

EcoGS Manual

Link: www.github.com/KaletaLab/EcoGS | **Commit:** 8f9268e3

Dr Georgios Marinos and Dr Samer Kadib Alban

Friday, December 19, 2025

Contents

1	Overview	1
2	Example workflow	1
2.1	Data Input and Initialisation	2
2.2	Generation of Experimental Data	2
2.3	Quantifying Metabolic Interactions	3
2.4	Weighing Methods	4
2.5	Statistical Correlation with Host Age	5
3	References	6

1 Overview

This manual describes a computational workflow for quantifying microbial metabolic interactions. The pipeline utilises the EcoGS package and gapseq-derived metabolic models to estimate community-level ecological relationships.

Please provide the authors with feedback or questions here:

- <https://github.com/KaletaLab/EcoGS/issues>

Further enquires to c.kaleta@iem.uni-kiel.de and/or s.kadibalban@iem.uni-kiel.de

To cite our pipeline, please use our *bioRxiv* article:

- <https://doi.org/10.1101/2025.09.23.678088>

2 Example workflow

This manual describes a computational workflow within the SIHUMIx (Simplified Human Intestinal Microbiota) community and correlating these interactions with host physiological parameters, such as age. Both abundance table and age values are randomly generated for illustration purposes.

2.1 Data Input and Initialisation

The analysis begins by loading a **named list** of metabolic models.

```
# Load EcoGS
library(EcoGS)
# Load the pre-defined SIHUMIx metabolic models
path <- system.file("extdata", "SIHUMIx_gapseq_EcoGS.RDS", package = "EcoGS")
SIHUMIx_gapseq_EcoGS <- readRDS(path)
```

These models represent the eight species of the SIHUMIx community, reconstructed using the gapseq tool. The models are constrained on the same northern German diet, as it described in the manuscript.

```
# Verify the models and species names
summary(SIHUMIx_gapseq_EcoGS)
#>                                         Length Class Mode
#> Anaerostipes_caccae_DSM_14662           1   modelorg S4
#> Bacteroides_thetaiotaomicron_VPI_5482    1   modelorg S4
#> Bifidobacterium_longum_NCC2705          1   modelorg S4
#> Blautia_productus_DSM_2950                1   modelorg S4
#> Clostridium_butyricum_DSM_10702          1   modelorg S4
#> Clostridium_ramosum_VPI_0427_DSM_1402     1   modelorg S4
#> Escherichia_coli_str_K_12_substr_MG1655    1   modelorg S4
#> Lactobacillus_plantarum_subsp_plantarum_ATCC_14917 1   modelorg S4
```

2.2 Generation of Experimental Data

To demonstrate the pipeline, we used synthetic, randomly generated, datasets representing microbial abundances and host age for 10 mice.

- Abundance Matrix: A 8×10 data frame where rows represent the SIHUMIx species and columns represent individual samples. Values are normalised such that each column sums to 1 (relative abundance). A **data.frame** is needed for further processing.
- Host Age: A vector of 10 random integers between 10 and 100 weeks, representing the adult lifespan of the mice.

```
set.seed(42)

# Create a random abundance table for 8 species across 10 mice
abundance <- as.data.frame(matrix(0,8,10))
row.names(abundance) <- names(SIHUMIx_gapseq_EcoGS)
names(abundance) <- paste0("mouse",1:10)
for (i in 1:8) {
  abundance[i,] <- sample(1:100, 10)
}

# Normalise each column by its sum to obtain relative abundance
abundance <- as.data.frame(apply(abundance, 2, function(x) x / sum(x)))

# Generate random age data in weeks
age <- sample(10:100,10)
```

2.3 Quantifying Metabolic Interactions

The core of the EcoGS pipeline involves predicting how species interact based on their metabolic requirements and outputs.

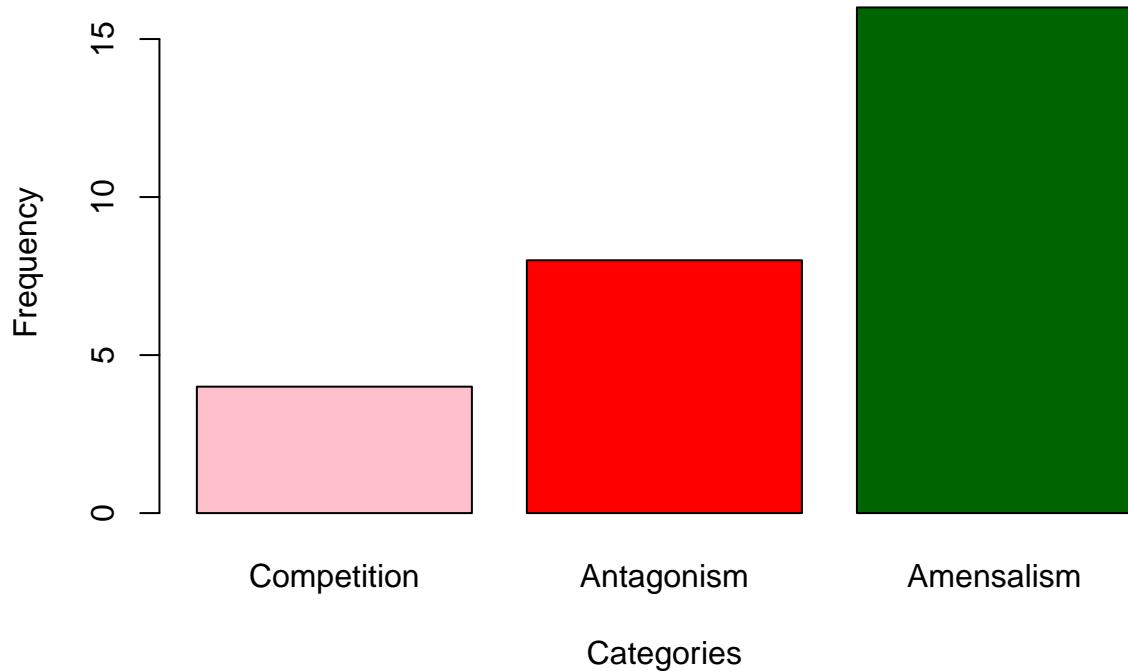
- Pairwise Interactions: *metabolic_interactions_with_MicrobiomeGS2* simulates growth for every possible pair of species to identify competitive, cooperative, or neutral interactions.
- Ecological Matrix: *make_eco_mat* converts these growth simulations into a structured ecological matrix.
- Visualisation: *plot_relations* provides a qualitative overview of the interaction network.

```
# Step 1: Simulate pairwise metabolic interactions
step1 <- metabolic_interactions_with_MicrobiomeGS2(list_of_models = SIHUMIx_gapseq_EcoGS,
                                                       cores = 1, save_pair = T)

# Step 2: Construct the ecological matrix
step2 <- make_eco_mat(growth_file = step1)
#> [1] "creating eco matrix, species Number: 1"
#> [1] "creating eco matrix, species Number: 2"
#> [1] "creating eco matrix, species Number: 3"
#> [1] "creating eco matrix, species Number: 4"
#> [1] "creating eco matrix, species Number: 5"
#> [1] "creating eco matrix, species Number: 6"
#> [1] "creating eco matrix, species Number: 7"
#> [1] "creating eco matrix, species Number: 8"

# Step 3: Visualise the predicted relations
step3 <- plot_relations(eco_mat = step2$eco_mat)
```

Eco relations



2.4 Weighing Methods

The ecological matrix is adjusted for each sample based on the observed relative abundance of the species. Two weighing methods are employed:

- Min-method: Uses the minimum abundance of the interacting pair.
- Multi-method: Uses the product of the abundances of the interacting pair.

The interaction ratios (e.g., cooperation vs. competition) are on a log10 scale.

```
# Weigh interactions
step4_min <- relation_per_sample(OTU_table = abundance, eco_mat = step2$eco_mat,
                                    weighing_method = "min")
#> [1] "Sample Number: 1"
#> [1] "Sample Number: 2"
#> [1] "Sample Number: 3"
#> [1] "Sample Number: 4"
#> [1] "Sample Number: 5"
#> [1] "Sample Number: 6"
#> [1] "Sample Number: 7"
#> [1] "Sample Number: 8"
#> [1] "Sample Number: 9"
#> [1] "Sample Number: 10"
step4_multi <- relation_per_sample(OTU_table = abundance, eco_mat = step2$eco_mat,
```

```

                weighing_method = "multi" )

#> [1] "Sample Number: 1"
#> [1] "Sample Number: 2"
#> [1] "Sample Number: 3"
#> [1] "Sample Number: 4"
#> [1] "Sample Number: 5"
#> [1] "Sample Number: 6"
#> [1] "Sample Number: 7"
#> [1] "Sample Number: 8"
#> [1] "Sample Number: 9"
#> [1] "Sample Number: 10"

# Calculate log10 interaction ratios
step5_min <- relation_ratios(relations_table = step4_min$weighed_relations)
step5_multi <- relation_ratios(relations_table = step4_multi$weighed_relations)

```

2.5 Statistical Correlation with Host Age

Finally, the weighted interaction ratios are correlated with the host's age using the non-parametric Spearman's rank correlation test. This step is not part of the pipeline. However, it serves as a suggestion of downstream statistical analyses.

```

# Test correlation between the three interaction ratios and host age
for (i in 1:3) {
  print(cor.test(x = step5_min[,i],y = age, method = "spearman"))
}
#>
#> Spearman's rank correlation rho
#>
#> data: step5_min[, i] and age
#> S = 142, p-value = 0.7072
#> alternative hypothesis: true rho is not equal to 0
#> sample estimates:
#>      rho
#> 0.1393939
#>
#>
#> Spearman's rank correlation rho
#>
#> data: step5_min[, i] and age
#> S = 136, p-value = 0.632
#> alternative hypothesis: true rho is not equal to 0
#> sample estimates:
#>      rho
#> 0.1757576
#>
#>
#> Spearman's rank correlation rho
#>
#> data: step5_min[, i] and age
#> S = 136, p-value = 0.632
#> alternative hypothesis: true rho is not equal to 0

```

```
#> sample estimates:  
#>      rho  
#> 0.1757576
```

3 References

- Becker *et al.*, Human intestinal microbiota: characterization of a simplified and stable gnotobiotic rat model, Gut Microbes. 2011 Jan-Feb;2(1):25-33. doi: 10.4161/gmic.2.1.14651.
- Zimmermann *et al.*, gapseq: informed prediction of bacterial metabolic pathways and reconstruction of accurate metabolic models, Genome Biol. 2021 Mar 10;22(1):81. doi: 10.1186/s13059-021-02295-1.