metaT

Anna Heintz-Buschart

Apr 27, 2022

SECTIONS:

The analysis of metatranscriptomes shares much of its concepts and techniques with metagenomics. In this session, you have the chance to compare the two omics levels at the whole-community level and at the level of a MAG.

Going through the parts under the 'Sections' header, you can perform some analyses on the HPC. The 'Processing steps' header holds example rules for handling metaT data. They are too slow to run with any fun during the course - we've prepared the data for you using these methods.

CHAPTER

ONE

PART 0: RECAP: WHAT'S HAPPENED WITH THE DATA SO FAR?

We're starting this practical session off with an annotated assembly and some metatranscriptomic reads. You've previously seen how genes are annotated on (metagenomic) contigs. This information forms the basis of the interpretation of metatranscriptomics.

The annotated assembly we are using in this practical is based on metagenomics and metatranscriptomics short reads. As a first step, they were trimmed to remove remainders of adapters and filtered for quality.

In the next step, certain uninformative reads were removed. In our case, we removed contaminant sequences, like the phiX174 genome. This step could also target host genomes.

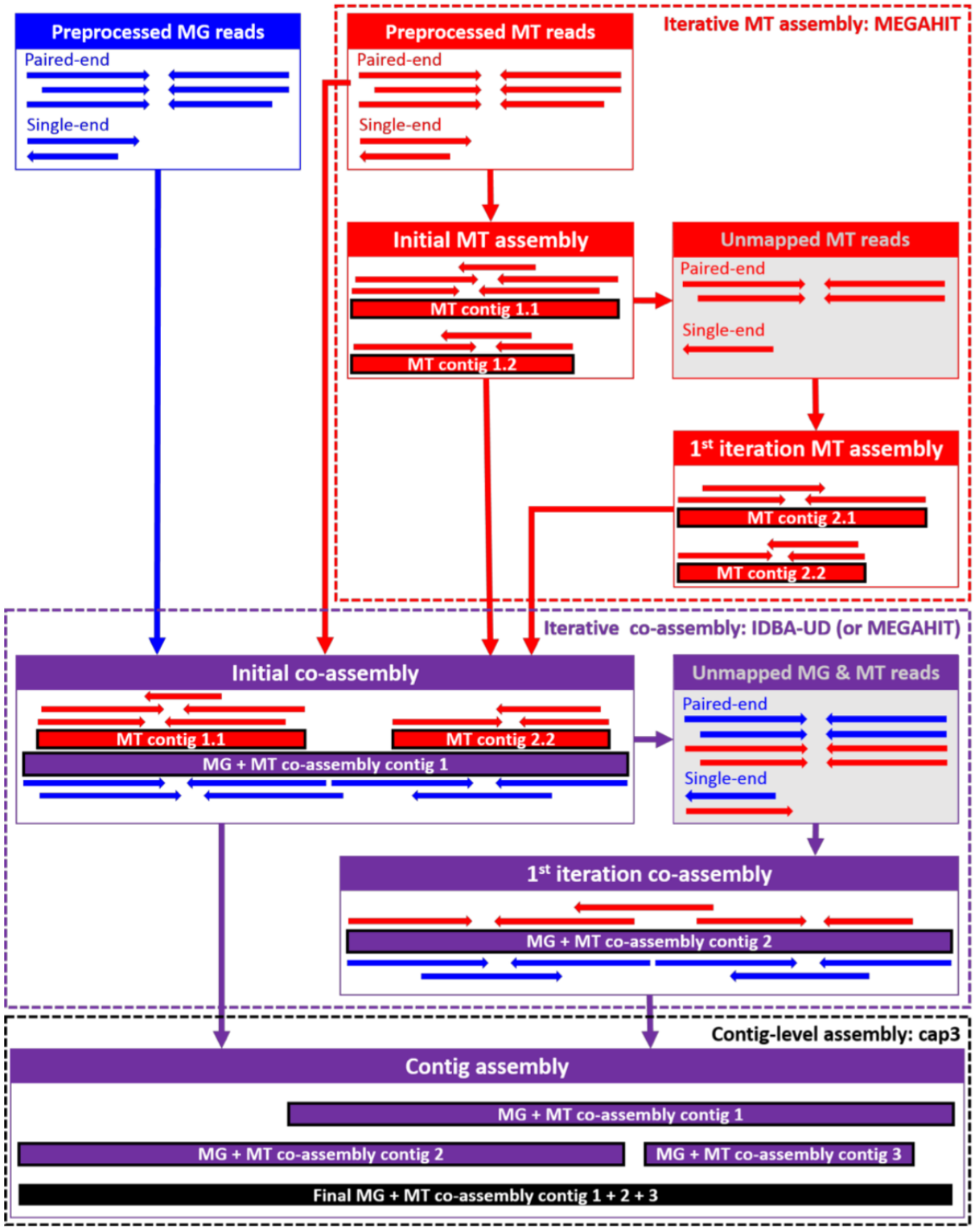
---

metaT-specific filtering

If you sequence metatranscriptomic reads after ribo-depletion, there will always be some rRNA-fragments left over. These are not representatitve of the metatranscriptomics information, so it's best practice to remove them from the metaT reads, too.

We achieved rRNA removal by mapping against an rRNA database.

The next step was assembly. In our case, we use an iterative (step-by-step) hybrid assembly of metatranscriptomics and metagenomics reads: the assembly starts with a metatranscriptomics assembly. The rationale is that there are some strongly expressed (and hence deeply covered) genes and operons which are better represented by the metatranscriptome than the metagenome. The assembled transcripts form one input to the hybrid assembly, together with the metagenomics and metatranscriptomics reads. In our case, we then map the reads back to the hybrid-assembled contigs and extract reads that do not map, i.e. that are not used/represented by the assembly. We assemble these like the original set of reads and then merge both sets of contigs.



This approach has been shown to lead to higher mapping rates, especially for the metatranscriptomic reads, than metagenomics-only assembly. There are however also situations, where you'd preferably assemble the metagenomics reads only – for example, if you expect to have intron-rich metatranscriptomes or if your metatranscriptome is sequenced too shallow.

What if I have only metaT reads?

It's possible to assembly metatranscriptomic reads on their own, of course. We will look at what the metatranscriptomics assembly recovers *later*.

In any case, the next step will be to call genes and annotate them. We have used prokka and HMMER3 for this purpose. In this hands-on session, we will use the KEGG orthologous group (KO) annotation. The annotation data is contained in a GFF file.

The other information that is needed for this session is the read alignment information, that is which read maps to the assembly at which position. We've generated this using BWA-MEM.

CHAPTER

TWO

PART 1: GENERATE GENE/FUNCTIONAL PROFILES

For this session, please access the HPC with X11-forwarding. (In mobaXterm, you need to check X11-Forwarding in the SSH session settings; on unix-systems, you add `-Y` to the ssh command). If you don't have either, you may also plot into a file which you open on your own compluter.

The data needed for this hands-on session sits in `/work/projects/embomicrobial2020/data/`.

```
cd /work/projects/embomicrobial2020/data/metaT
ls -ltrh
```

We have the data for a small number of samples from the time series from this article in this directory (labeled with the date). The first few steps of this hands-on session deal with per-sample analysis. Please pick a (sort of) random sample:

```
myIndex=$(($RANDOM % 5))
samples=("2011-07-08" "2011-08-29" "2011-10-12" "2011-11-29" "2012-01-19" "2012-03-08")
mySample=${samples[$myIndex]}
echo $mySample
```

2.1 Extracting per-feature metatranscriptomics reads

As the first part of the session, we will use feature counts to extract the number of reads per called gene or per KO. The output of this hands-on will be written to the `/scratch` directory:

```
cd /scratch/users/$USER
mkdir metaT
cd metaT
```

The idea of featureCounts is very simple: you supply the GFF file, which contains the positions of the genes (or open reading frames), and the .bam file, which contains for all the reads the positions where they map. Feature counts just counts how many reads map on each of the open reading frames. The numbers of reads per open reading frame can also be aggregated at the level of the same annotations, e.g. KOs.

First, let's calculate the number of reads mapping per open reading frame:

```
conda activate /work/projects/embomicrobial2020/envs/featureCounts/
featureCounts -p -O -t CDS -g ID -o mt.CDS_counts.tsv -s 2 -a /work/projects/
↪embomicrobial2020/data/metaT/annotations/$mySample/$mySample.annotation_CDS_RNA_hmms.
↪gff -T 1 /work/projects/embomicrobial2020/data/metaT/mapping/$mySample/mt.reads.sorted.
↪bam
featureCounts -p -O -t CDS -g ID -o mg.CDS_counts.tsv -s 0 -a /work/projects/
↪embomicrobial2020/data/metaT/annotations/$mySample/$mySample.annotation_CDS_RNA_hmms.
↪gff -T 1 /work/projects/embomicrobial2020/data/metaT/mapping/$mySample/mg.reads.sorted.
↪bam
```

(continues on next page)

(continued from previous page)

Check `featureCounts -h` to understand the difference in the arguments of the two commands.

As you can also see in the help, you could also run both analyses in one go and write the result to a single file.

```
conda activate /work/projects/embomicrobial2020/envs/featureCounts/
featureCounts -p -O -t CDS -g ID -o mgmt.CDS_counts.tsv -s 0,2 -a /work/projects/
↪embomicrobial2020/data/metaT/annotations/$mySample/$mySample.annotation_CDS_RNA_hmms.
↪gff -T 1 /work/projects/embomicrobial2020/data/metaT/mapping/$mySample/mg.reads.sorted.
↪bam /work/projects/embomicrobial2020/data/metaT/mapping/$mySample/mt.reads.sorted.bam
```

Your output (in `mt.CDS_counts.tsv` and `mg.CDS_counts.tsv`) contains a line with the full call to featureCounts and a header line and then one line for all features in the GFF file. As you can see in the header line, the first field in every line gives the actual feature and the last column contains the number of reads per feature.

To do the same for the open reading frames that were annotated with a KO, we first need to filter the GFF file, because featureCounts is a bit unflexible with its input.

```
grep "KEGG=" /work/projects/embomicrobial2020/data/metaT/annotations/$mySample/$mySample.
↪annotation_CDS_RNA_hmms.gff >> $mySample.annotation_KEGG.gff
```

Then, you can run the featureCounts command using KEGG as the `-g` argument as above, but with the filtered GFF file as input. Name the output `mgmt.KEGG_counts.tsv`, if you want to follow the next steps of the tutorial closely.

Comment

Files with counts generated in this way can be used for differential abundance analysis, e.g. by DESeq2.

CHAPTER

THREE

PART 2: COMPARE METAT & METAG AT GENE LEVEL

In this and the next part of the tutorial, you'll quickly visualize the results from the last step.

First, check how many open reading frames there are in your sample:

```
grep -P "\tCDS\t" -c /work/projects/embomicrobial2020/data/metaT/annotations/$mySample/
↪$mySample.annotation_CDS_RNA_hmms.gff
```

Note this number.

Then have a look at the correlation between metagenomics and metatranscriptomics read mapping to all the open reading frames in the dataset. Probably the easiest option is to plot this in R (R is installed in the featureCounts environment).

```
R
```

Within R, you can simply read the featureCounts results.

```
mgmt <- read.delim("mgmt.CDS_counts.tsv",skip=1,stringsAsFactors=F)[,c(1,7,8)]
colnames(mgmt)[2:3] <- c("metaG","metaT")
```

Warning: Make sure that you entered the metaG input first into the featureCounts call, in order for the above column names assignment to be correct. You can check the top of the mgmt file, using the `head` function (either in the shell or in R).

You can first compare the number of genes that are in your featureCounts output to the number of CDS' in the GFF file.

```
dim(mgmt)
```

Then you can check, how many genes are not mapped by either metagenomics or metatranscriptomics reads:

```
length(which(mgmt$metaG==0 & mgmt$metaT==0))
```

You can also see, how many genes are detected in only metagenomics:

```
length(which(mgmt$metaG>0 & mgmt$metaT==0))
```

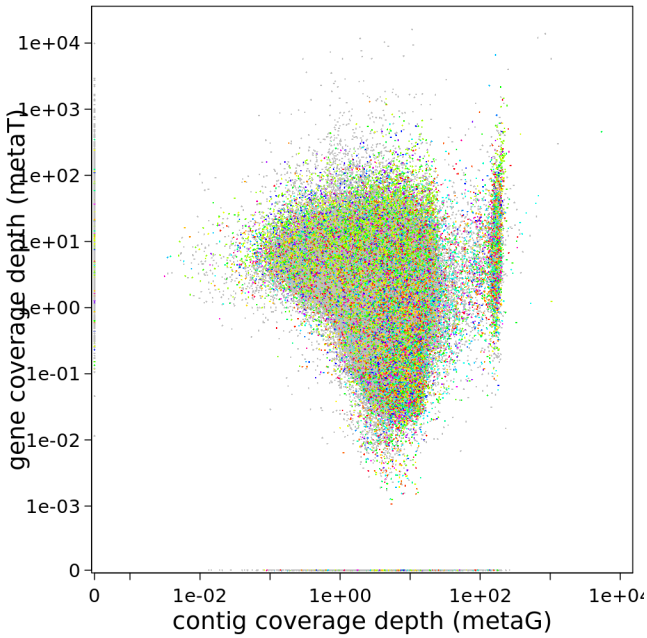
Can you add the code to figure out the number of genes that are only detected in metatranscriptomics? And the numbers of genes that are covered by both kinds of reads?

You can also visualize these relationships.

```
 min0 <- function(x){
   min(x[x>0])
 }

 png("scatterplot.genes.png",width=8.5,height=8.5,units = "cm",res=300,pointsize=7)
 par("mar"=c(2.9,4,0.5,0.5),tcl="-0.3",mgp=c(1.7,0.5,0),lwd=0.5)
 plot(mgmt$metaG,mgmt$metaT,
     pch=16,cex=0.2,log="xy",
     las=1,xlim=c(min0(mgmt$metaG)/11,max(mgmt$metaG)/1.2),
     xaxs="i",
     ylim=c(min0(mgmt$metaT)/11,max(mgmt$metaT)/1.2),
     yaxs="i",
     ylab="",
     xlab="gene coverage (metaG reads)",cex.lab=9/7,axes=F)
points(mgmt$metaG[mgmt$metaT==0],
       rep(min0(mgmt$metaT)/10,length(which(mgmt$metaT==0))),
       pch=16,cex=0.2)
points(rep(min0(mgmt$metaG)/10,length(which(mgmt$metaG==0))),
       mgmt$metaT[mgmt$metaG==0],
       pch=16,cex=0.2)
mtext("gene coverage (metaT reads)",2,2.8,cex=9/7)
axis(1,
     at=c(min0(mgmt$metaG)/10,10^c(-3:8)),
     labels=c(0,format(10^c(-3:8),scientific=T,digits = 2)),lwd=0.5,las=1)
axis(2,at=c(min0(mgmt$metaT)/10,10^c(-3:8)),
     labels=c(0,format(10^c(-3:8),scientific=T,digits = 2)),lwd=0.5,las=1)
box(bty="o")
dev.off()
```

In the IMP3-report, the same visualization is given, but here the average coverage depth is used, and the genes are coloured by function:



You can find the files containing the coverage information in `/work/projects/embomicrobial2020/data/`

`$mySample/run1hybrid/Stats/mg/annotation/mg.gene_depth.txt`. Visualize this, too, if you want some practice with R.

CHAPTER

FOUR

PART 3: COMPARE METAT & METAG AT KO LEVEL

Task

Try to do a similar visualization at the aggregated KO-level.

Check out the KOs with an usually high metaT:metaG ratio. What kinds of functions do you find?

CHAPTER

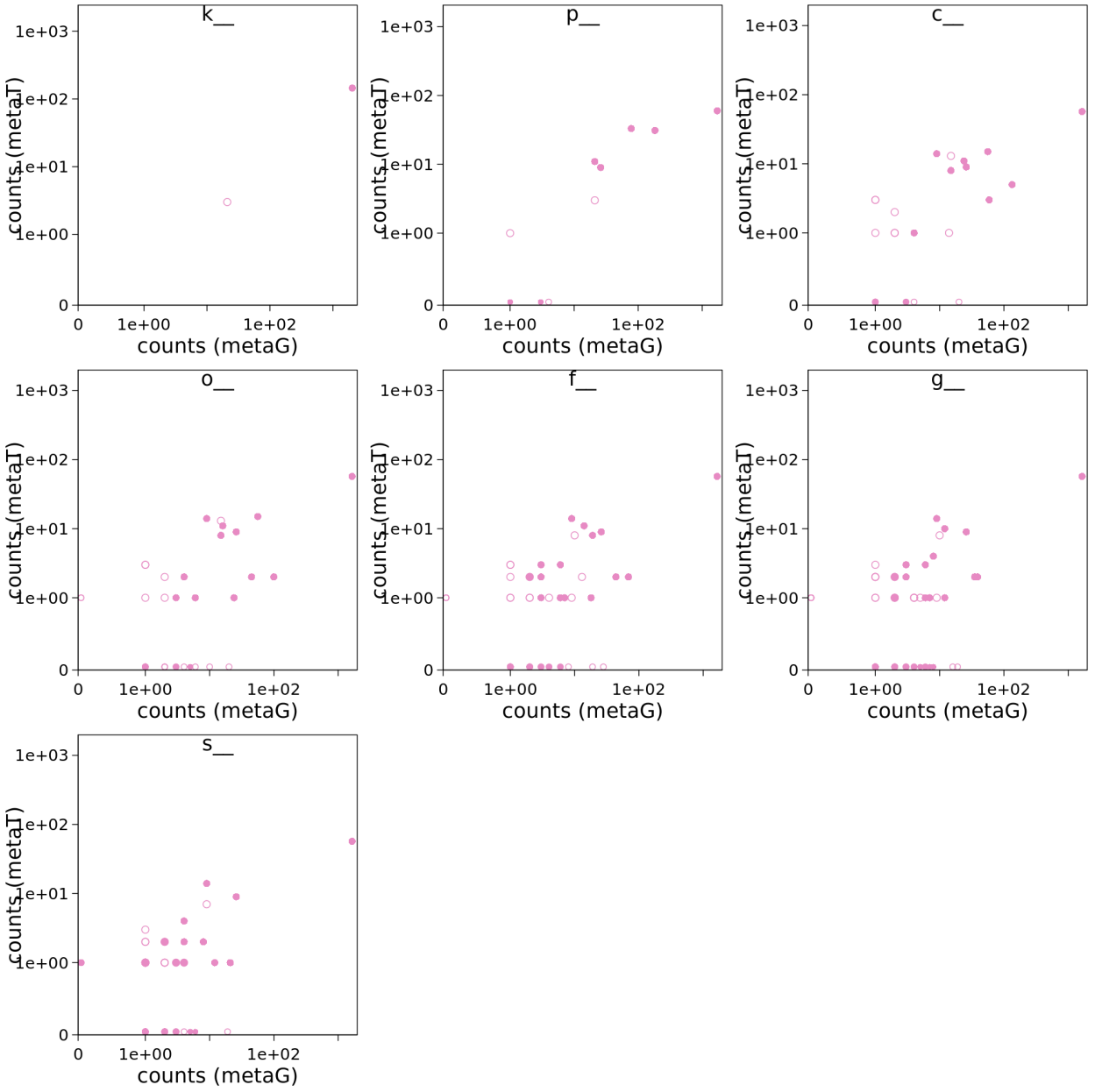
FIVE

PART 4: COMPARE METAT & METAG AT MOTU LEVEL

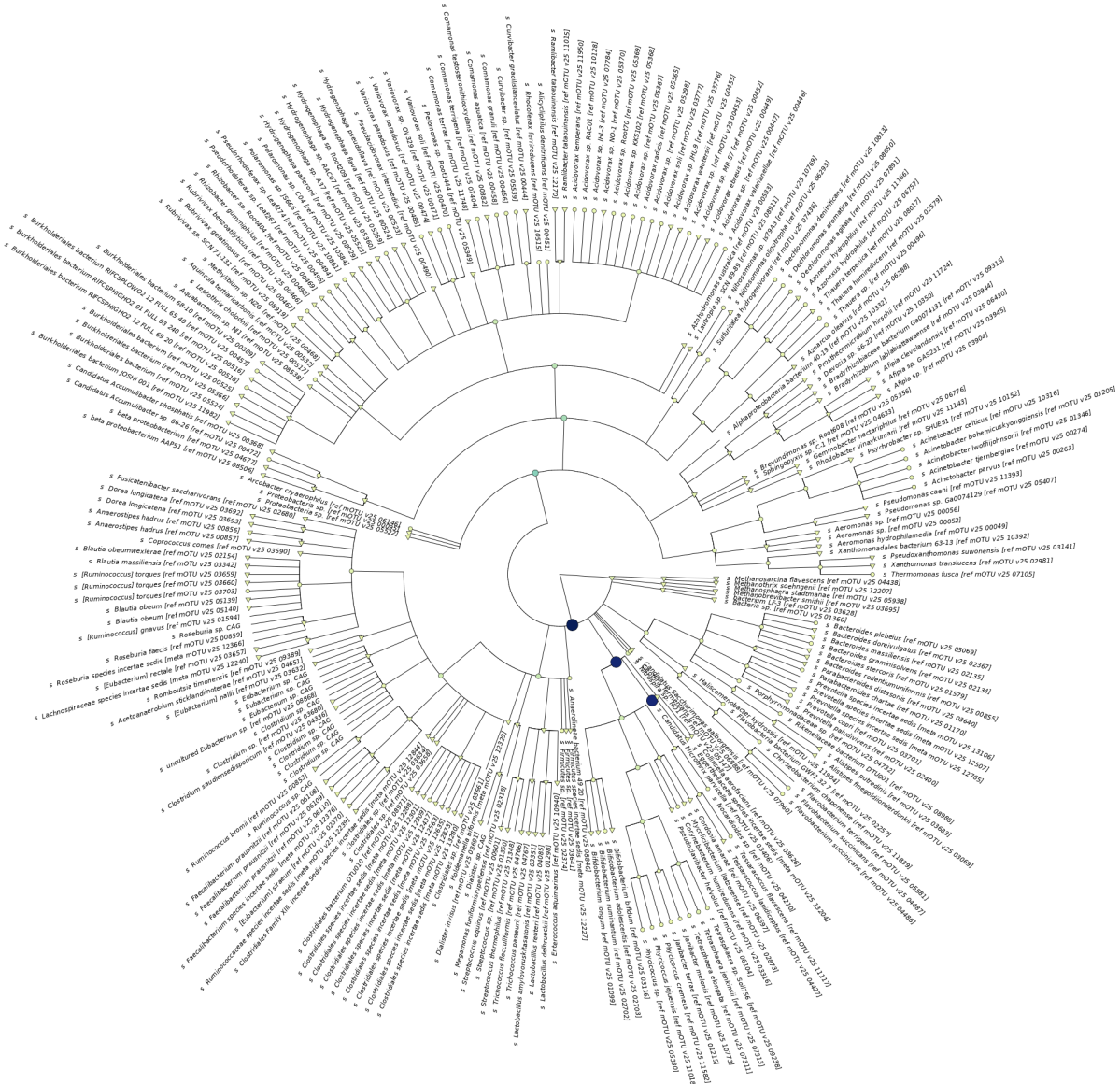
Because the mOTUs2 method uses the same housekeeping genes to estimate the abundance of taxa in a metagenomics sample, it can also be used to profile a metatranscriptome.

You can find the output of mOTUs2 at metagenomic and metatranscriptomic level in `/work/projects/embomicrobial2020/data/metaT/taxonomy/$mySample`. Try, if you can visualize this in the same way as the KO data in the previous part.

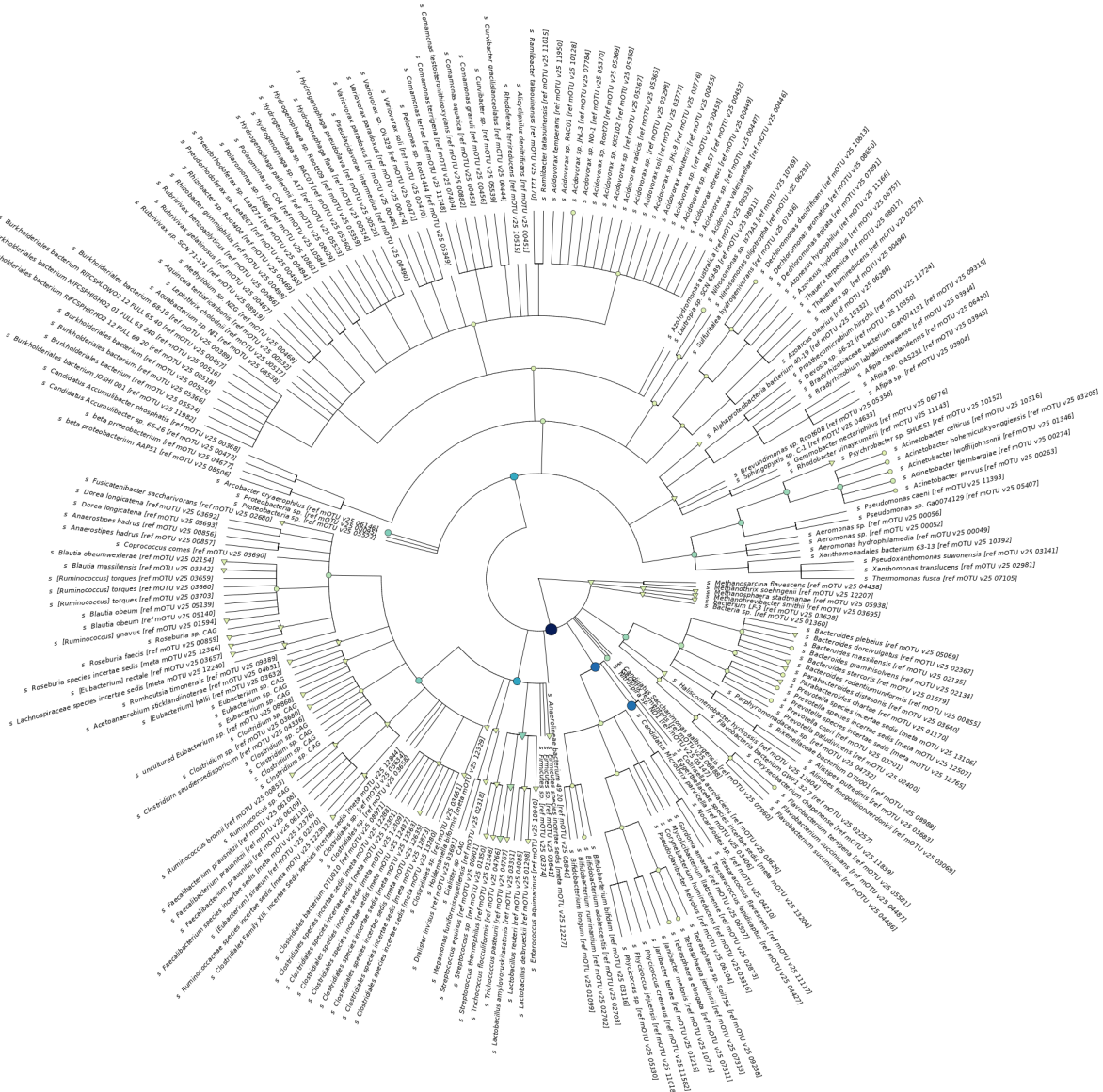
Here is an example with scatter plots at multiple taxonomic ranks:



In this tree-visualization, which is part of IMP3's report, you can also see the names and phylogeny of the taxa. The nodes are coloured according to the relative abundance.



If you compare this to metatranscriptomic output below, you see some subtle differences:



Comment

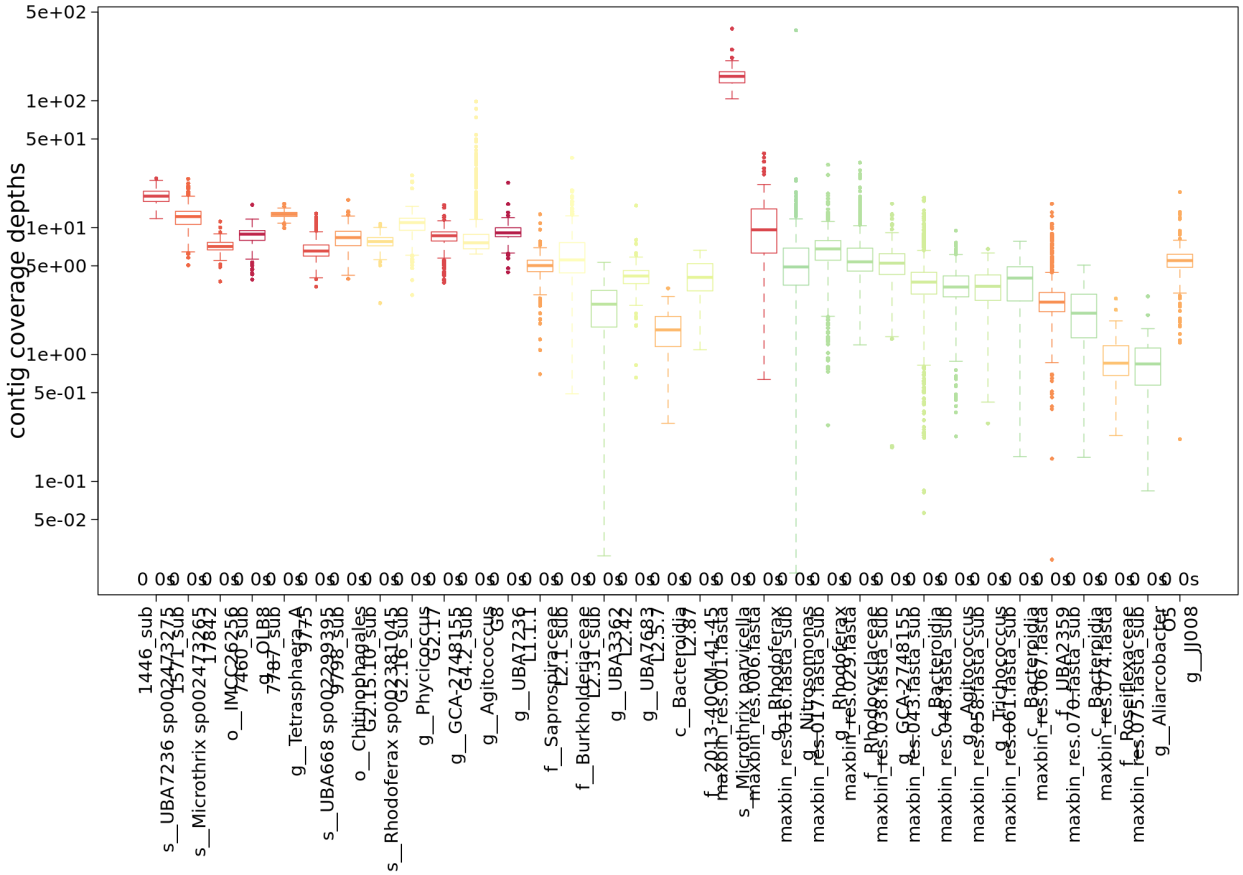
With a few lines of code, you can concatenate data from several samples to do statistical analyses on the mOTU-profiles. Since this taxonomic profile data does not meet all the assumptions of the DESeq2 test, it's better to analyse them with statistical packages developed for microbiome data, e.g. ANCOM-II or corncob.

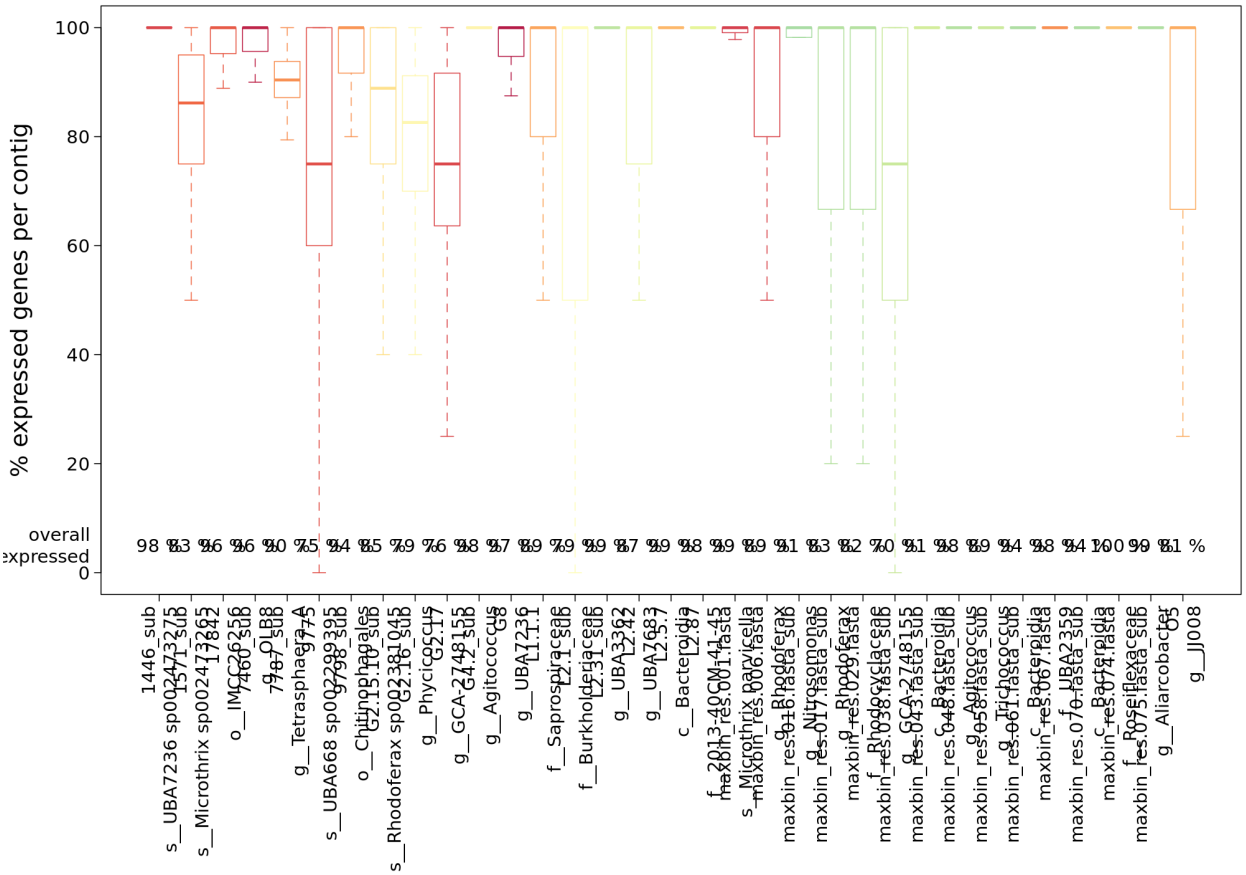
CHAPTER

SIX

PART 5: EXTRACT INFORMATION FOR ONE MAG

An informative analysis is to look at the transcript profiles for single MAGs. IMP gives a small summary of the coverage with metagenomics and metatranscriptomics reads of the MAGs in a sample:





For this hands-on session, we've extracted the contigs and some annotation as well as read-mapping data for a single, dominant MAG from the complete IMP3 output. In the figures above, it is the first bin represented by a red box in the centre of the plots.

Here is the code we have used to extract the GFF information:

```
for sample in 2011-07-08  2011-08-29  2011-10-12  2011-11-29  2012-01-19  2012-03-08
do
     cd /work/projects/embomicrobial2020/data/metaT/annotations/$sample
     bin=`cat /work/projects/embomicrobial2020/data/metaT/binning/$sample/bins`
     samplegff=$sample.annotation_CDS_RNA_hmms.gff
     bingff=$bin.$sample.annotation_CDS_RNA_hmms.gff

     for contig in `cat /work/projects/embomicrobial2020/data/metaT/binning/$sample/$bin.
↪ids`
     do
             grep $contig $samplegff
     done > $bingff
done
```

And this is the code for the extraction of only those reads which map to the contigs in the MAG.

```
conda activate /work/projects/embomicrobial2020/tools/IMP/conda/bff6861f
for sample in 2011-07-08  2011-08-29  2011-10-12  2011-11-29  2012-01-19  2012-03-08
do
     cd /work/projects/embomicrobial2020/data/metaT/binning/$sample
```

(continues on next page)

(continued from previous page)

```
    bin=`cat bins`
    grep ">" $bin.contigs.fa | sed "s/>//" >$bin.ids
    samtools view -bS ../../mapping/$sample/mg.reads.sorted.bam `cat $bin.ids` >$bin.mg.
↪bam
    samtools view -bS ../../mapping/$sample/mt.reads.sorted.bam `cat $bin.ids` >$bin.mt.
↪bam
    samtools index $bin.mg.bam
    samtools index $bin.mt.bam
done
```

Exercise

If you have time, extract the commands from our little loops to run the commands to write the data for your sample to your `/scratch/users/$USER` directory. You can also practice to make a snakemake rule.

CHAPTER

SEVEN

PART 6: METAT-ONLY ASSEMBLY?

You may have only metatranscriptomic reads and wonder what you can do with them. In this part of the hands-on session, you can compare a metatranscriptomics-only assembly with the integrated metagenomics and metatranscriptomics assembly. We have picked the most dominant MAG from the dataset as an example.

Metatranscriptomics-only assemblies can be found in `/work/projects/embomicrobial2020/data/metaT/assemblies/$mySample/final.contigs.fa`.

Information for the example MAG is in `/work/projects/embomicrobial2020/data/metaT/binning/$mySample/`.

To compare the contigs from the metaT-only assembly to the MAG, you can use nucmer (make sure you request sufficient cpus (8) in your interactive slurm job):

```
conda activate /work/projects/embomicrobial2020/envs/mummer4
cpus=8
sample=2011-07-08

ref=final.contigs.fa
bin=`cat /work/projects/embomicrobial2020/data/metaT/binning/$sample/bins`
qry=$bin.contigs.fa
prefix=$sample.genomalign

mkdir metaTGvsmetaT
cd metaTGvsmetaT

nucmer -t $cpus -p $prefix /work/projects/embomicrobial2020/data/metaT/binning/$sample/
↪$qry /work/projects/embomicrobial2020/data/metaT/assemblies/$sample/$ref

delta=$prefix.delta
dnadiff -p $prefix -d $delta
show-coords $prefix.1delta >$prefix.1delta.coords
```

Here is a description of the output in `$prefix.1delta.coords` from the nucmer manual: [S1] start of the alignment region in the reference sequence [E1] end of the alignment region in the reference sequence [S2] start of the alignment region in the query sequence [E2] end of the alignment region in the query sequence [LEN 1] length of the alignment region in the reference sequence [LEN 2] length of the alignment region in the query sequence [% IDY] percent identity of the alignment [% SIM] percent similarity of the alignment (as determined by the BLOSUM scoring matrix) [% STP] percent of stop codons in the alignment [LEN R] length of the reference sequence [LEN Q] length of the query sequence [COV R] percent alignment coverage in the reference sequence [COV Q] percent alignment coverage in the query sequence [FRM] reading frame for the reference and query sequence alignments respectively [TAGS] the reference and query FastA IDs respectively. All output coordinates and lengths are relative to the forward strand of the reference DNA sequence.

Excercise

Adapt the nucmer, dnadiff and show-coords to run the same comparison in the opposite direction (metaTGvsmetaT).

You can already see from the numbers in `$sample.genomalign.report` that only a part of the MAG that was assembled from metaG and metaT reads is also assembled from the transcriptome (the 53.3175% in the example below):

```
[Bases]
TotalBases                  3252111             59671729
AlignedBases       1733945(53.3175%)     1806914(3.0281%)
UnalignedBases     1518166(46.6825%)   57864815(96.9719%)
```

Excercise

For the next step, extract the alignment information as .bed format, as shown in the code below.

```
paste <(cut -f 12 /work/projects/embomicrobial2020/data/metaT/binning/$sample/
↪genomealignment/metaTGvsmetaT/$sample.genomalign.1coords) <(cut -f 1,2 /work/projects/
↪embomicrobial2020/data/metaT/binning/$sample/genomealignment/metaTGvsmetaT/$sample.
↪genomalign.1coords)  >> $sample.genomalign.bed
```

Copy the resulting file to your computer.

CHAPTER

EIGHT

PART 7: VISUALIZE GENE EXPRESSION IN IGV

For this part of the hands-on, you need to download some data to your computer to visualize it locally. These files are:

- the contigs of you genome of interest `/work/projects/embomicrobial2020/data/metaT/binning/<SAMPLE>/<MAG>.fasta.contigs.fa`
- the .bed file you created in the last step
- the alignment files and their indices `/work/projects/embomicrobial2020/data/metaT/binning/<SAMPLE>/<MAG>.mt.bam`, `/work/projects/embomicrobial2020/data/metaT/binning/<SAMPLE>/<MAG>.mg.bam`, `/work/projects/embomicrobial2020/data/metaT/binning/<SAMPLE>/<MAG>.mt.bam.bai`, and `/work/projects/embomicrobial2020/data/metaT/binning/<SAMPLE>/<MAG>.mg.bam.bai`
- the filtered GFF file `/work/projects/embomicrobial2020/data/metaT/annotations/<SAMPLE>/<MAG>.annotation_CDS_RNA_hmms.gff`

IGV is a graphical interface for genome visualization. You can run it on your computer. Here's how you can install it: `get IGV`.

Once you have installed IGV, you can open the GUI and follow the description `here` to visualize the MAG, the metagenomics and metatranscriptomics reads that map to it, the positions of the open reading frames, and the positions of the metatranscriptomics-only contigs we were able to recover in the last step.

Tasks

Explore the variation in coverage depth in the metagenome and metatranscriptome.

Focus on the metatranscriptome: can you see polycistronic operons? Compare the pattern of coverage with metatransctiptomic reads to the positions of the metatranscriptome-assembled contigs. Do you see a pattern? Can you find positions with a surprising coverage depth?

Use the search bar to find a gene with the KO annotation K00635. This is a key gene for the lipid-accumulating phenotype of *Microthrix parvicella* and the whole floating sludge community.

- genindex
- modindex
- search