New metabarcoding pipeline

User Guide

Overview

- Updated the packages after the SAGA system went through a complete update
- No longer necessary (almost…) to use a conda environment
- Should be even more (!) user friendly
- Two options for clustering after DADA2, VSEARCH and SWARM (more on this later)
- New naming convention of sequences, same names across samples
- More parallell computing, entire pipeline runs in one work day or less

Notes before starting

- All the scripts you need are collected in one folder, Zazzy_pipeline_v2_scripts – copy this entire folder to your work area
- Only one step requires an Anaconda environment, ITSx
- Follow the instructions in the READ_ME file
- Background scripts are still in the same place in the original zazzy folder
  - Main scripts that require the background scripts are properly pathed, no need to change anything

Step 1 – Demultiplex using cutadapt

- slurm_script_1_dem_pairedadapters.sh
- Set up to check for primer tags in BOTH ENDS of each sequence
- Important update/bug fix that will for the most part take care of tag-switching problems
- No change for the user on how it runs
- User only needs to properly format the map files
  - Example files in the v2 Zazzy folder

Step 2 – Clean samples before DADA2

- script_2_remove_demult_files_with_very_low_read_number.sh
- Sometimes the R1 and R2 have a slightly different amount of reads
- DADA2 can throw errors when only one of the R1/R2 have a very low amount of sequences
- Run this script first and check the DADA2_AS and DADA2_SS folder manually that all sample files have both an R1 and an R2 version
  - Not doing this always leads to DADA errors that are difficult to decode.

Step 3 – DADA2 denoising

- slurm_script_2_runDADA2_for_SWARM.sh
- script_3_dependency_R_code_sha1_hash_names.R – this needs to be copied into working folder

- A few changes to this:
- General change, only R v.4.1 and v4.2 are available on saga now
- Check your personal R libraries and make sure dada2 package is installed
- This script now gives each sequence a unique 40 character name, these are the same across all samples
- This is necessary for running SWARM, and does not affect the VSEARCH process
- There is no need for the user to change anything in the DADA2 outputs

Step 4 – Trim ITS sequences using ITSx

- Skip step 4 if not working with ITS
- ITSx is not a module on SAGA so you need to activate it manually
- ITSx is a module in Bioconda, so you can make your own environment
- Alternatively, install Zazzy_metabarcoding_pipeline and load this as described in the original folder
- Run ITSx on an interactive node
- From this step all the rest can be done in the same interactive node

Step 5 - Clustering

- Two options – SWARM and VSEARCH
- VSEARCH: Clustering based on pre-defined similarity
  - Threshold can be defined by the user, default 97%
- SWARM: single linkage clustering algortihm
  - Number of accepted differences can be changed in the main SWARM script
  - Will better be able to separate amplicons that are very similar
- Important to set the name of output files
- Outputs a centroid file and an OTU table
- SWARM also oputputs some other files
- SWARM is run without sending to the slurm queue

VSEARCH

```
#SETTING VARIABLES
VSEARCH=vsearch
THREADS=10
CLUSTER=0.97
LOW_ABUND_SAMPLE=2
PROJ="Vilde_fasteris_sequences"
TMP_FASTA1=$(mktemp)
TMP_FASTA2=$(mktemp)
TMP_FASTA3=$(mktemp)
TMP_FASTA4=$(mktemp)
TMP_FASTA5=$(mktemp)
TMP_FASTA7=$(mktemp)
```

SWARM

```
#VSEARCH=$(which vsearch)
#SWARM=$(which swarm)
TMP_FASTA=$(mktemp --tmpdir=".")
FINAL_FASTA="Solhomfjell_fungi.fas"
module purge
```

Step 6 – LULU post-clustering

- Run script_6_lulu_curation.sh
- No need to send to slurm, work in the same interactive node

- Remember to set the name of your centroid in the main LULU script
  - No need to do anything with the R scripts

- Two dependencies: Copy the correct R dependency script and activate the correct line in the lulu curation script

- Separate scripts for SWARM and VSEARCH clustered centroids

- This generates a new centroid file and OTU table without deleting the output from SWARM or VSEARCH in case you want to compare pre- and post-LULU

Step 7 – Taxonomic annotation

- Here you can choose between BLAST and SINTAX for assigning taxonomy
- SINTAX is recommended, BLAST with 5 hits per seq can be used as a control
- Running the SINTAX script will output an OTU table with taxonomy nicely formatted for downstream analysis

SINTAX

- Algoritm by Robert C. Edgar (USEARCH author)
- Adapted for VSEARCH by Rognes et al, slightly stripped down version
- Check for most likely hit in the database using a slightly different approach than BLAST
- Checks each taxonomic rank sequentially
- Outputs a bootstrap value per taxonomic rank for each sequence
- Less prone to false positives than BLAST