Last updated: June 28, 2013

APDP v1.1 Installation Notes and Tutorial

For new features and bug fixes in this version see separate Release Notes document. For comments, suggestions or questions email Matthew.Morgan@csiro.au.

Overview and aim of APDP

Amplicon Pyrosequencing Denoising Program, or APDP, identifies and removes errors from raw Roche 454 GS-FLX Titanium amplicon pyrosequencing data sets. APDP is a Perl implementation of the error-removal method described in Morgan et al. (in review). The aim of APDP is to provide a simple scripted workflow that is easy to use, and can process >1M titanium pyrosequencing reads from >100 samples on a single CPU without requiring access to a multi-processor computing cluster.

Sequences that pass all selection criteria are *validated*. Validated sequences are considered to be "real" in the sense that it is highly likely they are present in the DNA sample pre-amplification. Sequences flagged as probable errors are retained in designated files rather than discarded, and thus are available for separate analyses if required. Failure to properly account for errors has been shown to lead to inflated estimates of alpha diversity within a single sample and subsequent beta-diversity measures. It is therefore vital to identify and remove these errors prior to constructing operational taxonomic units (OTUs) or estimating diversity within or among samples.

What APDP does not do.

APDP does not perform OTU construction or taxonomic assignment. Many other algorithms and pipelines are available for these functions, although the accuracy and robustness of subsequent diversity estimates are dependent on the removal of methodological errors. APDP validated sequence outputs can be processed using other methods or pipelines, and we provide conversion scripts for this purpose.

APDP Installation

The included scripts have been tested on Linux (Ubuntu v.11.04), but should work on any Unix-based system (e.g. Mac OS X) with Perl plus the BioPerl module installed. The current version of APDP (v1.1) does not require access to a cluster and runs on a single CPU on a standard desktop PC (3.1GHz CPU, 4Gb RAM). Large data sets (>500k reads, or >100 samples) may take some time to run, but several obvious speed improvements are currently under development. A version of the alignment program MUSCLE (Edgar, 2004) is required to run APDP. The MUSCLE software and installation documentation can be obtained from:

http://www.drive5.com/muscle/.

Previous versions of MUSCLE (v3.7 and older) were provided as a program simply called *muscle*. Later versions (including the current v3.8.31) are run from a binary file with extra characters in the name e.g. *muscle3.8.31_i86linux64*. If you have a later version with such a name, you should create a copy of the binary file in the same directory and rename it just *muscle* (making a copy preserves the original file so you can keep track of the version you are using. Alternatively, you can make a link to the original binary file, and call it *muscle*),

You then need to ensure the binary file is accessible to your computer. To do this you can copy the muscle file to a directory in your path (e.g. /usr/local/bin or /usr/bin) and it will be accessible when required. Alternatively, you can put it in another directory and add that to your path. For example, if you put the file *muscle* in /home/bin/, to add it to your path, open a terminal window and type:

```
PATH=$PATH:/home/bin
export PATH
```

MUSCLE will now be in your executable search path.

APDP is supplied as a set of Perl scripts. To 'install' these scripts, first open a terminal window, navigate to the desired folder and unzip the compressed archive:

tar -zxvf APDPv1.1.tgz

This will extract a folder called “APDPv1.1” containing four subfolders “license”, “scripts”, “example1” and “example2”. The license and scripts folders are self-explanatory. The examples folder contains all data files required to follow this tutorial. These example files also act as templates for users to make their own input files.

That's it. APDP is ready to go. This tutorial assumes the user will navigate to the examples subfolder and run all analyses from there. To run empirical data sets following the same command line syntax, create a new subfolder in the APDP folder for the analysis, move into it and call the scripts using the same command lines as laid out below.

Roche 454 GS-FLX Titanium sequencing providers typically return three data files: a FASTA file (*.fna) containing raw sequencing reads trimmed of sequencing adapters and key, a quality file (*.qual) containing the quality scores for each base in the sequence file, and a flowgram file (*.sff). APDP only requires the FASTA file to run. The FASTA file you want to analyse should be placed in a subfolder within the APDP folder. APDP can also analyse sequence data in multiple FASTA files simultaneously. It is often that case that samples from a single experiment may be run over multiple regions of a sequencing plate, or even on separate sequencing plates, with the reads from each run contained in separate files. Details on how to set up your data file are provided in the Filter reads section below.

Data analysis with APDP

1. Filter reads, trim primers and barcodes, and bin by sequence.

There are two filtering approaches available through APDP. The first and original approach is to require a read to contain a perfect match to a forward and reverse primer and/or MID sequence to be accepted. This is implemented through the 1_Filter_sequences.pl script. The second approach is to require a read to contain a perfect match to only a forward primer and/or MID, reach a minimum sequence length, then truncates all reads at that length. This is implemented through the

1_Filter_by_truncation.pl script. This is particularly useful when amplicons are longer than the expected read length of the 454 technology and still outperforms other denoising approaches (Morgan et al., 2013).

To remove poor reads and consolidate accepted reads into unique sequences using the first approach, run:

perl 1_Filter_sequences.pl [file.fna] [gasket file] [forward primer+MID file] [reverse primer+MID file] [unused barcode combinations file]

- **Example: perl 1_Filter_sequences.pl subsample.fna gasket04 forward_primers_mids.txt reverse_primers_mids.txt**

To use the second (truncation) approach, run:

perl 1_Filter_by_truncation.pl [file.fna] [gasket file] [forward primer+MID file] [reverse primer file] [truncation length]

- **Example: perl 1_Filter_by_truncation.pl subsample.fna gasket04 forward_primers_mids.txt reverse_primers_mids.txt 110**

Note that you still need to supply a reverse primer file to the truncation script, but this is not used for filtering. Output files and subsequent steps are identical for both approaches.

INPUT FILES

- **file.fna :** the FASTA file containing the sequences of interest.
- **gasket file :** required by the script to know from which regions of the picotiter plate to expect pyrosequencing reads. The region a read came from is contained in the read name as a two-digit number from 01-16 (note the use of 0 for numbers below 10) following the run number. This information should be provided as a tab-delimited text file containing the run name plus one or more numbers (e.g. GY3J8KL01 to GY3J8KL16), with each number on a new line.

- **Primer+MID files :** The minimum requirement for APDP to accept reads is for a perfect match to a forward and reverse primer sequence. Forward and reverse barcodes (or MIDs) can be provided and are recommended (at least at the forward primer). Forward and reverse primers are provided in two separate files. Each file should be formatted as tab-delimited text with the Primer Name in the first field, and the primer (plus MID) sequence in the second field.

  APDP expects to see the primers in the same orientation as the pyrosequenced read. That is, the forward primer file should be oriented as MID sequence-primer sequence, whereas the reverse primer file should be reverse-complemented and oriented as primer sequence-MID sequence.

  APDP can detect degenerate primers (i.e. primers with ambiguous bases such as “N” or “Y”) using standard perl regular expression syntax. That is, all ambiguous base characters should be replaced with square brackets containing the possible matching nucleotides. For example, “N” should be replaced with “[ACGT]”. Thus the primer sequence AGGTNTGC should be entered in the text file as AGGT[ACGT]TGC. Likewise, the primer sequence AGGTYTGC should be entered in the text file as AGGT[TC]TGC.
- **Unused barcode combinations :** Optional file. APDP expects to be able to see reads from all combinations of the provided forward and reverse barcodes. This may not be the case, and users may want to ensure unused barcodes do not get used in downstream analyses. Unused barcodes are provided as a text file as region-forward mid-reverse mid (e.g. GY3J8KL02F01R03), one per line.

*One further note of caution: it should be noted that the text files for input must not contain any hidden characters (such as those inserted by some word processors).

INPUT PARAMETERS

APDP will prompt for three optional parameters.

- **Minimum sequence length :** Sets minimum acceptable sequence length for unique sequences after primer and MID trimming. If no value is set, default value of 40bp will be used.

- **Multiple primer sequences :** APDP can search for whether a primer exists more than once in the sequence, and exclude sequences where this is the case (rarely the case, but may happen). If required, APDP will prompt for each sequence to look for after primer and MID trimming. If no sequence is input, APDP ignores this step.

- **Example Usage: Press enter at all prompts i.e. use default Minimum sequence length and do not search for additional primer sequences**

OUTPUT FILES

This step outputs eight files.

- **Filtered_unique_sequences.txt :** Accepted multi-read unique sequences. For each unique sequence data output fields are: a unique sequence identifier; sequence length (post-primer and MID trimming); a read number from the raw data of a single read representing the unique sequence as a link to the raw data; the unique DNA sequence; the total number of observed reads assignable to barcodes, the distribution of reads across barcodes.
- **Filtered_unique_sequences.fasta :** Filtered unique sequences in FASTA format.
- **Reads_with_forward_and_reverse_MIDs.txt :** As Filtered_unique_sequences.txt except contains all observed unique sequences before sequence filtering (includes: singletons, short sequences, ambiguous bases).
- **Reads_with_forward_and_reverse_MIDs.fasta :** All observed unique sequences in FASTA format.
- **Rejected_reads.txt :** Same as Filtered_unique_sequences except contains only the rejected unique sequences
- **Read_status.txt :** Each read name, whether it was accepted and the observed barcode combination) or rejected (and reason for rejection).
- **Read_status_summary.txt :** Summarises number of reads accepted and rejected and reason for rejection.
- **Reads_by_sequence.txt :** Each unique sequence, the observed number of reads, and list of reads with that sequence.

APDP can analyse sequence data in multiple FASTA files simultaneously. To run data from multiple FASTA files simultaneously, you will need to concatenate the relevant FASTA files together into a single file. The easiest way is to use the “cat” command. For example:

```
cat input1.fna input2.fna >outputfile.fna
```

You will also need to add the region and/or plate names of all files to the gasket file.

2. Compare all filtered unique sequences to reference database (Genbank) and return best hit information

Downstream steps require valid sequences to be grouped in to clusters of similar sequences. The input for these next steps is essentially the information from Filtered_unique_sequences.txt with group information included. APDP does this using remote megablast against the NCBI database (again, to reduce the computational constraints of the user), which requires an internet connection. Any method can be substituted in here to assign sequences to groups, (e.g. local BLAST searches or sequence similarity clustering methods) as long as the group information is tagged to the Filtered_unique_sequences.txt sequence information in a text file, analogous to the output of this script (**Filtered_unique_sequences_groups.txt**).

To assign the filtered unique sequences from the previous step to groups based on the NCBI nr database, run:

perl 2_Assign_groups.pl [filename] [filter]

- **Example Usage: perl 2_Assign_groups.pl Filtered_unique_sequences.txt**

COMMAND LINE INPUT

- **filename :** Will normally just want to use sequences in

Filtered_unique_sequence.txt. For additional analyses some users may also want to use all observed sequences (Reads_with_forward_and_reverse_MIDs.txt).

INPUT PARAMETERS

APDP will prompt for two parameters:

BLAST filter : Determines whether to include ("Y" - case-sensitve) or exclude (any other keystroke) environmental and metagenomic sequences from the nr database during searches.

BLAST algorithm : To use BLASTN type "0"; to use MEGABLAST type "1".

OUTPUT FILES

This step outputs two files:

- Filtered_unique_sequences_groups.txt : Input sequences with NCBI information appended. Appended information fields are : sequence identifier (confirmation that correct information is appended); Bit score of best hit; NCBI GI of best hit (or most recent GI if multiple best hits); number of Genbank hits with same Bitscore (max=10); list of all GIs with same Bitscore (max=10).
- Filtered_unique_sequences_no_significant_similarity.txt : List of input sequences with no significant similarity to anything in the NCBI nr database using MEGABLAST. We see very few unique sequences (representing a very low proportion of pyrosequencing reads) that fall in this category, although exceptions may apply especially if uncommon amplicons are used. If many sequences are found in this category (or sequences representing a large number of reads) another method of assigning groups may be more appropriate. At this time, APDP does not implement any other grouping methods although they can be used as explained below Any sequences in this file are excluded from further analyses.

3. Preliminary validation within Genbank-defined similarity groups

This step uses the distribution of reads to determine the validity of a sequence within each sample that it occurs. APDP has two versions: the default multi-sample version (APDP-MS) assumes the data contains different samples. If your data comprises a single sample, you should use the single-sample version (APDP-SS; uses 3_Preliminary_validation_SS.pl). APDP-SS uses a modified Preliminary Validation script ignoring the criterion requiring low-abundance sequences to appear in multiple samples to be validated.

A spreadsheet program is useful to obtain the field (i.e. column number) information required at this step. To run Preliminary Validation on the grouped unique sequences, run:

perl 3_Preliminary_validation_MS.pl [Unique sequences plus Group info] [Group info field/column] [Total reads field/column] [First sample field/column] [Number of consecutive MID/sample columns i.e. number of samples]

- Example Usage: perl 3_Preliminary_validation_MS.pl Filtered_unique_sequences_groups.txt 53 5 6 45

COMMAND LINE INPUT

Unique sequences plus Genbank info : The output from 2_Assign_groups.pl (Filtered_unique_sequences_groups.txt), or analogous file containing sequence, read distribution and grouping information.

Group info field : Field or column of input file containing Group information.

Total reads field : Field or column of input file containing Total Number of reads for each sequence (normally 5).

First sample field : Field or column of input file containing first sample read abundance information (normally 6).

Number of samples : Number of samples for which to expect read information. Samples should be in consecutive columns.

INPUT PARAMETERS

APDP will prompt for an additional parameter:

Read proportion cut-off (default = 0.50) : In addition to validating the most-abundant sequence in each group (see original manuscript for exceptions) APDP will validate additional sequences if observed with sufficient reads in any given sample. This is calculated for each sample as a user-defined proportion of the number of reads observed in that sample for the overall most abundant read in the group. The default for this parameter is 0.50 i.e. any sequence in the group with >50% of the reads observed for the most abundant group sequence is preliminary valid. If set to 1.0, additional sequences will only be validated if they are the most abundant within any sample. Hit enter to accept the default value.

OUTPUT FILES

This step outputs seven files :

Preliminary_validated_sequences.txt : All input sequences passing Preliminary Validation. Includes read number and group information.

Preliminary_valid_sequence_names.txt : Sequence names of all preliminary valid sequences.

Preliminary_valid_sequences_by_group.txt : All preliminary valid sequences organised by group.

Top_group_Preliminary_Validated_sequences.txt : Sequences validated as most abundant within their group.

Additional_group_validated_seqs.txt : Sequences validated as additionally valid within their group (i.e. not most abundant within group)

Top_group_rejected_rare_sequences.txt : Sequences rejected, even though most abundant within their group. Comprises sequences observed in a single sample with <10 reads. We have observed that low-read, poorly reproducible sequences are highly likely to be errors. These are considered probable errors but may be rare real taxa, and so are output here for further investigation if required.

Additional_group_rejected_rare_sequences : Low-read sequences passing the cut-off

threshold in one or more samples. The output fields are sequence identifier; sample number passing cut-off; number of reads in the sample; sequence group name; number of reads in the sample for the most-abundant sequence in the group. As above, these are considered to be probable errors, although some users may want to investigate these sequences more closely.

4. Secondary validation within individual samples

This step examines all the sequences observed in each sample, and assesses whether each sequence is likely to be derived from one of the three defined critical error types (indels, DNA polymerase-error, chimera). To pass sequences through Secondary Validation, run :

perl 4_Secondary_validation.pl [Preliminary_validated_sequences.txt] [First sample number] [Number of samples]

Example Usage: perl 4_Secondary_validation.pl Preliminary_validated_sequences.txt 1 45

This step requires the alignment program MUSCLE is installed and that the folder containing the program is in your path. Our implementation uses hard-coded settings for MUSCLE that result in the fastest possible speed for nucleotide alignments (see Muscle user-guide for more information).

COMMAND LINE INPUT

Preliminary_validated_sequences.txt : Provisionally validated sequences from step 3.
First sample number : First sample (MID combination) to analyse. This step numbers samples from starting at 1. Normally, users will want to start with the first sample. This is user-definable in case users want to run this step in stages (i.e. stop then re-start at the same point) or wish to run the step on multiple computers to speed up the process (this step is the rate limiting portion of the analysis).
Number of samples : Number of samples or MID combinations in consecutive fields.

INPUT PARAMETERS

APDP will prompt for two additional parameters :

Cutoff for possible PCR error (default = 0.02 or 2%) : The maximum number of reads a 1nt-DNA polymerase error is expected to have, as a proportion of the observed reads for the parent sequence. A putative DNA polymerase error with more than the expected maximum number of reads is validated. The default maximum value has been determined from multiple independent positive controls in multiple Titanium runs from different service providers. Hit enter to accept the default.

Cutoff for possible chimera error (default = 0.15 or 15%) : The maximum number of reads a single-crossover chimera is expected to have, as a proportion of the observed reads for the parent sequence. A putative chimera with more than the expected maximum number of reads is validated. The default maximum value has been determined from multiple independent positive controls in multiple Titanium runs from different service providers. Hit enter to accept the default.

OUTPUT FILES

This step creates a validation results file for each sample. Each sequence observed and evaluated within the sample is output with its evaluation status. These files are retained in a new folder (Validation_by_sample). WARNING : Sample validation results files from previous script4.pl runs will be overwritten. The final step of the script calculates the number of valid and invalid observations for each sequence and outputs the results to a new file.

Validation.mid* : One validation results file for each of the *N* samples (numbered from 0 to *N*-1).

Final_validation.txt : Number of valid and invalid observations for each input sequence.

5. Final Validation: Filter for sequences valid in minimum number of samples

The last step filters the Preliminary Valid sequences (output from Step 3) for sequences passing Secondary Validation in a minimum number of samples. The minimum number of samples required will depend on features of the experimental design such as technical or biological replication.

perl 5_Final_validation.pl [Preliminary_validated_sequences.txt] [First sample column]

Example Usage: perl 5_Final_validation.pl Preliminary_validated_sequences.txt 7

COMMAND LINE INPUT

Preliminary_validated_sequences.txt : File from Step 3 output used in Step 4 input.
First sample: Field or column in input file containing first sample information (if the example protocols (below) are followed, this data will be in column 7).

OPTIONAL INPUT

APDP will prompt the user for a file containing information about biological or technical replicates in the experimental design that should be taken into account when validating sequences. For example, we routinely run 2-3 technical replicates for each sample (replicate PCRs performed on the same extracted DNA sample), and we expect real sequences to be highly reproducible between replicates. Therefore we require sequences to be validated in most or all technical replicates. Biological replicates are expected to be more variable (there is no *a priori* assumption that these will contain the exact same set of real sequences). We therefore normally require sequences to be validated in a single biological replicate (as we would expect some real sequences to be present in just one sample), although this parameter can be varied to assess sensitivity of biodiversity estimates.

Replicates file : a tab-delimited text file. See example2/replicates.txt for an example of the input format. Fields are : the location/experiment name; the total number of samples in the input file; the minimum number of valid observations to validate a sequence; list of MID numbers representing the sample. MID numbers can be input as ranges (numbers

separated by "-" or individually (numbers separated by ","). This file is optional. If no file is entered APDP will run this step as if the data set is a single experiment and each MID-combination represents an independent sample, and will prompt for the minimum number of valid observations to retain a sequence (default is 1 – hit enter to accept the default).

APDP also prompts for the minimum number of reads for "detection". This parameter does not affect which sequences are validated, but will affect the number of reads output to some of the auxillary files or some statistics (e.g. reads output to **Final_validated_sequences_All_Samples_invalid_reads_removed.txt,** and the confidence calculation for **Final_validated_sequences_All_Samples_invalid_reads_removed.txt** (see below)). To reiterate, these files and statistics are not used in the validation process. They are solely used for qualitative data exploration and the effects on downstream analyses of removing low frequency sequences or low confidence sequences from individual samples.

OUTPUT FILES

This step outputs six files. APDP creates a folder for all sequences validated across the data sets ("ALL_SAMPLES"). If a replicates file is specified with more than one location or experiment name, APDP also creates a new folder and outputs files for each one separately:

Final_validated_sequences_All_Samples_invalid_reads_retained.txt: This file contains the Final Validated sequences that we recommend to use for downstream analyses. The reference to "invalid reads retained" in the name means that all the reads observed for a validated sequence are retained in the output file, even if the sequence was flagged as potentially invalid in one or more samples. Output fields are the same as the input file, including the read data for all samples and all validated sequences, but also includes a column with a "confidence" calculation – this is the number of observed reads above the user-defined detection limit that were independently validated across samples. This value is not used to validate sequences, but it is possible to explore the effects on downstream analyses of removing "low confidence" sequences.

Final_validated_sequences_All_Samples_invalid_reads_removed.txt: Same as Final

validated sequences , except that all instances where validated sequences were evaluated as invalid and all observations below the user-defined detection limit (default is 2) have been replaced with 0 reads. Similar to the confidence calculation, this is not used in the validation process. This file is only used as a method for exploring the effects of removing low frequency sequences from individual samples. We recommend using the alternative **Final_validated_sequences_All_Samples_invalid_reads_retained.txt** for downstream analyses.

Invalid_or_ambiguous_sequences.txt : All sequences with only invalid evaluations and no valid observations (invalid sequences) and all sequences with a) 0<x<Min Valid valid observations, and b) zero valid or invalid observations (ambiguous sequences).

CONVERTING APDP OUTPUT

APDP validated sequences can be used with other tools for community diversity analyses. We provide a perl script that takes the validated sequences text file and converts it into formats usable by QIIME and mothur.

perl convertAPDPoutput.pl [Final_validated_sequences] [Number of samples/MIDs]

Example Usage: perl convertAPDPoutput.pl Final_validated_sequences.txt 9

OUTPUT

APDP_validated_sequences_qiime.fasta : This file is analogous to the inflated denoiser output (or the split libraries script) in QIIME. This file can be directly inserted in to a custom QIIME pipeline that requires a correctly-formatted fasta file. For example, this file can be used as input for the for pick_otus.py script.

APDP_validated_sequences_mothur.fasta (plus *.names and *.groups) : Sequences, names and groups files that can be inserted in to a mothur command pipeline for further analysis.

WORKED EXAMPLES

Example 1

This example simulates an analysis of Roche 454 GS FLX Titanium pyrosequencing data derived from three environmental samples. Each sample was taken from a different location. After DNA extraction, a region of the 18S rRNA gene was PCR amplified in each sample separately. Each sample was then labeled with a unique 10bp multiplex identifier (MID) or barcode at the 5' (or “forward”) end. The three samples were pooled and sequenced on a single region.

The data files required for this analysis are in “APDP/example1”. To use the commands as presented here, APDP must be run from within this folder.

The required data files are :

example.fna : 32,940 raw pyrosequencing reads in fasta format.
regions.txt : text file of the PTP regions in which the reads were sequenced. In this case, they all come from the same region (region 02).
forward_primers_mids.txt : text file of fusion primer (MID + 18S rRNA forward primer sequences).
reverse_primer.txt : text file of reverse primer used in 18S rRNA PCR amplification.

Step 1 : Filter reads, bin unique sequences

In a terminal window, type:

```
perl ../scripts/1_Filter_sequences.pl example.fna regions.txt forward_primers_mids.txt reverse_primer.txt
```

When prompted: Set a minimum sequence length of 120bp, and ignore the prompts for primer sequences (hit enter through them).

APDP should output:

F01 ACGAGTGCGTTGGTGCATGGCCGTTCTTAGT
F02 ACGCTCGACATGGTGCATGGCCGTTCTTAGT
F03 CGTGTCTCTATGGTGCATGGCCGTTCTTAGT
Rev GGTCTGTGATGCCCTTAGATG
Found 32940 raw reads in fasta format and converted to tab-delimited text
There are 5227 unique sequences

There are 1219 filtered unique sequences

30 (this is the time in seconds taken to process the data and may vary).

Step 2 : BLAST sequences and assign to groups

This step will take a while, so for this tutorial you can skip it and use the pre-run output in the examples_prerun_results folder. Move these files into the examples1 folder and continue to Step 3. If you want to run this and obtain the same files yourself,follow the instructions below.

perl ../scripts/2_Assign_groups.pl Filtered_unique_sequences.txt

Exclude metagenomic and environmental samples (type “n” at the prompt) and use the MEGABLAST algorithm (“1” at the prompt).

Will output two files: one with the group info attached and one with all the sequences that did not return a significant BLAST hit. In this example it should contain no sequences.

This may take up to eight hours depending on NCBI's server load. Again, remote BLAST is used to reduce the computational burden on the user. A local BLAST database significantly improves the speed of this step

Step 3 : Preliminary Validation

perl ../scripts/3_Preliminary_validation_MS.pl Filtered_unique_sequences_groups.txt 11 5 6 3

The first number (11) sets the input data field or column with the group information, the second (5) is the field with the Total reads, the third (6) contains the first MID read data, and the fourth (3) is the number of consecutive MID read data columns.

Use the default read proportion cut-off – either type “0.5” or just hit enter.

APDP should output something like:

Processed 31 of 31 groups
10 seconds taken
Done.

The output in Preliminary_validated_sequences.txt should contain 24 sequences.

Step 4 : SecondaryValidation

perl ../scripts/4_Secondary_validation.pl Preliminary_validated_sequences.txt 1 3

APDP will evaluate the sequences in each MID sample in turn and assess each one as “valid” or “invalid” within that MID. Use the default cut-off values for PCR and chimera errors by hitting enter through the following prompts.

This step should take around 90 seconds to complete. The final output will be a new folder (**Validation_by_sample**) containing a file for each sample with the status of each sequence, and a tab-delimited text file listing all pairs of sequences that differ by indels. Both are used by the next step to determine which sequences should be retained.

Step 5 : Filter by Valid Observations

perl ../scripts/5_Final_validation.pl Preliminary_validated_sequences.txt 7

APDP will prompt for a replicates file. There is no replicate information file for these data, so hit enter at the prompt. The total number of samples (MID combinations) is three. APDP will treat the three MID samples as independent, and a sequence will be validated if evaluated as valid in at least one sample. Use the default values for the next two prompts (hit enter through them). These parameters do not affect which sequences are validated, but affect some of the auxillary outputs.

Files (see instructions) will be output in a new folder “ALL_SAMPLES”. The output in **Final_validated_sequences_All_Samples_invalid_reads_retained.txt** and **Final_validated_sequences_All_Samples_invalid_reads_removed.txt** should contain 16 valid sequences and the associated read distributions. There should be eight invalid sequences.

Example 2

This example simulates a more complex experimental design than Example 1. It uses the same raw data as Example 1, but utilises all features of APDP. In this example, the sample taken from Location 1 (labelled with F01 in Example 1) actually comprise three sampling replicates. The sampling replicates were processed independently and labelled with the same forward barcode (F01) but a unique barcode was added to the 3' ("reverse") end of the amplicon (R01-R03). Location 2 (labelled F02) comprises two sampling replicates labelled with R01 and R02. Location 3 (labelled F03) has only a single sample, but was PCR amplified twice to give two technical or PCR replicates, labelled R01 and R03. The table below shows the barcode combinations used.

	R01	R02	R03
F01	LOC1	LOC1	LOC1
F02	LOC2	LOC2	NOT USED
F03	LOC3	NOT USED	LOC3

There are steps throughout the analysis that will account for the study design used here, the presence of reverse barcodes, and the existence of unused potential barcode combinations.

The data files required for this analysis are in "APDP/example2". To use the commands as presented here, APDP must be run from within this folder.

The required data files are :

example.fna : 32,940 raw pyrosequencing reads in fasta format.
regions.txt : text file of the PTP regions in which the reads were sequenced. In this case, they all come from the same region (region 02).
forward_primers_mids.txt : text file of forward fusion primers (MID + 18S rRNA forward

primer sequences).

reverse_primers_mids.txt : text file of reverse fusion primers (MID + 18S rRNA reverse primer sequences). Note these are reverse-complement sequences – they are oriented as in the raw reads.

unused_mids.txt : text file of the unused barcode combinations. These will appear in the output but with zero reads in all fields.

Step 1 : Filter reads, bin unique sequences

perl ../scripts/1_Filter_sequences.pl example.fna regions.txt forward_primers_mids.txt reverse_primers_mids.txt unused_mids.txt

Set a minimum sequence length of 120bp, and either enter the forward and reverse primer sequences or ignore the prompts for primer sequences (hit enter through them). APDP should output:

F01 ACGAGTGCGTTGGTGCATGGCCGTTCTTAGT
F02 ACGCTCGACATGGTGCATGGCCGTTCTTAGT
F03 CGTGTCTCTATGGTGCATGGCCGTTCTTAGT
R01 GGTCTGTGATGCCCTTAGATGCATAGTAGT
R02 GGTCTGTGATGCCCTTAGATGCGAGAGATA
R03 GGTCTGTGATGCCCTTAGATGATACGACGT
Unused combinations:
02F02R03 1
02F03R02 1
Found 32940 raw reads in fasta format and converted to tab-delimited text
There are 5220 unique sequences

There are 1219 filtered unique sequences

30 (this is the time in seconds taken to process the data and may vary).

Step 2 : BLAST sequences and assign to groups

This step will take a while, so you for this tutorial you can skip it and use the pre-run output

in the examples_prerun_results folder. Move these files into the examples2 folder and continue to Step 3. If you want to run this and obtain the same files yourself,follow the instructions below.

perl ../scripts/2_Assign_groups.pl Filtered_unique_sequences.txt

Exclude metagenomic and environmental samples (enter “n”) and use the MEGABLAST algorithm (“1”).

This step will output two files: one with the group info attached and one with all the sequences that did not return a significant BLAST hit. In this example it should contain no sequences.

Step 3 : Preliminary Validation

perl ../scripts/3_Preliminary_validation_MS.pl Filtered_unique_sequences_groups.txt 17 5 6 9

The first number (17) sets the input data field or column with the group information, the second (5) is the field with the Total reads, the third (6) contains the first MID read data, and the fourth (9) is the number of consecutive MID read data columns.

Use the default read proportion cut-off – either type “0.5” or just hit enter.

APDP should output something like:

Processed 31 of 31 groups
20 seconds taken
Done.

The output in Preliminary_validated_sequences.txt should contain 25 sequences.

Step 4 : Secondary Validation

perl ../scripts/4_Secondary_validation.pl Preliminary_validated_sequences.txt 1 9

APDP will evaluate the sequences in each MID sample in turn and assess each one as “valid” or “invalid” within that MID. Use the default cut-off values by hitting enter through the following prompts.

The output to screen will look like:

```
rm: cannot remove `validation*': No such file or directory
Set PCR error cut-off (default value = 0.02) :

PCR error cut-off set to 0.02

Set chimera error cut-off (default = 0.15) :

Chimera error cut-off set to 0.15

...

Took 70 seconds to complete
```

Step 5 : Final Validation

perl ../scripts/5_Final_validation.pl Preliminary_validated_sequences.txt 7

APDP will prompt for a replicates file. For this data set it is “**replicates.txt**”. The format for this file was explained earlier. Briefly, each experiment/set of samples is given a separate line. The minimum number of valid observations required to validate a sequence will be dependent on the experimental design. In this example, because LOC1 and LOC2 comprise biological (sampling) replicates, there is no a priori reason to assume they should contain identical real sequences. Thus a sequence valid in any one of the replicates is considered “real”. LOC3 comprises two technical (PCR) replicates, and sequences must be valid in both to be validated. Use default values for all other parameters when prompted. These parameters do not affect which sequences are

validated, but affect some of the auxillary outputs.

The output to screen will look like:

…
MIDS : 0 1 2 End
0
LOC1 MID: 0
LOC1 MID: 1
LOC1 MID: 2
Took 41 seconds to complete

The output files (see instructions) will be output in a new folder for all samples (ALL_SAMPLES), in addition to one for each experiment (LOC1-3). The output in **Final_validated_sequences_All_Samples_invalid_reads_retained.txt** and **Final_validated_sequences_All_Samples_invalid_reads_removed.txt** should contain 16 valid sequences for each LOC, and overall in **Final_validated_sequences_All_Samples_invalid_reads_retained.txt** (in ALL_SAMPLES).