Cockatiel microbiome bioinformatic protocols.

**Author names and affiliations**

Luis David Alcaraza, Apolinar M. Hernándezb, and Mariana Peimbertb,
a Laboratorio Nacional de Ciencias de la Sostenibilidad. Instituto de Ecología. Universidad Nacional Autónoma de México. AP 70-275, Ciudad Universitaria, UNAM, 04510, Cd. Mx., México.

b Departamento de Ciencias Naturales. Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Av. Vasco de Quiroga 4871, Col. Santa Fe Cuajimalpa, 05348, Cd. Mx., México.

1. Preprocessing, assembly, QC, OTU clustering, taxonomy assignments.

Raw MISeq sequences were assembled with Pandaseq and Quality Control with fastx\_tools generating a Sun Grid Engine job to be submitted to our deep-thought cluster, by the following script :

|  |
| --- |
| #!/bin/bashSEQS=/home/run\_ago\_15SALIDAS=/home/salidasBIN=/home/binBIN2=/home/binCOUNT=0for FAA in `cat lista`dolet COUNT=COUNT+1echo "#!/bin/bash" >$\*.$COUNT.screcho "#$ -cwd" >>$\*.$COUNT.screcho "#$ -j y" >>$\*.$COUNT.screcho "#$ -S /bin/bash" >>$\*.$COUNT.screcho "zcat $SEQS/$FAA"\_R1.fastq.gz" | $BIN2/fastx\_trimmer -l 250 > $SEQS/$FAA"tr\_R1.fastq" &" >>$\*.$COUNT.scr echo "zcat $SEQS/$FAA"\_R2.fastq.gz" | $BIN2/fastx\_trimmer -l 250 > $SEQS/$FAA"tr\_R2.fastq"" >>$\*.$COUNT.screcho "$BIN/pandaseq -B -f $SEQS/$FAA"tr\_R1.fastq" -r $SEQS/$FAA"tr\_R2.fastq" -t 0.95 -l 250 -L 470 -o 15 -w $SALIDAS/$FAA"\_4nov15.fasta" -G $SALIDAS/$FAA"-4nov15.log.bz2"" >>$\*.$COUNT.scrdone |

The manual of pandaseq is available here: <http://neufeldserver.uwaterloo.ca/~apmasell/pandaseq_man1.html>

Particularly the following options were modified:

−o minoverlap

Sets the minimum overlap between forward and reverse reads. By default, this is at least one nucleotide of overlap. Raising this number does not generally increase the quality of the output as alignments with small overlaps tend to score poorly and are discarded anyway. This was chosen to be a value of 15, rather than 1.

−O maxoverlap

Sets the maximum overlap between forward and reverse reads. By default, this is the read length. In highly overlapping sequences (i.e., those where the end of one read precede the start of the other), this parameter should be set to the sum of the input reads, or a value larger than that. The maxoverlap was set to default which is the read length

−t threshold

The score, between zero and one, that a sequence must meet to be kept in the output. Any alignments lower than this will be discarded as low quality. Increasing this number will not necessarily prevent uncalled bases (Ns) from appearing in the final sequence. It is also used as the threshold to match primers, if primers are supplied. The default value is 0.6. Our chosen value was 0.95 in a 0-1 scale.

Length was filtered up to a size of 470 bp because the expected amplicon size is around 460 bp from the 16S rRNA gene V3-V4 primers (<http://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/16s/16s-metagenomic-library-prep-guide-15044223-b.pdf>).

Afterwards, quality of the assembly was visualized by FastQC and their html generated reports.

Clustering of the OTUs at the 97% identity was done with cd-hit-est

|  |
| --- |
| $bin/cd-hit/cd-hit-est -c 0.97 -T 0 -i ~/names.fas -o ~/output.clstr |

Then the cluster file is converted to a file that looks exactly the same than the output of QIIME’s (v1.9.0) pick\_otus:

|  |
| --- |
| $perl -pne 's/\t//g;s/^.\*,//g;s/\.\.\..\*$//g;s/\n/\t/g;s/\>Cluster\ /\n/g;s/\>//g; eof && do{chomp; print "$\_ \n"; exit}' output.clstr >cockatiel.otu$sed -i '1d' cockatiel.otu |

With the previous OTU file use QIIME’s pick\_rep\_set.py to select representative *de novo* picked OTUs:

|  |
| --- |
| $pick\_rep\_set.py -i cockatiel.otu -f seqs.fna -o rep\_set1.fna |

Parallel assigning taxonomy:

|  |
| --- |
| $parallel\_assign\_taxonomy\_blast.py -i rep\_set1.fna -o taxonomy -r /qiime/gg\_otus-13\_8-release/rep\_set/97\_otus.fasta -t //qiime/qiime\_software/gg\_otus-13\_8-release/taxonomy/97\_otu\_taxonomy.txt |

BlastTaxonAssigner parameters:

Application:blastn/megablast

Max E value:0.001

Min percent identity:90.0

blast\_db:/home/salidas/perico\_tax/BTA\_L2SRA\_/97\_otus.fasta

id\_to\_taxonomy\_filepath:/home/qiime/1.9.0/software/gg\_otus-13\_8-release/taxonomy/97\_otu\_taxonomy.txt

Number of sequences inspected: 97

Number with no blast hits: 0

Result path: /tmp/3820.1.lda.q/assign-taxMcWg8H

BlastTaxonAssigner parameters:

Application:blastn/megablast

Max E value:0.001

Min percent identity:90.0

Build OTU table:

|  |
| --- |
| $make\_otu\_table.py -i cockatiel.otu -t perico\_tax/perico\_repset\_tax\_assignments.txt -o perico.biom |

Then the otu table is converted to text along with the taxonomy assignments:

|  |
| --- |
| $biom convert --to-tsv -i perico.biom -o ave.txt --table-type "Taxon table" --header-key=taxonomy |

To split the taxonomy from the OTU table:

|  |
| --- |
| $perl -pe 's/\; /\;/g' ave.txt | awk '{print $1,"\t",$NF}' | perl -pe 's/\;/\t/g' >aves.tax |

To clean up the OTU table in txt (tab separated) format:

|  |
| --- |
| $rev ave.txt | cut -f2- | rev >ave.biom.txt |

To prepare OTU table header to be loaded in phyloseq:

|  |
| --- |
| $cat ave.biom.txt | grep -v "biom file" | perl -pe 's/\#OTU\ ID/OTU\_ID/g' >ave.biom.clean.txt |

Cleaning up the OTU table. First step, remove OTUs with no matches to the 16S rRNA gene database (Greengenes):

|  |
| --- |
| $grep "no\_hit" perico.tax.txt | cut -f1 | while read x; do echo "sed -i.bak '/$x/d'" perico.clean.biom.txt; done | bash$grep -v "no\_hit" perico.tax.txt >perico.clean.tax #to clean no hits from the taxonomy file |

Cleaning up the OTU table. Second step, remove Chimeras, identified by ChimeraSlayer:

|  |
| --- |
| $parallel\_align\_seqs\_pynast.py -i ~/ninfa/ninfa.repset.fna -o ~/ninfa/ninfa.align -X ninfa$parallel\_identify\_chimeric\_seqs.py -i ~/ninfa/ninfa.repset\_aligned.fasta -r ~/qiime\_default\_reference/gg\_13\_8\_otus/rep\_set\_aligned/85\_otus.fasta -o ~/ninfa/ninfa.chimera.txt#generate a list of OTUs to be removed$cat ninfa.chimerablast.txt | awk '{print $1}' >remove#Remove Chimeric OTUs$for N in `cat remove`; do egrep -v \"$N \" test >test.t; mv test.t test; done  |

Cleaning up the OTU table. Third step, remove known contaminants (mitochondria, chloroplasts)

|  |
| --- |
| $grep -i "chloroplast" perico.tax.txt | cut -f1 | while read x; do echo "sed -i.bak '/$x/d'" perico.clean.biom.txt; done | bash$grep -iv "chloroplast" perico.tax.txt >perico.clean.tax $grep -i "mitochondria" perico.tax.txt | cut -f1 | while read x; do echo "sed -i.bak '/$x/d'" perico.clean.biom.txt; done | bash$grep -iv "mitochondria" perico.tax.txt >perico.clean.tax  |

\*\*The metadata file order does matter, check the OTU table sample order and the metadata should be exactly in the same order for phyloseq to load them correctly.

2. R analysis

All statistical analysis were done in R. R version 3.2.1 (2015-06-18; x86\_64). With the following packages and versions:

 [1] Rcpp\_0.12.0 locfit\_1.5-9.1 ape\_3.3

 [4] Biostrings\_2.36.1 digest\_0.6.8 foreach\_1.4.2

 [7] biom\_0.3.12 GenomeInfoDb\_1.4.1 plyr\_1.8.3

[10] chron\_2.3-47 futile.options\_1.0.0 acepack\_1.3-3.3

[13] stats4\_3.2.1 RSQLite\_1.0.0 ggplot2\_1.0.1

[16] zlibbioc\_1.14.0 data.table\_1.9.4 annotate\_1.46.1

[19] vegan\_2.3-0 S4Vectors\_0.6.3 rpart\_4.1-10

[22] Matrix\_1.2-4 proto\_0.3-10 splines\_3.2.1

[25] BiocParallel\_1.2.9 geneplotter\_1.46.0 stringr\_1.0.0

[28] foreign\_0.8-66 igraph\_1.0.1 munsell\_0.4.2

[31] BiocGenerics\_0.14.0 multtest\_2.24.0 mgcv\_1.8-12

[34] nnet\_7.3-12 gridExtra\_2.0.0 Hmisc\_3.16-0

[37] IRanges\_2.2.5 codetools\_0.2-14 XML\_3.98-1.4

[40] MASS\_7.3-45 grid\_3.2.1 nlme\_3.1-126

[43] xtable\_1.7-4 gtable\_0.1.2 DBI\_0.3.1

[46] magrittr\_1.5 scales\_0.2.5 stringi\_0.5-5

[49] XVector\_0.8.0 reshape2\_1.4.1 genefilter\_1.50.0

[52] RcppArmadillo\_0.5.300.4 latticeExtra\_0.6-26 futile.logger\_1.4.1

[55] Formula\_1.2-1 lambda.r\_1.1.7 iterators\_1.0.7

[58] tools\_3.2.1 RJSONIO\_1.3-0 ade4\_1.7-2

[61] Biobase\_2.28.0 DESeq2\_1.8.1 parallel\_3.2.1

[64] survival\_2.38-3 AnnotationDbi\_1.30.1 colorspace\_1.2-6

[67] cluster\_2.0.3 GenomicRanges\_1.20.5

The packages must be installed and loaded to reproduce the results.

|  |
| --- |
| #Data loadinglibrary(phyloseq)library(ggplot2)otu <- as.matrix(read.table("perico.clean.biom.txt", header=T, row.names=1))OTU = otu\_table(otu, taxa\_are\_rows=T)taximat = as.matrix(read.table("perico.tax.txt", header=T, row.names=1))taxi=tax\_table(taximat)ninfa = phyloseq (OTU, taxi)Ninfa\_data =read.table(“data.txt”, header=T, row.names=1))sampledata = sample\_data(data.frame(id=ninfa\_data$id, sampling=ninfa\_data$sampling, row.names=sample\_names(ninfa)))ninfa = phyloseq (OTU, taxi, sampledata)ninfaFilter = ninfaninfa0 = genefilter\_sample(ninfaFilter, filterfun\_sample(function(x) x > 5), A =0.5 \* nsamples(ninfaFilter))ninfa1F = prune\_taxa(ninfa0, ninfaFilter)p1 <- plot\_bar(ninfa1F, fill="Phylum", facet\_grid=~Genus)ggsave ("genus\_bar.pdf", p1)library(vegan)data(BCI)otuninfa2 <- t(read.table("perico.clean.biom.txt", header=T, row.names=1))rarecurve(otuninfa2, step=10000)col <- c("blue", "orange", "green")lty <- c("solid", "dashed", "dotdash")pars <- expand.grid(col = col, lty = lty, stringsAsFactors = FALSE)out <- with(pars[1:3, ], rarecurve(otuninfa2, step=10, col = col, lty=lty))pdf("rarefaction\_onlyninfa.pdf")p2 <- plot\_bar(ninfa1F, fill="Genus", facet\_grid=~Order) + theme\_bw()p2ggsave("plot\_barOrder\_genus.pdf", p2)#Venn diagrams / heatmapsinstall.package("venn")install.packages("venn")library(venn)library(limma)vennDiagram(ninfa)ninfaotuninfavennDiagram(otuninfa)pdf("vennNinfa.pdf")top20OTUs <- names(sort(taxa\_sums(ninfa), TRUE)[1:20])ninfa20 <- prune\_taxa(top20OTUs, ninfa)ninfa20otu\_table(ninfa20)ninfa20tax\_table(ninfa20)plot\_bar(ninfa20)plot\_bar(ninfa20, fill="Genus")plot\_heatmap(ninfa1F, "MDS", "bray", )plot\_heatmap(ninfa1F, "MDS", "jaccard", )plot\_heatmap(ninfa1F, "NMDS", "jaccard", )plot\_heatmap(ninfa1F, "PCoA", "jaccard", )plot\_heatmap(ninfa1F, "NMDS", "jaccard", )plot\_heatmap(ninfa1F, "NMDS", "jaccard", )plot\_heatmap(ninfa1F, "PCoA", "jaccard", )plot\_heatmap(ninfa1F, "CCA", "jaccard", )plot\_heatmap(ninfa1F, "RDA", "jaccard", )plot\_heatmap(ninfa1F, "DPCoA", "jaccard", )plot\_heatmap(ninfa1F, "CCA", "jaccard", )plot\_heatmap(ninfa1F, "CCA", "manhattan", )plot\_heatmap(ninfa1F, "CCA", "euclidean", )plot\_heatmap(ninfa1F, "NMDS", "euclidean", )plot\_heatmap(ninfa1F, "NMDS", "manhattan", )plot\_heatmap(ninfa1F, "CCA", "manhattan", )plot\_heatmap(ninfa1F, "NMDS", "manhattan", )plot\_heatmap(ninfa50, "NMDS", "manhattan", )plot\_heatmap(ninfa50, "CCA", "manhattan", )plot\_heatmap(ninfa50, "NMDS", "manhattan", )plot\_heatmap(ninfa20, "NMDS", "manhattan", )plot\_heatmap(ninfa50, "NMDS", "manhattan", )plot\_heatmap(ninfa100, "NMDS", "manhattan", )m= plot\_heatmap(ninfa100, "NMDS", "manhattan", )ggsave ("ninfas\_heatmap100.pdf" , m)m= plot\_heatmap(ninfa1F, "NMDS", "manhattan", )ggsave ("ninfas\_heatmap1F.pdf" , m)ninfa.scale = transform\_sample\_counts(ninfa, function(x) x/sum(x))ninfa.scale = prune\_taxa(taxa\_sums(ninfa.scale)>0, ninfa.scale)#phylum barplots from summary taxa datalibrary(ggplot2)library(phyloseq)phylum <- read.table("otu\_table\_L2.txt", header=T, row.names=1)abund\_table <- t(phylum)x <- abund\_table[,order(colSums(abund\_table), decreasing=T)]for (i in 1:dim(x)[2]){tmp<-data.frame(row.names=NULL,Sample=rownames(x),Taxa=rep(colnames(x)[i],dim(x)[1]),Value=x[,i])if(i==1){df<-tmp} else {df<-rbind(df,tmp)}}dat$V1 <- factor(df$Sample, levels=(unique(rownames(x))))N <- length(df$Taxa)colours <- c("darkorange1", "snow3", "green3", "navy", "greenyellow", "purple", "turquoise4", "skyblue", "violetred", "yellow", "red", "purple4", "turquoise1", "violet", "slateblue4", "olivedrab", "tomato4", "aquamarine")p <- ggplot (df, aes(Sample, Value, fill=Taxa)) + geom\_bar(stat="identity")p2 <- p + scale\_fill\_manual(values=colours[1:(N+1)])p2 <- p2 +theme\_bw() +theme(axis.text.x=element\_text(angle=90, hjust=1, vjust=0.5)) p2 <- p2 + xlab("") + ylab("Relative Frequency")ggsave("bar\_phylum.pdf")#ordinationlibrary(phyloseq)library(ggplot2)aves <- as.matrix(read.table("phyloseq.txt", header=T, row.names=1))OTU = otu\_table(otu, taxa\_are\_rows=T)taximat <- as.matrix(read.table("phyloseq.tax.txt", header=T, row.names=1))taxi = tax\_table(taximat)aves = phyloseq(OTU, taxi)aves\_data=read.table("phyloseq.map.txt", header=T, row.names=1)sampledata = sample\_data(data.frame(id=aves\_data$id, species=aves\_data$species, row.names=sample\_names(aves)))gp.ord <- ordinate(aves, "NMDS", "bray")p1 = plot\_ordination(aves, gp.ord, type = "split", color = "Phylum", shape="species")p1p2 <- p1 + scale\_shape\_manual(values=c(16:25))p2p2 <- p1 + scale\_shape\_manual(values=c(16:25)) + theme\_bw()p2p2 <- p1 + scale\_shape\_manual(values=c(16:25)) + theme\_bw() + label="species"p2 <- p1 + scale\_shape\_manual(values=c(16:25)) + theme\_bw() + label="specie"p1 = plot\_ordination(aves, gp.ord, type = "split", color = "Phylum", shape="species", label = "species")p1p1 = plot\_ordination(aves, gp.ord, type = "split", color = "Phylum", shape="species", label = "id")p1p2 <- p1 + scale\_shape\_manual(values=c(16:25)) + theme\_bw()p2gp.ord2 <- ordinate(aves, "PCoA", "bray")gp.ord3 <- ordinate(aves, "PCoA", "manhattan")p2 = plot\_ordination(aves, gp.ord2, type = "split", color = "Phylum", shape="species", label = "id") + scale\_shape\_manual(values=c(16:25)) + theme\_bw()p2p3 = plot\_ordination(aves, gp.ord3, type = "split", color = "Phylum", shape="species", label = "id") + scale\_shape\_manual(values=c(16:25)) + theme\_bw()p3p1 = plot\_ordination(aves, gp.ord, type = "split", color = "Phylum", shape="species", label = "id") + scale\_shape\_manual(values=c(16:25)) + theme\_bw()p1p1 + geom\_point(size=2)p1 + geom\_point(size=3)ggsave("ordination\_NMDS\_bray.pdf", p1 + geom\_point(size=3) + alpha = 0.6)ggsave("ordination\_NMDS\_bray.pdf", p1 + geom\_point(size=3+ alpha = 0.6))ggsave("ordination\_NMDS\_bray.pdf", p1 + geom\_point(size=3, alpha = 0.6))library(plyr)library(breakaway)peri <- read.table("perico.clean.biom.txt", header=T, sep="\t", row.names=1)breakaway((count(peri$S21)[-1,]))breakaway((count(peri$S22)[-1,]))breakaway((count(peri$S23)[-1,])) |

3. Comparative datasets

The comparative set species were downloaded from the following SRA Accessions/URLs.

**Kakapo:**

[Kakapo captivity study-Adult7\_2012](http://www.ncbi.nlm.nih.gov/sra/SRX385262%5Baccn%5D)

1 LS454 (454 GS FLX) run: 3,015 spots, 1.1M bases, 513,797b downloads

Accession:SRX385262

Select item 552349[ ] 2.

[Kakapo captivity study-Adult6\_2012](http://www.ncbi.nlm.nih.gov/sra/SRX385261%5Baccn%5D)

1 LS454 (454 GS FLX) run: 1,958 spots, 743,227 bases, 351,204b downloads

Accession: SRX385261

Select item 552348[ ] 3.

[Kakapo captivity study-Adult5\_2012](http://www.ncbi.nlm.nih.gov/sra/SRX385260%5Baccn%5D)

1 LS454 (454 GS FLX) run: 1,918 spots, 724,992 bases, 355,746b downloads

Accession: SRX385260

Select item 552347[ ] 4.

[Kakapo captivity study-Adult4\_2012](http://www.ncbi.nlm.nih.gov/sra/SRX385259%5Baccn%5D)

1 LS454 (454 GS FLX) run: 2,720 spots, 1M bases, 482,105b downloads

Accession: SRX385259

**Turkey:**

Data was downloaded form the following link: <http://conservancy.umn.edu/handle/11299/174930>

Reference: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4672264/>

**Chicken:**

Data was downloaded from the following project:

<http://metagenomics.anl.gov/metagenomics.cgi?page=MetagenomeProject&project=6177>

And from the following samples:

<http://metagenomics.anl.gov/metagenomics.cgi?page=DownloadMetagenome&metagenome=4537568.3>

<http://metagenomics.anl.gov/metagenomics.cgi?page=DownloadMetagenome&metagenome=4537569.3>

<http://metagenomics.anl.gov/metagenomics.cgi?page=DownloadMetagenome&metagenome=4537583.3>

**Wild duck (Aythya americana):**

Reference:<http://f1000research.com/articles/2-224/v2#article-reports>

Data are available as supplementary material to the paper: <https://ndownloader.figshare.com/files/1231298>

**Emu**

Reference: <https://www.researchgate.net/profile/Darin_Bennett/publication/249318695_Characterization_of_cecal_microbiota_of_the_emu_%28Dromaius_novaehollandiae%29/links/02e7e520ffa4f5fc10000000.pdf>

Downloaded from: <http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?exp=SRX254725&cmd=search&m=downloads&s=seq>

**Pigs (outgroup):**

Reference: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3277147/>

SRA project: <http://www.ncbi.nlm.nih.gov/sra/SRX038401>

Specific samples:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | [SRS151804](http://www.ncbi.nlm.nih.gov/biosample?term=SRS151804) : [pig 26-0](http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=search&amp;m=search&amp;s=seq&amp;exp=SRX038401&amp;member=pig%2026-0) | 8,411 | 2.3M |
| 2 | [SRS151805](http://www.ncbi.nlm.nih.gov/biosample?term=SRS151805) : [pig 35-0](http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=search&amp;m=search&amp;s=seq&amp;exp=SRX038401&amp;member=pig%2035-0) | 11,532 | 3.1M |
| 3 | [SRS151806](http://www.ncbi.nlm.nih.gov/biosample?term=SRS151806) : [pig 37-0](http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=search&amp;m=search&amp;s=seq&amp;exp=SRX038401&amp;member=pig%2037-0) | 9,860 | 2.7M |

For each sequence of the comparative dataset each study was processed individually by the following commands:

|  |
| --- |
| for N in \*.fastq; do fastq\_quality\_filter -Q 33 -p 97 -q 20 -i $N | fastx\_trimmer -m 100 | fastq\_to\_fasta -o $N.filter.fasta; done |

Each file was renamed in a sequential header with a number for each sequence, in a way similar of QIIME’s output of split libraries with the following perl script:

|  |
| --- |
| #!/usr/bin/perl# este script cambia cada uno de los identificadores de un archivo fasta a >prefix\_# donde prefix es el# primer argumento del script y # es una numeración de cada una de las secuencias que componen el archivo el segundo argumento es el nombre del archivo fasta a renombrar.# este script es útil si se quiere usar QIIME en archivos fasta desmultiplexados o individuales.# uso: ./header.fasta.numbers.pl prefix nombre\_del\_archivo.fasta# Luis David Alcaraz 2013-04-11my $prefix = $ARGV[0]; chomp $prefix;my $f = 1;my $fasta\_file = $ARGV [1]; chomp $fasta\_file;my $fh;open($fh, $fasta\_file) or die "can't open $fasta\_file: $!\n";open(OUT, ">$fasta\_file.numbered.fas") || die "can't open $fasta\_file.numbered.fas\n";my %sequence\_data;while (read\_fasta\_sequence($fh, \%sequence\_data)) { print OUT ">$sequence\_data{header}\n$sequence\_data{seq}\n";}close $fh;close OUT;sub read\_fasta\_sequence { my ($fh, $seq\_info) = @\_; $seq\_info->{seq} = undef; # clear out previous sequence # put the header into place $seq\_info->{header} = $seq\_info->{next\_header} if $seq\_info->{next\_header}; my $file\_not\_empty = 0; while (<$fh>) { $file\_not\_empty = 1; next if /^\s\*$/; # skip blank lines chomp;  if (/^>/) { # fasta header line my $h = $\_;  $h =~ s/>/$prefix\\_$f\ /; $f++; if ($seq\_info->{header}) { $seq\_info->{next\_header} = $h; return $seq\_info;  }  else { # first time through only $seq\_info->{header} = $h; }  }  else {  s/\s+//; # remove any white space s/\n\n/\n/; $seq\_info->{seq} .= $\_; }  }  if ($file\_not\_empty) { return $seq\_info; }  else { # clean everything up $seq\_info->{header} = $seq\_info->{seq} = $seq\_info->{next\_header} = undef; return;  } } |

Each species was concatenated independently (3 individuals per sample, all but wild duck).

Example:

|  |
| --- |
| cat chicken/chickens.fna kakapo/kakapo.fna ninfa/perico.nov15.fasta turkey/turkey.fas wild\_duck/duck.fna >comp-mic\_aves.fna |

There is need to generate a parameter\_pick for pick\_closed\_reference.otus.py with the following information to enable reverse strand matching:

|  |
| --- |
| $echo “pick\_otus:enable\_rev\_strand\_match True” -p parameter\_pick |

We made a regular QIIME’s pick\_closed\_reference.otus.py assignment, with the previous files :

|  |
| --- |
| $pick\_closed\_reference\_otus.py -f -i ~/comp-mic\_aves.fna -p ~/parameter\_pick -a -o ~/salidas/mic\_aves/ |

Pick closed otu parameters employed are described below:

-r REFERENCE\_FP, --reference\_fp=REFERENCE\_FP

The reference sequences [default:

/home/qiime/1.9.0/lib/python2.7/site-package

s/qiime\_default\_reference/gg\_13\_8\_otus/rep\_set/97\_otus

.fasta]. NOTE: If you do not pass -r to this script,

you will be using QIIME's default reference sequences.

In this case, QIIME will copy the corresponding

reference tree to the output directory. This is the

tree that should be used to perform phylogenetic

diversity analyses (e.g., with

core\_diversity\_analyses.py).

-p PARAMETER\_FP, --parameter\_fp=PARAMETER\_FP

path to the parameter file, which specifies changes to

the default behavior. See

http://www.qiime.org/documentation/file\_formats.html

#qiime-parameters . [if omitted, default values will

be used]

-t TAXONOMY\_FP, --taxonomy\_fp=TAXONOMY\_FP

the taxonomy map [default:

/home/qiime/1.9.0/software/gg\_otus-

13\_8-release/taxonomy/97\_otu\_taxonomy.txt]

-s, --assign\_taxonomy

Assign taxonomy to each sequence using

assign\_taxonomy.py (this will override --taxonomy\_fp,

if provided) [default: False]

-f, --force Force overwrite of existing output directory (note:

existing files in output\_dir will not be removed)

[default: none]

-w, --print\_only Print the commands but don't call them -- useful for

debugging [default: False]

-a, --parallel Run in parallel where available [default: False]

-O JOBS\_TO\_START, --jobs\_to\_start=JOBS\_TO\_START

Number of jobs to start. NOTE: you must also pass -a

to run in parallel, this defines the number of jobs to

be started if and only if -a is passed [default: 200]

--suppress\_taxonomy\_assignment

skip the taxonomy assignment step, resulting in an OTU

table without taxonomy (this will override

--taxonomy\_fp and --assign\_taxonomy, if provided)

[default: False]

REQUIRED options:

The following options must be provided under all circumstances.

-i INPUT\_FP, --input\_fp=INPUT\_FP

the input sequences [REQUIRED]

-o OUTPUT\_DIR, --output\_dir=OUTPUT\_DIR

the output directory [REQUIRED]

Comparative samples, beyond pick\_closed\_otus

Due to the poor performance of the QIIME’s *pick\_closed\_otus.py* script on OTU assignments for cockatiels, we decided to perform the very same analysis described in previous sections (1) and (2) of this log to process each of the animal species individually.

Briefly:

1. QC filtering
2. OTU clustering
3. OTU taxonomic assignment
4. OTU chimera, and contaminant removal
5. OTU table generation with taxonomy
6. Perform the summarize taxa script for each individual study (animal):

|  |
| --- |
| $for N in `cat list`; do echo "summarize\_taxa.py -i $N.biom -o $N.tax.sum"; done |

1. Merge the phyla level taxonomic summaries (Level 2) of the previous step by the taxonomic assignment. Until get a tab separated file of the following shape:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | kakapo1 | kakapo2 | kakapo3 | ninfa1 | ninfa2 | ninfa3 | … |
| No\_blast\_hit\_Other | 0.001604493 | 0.009179575 | 0.002186987 | 0.000684932 | 0.000360227 | 0.000334775 | … |
| None\_Other | 0.126754914 | 0.193344808 | 0.086386003 | 0 | 0 | 0 | … |
| Unknown | 0 | 0 | 0 | 0 | 0 | 0 | … |
| Acidobacteria | 0 | 0 | 0 | 2.21E-005 | 2.06E-005 | 0 | … |
| Actinobacteria | 0.016446049 | 0.001147447 | 0 | 0.000198851 | 0.000494025 | 4.65E-005 | … |
| Armatimonadetes | 0 | 0 | 0 | 0 | 0 | 0 | … |
| BHI80-139 | 0 | 0 | 0 | 0 | 0 | 0 | … |
| BRC1 | 0 | 0.000573723 | 0 | 0 | 0 | 0 | … |
| Bacteroidetes | 0 | 0.025243832 | 0.006014215 | 0.000419797 | 0.000535194 | 0.000158088 | … |
| Chloroflexi | 0 | 0.000573723 | 0 | 0 | 0 | 9.30E-006 | … |
| Cyanobacteria | 0.001604493 | 0.016064257 | 0.019682887 | 8.84E-005 | 0.003838989 | 0.001711071 | … |
| Deferribacteres | 0 | 0 | 0 | 0 | 0 | 0 | … |
| Elusimicrobia | 0 | 0 | 0 | 0 | 0 | 0 | … |

1. Use table (step 7) as input in R to elaborate barplots:

|  |
| --- |
| library(ggplot2)library(phyloseq)phylum <- read.table("otu\_table\_L2.txt", header=T, row.names=1)abund\_table <- t(phylum)x <- abund\_table[,order(colSums(abund\_table), decreasing=T)]for (i in 1:dim(x)[2]){tmp<-data.frame(row.names=NULL,Sample=rownames(x),Taxa=rep(colnames(x)[i],dim(x)[1]),Value=x[,i])if(i==1){df<-tmp} else {df<-rbind(df,tmp)}}dat$V1 <- factor(df$Sample, levels=(unique(rownames(x))))N <- length(df$Taxa)colours <- c("darkorange1", "snow3", "green3", "navy", "greenyellow", "purple", "turquoise4", "skyblue", "violetred", "yellow", "red", "purple4", "turquoise1", "violet", "slateblue4", "olivedrab", "tomato4", "aquamarine")p <- ggplot (df, aes(Sample, Value, fill=Taxa)) + geom\_bar(stat="identity")p2 <- p + scale\_fill\_manual(values=colours[1:(N+1)])p2 <- p2 +theme\_bw() +theme(axis.text.x=element\_text(angle=90, hjust=1, vjust=0.5))p2 <- p2 + xlab("") + ylab("Relative Frequency")ggsave("bar\_phylum.pdf") |