**### Originally post July 2018#######**

**### Platform: UBUNTU 16.04#######**

**###This is no longer supported. Please use the latest APPTAINER image.####**

**To do: List publications that used this version.**

**ECPP**

Exon Capture for Phylogenetics

Contributors:

Adnan Moussalli, Tim O’Hara, Mark Phuong, Luisa Teasdale, and Jose Grau

This workflow is essentially a BASH wrapper around a selection of established programs and in-house scripts that take you from raw high-throughput (e.g. illumina) sequence data to final alignments ready for downstream phylogenetic analyses.

For those interested in exploring the currently stable release, I have provided a bash script for the installation of all dependencies.

ECPP\_1.0.0\_install.sh

This pipeline has many dependencies, and indeed dependencies on specific versions. As much as possible I do try to keep up with updates and upgrades, but please do pay attention to the versions recommended for installation. An ideal approach is to install onto a new virtual machine, using vmware or virtualbox for instance. Ultimately, the objective is that wrap all this up in a DOCKER image.

Things you need:

1. Raw sequences, demultiplex. It is a good idea to rename them to something meaningful, e.g. EC1\_FAMILY\_GENUS\_SPECIES\_REGNO\_LOCALITY. This name will carry through all the way to the first tree you produce. Create a new folder for each projects, and in that folder place the raw sequences in a subfolder named “0\_raw”. Note, all file must end in \*\_R1.fastq.gz or \*\_R2.fastq.gz or \*\_RS.fastq.gz.
2. A reference file containing a single representative sequence for each exon, translated. This is typically your bait design before you chopped them up into probes.
3. A list of the samples you want to process. These will be the names associated with the raw sequence files, but without “\_R1.fastq.gz”.

ECPP has 5 modules and are run sequentially:

1. **CLEAN** - this module remove duplicates, subsets the raw reads to only those having a hit to the reference and quality trims and removes adapters using trimmomatic.
2. **TRINITY/SPADE** - this module assembles using the CLEAN reads. You have a choice of using either Trinity or Spade, I recommend Trinity. It then tblastn the reference (as query) against the sample specific assembly (as subject).  
   The best hit per reference exon is identified and the local hit coordinates used to extract the corresponding exon. This creates a new reference which is "sample specific".
3. **MAP** - this module maps the CLEAN reads onto the new "sample specific" reference using BBMAP, followed by variant calling using VARSCAN.
4. **ALIGN** - this module will produce a seprate alignment for each targeted exon (fasta files), and 'all exons combined' supermatrix files in fasta/nexus (with CHAR SET defined)/phylip format.
5. **SUBSET** – based on summary file from the last module, you identify either exons or taxa that should be removed. Here is you opportunity to do so.