Chapter 10

PURC Provides Improved Sequence Inference for Polyploid Phylogenetics and Other Manifestations of the Multiple-Copy Problem

Peter Schafran, Fay-Wei Li, and Carl J. Rothfels

Abstract

Inferring the true biological sequences from amplicon mixtures remains a difficult bioinformatics problem. The traditional approach is to cluster sequencing reads by similarity thresholds and treat the consensus sequence of each cluster as an "operational taxonomic unit" (OTU). Recently, this approach has been improved by model-based methods that correct PCR and sequencing errors in order to infer "amplicon sequence variants" (ASVs). To date, ASV approaches have been used primarily in metagenomics, but they are also useful for determining homeologs in polyploid organisms. To facilitate the usage of ASV methods among polyploidy researchers, we incorporated ASV inference alongside OTU clustering in PURC v2.0, a major update to `PURC` (Pipeline for Untangling Reticulate Complexes). In addition, PURC v2.0 features faster demultiplexing than the original version and has been updated to be compatible with `Python 3`. In this chapter we present results indicating that using the ASV approach is more likely to infer the correct biological sequences in comparison to the earlier OTU-based `PURC` and describe how to prepare sequencing data, run PURC v2.0 under several different modes, and interpret the output.

Key words Allopolyploidy, Amplicon sequence variants, Amplicon sequencing, Moderate data, OTU inference, PacBio, Polyploid phylogenetics, Reticulate evolution

1 Overview

1.1 The Multiple-Copy Problem

There are many situations in biology in which an individual organism may harbor multiple closely related gene copies and where knowing the number of copies and their individual sequences is important for downstream inferences. For example, a major system of self-incompatibility in plants (e.g., the ability of a plant to reject fertilization attempts from its own pollen and thus avoid mating with itself) is based on the presence of particular combinations of "S-alleles," and the characterization of these highly polymorphic S-alleles across individuals is thus essential for understanding the mechanism and consequences of this form of self-incompatibility

Yves Van de Peer (ed.), *Polyploidy: Methods and Protocols*,
Methods in Molecular Biology, vol. 2545, https://doi.org/10.1007/978-1-0716-2561-3_10,

[1, 2]. Similarly, many applications based on the multispecies coalescent, such as delimiting species with BPP [3, 4], rely on including the sequences from both alleles present in a diploid individual. This "multiple-copy problem" [5], however, is particularly pronounced in phylogenetic studies of polyploids. Because many polyploids unite subgenomes from different progenitor species (they are hybrid polyploids, or "allopolyploids"), their true evolutionary history is a network rather than a strictly bifurcating tree, and to infer this reticulate history, the homeologous sequences—the copies from each of the subgenomes—need to be recovered and reconstructed accurately [6].

The classic way of recovering multiple gene copies from a single individual is molecular cloning: the desired marker is amplified by PCR, the amplicon is cloned into plasmid vectors that are used to transform *Escherichia coli*, and multiple colonies of the transformed bacteria are re-amplified and sequenced [e.g. 7, 8]. This approach is labor-intensive and expensive (for a triploid, for example, one would need to sequence 11 colonies to have a 95% chance of getting all three copies), making datasets with many samples or many loci impractical. Short-read next-generation sequencers, such as those of the Illumina platform, offer some relief, in that the reads come from individual molecules (rather than representing a form of majority-rule consensus of the molecules present, as is the case with Sanger sequencing), and sequencing costs are dramatically reduced. However, in order to recover the individual copies, the reads need to be assembled accurately. This assembly step can be bioinformatically prohibitive, especially if there are more than two copies present, and always faces a hard limit: to correctly assemble the full length of each copy of a target sequence, consecutive variable sites have to be separated from each other by no longer than the read length.

1.2 The PURC Approach

To help facilitate the recovery of all homeologous sequences from polyploid accessions (and for other manifestations of the multiple-copy problem), [9] published a molecular lab workflow based on PacBio long-read sequencing, with an associated bioinformatics pipeline (the "Pipeline for Untangling Reticulate Complexes" [PURC]) to infer the individual copies from the PacBio reads. This approach capitalized on PacBio's circular consensus sequencing (CCS) technology to generate contiguous reads for long (>1000 bp) phylogenetically informative regions, thus avoiding the need for assembly and allowing for the accurate retrieval of all copies present in individual polyploids [9–11]. A single PacBio SMRT cell can generate sequences for multiple loci for hundreds of samples, providing an economical means to generate powerful "moderate data" datasets for polyploid phylogenetics.

The wetlab workflow involved standard PCR with barcoded forward primers; the amplicons are then pooled, roughly standardized by amplicon concentration and the anticipated number of copies present in each accession, and sequenced on a single PacBio SMRT cell. PURC can demultiplex the resulting reads by locus, barcode, and phylogenetic affinity (i.e., the same barcode can be used multiple times in a single sequencing run, once for each phylogenetically discernable group, such as a genus). By taking an iterative chimera-removal and clustering approach [12, 13], PURC then infers the underlying biological sequences. The output from PURC is one alignment for each locus, which includes all the copies present in each of the accessions, labelled by their accession ID and coverage (the number of reads that constitute that sequence). Since its release, PURC has been used to investigate allopolyploidy in genus-level datasets in diverse plant lineages [10, 14–16], in studies of cytotype variation within species [11], and for applications where polyploidy itself was incidental to the primary research questions [17–20]. Most notably, Blischak et al. [21] created a program to adapt PURC to data generated by microfluidic PCR, reducing one of the main limitations of amplicon-based approaches (the time and expense associated with PCR itself).

1.3 PURC v2.0

Since the release of PURC, several studies have demonstrated shortcomings of OTU clustering, especially a tendency to overestimate the number of sequences, difficulty in determining appropriate similarity thresholds, and inability to replicate and compare OTUs between analyses [22–25], with the overestimation problem reported for PURC specifically [14, 21]. An alternative approach to identify and separate PCR and sequencing errors from biological sequences is to apply an error model, where read abundance, composition, and quality scores are used to infer whether each unique read is likely to have been derived from another observed sequence [26]. Reads inferred to represent biological sequences by these methods are called amplicon sequence variants (ASVs), exact sequence variants, or zero-radius OTUs. DADA2 [26] is one of the most popular software packages for inferring ASVs and is particularly flexible because it incorporates separate error models for Illumina and PacBio CCS data. Additionally, DADA2 can take sequences as "priors," increasing the sensitivity of the algorithm for sequences that are similar to the priors (https://benjjneb.github.io/dada2/pseudo.html).

In order to take advantage of potential improvements of ASVs, PURC v2.0 incorporates DADA2 alongside Vsearch [27], an open-source alternative to Usearch [12], allowing PURC v2.0 to run in four different ways:

1. OTU clustering alone
2. ASV inference alone
3. OTU clustering and ASV inference in one run
4. OTU clustering followed by ASV inference with the OTUs as priors

To test our incorporation of DADA2, we reproduced OTUs and ASVs generated by [25], where a mock community of five cyanobacteria was created by combining equal quantities of *rbcL-X* amplicons generated from pure cultures. PURC v2.0 was run on four replicates of the five-taxon mock community, performing OTU clustering with default parameter values (clustering thresholds by round: 0.997, 0.995, 0.990, and 0.997; final size threshold = 4) and inferring ASVs (DADA2) with a maximum of five expected errors allowed per read and length outliers removed based on Tukey's outlier test [28]. We also tested data from species of *Isoetes* from [15] using the same parameters.

Our results agree with [25] in showing that ASV inference was more accurate than OTU clustering (Fig. 1). The same five ASVs were recovered from every replicate, whereas OTU clustering identified six to nine OTUs per replicate. Most replicates contained OTUs identical to ASVs, but OTUs with small sequence deviations were common and a few spurious OTUs with 75%–92% identity to ASVs were also generated. Because the *Isoetes* data from [15] are

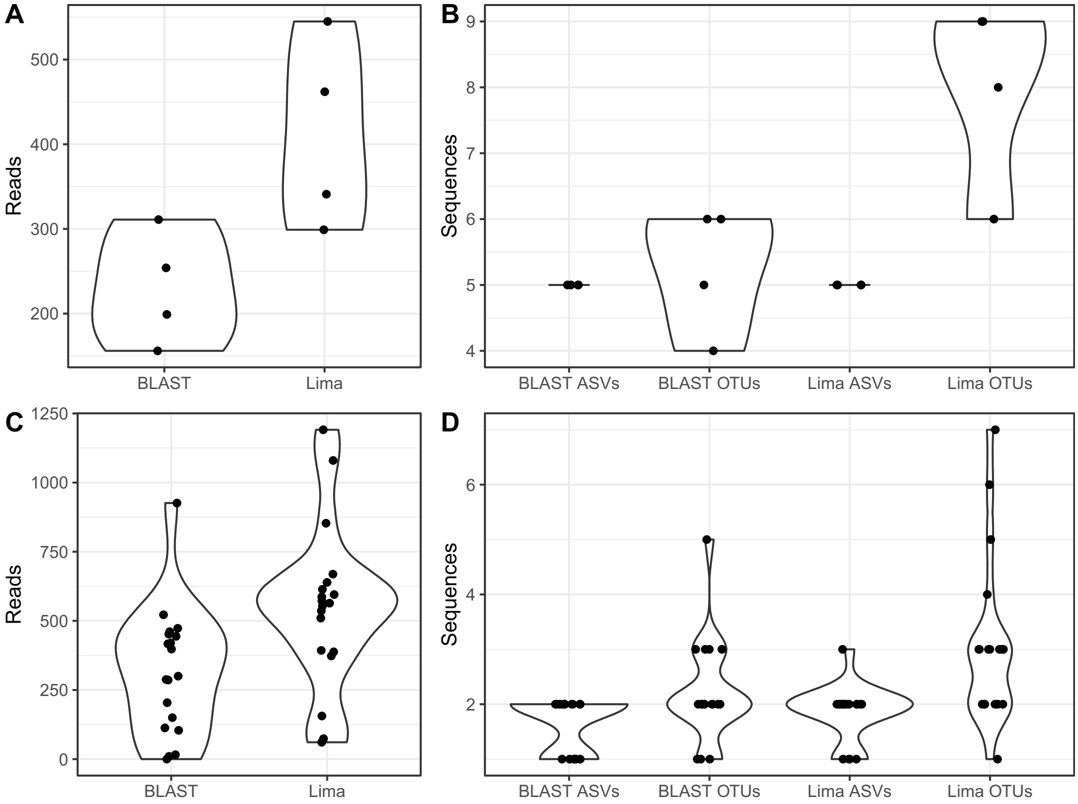

Fig. 1 Comparison of demultiplexing and sequence inference methods using data from [25] and [15]. (**a**) Number of reads assigned to four replicates of the five-taxon mock community. (**b**) Sequences inferred for the four replicates of the five-taxon mock community. (**c**) Number of reads assigned to each *Isoetes* sample. (**d**) Sequences inferred for each *Isoetes* sample

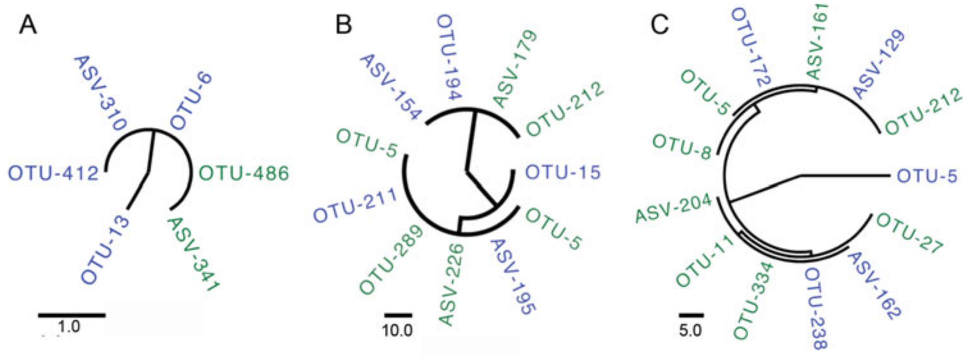

Fig. 2 Maximum-parsimony trees of OTUs and ASVs from two PCR replicates of three *Isoetes* species ((**a**) diploid *I. echinospora* Taylor #6989-1; (**b**) allotetraploid *I. maritima* Taylor #6983; (**c**) allotetraploid *I.* aff. "new hybrid A" Taylor #6988-2). PCR replicate 1 is colored green and replicate 2 is colored blue. Numbers following tip names indicate coverage (the number of reads constituting each OTU/ASV). Trees are rooted at their midpoints. Scale bar represents number of substitutions. (Data are from [15])

empirical and true sequences are unknown, we evaluated ASV- versus OTU-based inferences of these sequences using two independent PCR amplifications of each of three individuals, thought to represent one diploid and two allotetraploids (*I. echinospora* Taylor #6989-1, *I. maritima* Taylor #6983, and *I.* aff. "new hybrid A" Taylor #6988-2, respectively). For these accessions we inferred 20 OTUs versus 10 ASVs in total. Both PCR replicates yielded identical ASV inferences—one sequence for the diploid and two for each of the allotetraploids (Fig. 2)—and these sequences all had high coverage (they represented many individual reads; Fig. 3). However, OTU inference was less consistent. For the diploid, both PCR replicates resulted in a high-coverage OTU that was identical to the one from ASV inference, but one of the replicates inferred two additional low-coverage OTUs (for example, OTU-6 contained five deletions and two ambiguous nucleotides relative to OTU-412 and ASV-310; Fig. 2a.) Similarly, for the tetraploids, for both PCR replicates, OTU inference found the sequences corresponding to the ASVs, but also found additional, presumably spurious, low-coverage sequences (Figs. 2 and 3). One of these sequences—OTU-5 in replicate 2 of the allotetraploid *I.* aff. "new hybrid A" Taylor #6988-2—was uniquely divergent, with 36 SNPs and two deletions totaling 97 bases. The origins of the five reads that constitute this OTU are unclear, but were not due to low-quality basecalls or incorrect assignment by barcodes. In every case, the identical ASVs and OTUs were by far the most abundant (Fig. 3); they likely represent the true biological sequences.

PURC v2.0 also includes PacBio's lima tool (https://lima.how) as a new method for demultiplexing reads on Linux operating systems. Using lima, we could recover an average of 74% more reads

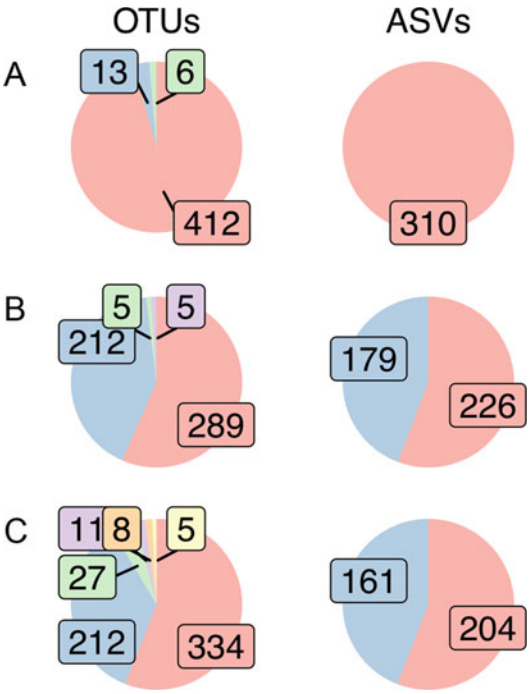

Fig. 3 Proportions of reads contributing to each OTU and ASV for three *Isoetes* species ((**a**) diploid *I. echinospora* Taylor #6989-1; (**b**) allotetraploid *I. maritima* Taylor #6983; (**c**) allotetraploid *I. aff.* "new hybrid A" Taylor #6988-2), from one PCR replicate. Labels indicate number of reads in each OTU/ASV and sequences are colored from largest to smallest within each chart. (Data are from [15])

from the [15] data than with the original BLAST-based method in PURC, including recovering one individual for which no sequences were identified using BLAST; the per-sample increase ranged from 24% to 640% (Fig. 1c). The number of both OTUs and ASVs increased in the lima-demultiplexed dataset but ASVs fluctuated less, with 74% of samples having the same number of ASVs as in the BLAST-demultiplexed data versus only 26% for OTUs.

There were some consistent trends in our analyses of mock community and real polyploid data. lima consistently recovered more reads from each sample, which resulted in an increase in the number of OTUs, possibly by inclusion of more divergent reads rising above the threshold for dropping low-abundance clusters. However, the inclusion of the additional reads had a much smaller effect on ASV inference. The variance in the number of ASVs was always smaller than that for OTUs and especially in the mock community analyses where every replicate was correctly inferred (Fig. 1). While [25] showed that by varying OTU clustering parameters they could more accurately reconstruct the true sample composition, this approach is unreliable in cases where the true sequences are unknown and where it may be tempting to change parameters until the results match expectations. Based on our results, we recommend lima demultiplexing and ASV inference as the primary method for running PURC v2.0. To compare OTU clustering and ASV inference on your own data, PURC v2.0 allows generating both simultaneously, with summary files to examine the number and abundance of sequences for each sample. If it appears that too few ASVs are found, a new analysis can be run with the OTU sequences as priors, increasing the sensitivity of the algorithm and reducing the detection limit for variants. This approach may be particularly useful for samples with little data or where two biological sequences are very similar, although it remains to be tested in polyploids.

2 Materials

2.1 Hardware

A personal computer with a multicore processor and at least 8-GB RAM, though this will vary based on the input data size.

2.2 Software

1. A Unix- or Linux-based operating system (e.g., macOS, Ubuntu), or on Windows, a Linux virtual machine can be used.
2. Conda package manager (https://docs.conda.io).

3 Methods

The following instructions as well as additional information for troubleshooting can be found in the main PURC v2.0 repository at https://bitbucket.org/peter_schafran/purc/.

3.1 Installing PURC v2.0

The most up-to-date version of PURC can be downloaded as a compressed TAR file in the main repository. When uncompressed, a new directory named `purc` will contain executable and installation files.

```
curl -L https://bitbucket.org/peter_schafran/purc/raw/master/
purc_v2.tar.gz -O
tar -xzf purc_v2.tar.gz
cd purc
```

We recommend installing dependencies through Conda. Two YAML files, one for macOS and one for Linux, are included in the repository and can be used to create a new PURC environment containing all necessary dependencies with one of these commands:

```
# macOS
conda env create -n purc --file purc_macos.yaml
conda activate purc
```

```
# Linux
conda env create -n purc --file purc_linux.yaml
conda activate purc
```

3.2 Preparing Input Files

Sequence data are expected to be 99% accurate amplicon sequences containing sample-specific nucleotide sequences (barcodes) on one or both ends of each read, as well as the priming sites used to generate the amplicons. If using PacBio reads generated by their standard protocol (https://ccs.how), they should not need any modification. Either FASTA or FASTQ formatted data are accepted, though FASTQ is required to perform ASV inference. In addition to the data, PURC v2.0 requires four files:

1. Barcode file listing the barcode sequences
2. Reference sequences file, so that reads can be oriented, assigned to the correct locus, and sorted into phylogenetic groups if individual barcodes are used for multiple accessions
3. Map file(s), which link barcode and group IDs to unique accessions (one map file for each locus)
4. Configuration file, which includes information on specific settings, the primer sequences, and other necessary information

3.2.1 Barcode File

Barcode sequences are provided in FASTA format. The sequence ID is the barcode name, and sequences should be oriented in the 5’-3’ direction. For example:

```
>BC01
ACTACATATGAGATGA
>BC02
TCATGAGTCGACACTA
>BC03
TATCTATCGTATACGC
```

If using the dual barcode function, the barcode names must be BCF1, BCF2, ... and BCR1, BCR2, ... for barcodes on forward and reverse primers, respectively. This information is used to check for barcodes found in the incorrect orientation. We recommend barcodes that are unique (including reverse complements), especially if using the option to demultiplex with lima. However, if a subset of duplicate barcodes is detected, PURC v2.0 will attempt to run separate analyses through lima with the unique and duplicate barcodes and then merge the output.

3.2.2 Reference Sequence File

Reference sequences must be provided in FASTA format. Each sequence should specify its locus name in the sequence ID (e.g., locus = ApP), even if the data being analyzed represent only one locus. A group name (e.g., group = A) can be provided to demultiplex by BLAST comparison to reference sequences if multiple samples share barcodes. A taxon name (e.g., ref_taxon = Cystopteris_bulbifera) or other descriptor can be included to provide additional information for the user but this information is not used by PURC. Fields are separated by forward slashes, so a complete sequence ID line may look like the following:

```
>locus=ApP/group=A/ref_taxon=Cystopteris_bulbifera
```

Note that the total sequence ID should not exceed 50 characters or else BLAST database construction will fail. All reference sequences for a locus must be in the same orientation to allow proper orientation of the reads.

3.2.3 Map File

The map file indicates which barcodes and/or groups correspond with each sample. This file is a tab-delimited text file with one of three configurations:

1. If each sample contains just one barcode (e.g., only the forward primer is barcoded) and each barcode is unique to one sample, the first column contains barcode names (from the barcode FASTA file) and the second column contains the name of the corresponding sample.

```
BC01 Cystopteris_fragilis_Utah
BC02 Cystopteris_fragilis_Arizona
BC03 Cystopteris_fragilis_Taiwan
```

2. If each sample contains one barcode, but individual barcodes are used for multiple samples, then the first column contains the barcode name, the second column contains a group ID (corresponding to a group specified in a reference sequence ID), and the third column contains the sample name.

```
BC01 A Acystopteris_japonica_Taiwan
BC01 B Gymnocarpium_dryopteris_Ontario
BC02 A Acystopteris_japonica_Japan
```

3. If samples are dual-barcoded, the first column contains the forward primer barcode name, the second column contains the reverse primer barcode name, and the third column the sample name.

```
BCF1 BCR1 Cystopteris_fragilis_Utah
BCF2 BCR1 Cystopteris_fragilis_Arizona
BCF1 BCR2 Cystopteris_fragilis_Taiwan
```

3.2.4 Config File

All PURC v2.0 operation is controlled via the config file, a text file containing information such as file names and run parameters. Paths to the data file, barcode file, reference sequence file, and map files are specified here, as well as primer sequences, run options, and optional parameter settings. Each line item is described by comments in the file (not shown here).

The first section of the config file specifies input files for reads, barcodes, and references and names of the prefix appended to output files, the output folder, and the log file.

```
[Files]
Input_sequence_file = 1-PacBio_seq.fastq
in_Barcode_seq_file = 2-barcodes.fasta
in_RefSeq_seq_file = 3-ref_sequences.fasta
Output_prefix = purc_run
Output_folder = purc_out
Log_file = log
```

The second section provides information about the locus or loci to be processed. The locus name must match that provided in the reference sequence file. If processing multiple loci, their order in `Locus_name` and `Locus-barcode-taxon_map` must match.

```
[Loci]
Locus_name = ApP, GAP
Locus-barcode-taxon_map = 4-map_APP.txt, 4-map_GAP.txt
```

The third section is used to provide primer sequences in the 5’-3’ direction: these, too, must be in the same order as, e.g., `Locus_name`. IUPAC ambiguous nucleotide codes are accepted.

```
[Primers]
Forward_primer  =  GGACCTGGSCTYGCTGARGAGTG,  TCTGCMCATGCMATT-
GAAAGAGAG
Reverse_primer  =  GGAAGVACCTTYCCTACTGCCTG,  TAGCTGCTCRAATTC-
CATKSAT
```

The fourth section allows the user to choose between several run modes depending on their data.

- `Mode` controls whether PURC v2.0 checks for interlocus concatemers or not (`Mode` should be set to `1` for single-locus data).
- `Multiplex_per_barcode` indicates if barcodes are reused for multiple individuals within the same locus or if each barcode (for a given locus) is unique to one sample (barcode reuse is not available for dual barcoding).

- Dual_barcode is used to specify how the barcodes are arranged on each read. Note that options 1 and 2 are only treated differently if using lima to demultiplex.
- Barcode_detection describes where to look for barcodes in each read. If set to 0, the "ErrMidBC" flag is included in the sequence name if barcodes are not at the ends of the sequence. This setting has no effect if using lima.
- Recycle_chimeric_seq controls whether PURC v2.0 splits interlocus chimeras into their respective loci and includes them in downstream analysis, or discards such chimeras. This setting has no effect if Mode = 1.
- Recycle_no_barcoded_seq allows for the use of Smith-Waterman local alignment [29] to try to identify barcodes in any sequences that failed to have a significant BLAST match. This setting has no effect if using lima to demultiplex.
- Clustering_method specifies which method to use for sequence inference. Option 2 does ASV inference and OTU clustering together and is required for Use_OTU_priors = TRUE.
- Align controls whether each final sequence file produced (one per locus per clustering method) is aligned using MAFFT [30].

For example:

```
[PPP_Configuration]
Mode = 0 # 0: Check concatemers and then full run
 # 1: Skip concatemer-checking
Multiplex_per_barcode = 0 # 0: Each barcode contains only one
sample
 # 1: Each barcode contains multiple samples
Dual_barcode = 0 # 0: Barcodes only on one primer
 # 1: Unique barcodes on both primers
 # 2: Same barcode on both primers
Barcode_detection = 1 # 0: Search barcode in entire sequences
 # 1: Search barcode only at the ends of sequences
Recycle_chimeric_seq = 0 # 0: Do not recycle
 # 1: Split chimeric sequences into respective locus
Recycle_no_barcoded_seq = 0 # 0: Do not recycle
 # 1: Use Smith-Waterman algorithm to find barcodes if BLAST
fails
Clustering_method = 0 # 0: Use DADA2 ASV inference
 # 1: Use Vsearch OTU clustering
 # 2: Use both clustering methods
Align = 1 # 0: No aligning attempted.
 # 1: Final consensus sequences will be aligned with MAFFT
```

Other optional parameters follow this section of the config file, including those to change the operation of DADA2 and Vsearch.

DADA2 options include specifying minimum and maximum lengths and maximum number of expected errors for reads to be included and whether to include each sample's OTU sequences as priors. If minimum and/or maximum read length is set to 0, that parameter is calculated using Tukey's equation for outliers [28]:

$$minLen = Q_1 - 1.5(Q_3 - Q_1), maxLen = Q_3 + 1.5(Q_3 - Q_1)$$

where Q_1 is the first quartile and Q_3 the third quartile, and results are rounded to the nearest integer. Outliers are recalculated for each sample. If minLen/maxLen values are user-supplied, they are applied globally. The maximum number of expected errors is estimated by DADA2 based on read quality scores. If putative spurious ASVs are produced, this number can be reduced, while if too many reads are discarded during filtering, it can be increased.

- minLen is the minimum length for a read to be included in analysis. Set to 0 to automatically detect short outliers.
- maxLen is the maximum length for a read to be included in analysis. Set to 0 to automatically detect long outliers.
- maxEE is the maximum number of expected errors estimated by DADA2 based on read quality scores. Reads with more expected errors than maxEE are discarded.
- Use_OTU_priors determines whether to use the OTU sequences as priors. When activated, the OTUs for each sample are used to infer ASVs for that sample. This can be useful if the number of reads per sample is low, or if well-supported OTUs seem to be lost during ASV inference. However, it can increase the risk of creating spurious ASVs. Requires Clustering_method = 2.

For example:

```
[DADA Filtering Parameters]
minLen = 0 # The minimum length to keep a read
maxLen = 0 # The maximum length to keep a read
maxEE = 5 # Reads with greater than maxEE "expected errors" are
discarded
Use_OTU_priors = FALSE # Set to TRUE to use OTU output
sequences as priors for ASV inference
```

Available Vsearch options are for the identity levels for each round of clustering, minimum size to retain a cluster, and the abundance skew for detecting chimeric sequences.

- clustIDn specifies the identity threshold for clustering reads during the *n*th round of clustering. For example, 0.997 means a read must have 99.7% identity to a cluster's centroid sequence in order to be included in that cluster.

- sizeThreshold1 sets the minimum number of reads in a cluster for it to be retained for the next round of OTU clustering.
- sizeThreshold2 is the same as sizeThreshold1, but is only applied to the final clustering output.
- abundance_skew is the minimum ratio of parent sequences to putative chimeric sequence required to classify a sequence as chimeric. Parent sequences are expected to be at least twice as abundant as their chimeras.

For example:

```
[Clustering Parameters]
clustID1 = 0.997 # The similarity criterion for the initial
VSEARCH clustering
clustID2 = 0.995 # The similarity criterion for the second
clustering
clustID3 = 0.990 # The similarity criterion for the third
clustering
clustID4 = 0.997 # The similarity criterion for the FINAL
clustering
sizeThreshold1 = 1 # The min. number of sequences/cluster for
that cluster to be retained
sizeThreshold2 = 4 # The min. number of sequences/cluster for
that cluster to be retained

[Chimera-killing Parameters]
abundance_skew = 1.9
```

If the user's operating system is detected as Linux-based, PURC v2.0 will default to using lima for demultiplexing. To override and use BLAST-based demultiplexing, change the override flag in the config file effect on other operating systems.

```
[Lima Override]
Lima_override = 0 # Set to 1 to use BLAST-based demultiplexing
```

3.3 Running PURC v2.0

3.3.1 Full Run with Demultiplexing and Sequence Inference

Once all input files are complete, PURC is initiated by calling the main PURC script with the config file as the argument:

```
purc.py config.txt
```

On completion, the output directory will have this structure, where log, output prefix, locus, and clustering method are replaced with those specified by the config file. It will contain a subdirectory for each locus and within each locus directory a subdirectory for each sample.

```
Output_folder
+-- log
+-- Output_prefix_1_bc_trimmed.fa
+-- Output_prefix_2_pr_trimmed.fa
+-- Output_prefix_3_annotated.fa
+-- Output_prefix_4_locus_clustering-method.fa
+-- Output_prefix_4_locus_clustering-method.aligned.fa
+-- Output_prefix_5_counts.xls
+-- Output_prefix_5_proportions.tsv
+-- Output_prefix_5_proportions.pdf
+-- Locus/
| +-- Locus.fa
| +-- Sample_1/
| +-- Sample_1_ASVs.fa
| +-- Sample_1_OTUs.fa
| +-- Sample_1_read_lengths.pdf
+--tmp/
```

- log—log file documenting the `PURC` run.
- Output_prefix_1_bc_trimmed.fa—all reads containing valid barcodes.
- Output_prefix_2_pr_trimmed.fa—reads with primer sequences removed (OTU clustering only).
- Output_prefix_3_annotated.fa—reads that could be assigned to samples based on barcodes or groups.
- Output_prefix_4_locus_clustering-method.fa—combined output sequences from all samples following OTU clustering and/or ASV inference. One file per locus per clustering method.
- Output_prefix_4_locus_clustering-method.aligned.fa—alignment of output sequence file.
- Output_prefix_5_counts.xls—summary of results containing number of reads surviving at each step in processing and final sequences per sample.
- Output_prefix_5_proportions(.tsv/.pdf)—summary of read coverage for the OTUs/ASVs for each sample. The PDF presents these data as pie charts with labels indicating the read coverage of each slice (as in Fig. 3).
- Locus/Locus.fa—all reads annotated to this locus.
- Locus/Sample_1/Sample_1_ASVs.fa—final ASV sequences for this sample (if applicable).
- Locus/Sample_1/Sample_1_OTUs.fa—final OTU sequences for this sample (if applicable).
- Locus/Sample_1/Sample_1_read_lengths.pdf—histogram of read lengths prior to ASV inference. Dotted lines indicate the limits for discarding too short/too long reads.

If PURC v2.0 is interrupted, it can be resumed by running again with the same config file. As long as all parameters are the same, it will determine the last completed step and continue.

3.3.2 Analyses on Previously Demultiplexed Data

PURC v2.0 includes a secondary script, `purc_recluster.py`, that can be used to perform OTU clustering on prior `PURC` runs or on data that have been demultiplexed by another method. Unlike the main script, `purc_recluster.py` operates with only command-line arguments.

```
./purc_recluster.py -f annotated_seq_file -o output_folder\
-c clustID1 clustID2 clustID3 clustID4\
-s sizeThreshold1 sizeThreshold2
```

If using PURC demultiplexed data, the input file is the `output_prefix_3_annotated.fa` file. If preparing your own data, the file must be FASTA formatted with name lines structured as follows:

```
>Isoetes_sp_Schafran1|LFY|BCF58^BCR2|
m170705_030709_42153_c10121536_s1_p0/62/ccs
```

The header line has elements separated by "|", where the sample name and locus name are the first and second elements, respectively. Any other elements after the sample and locus names, such as barcode, group, and read IDs, are not used. Four clustering identity thresholds are specified by the `-c/--clustering_identities` flag and two size thresholds specified by the `-s/--size_threshold` function identically to their respective parameters in the config file.

The reclustering script produces output organized similarly to the PURC output detailed above. This output contains two FASTA files for each locus, one of OTU sequences combined from all samples and the other an alignment of those sequences. There are folders for each locus and, within each, separate folders containing working files for each sample. A summary file called `purc_cluster_counts.xls` contains information about the number of reads, OTUs, and chimeras found per sample and per locus.

4 Conclusions

`PURC` [9], in conjunction with PacBio circular consensus sequencing, introduced an economical and effective alternative to time-consuming cloning and Sanger sequencing for generating broad multilocus datasets for groups containing polyploids. With PURC v2.0 we have significantly improved upon the earlier version, most notably by addressing shortcomings of OTU clustering by the incorporation of ASV inference. In addition to the generally greater accuracy of ASV (versus OTU) inference, our implementation of “straight” ASV inference and ASV inference with priors alongside OTU inference allows users to compare the results from all three approaches. PURC v2.0 can thus function as a data exploration tool, providing users with the opportunity to detect and interrogate unexpected patterns of variation in their amplicon sequencing datasets.

We anticipate that PURC v2.0 will prove to be a valuable component of biologists’ toolkits. While amplicon-based data generation does not scale as well as most other reduced-representation techniques, such as Hyb-Seq [e.g. 31, 32], it is a cost-effective way to sequence a greater number of samples for fewer—but more informative—loci, allows for the easy integration of new data with historical datasets, and is a powerful means of supporting “moderate data” approaches [33, 34] (i.e., the production of multilocus datasets that are large enough to be phylogenetically informative yet sufficiently small to allow for thorough curation and model selection; see also [35]). Moderate data are particularly effective for the phylogenetic study of polyploids, where the limiting factor is typically systematic error rather than stochastic error [36]: the accurate recovery and analysis of the full set of homeologous sequences (avoiding chimeras) is more important than the absolute amount of data available per se [6].

The main application of PURC v2.0 is thus likely to be as a component of a “polyploid phylogenetics” workflow [6]. For example, a researcher can use PURC v2.0 to generate a multilocus nuclear dataset for a broad taxon sample, including polyploids, phase the loci (determine, for each locus, which copy of each polyploid comes from which subgenome) with a `homologizer` [37], and use those phased multilocus data for downstream phylogenetic inference such as divergence-time or species-tree estimation. Such a workflow would allow for the investigation of many outstanding questions related to polyploid evolution and would also permit the phylogenetic study of groups that contain polyploids regardless of whether polyploidy itself is of central interest. In addition, PURC v2.0 is not restricted to cases of the multiple-copy problem. For example, to answer questions of hybrid parentage, plastid regions can be amplified and co-sequenced with nuclear loci and processed in the same `PURC` run, and PURC v2.0 is also an effective way to generate sequence data for basic diploids. Finally, while not specifically designed for metabarcoding, the underlying programs in PURC v2.0 are widely used in this field, making `PURC` useful for demultiplexing samples and sequence inference for other downstream analyses (e.g., [38]).

References

1. Ramanauskas K, Igić B (2017) The evolutionary history of plant T2/S-type ribonucleases. PeerJ 5:e3790
2. Goldberg EE, Kohn JR, Lande R et al (2010) Species selection maintains self-incompatibility. Science 330:493–495
3. Yang Z, Rannala B (2010) Bayesian species delimitation using multilocus sequence data. Proc Natl Acad Sci 107:9264–9269
4. Yang Z (2015) The BPP program for species tree estimation and species delimitation. Curr Zool 61:854–865
5. Griffin PC, Robin C, Hoffmann AA (2011) A next-generation sequencing method for overcoming the multiple gene copy problem in polyploid phylogenetics, applied to *Poa* grasses. BMC Biol 9:19. https://doi.org/10.1186/1741-7007-9-19
6. Rothfels CJ (2021) Polyploid phylogenetics. New Phytol 230:66–72
7. Schuettpelz E, Grusz AL, Windham MD, Pryer KM (2008) The utility of nuclear *gapCp* in resolving polyploid fern origins. Syst Bot 33: 621–629
8. Li F-W, Pryer KM, Windham MD (2012) *Gaga*, a new fern genus segregated from *Cheilanthes* (Pteridaceae). Syst Bot 37:845–860. https://doi.org/10.1600/036364412X656626
9. Rothfels CJ, Pryer KM, Li F-W (2017) Next-generation polyploid phylogenetics: rapid resolution of hybrid polyploid complexes using PacBio single-molecule sequencing. New Phytol 213. https://doi.org/10.1111/nph.14111
10. Dauphin B, Grant JR, Farrar DR, Rothfels CJ (2018) Rapid allopolyploid radiation of moonwort ferns (*Botrychium*; Ophioglossaceae) revealed by PacBio sequencing of homologous and homeologous nuclear regions. Mol Phylogenet Evol 120:342–353. https://doi.org/10.1016/j.ympev.2017.11.025
11. Kao T-T, Rothfels CJ, Melgoza-Castillo A et al (2020) Infraspecific diversification of the star cloak fern (*Notholaena standleyi*) in the deserts of the United States and Mexico. Am J Bot 107:658–675
12. Edgar RC (2010) Search and clustering orders of magnitude faster than BLAST. Bioinformatics 26:2460–2461
13. Edgar RC, Haas BJ, Clemente JC et al (2011) UCHIME improves sensitivity and speed of chimera detection. Bioinformatics 27: 2194–2200
14. Morales-Briones DF, Tank DC (2019) Extensive allopolyploidy in the neotropical genus *Lachemilla* (Rosaceae) revealed by PCR-based target enrichment of the nuclear ribosomal DNA cistron and plastid phylogenomics. Am J Bot 106:415–437. https://doi.org/10.1002/ajb2.1253
15. Suissa JS, Kinosian SP, Schafran PW et al (2022) Homoploid hybrids, allopolyploids, and high ploidy levels characterize the evolutionary history of a western North American quillwort (Isoëtes) complex. Mol Phylogenet Evol 166:107332
16. Blischak PD, Thompson CE, Waight EM et al (2020) Inferring patterns of hybridization and polyploidy in the plant genus *Penstemon* (Plantaginaceae). bioRxiv
17. Kao T-T, Pryer KM, Freund FD et al (2019) Low-copy nuclear sequence data confirm complex patterns of farina evolution in notholaenid ferns (Pteridaceae). Mol Phylogenet Evol 138: 139–155. https://doi.org/10.1016/j.ympev.2019.05.016
18. Chery JG, Acevedo-Rodríguez P, Rothfels CJ, Specht CD (2019) Phylogeny of *Paullinia* L. (Paullinieae: Sapindaceae), a diverse genus of lianas with dynamic fruit evolution. Mol Phylogenet Evol 140:106577
19. Wolfe AD, Blischak PD, Kubatko L (2021) Phylogenetics of a rapid, continental radiation: Diversification, biogeography, and circumscription of the beardtongues (*Penstemon*; Plantaginaceae). bioRxiv
20. Frost LA, O'Leary N, Lagomarsino LP et al (2021) Phylogeny, classification, and character evolution of the tribe Citharexyleae (Verbenaceae). Am J Bot 108(10):1982–2001
21. Blischak PD, Latvis M, Morales-Briones DF et al (2018) `Fluidigm2PURC`: automated processing and haplotype inference for double-barcoded PCR amplicons. Appl Plant Sci 6:e01156
22. Callahan BJ, McMurdie PJ, Holmes SP (2017) Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. ISME J 11:2639–2643
23. Barnes CJ, Rasmussen L, Asplund M et al (2020) Comparing DADA2 and OTU clustering approaches in studying the bacterial communities of atopic dermatitis. J Med Microbiol 69:1293–1302
24. Joos L, Beirinckx S, Haegeman A et al (2020) Daring to be differential: Metabarcoding analysis of soil and plant-related microbial

communities using amplicon sequence variants and operational taxonomical units. BMC Genomics 21:733

25. Nelson JM, Hauser DA, Li F-W (2021) The diversity and community structure of symbiotic cyanobacteria in hornworts inferred from long-read amplicon sequencing. Am J Bot 108 (9):1731–1744
26. Callahan BP, McMurdie PJ, Rosen MJ et al (2016) DADA2: high-resolution sample inference from Illumina amplicon data. Nat Methods 13:581–583. https://doi.org/10.1038/nmeth.3869
27. Rognes T, Flouri T, Nichols B et al (2016) VSEARCH: a versatile open source tool for metagenomics. PeerJ 4. https://doi.org/10.7717/peerj.2584
28. Tukey JW (1977) Exploratory data analysis. Addison-Wesley Publishing Company, Reading
29. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195–197
30. Katoh K, Standley DM (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol 30:772–780
31. Breinholt JW, Carey SB, Tiley GP et al (2021) A target enrichment probe set for resolving the flagellate land plant tree of life. Appl Plant Sci 9. https://doi.org/10.1002/aps3.11406
32. Johnson MG, Pokorny L, Dodsworth S et al (2019) A universal probe set for targeted sequencing of 353 nuclear genes from any flowering plant designed using k-medoids clustering. Syst Biol 68:594–606
33. Rothfels CJ, Li F-W, Sigel EM et al (2015) The evolutionary history of ferns inferred from 25 low-copy nuclear genes. Am J Bot 102: 1089–1107
34. Rothfels CJ, Larsson A, Kuo L-Y et al (2012) Overcoming deep roots, fast rates, and short internodes to resolve the ancient rapid radiation of eupolypod II ferns. Syst Biol 61:490
35. Frost LA, Lagomarsino LP (2021) More-curated data outperforms more data: Treatment of cryptic and known paralogs improves phylogenomic analysis and resolves a northern Andean origin of *Freziera* (Pentaphylacaceae). bioRxiv
36. Philippe H, Brinkmann H, Lavrov DV et al (2011) Resolving difficult phylogenetic questions: why more sequences are not enough. PLoS Biol 9:e1000602
37. Freyman WA, Johnson MG, Rothfels CJ (2020) `homologizer`: phylogenetic phasing of gene copies into polyploid subgenomes. bioRxiv. https://doi.org/10.1101/2020.10.22.351486
38. Goldberg AR, Conway CJ, Tank DC et al (2020) Diet of a rare herbivore based on DNA metabarcoding of feces: selection, seasonality, and survival. Ecol Evol 10:7627–7643