Integrative genomic reconstruction of carbohydrate utilization networks in bifidobacteria: global trends, local variability, and dietary adaptation

Aleksandr A. Arzamasov Dmitry A. Rodionov Matthew C. Hibberd
Janaki L. Guruge Marat D. Kazanov Semen A. Leyn James E. Kent
Kristija Sejane Lars Bode Michael J. Barratt Jeffrey I. Gordon
Andrei L. Osterman

Contents

1 Background

This supplementary code file describes:

1. Summary of reference genomes and functional roles
2. Phylogenetic analysis of reference *Bifidobacterium* genomes
3. Representation of predicted carbohydrate utilization phenotypes in 263 reference *Bifidobacterium* strains
4. Analysis of CAZyme representation in 263 *Bifidobacterium* genomes
5. Representation of predicted metabolic pathways in 26 *Bifidobacterium longum* genomes
6. Representation of predicted metabolic pathways in 2967 *Bifidobacterium* genomes
7. Analysis of *in vitro* growth data
8. Analysis of human milk oligosaccharide (HMO) consumption data
9. Analysis of RNA-seq data

---

2 Reproducibility and accessibility

All code used in this analysis (including the Rmarkdown document used to compile this supplementary code file) is available on GitHub **here**. Once the GitHub repo has been downloaded, navigate to `compendium_manuscript/` to find the Rmarkdown document as well as the RProject file. This should be your working directory for executing code.

1. To fully reproduce the phylogenetic analysis of 263 reference *Bifidobacterium* genomes, you will need to download FNA files from **Figshare**. Downloaded FNA files should be placed to `data/genomes/263_NR_ref_genomes/fna/`

2. To fully reproduce the analysis of the CAZyme representation in 263 reference *Bifidobacterium* genomes, you will need to download FAA files from **Figshare**. Downloaded FAA files should be placed to `data/genomes/263_NR_ref_genomes/faa/`
3. To fully reproduce the RNA-seq data analysis, you will need to download raw FASTQ files from Gene Expression Omnibus under accession **GSE239955**. Downloaded FASTQ files should be placed to `data/rnaseq/fastq/`. Otherwise, `data/rnaseq/kallisto/` already contains Kallisto mapping outputs

3 R packages and external functions

A set of R packages was used for this analysis. The pacman package was used to simplify downloading and loading the required packages. All graphics and data wrangling were handled using the tidyverse suite of packages. To fully reproduce the R environment used in the analysis, use the renv package and `renv::restore()` to restore the environment from `renv.lock`.

```
# install/load the pacman package for the rapid installation of required packages
if (!require("pacman")) install.packages("pacman")
# use pacman to install/load all packages needed for the analysis
pacman::p_load("tidyverse", "patchwork", "ComplexHeatmap", "ggbeeswarm", "ggrepel",
               "circlize", "ggpubr", "tximport", "rhdf5", "gt", "edgeR", "cowplot", "limma",
               "vegan", "car", "AER", "emmeans")
set.seed(1992)
```

A set of external R functions was used to keep the code tidy. All used R scripts with functions can be found in `compendium_manuscript/code/`.

```
source("code/growth_curve_plotter.R") # plots growth curves
source("code/calculate_statistics.R") # calculates the number of true positives (TP),
# true negatives (TN), false positives (FP), false negatives (FN)
# by comparing predicted binary carbohydrate utilization phenotypes ("1" and "0")
# with growth phenotypes ("+/w" and "-")
source("code/profile.R") # calculates counts per million (CPM) for each gene and
# plots the distribution of CPM values for each sample
source("code/deg_list.R") # selects differentially expressed genes (DEGs) based on input cut-offs
# outputs an annotated table with DEGs to a file
```

4 Summary of reference genomes and functional roles

4.1 Introduction

This block contains code used for summarizing information about (i) 263 reference non-redundant *Bifidobacterium* genomes and (ii) functional roles.

4.2 Load data

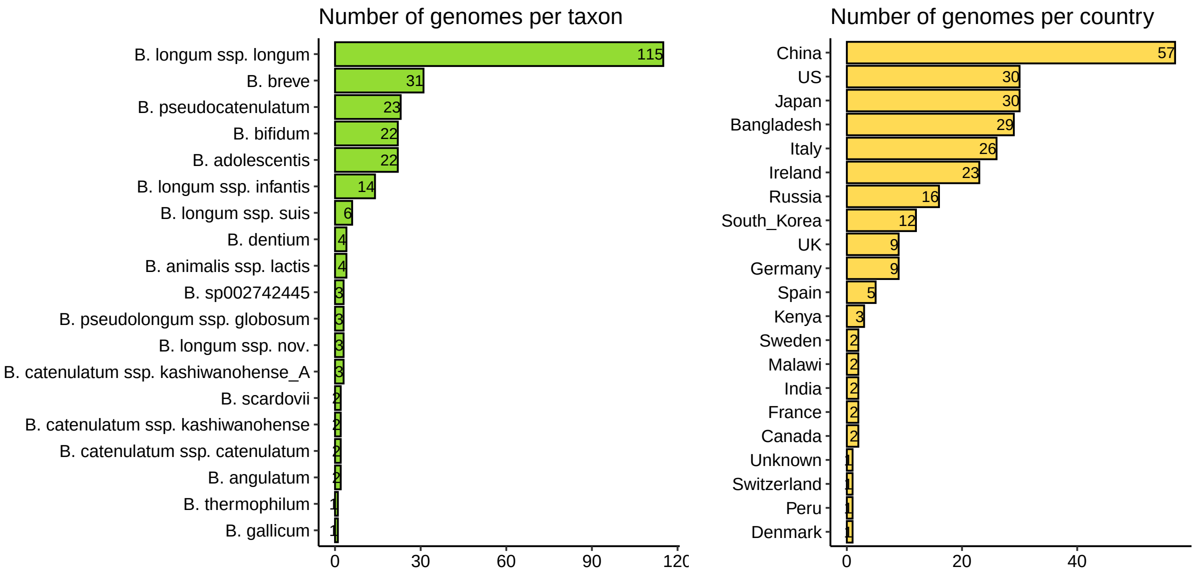
```r
# table with data on 263 genomes
info263.df <- read_tsv("data/tables/BPM_263_NR_genomes_carbs.txt") %>%
  dplyr::select(genome_ID, curated_taxonomy, country)
# table with functional roles
fr.df <- read_tsv("data/tables/2024-04-15_functional_roles.txt")
```

4.3 Summary of 263 reference *Bifidobacterium* genomes

Here we summarize information about the set of 263 reference genomes: (i) number per species/subspecies, (ii) number per country.

```r
# replace specific strings in curated_taxonomy column
info263.df <- info263.df %>%
  mutate(curated_taxonomy = str_replace_all(curated_taxonomy, "Bifidobacterium", "B."),
                             curated_taxonomy = str_replace_all(curated_taxonomy, "subsp.", "ssp."))
# change data type from character to (sorted) factor
info263.df$curated_taxonomy <- factor(info263.df$curated_taxonomy,
                                      levels = names(sort(table(info263.df$curated_taxonomy),
                                      increasing = TRUE)))
info263.df$country <- factor(info263.df$country,
                            levels = names(sort(table(info263.df$country),
                            increasing = TRUE)))

# barplot depicting the number of genomes per taxon
theme_set(theme_classic())
bp_taxa <- ggplot(info263.df, aes(x = curated_taxonomy)) +
  geom_bar(fill = "#79d400", colour = "black", alpha = 0.8, size = 0.5) +
  # count the number of genomes in each group and plot it
  stat_count(aes(label = ..count..), geom = "text", hjust = 1, size = 3) +
  labs(title = "Number of genomes per taxon") +
  # remove the axes labels
  xlab("") +
  ylab("") +
  coord_flip() +
  theme(axis.text.x = element_text(size = 10, color = "black"),
        axis.text.y = element_text(size = 10, color = "black"))

# barplot depicting the number of genomes per country
bp_country <- ggplot(info263.df, aes(x = country)) +
  geom_bar(fill = "#ffd12a", colour = "black", alpha = 0.8, size = 0.5) +
  # count the number of genomes in each group and plot it
  stat_count(aes(label = ..count..), geom = "text", hjust = 1, size = 3) +
  labs(title = "Number of genomes per country") +
  # remove the axes labels
  xlab("") +
  ylab("") +
  coord_flip() +
  theme(axis.text.x = element_text(size = 10, color = "black"),
        axis.text.y = element_text(size = 10, color = "black"))

# combine barplots using the patchwork package
bp_combined <- bp_taxa + bp_country
```

```r
plot(bp_combined)
```



```r
# save the figure to a file
ggsave("results/phylogeny/summary_263_genomes.pdf", device = "pdf", width = 15, height = 5)
```

4.4 Summary of functional roles

Here we calculate:

- The number of unique functional roles (total and stratified by (i) experimental evidence and (ii) type)
- The number of publications from which data about functional roles were collected

```r
# calculate the total number of unique functional roles
num_roles <- fr.df %>%
  distinct(annotation) %>%
  nrow()
# calculate the total number of unique characterized functional roles
num_characterized_roles <- fr.df %>%
  filter(evidence == "characterized") %>%
  distinct(annotation) %>%
  nrow()
# calculate the total number of unique predicted functional roles
num_predicted_roles <- fr.df %>%
  filter(evidence == "predicted") %>%
  distinct(annotation) %>%
  nrow()
# calculate the total number of unique novel predicted functional roles
num_predicted_new_roles <- fr.df %>%
  filter(evidence == "predicted_new") %>%
  distinct(annotation) %>%
  nrow()

# stratify the functional roles by type
num_transporters <- fr.df %>%
  filter(type == "transporter") %>%
  distinct(annotation) %>%
  nrow()
num_donwstream <- fr.df %>%
  filter(type == "downstream_catabolism") %>%
  distinct(annotation) %>%
  nrow()
num_cazy <- fr.df %>%
  filter(type == "CAZyme") %>%
  distinct(annotation) %>%
  nrow()
num_reg <- fr.df %>%
  filter(type == "regulator") %>%
  distinct(annotation) %>%
  nrow()

# calculate the total number of publications
# split PMID values and extract unique values
unique_pmids <- unique(unlist(strsplit(fr.df$PMID, ";")))
num_unique_pmids <- length(unique_pmids)

# print the counts
fr_output <- paste("Total functional roles:", num_roles, "\n",
                "Total characterized functional roles:", num_characterized_roles, "\n",
                "Total predicted functional roles:", num_predicted_roles - 1, "\n",
                "Total novel predicted functional roles:", num_predicted_new_roles, "\n",
                "Transporters and their components:", num_transporters, "\n",
                "Downstream catabolic enzymes:", num_donwstream, "\n",
                "CAZymes:", num_cazy, "\n",
                "Transcriptional regulators:", num_reg, "\n",
                "Total number of publications:", num_unique_pmids - 1, "\n")
cat(fr_output)
```

```
## Total functional roles: 565
##  Total characterized functional roles: 200
##  Total predicted functional roles: 225
##  Total novel predicted functional roles: 140
##  Transporters and their components: 226
##  Downstream catabolic enzymes: 67
##  CAZymes: 188
##  Transcriptional regulators: 84
##  Total number of publications: 144
```

5 Phylogenetic analysis of reference *Bifidobacterium* genomes

5.1 Introduction

This block contains the code for building:

1. Phylogenetic tree of 263 reference *Bifidobacterium* genomes. The topology of the resulting tree was manually inspected to check (and correct if needed) taxonomic assignments of genomes based on their co-clustering with branches corresponding to the type or well-characterized strains of various *Bifidobacterium* taxa

2. Average Nucleotide Identity (ANI) matrices for select strains belonging to *Bifidobacterium longum* and *Bifidobacterium catenulatum* species

The following software is required:

1. Prokka (v1.14.6)
2. Panaroo (v1.3.2)
3. CD-HIT (v4.8.1)
4. MAFFT (v7.515)
5. IQ-TREE (v2.2.0.3)
6. pyani (v0.2.12)

To install these tools, you can use mamba and yml files in `envs` to create respective environments. **Note**: installation via `mamba` was tested on macOS and may not always work on Linux-based operating systems. In the latter case, you may need to install the required software manually.

1. `mamba env create -f envs/prokka.yml` # Prokka (v1.14.6)
2. `mamba env create -f envs/panaroo.yml` # Panaroo (v1.3.2); CD-HIT (v4.8.1); MAFFT (v7.515)
3. `mamba env create -f envs/iqtree.yml` # IQ-TREE (v2.2.0.3)
4. `mamba env create -f envs/pyani.yml` # pyani (v0.2.12)

Note: FNA files could not be stored in the GitHub repo due to size limitations. Thus, you will need to download them from **Figshare**. Put downloaded FNA files to `data/genomes/263_NR_ref_genomes/fna/`.

5.2 Annotating genomes using Prokka

We used Prokka for annotating genomes. Prokka takes contig nucleotide fasta (FNA) and outputs annotated genomes in a standardized format (GFF3), which is recognized by many pangenome calculating tools. The script that performs annotation can be found in `code/run_prokka.sh/`. A simple bash operation at the end of the script collects all created GFF3 files and puts them in the `data/genomes/263_NR_ref_genomes/gff/` folder.

```
source ~/.bash_profile
source code/run_prokka.sh
```

5.3 Calculating pangenome using Panaroo

We used Panaroo to identify core genes shared among 263 reference *Bifidobacterium* genomes. Prokka-annotated GFF3 files were used as input. Since genomes of multiple different *Bifidobacterium* species were used, we relaxed the sequence identity threshold (`--threshold 0.8`) and length difference cutoff (`--len_dif_percent 0.9`). As part of the Panaroo pipeline, concatenated nucleotide sequences of 487 identified core genes were aligned via MAFFT.

```
source ~/.bash_profile
#set -ex

### SOFTWARE SETUP ##
####################
# required tools: panaroo=1.3.2; cd-hit=4.8.1; mafft=7.515
# set the name of the environment with installed tools
environment_name="panaroo"
# activate selected conda environment
eval "$(command conda 'shell.bash' 'hook' 2> /dev/null)" # initializes conda in sub-shell
conda activate ${environment_name}
conda info|egrep "conda version|active environment"

# run panaroo
mkdir -p data/pangenome
panaroo -i data/genomes/263_NR_ref_genomes/gff/*.gff \
-o data/pangenome/panaroo_strict_i80_l90 \
--clean-mode strict \
-a core --aligner mafft \
--threshold 0.8 \
--len_dif_percent 0.9 \
-t 16
```

5.4 Building phylogenetic tree using IQ-TREE

We used a maximum-likelihood-based algorithm with ultrafast bootstrap approximation (UFBoot) implemented in IQ-TREE to build a phylogenetic tree based on the alignment of the core genes. Depending on your computing power, this process may take multiple days; you can use the prebuilt tree (`data/phylogeny/tree_263_NR_ref_genomes/tree_263_NR_genomes.treefile`) as an alternative.

```
source ~/.bash_profile
#set -ex

### SOFTWARE SETUP ##
####################
# required tools: iqtree=2.2.0.3
# set the name of the environment with installed tools
environment_name="iqtree"
# activate selected conda environment
eval "$(command conda 'shell.bash' 'hook' 2> /dev/null)" # initializes conda in sub-shell
conda activate ${environment_name}
conda info|egrep "conda version|active environment"

# copy the filtered alignment file to a new folder
mkdir -p data/phylogeny/tree_263_NR_ref_genomes
```

```
cp data/pangenome/panaroo_strict_i80_l90/core_gene_alignment.aln \
data/phylogeny/tree_263_NR_ref_genomes/core_gene_alignment.aln
cd data/phylogeny/tree_263_NR_ref_genomes

# build the tree
iqtree -s core_gene_alignment.aln -o 561180.4 -m GTR+F+R10 -B 1000 -T 16
# if you do not have much time or computing power, turn on the fast tree search mode
#iqtree -s core_gene_alignment.aln -o 561180.4 -m GTR+F+R10 -T 16 -fast
```

5.5 Visualizing phylogenetic tree

The pheylogentic tree of 263 *Bifidobacterium* genomes was manually visualized in iTOL.

5.6 Calculating ANI of *Bifidobacterium longum* genomes

The phylogenomic analysis indicated that the *Bifidobacterium longum* species might have a more complex subspecies structure than previously described. To investigate it further, we computed pairwise ANI indices of 15 reference and 11 additional *Bifidobacterium longum* genomes using the ANIb algorithm implemented in pyani. For comparative purposes, the 11 additional genomes included isolates of non-human origin, such as type strains of *Bifidobacterium longum* subsp. *suis* and *Bifidobacterium longum* subsp. *suillum*.

To run the analysis, put the 26 corresponding FNA files to `data/genomes/26_Blongum_genomes/`.

```
source ~/.bash_profile
#set -ex

### SOFTWARE SETUP ##
####################
# required tools: pyani=0.2.12
# set the name of the environment with installed tools
environment_name="pyani"
# activate selected conda environment
eval "$(command conda 'shell.bash' 'hook' 2> /dev/null)" # initializes conda in sub-shell
conda activate ${environment_name}
conda info|egrep "conda version|active environment"

# run pyani
average_nucleotide_identity.py -i data/genomes/26_Blongum_genomes/ \
-o data/phylogeny/ANIb_Blongum \
-m ANIb
```

Load data to R.

```
# read the table with calculated ANI values for 26 Bifidobacterium longum genomes
ANIb_Blon <- read_tsv("data/phylogeny/ANIb_26_Blongum_genomes/ANIb_percentage_identity.tab") %>%
  # use the first column as row names
  column_to_rownames(var="...1")
```

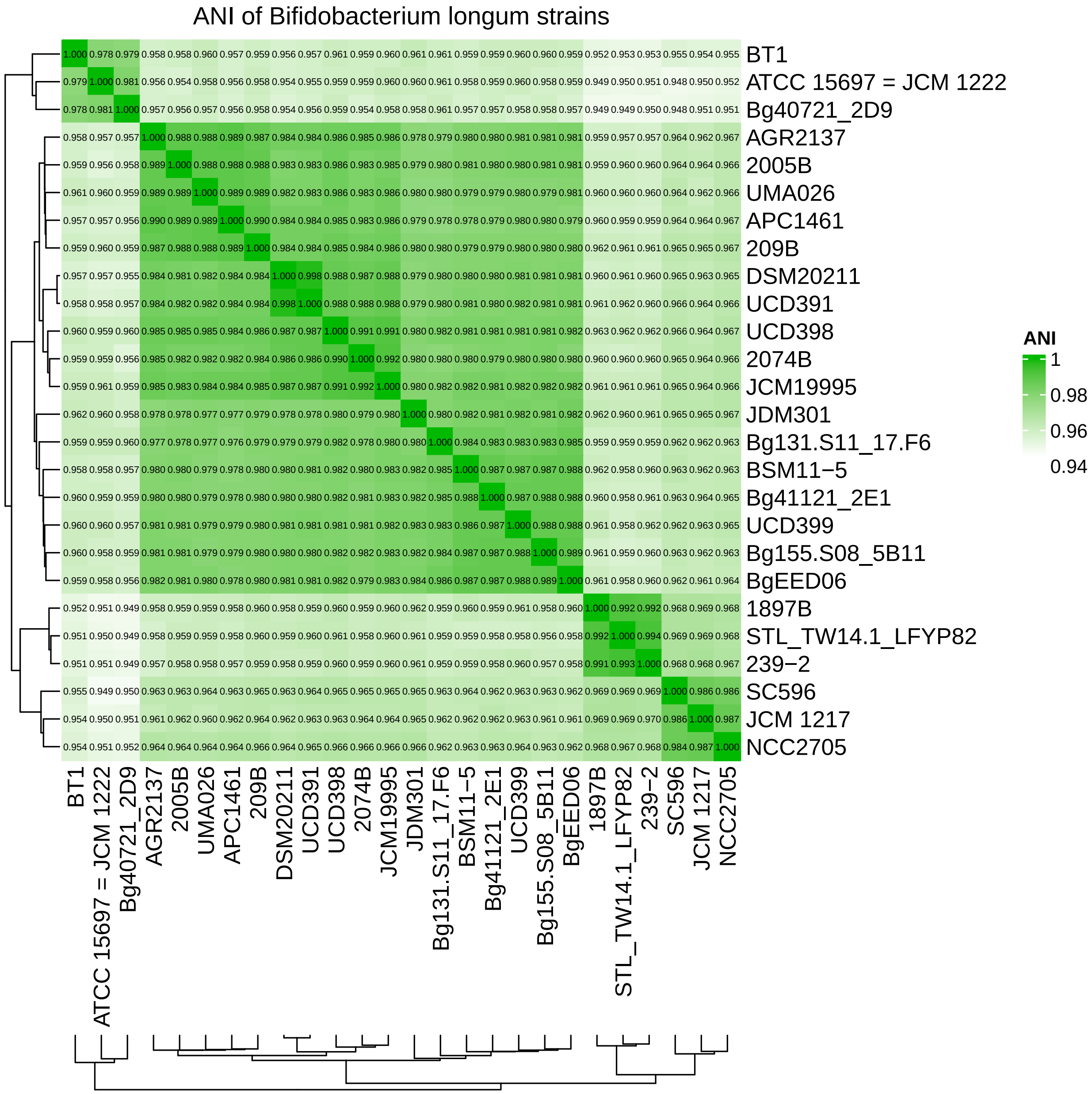
The following code chunk below creates a heatmap based on calculated ANI values.

Hierarchical clustering options:

- **Distance metric**: Maximum distance
- **Linkage method**:Average method

```
# convert the tibble with ANI values to a matrix
ANIb_Blon_mat <- as.matrix(ANIb_Blon)
ANIb_Blon_mat <- round(ANIb_Blon_mat, digits = 3)
old_to_new_names <- c("216816.186" = "APC1461",
                      "216816.1989" = "Bg155.S08_5B11",
                      "1682.76" = "Bg41121_2E1",
                      "216816.378" = "BgEED06",
                      "759350.3" = "JDM301",
                      "Bsuis_BSM11-5" = "BSM11-5",
                      "391904.5" = "ATCC 15697 = JCM 1222",
                      "1682.151" = "Bg40721_2D9",
                      "1682.24" = "BT1",
                      "565042.3" = "JCM 1217",
                      "206672.9" = "NCC2705",
                      "216816.144" = "1897B",
                      "216816.1981" = "STL_TW14.1_LFYP82",
                      "216816.262" = "239-2",
                      "1679.217" = "SC596",
                      "1695.38" = "Bg131.S11_17.F6")

# update row names only for specified genomes, preserving the rest
rownames(ANIb_Blon_mat) <- ifelse(rownames(ANIb_Blon_mat) %in% names(old_to_new_names),
                                  old_to_new_names[rownames(ANIb_Blon_mat)],
                                  rownames(ANIb_Blon_mat))

# update column names only for specified ones, preserving the rest
colnames(ANIb_Blon_mat) <- ifelse(colnames(ANIb_Blon_mat) %in% names(old_to_new_names),
                                  old_to_new_names[colnames(ANIb_Blon_mat)],
                                  colnames(ANIb_Blon_mat))

# set colors
ANI_col_fun <- circlize::colorRamp2(c(0.945, 1), c("white", "#00b900"))
# create a function that will add ANI values to each cell
ANI_cell_fun = function(j, i, x, y, w, h, fill){
  grid.rect(x, y, w, h, gp = gpar(fill = fill, col = fill))
  # add ANI values to each cell
   if(ANIb_Blon_mat[j, i] <= 1){
    grid.text(sprintf("%.3f", ANIb_Blon_mat[j, i]), x, y, gp = gpar(fontsize = 5))
  }
}

# define the distance measure and clustering method
dist_method = "maximum"
clust_method = "average"

# compute hierarchical clustering for rows
row_dist <- dist(t(ANIb_Blon_mat), method = dist_method)
row_clust <- hclust(row_dist, method = clust_method)

# compute hierarchical clustering for columns
```

```
col_dist <- dist(ANIb_Blon_mat, method = dist_method)
col_clust <- hclust(col_dist, method = clust_method)

# specify the name of the output
pdf("results/phylogeny/ANIb_26_Blongum.pdf", width=8, height=8)
# plot the heatmap
ANIb_Blon_ht <- ComplexHeatmap::Heatmap(ANIb_Blon_mat,
                             rect_gp = gpar(type = "none"),
                             column_dend_side = "bottom",
                             column_title = "ANI of Bifidobacterium longum strains",
                             name = "ANI",
                             col = ANI_col_fun,
                             cell_fun = ANI_cell_fun,
                             cluster_rows = row_clust,
                             cluster_columns = col_clust,
                             row_names_side = "right")

draw(ANIb_Blon_ht)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ANIb_Blon_ht)
```



5.7 Calculating ANI of *Bifidobacterium catenulatum* genomes

The phylogenetic analysis indicated that the *Bifidobacterium catenulatum* species might have a more complex subspecies structure than previously described. To investigate it further, we computed pairwise ANI indices of 10 reference *Bifidobacterium catenulatum* genomes using the ANIb algorithm implemented in pyani. To run the analysis, put the 10 corresponding FNA files to `data/genomes/10_Bcatenulatum_genomes/`.

```
source ~/.bash_profile
#set -ex

### SOFTWARE SETUP ##
####################
# required tools: pyani=0.2.12
# set the name of the environment with installed tools
environment_name="pyani"
# activate selected conda environment
eval "$(command conda 'shell.bash' 'hook' 2> /dev/null)" # initializes conda in sub-shell
conda activate ${environment_name}
conda info|egrep "conda version|active environment"

# run pyani
average_nucleotide_identity.py -i data/genomes/10_Bcatenulatum_genomes/ \
-o data/phylogeny/ANIb_Bcat \
-m ANIb
```

Load data into R.

```
# read the file with calculated ANI values for 10 Bifidobacterium catenulatum genomes
ANIb_Bcat <- read_tsv("data/phylogeny/ANIb_10_ref_Bcatenulatum_genomes/ANIb_percentage_identity.tab") %
  # use the first column as row names
  column_to_rownames(var="...1")
```

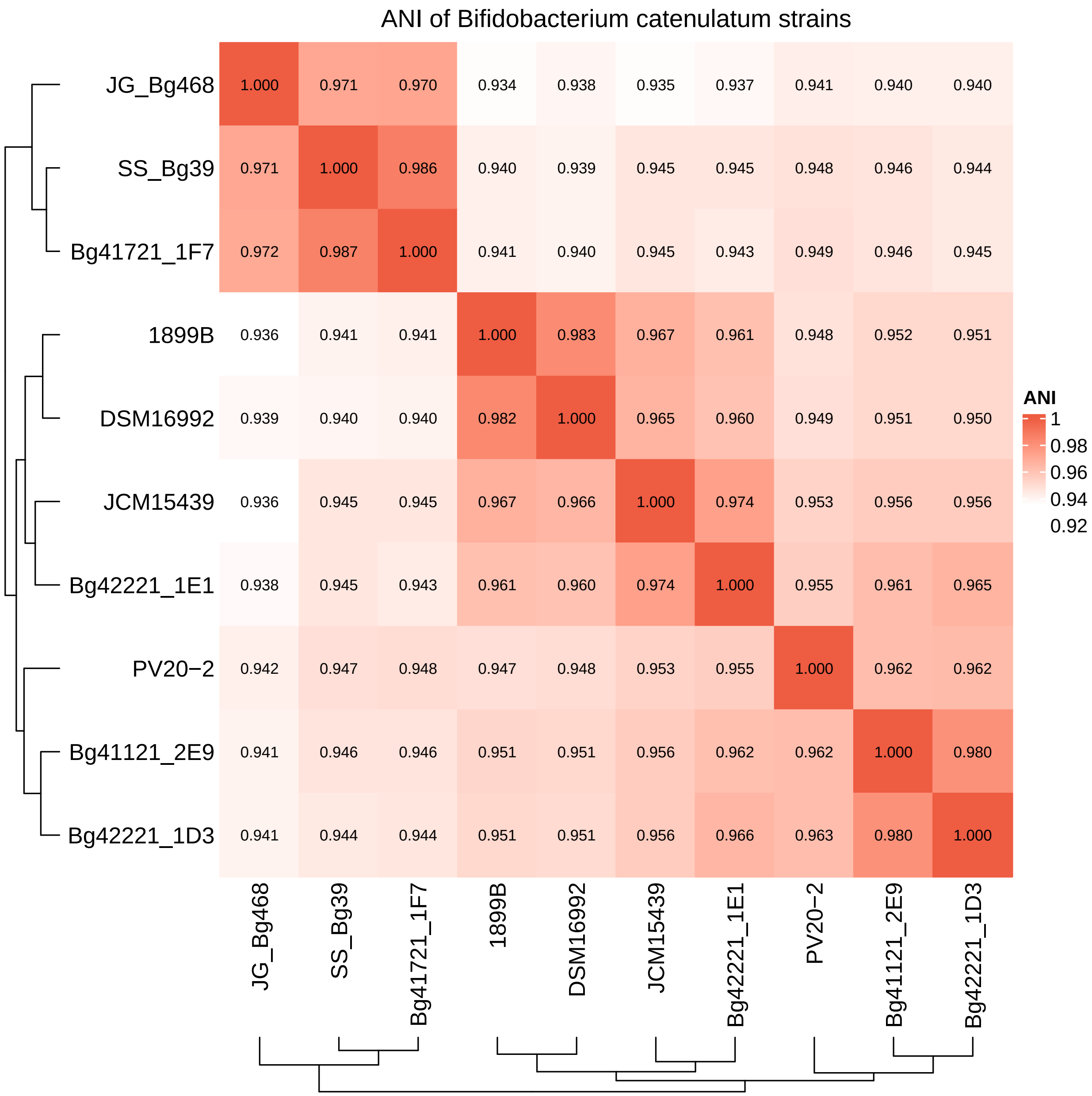
The following code chunk below creates a heatmap based on calculated ANI values.

Hierarchical clustering options:

- **Distance metric**: Maximum distance
- **Linkage method**:Average method

```
# convert the tibble with ANI values to a matrix
ANIb_Bcat_mat <- as.matrix(ANIb_Bcat)
ANIb_Bcat_mat <- round(ANIb_Bcat_mat, digits = 3)

# set colors
ANI_col_fun <- circlize::colorRamp2(c(0.935, 1), c("white", "tomato2"))
# create a function that will add ANI values to each cell
ANI_cell_fun = function(j, i, x, y, w, h, fill){
  grid.rect(x, y, w, h, gp = gpar(fill = fill, col = fill))
  # add ANI values to each cell
   if(ANIb_Blon_mat[j, i] <= 1){
    grid.text(sprintf("%.3f", ANIb_Bcat_mat[j, i]), x, y, gp = gpar(fontsize = 8))
  }
}

# define the distance measure and clustering method
dist_method = "maximum"
clust_method = "average"

# compute hierarchical clustering for rows
row_dist <- dist(t(ANIb_Bcat_mat), method = dist_method)
row_clust <- hclust(row_dist, method = clust_method)

# compute hierarchical clustering for columns
col_dist <- dist(ANIb_Bcat_mat, method = dist_method)
col_clust <- hclust(col_dist, method = clust_method)
```

```
# specify the name of the output
pdf("results/phylogeny/ANb_10_Bcatenulatum.pdf", width=8, height=8)
# plot the heatmap
ANIb_Bcat_ht <- ComplexHeatmap::Heatmap(ANIb_Bcat_mat,
                             rect_gp = gpar(type = "none"),
                             column_dend_side = "bottom",
                             column_title = "ANI of Bifidobacterium catenulatum strains",
                             name = "ANI",
                             col = ANI_col_fun,
                             cell_fun = ANI_cell_fun,
                             cluster_rows = row_clust,
                             cluster_columns = col_clust,
                             row_names_side = "left")

draw(ANIb_Bcat_ht)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ANIb_Bcat_ht)
```



	JG_Bg468	SS_Bg39	Bg41721_1F7	1899B	DSM16992	JCM15439	Bg42221_1E1	PV20−2	Bg41121_2E9	Bg42221_1D3
JG_Bg468	1.000	0.971	0.970	0.934	0.938	0.935	0.937	0.941	0.940	0.940
SS_Bg39	0.971	1.000	0.986	0.940	0.939	0.945	0.945	0.948	0.946	0.944
Bg41721_1F7	0.972	0.987	1.000	0.941	0.940	0.945	0.943	0.949	0.946	0.945
1899B	0.936	0.941	0.941	1.000	0.983	0.967	0.961	0.948	0.952	0.951
DSM16992	0.939	0.940	0.940	0.982	1.000	0.965	0.960	0.949	0.951	0.950
JCM15439	0.936	0.945	0.945	0.967	0.966	1.000	0.974	0.953	0.956	0.956
Bg42221_1E1	0.938	0.945	0.943	0.961	0.960	0.974	1.000	0.955	0.961	0.965
PV20−2	0.942	0.947	0.948	0.947	0.948	0.953	0.955	1.000	0.962	0.962
Bg41121_2E9	0.941	0.946	0.946	0.951	0.951	0.956	0.962	0.962	1.000	0.980
Bg42221_1D3	0.941	0.944	0.944	0.951	0.951	0.956	0.966	0.963	0.980	1.000

6 Representation of predicted carbohydrate utilization phenotypes in 263 reference *Bifidobacterium* strains

6.1 Introduction

The block describes the various analyses of Binary Phenotype Matrix (BPM) containing 66 binary carbohydrate utilization phenotypes predicted for 263 reference *Bifidobacterium* strains. Four additional phenotypes (GalNAc, ManNAc, Man, GalA) were excluded from the analysis since, for these glycans, all strains had predicted binary phenotype **0**.

6.2 Load data

```
# read the BPM (carbs)
bpm_263_carb_df <- read_tsv("data/tables/BPM_263_NR_genomes_carbs.txt",
                            col_types = cols(.default = "c")) %>%
                            mutate_at(c(4:73), as.numeric) %>%
                            dplyr::select(-c(ManNAc, GalNAc, Man, GalA)) %>%
                            arrange(genome_ID)
# read the table with metadata for predicted carbohydrate utilization phenotypes
phenotype_metadata <- read_tsv("data/tables/phenotype_metadata_carbs.txt",
                      col_types = cols(.default = "c")) %>%
                      filter(!(phenotype %in% c("ManNAc", "GalNAc", "Man", "GalA")))
```

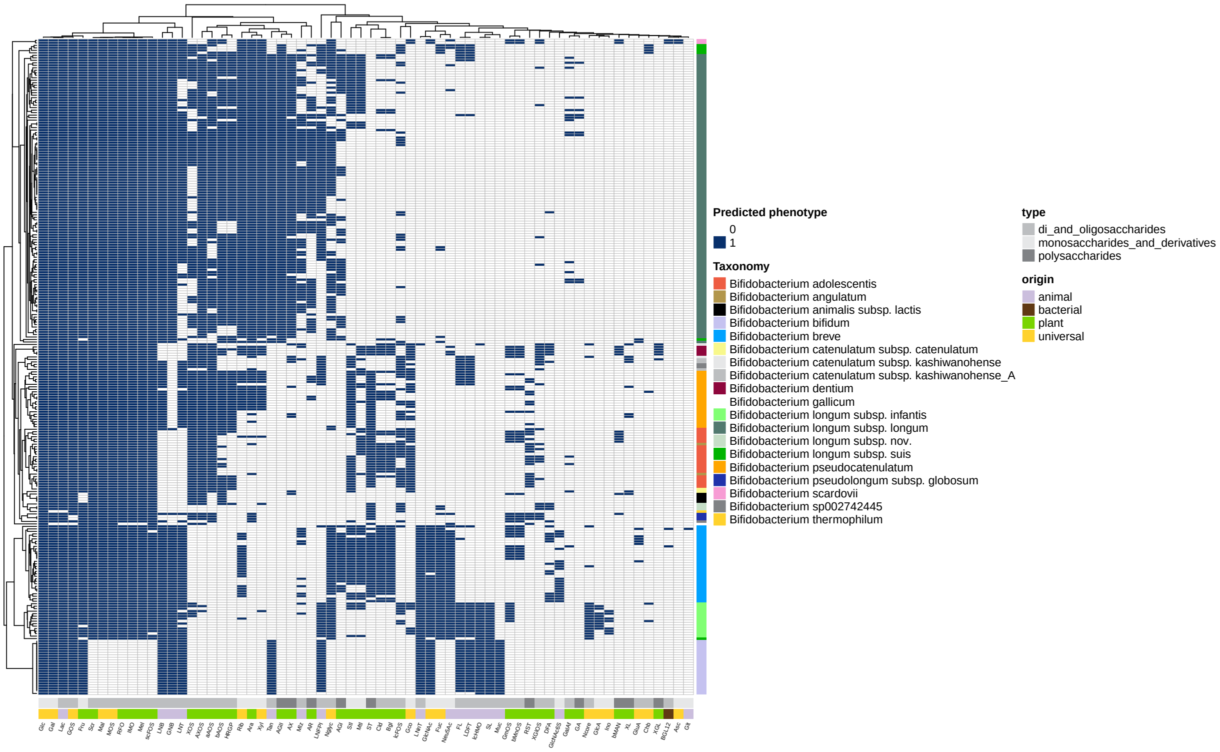
6.3 Hierarchical clustering of the BPM for 263 *Bifidobacterium* genomes

The following heatmap shows the hierarchical clustering of the BPM for 263 *Bifidobacterium* genomes.

Hierarchical clustering options:

- **Distance metric**: Hamming distance (equivalent to Manhattan distance for binary data)
- **Linkage method**: Average

```
# extract the binary matrix
bpm_263_mat <- as.matrix((bpm_263_carb_df[, 4:69]))
# add rownames to the matrix
rownames(bpm_263_mat) <- bpm_263_carb_df$genome_ID

# create a vector with taxonomy (group)
taxonomy <- bpm_263_carb_df$curated_taxonomy
# extract vectors containing data about glycan type and origin
glycan_type <- phenotype_metadata$type_group
glycan_origin <- phenotype_metadata$origin
# create a coloring function
col_fun <- structure(c("white", "#08306b"), names = c("0", "1"))
# create a row annotation specifying taxonomy
ha_263_1 <- HeatmapAnnotation(
  which = c("row"),
  Taxonomy = taxonomy,
  col = list(Taxonomy = c("Bifidobacterium adolescentis" = "tomato2",
                          "Bifidobacterium angulatum" = "#b2964b",
                          "Bifidobacterium animalis subsp. lactis" = "black",
                          "Bifidobacterium bifidum" = "#c5c2f0",
                          "Bifidobacterium breve" = "#00a2ff",
                          "Bifidobacterium catenulatum subsp. catenulatum" = "#f8f88b",
                          "Bifidobacterium catenulatum subsp. kashiwanohense" = "#E6E7E8",
                          "Bifidobacterium catenulatum subsp. kashiwanohense_A" = "#BCBEC0",
                          "Bifidobacterium dentium" = "#8e063c",
                          "Bifidobacterium gallicum" = "#ffffff",
                          "Bifidobacterium longum subsp. infantis" = "#81FF74",
                          "Bifidobacterium longum subsp. longum" = "#51796f",
                          "Bifidobacterium longum subsp. suis" = "#00b400",
                          "Bifidobacterium longum subsp. nov." = "#c6dec7",
                          "Bifidobacterium pseudocatenulatum" = "#ffa600",
                          "Bifidobacterium pseudolongum subsp. globosum" = "#2032ab",
                          "Bifidobacterium scardovii" = "#f79cd4",
                          "Bifidobacterium sp002742445" = "#808285",
                          "Bifidobacterium thermophilum" = "#ffd22d")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))
# create two column annotations specifying glycan type and origin
ha_263_2 <- HeatmapAnnotation(
  type = glycan_type,
  origin = glycan_origin,
  col = list(type = c("monosaccharides_and_derivatives" = "#E6E7E8",
                      "di_and_oligosaccharides" = "#BCBEC0",
                      "polysaccharides" = "#808285"),
             origin = c("universal" = "#ffd22d",
                        "animal" = "#cbbedd",
                        "plant" = "#79d400",
                        "bacterial" = "#603913")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))

# plot the heatmap
pdf("results/phenotypes/BPM_263_heatmap.pdf", width=15, height=10)
ht_263_all <- ComplexHeatmap::Heatmap(bpm_263_mat,
                              name = "Predicted phenotype",
                              right_annotation = ha_263_1,
                              bottom_annotation = ha_263_2,
                              col = col_fun,
                              clustering_distance_rows = function(m)
                                dist(m, method = "manhattan"),
                              clustering_distance_columns = function(m)
                                dist(m, method = "manhattan"),
                              clustering_method_rows = "average",
                              clustering_method_columns = "average",
                              rect_gp = gpar(col = "grey", lwd = 0.05),
                              # do not show row names
                              show_row_names = FALSE,
                              row_names_gp = gpar(fontsize = 3),
                              column_names_gp = gpar(fontsize = 5),
                              column_names_rot = 70,
                              width = unit(200, "mm"),
                              height = unit(200, "mm"))
draw(ht_263_all)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ht_263_all)
```



6.4 Conservation of predicted carbohydrate utilization phenotypes within taxonomic groups

```
# calcualte "average" phenotypes for each taxon
bpm_263_mean <- bpm_263_carb_df %>%
  group_by(curated_taxonomy) %>%
  summarise_at(vars(Glc:Asc), list(mean))

# count columns with only 0s and 1s
#  i.e., phenotypes that are always conserved within taxonomic groups
binary_columns <- sum(apply(bpm_263_mean , 2, function(col) all(col %in% c(0, 1))))

# count columns with intermediate values in the [0,1] range
# i.e., phenotypes that vary within taxonomic groups
range_columns <- sum(apply(bpm_263_mean, 2, function(col) any(col > 0 & col < 1)))
```

Number of predicted phenotypes that are always conserved within taxonomic groups: **11**

Number of predicted phenotypes for which there is variability at least in one taxon: **55**

7 Analysis of CAZyme representation in 263 *Bifidobacterium* genomes

7.1 Introduction

This block contains the code used to analyze the representation of genes encoding Carbohydrate Active Enzymes (CAZymes, specifically glycoside hydrolases (GHs), carbohydrate esterases (CEs), and polysaccharide lyases (PLs)) in 263 *Bifidobacterium* genomes. In addition, we checked how many of these CAZymes were captured in the curated mcSEED subsystems.

The following software is required:

1. dbCAN (v4.0.0)

Note: Given the potential challenges with installing and running dbCAN, we provide processed dbCAN outputs in `data/CAZyme`. If you wish to run dbCAN by yourself: (i) install dbCAN and its associated databases, check the installation instructions here, (ii) download FAA files from **Figshare**. Put downloaded FAA files to `data/genomes/263_NR_ref_genomes/faa/`.

Processed dbCAN outputs:

1. `GH_output_subfamilies` # concatenated dbCAN output
2. `CAZyme_families` # representation of GH/CE/PL families in 263 *Bifidobacterium* genomes. GH subfamilies are collapsed (e.g., GH43_22 and GH_43_24 are treated as GH43)
3. `CAZyme_subfamilies` # representation of GH/CE/PL families and subfamilies in 263 *Bifidobacterium* genomes. GH subfamilies are treated as distinct columns

7.2 Load data

```r
# read the table with metadata for 263 genomes
bpm_263_join <- read_tsv("data/tables/BPM_263_NR_genomes_carbs.txt",
                         col_types = cols(.default = "c")) %>%
                         mutate_at(c(4:73), as.numeric) %>%
                         arrange(genome_ID) %>%
                         dplyr::select(genome_ID, genome_name, curated_taxonomy)
# read the table with CAZyme representation
cazy_subfam_df <- read_tsv("data/CAZyme/CAZyme_subfamilies.txt",
                           col_types = cols(.default = "c")) %>%
                           mutate_at(c(3:111), as.numeric) %>%
                           left_join(bpm_263_join, by = c("Organism" = "genome_name")) %>%
                           dplyr::select(-seed_id, -CE0, -GH0) %>%
                           arrange(genome_ID)
# read the processed dbCAN output
cazy_df <- read_tsv("data/CAZyme/GH_output_subfamilies.txt",
                    col_types = cols(.default = "c"))
```

7.3 Set colors

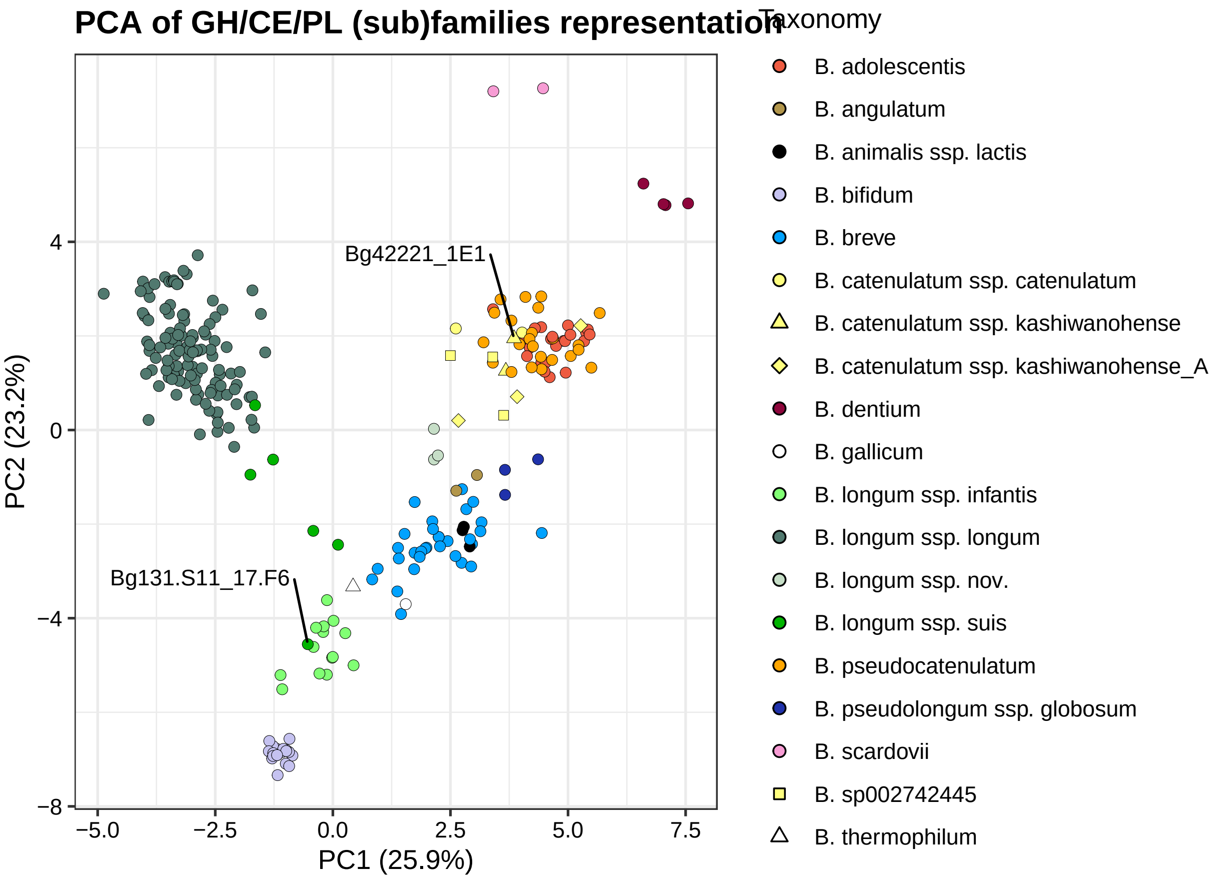
Define point colors and shapes used throughout this block.

```
# arguments for scale_shape_manual
genomes_263_shapes <- c(21,21,21,21,21,
                        21,24,23,21,21,
                        21,21,21,21,21,
                        21,21,22,24)
# arguments for scale_fill_manual
# breaks (how groups are encoded in the table)
genomes_263_breaks <- c("Bifidobacterium adolescentis",
                        "Bifidobacterium angulatum",
                        "Bifidobacterium animalis subsp. lactis",
                        "Bifidobacterium bifidum",
                        "Bifidobacterium breve",
                        "Bifidobacterium catenulatum subsp. catenulatum",
                        "Bifidobacterium catenulatum subsp. kashiwanohense",
                        "Bifidobacterium catenulatum subsp. kashiwanohense_A",
                        "Bifidobacterium dentium",
                        "Bifidobacterium gallicum",
                        "Bifidobacterium longum subsp. infantis",
                        "Bifidobacterium longum subsp. longum",
                        "Bifidobacterium longum subsp. nov.",
                        "Bifidobacterium longum subsp. suis",
                        "Bifidobacterium pseudocatenulatum",
                        "Bifidobacterium pseudolongum subsp. globosum",
                        "Bifidobacterium scardovii",
                        "Bifidobacterium sp002742445",
                        "Bifidobacterium thermophilum")
# values (color codes)
genomes_263_colors <- c("tomato2", # Bifidobacterium adolescentis
                        "#b2964b", # Bifidobacterium angulatum
                        "black", # Bifidobacterium animalis subsp. lactis
                        "#c5c2f0", # Bifidobacterium bifidum
                        "#00a2ff", # Bifidobacterium breve
                        "#ffff7f", # Bifidobacterium catenulatum subsp. catenulatum
                        "#ffff7f", # Bifidobacterium catenulatum subsp. kashiwanohense
                        "#ffff7f", # Bifidobacterium catenulatum subsp. kashiwanohense_A
                        "#8e063c", # Bifidobacterium dentium
                        "#ffffff", # Bifidobacterium gallicum
                        "#81FF74", # Bifidobacterium longum subsp. infantis
                        "#51796f", # Bifidobacterium longum subsp. longum
                        "#c6dec7", # Bifidobacterium longum subsp. nov.
                        "#00b400", # Bifidobacterium longum subsp. suis
                        "#ffa600", # Bifidobacterium pseudocatenulatum
                        "#2032ab", # Bifidobacterium pseudolongum subsp. globosum
                        "#f79cd4", # Bifidobacterium scardovii
                        "#ffff7f", # Bifidobacterium sp002742445
                        "#ffffff") # Bifidobacterium thermophilum)
# labels (what will appear in the legend)
genomes_263_species <- c("B. adolescentis",
                         "B. angulatum",
                         "B. animalis ssp. lactis",
                         "B. bifidum",
                         "B. breve",
                         "B. catenulatum ssp. catenulatum",
                         "B. catenulatum ssp. kashiwanohense",
                         "B. catenulatum ssp. kashiwanohense_A",
                         "B. dentium",
                         "B. gallicum",
                         "B. longum ssp. infantis",
                         "B. longum ssp. longum",
                         "B. longum ssp. nov.",
                         "B. longum ssp. suis",
                         "B. pseudocatenulatum",
                         "B. pseudolongum ssp. globosum",
                         "B. scardovii",
                         "B. sp002742445",
                         "B. thermophilum")
```

7.4 Ordination

Ordination techniques summarize the data in a reduced number of dimensions while accounting for as much of the variability in the original data set as possible. We used Principal Component Analysis (PCA) for ordination of a table containing the representation of GH/CE/PL (sub)families in 263 genomes.

```
# extract the matrix from the CAZyme tibble
cazy_subfam_mat <- as.matrix((cazy_subfam_df[, 2:108]))
# do PCA
cazy.pca.res <- prcomp(cazy_subfam_mat, scale.=F, retx=T)
# sdev^2 captures eigenvalues from the PCA result
cazy.pc.var <- cazy.pca.res$sdev^2
# calculate the percentage of the total variance explained by each PC
cazy.pc.per <- round(cazy.pc.var/sum(cazy.pc.var)*100, 1)
# extract PCA results to a tibble
cazy.pca.res.df <- as_tibble(cazy.pca.res$x)

# plot the PCA results
# create a vector with taxonomy (group)
cazy.species <- cazy_subfam_df$curated_taxonomy
# create a vector with genome names
cazy.genomes <- cazy_subfam_df$Organism
# select points (genomes) that will be labeled
cazy.genomes_short <- ifelse(cazy.genomes ==
                             "Bifidobacterium longum subsp. suis Bg131.S11_17.F6",
                             "Bg131.S11_17.F6",
                      ifelse(cazy.genomes ==
                             "Bifidobacterium catenulatum subsp. kashiwanohense Bg42221_1E1",
                             "Bg42221_1E1", ""))
# plot
ggplot(cazy.pca.res.df) +
  aes(x=PC1, y=PC2, fill=cazy.species, shape=cazy.species, stroke = 0.15) +
  geom_point(size=2) +
  scale_shape_manual(values=genomes_263_shapes) +
  guides(shape="none") +
  scale_fill_manual(name = "Taxonomy",
                    breaks=genomes_263_breaks,
                    values=genomes_263_colors,
                  labels=genomes_263_species) +
  guides(fill = guide_legend(override.aes=list(shape=genomes_263_shapes))) +
  # add text labels for selected genomes
  geom_text_repel(aes(label = cazy.genomes_short), size = 3, fontface=1,
                  color="black", min.segment.length = 0,
                  seed = 42, box.padding = 1, max.overlaps = 100) +
  labs(title= "PCA of GH/CE/PL (sub)families representation") +
  xlab(paste0("PC1 (",cazy.pc.per[1],"%",")")) +
  ylab(paste0("PC2 (",cazy.pc.per[2],"%",")")) +
  coord_fixed(1) +
  theme_bw() +
  theme(plot.title = element_text(face="bold"),
        axis.title = element_text(color = "black"),
        axis.text = element_text(color = "black"))
```

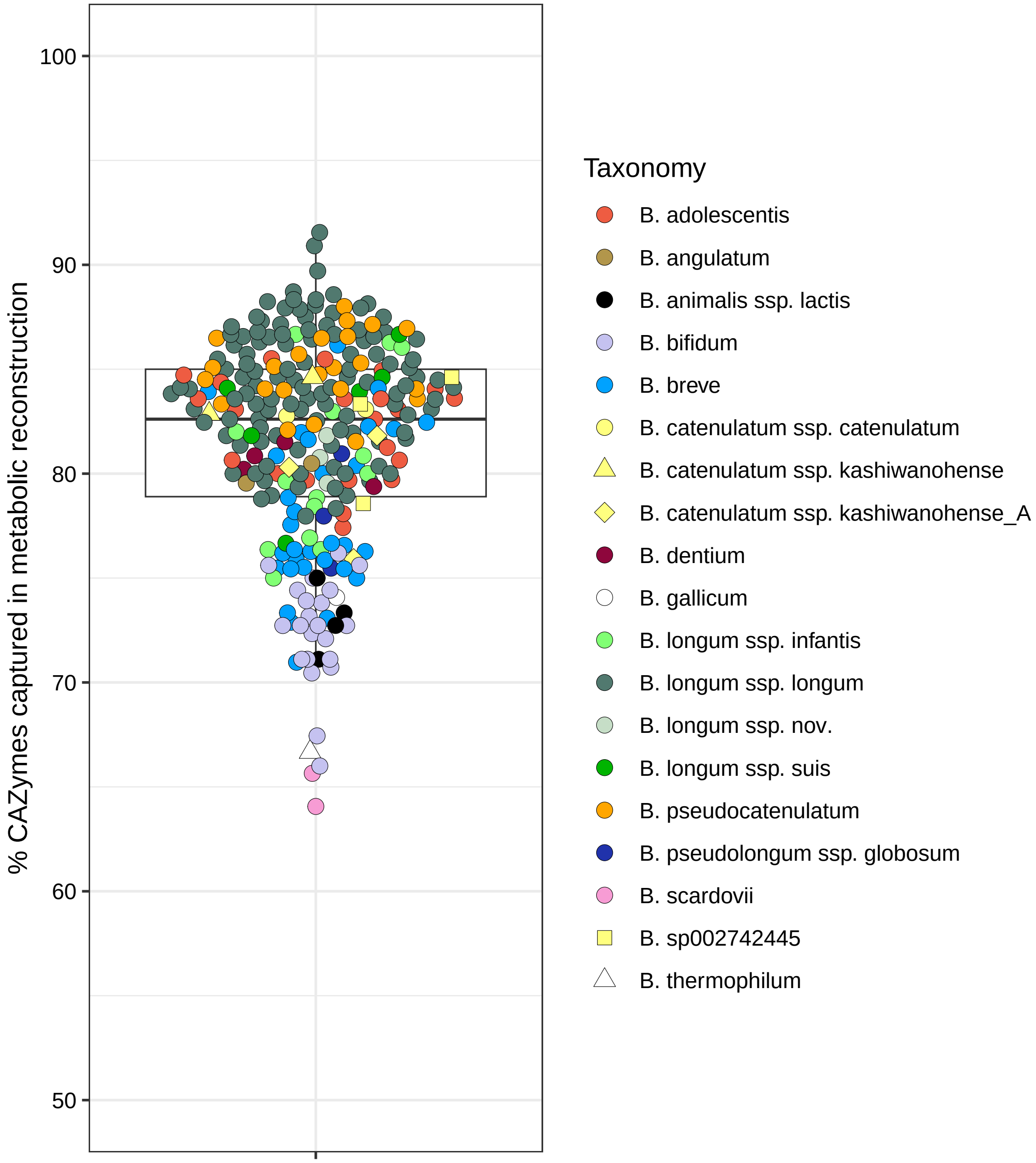


```
# save the plot to a file
ggsave("results/CAZyme/CAZyme_263_PCA.pdf", width = 7, height = 5)
```

7.5 Percent of CAZymes captured by metabolic reconstruction

For each genome, we calculated the percentage of CAZymes (GHs/CEs/PLs) captured by the metabolic reconstruction (i.e., ratio: number of GHs/CEs/PLs captured in mcSEED subsystems / total number of GHs/CEs/PLs identified by dbCAN).

```r
# calculate the ratio of CAZymes (GHs/CEs/PLs) captured in mcSEED subsystems
cazy_captured_mcseed <- cazy_df %>%
  group_by(Organism) %>%
  summarize(ratio = 100*(1 - (sum(Subsystem == "-") / n()))) %>%
  left_join(bpm_263_join, by = c("Organism" = "genome_name")) %>%
  arrange(genome_ID)

# plot a swarmplot + boxplot
# create a vector with curated taxonomy
cazy_species <- cazy_captured_mcseed$curated_taxonomy
# plot
ggplot() +
  geom_boxplot(data = cazy_captured_mcseed,
               mapping = aes(x="", y=ratio),
               outlier.shape = NA, width=0.9, lwd = 0.3) +
  ggbeeswarm::geom_quasirandom(data = cazy_captured_mcseed,
                               aes(x="", y=ratio, fill=cazy_species, shape=cazy_species),
                               color = "black", stroke = 0.15, size = 3) +
  scale_shape_manual(values=genomes_263_shapes) +
  guides(shape="none") +
  scale_fill_manual(name = "Taxonomy",
                    breaks=genomes_263_breaks,
                    values=genomes_263_colors,
                    labels=genomes_263_species) +
  guides(fill = guide_legend(override.aes=list(shape=genomes_263_shapes))) +
  theme_bw() +
  theme(plot.title = element_text(face="bold"),
        axis.title = element_text(color = "black"),
        axis.text = element_text(color = "black")) +
  coord_cartesian(ylim = c(50, 100)) +
  labs(x = "", y = "% CAZymes captured in metabolic reconstruction")
```



```
# save the plot to a file
ggsave("results/CAZyme/percent_captured_in_mcSEED.pdf", width = 6, height = 7)
```

The total and average (across all 263 genomes) percentages of captured CAZymes:

```
# calculate the total % of captured CAZymes
not_captured <- cazy_df %>%
  filter(Subsystem == "-") %>%
  nrow()
captured <- round(1 - not_captured / length(cazy_df$Subsystem), digits = 3) * 100
```

```
# calculate the average % of captured CAZymes
ratio_mean <- round(mean(cazy_captured_mcseed$ratio), digits = 1)
ratio_sd <- round(sd(cazy_captured_mcseed$ratio), digits = 1)
# print calculated values
print(paste0("Total percentage of captured CAZymes:"," ",captured, "%"))
```

```
## [1] "Total percentage of captured CAZymes: 81.8%"
```

```
print(paste0("Mean ± SD percentage of captured CAZymes:"," ",ratio_mean," ","±"," ",ratio_sd))
```

```
## [1] "Mean ± SD percentage of captured CAZymes: 81.5 ± 5"
```

8 Representation of predicted metabolic phenotypes in 26 *Bifidobacterium longum* genomes

8.1 Introduction

This block describes the analysis of the representation of various metabolic pathways in 26 *Bifidobacterium longum* genomes. This genomic dataset included 15 genomes from the reference set + 11 additional *Bifidobacterium longum* genomes. For comparative purposes, the 11 additional genomes included isolates of non-human origin, such as type strains of *Bifidobacterium longum* subsp. *suis* and *Bifidobacterium longum* subsp. *suillum*.

8.2 Load data

```
# read the table with BPM (carbohydrate utilization) for 26 B. longum genomes
bpm_26_Blon_df <- read_tsv("data/tables/BPM_26_Blongum_genomes_carbs.txt",
                           col_types = cols(.default = "c")) %>%
                            mutate_at(c(4:69), as.numeric)
# read the table with BPM (other pathways) for 26 B.longum genomes
bpm_26_Blon_df2 <- read_tsv("data/tables/BPM_26_Blongum_genomes_other.txt",
                            col_types = cols(.default = "c")) %>%
                            mutate_at(c(4:32), as.numeric)
# read the table with metadata for predicted carbohydrate utilization phenotypes
phenotype_metadata <- read_tsv("data/tables/phenotype_metadata_carbs.txt",
                               col_types = cols(.default = "c")) %>%
                               filter(!(phenotype %in% c("ManNAc", "GalNAc", "Man", "GalA")))
# read the table with metadata for other pathways
phenotype_metadata_other <- read_tsv("data/tables/phenotype_metadata_other.txt",
                               col_types = cols(.default = "c"))
```

8.3 Hierarchical clustering of the BPM (carbohydrate utilization) of 26 *Bifidobacterium longum* genomes

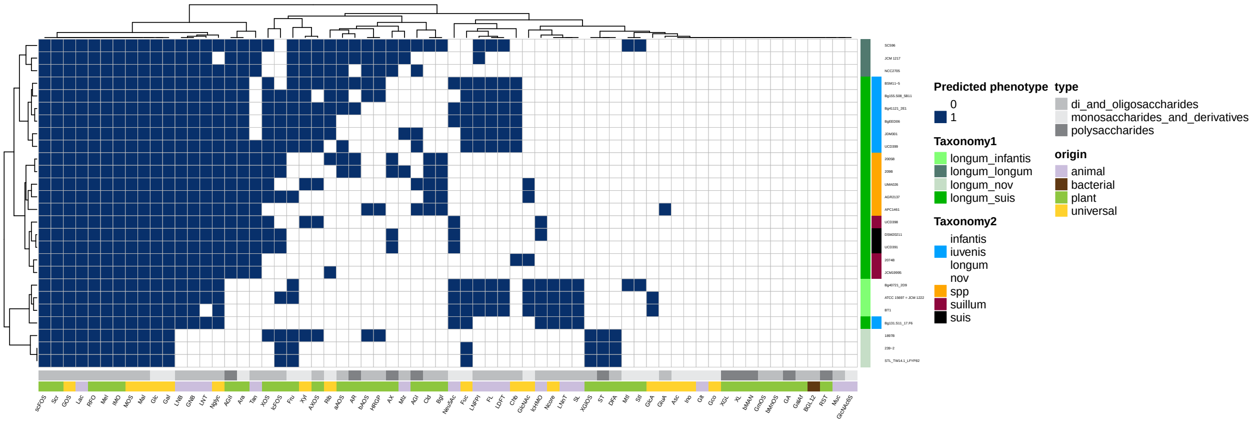
The following heatmap shows the hierarchical clustering of BPM with the representation of carbohydrate utilization phenotypes predicted for 26 *Bifidobacterium longum* strains.

Hierarchical clustering options:

- **Distance metric**: Hamming distance (equivalent to Manhattan distance for binary data)
- **Linkage method**: Average

```r
# extract the binary matrix
bpm_Blon_mat <- as.matrix((bpm_26_Blon_df[, 4:69]))
# add rownames to the matrix
genome_id_Blon <- bpm_26_Blon_df$genome_ID
rownames(bpm_Blon_mat) <- genome_id_Blon

# create vectors with taxonomy
Blon_tax1 <- bpm_26_Blon_df$group
Blon_tax2 <- bpm_26_Blon_df$add_tax
# extract vectors containing data about glycan type and origin
glycan_type <- phenotype_metadata$type_group
glycan_origin <- phenotype_metadata$origin

# create a coloring function
col_fun <- structure(c("white", "#08306b"), names = c("0", "1"))
# create two row annotations specifying taxonomy
ha_Blon1 <- HeatmapAnnotation(
  which = c("row"),
  Taxonomy1 = Blon_tax1,
  Taxonomy2 = Blon_tax2,
  col = list(Taxonomy1 = c("longum_infantis" = "#81FF74",
                           "longum_longum" = "#51796f",
                           "longum_suis" = "#00b400",
                           "longum_nov" = "#c6dec7"),
             Taxonomy2 = c("suis" = "black",
                           "spp" = "#ffa600",
                           "suillum" = "#8e063c",
                           "iuvenis" = "#00a2ff",
                           "longum" = "white",
                           "infantis" = "white",
                           "nov" = "white")),
             show_annotation_name = FALSE,
             simple_anno_size = unit(3, "mm"))
# create two column annotations specifying glycan type and origin
ha_Blon2 <- HeatmapAnnotation(
  type = glycan_type,
  origin = glycan_origin,
  col = list(type = c("monosaccharides_and_derivatives" = "#E6E7E8",
                      "di_and_oligosaccharides" = "#BCBEC0",
                      "polysaccharides" = "#808285"),
             origin = c("universal" = "#ffd22d",
                        "animal" = "#cbbedd",
                        "plant" = "#8dc63f",
                        "bacterial" = "#603913")),
             show_annotation_name = FALSE,
             simple_anno_size = unit(3, "mm"))

# plot the heatmap
```

```
pdf("results/phenotypes/Blon_carb_heatmap.pdf", width=15, height=10)
ht_Blon_carb <- ComplexHeatmap::Heatmap(bpm_Blon_mat,
                              name = "Predicted phenotype",
                              right_annotation = ha_Blon1,
                              bottom_annotation = ha_Blon2,
                              col = col_fun,
                              clustering_distance_rows = function(m)
                                dist(m, method = "manhattan"),
                              clustering_distance_columns = function(m)
                                dist(m, method = "manhattan"),
                              rect_gp = gpar(col = "grey", lwd = 0.05),
                              show_row_names = TRUE,
                              row_names_gp = gpar(fontsize = 3),
                              column_names_gp = gpar(fontsize = 5),
                              column_names_rot = 60,
                              width = unit(250, "mm"),
                              height = unit(100, "mm"))
draw(ht_Blon_carb)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ht_Blon_carb)
```



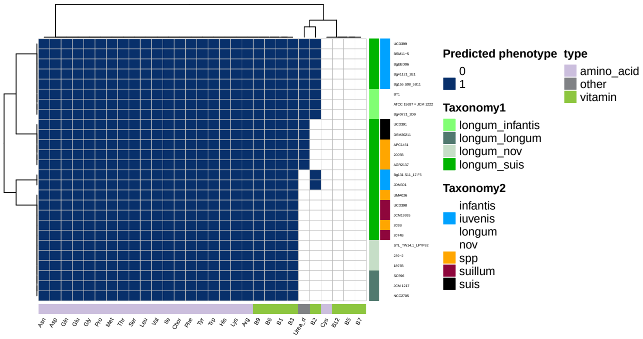
8.4 Hierarchical clustering of the BPM (other pathways) of 26 *Bifidobacterium longum* genomes

The following heatmap shows the hierarchical clustering of BPM with the representation of select metabolic pathways (amino acid/vitamin biosynthesis, urea utilization) predicted for 26 *Bifidobacterium longum* genomes.

Hierarchical clustering options:

- **Distance metric**: Hamming distance (equivalent to Manhattan distance for binary data)
- **Linkage method**: Average

```r
# extract the binary matrix
bpm_Blon_mat2 <- as.matrix((bpm_26_Blon_df2[, 4:32]))
# add rownames to the matrix
genome_id_Blon <- bpm_26_Blon_df2$genome_ID
rownames(bpm_Blon_mat2) <- genome_id_Blon

# create a vector with taxonomy (group)
Blon_tax1 <- bpm_26_Blon_df$group
Blon_tax2 <- bpm_26_Blon_df$add_tax
# extract vectors containing data about glycan type and origin
pathway_type <- phenotype_metadata_other$type
# create a coloring function
col_fun <- structure(c("white", "#08306b"), names = c("0", "1"))
# create two row annotations specifying taxonomy
ha_Blon1 <- HeatmapAnnotation(
  which = c("row"),
  Taxonomy1 = Blon_tax1,
  Taxonomy2 = Blon_tax2,
  col = list( Taxonomy1 = c("longum_infantis" = "#81FF74",
                            "longum_longum" = "#51796f",
                            "longum_suis" = "#00b400",
                            "longum_nov" = "#c6dec7"),
              Taxonomy2 = c("suis" = "black",
                            "spp" = "#ffa600",
                            "suillum" = "#8e063c",
                            "iuvenis" = "#00a2ff",
                            "longum" = "white",
                            "infantis" = "white",
                            "nov" = "white")),
              show_annotation_name = FALSE,
              simple_anno_size = unit(3, "mm"))
# create a column annotation specifying the pathway type
ha_Blon3 <- HeatmapAnnotation(
  type = pathway_type,
  col = list(type = c("vitamin" = "#8dc63f", "amino_acid" = "#cbbedd", "other" = "#808285")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))

# plot the heatmap
pdf("results/phenotypes/Blon_vit_heatmap.pdf", width=15, height=10)
ht_Blon_vit <- ComplexHeatmap::Heatmap(bpm_Blon_mat2,
                              name = "Predicted phenotype",
                              bottom_annotation = ha_Blon3,
                              right_annotation = ha_Blon1,
                              col = col_fun,
                              clustering_distance_rows = function(m)
                                dist(m, method = "manhattan"),
                              clustering_distance_columns = function(m)
                                dist(m, method = "manhattan"),
                              clustering_method_rows = "average",
                              clustering_method_columns = "average",
                              rect_gp = gpar(col = "grey", lwd = 0.05),
                              show_row_names = TRUE,
                              row_names_gp = gpar(fontsize = 3),
                              column_names_gp = gpar(fontsize = 5),
                              column_names_rot = 60,
                              width = unit(100, "mm"),
                              height = unit(80, "mm"))
draw(ht_Blon_vit)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ht_Blon_vit)
```



9 Representation of predicted metabolic phenotypes in 2967 *Bifidobacterium* genomes

9.1 Introduction

This section describes various analyses of two binary BPM for 2973 *Bifidobacterium* genomes. This genomic dataset was assembled by merging 263 reference and 2710 additional non-redundant genomes.

1. The first BPM depicts the presence/absence of carbohydrate utilization pathways
2. The second BPM depicts the presence/absence of biosynthetic pathways + urea utilization

9.2 Load data

```
# read BPM (carbs) for 263 reference genomes
bpm_263_df <- read_tsv("data/tables/BPM_263_NR_genomes_carbs.txt",
                            col_types = cols(.default = "c")) %>%
                            mutate_at(c(4:73), as.numeric) %>%
                            dplyr::select(-c(ManNAc, GalNAc, Man, GalA)) %>%
                            arrange(genome_ID)
# read BPM (carbs) for 2710 additional genomes
bpm_2710_df <- read_tsv("data/tables/BPM_2710_genomes_carbs.txt",
                        col_types = cols(.default = "c")) %>%
                        mutate_at(c(3:68), as.numeric) %>%
                        arrange(genome_ID)
# read the table with metadata for predicted carbohydrate utilization phenotypes
phenotype_metadata <- read_tsv("data/tables/phenotype_metadata_carbs.txt",
                      col_types = cols(.default = "c")) %>%
                      filter(!(phenotype %in% c("ManNAc", "GalNAc", "Man", "GalA")))
# extract BPM for 263 genomes for merging
bpm_263_carb_df <- bpm_263_df %>%
  dplyr::select(genome_ID, curated_taxonomy, c(Glc:Asc))
# extract BPM for 2710 genomes for merging
bpm_2710_carb_df <- bpm_2710_df %>%
  dplyr::select(genome_ID, curated_taxonomy, c(Glc:Asc))
# check that column names in both BPMs are identical
# colnames(bpm_263_carb_df) == colnames(bpm_2710_carb_df)
# merge the BPMs
bpm_2973_carb_df <- merge(bpm_263_carb_df, bpm_2710_carb_df, all=TRUE)

# read the table with BPM for 2973 genomes with predicted presence/absence of biosynthetic pathways
bpm_2973_other_df <- read_tsv("data/tables/BPM_2973_genomes_other.txt",
                              col_types = cols(.default = "c")) %>%
                     mutate_at(c(3:31), as.numeric) %>%
                     arrange(genome_ID)
# read the table with metadata for biosynthetic pathways
phenotype_metadata_other <- read_tsv("data/tables/phenotype_metadata_other.txt",
                               col_types = cols(.default = "c"))
```

9.3 Set colors

Defines point colors and shapes used throughout this block.

```
# arguments for scale_shape_manual
genomes_2973_shapes <- c(21,21,24,21,21,
                         21,21,24,23,21,
                         21,21,21,21,21,
                         21,21,21,22,21,
                         22,24,23)
# arguments for scale_fill_manual
```

```r
# breaks (how groups are encoded in the table)
genomes_2973_breaks <- c("Bifidobacterium adolescentis",
                         "Bifidobacterium angulatum",
                         "Bifidobacterium animalis subsp. animalis",
                         "Bifidobacterium animalis subsp. lactis",
                         "Bifidobacterium bifidum",
                         "Bifidobacterium breve",
                         "Bifidobacterium catenulatum subsp. catenulatum",
                         "Bifidobacterium catenulatum subsp. kashiwanohense",
                         "Bifidobacterium catenulatum subsp. kashiwanohense_A",
                         "Bifidobacterium dentium",
                         "Bifidobacterium gallicum",
                         "Bifidobacterium longum subsp. infantis",
                         "Bifidobacterium longum subsp. longum",
                         "Bifidobacterium longum subsp. nov.",
                         "Bifidobacterium longum subsp. suis",
                         "Bifidobacterium pseudocatenulatum",
                         "Bifidobacterium pseudolongum subsp. globosum",
                         "Bifidobacterium pullorum",
                         "Bifidobacterium ruminantium",
                         "Bifidobacterium scardovii",
                         "Bifidobacterium sp002742445",
                         "Bifidobacterium thermophilum",
                         "Bifidobacterium tsurumiense")
# values (color codes)
genomes_2973_colors <- c("tomato2", # Bifidobacterium adolescentis
                         "#b2964b", # Bifidobacterium angulatum
                         "black" , # Bifidobacterium animalis subsp. animalis
                         "black", # Bifidobacterium animalis subsp. lactis
                         "#c5c2f0", # Bifidobacterium bifidum
                         "#00a2ff", # Bifidobacterium breve
                         "#ffff7f", # Bifidobacterium catenulatum subsp. catenulatum
                         "#ffff7f", # Bifidobacterium catenulatum subsp. kashiwanohense
                         "#ffff7f", # Bifidobacterium catenulatum subsp. kashiwanohense_A
                         "#8e063c", # Bifidobacterium dentium
                         "#ffffff", # Bifidobacterium gallicum
                         "#81FF74", # Bifidobacterium longum subsp. infantis
                         "#51796f", # Bifidobacterium longum subsp. longum
                         "#c6dec7", # Bifidobacterium longum subsp. nov.
                         "#00b400", # Bifidobacterium longum subsp. suis
                         "#ffa600", # Bifidobacterium pseudocatenulatum
                         "#2032ab", # Bifidobacterium pseudolongum subsp. globosum
                         "#00FFF7", # Bifidobacterium pullorum
                         "#ffffff", # Bifidobacterium ruminantium
                         "#f79cd4", # Bifidobacterium scardovii
                         "#ffff7f", # Bifidobacterium sp002742445
                         "#ffffff", # Bifidobacterium thermophilum
                         "#ffffff") # Bifidobacterium tsurumiense
# labels (what will appear in the legend)
genomes_2973_species <- c("B. adolescentis",
                          "B. angulatum",
                          "B. animalis ssp. animalis",
                          "B. animalis ssp. lactis",
                          "B. bifidum",
                          "B. breve",
                          "B. catenulatum ssp. catenulatum",
                          "B. catenulatum ssp. kashiwanohense",
                          "B. catenulatum ssp. kashiwanohense_A",
                          "B. dentium",
                          "B. gallicum",
                          "B. longum ssp. infantis",
                          "B. longum ssp. longum",
                          "B. longum ssp. nov.",
                          "B. longum ssp. suis",
                          "B. pseudocatenulatum",
                          "B. pseudolongum ssp. globosum",
                          "B. pullorum",
                          "B. ruminantium",
                          "B. scardovii",
                          "B. sp002742445",
                          "B. thermophilum",
                          "B. tsurumiense")
```

9.4 Statistical analysis

The code chunks below describe the multivariate analysis of the BPM.

9.4.1 PERMANOVA

We used the Hamming distance (equivalent to Manhattan distance for binary data) to create a distance matrix from the BPM. We then performed a PERMANOVA (using the adonis2 function from the vegan) to assess the effect of taxonomic groupings on the constructed dissimilarity matrix.

```
# extract the binary matrix
bpm_2973_mat <- as.matrix((bpm_2973_carb_df[, 3:68]))
# calculate Hamming distance
bpm_2973_mat_dist <- vegdist(bpm_2973_mat, method ="manhattan")
# create a vector with taxonomy
taxonomy <- as.factor(bpm_2973_carb_df$curated_taxonomy)
# perform PERMANOVA
permanova_result <- adonis2(bpm_2973_mat_dist ~ taxonomy, permutations = 999, parallel = 8)
permanova_result
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bpm_2973_mat_dist ~ taxonomy, permutations = 999, parallel = 8)
##            Df SumOfSqs      R2    F Pr(>F)
## taxonomy   22   545430 0.89987 1205  0.001 ***
## Residual 2950    60693 0.10013
## Total    2972   606123 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The PERMANOVA results show a significant effect of taxonomy on the dissimilarity matrix (bpm_2973_mat_dist), explaining 89.99% of the variation ($R^2 = 0.89987$, $F = 1205$, $p = 0.001$), indicating that taxonomy significantly influences the differences observed in the data.

9.4.2 Homogeneity of multivariate dispersions

We assessed the homogeneity of multivariate dispersions using the betadisper function from the vegan package followed by a permutation test with permutest.

```
betadisper_result <- betadisper(bpm_2973_mat_dist, taxonomy)
permutest(betadisper_result)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##             Df  Sum Sq Mean Sq      F N.Perm Pr(>F)
## Groups      22  7688.7  349.49 94.422    999  0.001 ***
## Residuals 2950 10918.9    3.70
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test for homogeneity of multivariate dispersions shows a significant difference between groups ($F = 94.422$, $p = 0.001$), indicating that the variation in dispersion among taxonomic groups is not equal. This result, combined with the significant effect of taxonomy in the PERMANOVA analysis, suggests that the observed differences in the dissimilarity matrix are influenced by both the central tendency and dispersion of the groups.

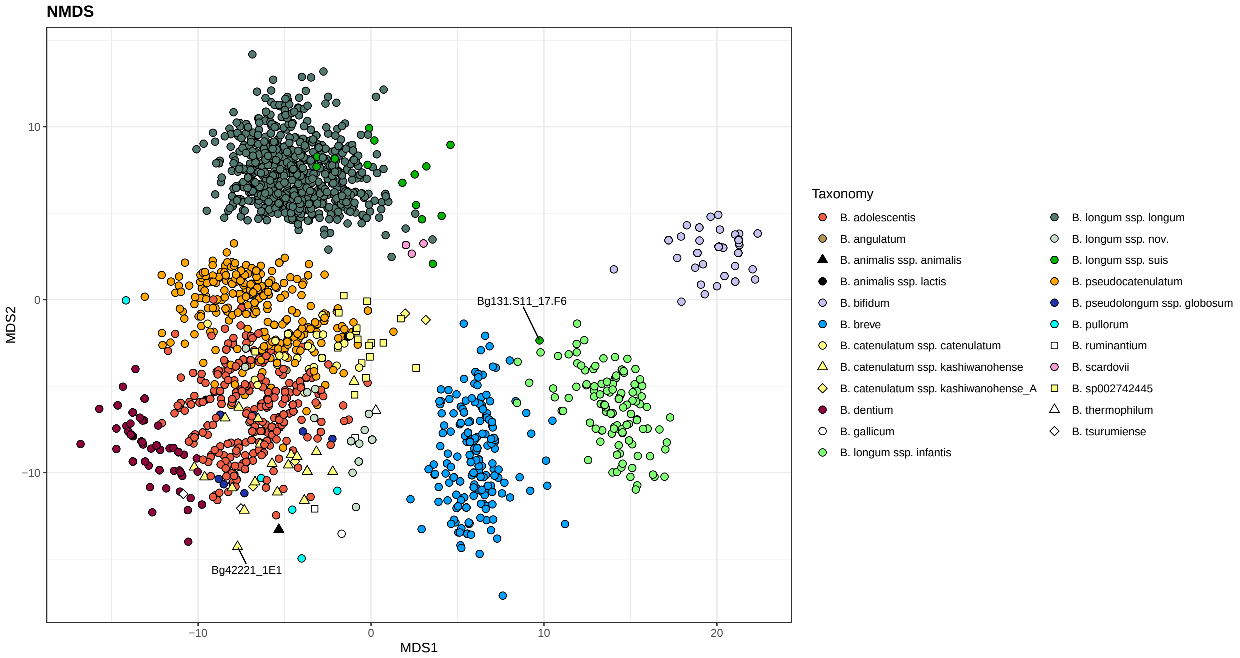
9.5 Ordination

Ordination techniques summarize the data in a reduced number of dimensions while accounting for as much of the variability in the original data set as possible. Here we use Non-metric MultiDimensional Scaling (NMDS) to visualize the level of similarity or dissimilarity between genomes based on a Hamming distance matrix.

```
# extract the binary matrix
bpm_2973_mat <- as.matrix((bpm_2973_carb_df[, 3:68]))
# do NMDS
nmds <- metaMDS(bpm_2973_mat,
             autotransform = FALSE,
             distance = "manhattan",
             engine = "monoMDS",
             k = 2,
             weakties = TRUE,
             model = "global",
             maxit = 300,
             try = 40,
             trymax = 100)
```

```
## Run 0 stress 0.1018357
## Run 1 stress 0.1018465
## ... Procrustes: rmse 0.0007692497  max resid 0.02521868
## Run 2 stress 0.1019231
## ... Procrustes: rmse 0.0007464143  max resid 0.0220277
## Run 3 stress 0.1018889
## ... Procrustes: rmse 0.0004315208  max resid 0.02184874
## Run 4 stress 0.1018397
## ... Procrustes: rmse 0.0004340168  max resid 0.02200739
## Run 5 stress 0.1022417
## ... Procrustes: rmse 0.0009023937  max resid 0.04395154
## Run 6 stress 0.1018643
## ... Procrustes: rmse 0.0009389056  max resid 0.02521889
## Run 7 stress 0.1022817
## ... Procrustes: rmse 0.001123376  max resid 0.04395164
## Run 8 stress 0.1019511
## ... Procrustes: rmse 0.0006598343  max resid 0.02521954
## Run 9 stress 0.1018515
## ... Procrustes: rmse 0.0005940397  max resid 0.02200846
## Run 10 stress 0.1018975
## ... Procrustes: rmse 0.0007182301  max resid 0.02202862
## Run 11 stress 0.1022469
## ... Procrustes: rmse 0.0009147552  max resid 0.043954
## Run 12 stress 0.1018384
## ... Procrustes: rmse 0.0006350611  max resid 0.025225
## Run 13 stress 0.1019158
## ... Procrustes: rmse 0.0009723215  max resid 0.02523245
## Run 14 stress 0.1018383
## ... Procrustes: rmse 0.0008984662  max resid 0.02522705
## Run 15 stress 0.1021991
## ... Procrustes: rmse 0.0009027652  max resid 0.04395509
## Run 16 stress 0.1018893
## ... Procrustes: rmse 0.0004614362  max resid 0.02185119
## Run 17 stress 0.1019126
## ... Procrustes: rmse 0.0009706455  max resid 0.02522992
## Run 18 stress 0.1019911
## ... Procrustes: rmse 0.0008768878  max resid 0.02523182
## Run 19 stress 0.1022378
## ... Procrustes: rmse 0.001025374  max resid 0.04395765
## Run 20 stress 0.1018439
## ... Procrustes: rmse 0.0001708423  max resid 0.008452249
## ... Similar to previous best
## Run 21 stress 0.1022398
## ... Procrustes: rmse 0.001026976  max resid 0.04397004
## Run 22 stress 0.1018911
## ... Procrustes: rmse 0.0007187335  max resid 0.0220374
## Run 23 stress 0.1018498
## ... Procrustes: rmse 0.0007543691  max resid 0.02524299
## Run 24 stress 0.1022852
## ... Procrustes: rmse 0.001024317  max resid 0.04398471
## Run 25 stress 0.1022041
## ... Procrustes: rmse 0.001003019  max resid 0.04397025
## Run 26 stress 0.1022074
## ... Procrustes: rmse 0.001029937  max resid 0.04396652
## Run 27 stress 0.102245
## ... Procrustes: rmse 0.001106648  max resid 0.04396681
## Run 28 stress 0.1019277
## ... Procrustes: rmse 0.0005987387  max resid 0.02193508
## Run 29 stress 0.1019652
## ... Procrustes: rmse 0.0008506045  max resid 0.02204635
## Run 30 stress 0.1018476
## ... Procrustes: rmse 0.0007724093  max resid 0.02522317
## Run 31 stress 0.1019599
## ... Procrustes: rmse 0.0007580452  max resid 0.02201325
## Run 32 stress 0.1019167
## ... Procrustes: rmse 0.0006277241  max resid 0.02522351
## Run 33 stress 0.1018628
## ... Procrustes: rmse 0.0007213101  max resid 0.02523264
## Run 34 stress 0.1018485
## ... Procrustes: rmse 0.0007742238  max resid 0.02523535
## Run 35 stress 0.1018951
## ... Procrustes: rmse 0.0006165552  max resid 0.02201999
## Run 36 stress 0.1019327
## ... Procrustes: rmse 0.0006103024  max resid 0.02193355
## Run 37 stress 0.1022415
## ... Procrustes: rmse 0.0009166491  max resid 0.04396071
## Run 38 stress 0.101893
## ... Procrustes: rmse 0.0004392224  max resid 0.02186677
## Run 39 stress 0.1018738
## ... Procrustes: rmse 0.0006678873  max resid 0.02521702
## Run 40 stress 0.1022471
## ... Procrustes: rmse 0.0009147389  max resid 0.04395287
## *** Best solution repeated 1 times
```

```r
# extract NMDS results to a tibble
nmds.df <- as_tibble(nmds$points)

# plot the NMDS results
# create a vector with taxonomy
taxonomy <- as.factor(bpm_2973_carb_df$curated_taxonomy)
# create a vector with genome names
IDs <- bpm_2973_carb_df$genome_ID
# select genomes that will be marked by text
genomes_short <- ifelse(IDs == "1695.38", "Bg131.S11_17.F6",
                        ifelse(IDs == "630129.38", "Bg42221_1E1", ""))
# plot
ggplot(nmds.df) +
  aes(x=MDS1, y=MDS2, fill=taxonomy, shape = taxonomy) +
  geom_point(size=2.5) +
  scale_shape_manual(values=genomes_2973_shapes) +
  guides(shape="none") +
  scale_fill_manual(name = "Taxonomy",
                    breaks=genomes_2973_breaks,
                    values=genomes_2973_colors,
                    labels=genomes_2973_species) +
  guides(fill = guide_legend(override.aes=list(shape=genomes_2973_shapes))) +
  geom_text_repel(aes(label = genomes_short), size = 3, fontface=1, color="black",
                  min.segment.length = 0, seed = 20, box.padding = 1,
                  max.overlaps = 100) +
  labs(title= "NMDS") +
  coord_fixed(1) +
  theme_bw() +
  theme(plot.title = element_text(face="bold"))
```



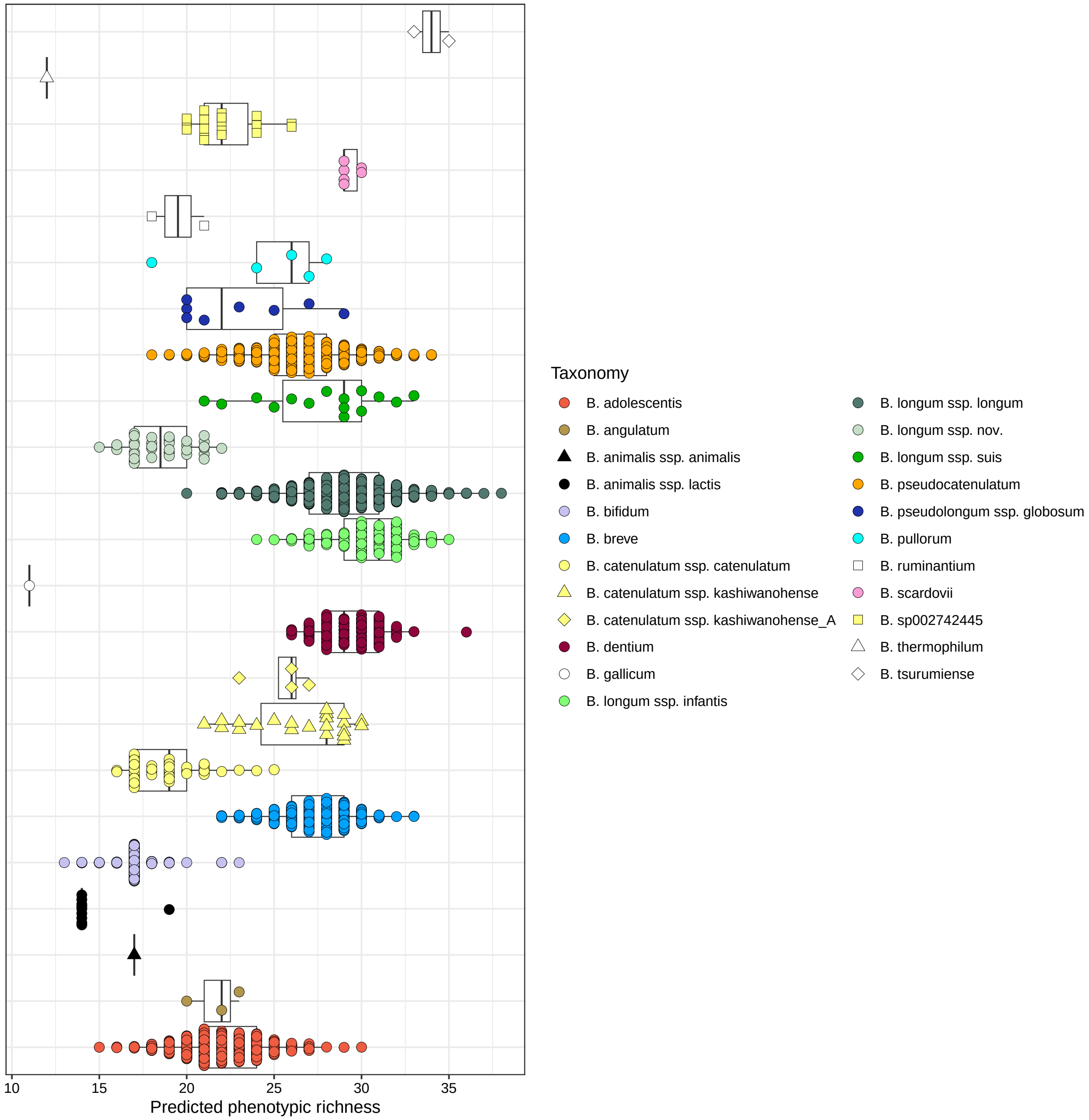
```
# save the file
ggsave("results/phenotypes/dim_reduction_testing/BPM_2973_Hamming_NMDS_weak_true.pdf",
       width = 14, height = 10)
```

The stress value of 0.1 indicated a good fit of the 2-dimensional model to the data, with stress type 1 and weak ties applied.

9.6 Predicted phenotypic richness

Here, we calculate predicted phenotypic richness, i.e., the total number of different predicted binary phenotypes “1” (utilizer) for each strain.

```
# create a new tibble
phenotype_richness <- bpm_2973_carb_df %>%
  mutate(curated_taxonomy = factor(curated_taxonomy, levels = genomes_2973_breaks)) %>%
  group_by(genome_ID, curated_taxonomy) %>%
  summarize(total_phenotypes = sum(across(Glc:Asc)))

# plot boxplot + swarmplot for each species/subspecies
ggplot() +
  geom_boxplot(data = phenotype_richness,
               mapping = aes(y = curated_taxonomy, x = total_phenotypes),
               outlier.shape = NA, width=0.9, lwd = 0.3) +
  ggbeeswarm::geom_quasirandom(data = phenotype_richness,
                               aes(y=curated_taxonomy, x=total_phenotypes,
                                   fill=curated_taxonomy,
                                   shape=curated_taxonomy),
                               color = "black", stroke = 0.1, size = 3) +
  scale_shape_manual(values=genomes_2973_shapes) +
  guides(shape="none") +
  scale_fill_manual(name = "Taxonomy",
                    breaks=c(genomes_2973_breaks),
                    values=c(genomes_2973_colors),
                    labels=c(genomes_2973_species)) +
  guides(fill = guide_legend(override.aes=list(shape=genomes_2973_shapes))) +
  theme_bw() +
  theme(plot.title = element_text(face="bold"),
        axis.title = element_text(color = "black"),
        axis.text = element_text(color = "black"),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank()) +
  labs(y = "", x = "Predicted phenotypic richness") +
  scale_x_continuous(breaks = seq(0, max(phenotype_richness$total_phenotypes), by = 5))
```



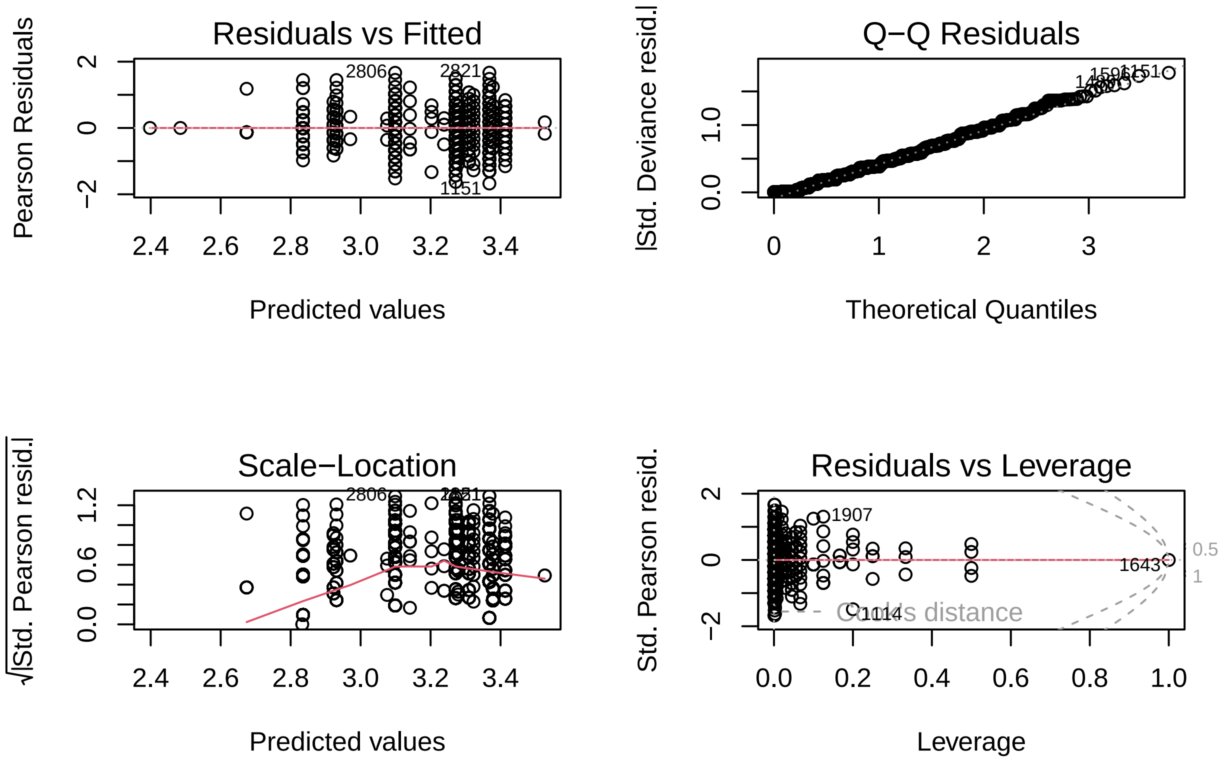
```
# save the plot to a file
ggsave("results/phenotypes/BPM_2973_phenotypic_richness.pdf", width = 10, height = 10)
```

9.6.1 Statistics for predicted phenotypic richness: generalized linear model (GLM)

Here we compare predicted phenotypic richness of diffrent *Bifidobacterum* clades using a generalized linear model (GLM) that assumes Poisson distribution. The phenotypic richness of *B. longum* subsp. *nov.* serves as a reference group.

```
# build a glm model
phenotype_richness$curated_taxonomy <- as.factor(phenotype_richness$curated_taxonomy)
phenotype_richness$curated_taxonomy <- relevel(phenotype_richness$curated_taxonomy,
                                               ref = "Bifidobacterium longum subsp. nov.")
```

```r
glm_result <- glm(total_phenotypes ~ curated_taxonomy, data = phenotype_richness, family = poisson)
# check the model
par(mfrow = c(2, 2))
plot(glm_result)
```



1. The residuals are randomly scattered around zero (but slightly shifted to the right)
2. The residuals are normally distributed
3. No extreme heteroscedasticity
4. One strong outlier based on Cook's distance (but kept for the analysis)

Overall, the Poisson regression model seems good; however, a check for overdispersion is required:

```r
dispersion_test <- dispersiontest(glm_result)
print(dispersion_test)
```

```
##
##  Overdispersion test
##
## data:  glm_result
## z = -130.74, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.1963478
```

The Poisson regression model does not exhibit overdispersion. In fact, it shows underdispersion with a dispersion parameter significantly less than 1. This means that the Poisson model is appropriate for our data, and we do not need to consider alternative models like the quasi-Poisson or negative binomial models to account for overdispersion. Multicollinearity is not an issue here, so we conclude that the model is appropriate for our data.

```r
summary(glm_result)
```

```
##
## Call:
## glm(formula = total_phenotypes ~ curated_taxonomy, family = poisson,
##     data = phenotype_richness)
##
## Coefficients:
##                                                                       Estimate
## (Intercept)                                                           2.922529
## curated_taxonomyBifidobacterium adolescentis                          0.176196
## curated_taxonomyBifidobacterium angulatum                             0.153246
## curated_taxonomyBifidobacterium animalis subsp. animalis             -0.089316
## curated_taxonomyBifidobacterium animalis subsp. lactis               -0.248380
## curated_taxonomyBifidobacterium bifidum                              -0.087145
## curated_taxonomyBifidobacterium breve                                 0.387197
## curated_taxonomyBifidobacterium catenulatum subsp. catenulatum        0.008403
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense     0.356330
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense_A  0.316150
## curated_taxonomyBifidobacterium dentium                               0.455445
## curated_taxonomyBifidobacterium gallicum                             -0.524634
## curated_taxonomyBifidobacterium longum subsp. infantis                0.490779
## curated_taxonomyBifidobacterium longum subsp. longum                  0.445559
## curated_taxonomyBifidobacterium longum subsp. suis                    0.400106
## curated_taxonomyBifidobacterium pseudocatenulatum                     0.349217
## curated_taxonomyBifidobacterium pseudolongum subsp. globosum          0.218385
## curated_taxonomyBifidobacterium pullorum                              0.280218
## curated_taxonomyBifidobacterium ruminantium                           0.047886
## curated_taxonomyBifidobacterium scardovii                             0.456196
## curated_taxonomyBifidobacterium sp002742445                           0.176061
## curated_taxonomyBifidobacterium thermophilum                         -0.437622
## curated_taxonomyBifidobacterium tsurumiense                           0.603832
##                                                                      Std. Error
## (Intercept)                                                            0.039778
## curated_taxonomyBifidobacterium adolescentis                           0.040894
## curated_taxonomyBifidobacterium angulatum                              0.130257
## curated_taxonomyBifidobacterium animalis subsp. animalis               0.245776
## curated_taxonomyBifidobacterium animalis subsp. lactis                 0.092081
## curated_taxonomyBifidobacterium bifidum                                0.041555
## curated_taxonomyBifidobacterium breve                                  0.041352
## curated_taxonomyBifidobacterium catenulatum subsp. catenulatum         0.051267
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense      0.057399
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense_A   0.106706
## curated_taxonomyBifidobacterium dentium                                0.043636
## curated_taxonomyBifidobacterium gallicum                               0.304124
## curated_taxonomyBifidobacterium longum subsp. infantis                 0.042537
## curated_taxonomyBifidobacterium longum subsp. longum                   0.040228
## curated_taxonomyBifidobacterium longum subsp. suis                     0.063136
## curated_taxonomyBifidobacterium pseudocatenulatum                     0.040985
## curated_taxonomyBifidobacterium pseudolongum subsp. globosum          0.083592
## curated_taxonomyBifidobacterium pullorum                              0.098551
## curated_taxonomyBifidobacterium ruminantium                           0.164995
## curated_taxonomyBifidobacterium scardovii                             0.085230
## curated_taxonomyBifidobacterium sp002742445                           0.063942
## curated_taxonomyBifidobacterium thermophilum                          0.291403
## curated_taxonomyBifidobacterium tsurumiense                           0.127625
##                                                                     z value
## (Intercept)                                                          73.471
## curated_taxonomyBifidobacterium adolescentis                          4.309
## curated_taxonomyBifidobacterium angulatum                             1.176
## curated_taxonomyBifidobacterium animalis subsp. animalis             -0.363
## curated_taxonomyBifidobacterium animalis subsp. lactis               -2.697
## curated_taxonomyBifidobacterium bifidum                              -2.097
## curated_taxonomyBifidobacterium breve                                 9.363
## curated_taxonomyBifidobacterium catenulatum subsp. catenulatum        0.164
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense     6.208
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense_A   2.963
## curated_taxonomyBifidobacterium dentium                              10.437
## curated_taxonomyBifidobacterium gallicum                             -1.725
## curated_taxonomyBifidobacterium longum subsp. infantis               11.538
## curated_taxonomyBifidobacterium longum subsp. longum                 11.076
## curated_taxonomyBifidobacterium longum subsp. suis                    6.337
## curated_taxonomyBifidobacterium pseudocatenulatum                     8.521
## curated_taxonomyBifidobacterium pseudolongum subsp. globosum          2.613
## curated_taxonomyBifidobacterium pullorum                              2.843
## curated_taxonomyBifidobacterium ruminantium                           0.290
## curated_taxonomyBifidobacterium scardovii                             5.353
## curated_taxonomyBifidobacterium sp002742445                           2.753
## curated_taxonomyBifidobacterium thermophilum                         -1.502
## curated_taxonomyBifidobacterium tsurumiense                           4.731
##                                                                     Pr(>|z|)
## (Intercept)                                                          < 2e-16
## curated_taxonomyBifidobacterium adolescentis                        1.64e-05
## curated_taxonomyBifidobacterium angulatum                            0.23940
## curated_taxonomyBifidobacterium animalis subsp. animalis             0.71630
## curated_taxonomyBifidobacterium animalis subsp. lactis               0.00699
## curated_taxonomyBifidobacterium bifidum                              0.03599
## curated_taxonomyBifidobacterium breve                                < 2e-16
## curated_taxonomyBifidobacterium catenulatum subsp. catenulatum       0.86980
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense   5.37e-10
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense_A  0.00305
## curated_taxonomyBifidobacterium dentium                              < 2e-16
## curated_taxonomyBifidobacterium gallicum                             0.08452
## curated_taxonomyBifidobacterium longum subsp. infantis               < 2e-16
## curated_taxonomyBifidobacterium longum subsp. longum                 < 2e-16
## curated_taxonomyBifidobacterium longum subsp. suis                  2.34e-10
## curated_taxonomyBifidobacterium pseudocatenulatum                    < 2e-16
## curated_taxonomyBifidobacterium pseudolongum subsp. globosum         0.00899
## curated_taxonomyBifidobacterium pullorum                             0.00446
## curated_taxonomyBifidobacterium ruminantium                          0.77164
## curated_taxonomyBifidobacterium scardovii                           8.67e-08
## curated_taxonomyBifidobacterium sp002742445                          0.00590
## curated_taxonomyBifidobacterium thermophilum                         0.13315
## curated_taxonomyBifidobacterium tsurumiense                         2.23e-06
##
## (Intercept)                                                         ***
## curated_taxonomyBifidobacterium adolescentis                        ***
## curated_taxonomyBifidobacterium angulatum
## curated_taxonomyBifidobacterium animalis subsp. animalis
## curated_taxonomyBifidobacterium animalis subsp. lactis              **
## curated_taxonomyBifidobacterium bifidum                             *
## curated_taxonomyBifidobacterium breve                               ***
## curated_taxonomyBifidobacterium catenulatum subsp. catenulatum
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense   ***
## curated_taxonomyBifidobacterium catenulatum subsp. kashiwanohense_A **
## curated_taxonomyBifidobacterium dentium                             ***
## curated_taxonomyBifidobacterium gallicum                            .
## curated_taxonomyBifidobacterium longum subsp. infantis              ***
## curated_taxonomyBifidobacterium longum subsp. longum                ***
## curated_taxonomyBifidobacterium longum subsp. suis                  ***
## curated_taxonomyBifidobacterium pseudocatenulatum                   ***
## curated_taxonomyBifidobacterium pseudolongum subsp. globosum        **
## curated_taxonomyBifidobacterium pullorum                            **
## curated_taxonomyBifidobacterium ruminantium
## curated_taxonomyBifidobacterium scardovii                           ***
## curated_taxonomyBifidobacterium sp002742445                         **
## curated_taxonomyBifidobacterium thermophilum
## curated_taxonomyBifidobacterium tsurumiense                         ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3061.53  on 2972  degrees of freedom
## Residual deviance:  586.84  on 2950  degrees of freedom
## AIC: 15663
##
## Number of Fisher Scoring iterations: 4
```

A Poisson regression model was fitted to evaluate the relationship between taxonomic groups and phenotypic richness. The overall model fit was assessed using a likelihood ratio test, comparing the full model to a null model containing only the intercept.

```r
# fit the null model (intercept-only model)
null_model <- glm(total_phenotypes ~ 1, data = phenotype_richness, family = poisson)
lr_test <- anova(null_model, glm_result, test = "Chisq")
print(lr_test)
```

```
## Analysis of Deviance Table
##
## Model 1: total_phenotypes ~ 1
## Model 2: total_phenotypes ~ curated_taxonomy
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      2972    3061.53
## 2      2950     586.84 22   2474.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The likelihood ratio test comparing the full model to the null model indicated a significant improvement in model fit when including the taxonomic groups as predictors. The full model showed a significant reduction in residual deviance (from 3061.53 to 586.84) with a chi-squared value of 2474.7 and 22 degrees of freedom, resulting in a p-value less than 2.2e-16.

Run post-hoc tests and save the result as a table:

```
posthoc_glm <- emmeans(glm_result, ~ curated_taxonomy)
pairwise_comparisons <- as_tibble(contrast(posthoc_glm, method = "pairwise", adjust = "bonferroni")) %>%
  write_tsv("results/phenotypes/phenotypic_richness_stats.txt")
#gt(pairwise_comparisons)
```

9.7 BPM collapsed at species/subspecies level (carbohydrate utilization)

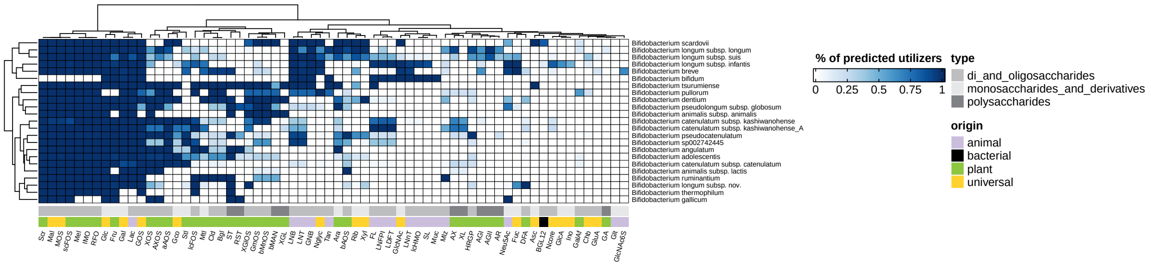
The following heatmap demonstrates percent of predicted utilizes (i.e., “average” phenotypes) at species/subspecies levels.

Hierarchical clustering options:

- **Distance metric**: Euclidean distance
- **Linkage method**: Ward’s D2

```
# create a tibble where phenotypes are averaged at species/subspecies levels
bpm_2973_mean <- bpm_2973_carb_df %>%
  group_by(curated_taxonomy) %>%
  summarise_at(vars(Glc:Asc), list(mean))
# extract the binary matrix
bpm_2973_mean_mat <- as.matrix(bpm_2973_mean[, 2:67])
# add rownames to the matrix
rownames(bpm_2973_mean_mat) <- bpm_2973_mean$curated_taxonomy

# extract vectors containing data about glycan type and origin
glycan_type <- phenotype_metadata$type_group
glycan_origin <- phenotype_metadata$origin
# create two column annotations specifying glycan type and origin
ha_2973_mean <- HeatmapAnnotation(
  type = glycan_type,
  origin = glycan_origin,
  col = list(type = c("monosaccharides_and_derivatives" = "#E6E7E8",
                      "di_and_oligosaccharides" = "#BCBEC0",
                      "polysaccharides" = "#808285"),
             origin = c("universal" = "#ffd22d",
                        "animal" = "#cbbedd",
                        "plant" = "#8dc63f",
                        "bacterial" = "black")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))
# add a coloring function
col_fun <- colorRamp2(c(0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1),
```

```r
                      c("#ffffff", "#DEEBF7", "#C6DBEF", "#9ECAE1", "#6BAED6",
                        "#4292C6", "#2171B5", "#08519C", "#08306B"))

# specify the name of the output file
pdf("results/phenotypes/BPM_2973_collapsed_heatmap.pdf", width=15, height=8)
# plot the heatmap
ht_2973 <- ComplexHeatmap::Heatmap(bpm_2973_mean_mat,
        col = col_fun,
        bottom_annotation = ha_2973_mean,
        clustering_distance_rows = function(m) dist(m, method = "euclidean"),
        clustering_distance_columns = function(m) dist(m, method = "euclidean"),
        clustering_method_rows = "ward.D2",
        clustering_method_columns = "ward.D2",
        column_names_side = "bottom",
        rect_gp = gpar(col = "black", lwd = 0.1),
        row_names_gp = gpar(fontsize = 6),
        column_names_gp = gpar(fontsize = 6),
        column_names_rot = 75,
        width = unit(180, "mm"),
        height = unit(50, "mm"),
        heatmap_legend_param = list(
          col_fun = col_fun,
          at = c(0, 0.25, 0.5, 0.75, 1),
          title = "% of predicted utilizers",
          direction = "horizontal",
          title_position = "topcenter",
          border = "black",
          legend_width = unit(40, "mm"))
        )
draw(ht_2973)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ht_2973)
```



Number of predicted phenotypes with variability within groups.

```r
# count columns with only 0s and 1s
binary_columns <- sum(apply(bpm_2973_mean_mat, 2, function(col) all(col %in% c(0, 1))))

# count columns with intermediate values in the [0,1] range
range_columns <- sum(apply(bpm_2973_mean_mat, 2, function(col) any(col > 0 & col < 1)))

# Print the counts
cat("Columns with only 0s and 1s:", binary_columns, "\n")
```

```
## Columns with only 0s and 1s: 2
```

```r
cat("Columns with intermediate values in [0,1] range:", range_columns, "\n")
```

```
## Columns with intermediate values in [0,1] range: 64
```

9.8 BPM collapsed at species/subspecies level (other phenotypes)

The following heatmap demonstrates percent of predicted “average” phenotypes at species/subspecies levels:

- Biosynthesis of B vitamins
- Biosynthesis of amino acids
- Urea utilization

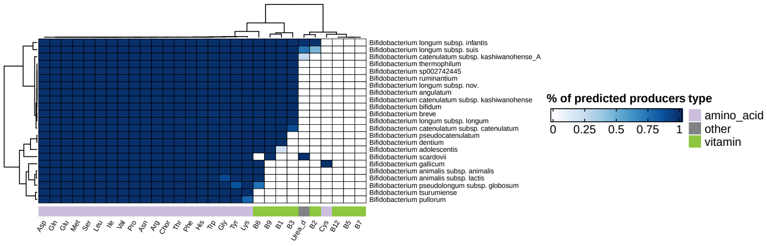
Hierarchical clustering options:

- **Distance metric**: Euclidean distance

- **Linkage method**: Ward's D2

```r
# create a tibble where phenotypes are averaged at species/subspecies levels
bpm_2973_mean2 <- bpm_2973_other_df %>%
  group_by(curated_taxonomy) %>%
  summarise_at(vars(B1:Urea_d), list(mean))
# extract the binary matrix
bpm_2973_mean2_mat <- as.matrix(bpm_2973_mean2[, 2:30])
# add rownames to the matrix
rownames(bpm_2973_mean2_mat) <- bpm_2973_mean2$curated_taxonomy

# extract vectors containing data about pathway type
pathway_type <- phenotype_metadata_other$type
# create a column annotations specifying pathway type
ha_2973_2_mean <- HeatmapAnnotation(
  type = pathway_type,
  col = list(type = c("vitamin" = "#8dc63f", "amino_acid" = "#cbbedd", "other" = "#808285")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))
# add a coloring function
col_fun <- colorRamp2(c(0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1),
                      c("#ffffff", "#DEEBF7", "#C6DBEF", "#9ECAE1", "#6BAED6",
                        "#4292C6", "#2171B5", "#08519C", "#08306B"))

# specify the name of the output file
pdf("results/phenotypes/BPM_2973_collapsed_heatmap_other.pdf", width=15, height=8)
# plot the heatmap
ht_2973_other <- ComplexHeatmap::Heatmap(bpm_2973_mean2_mat,
        col = col_fun,
        bottom_annotation = ha_2973_2_mean,
        clustering_distance_rows = function(m) dist(m, method = "euclidean"),
        clustering_distance_columns = function(m) dist(m, method = "euclidean"),
        clustering_method_rows = "ward.D2",
        clustering_method_columns = "ward.D2",
        column_names_side = "bottom",
        rect_gp = gpar(col = "black", lwd = 0.1),
        row_names_gp = gpar(fontsize = 6),
        column_names_gp = gpar(fontsize = 6),
        column_names_rot = 60,
        width = unit(100, "mm"),
        height = unit(50, "mm"),
        heatmap_legend_param = list(
          col_fun = col_fun,
          at = c(0, 0.25, 0.5, 0.75, 1),
          title = "% of predicted producers",
          direction = "horizontal",
          title_position = "topcenter",
          border = "black",
          legend_width = unit(40, "mm"))
        )
draw(ht_2973_other)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
```

```
draw(ht_2973_other)
```



9.9 Hierarchical clustering of the BPM (carbohydrate utilization) of 95 *Bifidobacterium catenulatum*-like genomes

The following heatmap shows the hierarchical clustering of BPM with the representation of carbohydrate utilization phenotypes predicted in 95 *Bifidobacterium catenulatum*-like genomes.

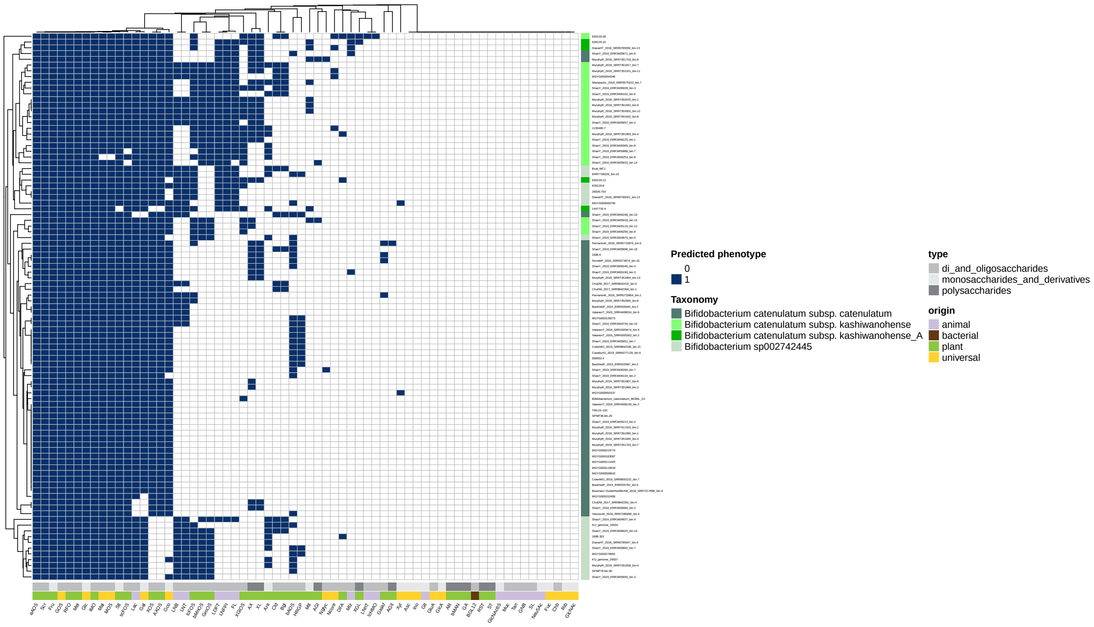
Hierarchical clustering options:

- **Distance metric**: Hamming distance (equivalent to Manhattan distance for binary data)
- **Linkage method**: Average

```
# create a tibble where phenotypes are averaged at species/subspecies levels
# extract BPM for 263 genomes for merging
bpm_cat_carb_df <- bpm_2973_carb_df %>%
  filter(curated_taxonomy %in% c("Bifidobacterium catenulatum subsp. catenulatum",
                                 "Bifidobacterium catenulatum subsp. kashiwanohense",
                                 "Bifidobacterium catenulatum subsp. kashiwanohense_A",
                                 "Bifidobacterium sp002742445")) %>%
  dplyr::select(genome_ID, curated_taxonomy, c(Glc:Asc))

# extract the binary matrix
bpm_cat_mat <- as.matrix((bpm_cat_carb_df[, 3:68]))
# add rownames to the matrix
genome_id_cat <- bpm_cat_carb_df$genome_ID
rownames(bpm_cat_mat) <- genome_id_cat

# create a vector with taxonomy (group)
```

```r
subsp_cat <- bpm_cat_carb_df$curated_taxonomy
# extract vectors containing data about glycan type and origin
phenotype_metadata <- phenotype_metadata %>%
  filter(!(phenotype %in% c("ManNAc", "GalNAc", "Man", "GalA")))
glycan_type <- phenotype_metadata$type_group
glycan_origin <- phenotype_metadata$origin
# create a coloring function
col_fun <- structure(c("white", "#08306b"), names = c("0", "1"))
# create a row annotation specifying taxonomy
ha_cat1 <- HeatmapAnnotation(
  which = c("row"),
  Taxonomy = subsp_cat,
  col = list(Taxonomy = c("Bifidobacterium catenulatum subsp. catenulatum" = "#51796f",
                          "Bifidobacterium catenulatum subsp. kashiwanohense" = "#81FF74",
                          "Bifidobacterium catenulatum subsp. kashiwanohense_A" = "#00b400",
                          "Bifidobacterium sp002742445" = "#c6dec7")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))
# create two column annotations specifying glycan type and origin
ha_cat2 <- HeatmapAnnotation(
  type = glycan_type,
  origin = glycan_origin,
  col = list(type = c("monosaccharides_and_derivatives" = "#E6E7E8",
                      "di_and_oligosaccharides" = "#BCBEC0",
                      "polysaccharides" = "#808285"),
             origin = c("universal" = "#ffd22d",
                        "animal" = "#cbbedd",
                        "plant" = "#8dc63f",
                        "bacterial" = "#603913")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))

# plot the heatmap
pdf("results/phenotypes/Bcatenulatum_heatmap.pdf", width=18, height=10)
ht_bcatenulatum <- ComplexHeatmap::Heatmap(bpm_cat_mat,
                              name = "Predicted phenotype",
                              right_annotation = ha_cat1,
                              bottom_annotation = ha_cat2,
                              col = col_fun,
                              clustering_distance_rows = function(m)
                              dist(m, method = "binary"),
                              clustering_distance_columns = function(m)
                              dist(m, method = "binary"),
                              clustering_method_rows = "average",
                              clustering_method_columns = "average",
                              rect_gp = gpar(col = "grey", lwd = 0.05),
                              show_row_names = TRUE,
                              row_names_gp = gpar(fontsize = 3),
                              column_names_gp = gpar(fontsize = 5),
                              column_names_rot = 60,
                              width = unit(200, "mm"),
                              height = unit(200, "mm"))
draw(ht_bcatenulatum)
```

```r
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ht_bcatenulatum)
```



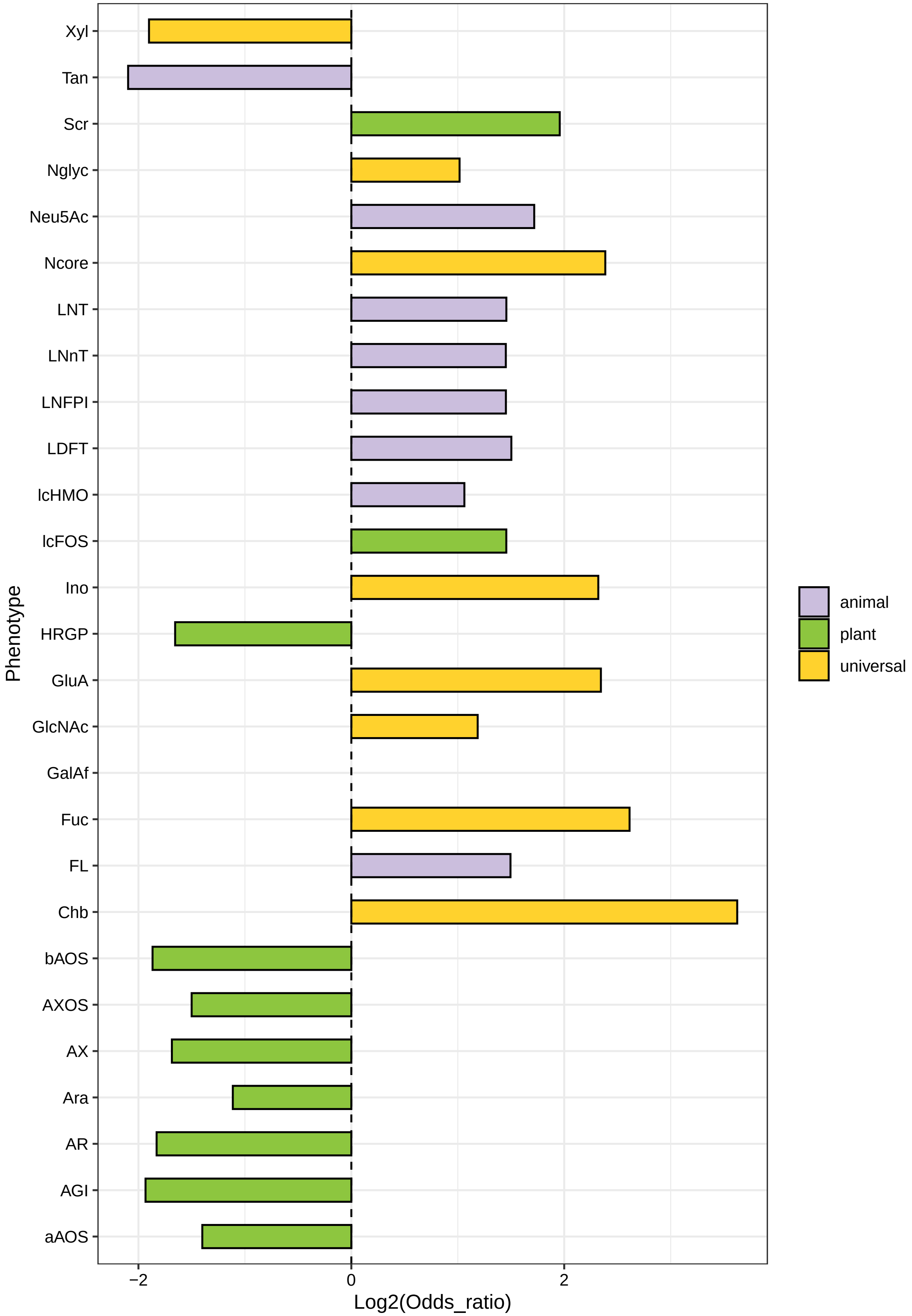
9.10 Pathway enrichment

Here we do a pathway enrichment analysis using Fisher's exact test. Pathways significantly enriched (P_{adj} 0.01) in specified groups are shown.

```r
# create a table for Fisher exact test
# extract data for 263 genomes
bpm_263_fish_df <- bpm_263_df %>%
  dplyr::select(genome_ID, curated_taxonomy, c(Glc:Asc), non_westernized, age_group)
# extract data for 2710 genomes
bpm_2710_fish_df <- bpm_2710_df %>%
  dplyr::select(genome_ID, curated_taxonomy, c(Glc:Asc), non_westernized, age_group)
# reorder columns in `bpm_263_fish_df` based on `bpm_2710_fish_df`
bpm_263_fish_df <- bpm_263_fish_df[, colnames(bpm_2710_fish_df)]
# merge the two tables
bpm_2973_fish_df <- merge(bpm_2710_fish_df, bpm_263_fish_df, all=TRUE)
```

9.10.1 Westernized (age < 3) vs. non-Westernized (age < 3)

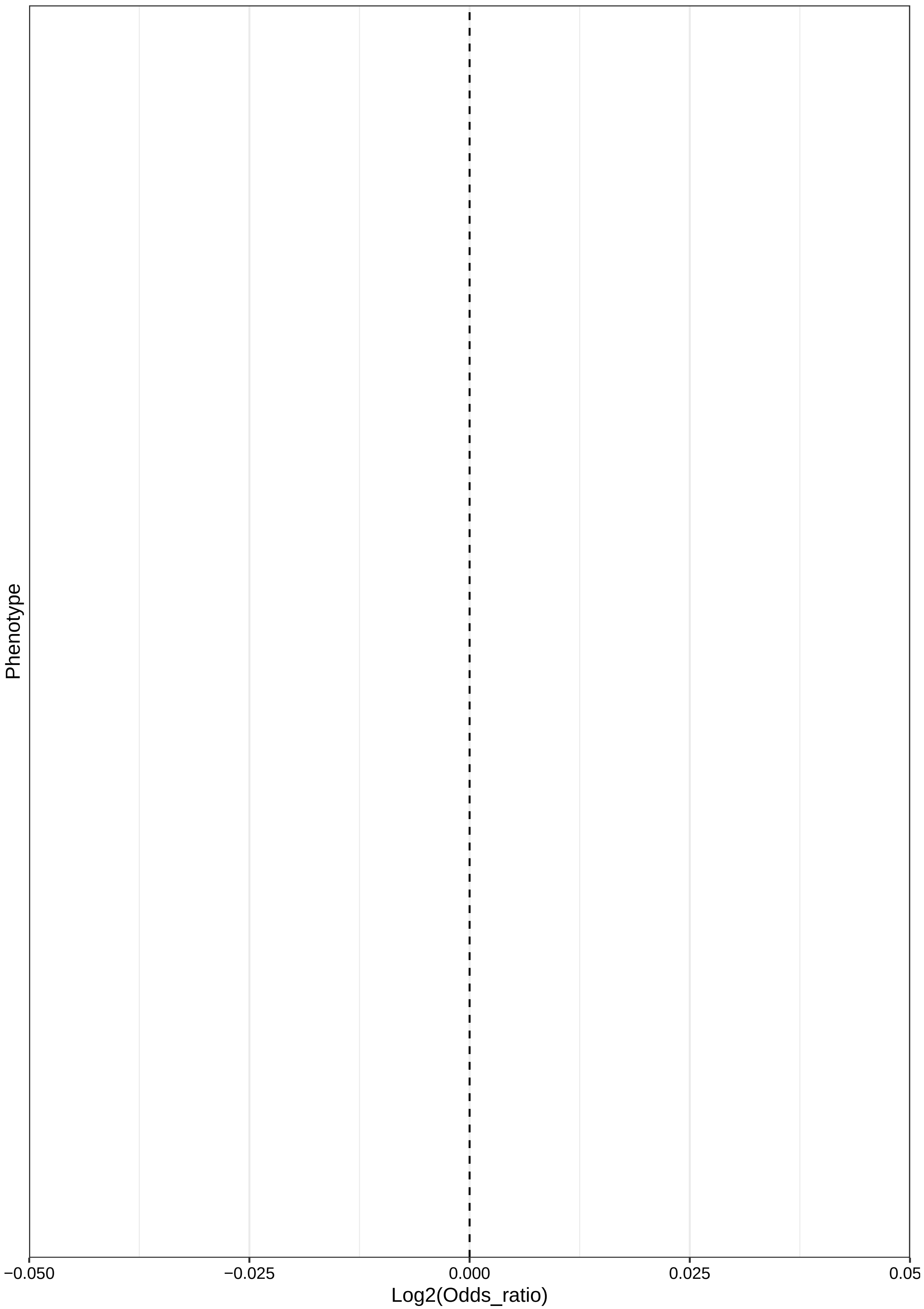
```r
bpm_west_inf_df <- bpm_2973_fish_df %>%
  filter(age_group == "child") %>%
  dplyr::select(genome_ID, c(Glc:Asc), non_westernized)
# create a contingency table
# i.e., calculate the number of occurrences of "yes" and "no" in the column `non_westernized` # for eac
contingency_table_west <- bpm_west_inf_df  %>%
  pivot_longer(cols = -c(genome_ID, non_westernized), names_to = "phenotype", values_to = "value") %>%
  group_by(phenotype, non_westernized, value) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = non_westernized, values_from = count) %>%
  replace(is.na(.), 0)
# keep only rows where neither all values are 0 nor all values are 1 within each group
contingency_table_west <- contingency_table_west %>%
  group_by(phenotype) %>%
  filter(!all(value == 0) & !all(value == 1))
# perform the Fisher's exact test, calculate odds ratios and adjusted p-values
result_west <- contingency_table_west %>%
  group_by(phenotype) %>%
  summarise(p_value = fisher.test(matrix(c(no, yes), nrow = 2))$p.value,
            odds_ratio = fisher.test(matrix(c(no, yes), nrow = 2))$estimate,
            conf_int_low = fisher.test(matrix(c(no, yes), nrow = 2))$conf.int[1],
            conf_int_high = fisher.test(matrix(c(no, yes), nrow = 2))$conf.int[2]) %>%
  mutate(p_adj = p.adjust(p_value, method = "fdr"))
# add information about phenotypes
results_west_ann <- left_join(result_west, phenotype_metadata, by = "phenotype")
# create a data frame with the log2-transformed odds ratios
odds_west_df <- results_west_ann %>%
  mutate(log2_odds_ratio = log2(odds_ratio)) %>%
  mutate(log2_conf_int_low = log2(conf_int_low)) %>%
  mutate(log2_conf_int_high = log2(conf_int_high)) %>%
  arrange(desc(log2_odds_ratio)) %>%
  dplyr::select(phenotype, origin, odds_ratio, conf_int_low, conf_int_high,
                log2_odds_ratio, p_value, p_adj)

# plot barplot
ggplot(data = subset(odds_west_df, p_adj <= 0.01)) +
  aes(x = log2_odds_ratio, y = phenotype, color= origin, fill = origin) +
  geom_bar(stat = "identity", width = 0.5, color = "black") +
  geom_vline(xintercept = 0, color = "black", linetype = "dashed") +
  scale_color_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  scale_fill_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  labs(x = "Log2(Odds_ratio)", y = "Phenotype") +
  theme_bw() +
  theme(legend.title = element_blank(),
        plot.title = element_text(face="bold"),
        axis.title = element_text(color = "black"),
        axis.text = element_text(color = "black"))
```


Xyl
Tan
Scr
Nglyc
Neu5Ac
Ncore
LNT
LNnT
LNFPI
LDFT
lcHMO
lcFOS
Ino
HRGP
GluA
GlcNAc
GalAf
Fuc
FL
Chb
bAOS
AXOS
AX
Ara
AR
AGI
aAOS
Phenotype
−2
0
2
Log2(Odds_ratio)
animal
plant
universal

```
# save the figure to a file
ggsave("results/phenotype_enrichment/child_westernized_vs_child_non_westernized.pdf", device = "pdf", wi
```

9.10.2 Westernized (age > 3) vs. non-Westernized (age >= 3)

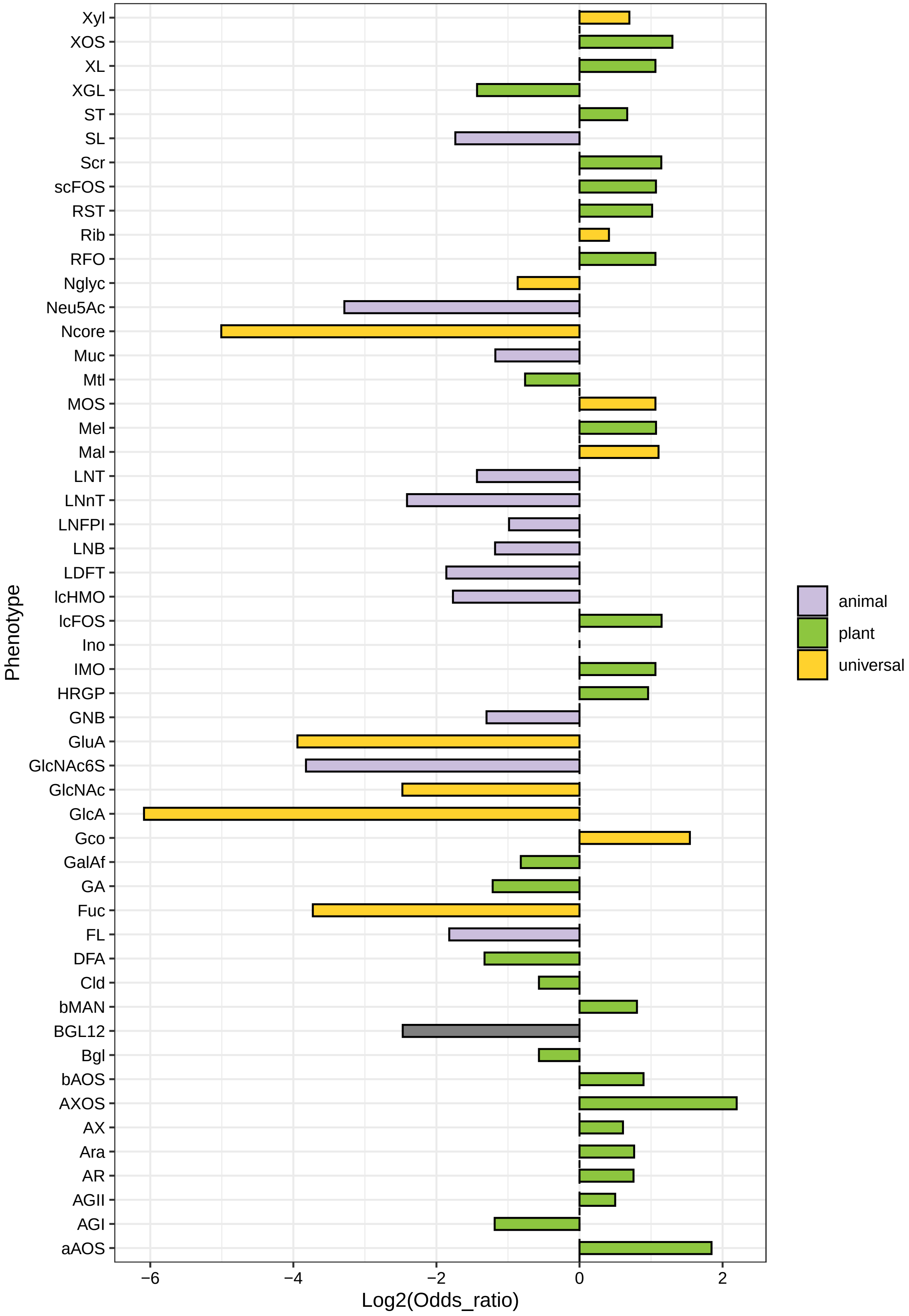
```
bpm_west_ad_df <- bpm_2973_fish_df %>%
  filter(age_group == "adult") %>%
  dplyr::select(genome_ID, c(Glc:Asc), non_westernized)
# create a contingency table
# i.e., calculate the number of occurrences of "yes" and "no" in the column `non_westernized` # for eac
contingency_table_west <- bpm_west_ad_df   %>%
  pivot_longer(cols = -c(genome_ID, non_westernized), names_to = "phenotype", values_to = "value") %>%
  group_by(phenotype, non_westernized, value) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = non_westernized, values_from = count) %>%
  replace(is.na(.), 0)
# keep only rows where neither all values are 0 nor all values are 1 within each group
contingency_table_west <- contingency_table_west %>%
  group_by(phenotype) %>%
  filter(!all(value == 0) & !all(value == 1))
# perform the Fisher's exact test, calculate odds ratios and adjusted p-values
result_west <- contingency_table_west %>%
  group_by(phenotype) %>%
  summarise(p_value = fisher.test(matrix(c(no, yes), nrow = 2))$p.value,
            odds_ratio = fisher.test(matrix(c(no, yes), nrow = 2))$estimate,
            conf_int_low = fisher.test(matrix(c(no, yes), nrow = 2))$conf.int[1],
            conf_int_high = fisher.test(matrix(c(no, yes), nrow = 2))$conf.int[2]) %>%
  mutate(p_adj = p.adjust(p_value, method = "fdr"))
# add information about phenotypes
results_west_ann <- left_join(result_west, phenotype_metadata, by = "phenotype")
# create a data frame with the log2-transformed odds ratios
odds_west_df <- results_west_ann %>%
  mutate(log2_odds_ratio = log2(odds_ratio)) %>%
  mutate(log2_conf_int_low = log2(conf_int_low)) %>%
  mutate(log2_conf_int_high = log2(conf_int_high)) %>%
  arrange(desc(log2_odds_ratio)) %>%
  dplyr::select(phenotype, origin, odds_ratio, conf_int_low, conf_int_high,
                log2_odds_ratio, p_value, p_adj)

# plot barplot
ggplot(data = subset(odds_west_df, p_adj <= 0.01)) +
  aes(x = log2_odds_ratio, y = phenotype, color= origin, fill = origin) +
  geom_bar(stat = "identity", width = 0.5, color = "black") +
  geom_vline(xintercept = 0, color = "black", linetype = "dashed") +
  scale_color_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  scale_fill_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  labs(x = "Log2(Odds_ratio)", y = "Phenotype") +
  theme_bw() +
  theme(legend.title = element_blank(),
        plot.title = element_text(face="bold"),
        axis.title = element_text(color = "black"),
        axis.text = element_text(color = "black"))
```


Phenotype
−0.050
−0.025
0.000
0.025
0.05
Log2(Odds_ratio)

```
# save the figure to a file
ggsave("results/phenotype_enrichment/adult_westernized_vs_adult_non_westernized.pdf",
       device = "pdf", width = 7, height = 10)
```

9.10.3 Westernized (age < 3) vs. Westernized (age >= 3)

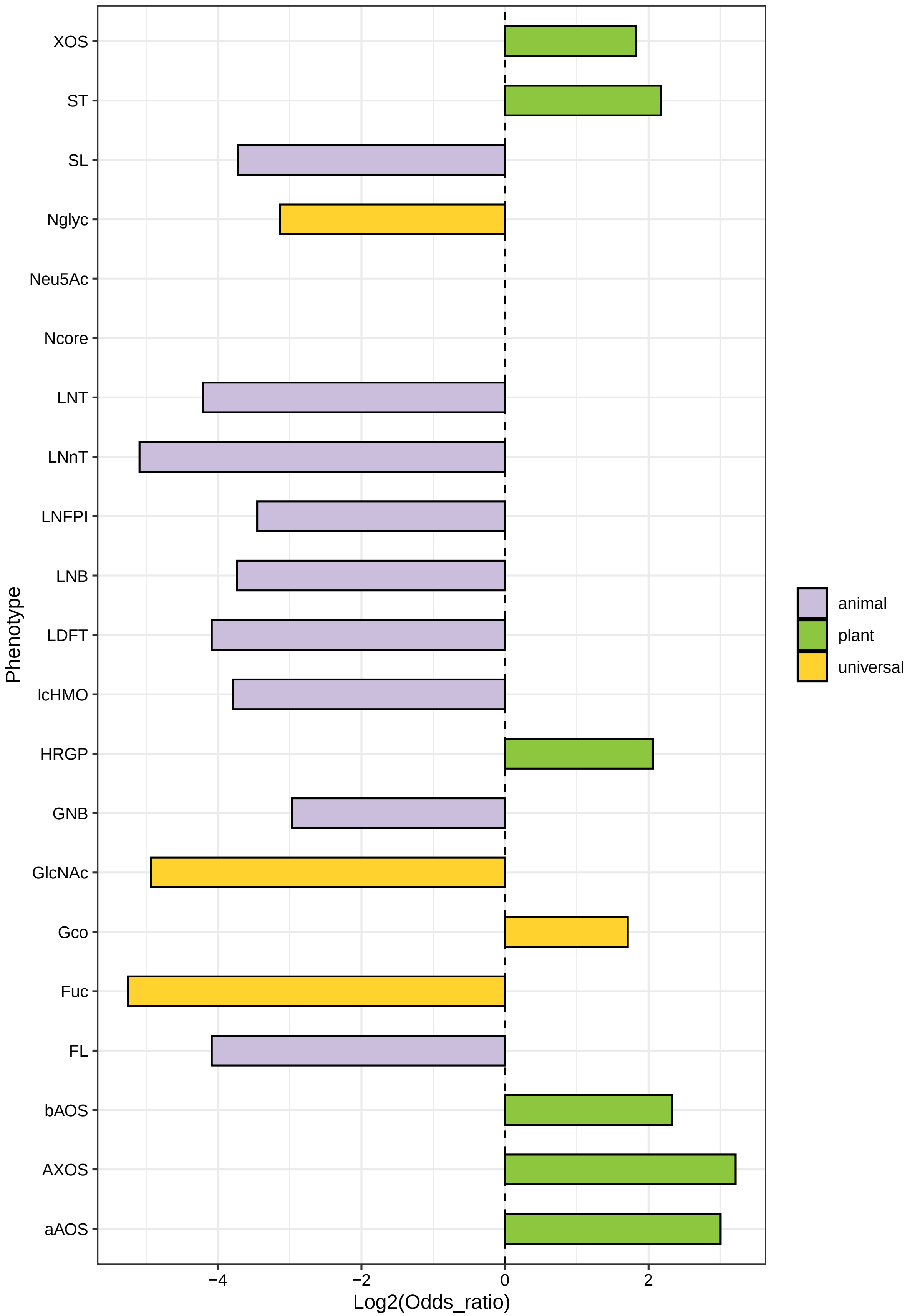
```
bpm_age_west_df <- bpm_2973_fish_df %>%
  filter(non_westernized == "no") %>%
  dplyr::select(genome_ID, c(Glc:Asc), age_group)
# create a contingency table
# i.e., calculate the number of occurrences of "yes" and "no" in the column `age_group` for # each bina
contingency_table_age <- bpm_age_west_df %>%
  pivot_longer(cols = -c(genome_ID, age_group), names_to = "phenotype", values_to = "value") %>%
  group_by(phenotype, age_group, value) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = age_group, values_from = count) %>%
  dplyr::select(phenotype, value, child, adult) %>%
  replace(is.na(.), 0)
# keep only rows where neither all values are 0 nor all values are 1 within each group
contingency_table_age <- contingency_table_age %>%
  group_by(phenotype) %>%
  filter(!all(value == 0) & !all(value == 1))
# perform the Fisher's exact test, calculate odds ratios and adjusted p-values
result_age <- contingency_table_age %>%
  group_by(phenotype) %>%
  summarise(p_value = fisher.test(matrix(c(child, adult), nrow = 2))$p.value,
            odds_ratio = fisher.test(matrix(c(child, adult), nrow = 2))$estimate,
            conf_int_low = fisher.test(matrix(c(child, adult), nrow = 2))$conf.int[1],
            conf_int_high = fisher.test(matrix(c(child, adult), nrow = 2))$conf.int[2]) %>%
  mutate(p_adj = p.adjust(p_value, method = "fdr"))
# add information about phenotypes
results_age_ann <- left_join(result_age, phenotype_metadata, by = "phenotype")
# create a data frame with the log2-transformed odds ratios
odds_age_df <- results_age_ann %>%
  mutate(log2_odds_ratio = log2(odds_ratio)) %>%
  mutate(log2_conf_int_low = log2(conf_int_low)) %>%
  mutate(log2_conf_int_high = log2(conf_int_high)) %>%
  arrange(desc(log2_odds_ratio)) %>%
  dplyr::select(phenotype, origin, odds_ratio, conf_int_low, conf_int_high,
                log2_odds_ratio, p_value, p_adj)

# plot barplot
ggplot(data = subset(odds_age_df, p_adj <= 0.01)) +
  aes(x = log2_odds_ratio, y = phenotype, color= origin, fill = origin) +
  geom_bar(stat = "identity", width = 0.5, color = "black") +
  geom_vline(xintercept = 0, color = "black", linetype = "dashed") +
  scale_color_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  scale_fill_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  labs(x = "Log2(Odds_ratio)", y = "Phenotype") +
  theme_bw() +
  theme(legend.title = element_blank(),
        plot.title = element_text(face="bold"),
        axis.title = element_text(color = "black"),
        axis.text = element_text(color = "black"))
```


Xyl
XOS
XL
XGL
ST
SL
Scr
scFOS
RST
Rib
RFO
Nglyc
Neu5Ac
Ncore
Muc
Mtl
MOS
Mel
Mal
LNT
LNnT
LNFPI
LNB
LDFT
lcHMO
lcFOS
Ino
IMO
HRGP
GNB
GluA
GlcNAc6S
GlcNAc
GlcA
Gco
GalAf
GA
Fuc
FL
DFA
Cld
bMAN
BGL12
Bgl
bAOS
AXOS
AX
Ara
AR
AGII
AGI
aAOS
Phenotype
−6
−4
−2
0
2
Log2(Odds_ratio)
animal
plant
universal

```
#
ggsave("results/phenotype_enrichment/child_westernized_vs_adult_westernized.pdf",
       device = "pdf", width = 7, height = 10)
```

9.10.4 non-Westernized (age < 3) vs. non-Westernized (age >= 3)

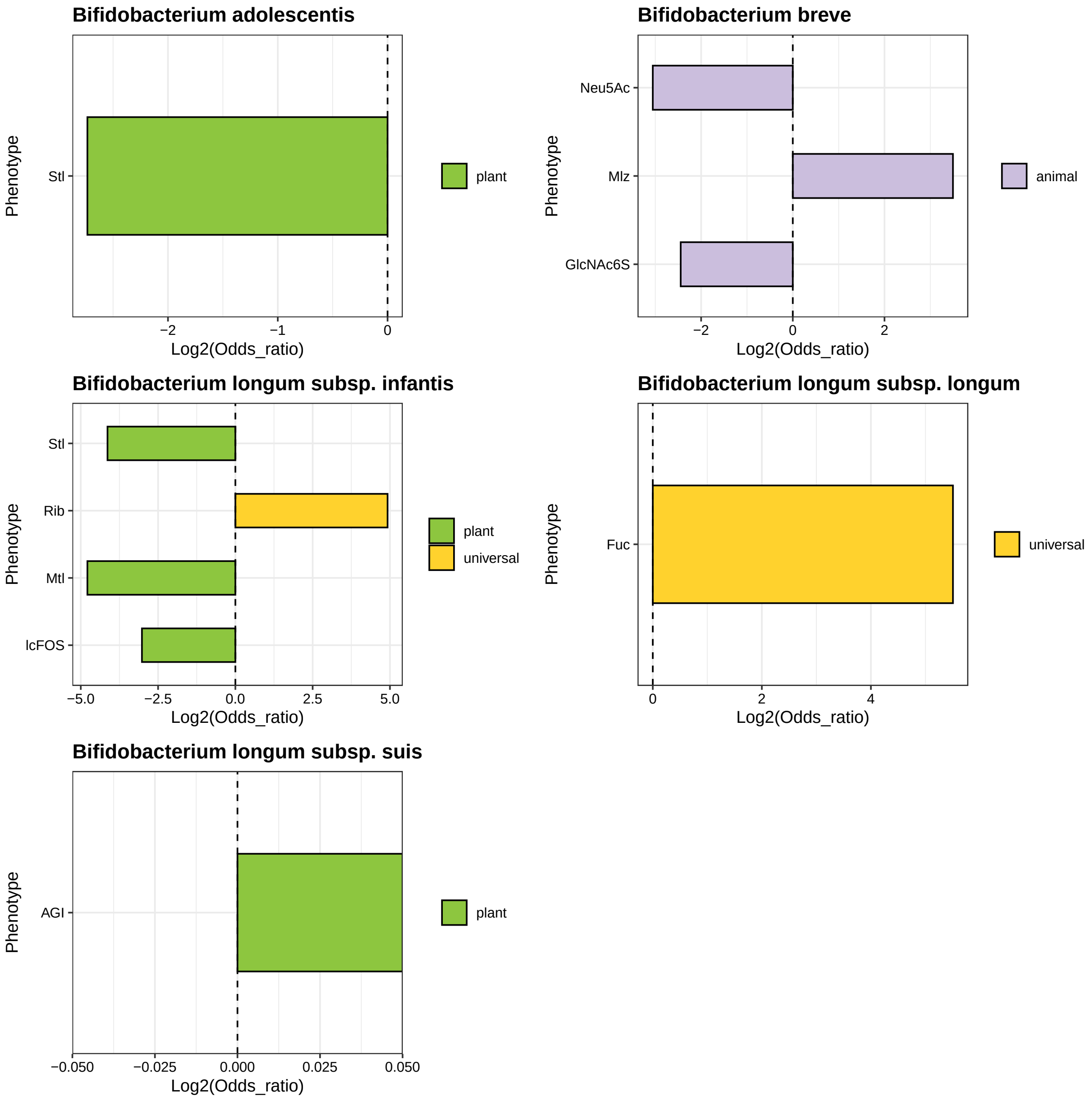
```
bpm_age_west_df <- bpm_2973_fish_df %>%
  filter(non_westerneized == "yes") %>%
  dplyr::select(genome_ID, c(Glc:Asc), age_group)
# create a contingency table
# i.e., calculate the number of occurrences of "yes" and "no" in the column `age_group` for # each bina
contingency_table_age <- bpm_age_west_df %>%
  pivot_longer(cols = -c(genome_ID, age_group), names_to = "phenotype", values_to = "value") %>%
  group_by(phenotype, age_group, value) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = age_group, values_from = count) %>%
  dplyr::select(phenotype, value, child, adult) %>%
  replace(is.na(.), 0)
# keep only rows where neither all values are 0 nor all values are 1 within each group
contingency_table_age <- contingency_table_age %>%
  group_by(phenotype) %>%
  filter(!all(value == 0) & !all(value == 1))
# perform the Fisher's exact test, calculate odds ratios and adjusted p-values
result_age <- contingency_table_age %>%
  group_by(phenotype) %>%
  summarise(p_value = fisher.test(matrix(c(child, adult), nrow = 2))$p.value,
            odds_ratio = fisher.test(matrix(c(child, adult), nrow = 2))$estimate,
            conf_int_low = fisher.test(matrix(c(child, adult), nrow = 2))$conf.int[1],
            conf_int_high = fisher.test(matrix(c(child, adult), nrow = 2))$conf.int[2]) %>%
  mutate(p_adj = p.adjust(p_value, method = "fdr"))
# add information about phenotypes
results_age_ann <- left_join(result_age, phenotype_metadata, by = "phenotype")
# create a data frame with the log2-transformed odds ratios
odds_age_df <- results_age_ann %>%
  mutate(log2_odds_ratio = log2(odds_ratio)) %>%
  mutate(log2_conf_int_low = log2(conf_int_low)) %>%
  mutate(log2_conf_int_high = log2(conf_int_high)) %>%
  arrange(desc(log2_odds_ratio)) %>%
  dplyr::select(phenotype, origin, odds_ratio, conf_int_low, conf_int_high,
                log2_odds_ratio, p_value, p_adj)

# plot barplot
ggplot(data = subset(odds_age_df, p_adj <= 0.01)) +
  aes(x = log2_odds_ratio, y = phenotype, color= origin, fill = origin) +
  geom_bar(stat = "identity", width = 0.5, color = "black") +
  geom_vline(xintercept = 0, color = "black", linetype = "dashed") +
  scale_color_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  scale_fill_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd", "plant" = "#8dc63f")) +
  labs(x = "Log2(Odds_ratio)", y = "Phenotype") +
  theme_bw() +
  theme(legend.title = element_blank(),
        plot.title = element_text(face="bold"),
        axis.title = element_text(color = "black"),
        axis.text = element_text(color = "black"))
```


XOS
ST
SL
Nglyc
Neu5Ac
Ncore
LNT
LNnT
LNFPI
LNB
LDFT
lcHMO
HRGP
GNB
GlcNAc
Gco
Fuc
FL
bAOS
AXOS
aAOS
Phenotype
−4
−2
0
2
Log2(Odds_ratio)
animal
plant
universal

```r
#
ggsave("results/phenotype_enrichment/child_non_westernized_vs_adult_non_westernized.pdf",
       device = "pdf", width = 7, height = 10)
```

9.10.5 Westernized vs. non-Westernized stratified by taxon

```r
# prepare data
bpm_2973_ado_df <- bpm_2973_fish_df %>%
  dplyr::select(curated_taxonomy, genome_ID, c(Glc:Asc), non_westernized)

# create a contingency table
contingency_table_west <- bpm_2973_ado_df %>%
  pivot_longer(cols = -c(genome_ID, curated_taxonomy, non_westernized),
               names_to = "phenotype", values_to = "value") %>%
  group_by(curated_taxonomy, phenotype, non_westernized, value) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = non_westernized, values_from = count) %>%
  replace(is.na(.), 0)

# keep only rows where neither all values are 0 nor all values are 1 within each group
contingency_table_west <- contingency_table_west %>%
  group_by(curated_taxonomy, phenotype) %>%
  filter(!all(value == 0) & !all(value == 1))

# perform the Fisher's exact test, calculate odds ratios and adjusted p-values
result_west <- contingency_table_west %>%
  group_by(curated_taxonomy, phenotype) %>%
  summarise(
    p_value = fisher.test(matrix(c(no, yes), nrow = 2))$p.value,
    odds_ratio = fisher.test(matrix(c(no, yes), nrow = 2))$estimate,
    conf_int_low = fisher.test(matrix(c(no, yes), nrow = 2))$conf.int[1],
    conf_int_high = fisher.test(matrix(c(no, yes), nrow = 2))$conf.int[2],
    .groups = 'drop'
  ) %>%
  group_by(curated_taxonomy) %>%
  mutate(p_adj = p.adjust(p_value, method = "fdr"))

# add information about phenotypes
results_west_ann <- left_join(result_west, phenotype_metadata, by = "phenotype")

# create a data frame with the log2-transformed odds ratios
odds_west_df <- results_west_ann %>%
  mutate(log2_odds_ratio = log2(odds_ratio)) %>%
  mutate(log2_conf_int_low = log2(conf_int_low)) %>%
  mutate(log2_conf_int_high = log2(conf_int_high)) %>%
  arrange(desc(log2_odds_ratio)) %>%
  dplyr::select(curated_taxonomy, phenotype, origin, odds_ratio, conf_int_low,
                conf_int_high, log2_odds_ratio, p_value, p_adj)

# create individual plots
plots <- odds_west_df %>%
  filter(p_adj <= 0.01) %>%
  split(.$curated_taxonomy) %>%
  map(~ ggplot(data = .x) +
        aes(x = log2_odds_ratio, y = phenotype, color = origin, fill = origin) +
        geom_bar(stat = "identity", width = 0.5, color = "black") +
        geom_vline(xintercept = 0, color = "black", linetype = "dashed") +
        scale_color_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd",
                                      "plant" = "#8dc63f")) +
        scale_fill_manual(values = c("universal" = "#ffd22d", "animal" = "#cbbedd",
                                     "plant" = "#8dc63f")) +
        labs(x = "Log2(Odds_ratio)", y = "Phenotype", title = .x$curated_taxonomy[1]) +
        theme_bw() +
        theme(legend.title = element_blank(),
              plot.title = element_text(face="bold"),
              axis.title = element_text(color = "black"),
              axis.text = element_text(color = "black")))

# combine plots using patchwork
combined_plot <- wrap_plots(plots, ncol = 2)
combined_plot
```



```
# save the combined plot
ggsave("results/phenotype_enrichment/taxon_west_nonwest_combined.pdf",
       combined_plot, device = "pdf", width = 14, height = 10)
```

9.10.6 Differences in taxonomic representation: Westernized vs. non-Westernized

```
tax_2973_west_df <- bpm_2973_fish_df %>%
  select(genome_ID, curated_taxonomy, non_westernized)
# create a contingency table
west_tax <- tax_2973_west_df %>%
  pivot_longer(cols = -c(genome_ID, non_westernized), names_to = "group", values_to = "value") %>%
  group_by(non_westernized, value) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = non_westernized, values_from = count) %>%
  replace(is.na(.), 0) %>%
  add_row(value = "total",
          `no` = sum(.[["no"]]),
          `yes` = sum(.[["yes"]]))
# calculate complement row for a given row
calculate_complement <- function(row, total_row) {
  new_row <- data.frame(
    value = paste0("NOT_", row$value),
    no = total_row$no - row$no,
    yes = total_row$yes - row$yes
  )
  return(new_row)
}
data <- west_tax
# filter out the "total" row
data_filtered <- data[data$value != "total", ]
# create a new data frame to store the results
new_data <- tibble()
# calculate complement rows
for (i in 1:nrow(data_filtered)) {
  original_row <- data_filtered[i, ]
  complement_row <- calculate_complement(original_row, data[data$value == "total", ])

  # append both original and complement rows to the new data frame
  new_data <- rbind(new_data, original_row, complement_row)
}
# add a column with taxonomy for grouping
west_tax_contingency <- new_data %>%
  mutate(curated_taxonomy = case_when(
    grepl("Bifidobacterium adolescentis", value) ~ "adolescentis",
    grepl("Bifidobacterium angulatum", value) ~ "angulatum",
    grepl("Bifidobacterium animalis subsp. animalis", value) ~ "animalis_animalis",
    grepl("Bifidobacterium animalis subsp. lactis", value) ~ "animalis_lactis",
    grepl("Bifidobacterium bifidum", value) ~ "bifidum",
    grepl("Bifidobacterium breve", value) ~ "breve",
    grepl("Bifidobacterium pseudocatenulatum", value) ~ "pseudocatenulatum",
    grepl("Bifidobacterium catenulatum subsp. catenulatum", value) ~ "catenulatum",
    grepl("Bifidobacterium catenulatum subsp. kashiwanohense_A", value) ~ "kashiwanohense_A",
    grepl("Bifidobacterium catenulatum subsp. kashiwanohense", value) ~ "kashiwanohense",
    grepl("Bifidobacterium sp002742445", value) ~ "sp002742445",
    grepl("Bifidobacterium dentium", value) ~ "dentium",
    grepl("Bifidobacterium gallicum", value) ~ "gallicum",
    grepl("Bifidobacterium longum subsp. infantis", value) ~ "longum_infantis",
    grepl("Bifidobacterium longum subsp. longum", value) ~ "longum_longum",
    grepl("Bifidobacterium longum subsp. nov.", value) ~ "longum_nov",
    grepl("Bifidobacterium longum subsp. suis", value) ~ "longum_suis",
    grepl("Bifidobacterium pseudolongum subsp. globosum", value) ~ "pseudolongum_globosum",
    grepl("Bifidobacterium pullorum", value) ~ "pullorum",
    grepl("Bifidobacterium scardovii", value) ~ "scardovii",
    grepl("Bifidobacterium thermophilum", value) ~ "thermophilum",
```

```r
    grepl("Bifidobacterium tsurumiense", value) ~ "tsurumiense",
    grepl("Bifidobacterium ruminantium", value) ~ "ruminantium",
    TRUE ~ NA_character_
  ))
# perform the Fisher's exact test, calculate odds ratios and adjusted p-values
result_west_tax <-  west_tax_contingency %>%
  select(curated_taxonomy , value, everything()) %>%
  group_by(curated_taxonomy ) %>%
  summarise(p_value = fisher.test(matrix(c(yes, no), nrow = 2))$p.value,
            odds_ratio = fisher.test(matrix(c(yes, no), nrow = 2))$estimate,
            conf_int_low = fisher.test(matrix(c(yes, no), nrow = 2))$conf.int[1],
            conf_int_high = fisher.test(matrix(c(yes, no), nrow = 2))$conf.int[2]) %>%
  mutate(p_adj = p.adjust(p_value, method = "fdr")) %>%
  select(curated_taxonomy, odds_ratio, conf_int_low, conf_int_high, p_value, p_adj)
gt(result_west_tax)
```

curated_taxonomy	odds_ratio	conf_int_low	conf_int_high	p_value	p_adj
adolescentis	1.1810377	0.715543666	1.8813341	4.644892e-01	8.217886e-01
angulatum	0.0000000	0.000000000	55.4726641	1.000000e+00	1.000000e+00
animalis_animalis	0.0000000	0.000000000	876.4214586	1.000000e+00	1.000000e+00
animalis_lactis	0.0000000	0.000000000	10.2492702	1.000000e+00	1.000000e+00
bifidum	0.4209965	0.176194849	0.8663903	1.569883e-02	4.011924e-02
breve	1.9676367	1.149058170	3.2305704	1.192031e-02	3.427088e-02
catenulatum	0.4513764	0.011118811	2.6807542	7.231021e-01	1.000000e+00
dentium	0.2107354	0.005245783	1.2226096	1.314462e-01	3.023263e-01
gallicum	0.0000000	0.000000000	876.4214586	1.000000e+00	1.000000e+00
kashiwanohense	1.0855868	0.026044289	6.8617442	6.126503e-01	1.000000e+00
kashiwanohense_A	Inf	15.292660522	Inf	2.983279e-06	2.287180e-05
longum_infantis	5.3494552	3.158973393	8.7831575	2.531235e-09	2.910921e-08
longum_longum	0.3421018	0.194104516	0.5711280	5.174685e-06	2.975444e-05
longum_nov	0.0000000	0.000000000	2.6177423	3.999317e-01	7.665357e-01
longum_suis	15.8562684	4.566544244	50.8901172	1.796849e-05	8.265503e-05
pseudocatenulatum	0.2115488	0.056379841	0.5607521	2.060663e-04	7.899209e-04
pseudolongum_globosum	3.2707330	0.072048471	25.7618718	2.911069e-01	6.086780e-01
pullorum	0.0000000	0.000000000	25.0474009	1.000000e+00	1.000000e+00
ruminantium	Inf	4.298491533	Inf	1.754236e-03	5.763918e-03
scardovii	0.0000000	0.000000000	19.5041101	1.000000e+00	1.000000e+00
sp002742445	30.7542058	10.698245666	91.3362767	3.976695e-10	9.146398e-09
thermophilum	0.0000000	0.000000000	876.4214586	1.000000e+00	1.000000e+00
tsurumiense	0.0000000	0.000000000	121.7222207	1.000000e+00	1.000000e+00

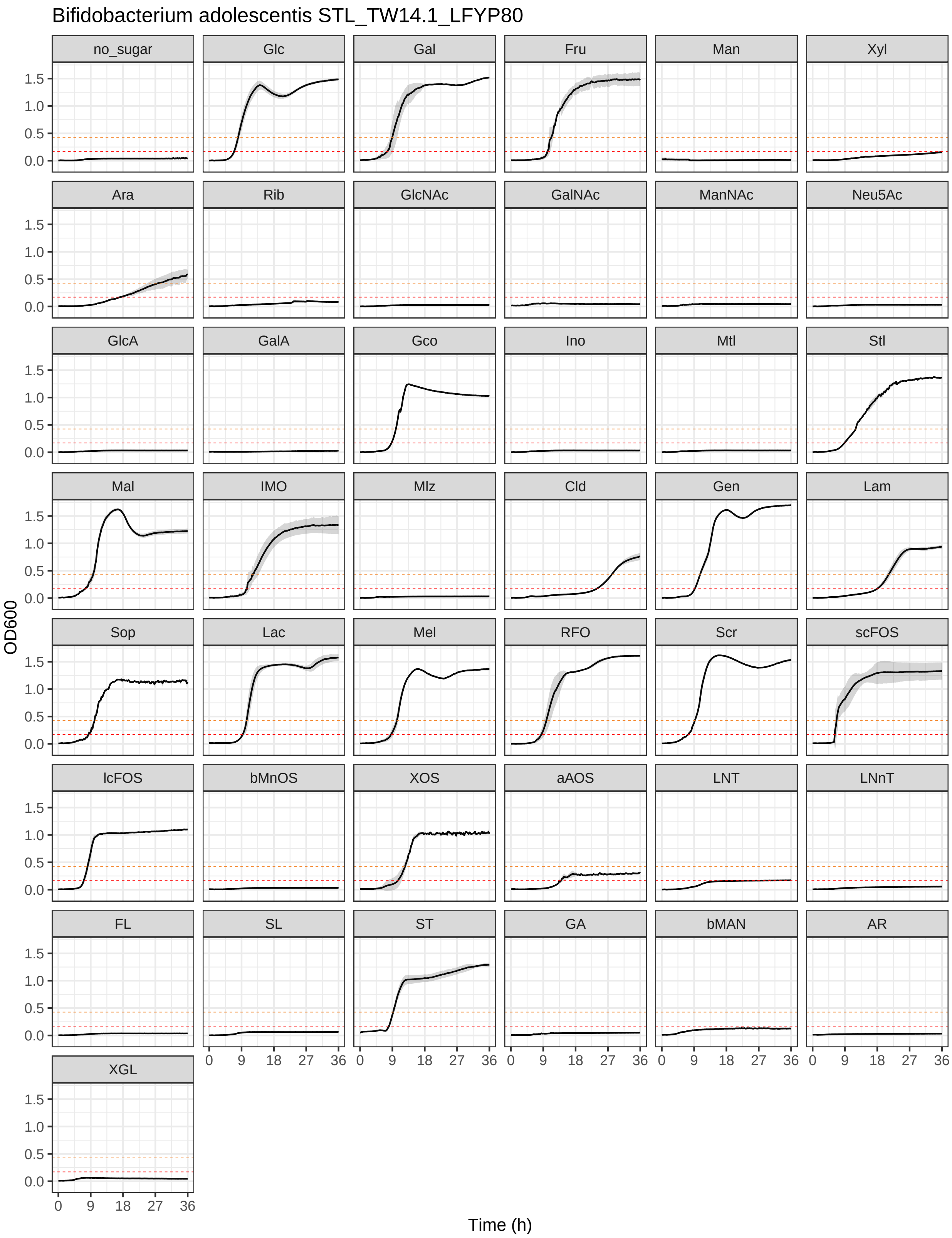
---

10 Analysis of *in vitro* growth data

10.1 *B. adolescentis* STL_TW14.1_LFYP80

Growth curves *B. adolescentis* STL_TW14.1_LFYP80 in MRS-AC supplemented with various (n = 42) carbon sources.

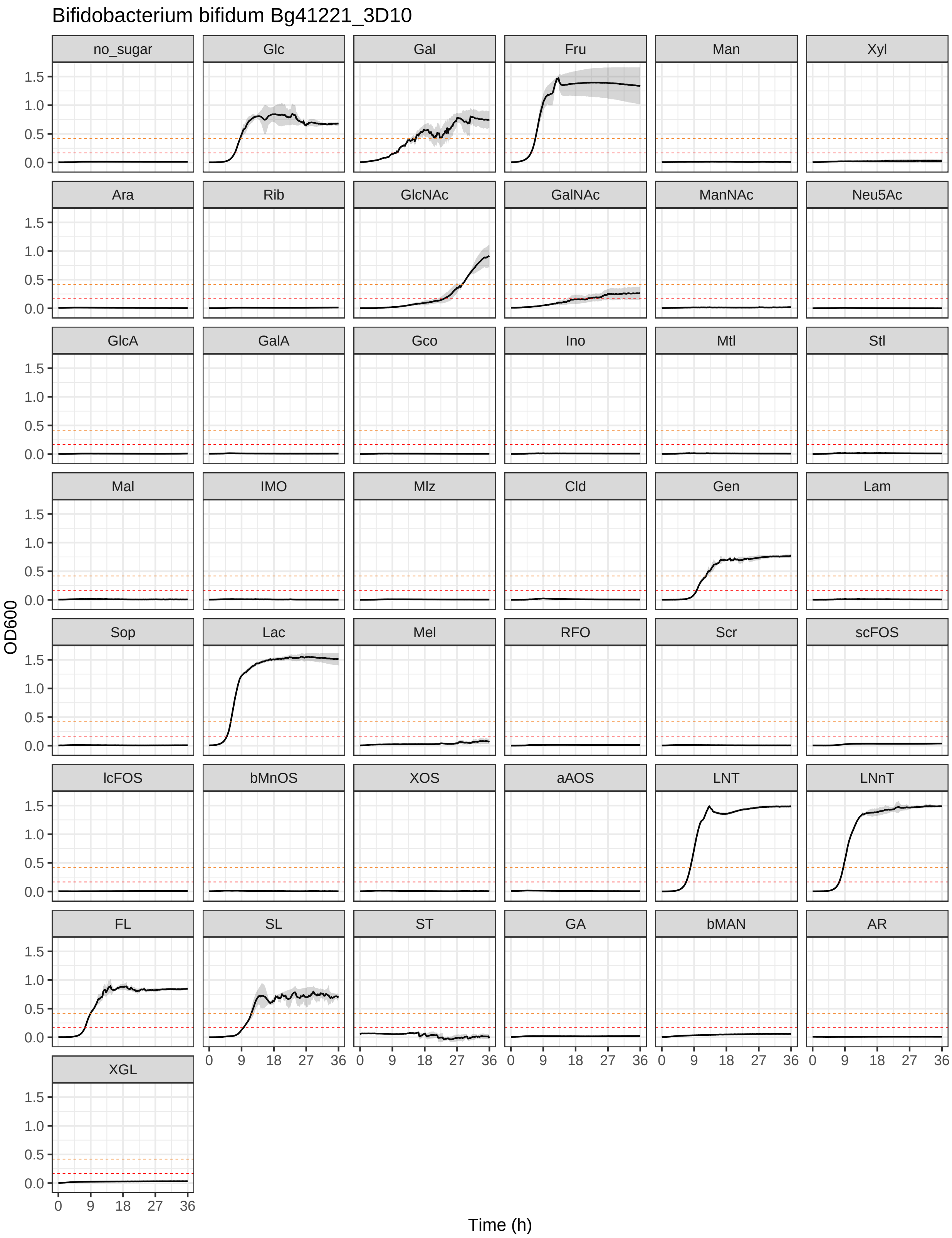
```r
growth_curves("Bifidobacterium adolescentis STL_TW14.1_LFYP80",
              "data/growth/formatted/Badolescentis_STL_TW14.1_LFYP80.txt",
              "results/growth/growth_curves/Badolescentis_STL_TW14.1_LFYP80.pdf",
              36)
```


Bifidobacterium adolescentis STL_TW14.1_LFYP80
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
1.5
1.0
0.5
0.0
0
9
18
27
36
Time (h)

10.2 *B. bifidum* Bg41221_3D10

Growth curves *B. bifidum* Bg41221_3D10 in MRS-AC supplemented with various (n = 42) carbon sources.

```r
growth_curves("Bifidobacterium bifidum Bg41221_3D10",
              "data/growth/formatted/Bbifidum_Bg41221_3D10.txt",
              "results/growth/growth_curves/Bbifidum_Bg41221_3D10.pdf",
              36)
```

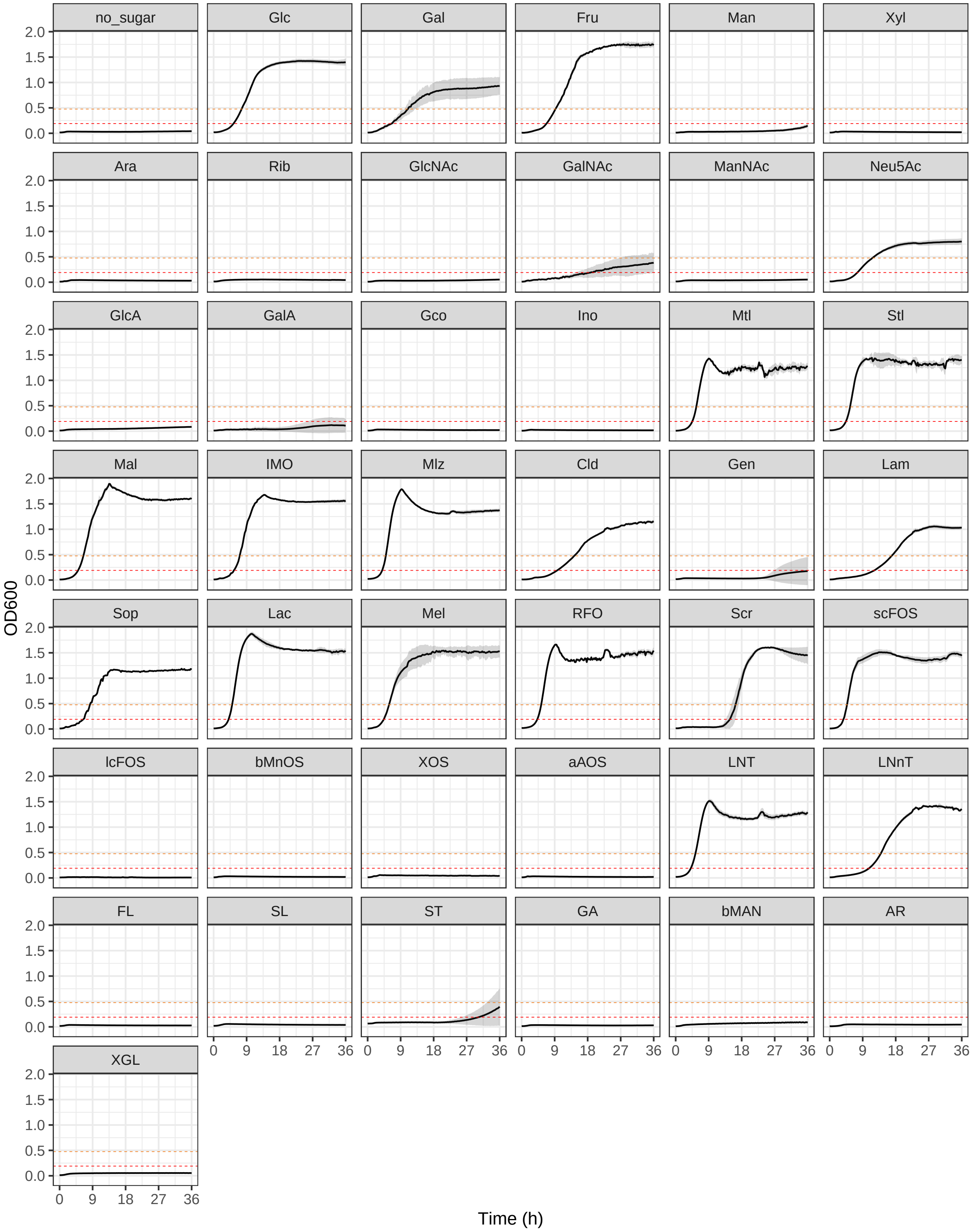

Bifidobacterium bifidum Bg41221_3D10
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
1.5
1.0
0.5
0.0
OD600
0
9
18
27
36
Time (h)

10.3 *B. breve* Bg155.S08_4F7

Growth curves *B. breve* Bg155.S08_4F7 in MRS-AC supplemented with various (n = 42) carbon sources.

```
growth_curves("Bifidobacterium breve Bg155.S08_4F7",
              "data/growth/formatted/Bbreve_Bg155.S08_4F7.txt",
              "results/growth/growth_curves/Bbreve_Bg155.S08_4F7.pdf",
              36)
```

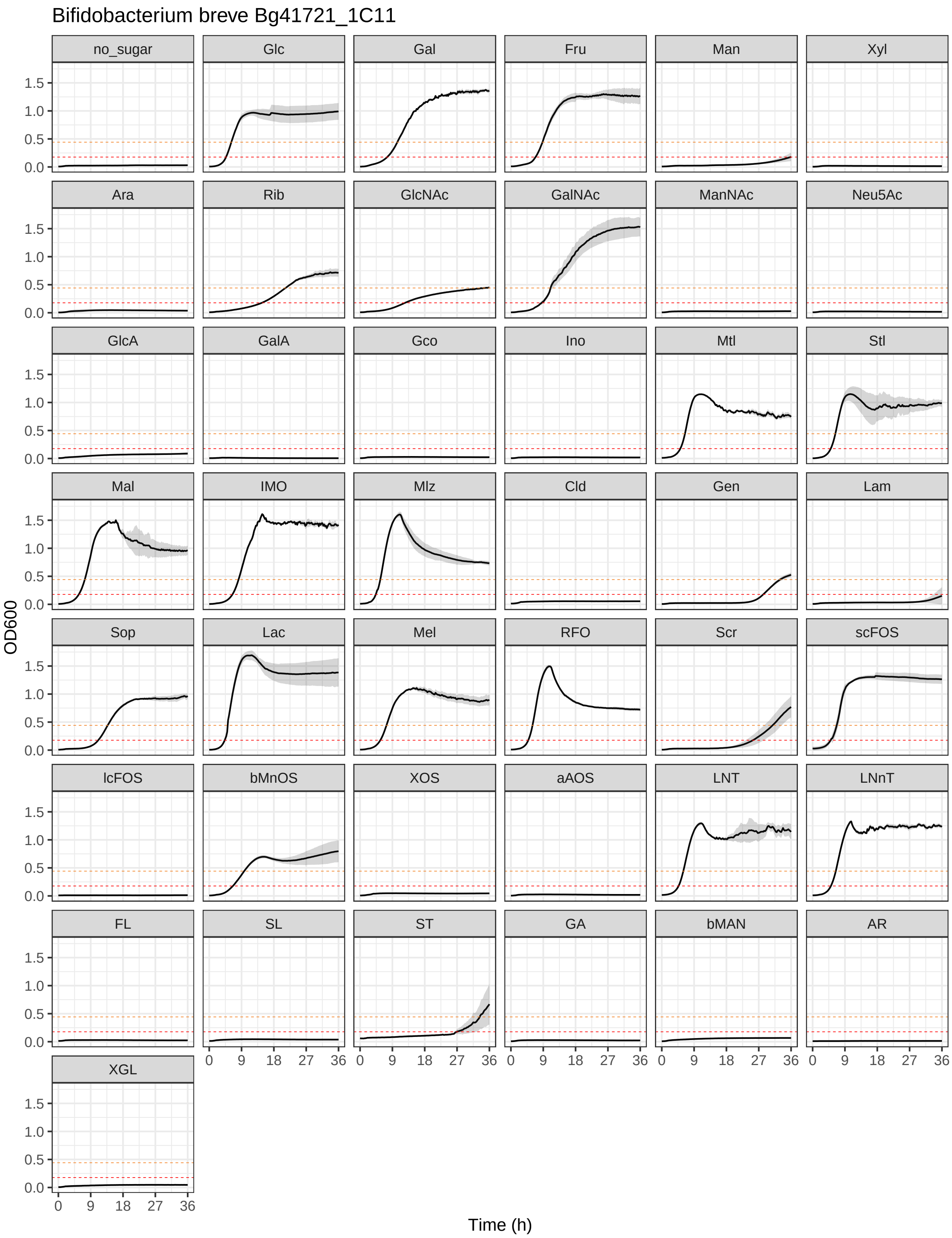
Bifidobacterium breve Bg155.S08_4F7



10.4 *B. breve* Bg41721_1C11

Growth curves *B. breve* Bg41721_1C11 in MRS-AC supplemented with various (n = 42) carbon sources.

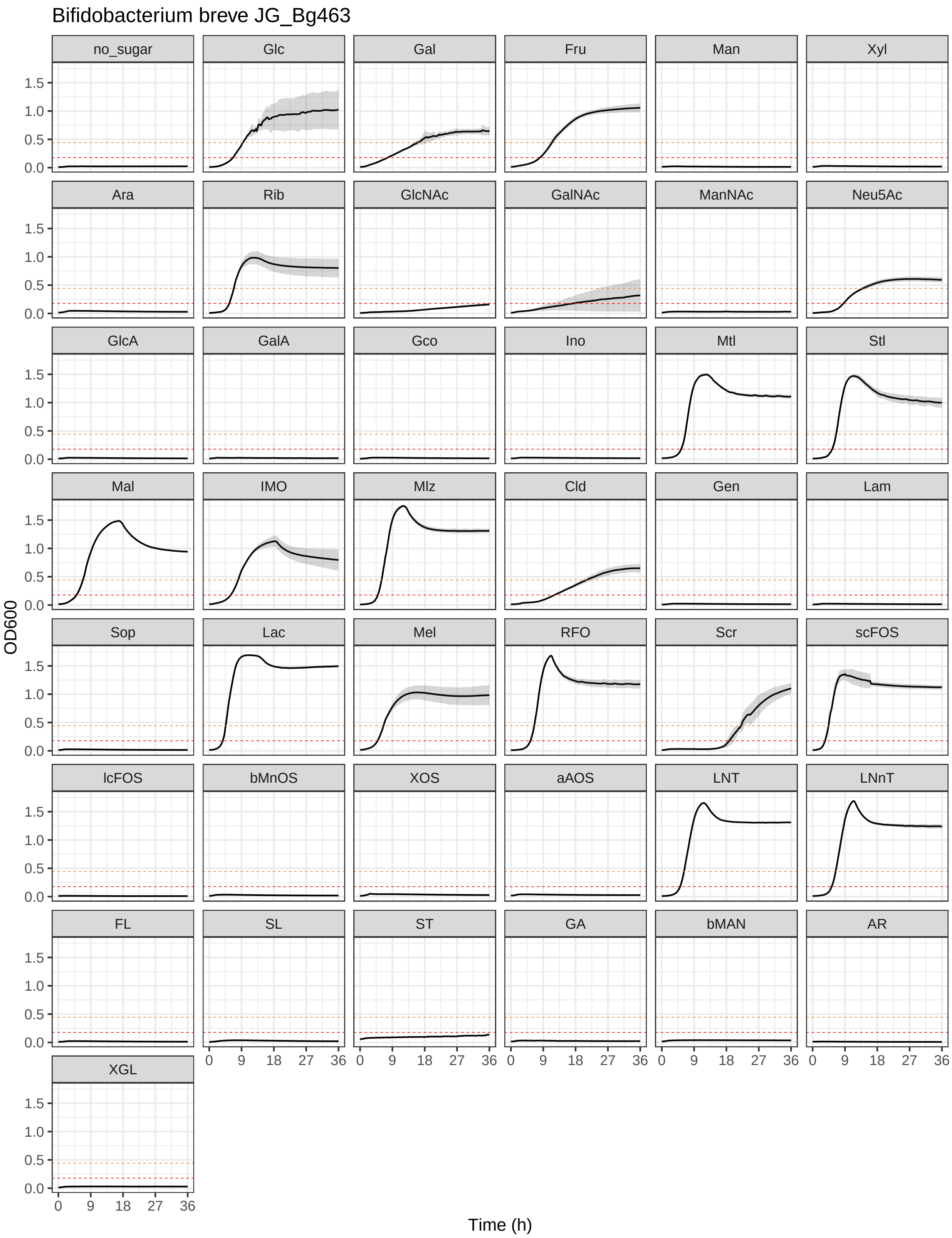
```r
growth_curves("Bifidobacterium breve Bg41721_1C11",
              "data/growth/formatted/Bbreve_Bg41721_1C11.txt",
              "results/growth/growth_curves/Bbreve_Bg41721_1C11.pdf",
              36)
```

Bifidobacterium breve Bg41721_1C11
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
1.5
1.0
0.5
0.0
0
9
18
27
36
Time (h)

10.5 *B. breve* JG_Bg463

Growth curves *B. breve* JG_Bg463 in MRS-AC supplemented with various (n = 42) carbon sources.

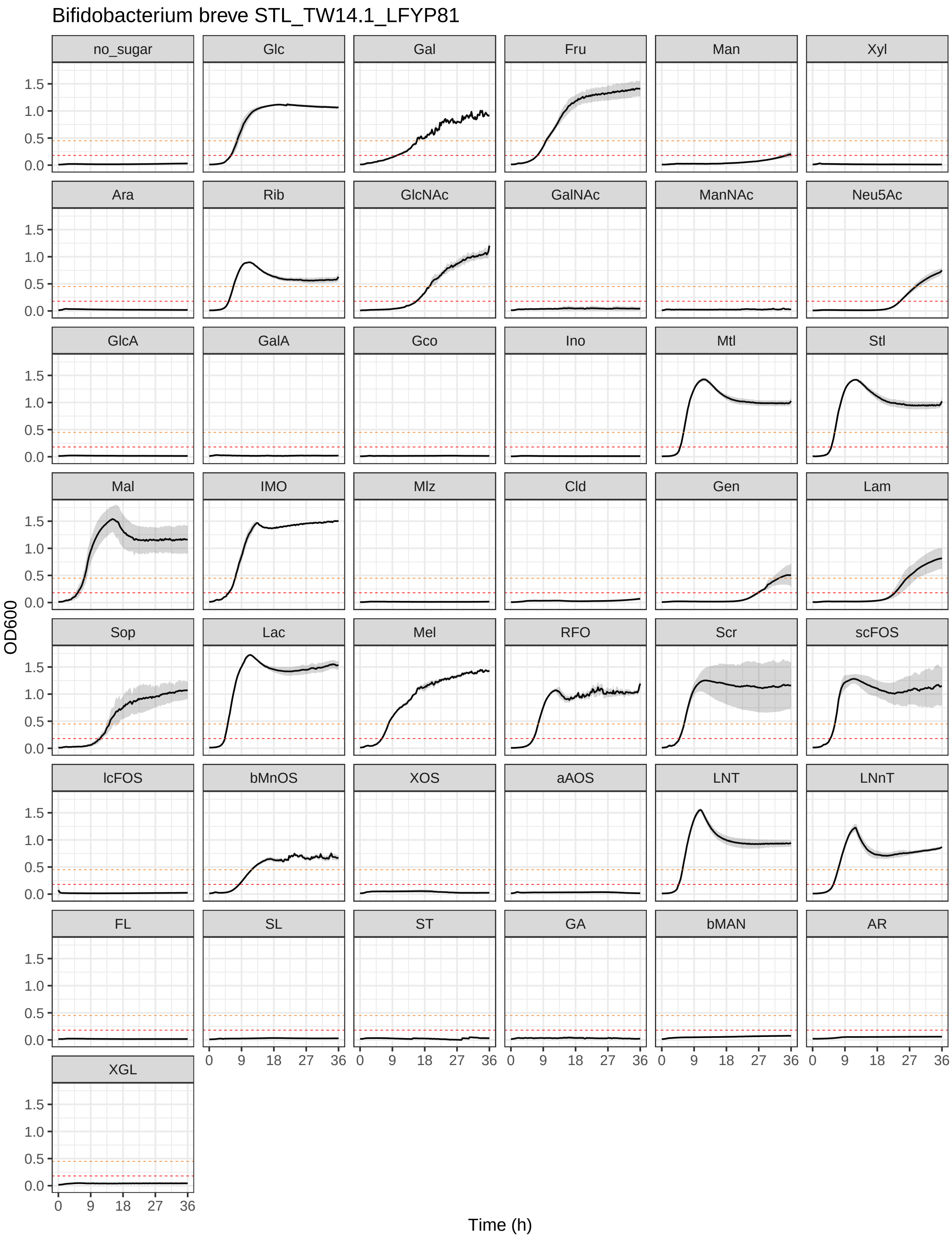
```
growth_curves("Bifidobacterium breve JG_Bg463",
              "data/growth/formatted/Bbreve_JG_Bg463.txt",
              "results/growth/growth_curves/Bbreve_JG_Bg463.pdf", 36)
```


Bifidobacterium breve JG_Bg463
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
0.0
0.5
1.0
1.5
0
9
18
27
36
Time (h)

10.6 *B. breve* STL_TW14.1_LFYP81

Growth curves *B. breve* STL_TW14.1_LFYP81 in MRS-AC supplemented with various (n = 42) carbon sources.

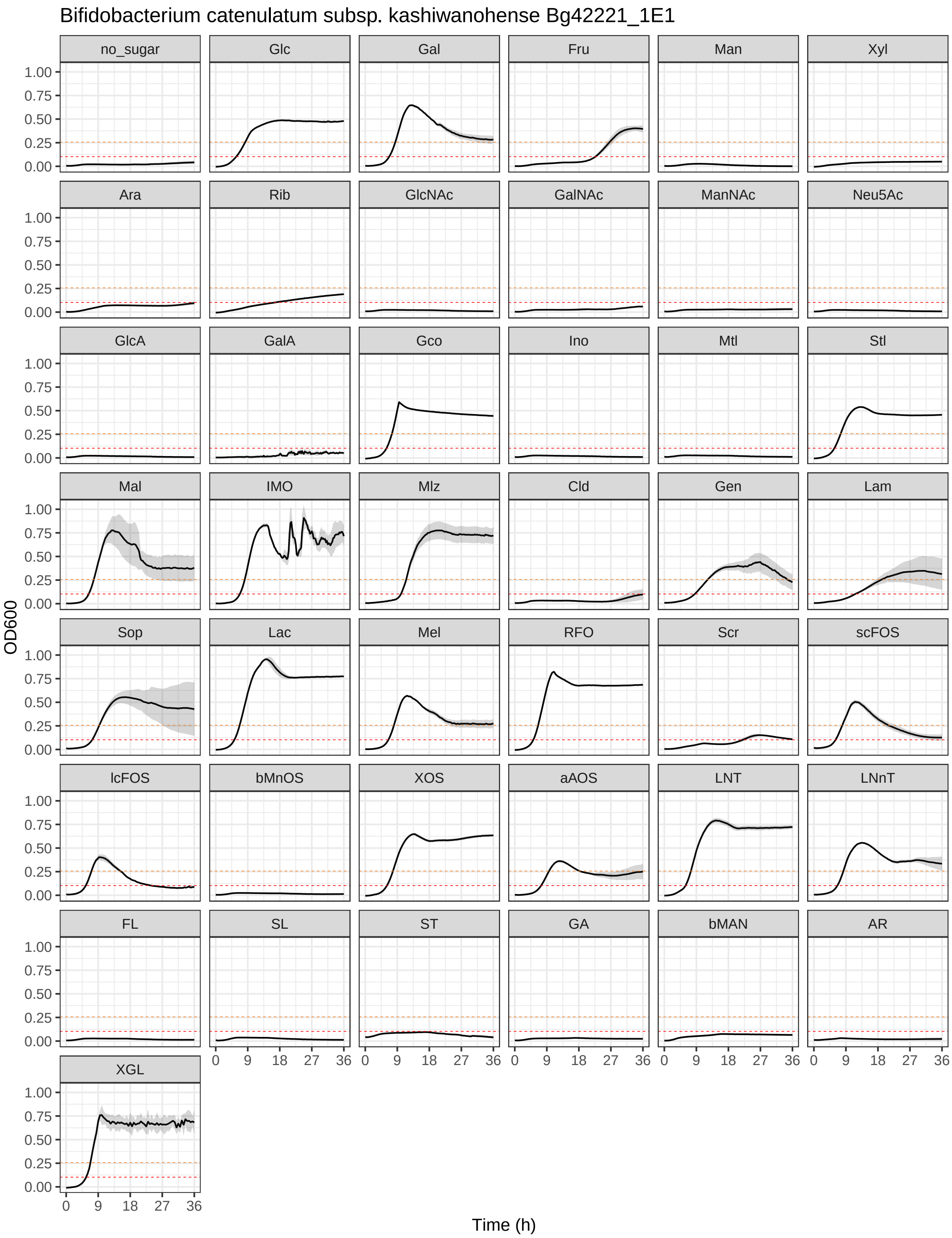
```
growth_curves("Bifidobacterium breve STL_TW14.1_LFYP81",
              "data/growth/formatted/Bbreve_STL_TW14.1_LFYP81.txt",
              "results/growth/growth_curves/Bbreve_STL_TW14.1_LFYP81.pdf",
              36)
```


Bifidobacterium breve STL_TW14.1_LFYP81
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
1.5
1.0
0.5
0.0
0
9
18
27
36
Time (h)

10.7 *B. catenulatum* subsp. *kashiwanohense* Bg42221_1E1

Growth curves *B. catenulatum* subsp. *kashiwanohense* Bg42221_1E1 in MRS-AC supplemented with various (n = 42) carbon sources.

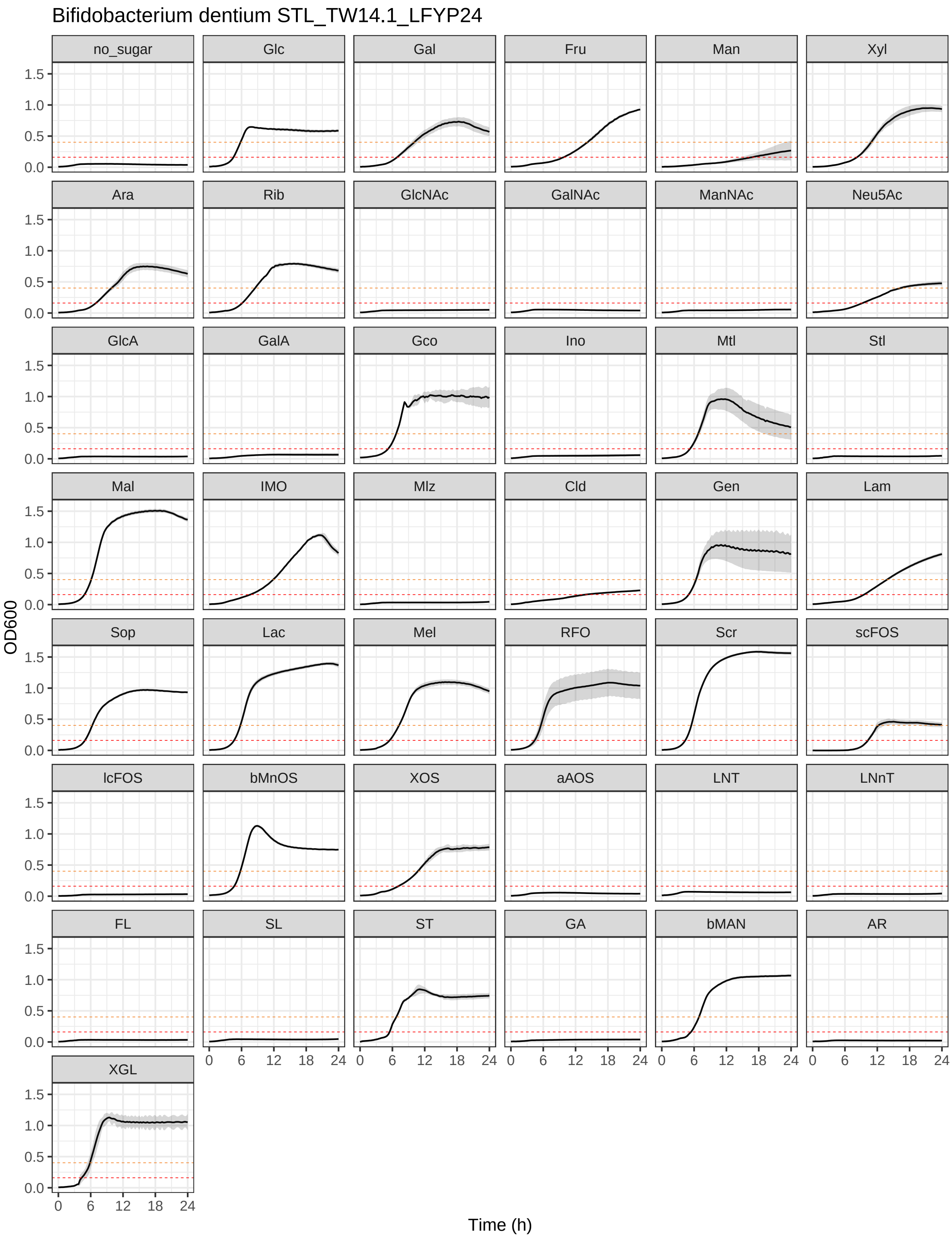
```r
growth_curves("Bifidobacterium catenulatum subsp. kashiwanohense Bg42221_1E1",
              "data/growth/formatted/Bc_kashiwanohense_Bg42221_1E1.txt",
              "results/growth/growth_curves/Bc_kashiwanohense_Bg42221_1E1.pdf",
              36)
```


Bifidobacterium catenulatum subsp. kashiwanohense Bg42221_1E1
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
1.00
0.75
0.50
0.25
0.00
0
9
18
27
36
Time (h)

10.8 *B. dentium* STL_TW14.1_LFYP24

Growth curves *B. dentium* STL_TW14.1_LFYP24 in MRS-AC supplemented with various (n = 42) carbon sources.

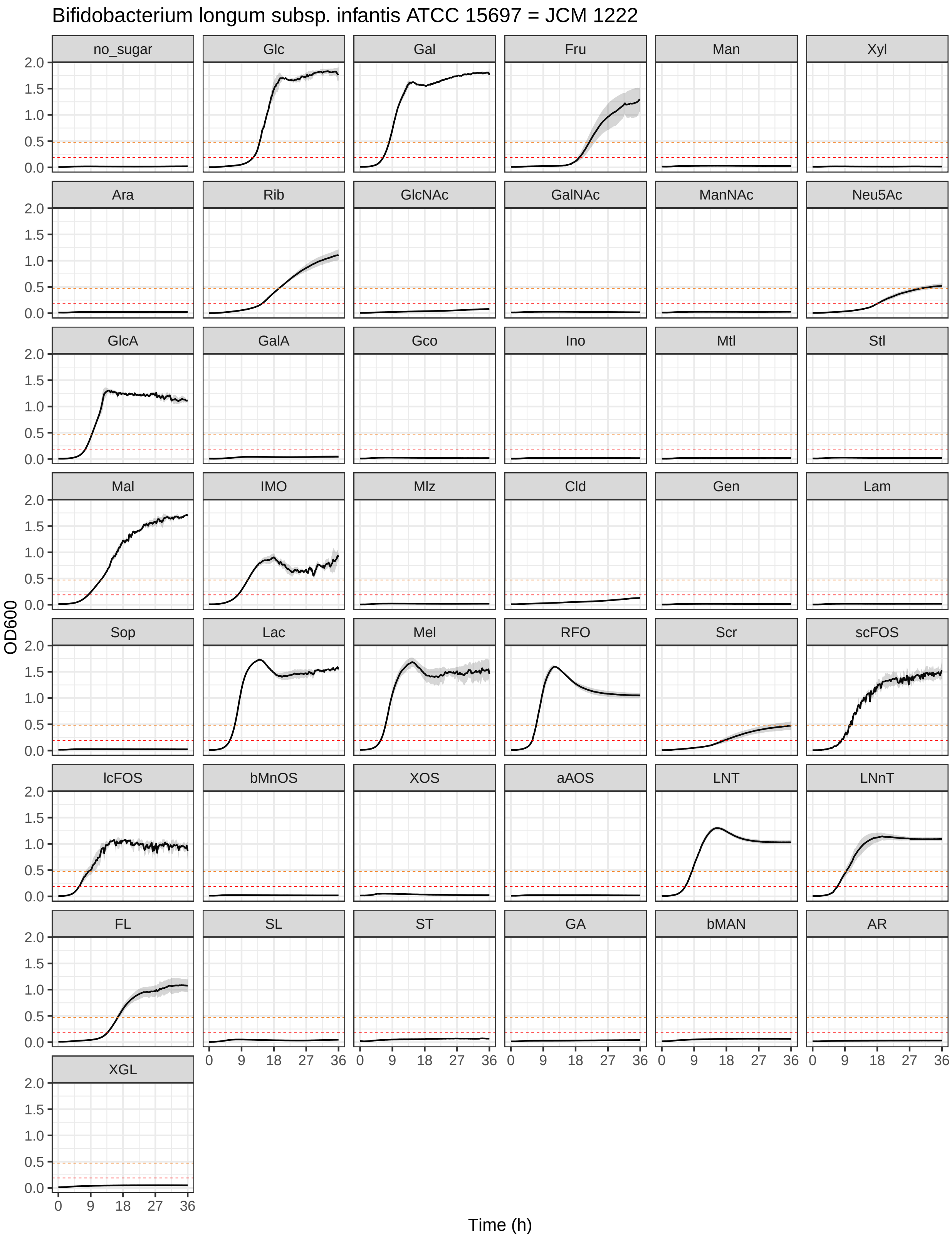
```r
growth_curves("Bifidobacterium dentium STL_TW14.1_LFYP24",
              "data/growth/formatted/Bdentium_STL_TW14.1_LFYP24.txt",
              "results/growth/growth_curves/Bdentium_STL_TW14.1_LFYP24.pdf", 24)
```


Bifidobacterium dentium STL_TW14.1_LFYP24
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
0.0
0.5
1.0
1.5
0
6
12
18
24
Time (h)

10.9 *B. longum* subsp. *infantis* ATCC 15697 = JCM 1222

Growth curves *B. longum* subsp. *infantis* ATCC 15697 = JCM 1222 in MRS-AC supplemented with various (n = 42) carbon sources.

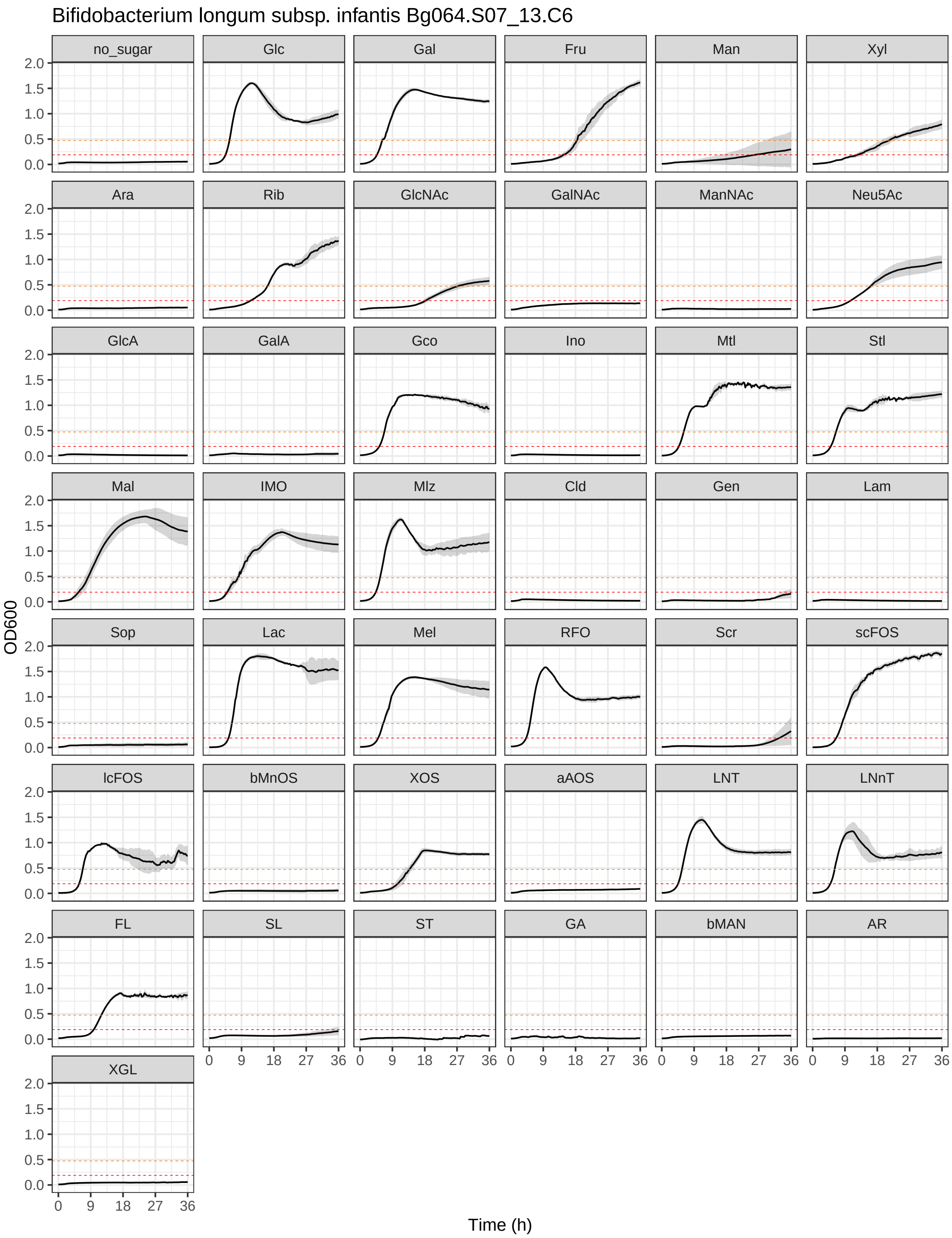
```
growth_curves("Bifidobacterium longum subsp. infantis ATCC 15697 = JCM 1222",
              "data/growth/formatted/Bl_infantis_ATCC15697.txt",
              "results/growth/growth_curves/Bl_infantis_ATCC15697.pdf",
              36)
```


Bifidobacterium longum subsp. infantis ATCC 15697 = JCM 1222
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
2.0
1.5
1.0
0.5
0.0
0
9
18
27
36
Time (h)

10.10 *B. longum* subsp. *infantis* Bg064.S07_13.C6

Growth curves *B. longum* subsp. *infantis* Bg064.S07_13.C6 in MRS-AC supplemented with various (n = 42) carbon sources.

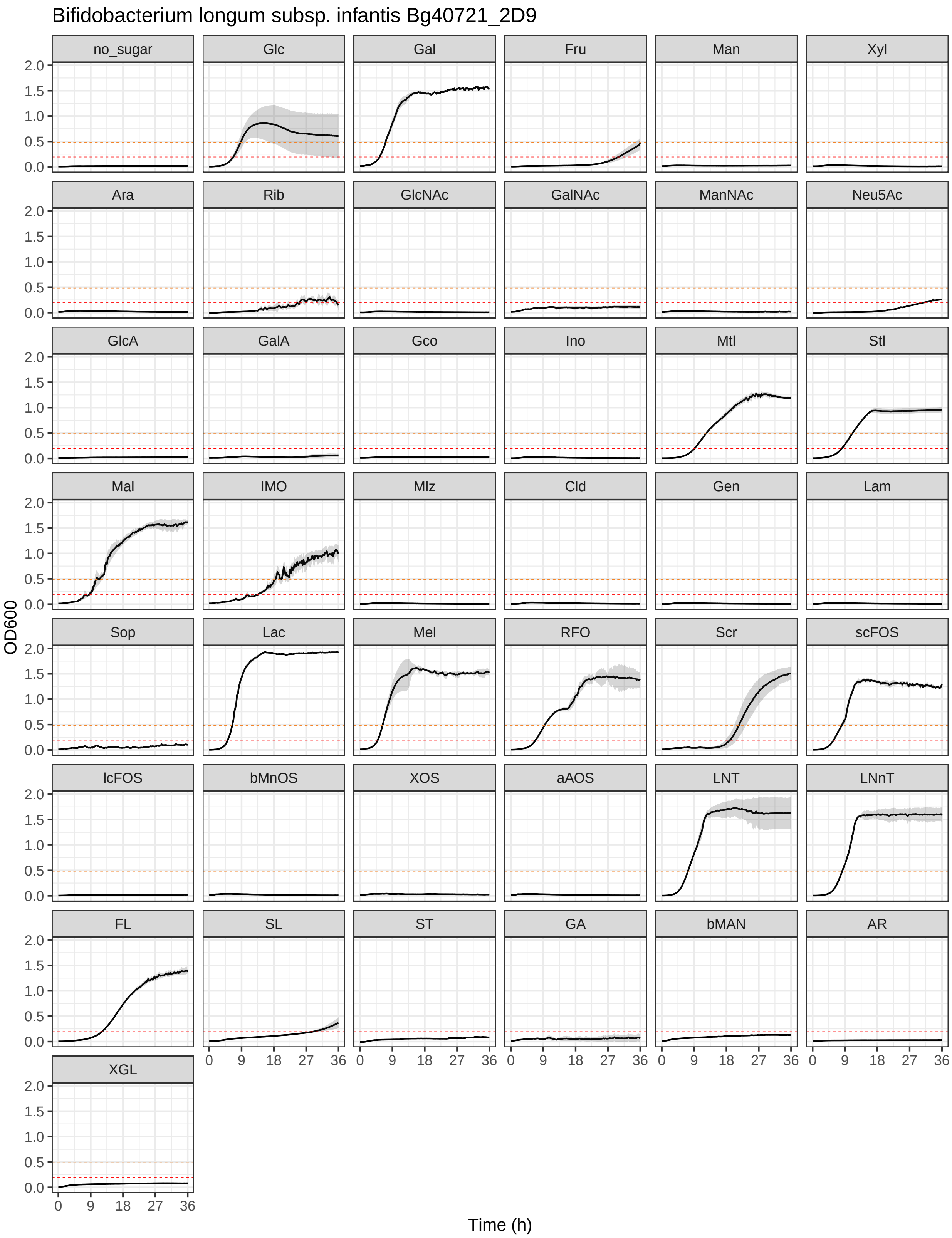
```r
growth_curves("Bifidobacterium longum subsp. infantis Bg064.S07_13.C6",
              "data/growth/formatted/Bl_infantis_Bg064.S07_13.C6.txt",
              "results/growth/growth_curves/Bl_infantis_Bg064.S07_13.C6.pdf",
              36)
```


Bifidobacterium longum subsp. infantis Bg064.S07_13.C6
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
0.0
0.5
1.0
1.5
2.0
0
9
18
27
36
Time (h)

10.11 *B. longum* subsp. *infantis* Bg40721_2D9

Growth curves *B. longum* subsp. *infantis* Bg40721_2D9 in MRS-AC supplemented with various (n = 42) carbon sources.

```r
growth_curves("Bifidobacterium longum subsp. infantis Bg40721_2D9",
              "data/growth/formatted/Bl_infantis_Bg40721_2D9.txt",
              "results/growth/growth_curves/Bl_infantis_Bg40721_2D9.pdf",
              36)
```

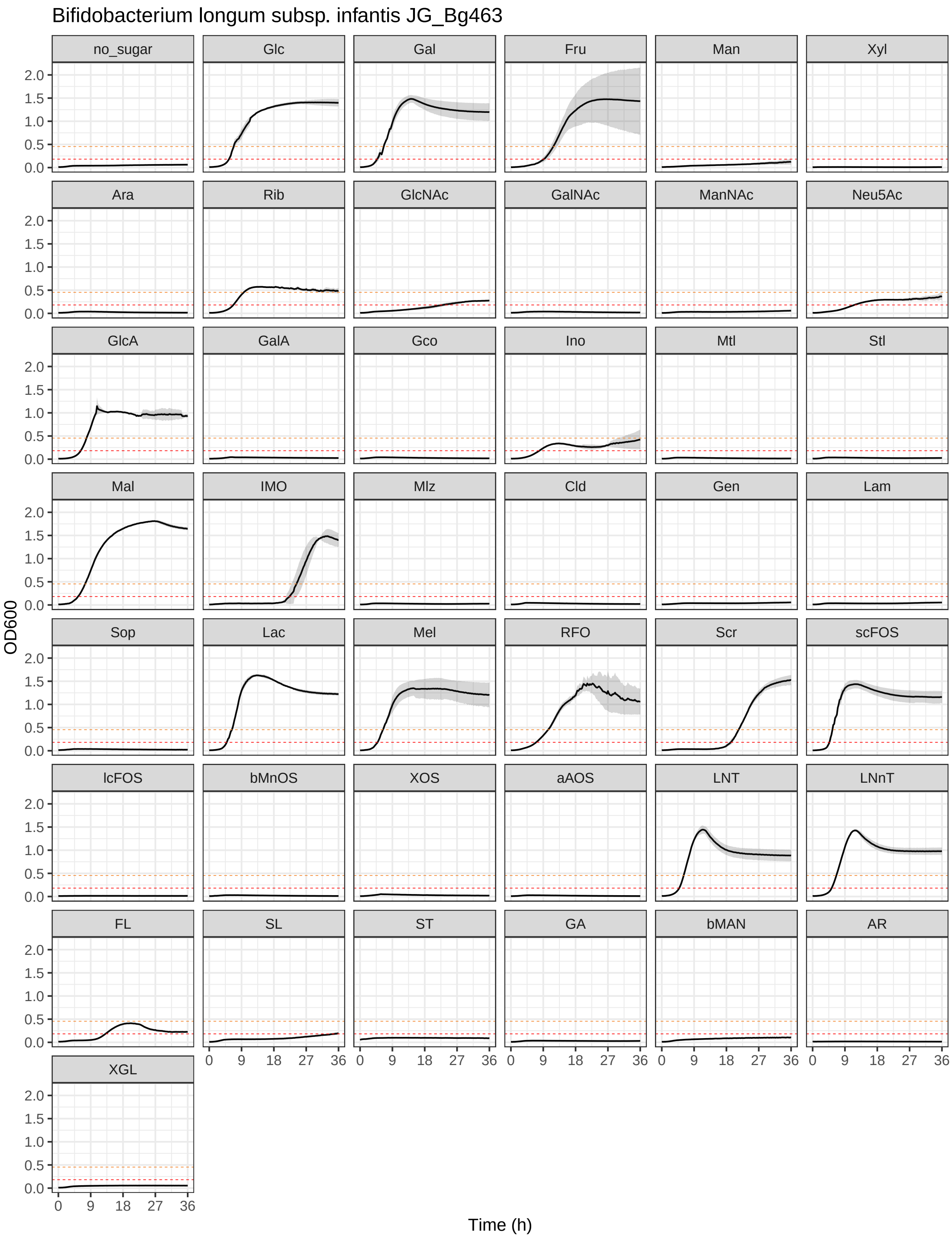


Bifidobacterium longum subsp. infantis Bg40721_2D9
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
2.0
1.5
1.0
0.5
0.0
0
9
18
27
36
Time (h)

10.12 *B. longum* subsp. *infantis* JG_Bg463

Growth curves *B. longum* subsp. *infantis* JG_Bg463 in MRS-AC supplemented with various (n = 42) carbon sources.

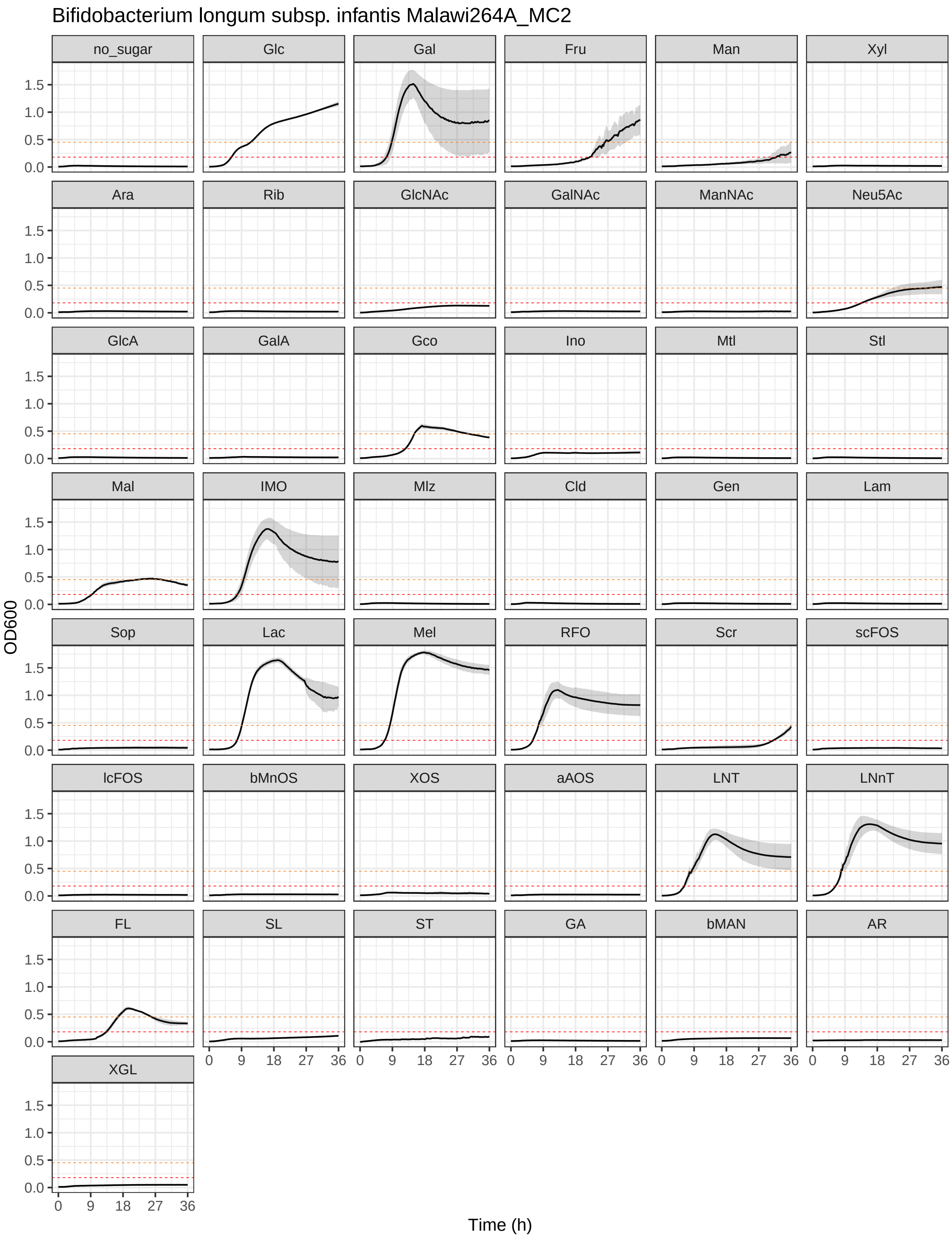
```r
growth_curves("Bifidobacterium longum subsp. infantis JG_Bg463",
              "data/growth/formatted/Bl_infantis_JG_Bg463.txt",
              "results/growth/growth_curves/Bl_infantis_JG_Bg463.pdf",
              36)
```


Bifidobacterium longum subsp. infantis JG_Bg463
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
0.0
0.5
1.0
1.5
2.0
0
9
18
27
36
Time (h)

10.13 *B. longum* subsp. *infantis* Malawi264A_MC2

Growth curves *B. longum* subsp. *infantis* Malawi264A_MC2 in MRS-AC supplemented with various (n = 42) carbon sources.

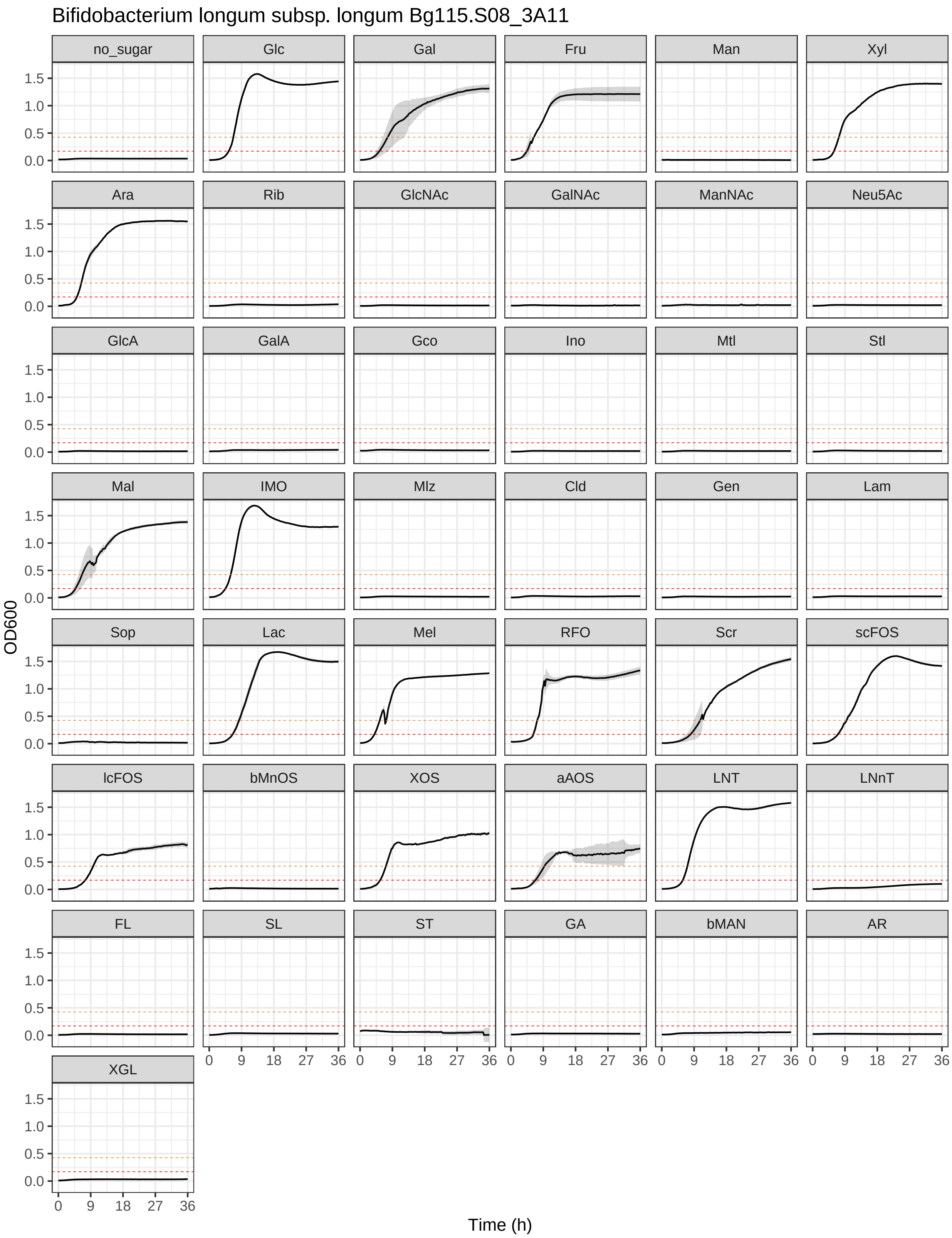
```
growth_curves("Bifidobacterium longum subsp. infantis Malawi264A_MC2",
              "data/growth/formatted/Bl_infantis_Malawi264A_MC2.txt",
              "results/growth/growth_curves/Bl_infantis_Malawi264A_MC2.pdf",
              36)
```


Bifidobacterium longum subsp. infantis Malawi264A_MC2
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
1.5
1.0
0.5
0.0
0
9
18
27
36
Time (h)

10.14 *B. longum* subsp. *longum* Bg115.S08_3A11

Growth curves *B. longum* subsp. *longum* Bg115.S08_3A11 in MRS-AC supplemented with various (n = 42) carbon sources.

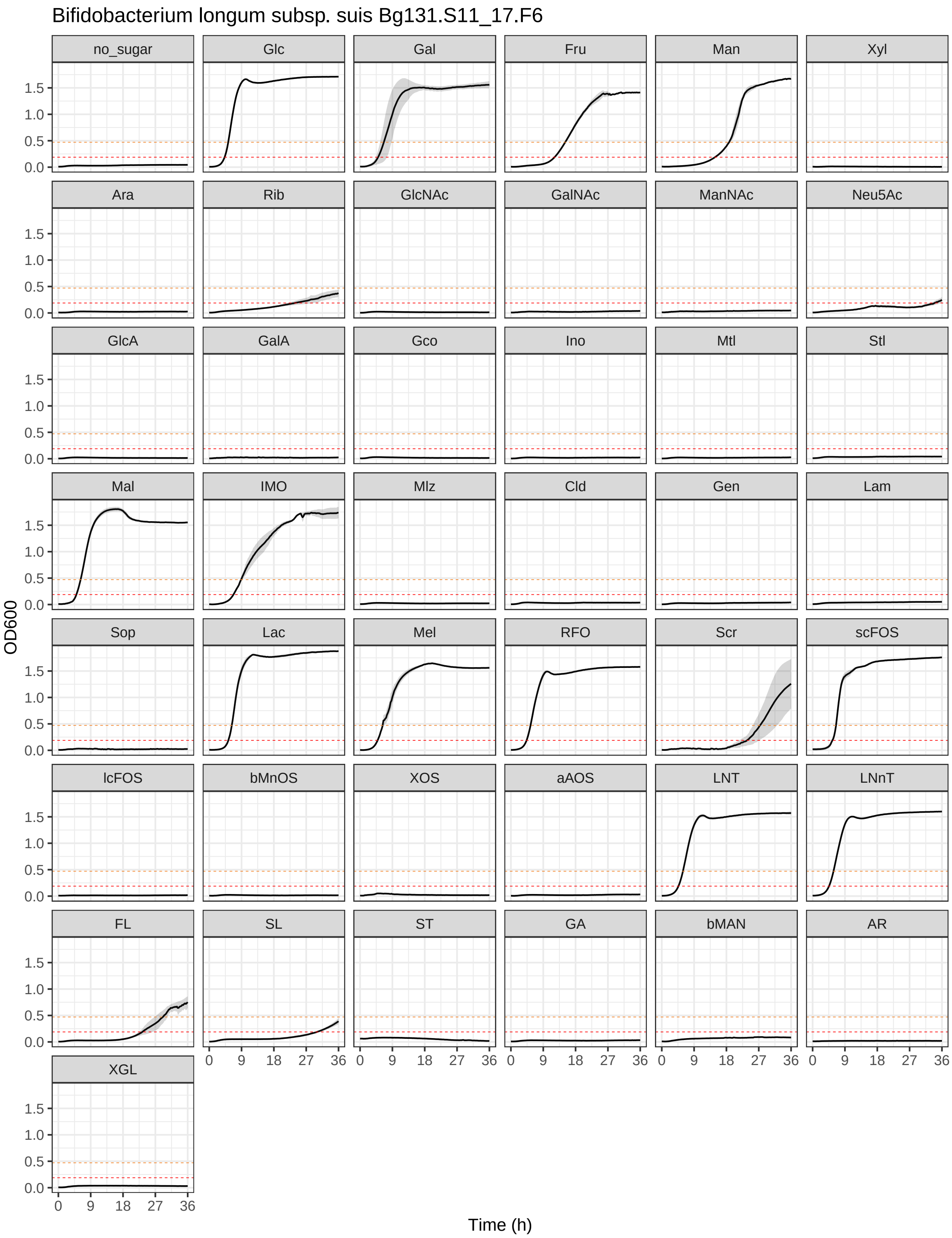
```
growth_curves("Bifidobacterium longum subsp. longum Bg115.S08_3A11",
              "data/growth/formatted/Bl_longum_Bg115.S08_3A11.txt",
              "results/growth/growth_curves/Bl_longum_Bg115.S08_3A11.pdf",
              36)
```


Bifidobacterium longum subsp. longum Bg115.S08_3A11
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
0.0
0.5
1.0
1.5
0
9
18
27
36
Time (h)

10.15 *B. longum* subsp. *suis* Bg131.S11_17.F6

Growth curves *B. longum* subsp. *suis* Bg131.S11_17.F6 in MRS-AC supplemented with various (n = 42) carbon sources.

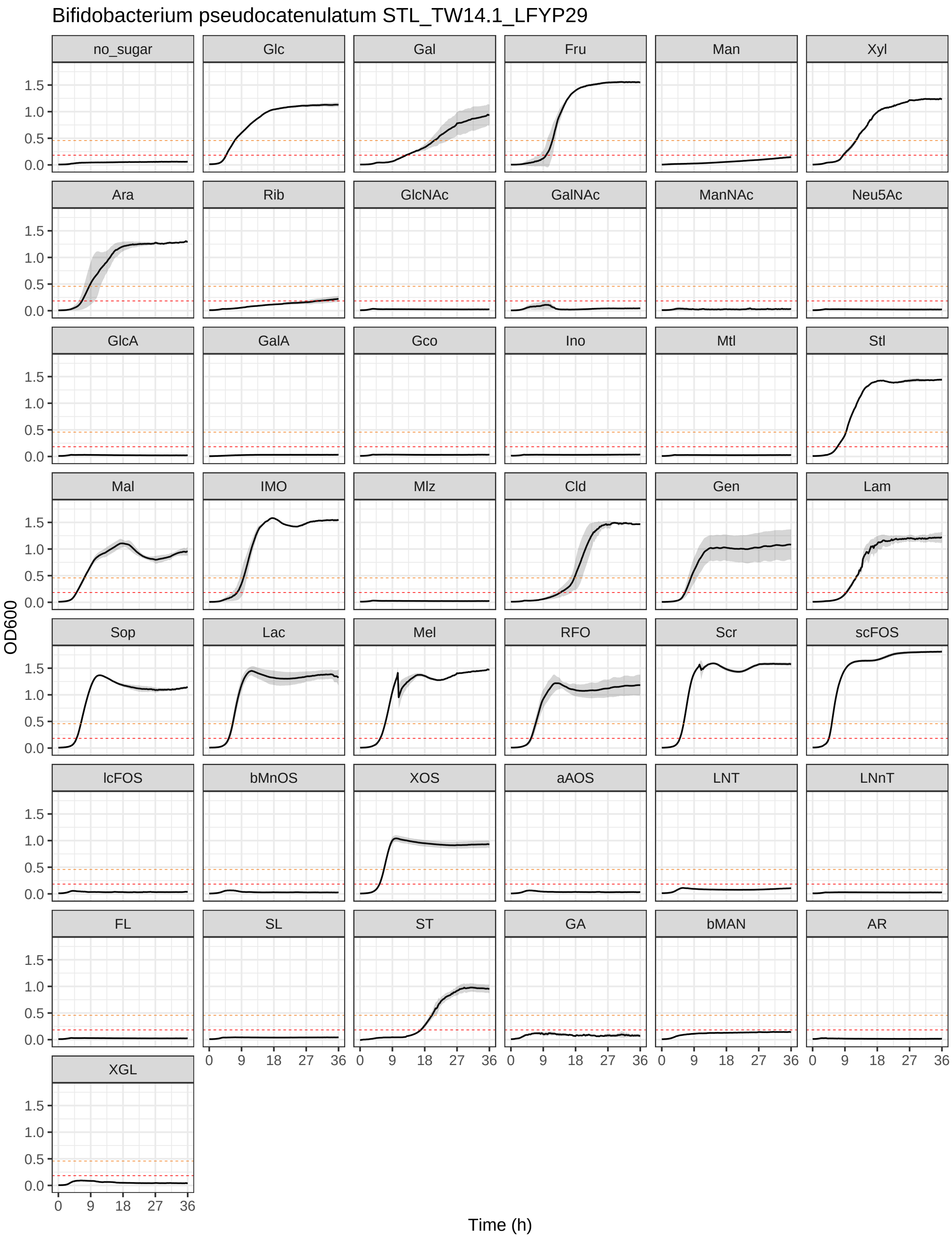
```r
growth_curves("Bifidobacterium longum subsp. suis Bg131.S11_17.F6",
              "data/growth/formatted/Bl_suis_Bg131.S11_17.F6.txt",
              "results/growth/growth_curves/Bl_suis_Bg131.S11_17.F6.pdf",
              36)
```


Bifidobacterium longum subsp. suis Bg131.S11_17.F6
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
0.0
0.5
1.0
1.5
0
9
18
27
36
Time (h)

10.16 *B. pseudocatenulatum* STL_TW14.1_LFYP29

Growth curves *B.* p*seudocatenulatum* STL_TW14.1_LFYP29 in MRS-AC supplemented with various (n = 42) carbon sources.

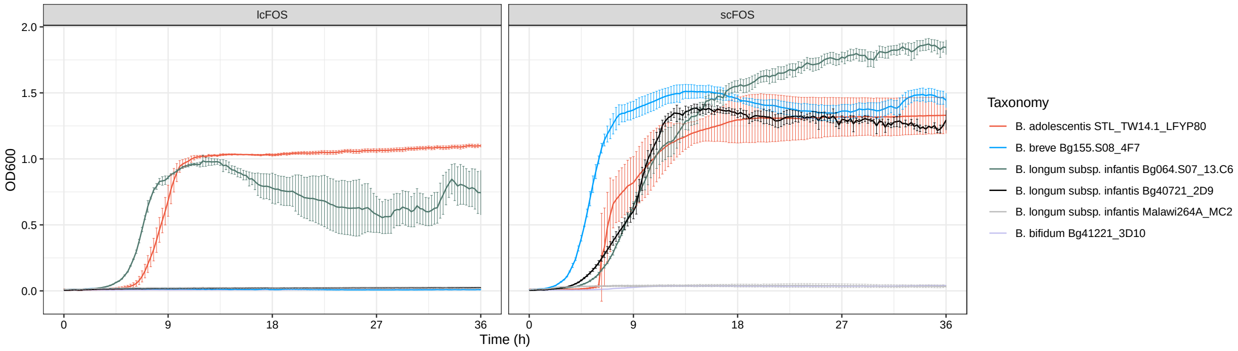
```r
growth_curves("Bifidobacterium pseudocatenulatum STL_TW14.1_LFYP29",
              "data/growth/formatted/Bpseudocatenulatum_STL_TW14.1_LFYP29.txt",
              "results/growth/growth_curves/Bpseudocatenulatum_STL_TW14.1_LFYP29.pdf",
              36)
```


Bifidobacterium pseudocatenulatum STL_TW14.1_LFYP29
no_sugar
Glc
Gal
Fru
Man
Xyl
Ara
Rib
GlcNAc
GalNAc
ManNAc
Neu5Ac
GlcA
GalA
Gco
Ino
Mtl
Stl
Mal
IMO
Mlz
Cld
Gen
Lam
Sop
Lac
Mel
RFO
Scr
scFOS
lcFOS
bMnOS
XOS
aAOS
LNT
LNnT
FL
SL
ST
GA
bMAN
AR
XGL
OD600
1.5
1.0
0.5
0.0
0
9
18
27
36
Time (h)

10.17 Selected growth curves

10.17.1 Growth in MRS-AC supplemented with 1% scFOS/lcFOS

```
# set input files
FOS_output_file <- "results/growth/FOS.pdf"
FOS_data <- "data/growth/formatted_selected/FOS/growth_FOS.txt"
FOS_annotation <- "data/growth/formatted_selected/FOS/growth_FOS_ann.txt"
FOS_trim_at_time <- 36 # measurements until [input] hours
FOS_blank_wells <- c("blank_1","blank_2","blank_3") # specify blank wells

# read a file with measurements
FOS_d <- read_tsv(FOS_data) %>%
  filter(time <= FOS_trim_at_time)
# calculate mean blank measurements (MRS-AC-Lac without added cells)
FOS_blank <- FOS_d %>%
  select(all_of(FOS_blank_wells)) %>%
  rowMeans()
# subtract blank measurements and round OD600 values to 3 digits
FOS_d_bl <- FOS_d %>%
  mutate(across(2:last_col(), ~ .x - FOS_blank)) %>%
  mutate(across(2:last_col(), ~ round(.x, 3)))
# read files with plate annotations
FOS_ann <- read_tsv(FOS_annotation)
FOS_d_bl_long <- FOS_d_bl %>%
  gather(., well, od, -time)
# create annotated tables
FOS_d_bl_long_ann <- left_join(FOS_d_bl_long, FOS_ann, by="well") %>%
  filter(carbon_source != "blank")
# plot growth curves
ggplot(FOS_d_bl_long_ann, aes(time, od, color = strain, fill = strain)) +
  #geom_point() +
  # add line connecting means from three replicates
  stat_summary(
    fun = mean,
    geom='line',
    size=0.5) +
  # add errorbars (standard deviation)
  stat_summary(
    fun.data=mean_sd,
    geom='errorbar',
    size=0.2,
    width=0.2,
    alpha=1) +
  scale_color_manual(name = "Taxonomy",
                     breaks=c("B. adolescentis STL_TW14.1_LFYP80",
                              "B. breve Bg155.S08_4F7",
                              "B. longum subsp. infantis Bg064.S07_13.C6",
                              "B. longum subsp. infantis Bg40721_2D9",
                              "B. longum subsp. infantis Malawi264A_MC2",
                              "B. bifidum Bg41221_3D10"),
                     values=c("tomato2", "#00a2ff", "#51796f", "black", "grey", "#c5c2f0")) +
  scale_x_continuous(limits=c(0, FOS_trim_at_time),
                     breaks=c(0, FOS_trim_at_time*0.25, FOS_trim_at_time*0.5,
                              FOS_trim_at_time*0.75, FOS_trim_at_time)) +
  xlab("Time (h)") +
  ylab("OD600") +
  theme_bw() +
  theme(axis.text = element_text(color = "black")) +
  facet_wrap(~carbon_source)
```

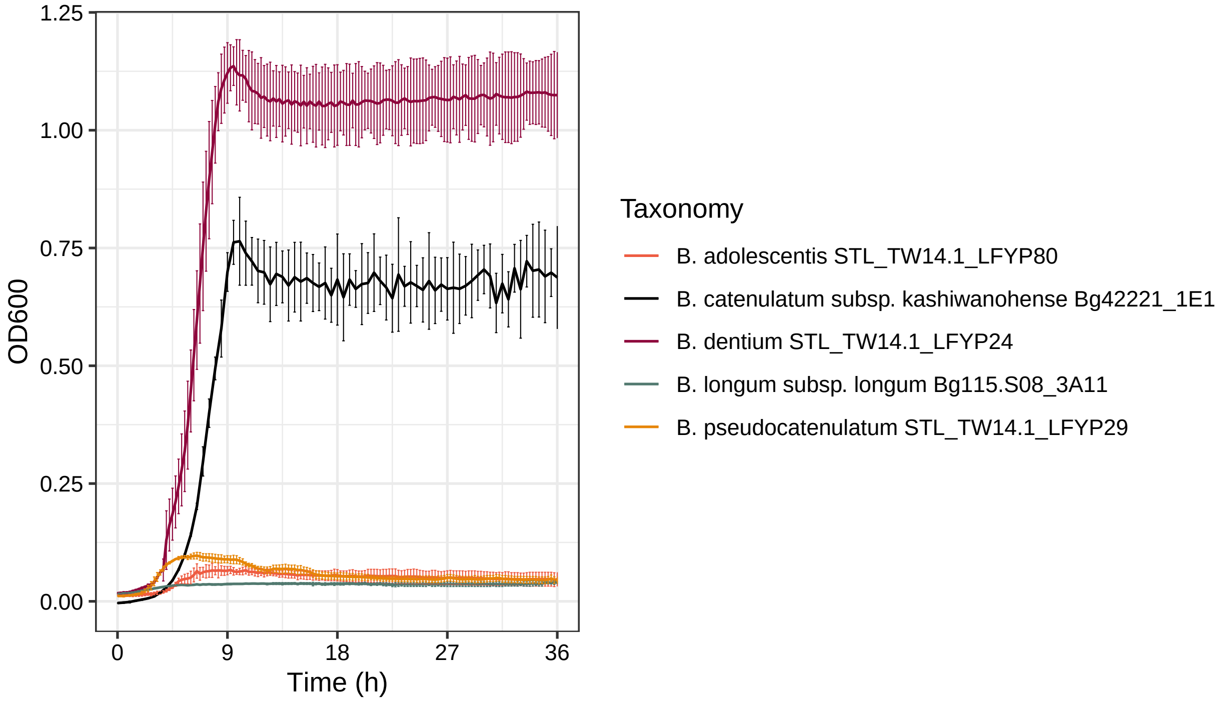


```
# save the plot as pdf
ggsave(filename = FOS_output_file, width = 14, height = 4, units = "in", dpi = 300)
```

10.17.2 Growth in MRS-AC supplemented with 0.5% tamarind xyloglucan

```
# set input files
XGL_output_file <- "results/growth/XGL.pdf"
XGL_data <- "data/growth/formatted_selected/XGL/growth_xyloglucan.txt"
XGL_annotation <- "data/growth/formatted_selected/XGL/growth_xyloglucan_ann.txt"
XGL_trim_at_time <- 36 # measurements until [input] hours
XGL_blank_wells <- c("blank_1","blank_2","blank_3") # specify blank wells

# read a file with measurements
XGL_d <- read_tsv(XGL_data) %>%
  filter(time <= XGL_trim_at_time)
# calculate mean blank measurements (MRS-AC-Lac without added cells)
XGL_blank <- XGL_d %>%
  select(all_of(XGL_blank_wells)) %>%
  rowMeans()
# subtract blank measurements and round OD600 values to 3 digits
XGL_d_bl <- XGL_d %>%
  mutate(across(2:last_col(), ~ .x - XGL_blank)) %>%
  mutate(across(2:last_col(), ~ round(.x, 3)))
# read files with plate annotations
XGL_ann <- read_tsv(XGL_annotation)
XGL_d_bl_long <- XGL_d_bl %>%
  gather(., well, od, -time)
# create annotated tables
XGL_d_bl_long_ann <- left_join(XGL_d_bl_long, XGL_ann, by="well") %>%
  filter(carbon_source != "blank")
```

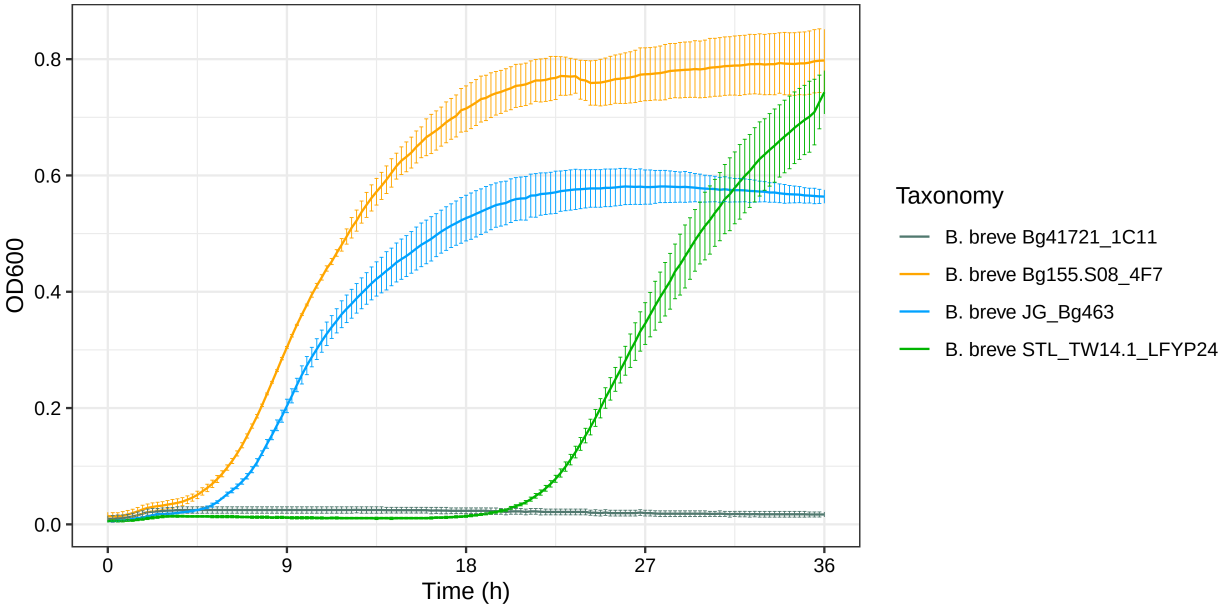
```
# plot growth curves
ggplot(XGL_d_bl_long_ann, aes(time, od, color = strain, fill = strain)) +
  #geom_point() +
  # add line connecting means from three replicates
  stat_summary(
    fun = mean,
    geom='line',
    size=0.5) +
  # add errorbars (standard deviation)
  stat_summary(
    fun.data=mean_sd,
    geom='errorbar',
    size=0.2,
    width=0.2,
    alpha=1) +
  scale_color_manual(name = "Taxonomy",
                     breaks=c("B. adolescentis STL_TW14.1_LFYP80",
                              "B. catenulatum subsp. kashiwanohense Bg42221_1E1",
                              "B. dentium STL_TW14.1_LFYP24",
                              "B. longum subsp. longum Bg115.S08_3A11",
                              "B. pseudocatenulatum STL_TW14.1_LFYP29"),
                     values=c("tomato2", "black", "#8e063c", "#51796f", "#e68607")) +
  scale_x_continuous(limits=c(0, XGL_trim_at_time),
                     breaks=c(0, XGL_trim_at_time*0.25, XGL_trim_at_time*0.5,
                              XGL_trim_at_time*0.75, XGL_trim_at_time)) +
  xlab("Time (h)") +
  ylab("OD600") +
  theme_bw() +
  theme(axis.text = element_text(color = "black"))
```



```
# save the plot as pdf
ggsave(filename = XGL_output_file, width = 7, height = 4, units = "in", dpi = 300)
```

10.17.3 Growth in MRS-AC supplemented with 1% N-acetylneuraminic acid.

```
# set input files
NANA_output_file <- "results/growth/Neu5Ac.pdf"
NANA_data <- "data/growth/formatted_selected/Neu5Ac/growth_sialic_acid.txt"
NANA_annotation <- "data/growth/formatted_selected/Neu5Ac/growth_sialic_acid_ann.txt"
NANA_trim_at_time <- 36 # measurements until [input] hours
NANA_blank_wells <- c("blank_1","blank_2","blank_3") # specify blank wells

# read a file with measurements
NANA_d <- read_tsv(NANA_data) %>%
  filter(time <= NANA_trim_at_time)
# calculate mean blank measurements (MRS-AC-Lac without added cells)
NANA_blank <- NANA_d %>%
  select(all_of(NANA_blank_wells)) %>%
  rowMeans()
# subtract blank measurements and round OD600 values to 3 digits
NANA_d_bl <- NANA_d %>%
  mutate(across(2:last_col(), ~ .x - NANA_blank)) %>%
  mutate(across(2:last_col(), ~ round(.x, 3)))
# read files with plate annotations
NANA_ann <- read_tsv(NANA_annotation)
NANA_d_bl_long <- NANA_d_bl %>%
  gather(., well, od, -time)
# create annotated tables
NANA_d_bl_long_ann <- left_join(NANA_d_bl_long, NANA_ann, by="well") %>%
  filter(carbon_source != "blank")
# plot growth curves
ggplot(NANA_d_bl_long_ann, aes(time, od, color = strain, fill = strain)) +
  # add line connecting means from three replicates
  stat_summary(
    fun = mean,
    geom='line',
    size=0.5) +
  # add errorbars (standard deviation)
  stat_summary(
    fun.data=mean_sd,
    geom='errorbar',
    size=0.2,
    width=0.2,
    alpha=1) +
  scale_color_manual(name = "Taxonomy",
                     breaks=c("B. breve Bg41721_1C11",
                              "B. breve Bg155.S08_4F7",
                              "B. breve JG_Bg463",
                              "B. breve STL_TW14.1_LFYP24"),
                     values=c("#51796f", "#ffa600", "#00a2ff", "#00b400")) +
  scale_x_continuous(limits=c(0, NANA_trim_at_time),
                   breaks=c(0, NANA_trim_at_time*0.25, NANA_trim_at_time*0.5,
                                    NANA_trim_at_time*0.75, NANA_trim_at_time)) +
  xlab("Time (h)") +
  ylab("OD600") +
  theme_bw() +
  theme(axis.text = element_text(color = "black"))
```

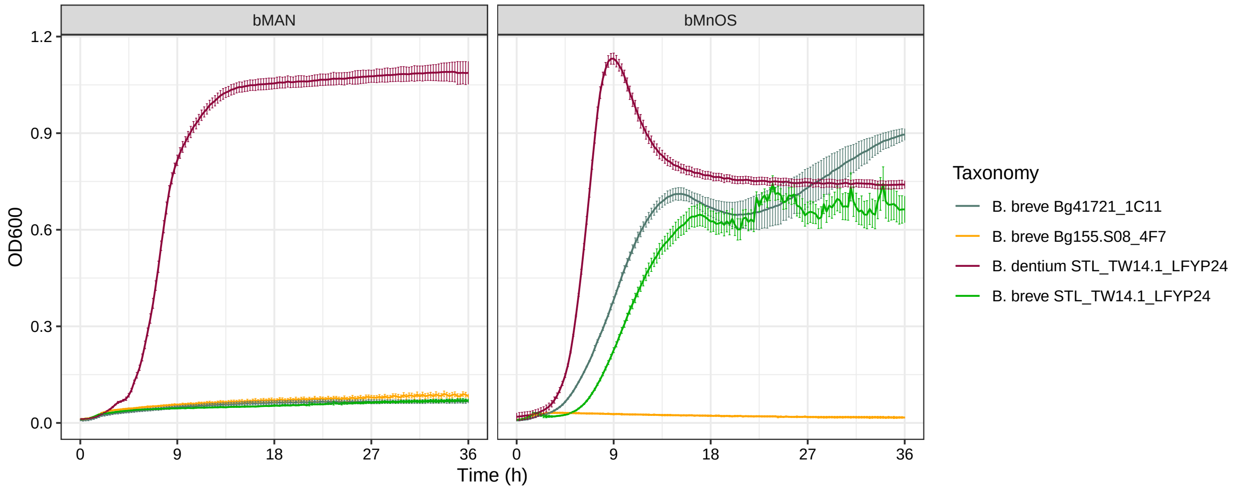


```
# save the plot as pdf
ggsave(filename = NANA_output_file, width = 8, height = 4, units = "in", dpi = 300)
```

10.17.4 Growth in MRS-AC supplemented with 0.5% mannotriose (bMnOS) or 0.5% konjac glucomannan (bMAN).

```
bMAN_output_file <- "results/growth/bMAN.pdf"
bMAN_data <- "data/growth/formatted_selected/bMAN/growth_bMAN.txt"
bMAN_annotation <- "data/growth/formatted_selected/bMAN/growth_bMAN_ann.txt"
bMAN_trim_at_time <- 36 # measurements until [input] hours
bMAN_blank_wells <- c("blank_1","blank_2","blank_3") # specify blank wells

# read a file with measurements
bMAN_d <- read_tsv(bMAN_data) %>%
  filter(time <= bMAN_trim_at_time)
# calculate mean blank measurements (MRS-AC-Lac without added cells)
bMAN_blank <- bMAN_d %>%
  select(all_of(bMAN_blank_wells)) %>%
  rowMeans()
# subtract blank measurements and round OD600 values to 3 digits
bMAN_d_bl <- bMAN_d %>%
  mutate(across(2:last_col(), ~ .x - bMAN_blank)) %>%
  mutate(across(2:last_col(), ~ round(.x, 3)))
# read files with plate annotations
```

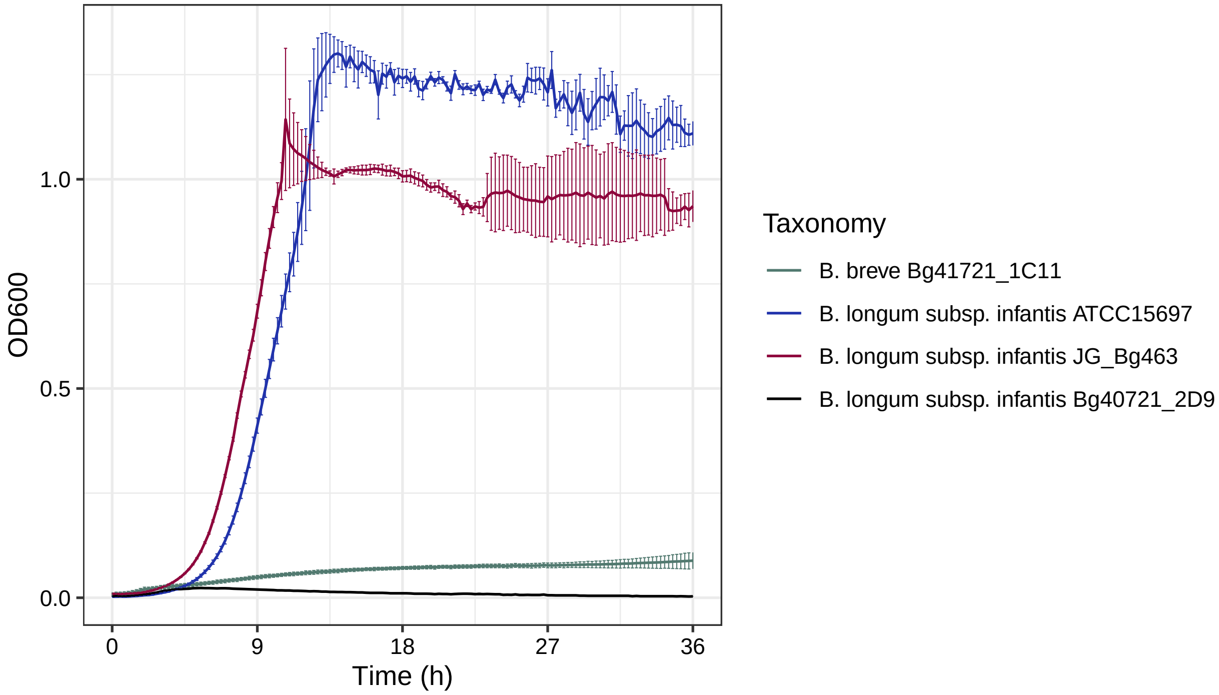
```r
bMAN_ann <- read_tsv(bMAN_annotation)
bMAN_d_bl_long <- bMAN_d_bl %>%
  gather(., well, od, -time)
# create annotated tables
bMAN_d_bl_long_ann <- left_join(bMAN_d_bl_long, bMAN_ann, by="well") %>%
  filter(carbon_source != "blank")
# plot growth curves
ggplot(bMAN_d_bl_long_ann, aes(time, od, color = strain, fill = strain)) +
  # add line connecting means from three replicates
  stat_summary(
    fun = mean,
    geom='line',
    size=0.5) +
  # add errorbars (standard deviation)
  stat_summary(
    fun.data=mean_sd,
    geom='errorbar',
    size=0.2,
    width=0.2,
    alpha=1) +
  scale_color_manual(name = "Taxonomy",
                     breaks=c("B. breve Bg41721_1C11",
                              "B. breve Bg155.S08_4F7",
                              "B. dentium STL_TW14.1_LFYP24",
                              "B. breve STL_TW14.1_LFYP24"),
                     values=c("#51796f", "#ffa600", "#8e063c", "#00b400")) +
  scale_x_continuous(limits=c(0, bMAN_trim_at_time),
                     breaks=c(0, bMAN_trim_at_time*0.25, bMAN_trim_at_time*0.5,
                              bMAN_trim_at_time*0.75, bMAN_trim_at_time)) +
  xlab("Time (h)") +
  ylab("OD600") +
  facet_wrap(~carbon_source) +
  theme_bw() +
  theme(axis.text = element_text(color = "black"))
```



```
# save the plot as pdf
ggsave(filename = bMAN_output_file, width = 10, height = 4, units = "in", dpi = 300)
```

10.17.5 Growth in MRS-AC supplemented with 1% D-glucuronic acid.

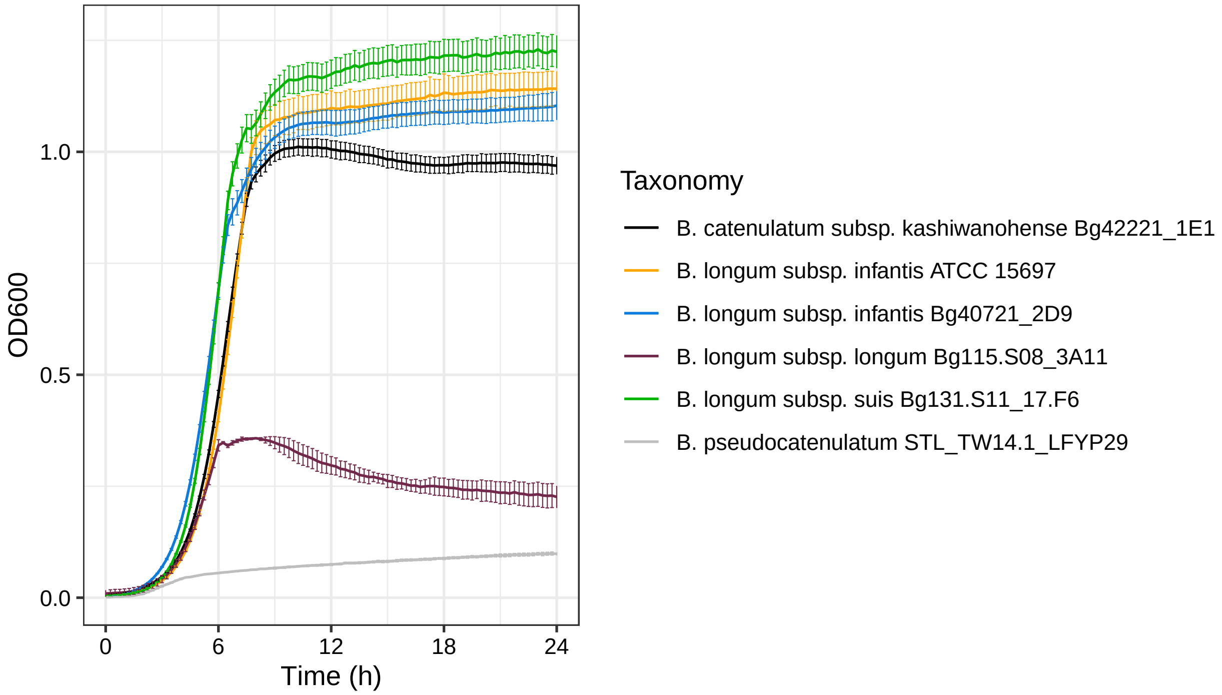
```
GlcA_output_file <- "results/growth/GlcA.pdf"
GlcA_data <- "data/growth/formatted_selected/GlcA/growth_GlcA.txt"
GlcA_annotation <- "data/growth/formatted_selected/GlcA/growth_GlcA_ann.txt"
GlcA_trim_at_time <- 36 # measurements until [input] hours
GlcA_blank_wells <- c("blank_1","blank_2","blank_3") # specify blank wells

# read a file with measurements
GlcA_d <- read_tsv(GlcA_data) %>%
  filter(time <= GlcA_trim_at_time)
# calculate mean blank measurements (MRS-AC-Lac without added cells)
GlcA_blank <- GlcA_d %>%
  select(all_of(GlcA_blank_wells)) %>%
  rowMeans()
# subtract blank measurements and round OD600 values to 3 digits
GlcA_d_bl <- GlcA_d %>%
  mutate(across(2:last_col(), ~ .x - GlcA_blank)) %>%
  mutate(across(2:last_col(), ~ round(.x, 3)))
# read files with plate annotations
GlcA_ann <- read_tsv(GlcA_annotation)
GlcA_d_bl_long <- GlcA_d_bl %>%
  gather(., well, od, -time)
# create annotated tables
GlcA_d_bl_long_ann <- left_join(GlcA_d_bl_long, GlcA_ann, by="well") %>%
  filter(carbon_source != "blank")
# plot growth curves
ggplot(GlcA_d_bl_long_ann, aes(time, od, color = strain, fill = strain)) +
  # add line connecting means from three replicates
  stat_summary(
    fun = mean,
    geom='line',
    size=0.5) +
  # add errorbars (standard deviation)
  stat_summary(
    fun.data=mean_sd,
    geom='errorbar',
    size=0.2,
    width=0.2,
    alpha=1) +
  scale_color_manual(name = "Taxonomy",
                     breaks=c("B. breve Bg41721_1C11",
                              "B. longum subsp. infantis ATCC15697",
                              "B. longum subsp. infantis JG_Bg463",
                              "B. longum subsp. infantis Bg40721_2D9"),
                     values=c("#51796f", "#2032ab", "#8e063c", "black")) +
  scale_x_continuous(limits=c(0, GlcA_trim_at_time),
                     breaks=c(0, GlcA_trim_at_time*0.25, GlcA_trim_at_time*0.5,
                              GlcA_trim_at_time*0.75, GlcA_trim_at_time)) +
  xlab("Time (h)") +
  ylab("OD600") +
  theme_bw() +
  theme(axis.text = element_text(color = "black"))
```



```
# save the plot as pdf
ggsave(filename = GlcA_output_file, width = 7, height = 4, units = "in", dpi = 300)
```

10.17.6 Growth in MRS-AC supplemented with 1% HMOs from pooled human milk (pHMOs); first experiment (24 h)

```
# load libraries
library(tidyverse)
library(ggpubr)
# read_data
pHMO_output_file <- "results/growth/pHMO_24h.pdf"
pHMO_data <- "data/growth/formatted_selected/pHMO/growth_pHMO_24h.txt"
pHMO_annotation <- "data/growth/formatted_selected/pHMO/growth_pHMO_24h_ann.txt"
trim_at_time <- 24 # measurements until [input] hours
blank_wells_HMO <- c("blank_1","blank_2","blank_3") # specify blank wells

# read a file with measurements
d_HMO <- read_tsv(pHMO_data) %>%
  filter(time <= trim_at_time)
# calculate mean blank measurements (MRS-AC-Lac without added cells)
blank_HMO <- d_HMO %>%
  dplyr::select(all_of(blank_wells_HMO)) %>%
  rowMeans()
# subtract blank measurements and round OD600 values to 3 digits
d_bl_HMO <- d_HMO %>%
  mutate(across(2:last_col(), ~ .x - blank_HMO)) %>%
  mutate(across(2:last_col(), ~ round(.x, 3)))
# read files with plate annotations
ann <- read_tsv(pHMO_annotation)
d_bl_long_HMO <- d_bl_HMO %>%
  gather(., well, od, -time)
# create annotated tables
d_bl_long_HMO_ann <- left_join(d_bl_long_HMO, ann, by="well") %>%
  filter(carbon_source != "blank") %>%
  filter(carbon_source != "Lac")

# plot growth curves
ggplot(d_bl_long_HMO_ann, aes(time, od, color = strain, fill = strain)) +
  #geom_point() +
  # add line connecting means from three replicates
  stat_summary(
    fun = mean,
    geom='line',
    size=0.5) +
  # add errorbars (standard deviation)
  stat_summary(
    fun.data=mean_sd,
    geom='errorbar',
    size=0.2,
    width=0.2,
    alpha=1) +
  scale_color_manual(name = "Taxonomy",
                     breaks=c("B. catenulatum subsp. kashiwanohense Bg42221_1E1",
                              "B. longum subsp. infantis ATCC 15697",
                              "B. longum subsp. infantis Bg40721_2D9",
                              "B. longum subsp. longum Bg115.S08_3A11",
                              "B. longum subsp. suis Bg131.S11_17.F6",
                              "B. pseudocatenulatum STL_TW14.1_LFYP29"),
                     values=c("black", "#ffa600","#0C7BDC",
                              "#71284a", "#00b400", "grey")) +
scale_x_continuous(limits=c(0, trim_at_time),
                   breaks=c(0, trim_at_time*0.25, trim_at_time*0.5,
                            trim_at_time*0.75, trim_at_time)) +
  xlab("Time (h)") +
  ylab("OD600") +
  theme_bw() +
  theme(axis.text = element_text(color = "black"))
```



```
# save the plot as pdf
ggsave(filename = pHMO_output_file, width = 7, height = 4, units = "in", dpi = 300)
```

10.18 Summary of growth data

The code chunks below describe the comparison of predicted binary carbohydrate utilization phenotypes with *in vitro* growth data obtained in this work or found in literature (Bottacini et al., 2014 and Arboleya et al., 2018).

Here we compare predicted binary phenotypes (**1** or **0**) with growth phenotypes (**+**/**w** and **-**) and calculate the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). We then calculate various metrics (precision, recall, specificity, accuracy, F-score, and Matthew's correlation coefficient) based on TP, TN, FP, and FN counts.

```
# define a function that will calculate various metrics based on TP, TN, FP, and FN counts
compute_metrics <- function(df) {
  df <- df %>%
    mutate(
      Precision = round(TruePositive / (TruePositive + FalsePositive), 2),
      Recall = round(TruePositive / (TruePositive + FalseNegative), 2),
      Specificity = round(TrueNegative / (TrueNegative + FalsePositive), 2),
      Accuracy = round((TruePositive + TrueNegative) /
                         (TruePositive + TrueNegative + FalsePositive + FalseNegative), 2),
      F_score = round(2 * (Precision * Recall) / (Precision + Recall), 2),
      MCC = round((TruePositive * TrueNegative - FalsePositive * FalseNegative) /
        sqrt((TruePositive + FalsePositive) *
               (TruePositive + FalseNegative) * (TrueNegative + FalsePositive) *
               (TrueNegative + FalseNegative)), 2)
    )
```

```r
  return(df)
}

# read files
# BPM and the summary of growth data for 16 strains from this work
bpm_ar <- read_tsv("data/growth/summary/bif_16_strains_BPM.txt")
growth_ar <- read_tsv("data/growth/summary/bif_16_strains_growth_data.txt")
# BPM and the summary of growth data for 6 strains from Bottacini et al., 2014
bpm_bot <- read_tsv("data/growth/summary/Bottacini_2014_BPM.txt")
growth_bot <- read_tsv("data/growth/summary/Bottacini_2014_growth_data.txt")
# BPM and the summary of growth data for 19 strains from Arboleya et al., 2018
bpm_arb <- read_tsv("data/growth/summary/Arboleya_2018_BPM.txt")
growth_arb <- read_tsv("data/growth/summary/Arboleya_2018_growth_data.txt")
# phenotype metadata
phenotype_metadata_gr <- read_tsv("data/tables/phenotype_metadata_carbs.txt") %>%
  dplyr::select(phenotype, type_group)

# calculate the number TP, TN, FP, and FN counts
# growth data from this work
results_ar <- calculate_metrics_within_groups(bpm_ar, growth_ar, phenotype_metadata_gr)
results_ar_df <- bind_rows(results_ar, .id = "Group")
# growth data from Bottacini et al., 2014
results_bot <- calculate_metrics_within_groups(bpm_bot, growth_bot, phenotype_metadata_gr)
results_bot_df <- bind_rows(results_bot, .id = "Group")
# growth data from Arboleya et al., 2018
results_arb <- calculate_metrics_within_groups(bpm_arb, growth_arb, phenotype_metadata_gr)
results_arb_df <- bind_rows(results_arb, .id = "Group")
# combine TP, TN, FP, and FN count data: Bottacini et al., 2014 + Arboleya et al., 2018
desired_order <- c(
  "monosaccharides_and_derivatives",
  "di_and_oligosaccharides",
  "polysaccharides",
  "AllGroups"
)
results_papers_df <- bind_rows(results_bot_df, results_arb_df) %>%
  group_by(Group) %>%
  summarise(
    Tested = sum(Tested),
    TruePositive = sum(TruePositive),
    TrueNegative = sum(TrueNegative),
    FalsePositive = sum(FalsePositive),
    FalseNegative = sum(FalseNegative)
  ) %>%
  arrange(match(Group, desired_order))
```

Summary for data from Bottacini et al., 2014 and Arboleya et al., 2018:

```r
# calculate the metrics for TP, TN, FP, and FN counts
results_papers_df_m <- compute_metrics(results_papers_df)
# use the gt package to produce the table
gt(results_papers_df_m  ) %>%
  cols_align(
    align = "left",
    columns = everything()
  )
```

Group	Tested	TruePositive	TrueNegative	FalsePositive	FalseNegative	Precision	F
monosaccharides_and_derivatives	244	92	136	12	4	0.88	0
di_and_oligosaccharides	169	95	67	7	0	0.93	1
polysaccharides	126	70	52	4	0	0.95	1
AllGroups	539	257	255	23	4	0.92	0

Summary for data obtained in this work:

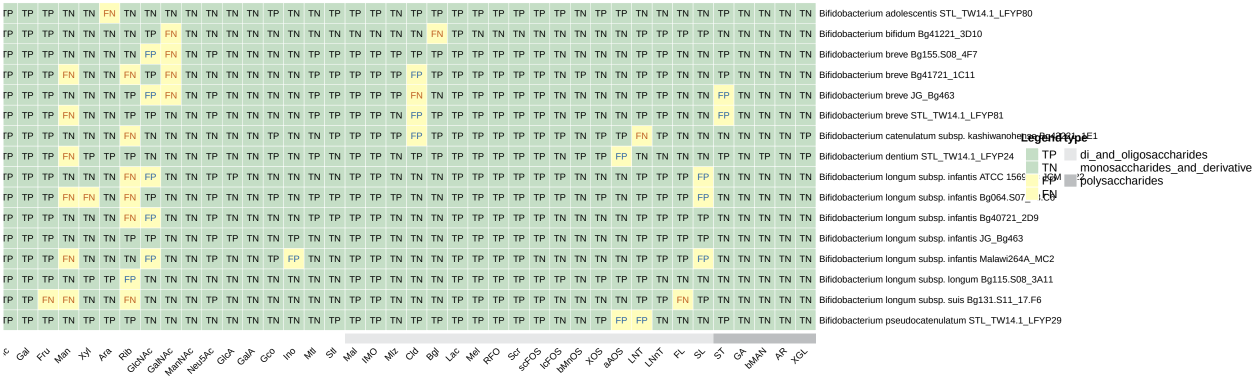
```
# calculate the metrics for TP, TN, FP, and FN counts
results_ar_df_m <- compute_metrics(results_ar_df)
# use the gt package to produce the table
gt(results_ar_df_m ) %>%
  cols_align(
    align = "left",
    columns = everything()
  )
```

Group	Tested	TruePositive	TrueNegative	FalsePositive	FalseNegative	Precision	F
monosaccharides_and_derivatives	272	97	149	7	19	0.93	0
di_and_oligosaccharides	288	172	103	9	4	0.95	0
polysaccharides	80	8	70	2	0	0.80	1
AllGroups	640	277	322	18	23	0.94	0

Plot the heatmap with results:

```
# Assuming bpm_ar, growth_ar, phenotype_metadata_gr are already defined
comparison_table <- bpm_ar %>%
  mutate(across(-genome_name, ~case_when(
    . == 1 & growth_ar[[cur_column()]] %in% c("+", "w") ~ "TP",
    . == 0 & growth_ar[[cur_column()]] %in% c("-") ~ "TN",
    . == 1 & growth_ar[[cur_column()]] %in% c("-") ~ "FP",
    . == 0 & growth_ar[[cur_column()]] %in% c("+", "w") ~ "FN",
    TRUE ~ NA_character_ # for any case that does not match the above
  )))

# Extract the binary matrix
comparison_table_mat <- as.matrix(comparison_table[, 2:ncol(comparison_table)])
# Add rownames to the matrix
rownames(comparison_table_mat) <- comparison_table$genome_name

# Extract vectors containing data about glycan type and origin
# Get the column names of the matrix, excluding the first one
matrix_column_names <- colnames(comparison_table_mat)

# Find the matching indices of these names in the phenotype column of phenotype_metadata_67
matching_indices <- match(matrix_column_names, phenotype_metadata_gr$phenotype)
```

```
# Use the indices to get the corresponding type values
glycan_type_vector <- phenotype_metadata_gr$type_group[matching_indices]

# Create a coloring function
comp_col_fun <- structure(c("#C5DEC6", "#C5DEC6", "#fffdbc", "#fffdbc"),
                          names = c("TP", "TN", "FP", "FN"))

# Create a column annotation specifying glycan type
comp_ha_ann <- HeatmapAnnotation(
  type = glycan_type_vector,
  col = list(type = c("monosaccharides_and_derivatives" = "#FFFFFF",
                      "di_and_oligosaccharides" = "#E6E7E8",
                      "polysaccharides" = "#BCBEC0")),
  show_annotation_name = FALSE,
  simple_anno_size = unit(3, "mm"))

# Plot the heatmap
pdf("results/growth/predictions_vs_growth.pdf", width=15, height=8)
ht_growth <- ComplexHeatmap::Heatmap(comparison_table_mat,
                              name = "Legend",
                              heatmap_legend_param = list(title_position = "topcenter"),
                              bottom_annotation = comp_ha_ann,
                              col = comp_col_fun,
                              cluster_rows = F,
                              cluster_columns = F,
                              rect_gp = gpar(col = "white", lwd = 0.5),
                              show_row_names = TRUE,
                              row_names_gp = gpar(fontsize = 8),
                              column_names_gp = gpar(fontsize = 8),
                              column_names_rot = 45,
                              width = unit(250, "mm"),
                              height = unit(100, "mm"),
                              cell_fun = function(j, i, x, y, width, height, fill) {
                                if (comparison_table_mat[i, j] == "FP") {
                                  grid.text(comparison_table_mat[i, j], x, y,
                                            gp = gpar(fontsize = 8, col = "#2c64a3"))
                                } else if (comparison_table_mat[i, j] == "FN") {
                                  grid.text(comparison_table_mat[i, j], x, y,
                                            gp = gpar(fontsize = 8, col = "#be5d20"))
                                } else {
                                  grid.text(comparison_table_mat[i, j], x, y,
                                            gp = gpar(fontsize = 8))
                                }
                              })
draw(ht_growth, heatmap_legend_side = "top", annotation_legend_side = "top")
# Save the heatmap to a file
invisible(dev.off())
# Draw the heatmap again to show it in the compiled markdown file
draw(ht_growth)
```



c	Gal	Fru	Man	Xyl	Ara	Rib	GlcNAc	GalNAc	ManNAc	Neu5Ac	GlcA	GalA	Gco	Ino	Mtl	Stl	Mal	IMO	Mlz	Cld	Bgl	Lac	Mel	RFO	Scr	scFOS	lcFOS	bMnOS	XOS	aAOS	LNT	LNnT	FL	SL	ST	GA	bMAN	AR	XGL	
TP	TP	TP	TN	TN	FN	TN	TN	TN	TN	TN	TN	TN	TP	TN	TN	TP	TP	TP	TN	TP	TP	TP	TP	TP	TP	TP	TP	TN	TP	TP	TN	TN	TN	TN	TP	TN	TN	TN	TN	Bifidobacterium adolescentis STL_TW14.1_LFYP80
TP	TP	TP	TN	TN	TN	TN	TP	FN	TN	TN	TN	TN	TN	TN	TN	TN	TN	TN	TN	TN	FN	TP	TN	TN	TN	TN	TN	TN	TN	TN	TP	TP	TP	TP	TN	TN	TN	TN	TN	Bifidobacterium bifidum Bg41221_3D10
TP	TP	TP	TN	TN	TN	TN	FP	FN	TN	TP	TN	TN	TN	TN	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP	TN	TN	TN	TN	TP	TP	TN	TN	TP	TN	TN	TN	TN	Bifidobacterium breve Bg155.S08_4F7
TP	TP	TP	FN	TN	TN	FN	TP	FN	TN	TN	TN	TN	TN	TN	TP	TP	TP	TP	TP	FP	TP	TP	TP	TP	TP	TP	TN	TP	TN	TN	TP	TP	TN	TN	TP	TN	TN	TN	TN	Bifidobacterium breve Bg41721_1C11
TP	TP	TP	TN	TN	TN	TP	FP	FN	TN	TP	TN	TN	TN	TN	TP	TP	TP	TP	TP	FN	TN	TP	TP	TP	TP	TP	TN	TN	TN	TN	TP	TP	TN	TN	FP	TN	TN	TN	TN	Bifidobacterium breve JG_Bg463
TP	TP	TP	FN	TN	TN	TP	TP	TN	TN	TP	TN	TN	TN	TN	TP	TP	TP	TP	TN	FP	TP	TP	TP	TP	TP	TP	TN	TP	TN	TN	TP	TP	TN	TN	FP	TN	TN	TN	TN	Bifidobacterium breve STL_TW14.1_LFYP81
TP	TP	TP	TN	TN	TN	FN	TN	TN	TN	TN	TN	TN	TP	TN	TN	TP	TP	TP	TP	FP	TP	TP	TP	TP	TP	TP	TP	TN	TP	TP	FN	TP	TN	TN	TN	TN	TN	TN	TP	Bifidobacterium catenulatum subsp. kashiwanohense Bg42221_1E1
TP	TP	TP	FN	TP	TP	TP	TN	TN	TN	TP	TN	TN	TP	TN	TP	TN	TP	TP	TN	TP	TP	TP	TP	TP	TP	TP	TN	TP	TP	FP	TN	TN	TN	TN	TP	TN	TP	TN	TP	Bifidobacterium dentium STL_TW14.1_LFYP24
TP	TP	TP	TN	TN	TN	FN	FP	TN	TN	TP	TP	TN	TN	TN	TN	TN	TP	TP	TN	TN	TN	TP	TP	TP	TP	TP	TP	TN	TN	TN	TP	TP	TP	FP	TN	TN	TN	TN	TN	Bifidobacterium longum subsp. infantis ATCC 1569[illegible] JCM [illegible]2
TP	TP	TP	FN	FN	TN	FN	TP	TN	TN	TP	TN	TN	TP	TN	TP	TP	TP	TP	TP	TN	TN	TP	TP	TP	TP	TP	TP	TN	TP	TN	TP	TP	TP	FP	TN	TN	TN	TN	TN	Bifidobacterium longum subsp. infantis Bg064.S07_[illegible].C6
TP	TP	TP	TN	TN	TN	FN	FP	TN	TN	TP	TN	TN	TN	TN	TP	TP	TP	TP	TN	TN	TN	TP	TP	TP	TP	TP	TN	TN	TN	TN	TP	TP	TP	TP	TN	TN	TN	TN	TN	Bifidobacterium longum subsp. infantis Bg40721_2D9
TP	TP	TP	TN	TN	TN	TP	TP	TN	TN	TP	TP	TN	TN	TP	TN	TN	TP	TP	TN	TN	TN	TP	TP	TP	TP	TP	TN	TN	TN	TN	TP	TP	TP	TP	TN	TN	TN	TN	TN	Bifidobacterium longum subsp. infantis JG_Bg463
TP	TP	TP	FN	TN	TN	TN	FP	TN	TN	TP	TN	TN	TP	FP	TN	TN	TP	TP	TN	TN	TN	TP	TP	TP	TP	TN	TN	TN	TN	TN	TP	TP	TP	FP	TN	TN	TN	TN	TN	Bifidobacterium longum subsp. infantis Malawi264A_MC2
TP	TP	TP	TN	TP	TP	FP	TN	TN	TN	TN	TN	TN	TN	TN	TN	TN	TP	TP	TN	TN	TN	TP	TP	TP	TP	TP	TP	TN	TP	TP	TP	TN	TN	TN	TN	TN	TN	TN	TN	Bifidobacterium longum subsp. longum Bg115.S08_3A11
TP	TP	FN	FN	TN	TN	FN	TN	TN	TN	TP	TN	TN	TN	TN	TN	TN	TP	TP	TN	TN	TN	TP	TP	TP	TP	TP	TN	TN	TN	TN	TP	TP	FN	TP	TN	TN	TN	TN	TN	Bifidobacterium longum subsp. suis Bg131.S11_17.F6
TP	TP	TP	TN	TP	TP	TP	TN	TN	TN	TN	TN	TN	TN	TN	TN	TP	TP	TP	TN	TP	TP	TP	TP	TP	TP	TP	TN	TN	TP	FP	FP	TN	TN	TN	TP	TN	TN	TN	TN	Bifidobacterium pseudocatenulatum STL_TW14.1_LFYP29

11 Analysis of HMO consumption data

11.1 Introduction

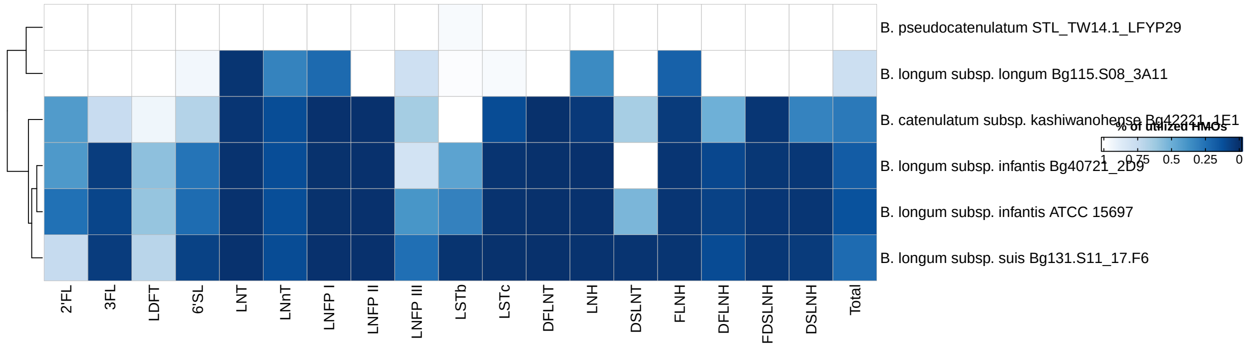
This block describes the analysis of tables with information about the concentration of individual HMOs (nmol/mL) in culture supernatants of select *Bifidobacterium* strains grown in MRS-AC-pHMO for (a) 8 hours or (b) 24 hours. The supernatants of two different timepoints were collected in two separate experiments

11.2 Load data

```
# read data
pHMO_data_8h <- "data/hmo_consumption/HMO_consumption_8h.txt"
pHMO_data_24h <- "data/hmo_consumption/HMO_consumption_24h.txt"
```

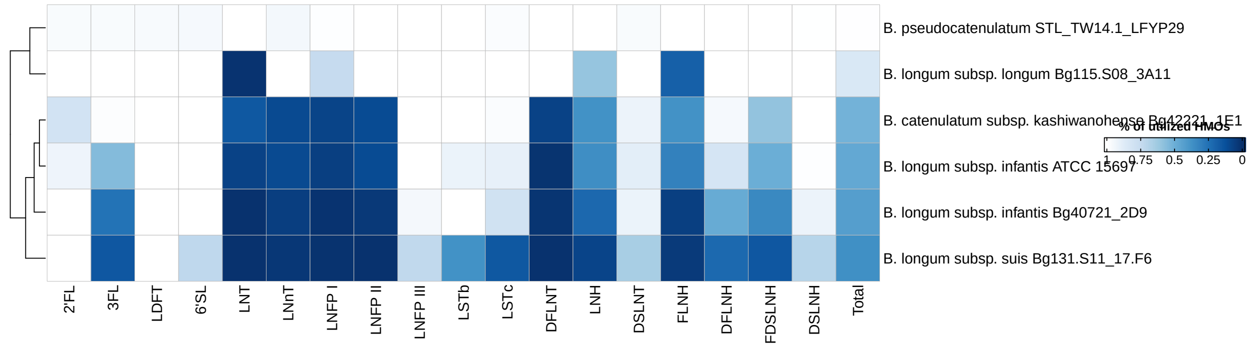
11.3 First experiment (24 h)

```
# read files with measurements
HMO_24h <- read_tsv(pHMO_data_24h) %>%
  mutate(across(3:last_col(), ~ round(.x, 1))) %>%
  rowwise() %>%
  mutate(Total = sum(c_across(3:last_col())))
# calculate average values
HMO_24h_avg <- HMO_24h %>%
  group_by(strain) %>%
  summarize(across(where(is.numeric), mean, na.rm = TRUE)) %>%
  mutate(across(2:last_col(), ~ round(.x, 1)))
# calculate percent of utilized HMOs
HMO_24h_ut <- HMO_24h_avg %>%
  mutate(across(-strain, ~ . / .[strain == "medium"])) %>%
  mutate(across(2:last_col(), ~ round(.x, 3))) %>%
  filter(strain != "medium")
# extract matrices
hmo_mat_24h <- as.matrix(HMO_24h_ut[,2:ncol(HMO_24h_ut)])
rownames(hmo_mat_24h) <- pull(HMO_24h_ut, strain)

# add a coloring function
col_fun <- colorRamp2(c(1, 0.875, 0.75, 0.625, 0.5, 0.375, 0.25, 0.125, 0),
                      c("#ffffff", "#DEEBF7", "#C6DBEF", "#9ECAE1", "#6BAED6",
                        "#4292C6", "#2171B5", "#08519C", "#08306B"))
# modify the legend 1, 0.875, 0.75, 0.625, 0.5, 0.375, 0.25, 0.125, 0
lgd <- Legend(col_fun = col_fun,
              title = "% of predicted utilizers",
              direction = "horizontal",
              title_position = "lefttop")
pdf("results/HMO_utilization/heatmap_24h.pdf", width=15, height=4)
ht_hmo_24h <- ComplexHeatmap::Heatmap(hmo_mat_24h,
                                     name = "HMO consumption at 8h",
                                     col = col_fun,
                                     cluster_rows = TRUE,
                                     cluster_columns = FALSE,
                                     rect_gp = gpar(col = "grey", lwd = 0.05),
                                     heatmap_legend_param = list(
                                       col_fun = col_fun,
                                       at = c(1, 0.75, 0.5, 0.25, 0),
                                       title = "% of utilized HMOs",
                                       direction = "horizontal",
                                       title_position = "topcenter",
                                       border = "black",
                                       legend_width = unit(40, "mm"))
)
draw(ht_hmo_24h)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ht_hmo_24h)
```



11.4 Second experiment (8 h)

```
# read files with measurements
HMO_8h <- read_tsv(pHMO_data_8h) %>%
  mutate(across(3:last_col(), ~ round(.x, 1))) %>%
  rowwise() %>%
  mutate(Total = sum(c_across(3:last_col())))
# calculate average values
HMO_8h_avg <- HMO_8h %>%
  group_by(strain) %>%
  summarize(across(where(is.numeric), mean, na.rm = TRUE)) %>%
  mutate(across(2:last_col(), ~ round(.x, 1)))
# calculate percent of utilized HMOs
HMO_8h_ut <- HMO_8h_avg %>%
  mutate(across(-strain, ~ . / .[strain == "medium"])) %>%
  mutate(across(2:last_col(), ~ round(.x, 3))) %>%
  filter(strain != "B. breve Bg155.S08_4F7") %>%
  filter(strain != "medium")
# extract matrices
hmo_mat_8h <- as.matrix(HMO_8h_ut[,2:ncol(HMO_8h_ut)])
rownames(hmo_mat_8h) <- pull(HMO_8h_ut, strain)

# add a coloring function
col_fun <- colorRamp2(c(1, 0.875, 0.75, 0.625, 0.5, 0.375, 0.25, 0.125, 0),
                      c("#ffffff", "#DEEBF7", "#C6DBEF", "#9ECAE1", "#6BAED6",
                        "#4292C6", "#2171B5", "#08519C", "#08306B"))
# modify the legend 1, 0.875, 0.75, 0.625, 0.5, 0.375, 0.25, 0.125, 0
lgd <- Legend(col_fun = col_fun,
              title = "% of predicted utilizers",
              direction = "horizontal",
              title_position = "lefttop")

# specify the name of the output file
pdf("results/HMO_utilization/heatmap_8h.pdf", width=15, height=4)
ht_hmo_8h <- ComplexHeatmap::Heatmap(hmo_mat_8h,
                        name = "HMO consumption at 8h",
                        col = col_fun,
                        cluster_rows = TRUE,
                        cluster_columns = FALSE,
                        rect_gp = gpar(col = "grey", lwd = 0.05),
                        heatmap_legend_param = list(
                          col_fun = col_fun,
                          at = c(1, 0.75, 0.5, 0.25, 0),
                          title = "% of utilized HMOs",
                          direction = "horizontal",
                          title_position = "topcenter",
                          border = "black",
                          legend_width = unit(40, "mm"))
                        )
draw(ht_hmo_8h)
# save the heatmap to a file
invisible(dev.off())
# draw the heatmap again to show it in the compiled markdown file
draw(ht_hmo_8h)
```



12 Analysis of RNA-seq data

12.1 Processing of raw FASTQ files and read mapping

The code chuck below describes the processing of raw FASTQ files and mapping reads to the *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg42221_1E1 transcriptome. The following software is required (can be installed in one conda/mamba environment):

1. FastQC (v0.11.9)
2. Cutadapt (v4.1)
3. Bowtie2 (v 2.4.5)
4. Kallisto (v0.48)
5. MultiQC (v1.13)
6. Parallel (v20220722)

Note: due to size limitation, raw FASTQ files could not be stored in the GitHub repo. Thus, you will need to download the FASTQ files from the Gene Expression Omnibus, under **accession**. Put downloaded fastq.gz files to `data/rnaseq/fastq/`.

The reference FASTA files used for building Bowtie2 and Kallisto indeces can be found in `data/rnaseq/refs/`.

Note: exact file names (without the .fastq.qz extension) should be entered into `data/rnaseq/runids.txt`.

Alternatively, you can run `code/qc_readmapping.sh` instead of the code chunk below.

Summary of the script:

1. Quality control of raw reads was carried out using FastQC
2. Quality trimming and removal of Illumina sequencing adapters and short reads (< 20 bp) were performed via Cutadapt
3. Reads were mapped against rRNA and tRNA gene sequences extracted from the *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg42221_1E1 genome (GenBank accession no.) using Bowtie2. Unmapped (filtered) reads were saved and used further
4. Filtered reads were mapped to the *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg_1E1 transcriptome using Kallisto
5. The quality of raw/filtered reads, as well as the results of Bowtie2/Kallisto mapping were summarized in `data/rnaseq/multiqc_report.html` generated via MultiQC

```bash
source ~/.bash_profile
echo $BASH_VERSION
#set -ex

####################
## SOFTWARE SETUP ##
####################
# required tools: fastqc (v0.11.9), cutadapt (v4.1), bowtie2 (v2.4.5), kallisto (v0.48)
# multiqc (v1.13), and parallel (v20220722)
# set the name of the environment with the tools
environment_name="transcriptomics"
# activate conda environment with the required tools
eval "$(command conda 'shell.bash' 'hook' 2> /dev/null)" # initializes conda in sub-shell
conda activate ${environment_name}
conda info|egrep "conda version|active environment"

################
## USER INPUT ##
################
# fastq files should be in data/rnaseq/fastq/
# txt file containing names of fastq files (without the fastq.gz extension)
sample_names="data/rnaseq/runids.txt"
# fasta file containing sequences of rRNA and tRNA genes
rRNA_tRNA_fasta="data/rnaseq/refs/Bc_kashiwanohense_Bg42221_1E1_RNA.fasta"
# fasta file containing the whole transcriptome
transcriptome_fasta="data/rnaseq/refs/Bc_kashiwanohense_Bg42221_1E1.fasta"
# name of the bowtie2 index
bowtie2_index_name="data/rnaseq/refs/Bc_kashiwanohense_Bg42221_1E1_rRNA_tRNA"
# name of the kallisto index
kallisto_index_name="data/rnaseq/refs/Bc_kashiwanohense_Bg42221_1E1_transcriptome.index"

# create directories
echo "Creating directories"
mkdir -p data/rnaseq/qc1 # qc results for raw reads
mkdir -p data/rnaseq/qc2 # qc results for filtered reads
mkdir -p data/rnaseq/fq_trim # trimmed reads
mkdir -p data/rnaseq/fq_filt # filtered reads
mkdir -p data/rnaseq/sam # sam files produced during bowtie2 alignment; will be deleted
mkdir -p data/rnaseq/kallisto # kallisto mapping results

# run fastqc on raw reads
echo "Running FastQC on raw reads"
cat ${sample_names} | parallel "fastqc \
data/rnaseq/fastq/{}.fastq.gz \
--outdir data/rnaseq/qc1"

# trim adapters with cutadapt
echo "Trimming adapters with Cutadapt"
cat ${sample_names} | parallel "cutadapt \
--nextseq-trim=20 \
-m 20 \
-a AGATCGGAAGAGCACACGTCTGAACTCCAGTC \
-o data/rnaseq/fq_trim/{}.fastq.gz data/rnaseq/fastq/{}.fastq.gz \
&> data/rnaseq/fq_trim/{}.fastq.qz.log"

# filter reads mapped to rRNA and tRNA
echo "Filtering reads that map to rRNA and tRNA using Bowtie2"
## build bowtie2 index
bowtie2-build ${rRNA_tRNA_fasta} \
${bowtie2_index_name}
## align reads via bowtie2; save ones that did not align to a separate file
cat ${sample_names} | \
parallel "bowtie2 -x ${bowtie2_index_name} \
-U data/rnaseq/fq_trim/{}.fastq.gz \
-S data/rnaseq/sam/{}.sam \
--un data/rnaseq/fq_filt/{}.fastq \
&> data/rnaseq/fq_filt/{}_bowtie2.log"
cat ${sample_names} | parallel "gzip data/rnaseq/fq_filt/{}.fastq"

# run fastqc on filtered reads
echo "Running FastQC on filtered reads"
cat ${sample_names} | parallel "fastqc \
data/rnaseq/fq_filt/{}.fastq.gz \
--outdir data/rnaseq/qc2"

# pseudolalign reads to transcriptome
echo "Mapping reads to the transcriptome via Kallisto"
## build kallisto index
kallisto index -i ${kallisto_index_name} \
${transcriptome_fasta}
## map reads to indexed reference via kallisto
cat ${sample_names} | parallel "kallisto quant \
-i ${kallisto_index_name} \
-o data/rnaseq/kallisto/{} \
--single \
--rf-stranded \
-l 150 \
-s 20 \
data/rnaseq/fq_filt/{}.fastq.gz \
&> data/rnaseq/kallisto/{}_kallisto.log"
```

```
# run multiqc
echo "Running MultiQC"
export LC_ALL=en_US.utf-8
export LANG=en_US.utf-8
multiqc -d data/rnaseq -o data/rnaseq

# remove directories with intermediate files
echo "Removing directories"
rm -rf data/rnaseq/qc1
rm -rf data/rnaseq/qc2
rm -rf data/rnaseq/fq_trim
rm -rf data/rnaseq/fq_filt
rm -rf data/rnaseq/sam
```

12.2 Importing count data into R

TxImport was used to read Kallisto output into the R environment.

Note: before running the code, double-check that file names in the file_names column in `data/rnaseq/studydesign.txt` are identical to file names in `data/rnaseq/runids.txt`.

```
# read the study design file
targets <- read_tsv("data/rnaseq/studydesign.txt")
# set file paths to Kallisto output folders with quantification data
files <- file.path("data/rnaseq/kallisto", targets$file_name, "abundance.tsv")
# check that all output files are present
all(file.exists(files))
```

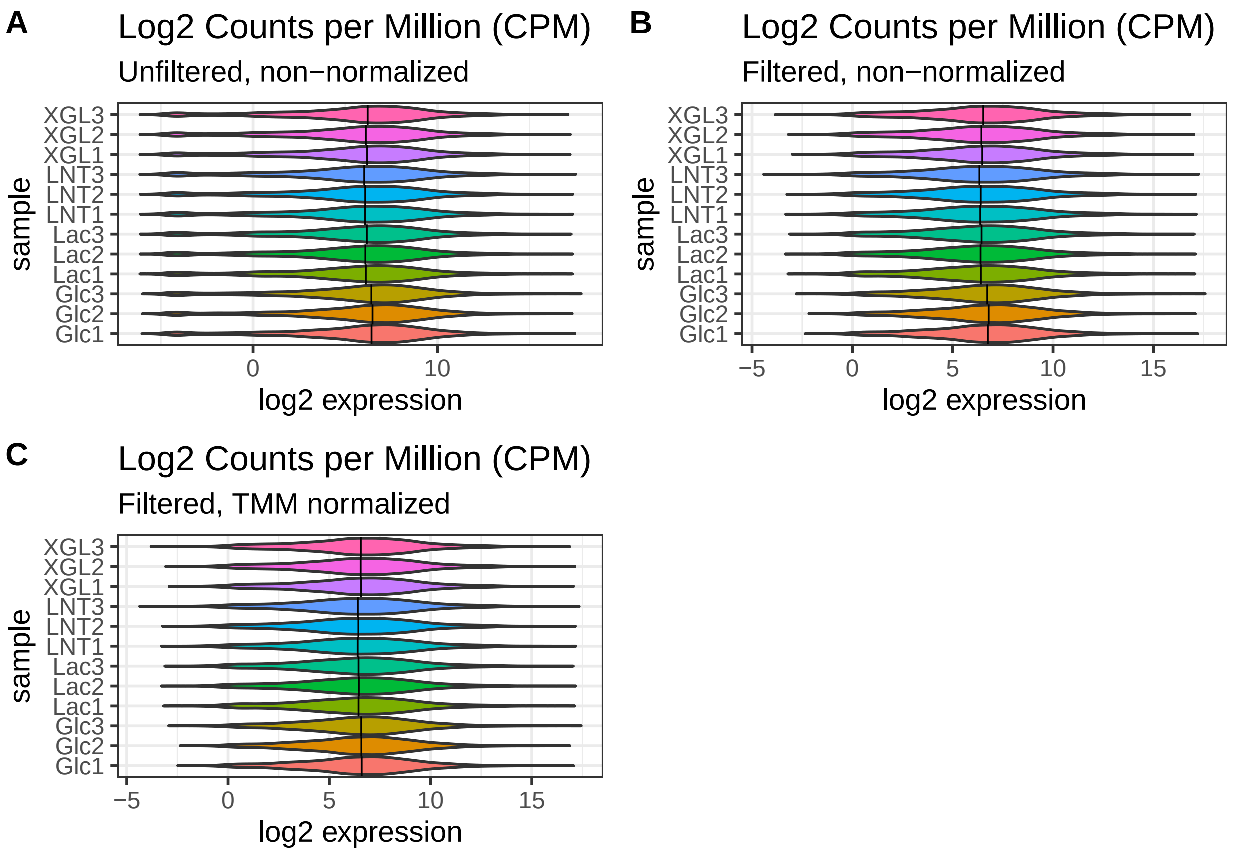
```
## [1] TRUE
```

```
# use 'tximport' to import Kallisto output into R
txi_kallisto <- tximport(files,
                         type = "kallisto",
                         txOut = TRUE, # import at transcript level
                         countsFromAbundance = "lengthScaledTPM")

# capture variables of interest from the study design
condition <- as.factor(targets$condition)
condition <- factor(condition, levels = c("Glc", "Lac", "LNT", "XGL"))
batch <- as.factor(targets$batch)
# capture sample labels for later use
sampleLabels <- targets$sample

# create a table with raw counts for GEO submission
raw_counts <- as_tibble(txi_kallisto$counts, rownames = "locus_tag")
colnames(raw_counts) <- c("geneID", sampleLabels)
write_tsv(raw_counts, "results/rnaseq/tables/Arzamasov_raw_count_matrix.txt")

# use the gt package to produce the study design table
gt(targets) %>%
  cols_align(
    align = "left",
    columns = everything()
  )
```

sample	file_name	condition	batch
Glc1	Glc1	Glc	1
Glc2	Glc2	Glc	1
Glc3	Glc3	Glc	1
Lac1	Lac1	Lac	2
Lac2	Lac2	Lac	2
Lac3	Lac3	Lac	2
LNT1	LNT1	LNT	2
LNT2	LNT2	LNT	2
LNT3	LNT3	LNT	2
XGL1	XGL1	XGL	1
XGL2	XGL2	XGL	1
XGL3	XGL3	XGL	1

12.3 Filtering and normalization

```
# extract counts
myDGEList <- DGEList(txi_kallisto$counts)
# plot unfiltered, non-normalized CPM
p1 <- profile(myDGEList, sampleLabels, "Unfiltered, non-normalized")
# filter counts
cpm <- cpm(myDGEList)
keepers <- rowSums(cpm>1)>=3 # only keep genes that have cpm>1 (== not zeroes)
# in more than 3 samples (minimal group size)
myDGEList.filtered <- myDGEList[keepers,]
# plot filtered, non-normalized CPM
p2 <- profile(myDGEList.filtered, sampleLabels, "Filtered, non-normalized")
# normalize counts via the TMM method implemented in edgeR
myDGEList.filtered.norm <- calcNormFactors(myDGEList.filtered, method = "TMM")
# plot filtered, normalized CPM
p3 <- profile(myDGEList.filtered.norm, sampleLabels, "Filtered, TMM normalized")
# compare distributions of the CPM values
plot_grid(p1, p2, p3, labels = c('A', 'B', 'C'), label_size = 12)
```

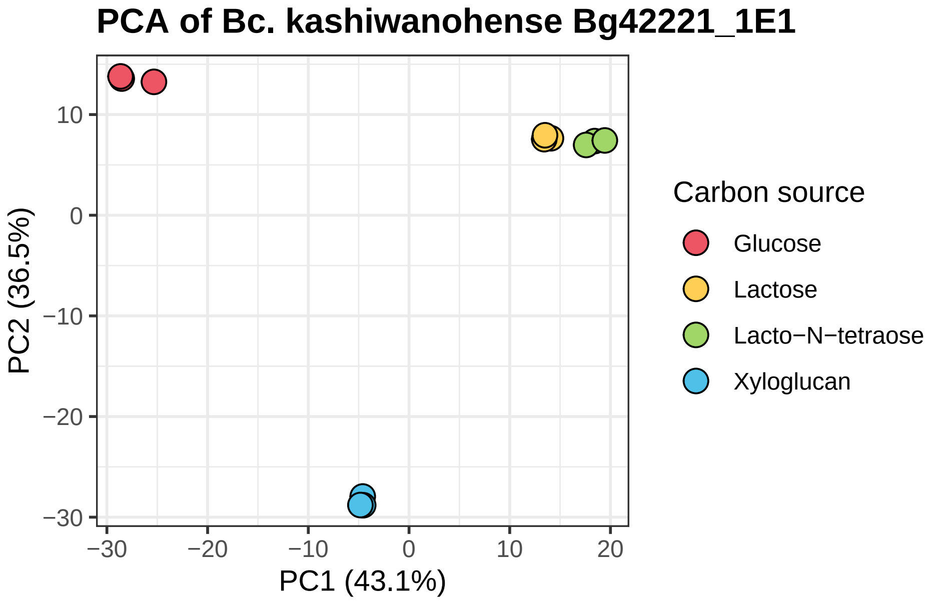


Filtering was carried out to remove lowly expressed genes. Genes with less than 1 count per million (CPM) in at least 3 or more samples were filtered out. This procedure reduced the number of genes from **2099** to **1927**. In addition, the TMM method was used for between-sample normalization.

12.4 PCA plot

Principal Component Analysis (PCA) plots reduce complex datasets to a 2D representation where each axis represents a source of variance (known or unknown) in the dataset. Principal Component 1 (PC1; X-axis) accounted for >43% of the variance in the data. PC2 (Y-axis) accounted for >36% of the variance in the data.

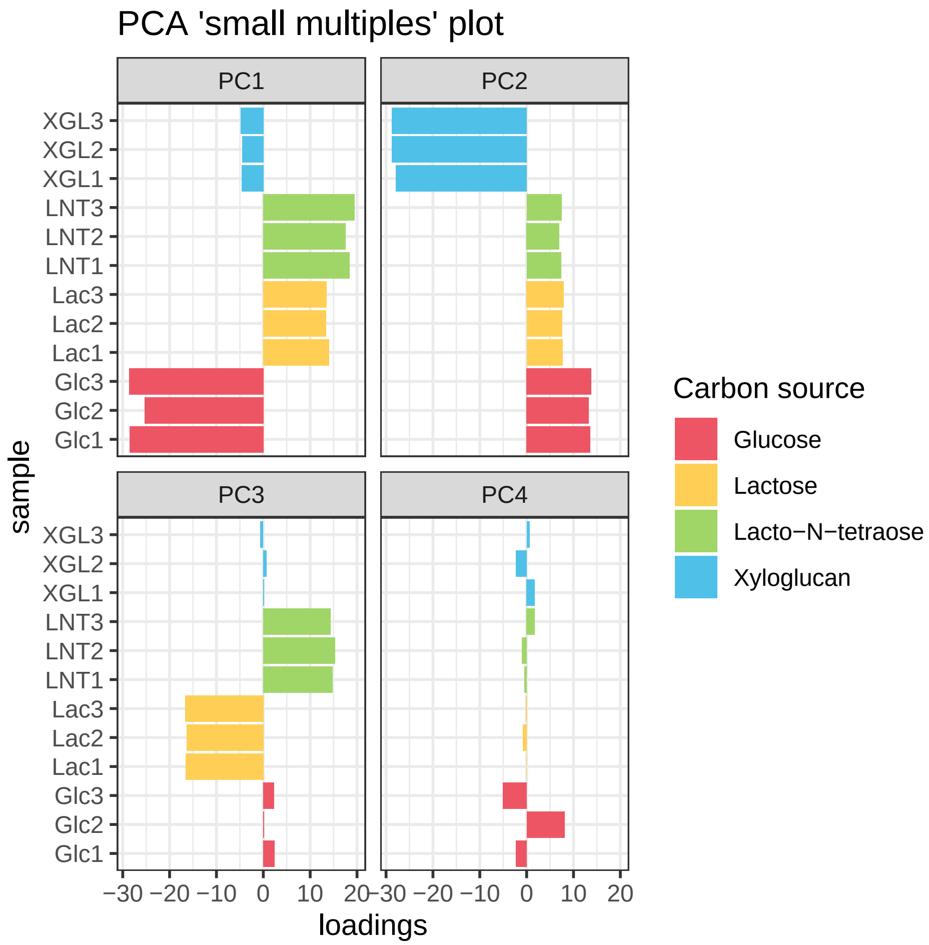
```
# running PCA
log2.cpm.filtered.norm <- cpm(myDGEList.filtered.norm, log=TRUE)
pca.res <- prcomp(t(log2.cpm.filtered.norm), scale.=F, retx=T)
pc.var <- pca.res$sdev^2 # sdev^2 captures eigenvalues from the PCA result
pc.per <- round(pc.var/sum(pc.var)*100, 1) # calculate percentage of the total variation
# explained by each eigenvalue
# converting PCA result into a tibble for plotting
pca.res.df <- as_tibble(pca.res$x)

# plotting PCA
ggplot(pca.res.df) +
  aes(x=PC1, y=PC2, label=sampleLabels, fill = condition) +
  geom_point(size=4, shape = 21) +
  scale_fill_manual(name = "Carbon source",
                     breaks=c("Glc","Lac", "LNT", "XGL"),
                     values=c("#ed5564", "#ffce54", "#a0d568", "#4fc1e8"),
                     labels=c("Glucose","Lactose", "Lacto-N-tetraose", "Xyloglucan")) +
  guides(fill = guide_legend(override.aes=list(shape=21))) +
  xlab(paste0("PC1 (",pc.per[1],"%",")")) +
  ylab(paste0("PC2 (",pc.per[2],"%",")")) +
  labs(title= "PCA of Bc. kashiwanohense Bg42221_1E1") +
  coord_fixed() +
  theme_bw() +
  theme(plot.title = element_text(face="bold"))
```



```
# save the figure
ggsave("results/rnaseq/figures/PCA.pdf", device = "pdf", width = 5, height = 5)
```

```
# PCA 'small multiples' chart
# view PCA loadings to understand impact of each sample on each principal component
pca.res.df <- pca.res$x[,1:4] %>%
  as_tibble() %>%
  add_column(sample = sampleLabels,
             group = condition)
```

```
pca.pivot <- pivot_longer(pca.res.df, # data frame to be pivoted
                          cols = PC1:PC4, # column names to be stored as a SINGLE variable
                          names_to = "PC", # name of that new variable (column)
                          values_to = "loadings") # name of new variable (column) storing all the value

# plot the chart
ggplot(pca.pivot) +
  aes(x=sample, y=loadings, fill=group) +
  geom_bar(stat="identity") +
  facet_wrap(~PC) +
  scale_fill_manual(name = "Carbon source",
                    breaks=c("Glc","Lac", "LNT", "XGL"),
                    values=c("#ed5564", "#ffce54", "#a0d568", "#4fc1e8"),
                    labels=c("Glucose","Lactose", "Lacto-N-tetraose", "Xyloglucan")) +
  labs(title="PCA 'small multiples' plot") +
  theme_bw() +
  coord_flip()
```

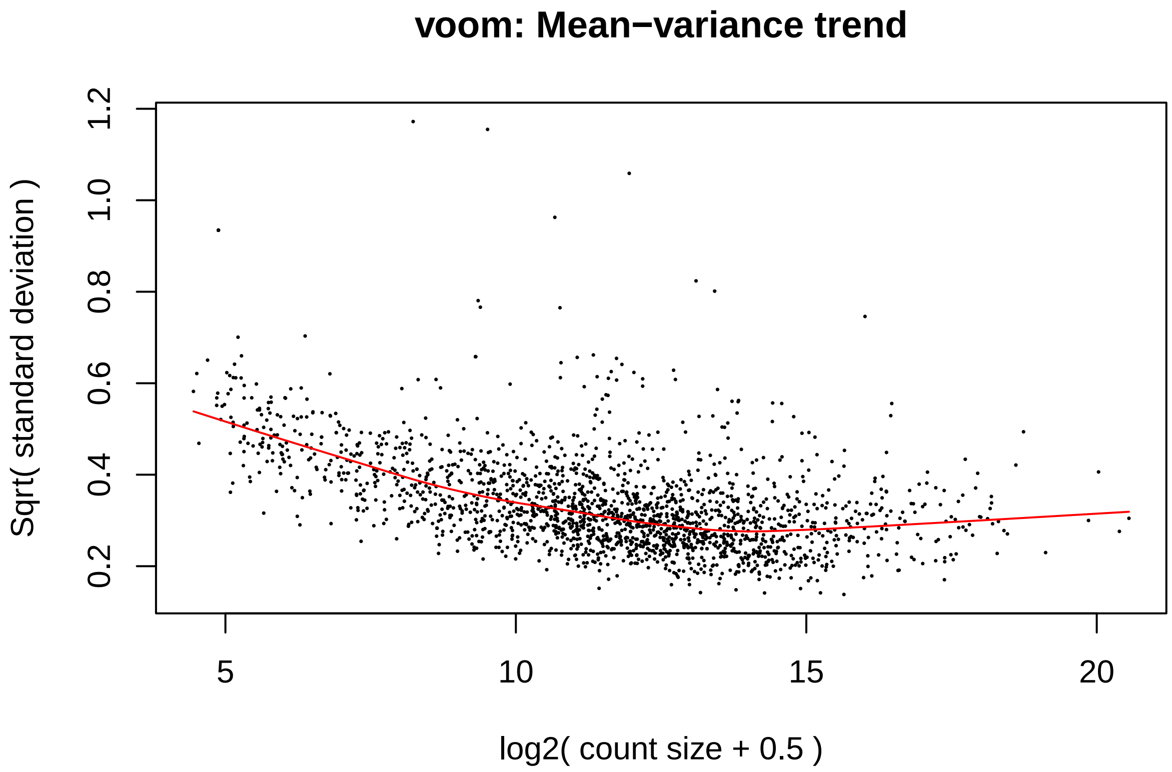


```
# save the figure
ggsave("results/rnaseq/figures/PCA_small_multiples.pdf", device = "pdf", width = 8, height = 5)
```

12.5 Differentially expressed genes

To identify differentially expressed genes (DEGs), precision weights were first applied to each gene based on its mean-variance relationship using VOOM. Linear modeling and bayesian stats were employed using Limma to find genes that were up- or down-regulated more than 4-fold at false-discovery rate (FDR) of 0.01.

```
# setting up model matrix without intercept
design <- model.matrix(~0 + condition)
colnames(design) <- levels(condition)
# using VOOM function from Limma package to apply precision weights to each gene
v.DEGList.filtered.norm <- voom(myDGEList.filtered.norm, design, plot = TRUE)
```



```
fit <- lmFit(v.DEGList.filtered.norm, design)
# setting up contrast matrix for pairwise comparisons of interest
contrast.matrix <- makeContrasts(XGL = XGL - Lac,
                                 LNT = LNT - Lac,
                                 levels=design)
fits <- contrasts.fit(fit, contrast.matrix)
# extracting stats
ebFit <- eBayes(fits)
```

DEGs were annotated based on a RAST-annotated version of the *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg42221_1E1 genome, which was additionally subjected to extensive manual curation in mcSEED, a private clone of the publicly available SEED platform. The manual curation focused on annotating genes encoding functional roles (transporters, glycoside hydrolases, downstream catabolic enzymes, transcriptional regulators) involved in carbohydrate metabolism.

DEGs: *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg42221_1E1 grown in MRS-AC-XGL vs. MRS-AC-Lac

```
# create a master annotation table
seed.ann <- read_tsv('data/rnaseq/annotation/mcSEED_annotations.txt')
corr <- read_tsv('data/rnaseq/annotation/locus_tag_comparison.txt')
final.ann <- right_join(seed.ann, corr, by = c('seed_id' = 'seed_id')) %>%
  dplyr::select(locus_tag, annotation)

# annotate DEGs
# xyloglucan vs lactose
myTopHits.XGLvsLac <- topTable(ebFit, adjust ="BH", coef=1, number=2600, sort.by="logFC")
deg_list(myTopHits.XGLvsLac, -2, 2, 0.01, "results/rnaseq/tables/table_S15A.txt")
```

DEGs: *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg42221_1E1 grown in MRS-AC-LNT vs. MRS-AC-Lac

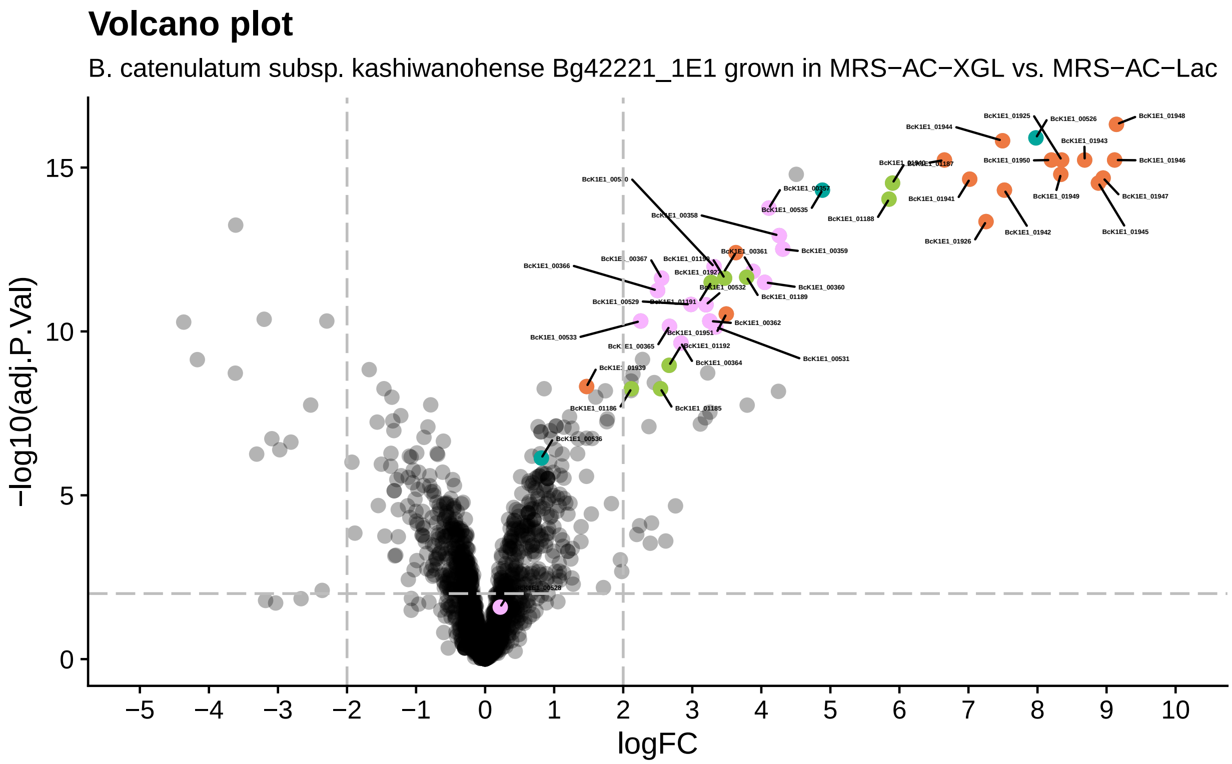
```
# annotate DEGs
# lacto-N-tetraose vs lactose
myTopHits.LNTvsLac <- topTable(ebFit, adjust ="BH", coef=2, number=2600, sort.by="logFC")
deg_list(myTopHits.LNTvsLac, -2, 2, 0.01, "results/rnaseq/tables/table_S15B.txt")
```

12.6 Volcano plot: *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg42221_1E1 grown in MRS-AC-XGL vs MRS-AC-Lac

Volcano plots are convenient ways to represent gene expression data because they combine magnitude of change (X-axis) with significance (Y-axis). Since the Y-axis is the inverse log10 of the adjusted Pvalue, higher points are more significant. In the case of this particular plot, there are many genes in the upper right of the plot, which represent genes that are significantly **upregulated** in when the bacterium is grown in MRS-AC-XGL, compared MRS-AC-Lac.

```
# list stats for all genes in the dataset to be used for making the volcano plot
myTopHits <- topTable(ebFit, adjust ="BH", coef=1, number=3000, sort.by="logFC")
myTopHits.df <- myTopHits %>%
  as_tibble(rownames = "geneID")
# select only genes with significant logFC and adj.P.Val for the heatmap
myTopHits.df.de <- subset(myTopHits.df, (logFC > 2 | logFC < 2) & adj.P.Val < 0.01)
# create vectors containing locus_tags of genes constituting reconstructed regulons
targets.XglR <- c("BcK1E1_01925", "BcK1E1_01926", "BcK1E1_01927", "BcK1E1_01944",
                  "BcK1E1_01943", "BcK1E1_01942", "BcK1E1_01941", "BcK1E1_01940",
                  "BcK1E1_01939", "BcK1E1_01950", "BcK1E1_01949", "BcK1E1_01948",
                  "BcK1E1_01947", "BcK1E1_01946", "BcK1E1_01945", "BcK1E1_01951")
targets.XylR <- c("BcK1E1_00526", "BcK1E1_00535", "BcK1E1_00536")
targets.XosR <- c("BcK1E1_00357", "BcK1E1_00358", "BcK1E1_00359", "BcK1E1_00360",
                  "BcK1E1_00361", "BcK1E1_00362", "BcK1E1_00363", "BcK1E1_00364",
                  "BcK1E1_00365", "BcK1E1_00366", "BcK1E1_00367", "BcK1E1_00528",
                  "BcK1E1_00529", "BcK1E1_00530", "BcK1E1_00531", "BcK1E1_00532", "BcK1E1_00533")
targets.XosR2 <- c("BcK1E1_01192", "BcK1E1_01191", "BcK1E1_01190", "BcK1E1_01189",
                   "BcK1E1_01188", "BcK1E1_01187", "BcK1E1_01186", "BcK1E1_01185")
# subset data based on targets.XglR/xylR/xosR/xosR2
myTopHits.XglR <- subset(myTopHits.df, geneID %in% targets.XglR)
myTopHits.XylR <- subset(myTopHits.df, geneID %in% targets.XylR)
```

```r
myTopHits.XosR <- subset(myTopHits.df, geneID %in% targets.XosR)
myTopHits.XosR2 <- subset(myTopHits.df, geneID %in% targets.XosR2)
# subset data labels (genes in regulons)
## XglR
myTopHits.df$XglR <- myTopHits.df$geneID
myTopHits.XglR_selected <- myTopHits.df$XglR %in% myTopHits.XglR$geneID
myTopHits.df$XglR[!myTopHits.XglR_selected] <- NA
## XylR
myTopHits.df$XylR <- myTopHits.df$geneID
myTopHits.XylR_selected <- myTopHits.df$XylR %in% myTopHits.XylR$geneID
myTopHits.df$XylR[!myTopHits.XylR_selected] <- NA
## XosR
myTopHits.df$XosR <- myTopHits.df$geneID
myTopHits.XosR_selected <- myTopHits.df$XosR %in% myTopHits.XosR$geneID
myTopHits.df$XosR[!myTopHits.XosR_selected] <- NA
## XosR
myTopHits.df$XosR2 <- myTopHits.df$geneID
myTopHits.XosR2_selected <- myTopHits.df$XosR2 %in% myTopHits.XosR2$geneID
myTopHits.df$XosR2[!myTopHits.XosR2_selected] <- NA

# create the volcano plot
ggplot(myTopHits.df) +
  aes(y=-log10(adj.P.Val), x=logFC, text = paste("Symbol:", geneID)) +
  geom_point(size=3, shape = 16, color="black", alpha=.3) +
  geom_point(mapping=NULL, myTopHits.XglR, size = 3, shape = 16, color= "#ee7942",
             inherit.aes = TRUE) +
  geom_point(mapping=NULL, myTopHits.XylR, size = 3, shape = 16, color= "#00a69c",
             inherit.aes = TRUE) +
  geom_point(mapping=NULL, myTopHits.XosR, size = 3, shape = 16, color= "#f8b4ff",
             inherit.aes = TRUE) +
  geom_point(mapping=NULL, myTopHits.XosR2, size = 3, shape = 16, color= "#9ac946",
             inherit.aes = TRUE) +
  geom_text_repel(aes(label = XglR),
                  size = 1, fontface=2, color="black", min.segment.length = 0,
                  seed = 42, box.padding = 0.5, max.overlaps = 100) +
  geom_text_repel(aes(label = XylR),
                  size = 1, fontface=2, color="black", min.segment.length = 0,
                  seed = 42, box.padding = 0.5, max.overlaps = 100) +
  geom_text_repel(aes(label = XosR),
                  size = 1, fontface=2, color="black", min.segment.length = 0,
                  seed = 42, box.padding = 0.5, max.overlaps = 100) +
  geom_text_repel(aes(label = XosR2),
                  size = 1, fontface=2, color="black", min.segment.length = 0,
                  seed = 42, box.padding = 0.5, max.overlaps = 100) +
  geom_hline(yintercept = -log10(0.01), linetype="longdash", colour="grey", size=0.6) +
  geom_vline(xintercept = 2, linetype="longdash", colour="grey", size=0.6) +
  geom_vline(xintercept = -2, linetype="longdash", colour="grey", size=0.6) +
  annotate("text", x=-6, y=-log10(0.01)+0.3,
           label=paste("Padj<0.01"), size=5, fontface="bold") +
  scale_x_continuous(limits=c(-5,10), breaks = -5:10) +
  labs(title="Volcano plot",
       subtitle = "B. catenulatum subsp. kashiwanohense Bg42221_1E1 grown in MRS-AC-XGL vs. MRS-AC-Lac"
  theme(plot.title = element_text(face="bold")) +
  theme_cowplot()
```

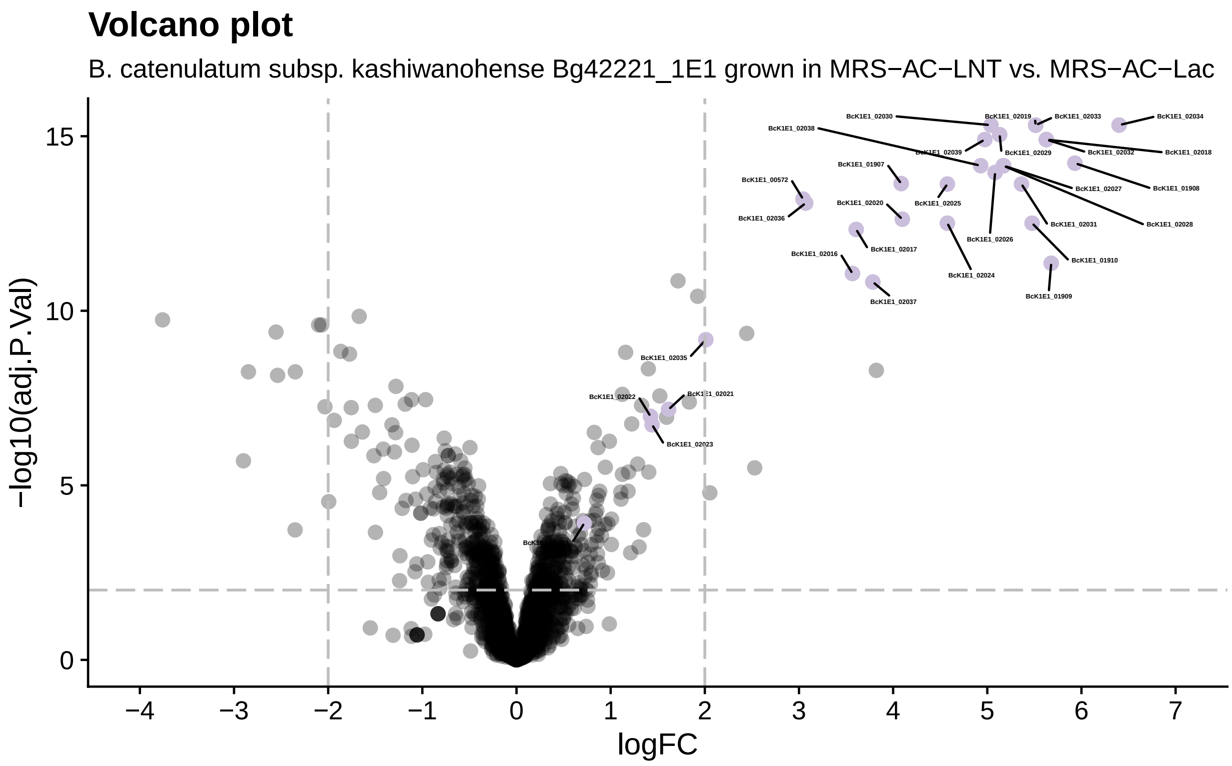


```
# save the figure
ggsave("results/rnaseq/figures/XGL_vs_Lac.pdf", device = "pdf", width = 8, height = 5)
```

12.7 Volcano plot: *Bifidobacterium catenulatum* subsp. *kashiwanohense* Bg42221_1E1 grown in MRS-AC-LNT vs MRS-AC-Lac

```
# listing stats for all genes in the dataset to be used for making volcano plot
myTopHits2 <- topTable(ebFit, adjust ="BH", coef=2, number=3000, sort.by="logFC")
myTopHits.df2 <- myTopHits2 %>%
  as_tibble(rownames = "geneID")
# select only genes with significant logFC and adj.P.Val for the heatmap
myTopHits.df2.de <- subset(myTopHits.df2, (logFC > 2 | logFC < 2) & adj.P.Val < 0.01)
# create a vector containing locus_tags of genes predicted to be in the HMO cluster
targets.NagR <- c("BcK1E1_00572", "BcK1E1_01910", "BcK1E1_01909", "BcK1E1_01908",
                  "BcK1E1_01907", "BcK1E1_01911", "BcK1E1_01907", "BcK1E1_02020",
                  "BcK1E1_02019", "BcK1E1_02018", "BcK1E1_02017", "BcK1E1_02016",
                  "BcK1E1_02034", "BcK1E1_02033", "BcK1E1_02032", "BcK1E1_02031",
                  "BcK1E1_02030", "BcK1E1_02029", "BcK1E1_02028", "BcK1E1_02027",
                  "BcK1E1_02026", "BcK1E1_02025", "BcK1E1_02024", "BcK1E1_02023",
                  "BcK1E1_02022", "BcK1E1_02021", "BcK1E1_02035", "BcK1E1_02036",
                  "BcK1E1_02037", "BcK1E1_02038", "BcK1E1_02039")
# subset data based on targets.nagR
myTopHits.NagR <- subset(myTopHits.df2, geneID %in% targets.NagR)
```

```r
# subset data labels (genes in regulons)
myTopHits.df2$NagR <- myTopHits.df2$geneID
myTopHits.NagR_selected <- myTopHits.df2$NagR %in% myTopHits.NagR$geneID
myTopHits.df2$NagR[!myTopHits.NagR_selected] <- NA

# create the volcano plot
ggplot(myTopHits.df2) +
  aes(y=-log10(adj.P.Val), x=logFC, text = paste("Symbol:", geneID)) +
  geom_point(size=3, shape = 16, color="black", alpha=.3) +
  geom_point(mapping=NULL, myTopHits.NagR, size = 3, shape = 16, color= "#cbbedd",
             inherit.aes = TRUE) +
  geom_text_repel(aes(label = NagR),
                  size = 1, fontface=2, color="black", min.segment.length = 0,
                  seed = 42, box.padding = 0.5, max.overlaps = 100) +
  geom_hline(yintercept = -log10(0.01), linetype="longdash", colour="grey", size=0.6) +
  geom_vline(xintercept = 2, linetype="longdash", colour="grey", size=0.6) +
  geom_vline(xintercept = -2, linetype="longdash", colour="grey", size=0.6) +
  annotate("text", x=-6, y=-log10(0.01)+0.3,
           label=paste("Padj<0.01"), size=5, fontface="bold") +
  scale_x_continuous(limits=c(-4,7), breaks = -4:7) +
  labs(title="Volcano plot",
       subtitle = "B. catenulatum subsp. kashiwanohense Bg42221_1E1 grown in MRS-AC-LNT vs. MRS-AC-Lac")
  theme(plot.title = element_text(face="bold")) +
  theme_cowplot()
```



```r
# save the figure
ggsave("results/rnaseq/figures/LNT_vs_Lac.pdf", device = "pdf", width = 8, height = 5)
```

13 Session info

The output from running 'sessionInfo' is shown below and details all packages necessary to reproduce the results in this report.

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.1.2
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] emmeans_1.10.2        AER_1.2-12            survival_3.6-4
##  [4] sandwich_3.1-0        lmtest_0.9-40         zoo_1.8-12
##  [7] car_3.1-2             carData_3.0-5         vegan_2.6-6.1
## [10] lattice_0.22-6        permute_0.9-7         cowplot_1.1.3
## [13] edgeR_4.0.16          limma_3.58.1          gt_0.10.1
## [16] rhdf5_2.46.1          tximport_1.30.0       ggpubr_0.6.0
## [19] circlize_0.4.16       ggrepel_0.9.5         ggbeeswarm_0.7.2
## [22] ComplexHeatmap_2.18.0 patchwork_1.2.0       lubridate_1.9.3
## [25] forcats_1.0.0         stringr_1.5.1         dplyr_1.1.4
## [28] purrr_1.0.2           readr_2.1.5           tidyr_1.3.1
## [31] tibble_3.2.1          ggplot2_3.5.1         tidyverse_2.0.0
## [34] pacman_0.5.1          knitr_1.45            tinytex_0.51
## [37] rmarkdown_2.27
##
## loaded via a namespace (and not attached):
##  [1] RColorBrewer_1.1-3  rstudioapi_0.16.0   shape_1.4.6.1
##  [4] magrittr_2.0.3      TH.data_1.1-2       estimability_1.5.1
##  [7] farver_2.1.2        GlobalOptions_0.1.2 ragg_1.3.2
## [10] vctrs_0.6.5         rstatix_0.7.2       htmltools_0.5.8.1
## [13] broom_1.0.6         Rhdf5lib_1.24.2     Formula_1.2-5
## [16] lifecycle_1.0.4     iterators_1.0.14    pkgconfig_2.0.3
## [19] Matrix_1.6-4        R6_2.5.1            fastmap_1.2.0
## [22] clue_0.3-65         digest_0.6.35       colorspace_2.1-0
## [25] S4Vectors_0.40.2    textshaping_0.3.7   labeling_0.4.3
## [28] fansi_1.0.6         timechange_0.3.0    abind_1.4-5
## [31] mgcv_1.9-1          compiler_4.3.2      bit64_4.0.5
## [34] withr_3.0.0         doParallel_1.0.17   backports_1.4.1
## [37] highr_0.10          ggsignif_0.6.4      MASS_7.3-60
## [40] rjson_0.2.21        tools_4.3.2         vipor_0.4.7
## [43] beeswarm_0.4.0      glue_1.7.0          nlme_3.1-164
## [46] rhdf5filters_1.14.1 cluster_2.1.6       generics_0.1.3
## [49] gtable_0.3.5        tzdb_0.4.0          hms_1.1.3
## [52] xml2_1.3.6          utf8_1.2.4          BiocGenerics_0.48.1
## [55] foreach_1.5.2       pillar_1.9.0        vroom_1.6.5
## [58] splines_4.3.2       bit_4.0.5           tidyselect_1.2.1
## [61] locfit_1.5-9.9      IRanges_2.36.0      stats4_4.3.2
## [64] xfun_0.44           statmod_1.5.0       matrixStats_1.3.0
## [67] stringi_1.8.4       yaml_2.3.8          evaluate_0.23
## [70] codetools_0.2-20    cli_3.6.2           xtable_1.8-4
## [73] systemfonts_1.1.0   munsell_0.5.1       Rcpp_1.0.12
## [76] coda_0.19-4.1       png_0.1-8           parallel_4.3.2
## [79] mvtnorm_1.2-5       scales_1.3.0        crayon_1.5.2
## [82] GetoptLong_1.0.5    rlang_1.1.3         multcomp_1.4-25
```