**The pipeline**

$ mkdir fastq-join\_joined; cp \*R1\* DCM1.fastq; cp \*R2\* DCM2.fastq; fastqc DCM1.fastq; fastqc DCM2.fastq; trim\_quality -t fastq -o DCM1\_trimmed.fastq --paired\_reads DCM1.fastq; trim\_quality -t fastq -o DCM2\_trimmed.fastq --paired\_reads DCM2.fastq; fastqc DCM1\_trimmed.fastq; fastqc DCM2\_trimmed.fastq; mv \*.py fastq-join\_joined; mv DCM1.fastq fastq-join\_joined; mv DCM2.fastq fastq-join\_joined

$ qiime

$ join\_paired\_ends.py -f DCM1\_trimmed.fastq -r DCM2\_trimmed.fastq -o fastq-join\_joined

$ exit

$ mv \*trimmed.fastq fastq-join\_joined; rm \*.zip; cd fastq-join\_joined; fastqc fastqjoin.join.fastq; convert\_format -t fastq -o fastqjoin.join.fasta -f fasta fastqjoin.join.fastq; python trim\_longitudes.py; blastclust -L 0.90 -S 99 -i fastqjoin.join.fasta -o fastqjoin.join.blastclust99.lst -a 4 -p F; python blastclust\_lst2fasta.py; mkdir data; mkdir scripts; mv \*.py scripts; mkdir align; mv \*OTU\* align; mv \*fastq data; mv \*lst data; mv \*.fasta data; mv \*txt data; rm \*.zip; mv \*error\* data; cd align; for i in \*fasta;do muscle -in $i -out $i"\_muscle.fasta"; done

**In yellow are the programs used to run the pipeline**

**These are**

1) Fastqc software (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>)

2) trim\_quality included in ngs\_crumbs (https://github.com/JoseBlanca/seq\_crumbs)

3) Ngs\_crumbs is a more updated version than seq crumbs (<http://bioinf.comav.upv.es/seq_crumbs/>)

4) Join\_paired\_ends.py is included in qiime ( <http://qiime.org/install/install.html>). Alternatively qiime is included in biolinux

5) convert\_format included in ngs\_crumbs (https://github.com/JoseBlanca/seq\_crumbs)

6) trim\_longitudes.py is a home made script and is attached

7) AND 8) blastclust and muscle are included in biolinux

9) blastclust\_lst2fasta.py is a home made script and is attached

**Bioinformatics (methods written up by Santiago Garcia )**

Sequences were demultiplexed into separated fastq files based on their index sequences. Fastqc software (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>) was used to visualize quality and length distribution of each dataset. Both pair-ends from each individual library were trimmed based on quality scores using trim\_quality software included in seq\_crumbs (<http://bioinf.comav.upv.es/seq_crumbs/>) using default parameters. After the trimming, the join\_paired\_ends python script included in Qiime pipeline (Caporaso et al., 2010) was used to combine both paired ends to create a single fastq file. The resultant fastq file from each library was converted into a fasta file using convert\_format software included in seq\_crumbs. A custom python script was used to remove forward and reverse primer sequences from the fasta file, which was clustered into Operational Taxonomic Units (OTUs) using blastclust software (Altschul et al., 1997) with a 99% of barcoding threshold. Resultant list file containing all the OTUs was converted into separate fasta files using a custom python script, and then were aligned using MUSCLE (Edgar, 2004). Singletons were removed from the analysis. A consensus sequence was generated from each OTU and identified using local blast against a custom-curated database and phylogenetics analyses.

**References**

Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25(17): 3389–3402. pmid:9254694 doi: 10.1093/nar/25.17.3389

Andrews S. FastQC: a quality control tool for high throughput sequence data. Available: [http://www.bioinformatics.bbsrc.ac.uk/pr​ojects/fastqc/](http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/). Accessed 2015 Mar 25.

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*(5), 335-336.

Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004;32(5): 1792–1797. pmid:15034147 doi: 10.1093/nar/gkh340