




microDecon: A highly accurate read-subtraction tool for the post-sequencing removal of contamination in metabarcoding studies

Donald T. McKnight¹  | Roger Huerlimann¹  | Deborah S. Bower^{1,2} |
Lin Schwarzkopf¹  | Ross A. Alford¹ | Kyall R. Zenger¹

¹College of Science and Engineering, James Cook University, Townsville, Queensland, Australia

²University of New England, Armidale, New South Wales, Australia

Correspondence

Donald T. McKnight, College of Science and Engineering, James Cook University, Townsville, Queensland, Australia.
Email: donald.mcknight@my.jcu.edu.au

Funding information

Australian Wildlife Society; Skyrail Rainforest Foundation; Australian Society of Herpetologists; Holsworth Wildlife Research Endowment

Abstract

Contamination is a ubiquitous problem in microbiome research and can skew results, especially when small amounts of target DNA are available. Nevertheless, no clear solution has emerged for removing microbial contamination. To address this problem, we developed the R package *microDecon* (<https://github.com/donaldtmcknight/microDecon>), which uses the proportions of contaminant operational taxonomic units (OTUs) or amplicon sequence variants (ASVs) in blank samples to systematically identify and remove contaminant reads from metabarcoding data sets. We rigorously tested *microDecon* using a series of computer simulations and a sequencing experiment. We also compared it to the common practice of simply removing all contaminant OTUs/ASVs and other methods for removing contamination. Both the computer simulations and our sequencing data confirmed the utility of *microDecon*. In our largest simulation (100,000 samples), using *microDecon* improved the results in 98.1% of samples. Additionally, in the sequencing data and in simulations involving groups, it enabled accurate clustering of groups as well as the detection of previously obscured patterns. It also produced more accurate results than the existing methods for identifying and removing contamination. These results demonstrate that *microDecon* effectively removes contamination across a broad range of situations. It should, therefore, be widely applicable to microbiome studies, as well as to metabarcoding studies in general.

KEYWORDS

16S, bacteria, bioinformatics, controls, decontaminate, microbiome

1 | INTRODUCTION

Advances in sequencing technology have greatly expanded our ability to harness the power of metabarcoding for studying microbial communities, and it is now possible to sequence an entire community using a minuscule amount of starting material. However, our

ability to detect organisms from just a few fragments of nucleic acid is both a blessing and a curse; while it greatly improves our detection of target species, it also carries the risk of sequence contamination. Indeed, there is growing recognition that contamination (especially bacterial contamination) is a serious hindrance in microbiome studies, and several studies have documented that contamination is

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2019 The Authors. *Environmental DNA* published by John Wiley & Sons Ltd

ubiquitous, even in places that should be DNA/RNA free, such as molecular grade water, PCR polymerases, and DNA extraction kits (Corless et al., 2000; Hang et al., 2014; Kulakov, McAlister, Ogden, Larkin, & O'Hanlon, 2002; Peters et al., 2004; Shen, Rogelj, & Kieft, 2006; Weiss et al., 2014). Contamination is particularly problematic for studies using low-biomass samples, where even a small amount of contamination can severely affect the results (Salter et al., 2014).

Although this problem is widespread, no clear solution has emerged. Good laboratory techniques are important but cannot eliminate contamination, because many kits and PCR reagents are contaminated (Salter et al., 2014) and contamination can occur when the samples are being collected. To address these issues, strategies such as using a single kit for all extractions or randomizing samples across kits and PCR runs have been recommended (Salter et al., 2014; Weiss et al., 2014). Additionally, various methods have been proposed for removing contamination from kits and reagents, but mixed levels of success have been reported, and they often cause PCR inhibition (Champlot et al., 2010; Mohammadi, Reesink, Vandembroucke-Grauls, & Savelkoul, 2005; Rueckert & Morgan, 2007).

None of the proposed methods are likely to eliminate contamination in all cases; therefore, there is still a need to identify and deal

with contamination postsequencing. Some researchers have advocated for a log-ratio test for identifying contamination (Robinson, Crabtree, Mattick, Anderson, & Dunning Hotopp, 2017), while others have suggested that contaminants can be identified by looking for negative correlations between prestandardization amplicon concentration and the relative abundance of operational taxonomic units (OTUs) postsequencing (Jervis-Bardy et al., 2015). Similarly, Davis, Proctor, Holmes, Relman, and Callahan (2018) proposed the R package decontam for using presequencing quantification data to identify contaminant amplicon sequencing variants (ASVs; for simplicity, we will refer to OTUs hereafter, but all concepts and methods we will discuss also apply to ASVs). Perhaps the most effective and straightforward suggestion is simply to use negative controls (hereafter called "blanks") that are carried through the entire collection, extraction, amplification, and sequencing process (Barton, Taylor, Lubbers, & Pemberton, 2006; Salter et al., 2014). These blanks can then be used to quantify the levels of contamination present.

Regardless of the mechanism used to detect contamination, the problem of what to do once it has been detected remains. One option is to simply report the level of contamination, but this is unsatisfactory as it is difficult to know the influence of contamination on comparisons among groups. To solve this dilemma, some researchers

Box 1. Definitions of Terms

- Blank = a negative control collected at the same time as the samples and carried through the entire extraction, amplification, and sequencing process
- Constant = an OTU that is entirely contamination and is used as the basis for decontaminating samples
- Contaminant OTUs = OTUs that amplified in the blank
- Entirely contamination = contaminant OTUs that would not be found on an uncontaminated sample (i.e., they do not occur on the species, substrate, etc. that is being studied)
- OTU = operational taxonomic unit. For simplicity and consistency with our sequencing experiment, we will refer to "OTUs" throughout, but this method is not specific to OTUs and works equally well for amplicon sequencing variants (ASVs).
- OTUs not in the blank = OTUs that did not amplify in the blank
- Overlapping OTUs (overlap) = contaminant OTUs that would also be found on an uncontaminated sample (i.e., they occur on the species, substrate, etc. that is being studied as well as in the source of contamination; thus, some of their reads are real and some are from contamination)
- Simulation control = a comparison between uncontaminated and decontaminated/contaminated samples using only the OTUs that were not in the blank (subsetting is done before any transformations). Because those OTUs are unaffected by contamination, they act as a control for background heterogeneity.

	OTU ID	Blank	Uncontaminated sample	
Contaminant OTUs	OTU1	100	0	Entirely contamination
	OTU2	50	0	
	OTU3	20	0	
	OTU4	10	30	
OTUs not in the blank	OTU5	5	500	Overlapping OTUs
	OTU6	1	40	
	OTU7	0	300	
	OTU8	0	10	

Box 1 Figure 1. Hypothetical sequencing reads, illustrating the terms used in this paper (in an actual study, the uncontaminated sample would be unknown)

have advocated the use of mock communities that are extracted, amplified, and sequenced alongside actual samples (Brooks, 2016; Wilner et al., 2013). In some situations, this is likely to be a very useful approach, especially when working with low-diversity communities and in situations where a research group frequently works with similar communities. Indeed, in situations with little contamination, it may even be possible to use the mock community to establish an abundance threshold that can be used to filter out contamination (Brooks, 2016; Wilner et al., 2013). For many applications, such as sequencing diverse communities and exploratory research, however, constructing a meaningful mock community is often not feasible, and thresholds will not be effective for communities with either many rare OTUs or high quantities of contamination.

One obvious solution is to simply remove any contaminant OTUs from all samples (Jervis-Bardy et al., 2015; Segal et al., 2013). In cases where there are very few contaminant OTUs, or there is a solid biological basis for thinking those OTUs should not be present, or both, that may be a good solution. In many cases, however, contaminant OTUs are likely to occur naturally on the host or in the environment being studied, as well as being present as contamination (hereafter these will be called “overlapping OTUs”). Therefore, simply removing any contaminant OTUs removes potentially important data and can either artificially exaggerate or reduce any differences among groups (depending on whether those OTUs are equally abundant across groups). The recently developed R package *decontam* (Davis et al., 2018) attempts to solve this by using statistical models to identify OTUs in the blanks that should be removed, but there is still a risk of removing OTUs that were actually present in low numbers in the system being studied. A final option is to simply subtract the contaminant reads from the reads in the samples; however, this is also problematic because read depth typically differs among samples. Furthermore, because samples are amplified and standardized prior to sequencing, samples with few OTUs (such as contaminated blanks) will have more reads per OTU than diverse samples.

Because of the problems associated with the removal of contamination enumerated above, a better solution is clearly needed. Thus, we developed, and rigorously tested, the R package *microDecon*, which provides several easy-to-use tools for identifying and removing contamination. *microDecon* uses information from blank samples to calculate and remove the contaminant reads for each OTU, rather than simply consigning an entire OTU to contamination. As such, it provides a substantial improvement over current methods, and importantly, avoids the loss of useful data.

2 | METHODS

2.1 | *microDecon*

The package *microDecon* operates on the principle that all the samples will receive the same proportions of contamination from a common source. For example, if a contaminated reagent contains 100 ng/μl of OTU1 and 50 ng/μl of OTU2, then each sample should receive approximately twice as much OTU1 contamination as OTU2

contamination. Thus, if we can identify an OTU that is entirely contamination (hereafter referred to as the “constant”), we can use it to calculate the number of reads in the actual sample that originated from contamination. *microDecon* does this in the following steps (illustrated in Figure 1). First, it subsets the data to include only the contaminant OTUs (i.e., OTUs that amplified in the blank). Second, it estimates the number of overlapping OTUs and uses that estimate to identify the best OTU to use as the constant (the algorithms it uses are based on regression equations that we developed through numerous simulations; details in Appendix S1). Third, it divides the reads for each OTU in the blank by the number of reads for the constant in the blank. Fourth, it multiplies those values by the number of reads for the constant in the actual sample. This produces the number of reads in the actual sample that are from contamination, and those reads are then subtracted. This entire process is done iteratively for each sample. Thus, each sample is treated completely independently.

As an example, consider a sample and blank with two OTUs that amplified in the blank. In the blank, OTU1 has 1,000 reads, and OTU2 has 100 reads. Thus, the ratio for those OTUs in the blank is 10:1. If we also know that one of those OTUs is entirely contamination (i.e., a constant), we can use that to determine the number of reads in the sample that are from contamination for both OTUs. If, for example, we know that OTU1 is entirely contamination, and in the sample, OTU1 has 600 reads while OTU2 has 100 reads, we can deduce that all 600 reads for OTU1 are from contamination and, based on the 10:1 ratio in the blank, 60 of the reads for OTU2 are from contamination. Therefore, a decontaminated sample would have zero reads for OTU1 and 40 reads for OTU2. Because this method relies on the proportions of OTUs in the blank relative to a constant, rather than the raw number of reads, it does not require samples to have consistent amounts of starting material or read depths. Thus, the results of the example with two OTUs would be the same if the OTUs in the blank had one million reads and one hundred thousand reads (respectively) or ten reads and one read (respectively).

This method is clearly dependant on identifying an appropriate constant. The algorithms for doing this are described in detail in Appendix S1, but briefly, the percent difference between the proportions of reads in the blanks and portions of reads in the samples (i.e., the fourth table in Figure 1) are useful for determining if an OTU is entirely contamination. When the percent difference is positive, it suggests that an OTU is under-represented in the sample, likely indicating that it is entirely contamination; whereas when it is negative, it suggests that the OTU is over-represented in the sample, likely indicating that it is an overlapping OTU. Based on our simulations, most OTUs with a positive percent difference will perform well as a constant, but both very large and very small positive percent differences tend not to perform optimally. Therefore, we used extensive simulations to examine correlations between known parameters in a dataset and the rank of the best OTU to use as the constant. From those simulations, we developed several regression equations for identifying the constant, and *microDecon* automatically selects

An example showing a blank, uncontaminated sample and its contaminated counterpart. In a real study, the uncontaminated sample would be unknown.

	Blank (reads)	Uncontaminated sample (reads)	Contaminated sample (reads)
OTU1	5000	0	2500
OTU2	3000	2000	3500
OTU3	2000	100	1100
OTU4	1500	10	760
OTU5	600	0	300
OTU6	400	40	240
OTU7	50	0	25
OTU8	30	0	15
OTU9	20	20	30
OTU10	10	1	6
OTU11	0	4000	4000
OTU12	0	3000	3000
OTU13	0	500	500
OTU14	0	50	50
OTU15	0	10	10

Subset the data to just the contaminant OTUs (OTUs that amplified in the blank).

	Blank (reads)	Contaminated sample (reads)
OTU1	5000	2500
OTU2	3000	3500
OTU3	2000	1100
OTU4	1500	760
OTU5	600	300
OTU6	400	240
OTU7	50	25
OTU8	30	15
OTU9	20	30
OTU10	10	6

Convert reads to proportions. Do this separately for both the blank and the sample (for the sample use: sum of reads+[0.1*sum of reads]).

	Blank (proportions)	Contaminated sample (proportions)
OTU1	0.3965	0.2681
OTU2	0.2379	0.3754
OTU3	0.1586	0.1180
OTU4	0.1190	0.0815
OTU5	0.0476	0.0322
OTU6	0.0317	0.0257
OTU7	0.0040	0.0027
OTU8	0.0024	0.0016
OTU9	0.0016	0.0032
OTU10	0.0008	0.0006

Calculate the percent difference between the proportions, sort from highest percent difference to lowest, and use an algorithm* to select the best constant.

	Percent difference
OTU1	32.4
OTU5	32.4
OTU7	32.4
OTU8	32.4
OTU4	31.5
OTU3	25.6
OTU6	18.9
OTU10	18.9
OTU2	-57.8
OTU9	-102.9

Subtract the contaminant reads from the contaminated sample. This produces a decontaminated sample that matches the uncontaminated sample.

	Contaminated sample (reads)	Subtract contaminant reads from contaminated sample	Decontaminated sample (reads)	Uncontaminated sample (reads)
OTU1	2500	2500-2500	0	0
OTU2	3500	3500-1500	2000	2000
OTU3	1100	1100-1000	100	100
OTU4	760	760-750	10	10
OTU5	300	300-300	0	0
OTU6	240	240-200	40	40
OTU7	25	25-25	0	0
OTU8	15	15-15	0	0
OTU9	30	30-10	20	20
OTU10	6	6-5	1	1

Multiply the results by the number of reads for the constant in the contaminated sample.

	Blank divided by constant	Multiply result by constant in the contaminated sample	Contaminant reads
OTU1	166.7	166.7*15	2500
OTU2	100	100*15	1500
OTU3	66.7	66.7*15	1000
OTU4	50	50*15	750
OTU5	20	20*15	300
OTU6	13.3	13.3*15	200
OTU7	1.7	1.7*15	25
OTU8	1	1*15	15
OTU9	0.7	0.7*15	10
OTU10	0.3	0.3*15	5

Divide the reads for each OTU in the blank by the number of reads for the constant in the blank.

	Blank (reads)	Divide by constant in the blank	Blank divided by constant
OTU1	5000	5000/30	166.7
OTU2	3000	3000/30	100
OTU3	2000	2000/30	66.7
OTU4	1500	1500/30	50
OTU5	600	600/30	20
OTU6	400	400/30	13.3
OTU7	50	50/30	1.7
OTU8	30	30/30	1
OTU9	20	20/30	0.7
OTU10	10	10/30	0.3

FIGURE 1 The basic steps used by microDecon to decontaminate samples. The process is iterative and each sample is treated completely independently. The constant is an OTU that is entirely contamination (i.e., should not be present in an uncontaminated sample). Because the constant is entirely contamination, it can be used as a point of comparison to determine how many reads in the sample are from contamination. Percent differences are calculated as: $([\text{blank proportion} - \text{sample proportion}] / \text{blank proportion}) \times 100$. Some numbers reported in the fourth table appear to be slight deviations of the expected values based on the third table. This is simply an artefact of rounding the values in the third table to four decimal places. *Full details on the algorithms are available in Appendix S1

among those regressions based on the data set it is given (see Appendix S1 for details).

Due to the potential pitfalls of any novel method, we rigorously tested microDecon over a wide range of situations, including both simulated 16S data sets and a real, sequenced data set, to ensure that the method was robust. We also compared microDecon with the common strategy of simply removing all contaminant OTUs, the method of detecting and removing contaminant OTUs proposed in Jervis-Bardy et al. (2015), and the decontam R package (Davis et al., 2018). We used the primary function in the microDecon package (decon()) on its default values for all tests. The function, its input parameters, and the tests we used to identify the best default values are explained in the microDecon User's Guide.

2.2 | Simulation 1: Individual samples

We wrote a simulation in R (R Core Team, 2017) to test the utility of microDecon (Appendix S2). For each iteration, this simulation creates an uncontaminated microbial sample, as well as an artificial contaminant community. It then uses the contaminant community to contaminate the sample (a copy of the contaminant community is saved as a blank). Next it processes and “sequences” the sample and the blank. Finally, it uses microDecon to decontaminate the contaminated sample.

Within each iteration, each OTU in the contaminant community is multiplied by a number that is randomly selected from a user-defined normal distribution before adding the contamination to the sample (a new number is selected for each OTU). This simulates heterogeneity from DNA extraction and library preparation. Additionally, the communities are in-silico “sequenced” by repeatedly randomly selecting DNA copies from the entire community (each OTU is coded as a number of DNA copies), which simulates heterogeneity from actual sequencing. Full details on the simulation and input OTU distribution are available in Appendices S3 and S4.

We ran 100,000 iterations of this simulation over a broad range of situations, including varying amounts of starting material and varying amounts of contamination (varied both in terms of numbers of OTUs and DNA yield for those OTUs). For each iteration, the input parameters were randomly selected from the following values: number of OTUs that were entirely contamination = 0–150, number of OTUs not in the blank = 50–1000, and number of overlapping OTUs = 0–150 (OTUs were randomly sampled from a supplied distribution, resulting in varying amounts of DNA per OTU). We created within-iteration heterogeneity in the contamination that was applied to the sample by multiplying each OTU by a number that was randomly selected from a normal distribution with a mean between 0.15–1.0 and *SD* that was the mean multiplied by 0.1–0.7 (a new number was randomly selected for each OTU, and a new mean and *SD* were randomly selected for each iteration). This produced a median contamination level of 0.12 (range = 0.0002–10.4; i.e., the amount of contaminant DNA that was applied to a sample divided by the amount of DNA in the uncontaminated sample). Finally, the number of sequencing reads

for the blank and the sample were independently selected from a range of 18,000–20,000.

For each iteration, we calculated Bray–Curtis dissimilarities (BC) between the uncontaminated versus contaminated sample and uncontaminated versus decontaminated sample and used those dissimilarities to judge the effectiveness of microDecon. Throughout this study, we calculated all BC by transforming the data to proportions (McKnight et al., 2018) and using the vegan package in R (Oksanen et al., 2017). Additionally, we applied multiple linear regression to the results to see how different factors influenced the effectiveness of microDecon (results are presented in Appendix S3).

Finally, we ran 10,000 iterations of a slightly modified version of simulation 1 that tested the effects of simply removing contaminant OTUs (i.e., all contaminant OTUs were set to zero in the final sample). It returned BC for the contaminated versus uncontaminated sample, decontaminated (with microDecon) versus uncontaminated sample, and sample with contaminant OTUs removed versus uncontaminated sample. We used the same settings as simulation 1.

2.3 | Simulation 2: Groups of Samples

We used a second simulation to examine the effects of microDecon at a group level (i.e., the effects when examining multiple samples from different populations, species, environments, etc.; Appendix S5). The core code and functionality of this simulation is similar to simulation 1, but there are a few key differences. First, it simulates two groups with a user-defined number of samples per group (samples in each group are more similar to each other than to samples in the other group). Additionally, it creates variability in the amount of DNA present in each sample. The samples are then contaminated as in simulation 1, but the procedure for producing heterogeneity in the contaminated community is applied separately for each sample. Thus, there is variation in the proportions of OTUs in the contamination applied to each sample. Within each group, it returns mean BC for comparisons between the uncontaminated and contaminated samples as well as the uncontaminated and decontaminated samples. Additionally, it returns mean BC for comparisons between the groups for the uncontaminated, contaminated, and decontaminated samples. Full details on the simulation and input OTU distribution are available in Appendices S3 and S6.

We used this simulation to compare groups of 5, 10, and 20 samples each (100 iterations per group size). For each iteration, there were a total of ~500 OTUs, of which ~120 amplified in the blank (the exact numbers varied because of stochasticity in the simulation). Of the ~120 contaminant OTUs, ~30 were entirely contamination, ~30 overlapped with group 1, but not group 2, ~30 overlapped with group 2 but not group 1, and ~30 overlapped with both groups. We varied the level of contamination between groups by giving samples in group 1 an average of 2.2 times the amount of starting material as samples in group 2. As a result, the level of contamination (DNA yield in contamination/DNA yield in sample) in group 1 had a mean of 0.05 (range = 0.02–0.13) and group 2 had a mean of 0.11 (range = 0.05–0.27).

2.4 | Sequencing experiment

We constructed a sequencing experiment using fungal microbiota. We used fungal microbiomes because they are less prone to contamination than are bacterial microbiomes and eliminating unwanted background contamination was vital for this experiment. Therefore, conducting this experiment on bacteria was not possible because contamination-free bacterial samples are extremely difficult to achieve. Nevertheless, because microDecon simply uses ratios of OTUs, it is not taxa-specific, and there is no a priori reason to expect it to behave differently for different taxa. Indeed, this becomes obvious when one considers the fact that microbiome simulations do not specify the taxa, and simulated OTUs can be discussed as bacterial OTUs, fungal OTUs, protist OTUs, etc. (similarly, “OTU” can be replaced with “ASV”). Thus, given that the same methodologies are used to produce bacterial and fungal OTU tables, testing this method on fungi rather than bacteria is completely valid and does not affect the applicability of our results.

Briefly, we constructed a contaminant fungal community (consisting of cells, rather than DNA). We then collected eight soil samples: four from a forest (group 1) and four from a nearby dry streambed (group 2) and added two fungal species that we included in our contaminant community. We did this to ensure that at least a few OTUs would be present among all samples, as well as in our contamination. Next, we homogenized the samples, split them in half, and added 90 μ l of our contaminant community to one of the halves of each sample, producing both an uncontaminated and contaminated copy of each sample. For one sample from each group, we split it into thirds and only contaminated one third so that we would have replicate uncontaminated samples; unless otherwise noted, we only used the first of those two replicates in the analyses and summary statistics to avoid pseudo-replication. We also added 90 μ l of contamination to each of four empty vials. These served as our blanks and allowed us to test the assumption that the contamination ratios would be homogeneous across samples. To account for background contamination, we also analyzed a control vial that did not receive our contaminant community. This produced a total of three reads from only two OTUs; therefore, given that the actual samples consisted of thousands of reads and had been diluted to a standard concentration prior to sequencing (whereas this control sample did not have detectable levels of DNA by either gel electrophoresis or Enspire quantification), we considered that level of background contamination to be inconsequential and do not discuss it further.

We extracted the DNA from all samples using a CTAB protocol (Doyle & Doyle, 1987) modified to include a bead beating step, and, with a few exceptions, we followed the Illumina 16S Metagenomics Sequencing Library Preparation guide (“16S Metagenomic Sequencing Library Preparation,” 2017) to prepare our samples. We used the ITS3_KY02/ITS4 primer pair to amplify the ITS2 region of the fungal genome (Toju, Tanabe, Yamamoto, & Sato, 2012). Also, we used 10 μ l reactions and 30 cycles for the amplification PCR, and 40 μ l reactions for the indexing PCR. For clean-ups, we used Sera-mag SpeedBeads rather than AMPure beads. We sequenced

the samples on an Illumina Miseq (Reagent kit V3 600 cycles PE, Illumina, USA). More details of our experimental design and methods are available in Appendix S3.

After sequencing, we used PIPITS (v1.4.5) (Gweon et al., 2015) to prepare a read pairs list (*pipits_getreadpairlist*), process the reads (*pipits_prep*) using PEAR (Zhang, Kassian, Flouri, & Stamatakis, 2013), and extract the ITS region (*pipits_funits*), according to the user manual. We followed this with chimera checking (*identify_chimeric_seqs.py*) using usearch61 (Edgar, Haas, Clemente, Quince, & Knight, 2011), and de novo OTU picking (*pick_de_novo_otus.py*) in QIIME (v1.9) (Caporaso et al., 2010), using the 97% sequence similarity UNITE database (12_11, alpha release) (Abarenkov et al., 2010). In some cases, multiple OTUs were identified as the same species; therefore, we combined those OTUs for each fungal species. Sequencing results are available in Appendices S7 and S8.

Following sequencing, filtering, and annotation, we applied microDecon to the contaminated samples, producing three data sets: uncontaminated, contaminated, and decontaminated. We used the data from all four blanks to decontaminate the samples (tests comparing the effects of using multiple blanks are available in Appendix S1).

We tested the utility of microDecon in several ways. First, we used PERMANOVAs via the *adonis2()* function in the *vegan* package (Oksanen et al., 2017) to compare the uncontaminated and contaminated samples, as well as the uncontaminated and decontaminated samples (contamination status and group were factors; sample was the strata; 5,000 permutations). To avoid spurious signals from heterogeneity in OTUs that were not present in the blanks and more effectively test microDecon, we subset the data to include just the contaminant OTUs. Additionally, we examined BC both within and among groups.

To compare microDecon with existing methods, we applied two other methods to our data set. First, we used the method of detecting and removing contaminant OTUs proposed in Jervis-Bardy et al. (2015). This test and its results are available in Appendix S3. Second, we used the *decontam* R package (Davis et al., 2018). We tested both the “frequency” method (which does not require data from blanks) and the “prevalence” method (which requires blanks), as well as the built-in method for combining approaches. We ran all methods using, the default threshold (0.1), a threshold of 0.5, and a threshold that we selected separately for each method based on the approach described in Davis et al. (2018).

3 | RESULTS AND DISCUSSION

3.1 | Simulation 1: Individual samples

microDecon reduced or eliminated the contamination in 98.1% of simulated samples (out of 100,000, each with a different starting community and different contaminant community). As expected, the BC between the uncontaminated and decontaminated samples was consistently lower than the BC between the uncontaminated and contaminated samples, with the effect becoming exaggerated

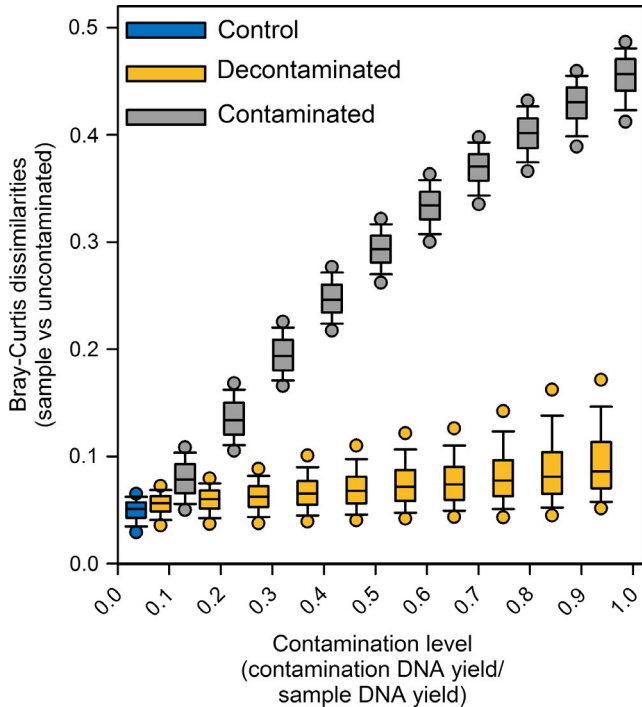


FIGURE 2 Simulation 1 results showing the ability of microDecon (“Decontaminated”) to corrected contaminated samples. Data (Bray–Curtis dissimilarity between the sample and uncontaminated copy of the sample) were grouped based on the proportion of contamination. The simulation control box is based on subsetting the data to only the OTUs that did not amplify in the blank. Whiskers represent the 90th and 10th percentile. For readability, outliers represent the 95th and fifth percentile. A total of 100,000 iterations were run, but 2,395 had contamination levels higher than 1 and are excluded (all iterations and outliers are visible in Appendix S3)

as the amount of contamination increased relative to the amount of DNA in the sample (Figure 2). This indicates that microDecon was accurately removing contamination and restoring samples to their proper OTU distributions.

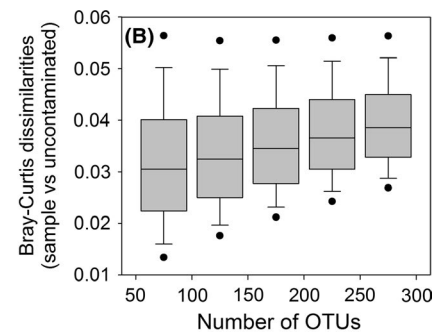
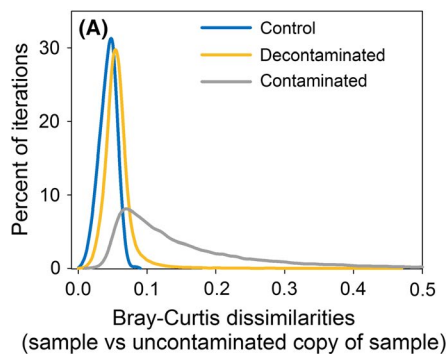


FIGURE 3 (a) Distributions of Bray–Curtis dissimilarities (BC) from 100,000 iterations of simulating individual samples. For readability, the X axis stops at 0.5, but there were 1,756 contaminated points and 20 decontaminated points greater than that (max = 0.906 and 0.712 respectively). The simulation control distribution is from the OTUs in the decontaminated sample that did not amplify in the blank. (b) Relationship between the number OTUs and the BC for the simulation controls (i.e., stochastic variation). Increasing numbers of OTUs resulted in greater dissimilarities, which were partially responsible for the slight shift in the decontaminated distribution in Figure 3a. Whiskers represent the 90th and 10th percentile, and outliers are shown as the 95th and fifth percentile

Nevertheless, because our simulations included heterogeneity from extraction and sequencing, as would occur in actual studies, we did not expect decontaminated samples to perfectly match their uncontaminated counterparts, even if they were fully decontaminated. To assess this background heterogeneity, for each decontaminated sample, we used “simulation controls” by subsetting the sample to only the OTUs that did not amplify in the blank and comparing that subset community with the corresponding OTUs in the uncontaminated sample. Because microDecon only affects the OTUs that amplified in the blank (i.e., contaminant OTUs), the OTUs that did not amplify in the blank would have been unaffected by microDecon but would have been affected by stochasticity in the simulation. Therefore, they could be used to measure the background heterogeneity.

We compared the BC frequency distribution between the simulation controls, decontaminated samples, and contaminated samples, with the expectation that the simulation controls and decontaminated samples should have similar distributions, while the contaminated samples should be shifted towards high BC. The results largely matched our predictions, suggesting that microDecon was successfully removing contamination (Figure 3a). The decontaminated distribution was shifted slightly from the simulation control distribution, but this was not unexpected, because BC increased as the number of OTUs increased (Figure 3b), and the control communities consisted of a subset of the OTUs in the decontaminated communities. Thus, the decontaminated communities always contained more OTUs and, therefore, we expected them to always have slightly higher BC.

To examine the failure rate of microDecon, we examined the number of iterations in which the decontaminated sample versus the uncontaminated sample had a higher BC than the contaminated sample versus the uncontaminated sample (i.e., cases where microDecon shifted the community further from the uncontaminated community). If microDecon was effective, then we expected that there would be few of these cases, the increases in BC should be small, and most “failures” should occur when then contamination levels

were extremely low (in terms of DNA yield), thus making them indistinguishable from stochastic fluctuations in the simulation (Figure 3). These expectations were met. Out of the 100,000 iterations, only 1,885 (1.9%) were “failures,” and those samples were characterized by low levels of contamination, resulting in low BC when either the contaminated or decontaminated samples were compared to the uncontaminated samples (Appendix S3). Additionally, the shifts in BC were generally small. For 1,019 of these samples (54.1%) the decontaminated BC were less than 0.005 BC units higher than the contaminated BC, for 1,463 (77.6%) the BC were less than 0.01 higher, and for 1,720 (91.2%) the BC were less than 0.02 higher. Only 20 iterations were off by more than 0.05.

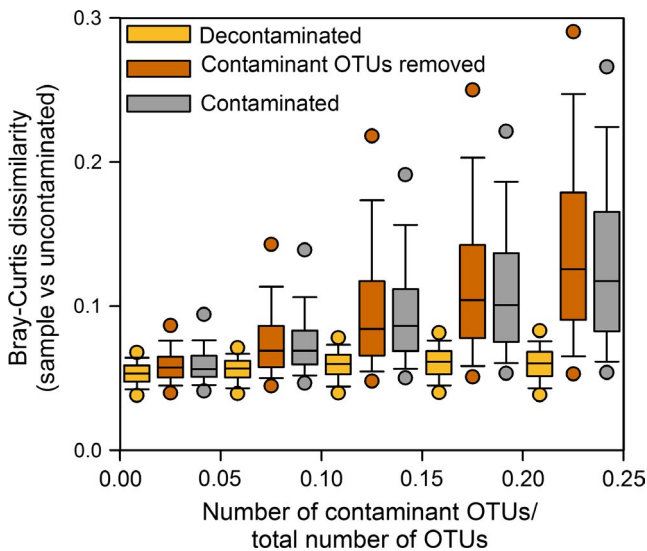


FIGURE 4 A comparison of the effectiveness of microDecon versus removing all contaminant OTUs for simulated data. Using microDecon (“Decontaminated”) was superior to either removing contaminant OTUs (“Contaminated OTUs removed”) or making no adjustments for contamination (“contamination”). Whiskers represent the 90th and 10th percentile. For readability, outliers are shown as the 95th and fifth percentile (full data in Appendix S3)

Nevertheless, a few of the iterations with higher BC do appear to be true microDecon failures and merit further discussion. These generally occurred when samples had very few OTUs that were entirely contamination (Appendix S3). Indeed 84 of the “failures” (including the worst one) had no OTUs that were entirely contamination. Given that microDecon operates by finding an OTU that is entirely contamination (the constant), it makes sense that it would struggle in situations where no OTUs are entirely contamination. Nevertheless, in the entire data set (all 100,000 iterations), there were 605 cases with no OTUs that were entirely contamination, and in every case except for these 84, microDecon still improved the results, which can be viewed as an 86.1% success rate even under the worst situation for this method. Additionally, in real microbiome studies, it is unlikely that all of the contaminant OTUs would overlap with the sample’s natural (noncontaminant) OTUs. Also, it should be stressed that these results are for individual samples. Thus, the net effect on a group may still be positive, even if one particular sample was negatively affected. Finally, these samples all used two runs of the decon() function (default), but for samples with very low contamination, the results can be improved by only using one run (see microDecon User’s Guide).

Finally, our comparison of microDecon versus the method of simply removing contaminant OTUs showed that microDecon produced more accurate results (Figure 4). As expected, the problems with simply removing contaminant OTUs became exaggerated as the proportion of OTUs that were contaminants increased, and when over roughly 20% of the OTUs were contaminants, removing them was actually worse than making no correction at all (Appendix S3). However, even when fewer than 5% of the OTUs were contaminants, applying microDecon was superior (mean BC = 0.054; *SD* = 0.01) to removing the contaminant OTUs (mean = 0.06; *SD* 0.02).

3.2 | Simulation 2: Groups of samples

If microDecon was effective, then we expected the mean BC per group to be lower for decontaminated versus uncontaminated samples than for contaminated versus uncontaminated samples

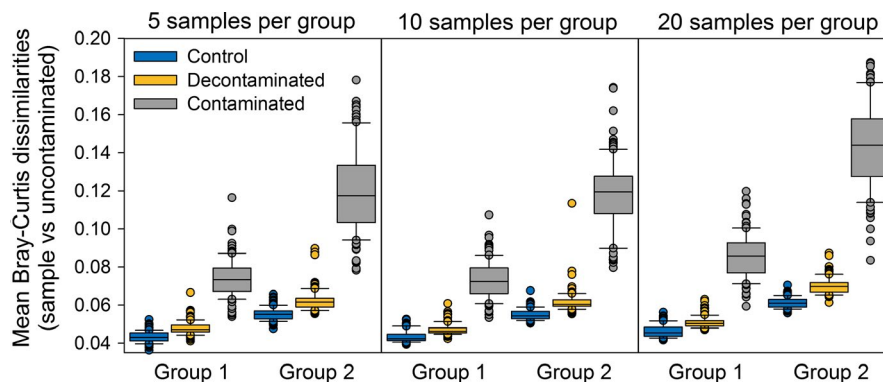


FIGURE 5 Results of simulations on entire groups (Simulation 2), showing the ability of microDecon (“Decontaminated”) to correct contaminated samples. Means are per group per iteration. For the simulation controls, comparisons were made between the decontaminated and uncontaminated samples using only the OTUs that were not in the blank (i.e., the ones unaffected by contamination and decontamination). Controls were expected to be slightly lower than decontaminated samples because they contained fewer OTUs (see Figure 3). Whiskers represent the 90th and 10th percentile, and all outliers are shown

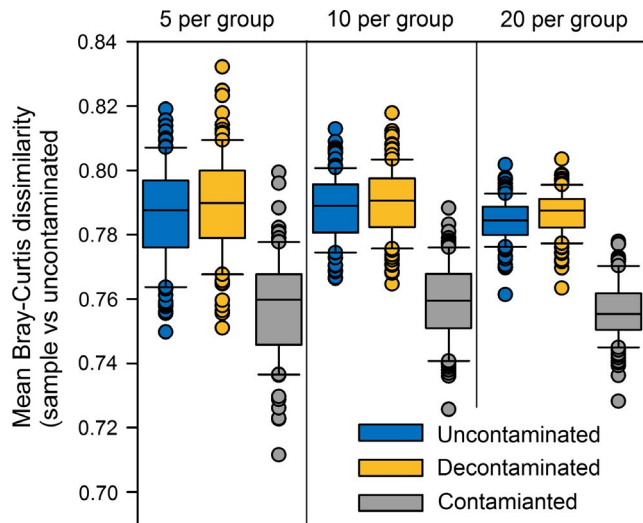


FIGURE 6 Mean Bray-Curtis dissimilarities for comparisons between groups (groups consisted of 5, 10, or 20 samples). For each iteration (100 per panel), comparisons were made between groups for the uncontaminated, decontaminated (with microDecon), and contaminated samples. Whiskers represent the 90th and 10th percentile, and all outliers are shown. Note: The Y axis should simply say "Mean Bray-Curtis dissimilarity" and not "(sample vs uncontaminated)"

(Figure 5). This prediction was met for both groups in all 300 iterations, once again demonstrating that microDecon restores samples to their correct distributions. This was particularly true for group 2, which had less than half the sample DNA of group 1 (on average).

The benefits of decontamination could also be seen when the two groups were compared within an iteration (Figure 6). Because contamination affected all samples in an extraction/sequencing run (iteration), we expected it to make samples more similar to each other, and that is what we observed. Furthermore, the decontamination procedure corrected this, and returned the groups to approximately the correct level of difference (Figure 6). We also visualized this using PCoAs (we used the `cmdscale()` function in the package `vegan`) (Figure 7a-c). Although the decontamination procedure clearly improved the samples, it did not produce BC that were quite as low as the simulation controls. As explained in the Simulation 1 section, this is at least partially an artefact caused by more OTUs being present in the decontaminated samples. We also used stacked bar plots (with each OTU as a bar) to visually examine the effects of microDecon. These visualizations further confirmed that it was successfully removing contamination and restoring communities (plots are available in Appendix S3: Figure 7).

3.3 | Sequencing experiment

The sequencing experiment provided powerful evidence that microDecon performs well under experimental conditions and accurately removes contaminant reads while retaining the reads from the actual sample. It also demonstrated the validity of our assumption that each sample would receive roughly equal ratios of contaminants.

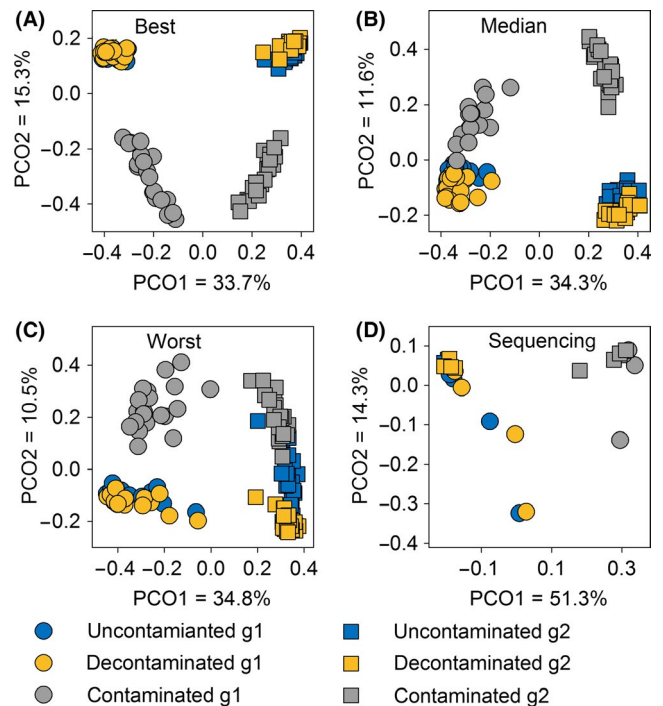


FIGURE 7 PCoAs (based on square root transformed Bray-Curtis dissimilarities [BC]) comparing groups ("g1" and "g2") for uncontaminated, decontaminated, and contaminated samples. The data were subset to the OTUs that amplified in the blank so that the effects of contamination and microDecon ("Decontaminated") could be seen more clearly. (a-c) Best, median, and worst results out of 100 iterations (judged based on mean BC between the uncontaminated and decontaminated samples for group 2). Group 2 had lower DNA yield and, therefore, was more affected by contamination. (d) Results from the sequencing experiment, showing that microDecon effectively removed the contamination

Sequencing produced a total of 1,598 OTUs; however, the majority of OTUs were not present in most samples, and on average, the uncontaminated samples contained only 361 OTUs (range = 183–511). There were 74 OTUs in the contaminated blanks (when all four blanks were averaged), 47 of which overlapped with the uncontaminated samples in group 1, and 48 of which overlapped with the uncontaminated samples in group 2. Additionally, the second most common OTU in the blank (*S. cerevisiae*; mean = 31.6% of reads in the blank) was also highly abundant in the uncontaminated samples (mean = 44.4%; range = 25.6%–69.4%), and the most abundant OTU in the blank (an unidentified fungus; mean = 33.6% of reads in the blank) was present in the uncontaminated samples at low levels (mean = 0.1%, range = 0.03%–0.38%). Finally, to obtain a proxy for contamination level, for each sample we divided the number of reads that were removed by microDecon by the number of reads in the decontaminated sample, which resulted in a mean contamination level of 0.31 (range = 0.14–0.63). This combination of high levels of contamination, and lots of OTUs that overlapped between the blank and the sample (including overlap with one of the most numerous members of each community) produced a situation approaching a worst-case scenario for microDecon.

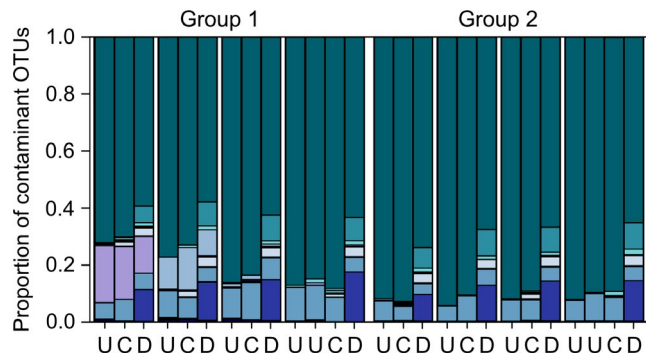


FIGURE 8 Comparison of uncontaminated (U), decontaminated (D), and contaminated (C) samples for the sequencing test. Stacked bars show the percent of each sample that was comprised by each OTU (each color/section is an OTU). Each group of 3–4 bars is a sample. The last sample in each group has a replicate uncontaminated sample. Data were subset to the OTUs that amplified in the blank (contaminant OTUs) so that trends could easily be seen. There were several prominent OTUs in the contaminated samples that were removed or greatly reduced by microDecon. Note: The "D" and "C" labels are flipped (i.e., C = decontaminated)

Therefore, this experiment should provide a useful test of the method's effectiveness.

Several tests confirmed the utility of microDecon. At the broadest scale, and as we would expect given that microDecon should be decontaminating samples and making them more similar to uncontaminated samples, there was no significant difference between the uncontaminated and decontaminated samples (PERMANOVA; pseudo- $F = 0.25$, $p = 0.939$), whereas there was a significant difference between the uncontaminated and contaminated samples (pseudo- $F = 8.14$, $p < 0.001$). These results demonstrate that contamination caused the communities to shift away from their true values, and microDecon restored them to approximately their proper (uncontaminated) distributions. For these tests, all four blanks were used, however, there was little heterogeneity among the blanks and the choice of blank had little impact on the results, thus supporting the assumption that the contamination ratios would be similar across samples (Appendix S1).

The utility of microDecon was also supported by the BC. For all eight samples, the BC was lower for the uncontaminated versus decontaminated sample than it was for the uncontaminated versus contaminated sample. This is also reflected in the PCoAs (Figure 7d) and stacked bar plot (Figure 8). Because heterogeneity in the OTUs that were not in the blank partially obscured the effects of both contamination and decontamination, subsetting the data allowed the trends to be seen more clearly, so we subset the data to just the contaminant OTUs for both visualizations. In Figure 7d, it is clear that contamination made the two groups more similar to each other and resulted in greater overlap between them, while the decontaminated results aligned closely with the uncontaminated results. Similarly, in Figure 8, there are several prominent OTUs in the contaminated samples that were completely or largely removed in the decontaminated samples. While the proportions for the OTUs that

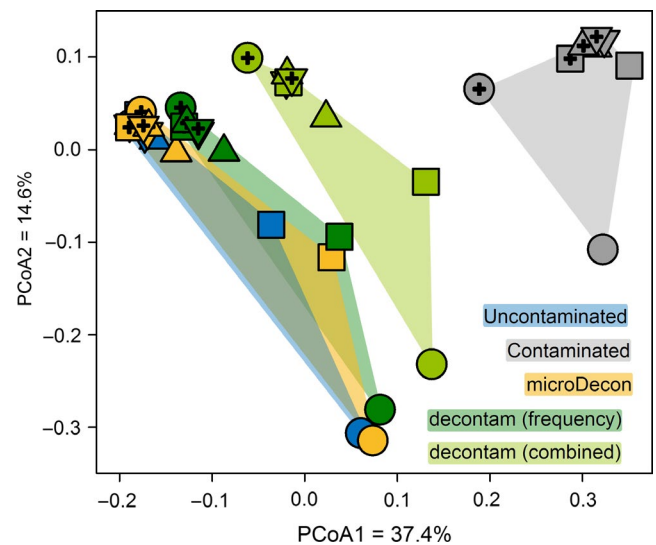


FIGURE 9 PCoA based on Bray–Curtis dissimilarities comparing uncontaminated samples, contaminated samples, and samples that were decontaminated using microDecon or decontam. Each shape is a sample and should be compared across methods. Hollow shapes are from group 1 and shapes with crosses are from group 2. Using decontam, we tested three methods with three thresholds each and have only presented the best (“frequency,” 0.5 threshold) and third best methods (“combined,” 0.5 threshold; the second best method, [“frequency,” 0.4 threshold] was nearly identical to the best method, therefore we did not visualize it here). Results of other methods were almost indistinguishable from contaminated samples, or in some cases, worse than contaminated samples. To better visualize the effects of contamination and decontamination, the data were subset to just OTUs that amplified in the blanks, but decontam also had many false positives when looking at the full data sets (see Appendix S3 for additional details and tests)

were retained in the decontaminated samples closely matched the proportions for the OTUs in the uncontaminated samples, they did not match perfectly, but that was expected because background heterogeneity causes small variations among groups, illustrated by the differences between the replicate uncontaminated samples.

Finally, in our tests, microDecon outperformed the decontam package (Figure 9; Appendix S3). When looking just at the OTUs in the blanks (to avoid heterogeneity from sequencing and to see results more clearly), contaminated samples compared to their uncontaminated counterparts had a mean BC of 0.27 ($SD = 0.04$). Samples that were decontaminated with microDecon had a mean BC of 0.04 ($SD = 0.01$) when compared to uncontaminated samples (indicating successful removal of contamination), whereas samples that were decontaminated with decontam had mean a mean BC of 0.07–0.98 ($SD = 0.01$ –0.04) depending on the method and settings used. Furthermore, decontam often had high false positive rates when looking at entire communities. These results likely occurred because microDecon has two distinct advantages over decontam. First, decontam is sensitive to the number of samples being used, and our test used only eight samples. In contrast, the core function of microDecon treats each sample separately and is not affected by sample size. Second, the most important innovation of microDecon is the ability to remove contaminant reads,

rather than entire OTUs. Thus, unlike other existing methods, it can correct OTUs that occur in both the contamination and in real samples. Additional details, discussion, *in silico* tests, and comparisons of microDecon and decontam using the Salter et al. (2014) 16S data set are presented in Appendices S3 and S9.

4 | CONCLUSION AND RECOMMENDATIONS

We have demonstrated the usefulness of the microDecon package for decontaminating samples *via* both computer simulations and a sequencing experiment, and we believe that this package will be broadly applicable across the microbiome research community. Our tests covered a wide range of situations, including low-yield samples and samples with high levels of contamination, and our method is robust to these situations. Indeed, our sequencing experiment included high contamination levels and a large overlap between the contaminant community and real community, but microDecon was still able to closely recover the real community. Therefore, we recommend that researchers use the following steps in their research.

1. Collect several blank samples at the same time and in the same manner as the actual samples are collected. These should be carried through the entire extraction process, rather than simply using no template PCR controls.
2. If possible, do all DNA extractions using a single kit and single batch of reagents. If this is not possible, then use several blanks (at least 3–4) per kit and per batch of reagents. Treat these statistically as blocks and randomize your samples across the blocks.
3. Sequence the samples and blanks, including several blanks per block. If a study involves many blocks and has insufficient sequencing depth for all of the blanks, then pool the blanks per block prior to indexing. If multiple blanks are included within a block in the final analysis, microDecon converts them to proportions and uses the mean of those proportions (see User's Guide for details).
4. Use standard filtering and bioinformatic processing steps to produce an OTU table, but do not transform, normalize, rarefy, or otherwise modify the read counts prior to using microDecon. Do not remove OTUs that are suspected to be entirely from contamination prior to running microDecon.
5. Carefully examine the blanks to ensure that they are reasonably consistent (e.g., *via* stacked bar plots and ordination plots). microDecon inherently assumes a common source of contamination. Therefore, if the contamination was from poor laboratory practices (e.g., cross-contamination among samples), the method will not be effective. If substantial differences among blanks occur only across experimental blocks, such as extraction kits (suggesting consistent contamination within a block), then use microDecon separately for each block. If, however, there is substantial variability among blanks within blocks (suggesting contamination from poor laboratory techniques), microDecon will not be effective.

6. Run microDecon (we recommend the `decon()` function on default settings).
7. Examine the OTUs in the blank and compare the contaminated and decontaminated samples to ensure that the results are reasonable for the given study system (the `decon()` and `decon.diff()` functions provide useful outputs for making these comparisons).

ACKNOWLEDGMENTS

We would like to thank the members of MEEL for their help and advice throughout this project. This work was supported by the Holsworth Wildlife Research Endowment via the Ecological Society of Australia, Skyrail Rainforest Foundation, Australian Society of Herpetologists, and Australian Wildlife Society. The environmental fungi in the sequencing experiment were collected under Queensland Department and Wildlife Protection Permit #WITK16243115 and with the approval of the James Cook University animal ethics committee (#A2209).

CONFLICT OF INTEREST

All authors affirm that they have no conflict of interest to declare.

AUTHOR CONTRIBUTIONS

DTM designed and wrote the R package and scripts and led the analyses and writing. DTM and RH conducted the sequencing experiment. RH, DSB, LS, RAA, and KRZ supervised the project, including providing input and advice for the design of the project and analysis of the data. All authors edited, read, and approved the final manuscript.

DATA AVAILABILITY STATEMENT

All data and simulation scripts are included in this article and its Appendices. The microDecon package and user manual are available from github (<https://github.com/donaldtmcknight/microDecon>).

ORCID

Donald T. McKnight  <https://orcid.org/0000-0001-8543-098X>

Roger Huerlimann  <https://orcid.org/0000-0002-6020-334X>

Lin Schwarzkopf  <https://orcid.org/0000-0002-1009-670X>

REFERENCES

- 16S Metagenomic Sequencing Library Preparation. (2017). Illumina.
- Abarenkov, K., Henrik Nilsson, R., Larsson, K.-H., Alexander, I. J., Eberhardt, U., Erland, S., ... Kõljalg, U. (2010). The UNITE database for molecular identification of fungi—recent updates and future perspectives. *New Phytologist*, 186(2), 281–285. <https://doi.org/10.1111/j.1469-8137.2009.03160.x>

- Barton, H. A., Taylor, N. M., Lubbers, B. R., & Pemberton, A. C. (2006). DNA extraction from low-biomass carbonate rock: An improved method with reduced contamination and the low-biomass contaminant database. *Journal of Microbiological Methods*, *66*, 21–31. <https://doi.org/10.1016/j.mimet.2005.10.005>
- Brooks, J. P. (2016). Challenges for case-control studies with microbiome data. *Annals of Epidemiology*, *26*(5), 336–341. <https://doi.org/10.1016/j.annepidem.2016.03.009>
- Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., ... Knight, R. (2010). QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*, *7*, 335–336. <https://doi.org/10.1038/nmeth.f.303>
- Champlot, S., Berthelot, C., Pruvost, M., Andrew Bennett, E., Grange, T., & Geigl, E. M. (2010). An efficient multistrategy DNA decontamination procedure of PCR reagents for hypersensitive PCR applications. *PLoS ONE*, *5*(9), e13042. <https://doi.org/10.1371/journal.pone.0013042>
- Corless, C. E., Guiver, M., Borrow, R., Edwards-Jones, V., Kaczmarek, E. B., & Fox, A. J. (2000). Contamination and sensitivity issues with a real-time universal 16S rRNA PCR. *Journal of Clinical Microbiology*, *38*, 1747–1752.
- Davis, N. M., Proctor, D. M., Holmes, S. P., Relman, D. A., & Callahan, B. J. (2018). Simple statistical identification and removal of contaminant sequences in marker-gene and metagenomics data. *Microbiome*, *6*, 226. <https://doi.org/10.1186/s40168-018-0605-2>
- Doyle, J. J., & Doyle, J. L. (1987). A rapid procedure for DNA purification from small quantities of fresh leaf tissue. *Phytochemical Bulletin*, *19*, 11–15.
- Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C., & Knight, R. (2011). UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics*, *27*, 2197–2200. <https://doi.org/10.1093/bioinformatics/btr381>
- Gweon, H. S., Oliver, A., Taylor, J., Booth, T., Gibbs, M., Read, D. S., ... Schonrogge, K. (2015). PIPITS: An automated pipeline for analyses of fungal internal transcribed spacer sequences from the Illumina sequencing platform. *Methods in Ecology and Evolution*, *6*(8), 973–980. <https://doi.org/10.1111/2041-210X.12399>
- Hang, J., Desai, V., Zavaljevski, N., Yang, Y. U., Lin, X., Satya, R., ... Kuschner, R. A. (2014). 16S rRNA gene pyrosequencing of reference and clinical samples and investigation of the temperature stability of microbiome profiles. *Microbiome*, *2*(1), 31. <https://doi.org/10.1186/2049-2618-2-31>
- Jervis-Bardy, J., Leong, L. E. X., Marri, S., Smith, R. J., Choo, J. M., Smith-Vaughan, H. C., ... Marsh, R. L. (2015). Deriving accurate microbiota profiles from human samples with low bacterial content through post-sequencing processing of Illumina MiSeq data. *Microbiome*, *3*(1), 19. <https://doi.org/10.1186/s40168-015-0083-8>
- Kulakov, L. A., McAlister, M. B., Ogden, K. L., Larkin, M. J., & O'Hanlon, J. F. (2002). Analysis of bacteria contaminating ultrapure water in industrial systems. *Applied and Environmental Microbiology*, *68*, 1548–1555. <https://doi.org/10.1128/AEM.68.4.1548-1555.2002>
- McKnight, D. T., Huerlimann, R., Bower, D. S., Schwarzkopf, L., Alford, R. A., & Zenger, K. R. (2018). Methods for normalizing microbiome data: An ecological perspective. *Methods in Ecology and Evolution*, <https://doi.org/10.1111/2041-210X.13115>
- Mohammadi, T., Reesink, H. W., Vandenbroucke-Grauls, C. M. J. E., & Savelkoul, P. H. M. (2005). Removal of contaminating DNA from commercial nucleic acid extraction kit reagents. *Journal of Microbiological Methods*, *61*(2), 285–288. <https://doi.org/10.1016/j.mimet.2004.11.018>
- Oksanen, J. F., Blanchet, F. G., Friendly, M., Kindt, R., Legendre, P., McGlenn, D., ... Wagner, H. (2017). *vegan: Community ecology package*. Retrieved from <https://cran.r-project.org/package=vegan>.
- Peters, R. P. H., Mohammadi, T., Vandenbroucke-Grauls, C. M. J. E., Danner, S. A., van Agtmael, M. A., & Savelkoul, P. H. M. (2004). Detection of bacterial DNA in blood samples from febrile patients: Underestimated infection or emerging contamination? *FEMS Pathogens and Disease*, *42*, 249–253.
- R Core Team (2017). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.
- Robinson, K. M., Crabtree, J., Mattick, J. S. A., Anderson, K. E., & Dunning Hotopp, J. C. (2017). Distinguishing potential bacteria-tumor associations from contamination in a secondary data analysis of public cancer genome sequence data. *Microbiome*, *5*(1), 9. <https://doi.org/10.1186/s40168-016-0224-8>
- Rueckert, A., & Morgan, H. W. (2007). Removal of contaminating DNA from polymerase chain reaction using ethidium monoazide. *Journal of Microbiological Methods*, *68*, 596–600. <https://doi.org/10.1016/j.mimet.2006.11.006>
- Salter, S. J., Cox, M. J., Turek, E. M., Calus, S. T., Cookson, W. O., Moffatt, M. F., ... Walker, A. W. (2014). Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biology*, *12*(1), 87. <https://doi.org/10.1186/s12915-014-0087-z>
- Segal, L. N., Alekseyenko, A. V., Clemente, J. C., Kulkarni, R., Wu, B., Chen, H., ... Weiden, M. D. (2013). Enrichment of lung microbiome with supraglottic taxa is associated with increased pulmonary inflammation. *Microbiome*, *1*, 19. <https://doi.org/10.1186/2049-2618-1-19>
- Shen, H., Rogelj, S., & Kieft, T. L. (2006). Sensitive, real-time PCR detects low-levels of contamination by *Legionella pneumophila* in commercial reagents. *Molecular and Cellular Probes*, *20*, 147–153. <https://doi.org/10.1016/j.mcp.2005.09.007>
- Toju, H., Tanabe, A. S., Yamamoto, S., & Sato, H. (2012). High-coverage ITS primers for the DNA-based identification of ascomycetes and basidiomycetes in environmental samples. *PLoS ONE*, *7*(7), e40863. <https://doi.org/10.1371/journal.pone.0040863>
- Weiss, S., Amir, A., Hyde, E. R., Metcalf, J. L., Song, S. J., & Knight, R. (2014). Tracking down the sources of experimental contamination in microbiome studies. *Genome Biology*, *15*(12), 564. <https://doi.org/10.1186/s13059-014-0564-2>
- Wilner, D., Daly, J., Whiley, D., Grimwood, K., Wainwright, C. E., & Hugenholtz, P. (2013). Comparison of DNA extraction methods for microbial community profiling with an application to pediatric bronchoalveolar lavage samples. *PLoS ONE*, *7*, e34605.
- Zhang, J., Kassian, K., Flouri, T., & Stamatakis, A. (2013). PEAR: A fast and accurate Illumina paired-end read merger. *Bioinformatics*, *30*, 614–620. <https://doi.org/10.1093/bioinformatics/btt593>

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: McKnight DT, Huerlimann R, Bower DS, Schwarzkopf L, Alford RA, Zenger KR. microDecon: A highly accurate read-subtraction tool for the post-sequencing removal of contamination in metabarcoding studies. *Environmental DNA*. 2019;1:14–25. <https://doi.org/10.1002/edn3.11>