be useful for other forms of data analysis that seek to separate valid peaks from sample noise peaks.

The software is implemented in R (R Development Core Team, 2012). The code is generally available to the research community and can be found in the Supplementary Material or on the Web site http://verenastarke.wordpress.com.

ACKNOWLEDGEMENTS

The authors thank Frank Robb, Russell Hill, Jocelyne DiRuggierio, Hal Schreier and Feng Cheng for valuable input and feedback.

Funding: NASA ASTEP grants NNX09AB74G and NNX12AP77G, the CIW-NASA Astrobiology Institute (NNA09DA81A) and the W. M. Keck Foundation (June 29, 2007).

Conflict of interest: none declared.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

REFERENCES

Boeer,S.I. *et al.* (2009) Time- and sediment depth-related variations in bacterial diversity and community structure in subtidal sands. *ISME J.*, **3**, 780–791.

Borg,I. and Groenen,P.J.F. (2005) *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York.

Bray,J.R. and Curtis,J. (1957) An ordination of the upland forest communities of southern Wisconsin. *Ecol. Monogr.*, **27**, 326–349.

Buchan,A. *et al.* (2003) Dynamics of bacterial and fungal communities on decaying salt marsh grass. *Appl. Environ. Microbiol.*, **69**, 6676–6687.

Caravati,E. *et al.* (2010) Picocyanobacterial assemblages in ultraoligotrophic Andean lakes reveal high regional microdiversity. *J. Plankton Res.*, **32**, 357–366.

Carson,J.K. *et al.* (2009) Minerals in soil select distinct bacterial communities in their microhabitats. *FEMS Microbiol. Ecol.*, **67**, 381–388.

Danovaro,R. *et al.* (2006) Comparison of two fingerprinting techniques, terminal restriction fragment length polymorphism and automated ribosomal intergenic spacer analysis, for determination of bacterial diversity in aquatic environments. *Appl. Environ. Microbiol.*, **72**, 5982–5989.

Fisher,M. and Triplett,E.W. (1999) Automated approach for ribosomal intergenic spacer analysis of microbial diversity and its application to freshwater bacterial communities. *Appl. Environ. Microbiol.*, **65**, 4630–4636.

Fuhrman,J.A. *et al.* (2008) A latitudinal diversity gradient in planktonic marine bacteria. *Proc. Natl Acad. Sci. USA*, **105**, 7774–7778.

Fuhrman,J.A. *et al.* (2006) Annually reoccurring bacterial communities are predictable from ocean conditions. *Proc. Natl Acad. Sci. USA*, **103**, 13104–13109.

Hewson,I. and Fuhrman,J.A. (2006) Improved strategy for comparing microbial assemblage fingerprints. *Microb. Ecol.*, **51**, 147–153.

Hewson,I. and Fuhrman,J.A. (2004) Richness and diversity of bacterioplankton species along an estuarine gradient in Moreton Bay, Australia. *Appl. Environ. Microbiol.*, **70**, 3425–3433.

Jaccard,P. (1901) Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bull. Soc. Vaudoise Sci Nat.*, **37**, 241–272.

Kent,A.D. *et al.* (2007) Synchrony in aquatic microbial community dynamics. *ISME J.*, **1**, 38–47.

Lee,C.K. *et al.* (2012) The Inter-Valley Soil Comparative Survey: the ecology of Dry Valley edaphic microbial communities. *ISME J.*, **6**, 1046–1057.

Liu,W.T. *et al.* (1997) Characterization of microbial diversity by determining terminal restriction fragment length polymorphisms of genes encoding 16S rRNA. *Appl. Environ. Microbiol.*, **63**, 4516–4522.

Luna,G.M. *et al.* (2006) DNA extraction procedure: a critical issue for bacterial diversity assessment in marine sediments. *Environ. Microbiol.*, **8**, 308–320.

Popa,R. *et al.* (2009) Limitations and benefits of ARISA intra-genomic diversity fingerprinting. *J. Microbiol. Methods*, **78**, 111–118.

R Development Core Team. (2012) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramette,A. (2009) Quantitative community fingerprinting methods for estimating the abundance of operational taxonomic units in natural microbial communities. *Appl. Environ. Microbiol.*, **75**, 2495–2505.

Ruan,Q. *et al.* (2006) A dynamic programming algorithm for binning microbial community profiles. *Bioinformatics*, **22**, 1508–1514.

Starke,V. *et al.* (2013) Microbial community composition and endolith colonization at an Arctic thermal spring are driven by calcite precipitation. *Environ. Microbiol. Rep.*, **5**, 648–659.

Wood,S.A. *et al.* (2008) Sources of edaphic cyanobacterial diversity in the Dry Valleys of Eastern Antarctica. *ISME J.*, **2**, 308–320.

Yannarell,A.C. and Triplett,E.W. (2005) Geographic and environmental sources of variation in lake bacterial community composition. *Appl. Environ. Microbiol.*, **71**, 227–239.

Yannarell,A.C. and Triplett,E.W. (2004) Within- and between-lake variability in the composition of bacterioplankton communities: investigations using multiple spatial scales. *Appl. Environ. Microbiol.*, **70**, 214–223.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016