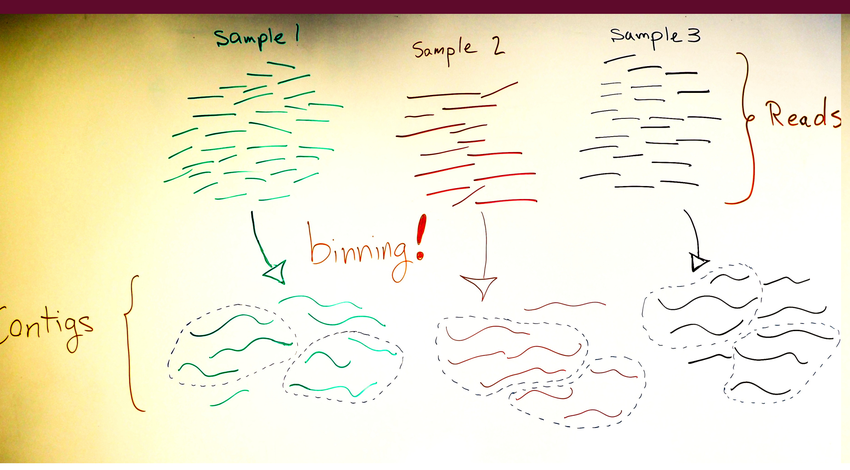
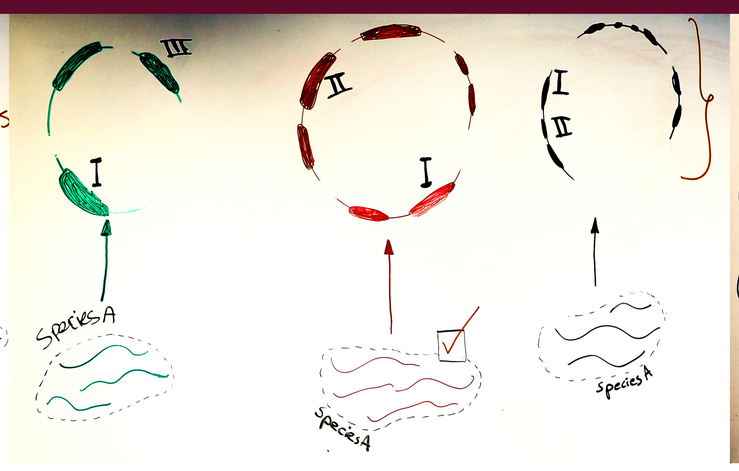
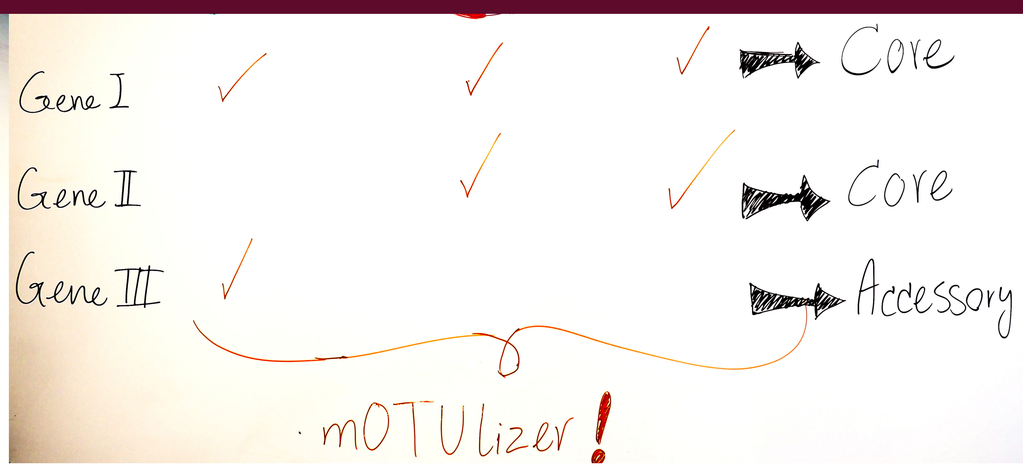
mOTUlizer: Bayesian approach to leverage imperfect metagenomic bins for pan-genome analysis

Moritz Buck[1], Maliheh Mehrshad[2], and Stefan Bertilsson[1,2]
[1]Swedish University of Agricultural Sciences
[2]Uppsala University, Sweden







Many assemblies are generated from multiple metagenomic samples or using different assembly strategies. From these, many bins are clustered. These collections of bins can contain many replicates of a genome of the same population/species/clade. Replicates can be clustered into so-called metagenomic Operation Taxonomic Units (with the mOTUlize.py program of mOTUlizer), shortened mOTU.

Classically, only the best genome, as in most complete and less contaminated, of an mOTU is kept. Losing a lot of information on gene distribution. With the mOTUpan.py program of mOTUlizer, we use the presence/absence pattern of genes, combined with estimated completenesses to predict which genes are core to an mOTU or part of the accessory.

The method is fast and flexible and can be used to compute pan-genomes traits fast and accurately for noisy as well as cleaner data. It can also be used to predict the distribution in a population of any other genome encoded trait, e.g. pathways, annotations, COGs.

The math bits: Some unresolved probabilities

Each genome (we will use genome as a shorthand for any set of nucleotide sequences belonging to the same biological entity, e.g. draft genome, complete genome, or Metagenome Assembled Genome), is described as a set of traits (in the case of this analysis a set of COGs, but mOTUlizer is agnostic to the type of traits). And each genome it self is part of a set of genomes which we will call an mOTU (metagenomic Operational Taxonomic Unit, due to the nature of the data we analyse here). mOTUlizer uses an iterative bayesian approach to classify each trait of the genome in an mOTU as a "core"-trait or "auxiliary"-trait based on a likelihood ratio. For each of the two hypotheses (core-trait or auxiliary-trait) a probability is computed assuming a certain completeness values for the each genome. Whicever of these is more likely is picked as class for that trait. Using this new classification we update the completeness estimate and recalculate the likelyhood ratios and repeat the process until convergence.

To compute the probability of a distribution of a specific trait in an mOTU under the assumption that it is in the core, we will simply multiply the completeness c_g of the genomes g that have that COG, with the inverse probability $1\text{-}c_p$ for the genomes that to not have that trait, e.g. equations 1 and 2.

1) $$\log(p_{\text{trait}|\text{core}}) = \sum_{\substack{\text{g in mOTU}\\ \text{if trait in g}}} \log(p_{\text{trait in g}|\text{core}}) + \sum_{\substack{\text{g in mOTU}\\ \text{if trait not in g}}} \log(1 - p_{\text{trait in g}|\text{core}})$$

2) $$p_{\text{trait in g}|\text{core}} = c_g$$

For the probability under the assumption that it is in the auxiliary fraction of the genome, we will have to make some assumptions on the structure of the pangenome. We have assumed that the traitsin the pangenome that are not in the core, are independent, and each trait has a frequency |trait|/|G| where |trait| is the number of genomes in the mOTU that have that trait, and |G| the total size of the traits-pool. We draw "|g|"-times, which is the number of traits estimated in that partial genome. Resulting in equations (3) and (4).

3) $$\log(p_{\text{trait}|\text{access}}) = \sum_{\substack{\text{g in mOTU}\\ \text{if trait in g}}} \log(1-\bar{p}_{\text{trait in g}|\text{access}}) + \sum_{\substack{\text{g in mOTU}\\ \text{if trait not in g}}} \log(\bar{p}_{\text{trait in g}|\text{access}})$$

4) $$\bar{p}_{\text{trait in g }|\text{access}} = (1 - \frac{|\text{trait}|}{|G|})^{|g|}$$

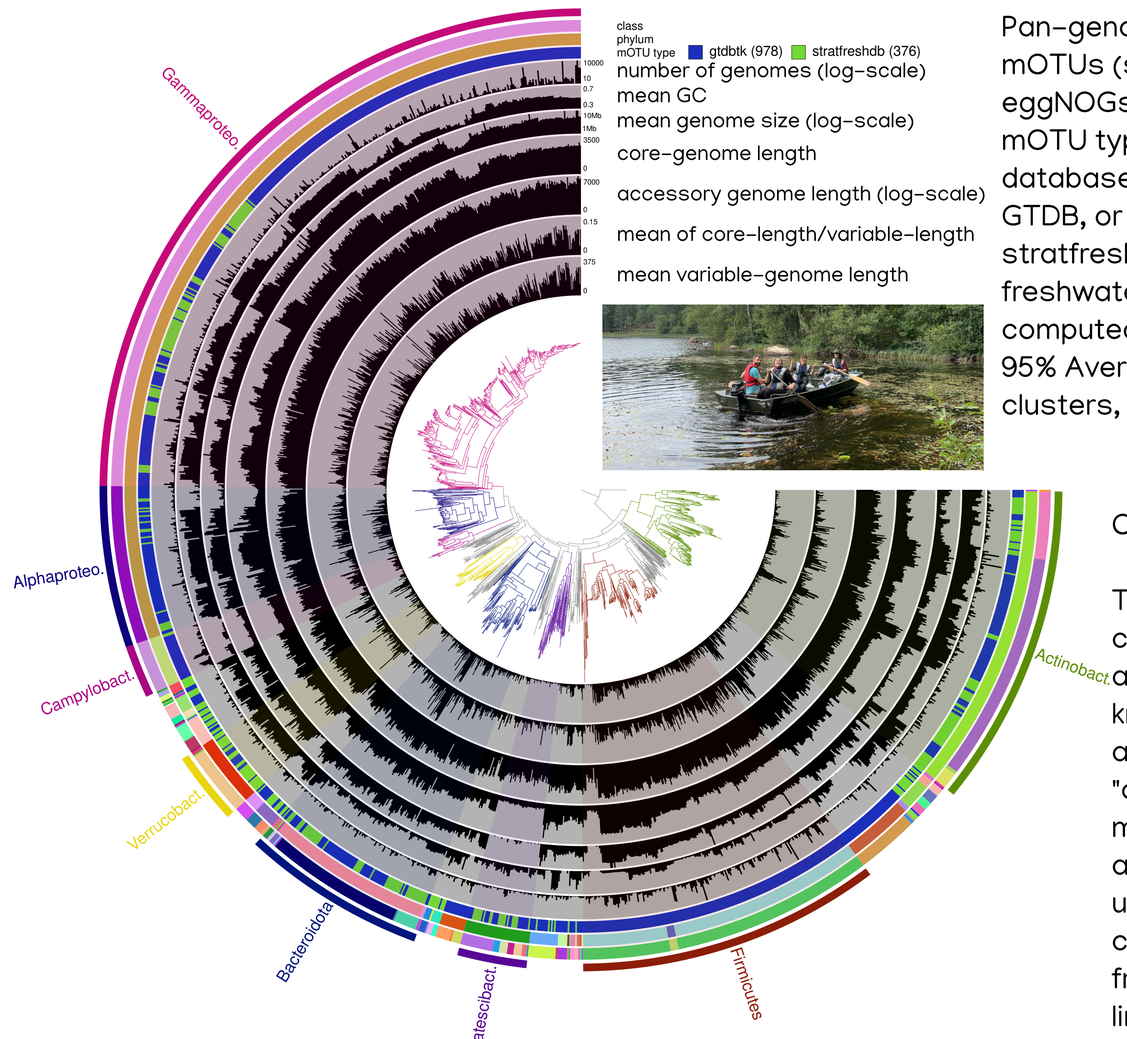
Resulting into a log-likelyhood ratio:

5) $$LLHR = \log(p_{\text{trait}|\text{core}}) - log(p_{\text{trait}|\text{access}})$$

if this is positive, the trait is considered core, if negative, auxiliary. Using this classification we recompute for each genome an updated completeness:

6) $$c_g = \frac{|\text{core}_{\text{mOTU in g}}|}{|\text{core}_{\text{mOTU}}|}$$

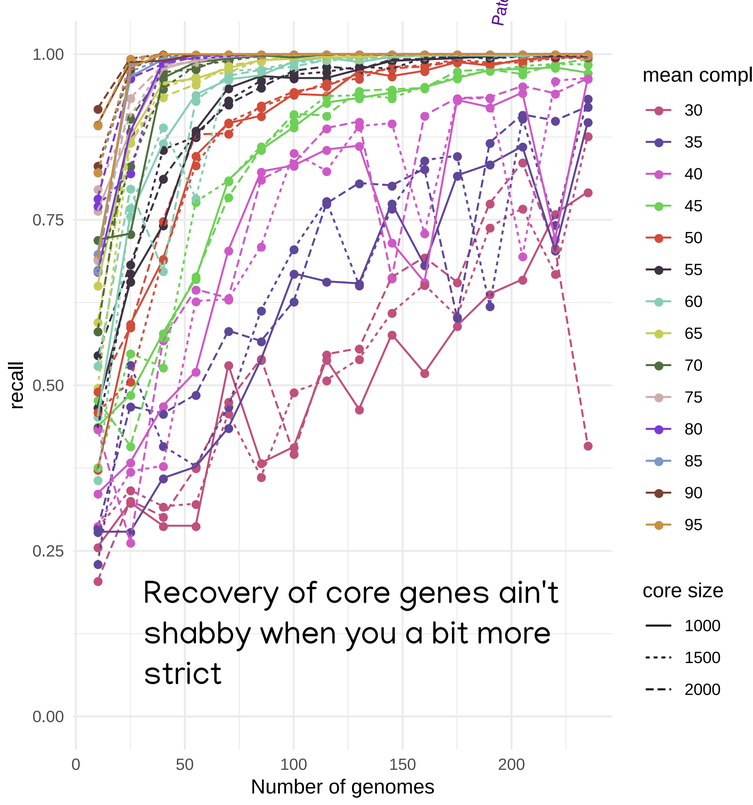
And rerun the likelihood computation. This is repeated until convergence, to obtain a final set of core-traits and and auxiliary-traits. Unique convergence depending on initial completeness scores has been tested (Supp ?) and stable convergence is obtained if the number of genomes is larger then 5.

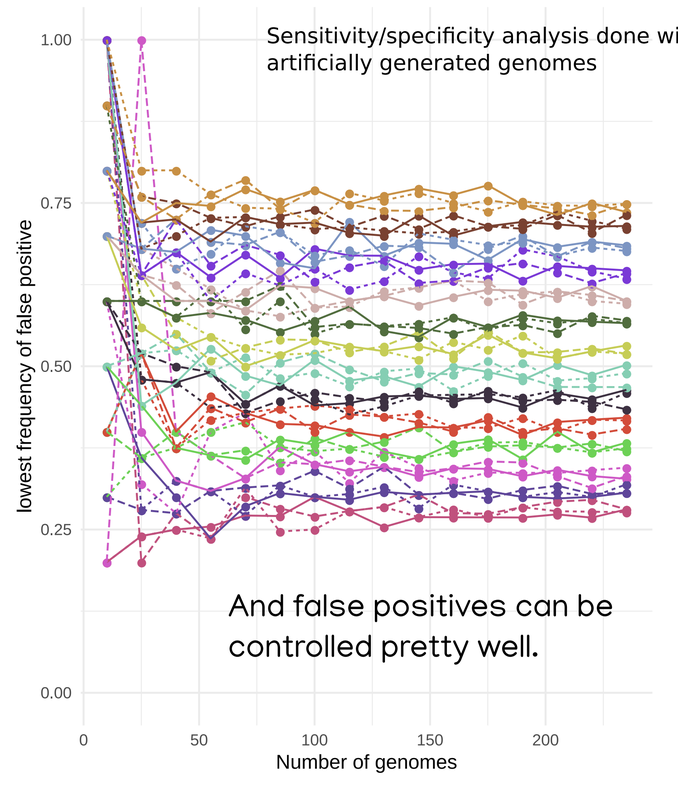


Pan-genome traits for about 1300 mOTUs (species). Computed using eggNOGs as genome encoded traits. mOTU type is depending on the database, either the species from the GTDB, or mOTUs from the stratfreshDB, a database of freshwater-metagenomes. These were computed using mOTUlize (basically 95% Average Nucleotide Identity clusters, as computed with FastANI.

Other alternatives:

There are tools out there for computing core and accessory- genome, most well known is roary. However roary and the others are limited to "closed" genomes making the mass of metagenomic assembled genomes (MAGs) unuseable. Alternatively, you can simply threshold by frequency, with its obivous limitations





@metamoritz
@moritzbuck
@metamoritz

Used Tools:

FastANI : average nucleotide computation
GTDBtk : tree and taxonomy
diamond : similarities for silix
silix : fast COG computation
checkm : initial completenss computation
eggnog-mapper : annotation/traits
etetoolkit : great python-pkg for phylo-stuff
anvi'o : amazing viz-tool

Databases:

GTDB : genome database
stratfreshDB : TBP, NCBI under PRJEB38681

Moritz Buck and secret author