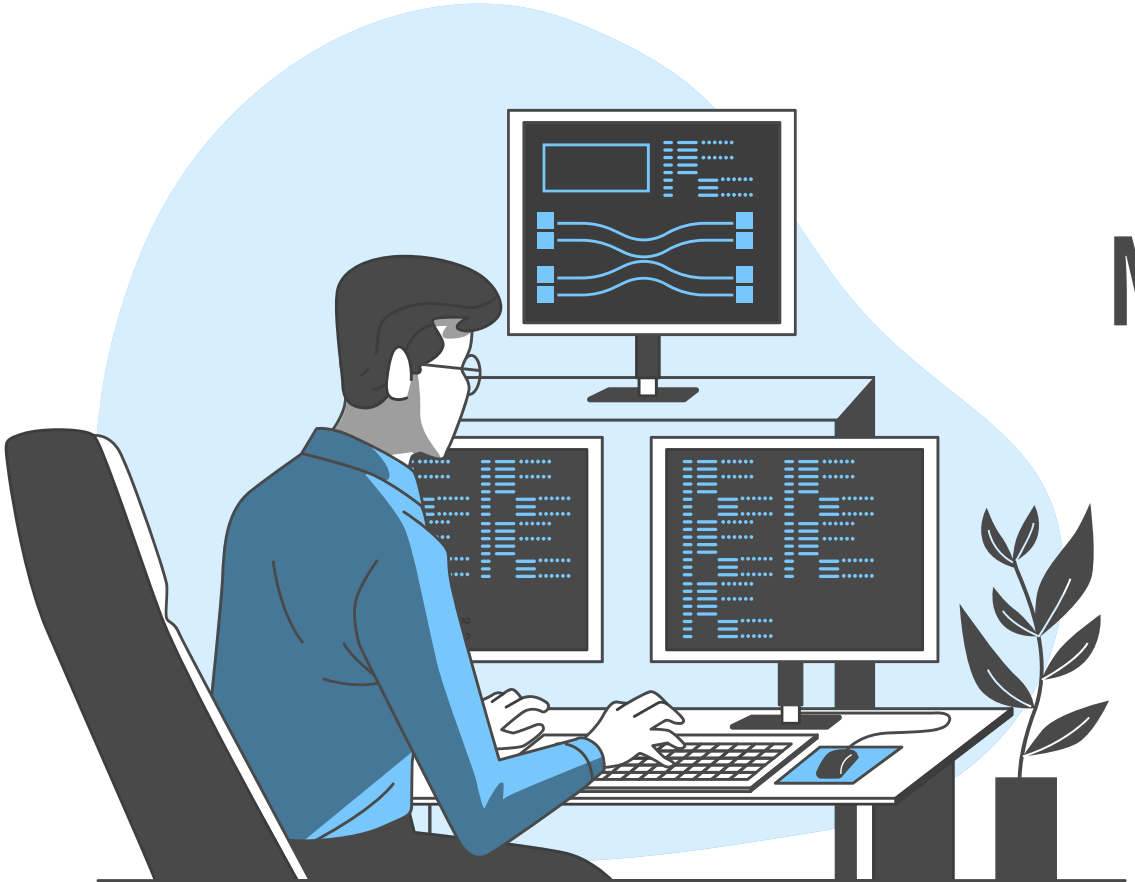


Metagenomics workshop

Brain-Gut Axis Conference

Eva Pavlinek

exaltum
scientific knowledge management

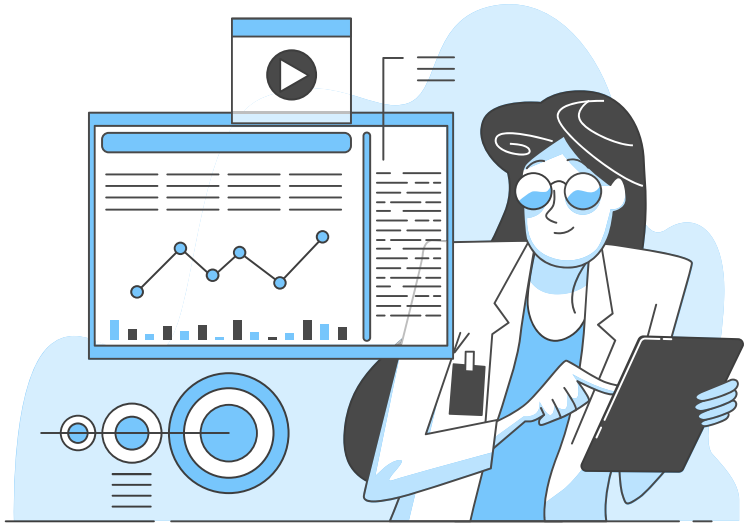
Workshop structure

01 Introduction to Galaxy

02 Introduction to metagenomics

03 Practical work

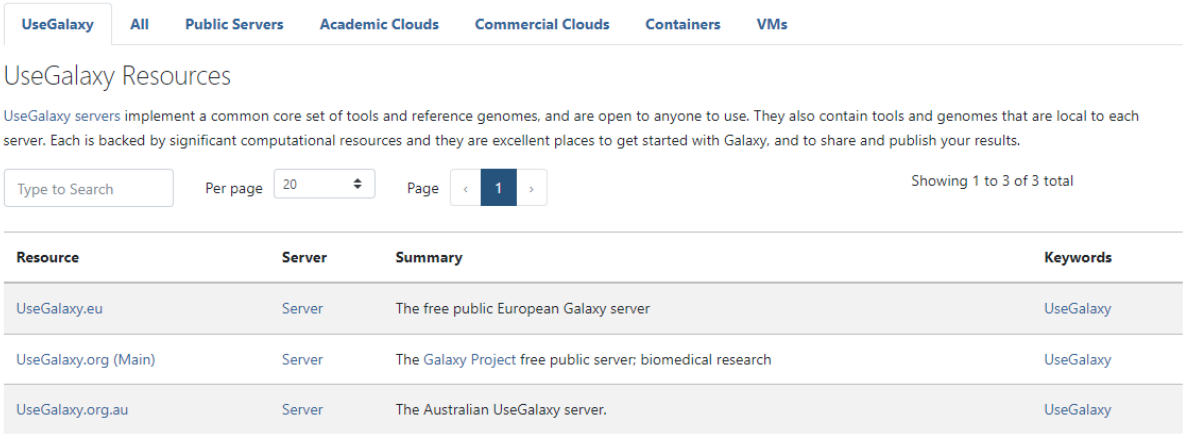
04 Conclusions & additional resources



Introduction to Galaxy

- free web-based data analysis platform
- homepage: https://galaxyproject.org/

- **the server we are going to use:** https://usegalaxy.org/



Task 1: Create an account

Task 2: Upload the data

- go to **Metagenomics workshop** folder on your **Desktop**
- **Upload Data** from the **data** folder (everything except the silva.txt file)
  - change the **Type** for the Screen.seqs file to **mothur.count_table**
  - change the **Type** for the OTU_List file to **mothur.list**
- **Upload Data** → Paste/Fetch data → paste the link from the **silva link.txt** file

waiting
running
success
failed

- All data and materials for this workshop can be found at:
- https://github.com/epavlinek/BGA_conference_metagenomics_workshop

Introduction to metagenomics

- Metagenome – the complete genetic material extracted directly from an environmental sample
- Metagenomics – the study of such genetic material

- most of the microorganisms cannot be cultured by plating (traditional methods)

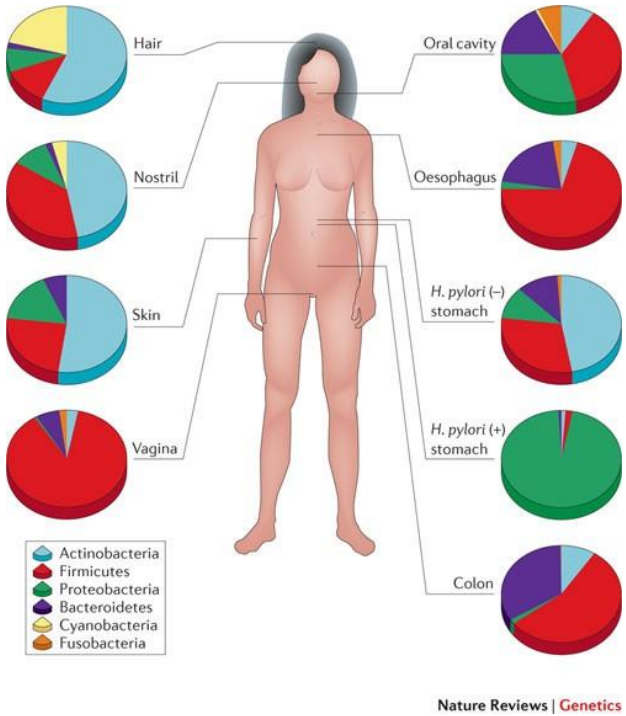



Human microbiome

- unique as a person's fingerprint
- bacteria, archaea, fungi, viruses

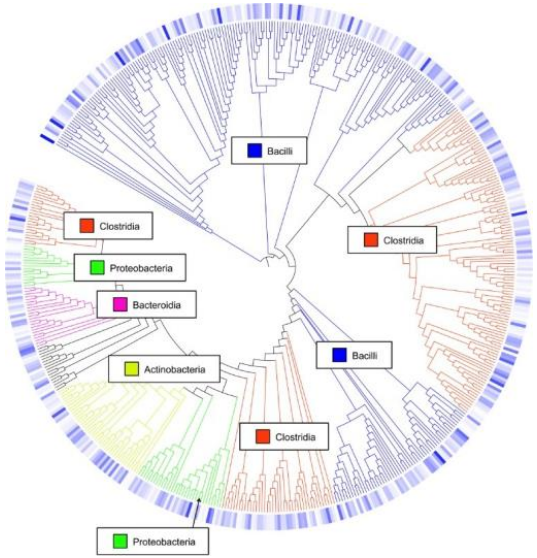
~10 times more cells than you
~100 times more genes than you
~1000s different species

- gut microbiome holds the vast majority of bacteria (good and bad)



What are we searching for?

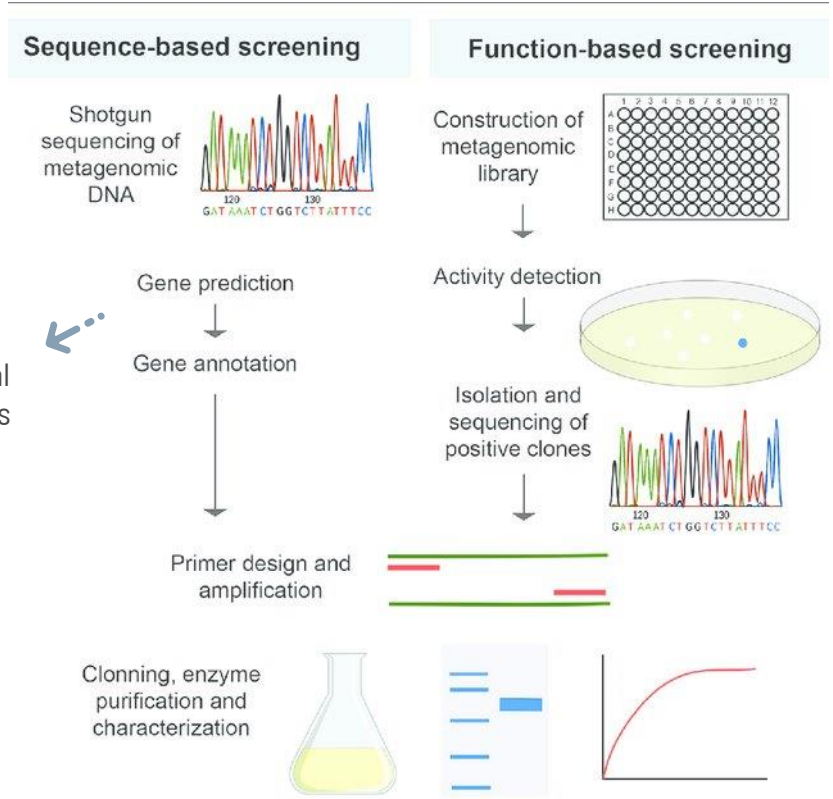
- **Who is there?**
- phylogenetic diversity



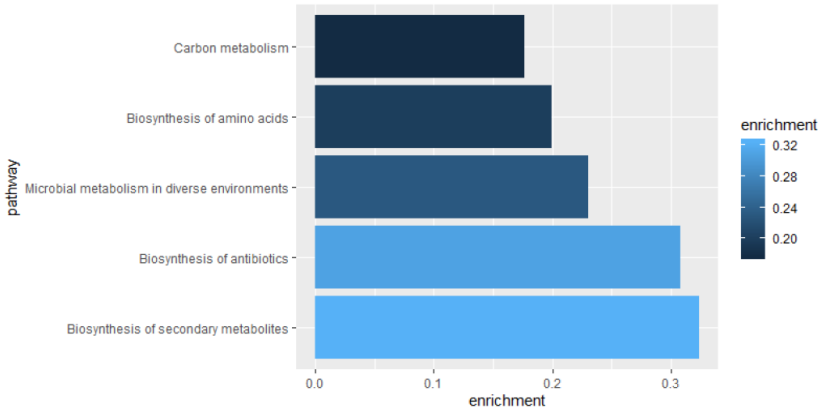
- **What do they do?**
- functional diversity
- screen for enzymatic activities
- novel genes?
- novel genomes?

Functional metagenomics

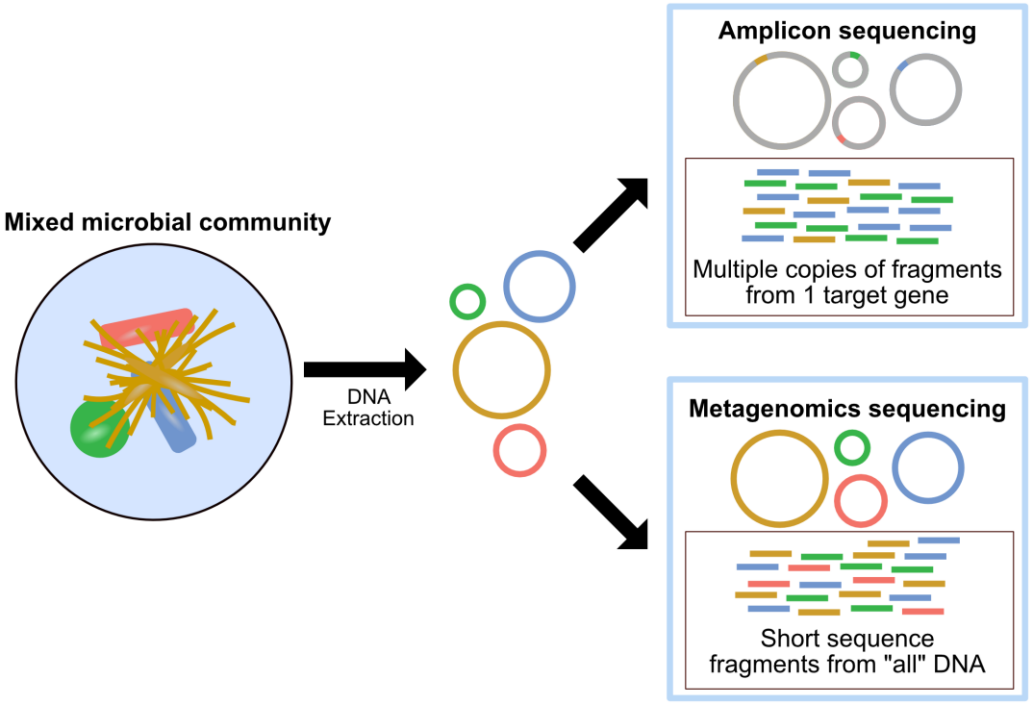
search functional databases



1. DNA fragmentation by restriction enzymes or mechanical digestion
2. ligation to a vector -> recombinant vectors
3. bacterial transformation
4. metagenomic library screening
5. gene sequencing or protein isolation

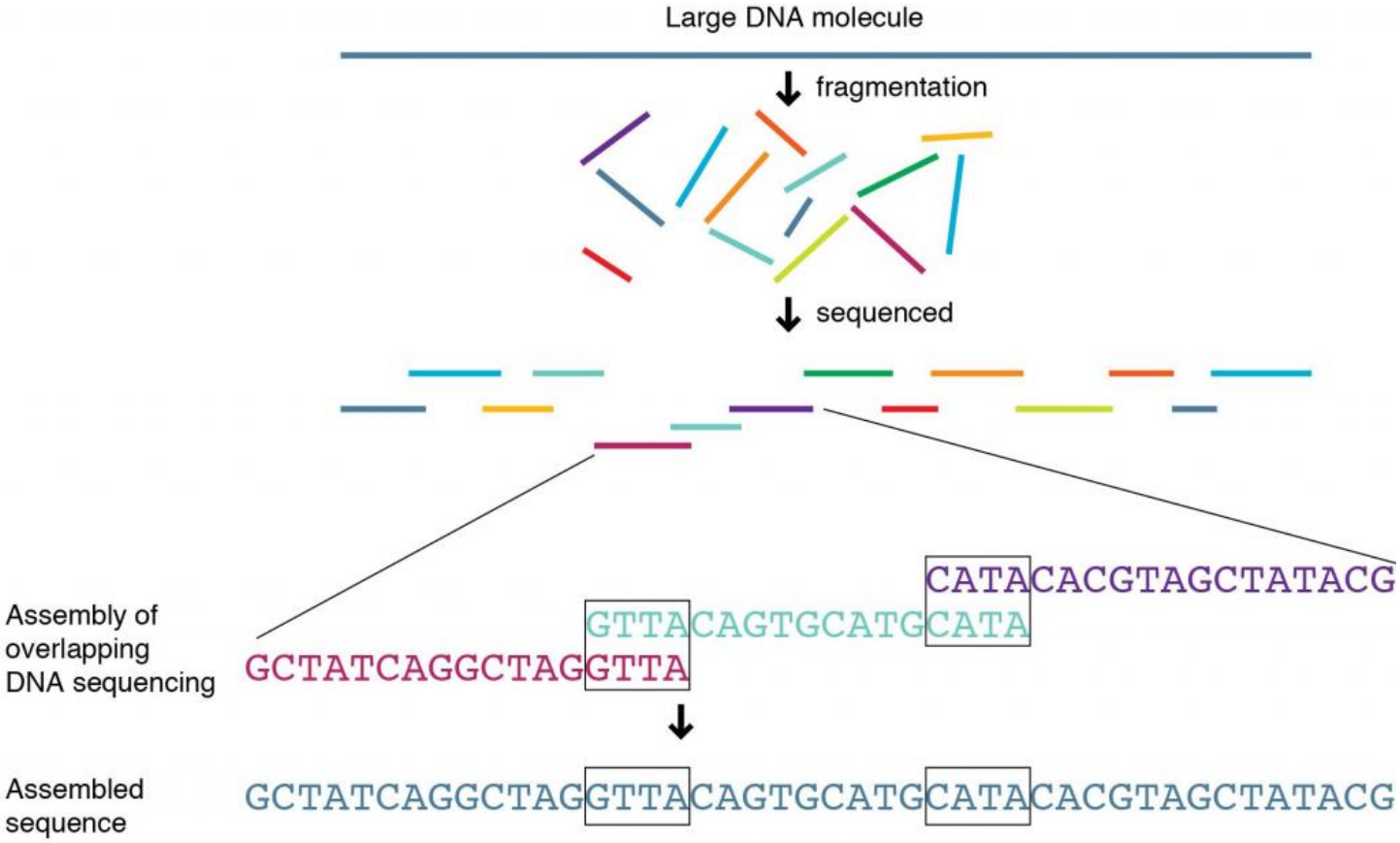


Metagenomic sequencing – amplicon/targeted vs. shotgun



- sequence only targeted regions
- cheaper and less complex to anlyse
- no functional information

- sequence all DNA
- higher cost and complexity
- more information

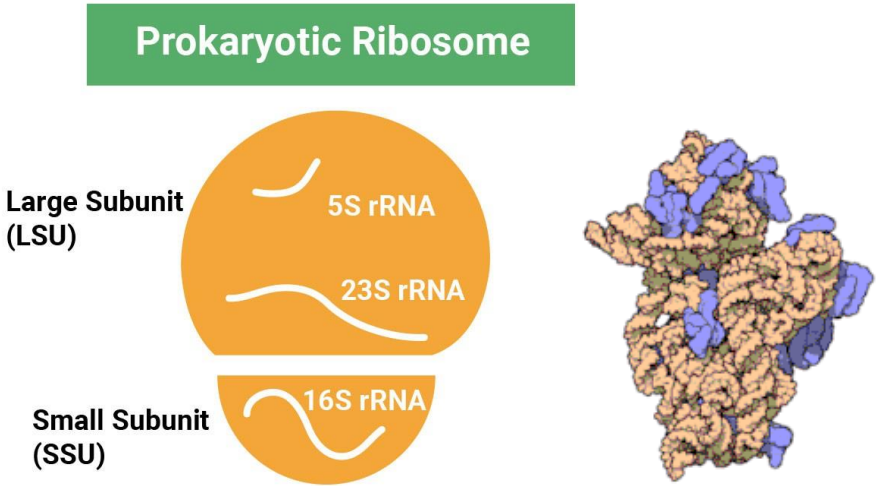

Large DNA molecule
fragmentation
sequenced
Assembly of overlapping DNA sequencing
GCTATCAGGCTAGGTTA
GTTACAGTGCATGCATA
CATACACGTAGCTATACG
Assembled sequence
GCTATCAGGCTAGGTTACAGTGCATGCATACACGTAGCTATACG

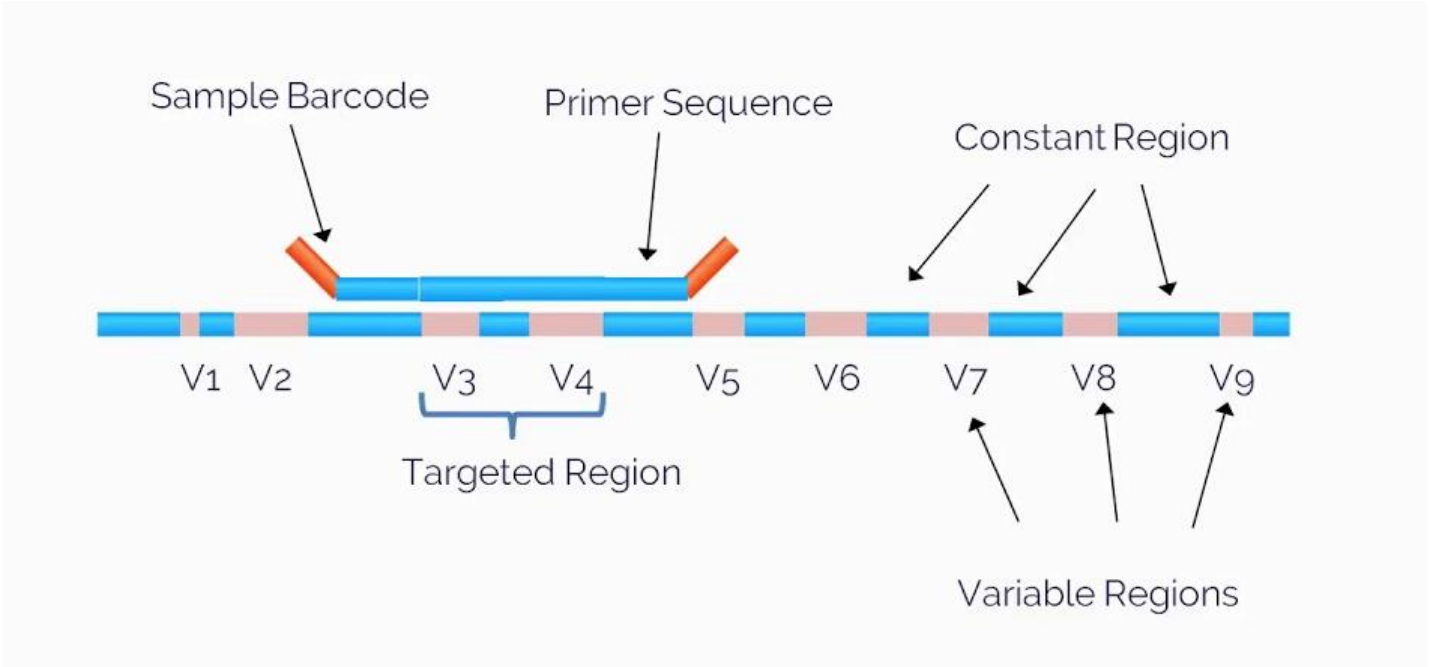
Amplicon / targeted sequencing

- **16S rRNA gene**

- Present in all bacteria
- Highly conserved + highly variable regions
- Huge reference databases

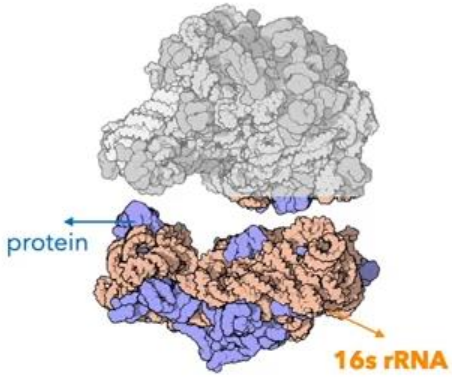
- 18S rRNA for fungi



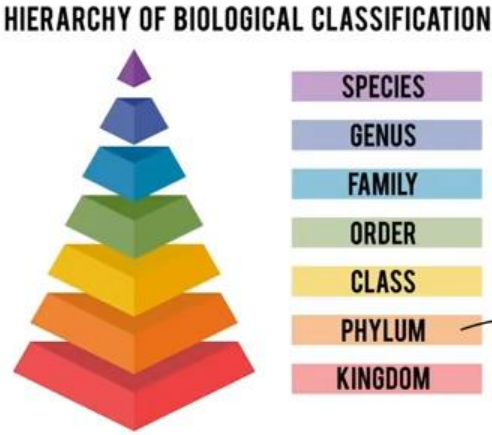


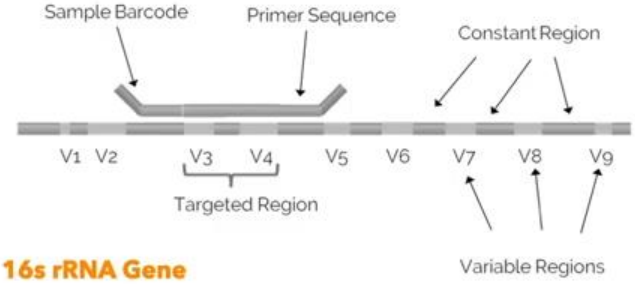
The highly conserved regions make it easy to target the gene across different organisms, while the highly variable regions allow us to distinguish between different species.

16s Taxonomic Classification

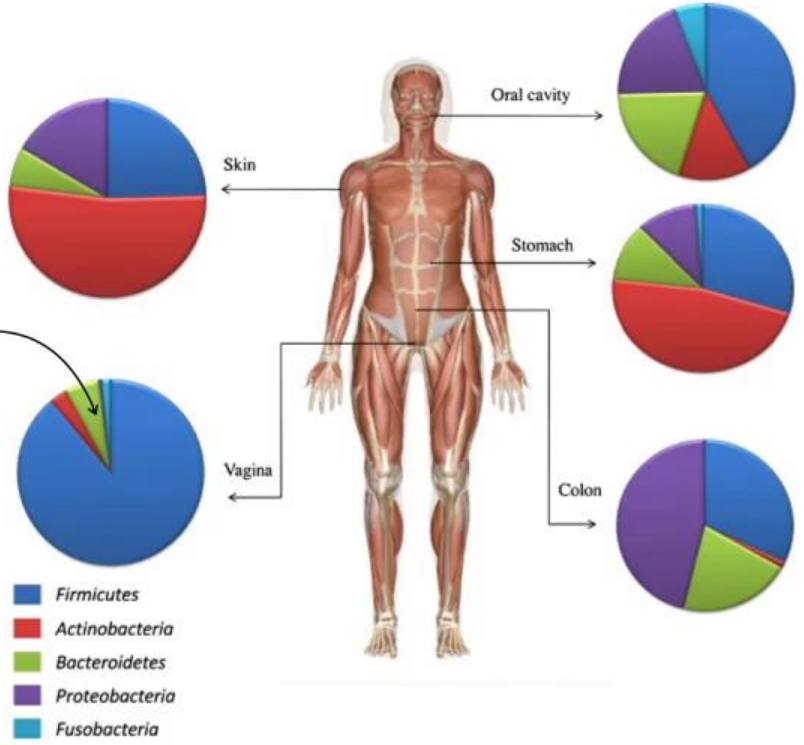


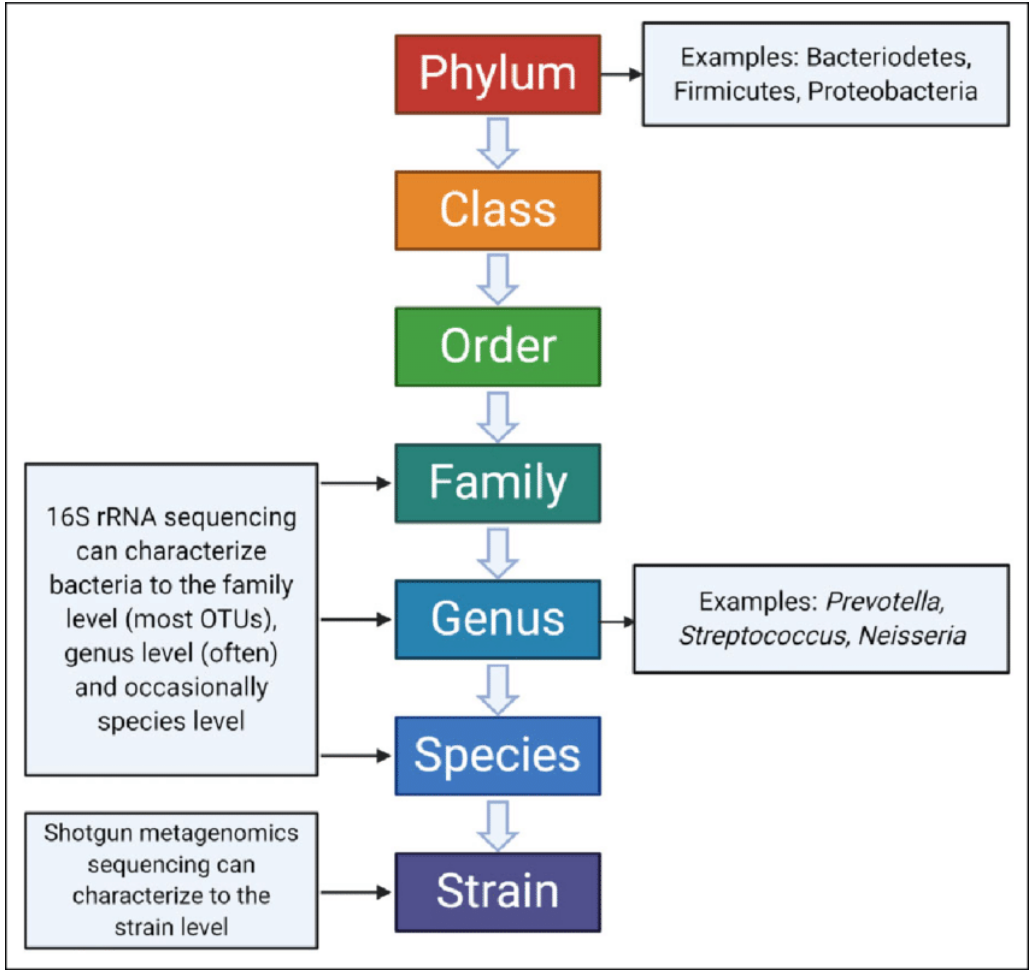
30S ribosomal subunit



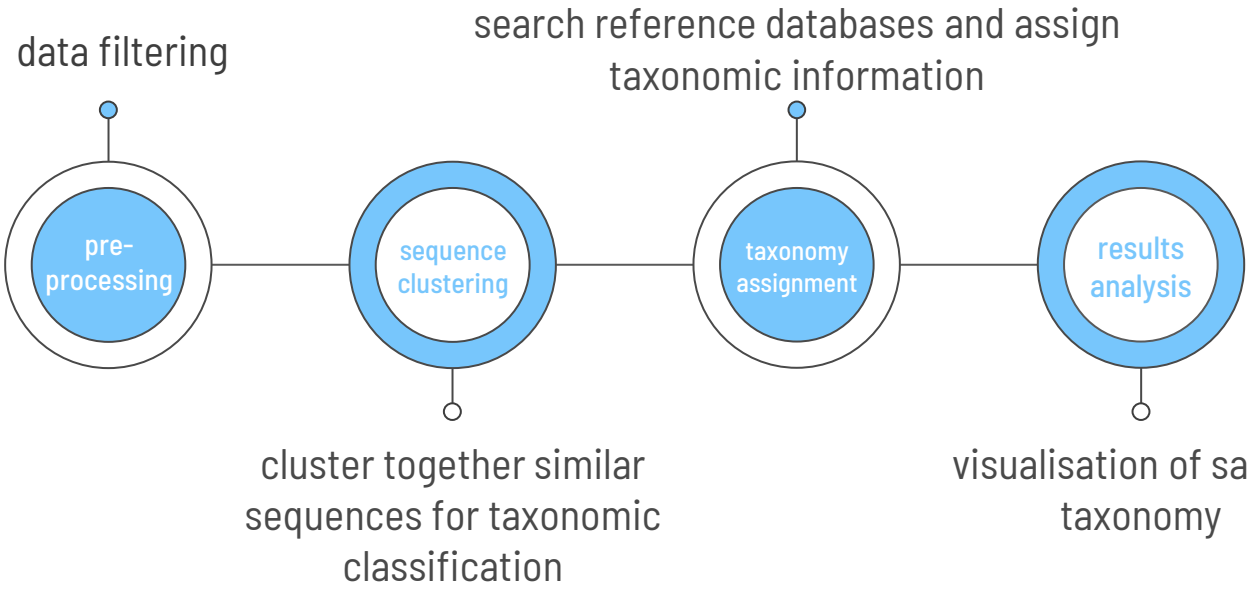


16s rRNA Gene




Phylum
Examples: Bacteriodetes, Firmicutes, Proteobacteria
Class
Order
Family
16S rRNA sequencing can characterize bacteria to the family level (most OTUs), genus level (often) and occasionally species level
Genus
Examples: Prevotella, Streptococcus, Neisseria
Species
Shotgun metagenomics sequencing can characterize to the strain level
Strain

Bioinformatics – a simplified workflow



BIOINFORMATICS ANALYSIS

What are our samples?

- 2 human (female) gut microbiome samples (SRR11389268, SRR11389279)

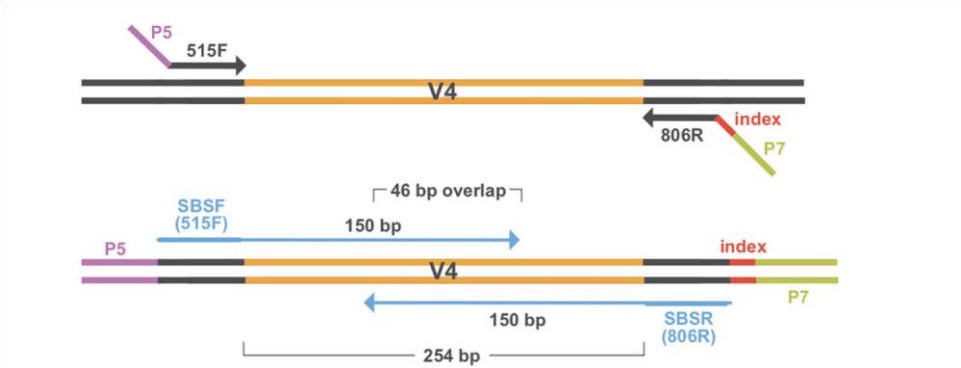
What is our goal?

- analyse the data and perform the taxonomic classification
  - Can we conclude something about the microbial composition?
  - Which species are present and in what percentage?

1. SAMPLE PREPARATION

What are our samples?

- 2 samples → paired-end reads in fastq files
  - SRR11389268_1.fastq (forward reads)
  - SRR11389268_2.fastq (reverse reads)
  - SRR11389279_1.fastq (forward reads)
  - SRR11389279_2.fastq (reverse reads)



Identifier — `@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1`

Sequence — `TTAATTGGTAAATAAATCTCCTAATAGCTTAGATNTTACCTTNNNNNNNNNNTAGTTTCTTGAGA`

+ sign & identifier — `+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1`

Quality scores — `efcffffcfeefffcffffffddf`feed]`]_Ba_^__[YBBBBBBBBBBRTT\]][]dddd`

Base T

phred Quality] = 29

encoded probability that the corresponding base call is incorrect

1. SAMPLE PREPARATION

Exercise 1: Create a paired collection (pair the forward and reverse reads)

1. Click on the **checkmark icon** at top of your history
2. Select all 4 sample files
3. Click on **4 of 8 selected**
4. Select **Build List of Dataset Pairs** from the dropdown menu:
5. Name your collection at the bottom right of the screen

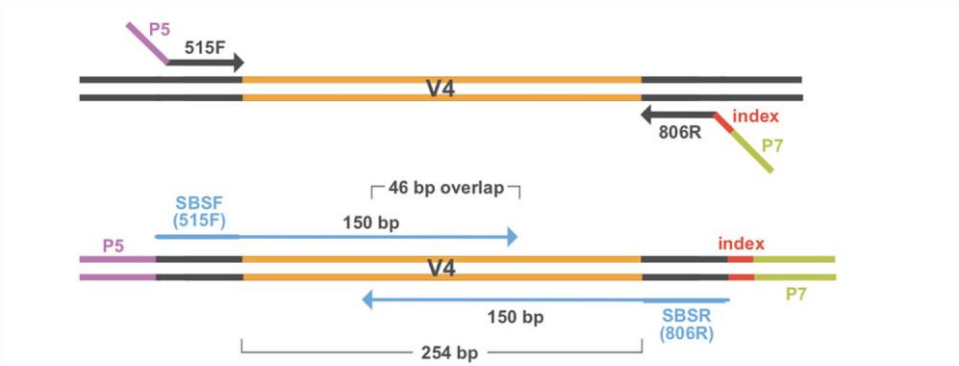
   In the next dialog window you can create the list of pairs. You should see a list of pairs suggested by Galaxy.
6. Click the **Create Collection** button

2. DATA PRE-PROCESSING

Exercise 2: Create contigs from paired-end reads

The sequencing was done from either end of each fragment. Because the reads are about 175 bp in length, there is an overlap between the forward and reverse reads in each pair. We will combine these pairs of reads into contigs.

1. **Make.contigs** tool with the following parameters:
- "*Way to provide files*": **Multiple pairs - Combo mode**
- "*Fastq pairs*": the collection you just created
- leave all other parameters to the default settings
- **Execute**



The output:

This step combined the forward and reverse reads for each sample, and also combined the resulting contigs from all samples into a single file. So we have gone from a paired collection of 2x2 FASTQ files, to a single FASTA file. In order to retain information about which reads originated from which samples, the tool also outputs a group file.

You can view the **trim.contigs.fasta** file and the **group** file

2. DATA PRE-PROCESSING

Exercise 3: Optimisation of files for computation

We are sequencing many of the same organisms, we anticipate that many of our sequences are duplicates of each other. Because it's computationally wasteful to align the same thing a bazillion times, we'll extract the unique sequence.

1. **Unique.seqs** tool with the following parameters:
- "*fasta*": **trim.contigs.fasta file from the Make.contigs**
- "*output format*": **Name file**
- leave all other parameters to the default settings
- **Execute**

The output:

This tool outputs two files: a fasta file containing only the unique sequences, and a names file. The names file consists of two columns, the first contains the sequence names for each of the unique sequences, and the second column contains all other sequence names that are identical to the representative sequence in the first column.

How many sequences were unique? How many duplicates were removed?

209,581 unique sequences and 47,075 were removed. This can be determined from the number of lines in the output, compared to the number of lines in the fasta file in the trim.contigs.fasta file.

2. DATA PRE-PROCESSING

Exercise 4: Count sequences

We want to keep track of the number of sequences represented by each unique sequence across multiple samples.

1. **Count.seqs** tool with the following parameters:
- *"name"*: **name file from the Unique.seqs**
- *"Use a Group file to include counts for groups"*: **yes**
- *„group - Group file for the tree"*: **the group file from Make.contigs**
- leave all other parameters to the default settings
- **Execute**

The output:

A table with a list of representative (unique) sequences, thier total number and their count in each sample.

Should we skip a few steps?

Do we have ~1:20 left?

2. DATA PRE-PROCESSING

Exercise 5: Quality control – part 1

Let's get the information about the quality of our data.

1. **Summary.seqs** tool with the following parameters:
- *"fasta - Dataset"*: **fasta from Unique.seqs**
- *"count – a count_table"*: **count_table from Count.seqs**
- *„output logfile"*: **yes**
- leave all other parameters to the default settings
- **Execute**

The output:

Let's view the logfile.

We have a total of 209581 unique sequences, representing 256656 total sequences that vary in length between 174 and 350 bases. Also, note that at least some of our sequences had some ambiguous base calls. Furthermore, at least one read had a homopolymer stretch of 156 bases, this is likely an error so we would like to filter such reads out as well.

	Start	End	NBases	Ambigs	Polymer	NumSeqs
Minimum:	1	174	174	0	3	1
2.5%-tile:	1	252	252	0	3	6417
25%-tile:	1	253	253	0	4	64165
Median:	1	253	253	1	4	128329
75%-tile:	1	253	253	2	5	192493
97.5%-tile:	1	255	255	10	6	250240
Maximum:	1	350	350	48	156	256656
Mean: 1	254.099	254.099	1.75228	4.4505		
# of unique seqs:		209581				
total # of seqs:		256656				

2. DATA PRE-PROCESSING

Exercise 6: Quality control – part 2

We want to filter our dataset on length, base quality, and maximum homopolymer length.

2. Screen.seqs tool with the following parameters:

- "*fasta – Fasta to screen*": **fasta from Unique.seqs**
- "minlength" parameter to **252**
- "maxlength" parameter to **255**
- "maxambig" parameter to **0**
- "maxhomop" parameter to **8**
- "count – a count_table": **count_table from Count.seqs**
- leave all other parameters to the default settings
- **Execute**

The output:

We removed any sequences with ambiguous bases (maxambig parameter), homopolymer stretches of 8 or more bases (maxhomop parameter) and any reads longer than 255 bp or shorter than 252 bp.

We removed ~130,000 reads (check the bad.accnos output).

3. SEQUENCE ALIGNMENT

Exercise 7: Sequence alignment to the rRNA reference database

We want to align our sequences to a reference (the V4 variable region of the 16S rRNA).

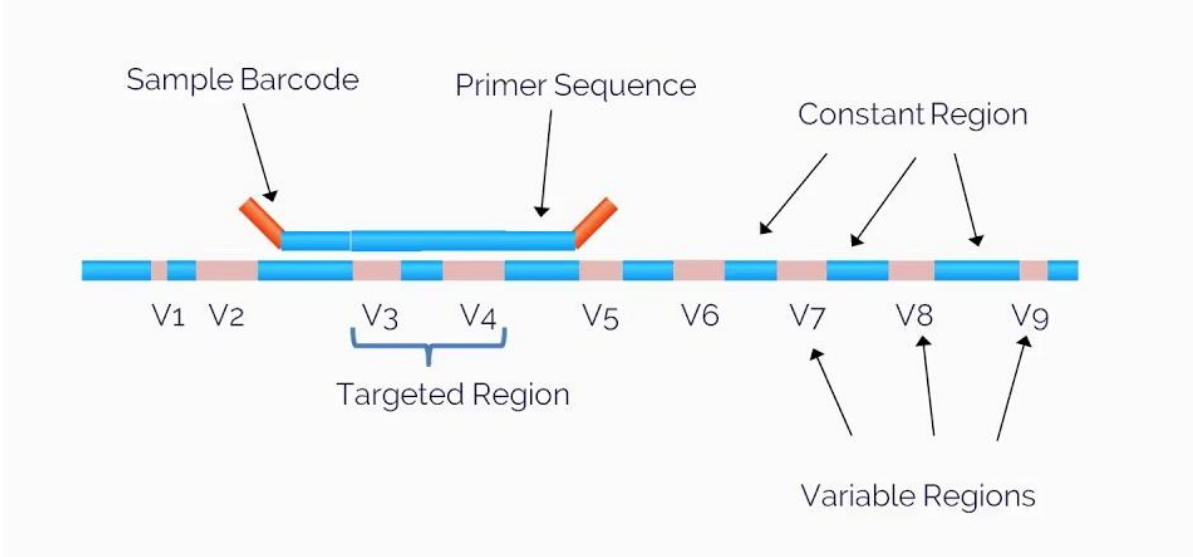
1. Align.seqs tool with the following parameters:

- "*fasta – Candidate Sequences*": **the good.fasta output from Screen.seqs**
- "*Select Reference Template from*": **Your history**
- "*reference*": **the silva.v4.fasta** reference file
- "*flip*": **Yes**
- leave all other parameters to the default settings
- **Execute**

3. SEQUENCE ALIGNMENT

What is the point of this step?

Aligning out sequences to the V4 variable region of 16S rRNA allows us to filter out the sequences which do not cover this part of the gene, as well as improve taxonomy assignment in the next steps.



3. SEQUENCE ALIGNMENT

Exercise 8: Sequence alignment to the rRNA reference database – part 2

We want to analyse the alignment results.

2. Summary.seqs tool with the following parameters:

- *"fasta – Dataset"*: **the align output from Align.seqs**
- *"count – a count_table"*: **count output from Screen.seqs**
- *„output logfile"*: **yes**
- leave all other parameters to the default settings
- **Execute**

The output:

Let's look at the quality of the alignment, we can view the log output from the summary step.

76581 sequences have been aligned, mostly between the positions 1968 and 11550 (~ where the V4 target region of the 16S gene is).

	Start	End	NBases	Ambigs	Polymer	NumSeqs
Minimum:	1965	10694	252	0	3	1
2.5%-tile:	1968	11550	252	0	4	3044
25%-tile:	1968	11550	253	0	4	30438
Median:	1968	11550	253	0	4	60875
75%-tile:	1968	11550	253	0	5	91312
97.5%-tile:	1968	11550	253	0	6	118705
Maximum:	1976	11550	255	0	8	121748
Mean:	1968	11550	252.912	0	4.47801	
# of unique seqs:		76581				
total # of seqs:		121748				

3. SEQUENCE ALIGNMENT

Exercise 9: Sequence alignment to the rRNA reference database – part 3

We want our sequences to overlap the same region so we will keep only those which start at 1968 and end at 11550 (we'll filter the sequences to remove the overhangs at both ends). In addition, there are many columns in the alignment that only contain gap characters ("."), look at the align file from Align.seqs) and these can be removed without losing any information.

3. Screen.seqs tool with the following parameters:

- *"fasta – Fasta to scren"*: **the align output from Align.seqs**
- *"start"*: **1968**
- *"end"*: **11550**
- *"count – a count_table"*: **the count file created by the previous run of Screen.seqs**
- leave all other parameters to the default settings
- **Execute**

4. Filter.seqs tool with the following parameters:

- *"fasta – Alignment Fasta"*: **good.fasta output from Screen.seqs**
- *"trump – Trump character"*: **. (the dot sign)**

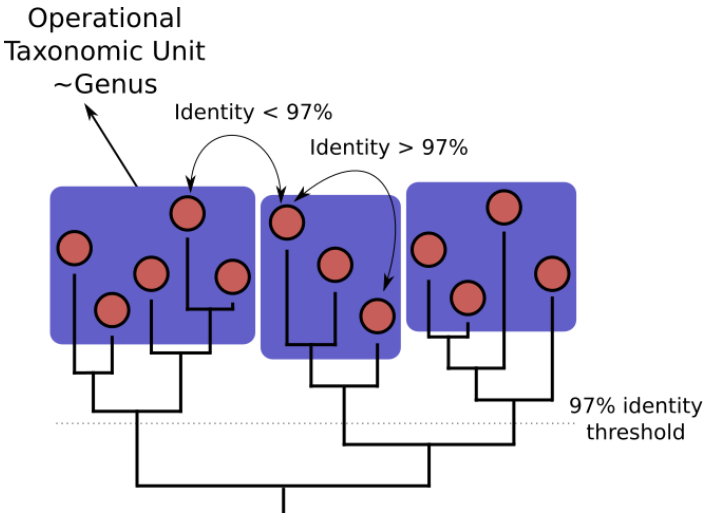
The output:

We removed 95 sequences (check the bad.accnos output from the Screen.seqs) and we only have the sequences which cover the same region.

Continue the analysis from the next step...

4. TAXONOMIC CLASSIFICATION

In order to answer which microorganisms are present in our samples (and in what proportion), we want to assign our sequences to a taxon. To do that, we group (or cluster) sequences based on their similarity to defined **Operational Taxonomic Units (OTUs)**: groups of similar sequences that can be treated as a single "genus" or "species" (depending on the clustering threshold).



Typically, OTU clusters are defined by a 97% identity threshold of the 16S gene sequence variants at genus level. 98% or 99% identity is suggested for species separation.

4. TAXONOMIC CLASSIFICATION

Exercise 10: Preclustering of the sequences

We will group together the sequences that differ by no more than 2 nucleotides from one another and remove all sequences that differ too much from all other sequences.

1. Pre.cluster tool with the following parameters:

- *"fasta – Sequence Fasta"*: **the fasta output from the last Filter.seqs run (UPLOADED FILE)**
- *"name file or count table"*: **the count file from the last Screen.seqs step (UPLOADED FILE) → might need to convert this file to mothur.count_table format**
- *"diffs - Number of mismatched bases to allow between sequences in a group (default 1)"*: **2**
- leave all other parameters to the default settings
- **Execute**

The output:

46162 unique sequences are left after the clustering of highly similar sequences (check the precluster.fasta output).

4. TAXONOMIC CLASSIFICATION

Exercise 11: Sequence classification

We want to assign the taxonomic information to each sequence. For that, we will use a reference set (a fasta file which holds information which sequence belongs to which taxonomic classificatio → look at the taxonomy_reference_db fasta file).

1. Classify.seqs tool with the following parameters:

- *"fasta – Candidate Sequences"*: **fasta output from Pre.cluster**
- *"Select Reference Template from"*: **History**
- *"reference – Reference to align with"*: **taxonomy_reference_db.pds**
- *"Select Taxonomy from"*: **History**
- *"taxonomy – Taxonomy referene"*: **taxonomy_reference_db.pds.tax from your history**
- *"count file"*: **count_table from Pre.cluster**
- leave all other parameters to the default settings
- **Execute**

The output:

View the taxonomy output → we see that every read has a classification.

name	taxonomy
1061_SRR11389268.fastq	Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Bacteroidaceae(100);Bacteroides(100);
389_SRR11389268.fastq	Bacteria(100);Firmicutes(100);Clostridia(100);Clostridiales(100);Ruminococcaceae(100);Faecalibacterium(95);
479_SRR11389268.fastq	Bacteria(100);Firmicutes(100);Clostridia(100);Clostridiales(100);Ruminococcaceae(100);Faecalibacterium(100);
1421_SRR11389268.fastq	Bacteria(100);Fusobacteria(100);Fusobacteriia(100);Fusobacteriales(100);Fusobacteriaceae(100);Fusobacterium(100);
4118_SRR11389268.fastq	Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Bacteroidaceae(100);Bacteroides(100);
35701_SRR11389268.fastq	Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Rikenellaceae(100);Alistipes(100);
779_SRR11389268.fastq	Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Bacteroidaceae(100);Bacteroides(100);
6562_SRR11389268.fastq	Bacteria(100);Actinobacteria(100);Actinobacteria(100);Bifidobacteriales(100);Bifidobacteriaceae(100);Bifidobacterium(100);
3468_SRR11389268.fastq	Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Bacteroidaceae(100);Bacteroides(100);
29518_SRR11389268.fastq	Bacteria(100);Firmicutes(100);Clostridia(100);Clostridiales(100);Ruminococcaceae(100);Ruminococcaceae_unclassified(100);
36123_SRR11389268.fastq	Bacteria(100);Proteobacteria(100);Betaproteobacteria(100);Burkholderiales(100);Sutterellaceae(100);Sutterella(100);
38943_SRR11389268.fastq	Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Bacteroidaceae(100);Bacteroides(100);
2814_SRR11389268.fastq	Bacteria(100);Bacteroidetes(100);Bacteroidia(100);Bacteroidales(100);Bacteroidaceae(100);Bacteroides(100);
40239_SRR11389268.fastq	Bacteria(100);Actinobacteria(100);Actinobacteria(100);Bifidobacteriales(100);Bifidobacteriaceae(100);Bifidobacterium(100);
2078_SRR11389268.fastq	Bacteria(100);Firmicutes(100);Clostridia(100);Clostridiales(100);Lachnospiraceae(100);Clostridium_XlVa(99);

4. TAXONOMIC CLASSIFICATION

Now we want to determine the abundances of the different found taxa. This consists of three steps:

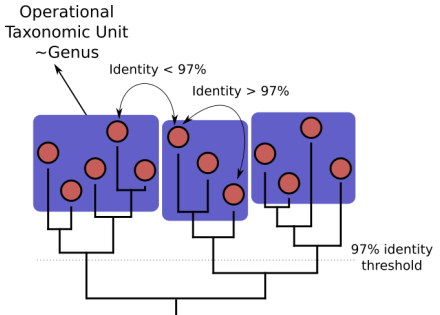
1. All individual sequences are classified, and get assigned a confidence score (0-100%)
2. Sequences are grouped at 97% identity threshold (not using taxonomy info)
3. For each cluster, a consensus classification is determined based on the classification of the individual sequences and taking their confidence scores into account

4. TAXONOMIC CLASSIFICATION

Exercise 12: Sequence classification – part 2: Assign sequences to OTUs

2. Cluster.split tool with the following parameters:

- *"Split by"*: **Classification using fasta**
- *"Fasta"*: **the fasta output from Pre.cluster**
- *"Taxonomy"*: **the taxonomy output from Classify.seqs**
- *"name file or count table - Sequences Name reference"*: **the count table output from Pre.cluster**
- "*Clustering method*": **Average Neighbour**
- *"cutoff - Distance Cutoff threshold - ignored if not > 0"*: **0.15**
- leave all other parameters to the default settings
- **DO NOT EXECUTE! THIS STEP WILL TAKE TOO LONG → YOU ALREDY HAVE THIS OUTPUT UPLOADED AS OTU LIST!!!**



4. TAXONOMIC CLASSIFICATION

Exercise 13: Sequence classification – part 3: Estimate OTU abundance

3. Make.shared tool with the following parameters:

- *"Select input type":* **OTU list**
- *"list – OTU List":* **OTU list output from Cluster.split (uploaded OTU List) → might need to change the datatype to mothur.list**
- *"supply group or count table if you supplied OTU list":* **the count_table from Pre.cluster**
- *"label":* **0.03**
- leave all other parameters to the default settings
- **Execute**

4. TAXONOMIC CLASSIFICATION

Exercise 14: Sequence classification – part 4: Classify the OTUs

4. Classify.otu tool with the following parameters:

- *"list":* **OUT list output from Cluster.split (uploaded OTU List)**
- *"count - used to represent the number of duplicate sequences for a given representative sequence"*: **the count_table from Pre.cluster**
- *"Select Taxonomy from":* **History**
- *"taxonomy - Taxonomy Reference":* **the taxonomy output from Classify.seqs**
- *"label":* **0.03**
- *„persample - allows you to find a consensus taxonomy for each group"*: **Yes**
- leave all other parameters to the default settings
- **Execute**

The output:

View the Classify.out tax.summary output → we see how many OTUs belong to which taxonomy class and how many are found in each sample.

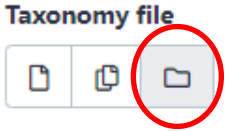
View the Classify.out taxonomy output → we see how many and which OTUs belong to which taxonomy

5. VISUALISATION

Exercise 15: Sequence classification – part 4: Classify the OTUs

We want to visualise our classified OTUs → we will do that using Krona.
We need to convert our taxonomy file to a format compatible with Krona.

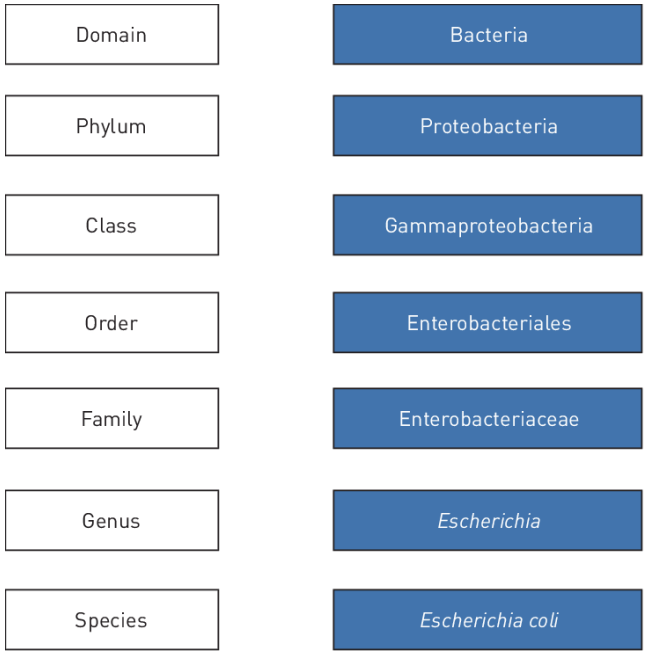
1. Taxonomy-to-Krona tool with the following parameters:
- *"Taxonomy file":* **the taxonomy output from Classify.otu (note: this is a collection input)**
- leave all other parameters to the default settings
- **Execute**



2. Krona pie chart tool with the following parameters:
- *"What is the type of your input data":* **Tabular**
- *"Input file":* **taxonomy output from Taxonomy-to-Krona (collection)**

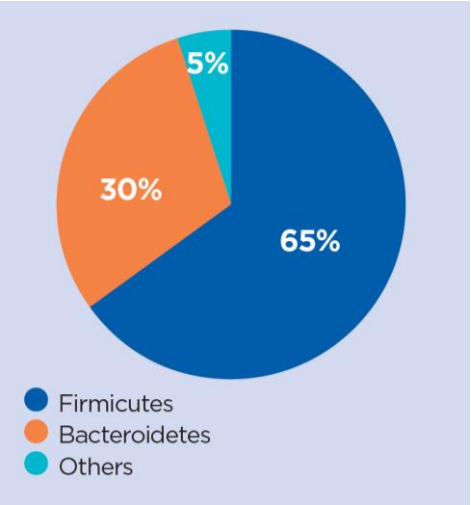
The output:

View the Krona pie chart HTML file. Switch between the samples in the left upper corner.



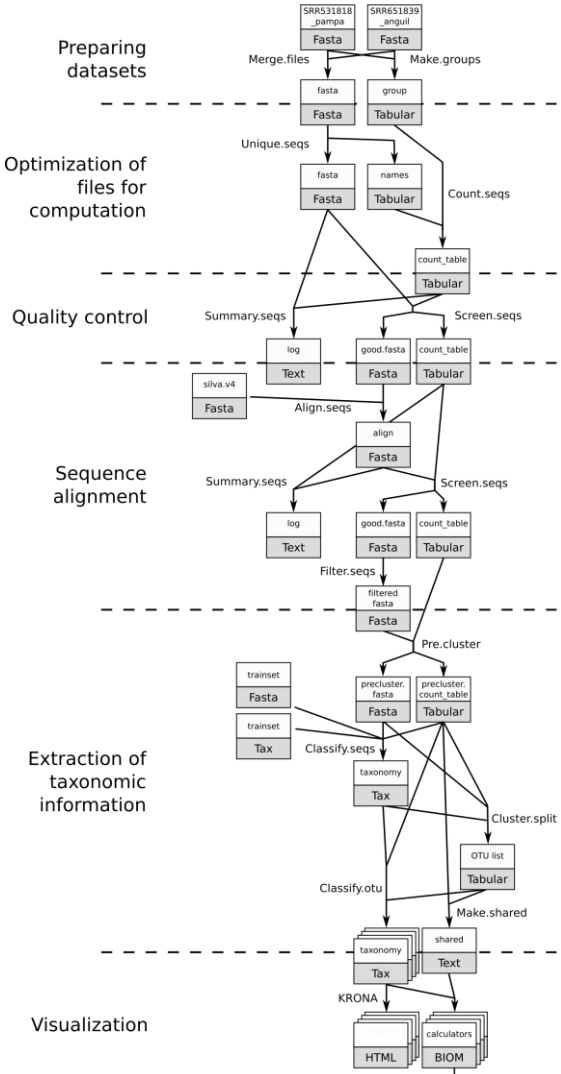
Discussion:

Are there any obvious differences between the samples?
Which sample has a higher percentage of *Bifidobacterium*?
Do you know any good or bad bacteria?
If not, google a bit if there is enough time and share your findings.



Do you have any idea how you might use these results in a real research; what could you compare? How can you use this for a gut-brain axis research?
Are you currently working on anything where you think this kind of analysis might be useful?

We've done something similar to this workflow...



If you want to learn more about metagenomics...

Read about:

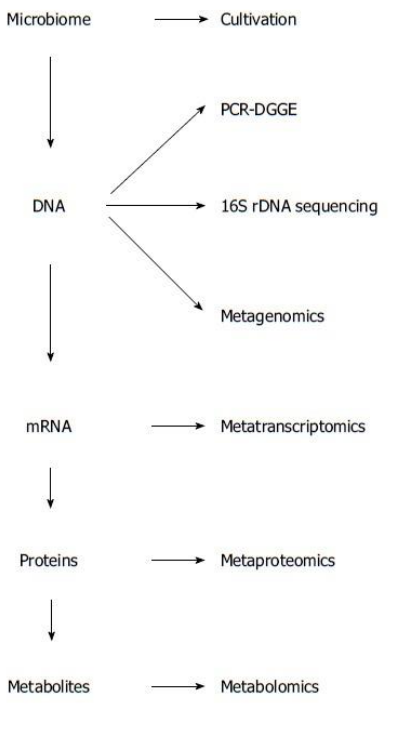
- paired-end sequencing
- fastq format & Phred quality score → FastQC for the quality control
- alpha and beta diversity
- different reference databases
- https://galaxyproject.eu/index-metagenomics.html
- http://qiime.org/

Watch some videos:

- Illumina Sequencing by Synthesis
- 16s rRNA sequencing

Most of the analysis is based on Galaxy Training materials.
Definitely check it out if you want to learn more!

Other meta-omics research



	Characteristics	Limitations	Applications
Microbiome → Cultivation	Visible bacterial colonies Low costs	Can not detect uncultured microbiota	Clinical diagnosis Obtaining target bacterial colonies
DNA → PCR-DGGE	Using 16S rDNA Revealing bacterial diversity Detecting microbial dysbiosis	Only baxonomic information Chimera production and PCR bias Except archaea and virus	Microbial composition dysbiosis Identifying healthy or disease specific species
DNA → 16S rDNA sequencing	Using 16S rDNA Revealing bacterial diversity Detecting microbial dysbiosis	Only baxonomic information Except archaea and virus	Microbial composition dysbiosis Identifying healthy or disease specific species
DNA → Metagenomics	Sequencing the total genes Uncovering microbial diversity Finding the novel genes	No microbial expressed functions Complex bioinformatic analysis Consuming costs and time	Revealing functional dysbiosis Finding disease specific microbial genes Identifying functional based studies
mRNA → Metatranscriptomics	Obtaining gene expression profiling Revealing different microbial gene expression among different physiological conditions	Poor stability of bacterial mRNA Requiring multiple purification steps Insufficient reference databases No unique protocol	Revealing functional dysbiosis Finding microbial activity kinetics Specific monitoring active bacteria
Proteins → Metaproteomics	Obtaining protein profiles Comparing microbial proteins among different physiological conditions	Insufficient reference databases Hard to extract total protein No unique protocol	Confirming microbial function Identifying eucaryotes-procaryotes analogs Clinical protein biomarkers
Metabolites → Metabolomics	Obtaining metabolic profiles Identifying metabolites among different physiological conditions	Insufficient reference databases Difficult to identify host or microbial metabolites No unique protocol	Revealing and confirming new pathways Identifying novel metabolic biomarkers

Thank you for your attention!

epavlinek@exaltum.eu

exaltum
scientific knowledge management