

# Appendix

## Overview and reproducibility

This document compiles methodological details, supplementary analyses, and reproducibility outputs supporting the main manuscript. Computationally intensive steps (e.g. phylogenetic tree inference, GLLVM fitting, iNEXT3D diversity estimation) were executed on a remote Ubuntu-based computing environment. Their outputs (figures, tables, and selected model objects) are integrated here as static results.

All figures (PNG) and result tables (CSV) are stored within the project repository using relative paths. Rendering this document therefore requires cloning the repository and opening the .qmd file from the repository root.

### Data availability.

Repository: To be provided upon acceptance

Sequence data: ENA accession number PRJEB107725

Metadata: Data\_S1\_metadata.csv, Data\_S3\_soil.csv

```
# Output folders used by table/figure chunks
out_fig_dir <- "figures"
out_tab_dir <- "tables"
out_obj_dir <- "objects"

# Main analysis phyloseq object (root-only; replicates collapsed; soil reads removed; decontaminated)
ps_individual <- readRDS(file.path(out_obj_dir, "ps_individual.rds"))

# Phyloseq object that includes the phylogeny and keeping the 2 root replicates separate
tree_ps <- readRDS(file.path(out_obj_dir, "tree_ps.rds"))
```

## Study sites and sampling

This study was conducted across four humid páramo regions in the Colombian Andes, spanning the Central and Eastern Cordilleras. Sampling locations covered elevations from 2,715 to 3,828 m a.s.l., representing the ecological transition from Andean forest to páramo (see main text, Fig. 1). The full sample metadata table is provided in Data\_S1\_metadata.csv.

Root sampling followed standardized sterile procedures. Fine roots of *Gaultheria myrsinoides* were excavated using a spade, and 3–5 cm root tips were clipped using a sterilized pruner. Tools were disinfected with 75% ethanol between samples, and gloves were changed between individual plants to minimize contamination. Root fragments were placed into sealed polybags stored in ice-cooled containers in the field. Samples were processed at Universidad ICESI, where roots were rinsed in sterile water to remove debris, dried briefly under a laminar flow hood, and stored at –20°C prior to shipment to Norway, and long-term storage at –80°C. Soil samples (top 15 cm) were collected at each sampling location, and five aliquots of rinse water were retained as laboratory and transport controls.

**Table S1:** GPS coordinates, elevation, and vegetation type classification of the 12 sampling locations across four páramo regions. These sites span the forest–subpáramo–páramo gradient and were used for root and soil sampling.

Site ID	Site	Altitude (m)	Latitude (N)	Longitude (W)	Ecosystem
NV_1	Páramo La Nevera	3828	03°30'58.1"	076°03'05.4"	Páramo
NV_2	Páramo La Nevera	3550	03°31'37.7"	076°03'38.7"	Subpáramo
NV_3	Páramo La Nevera	3333	03°32'07.0"	076°04'18.3"	Andean forest
NV_4	Páramo La Nevera	3078	03°32'48.9"	076°04'31.3"	Andean forest
DOM_1	Páramo Las Domínguez	3815	03°43'51.8"	076°06'38.1"	Páramo
DOM_2	Páramo Las Domínguez	3534	03°44'07.2"	076°05'53.2"	Subpáramo
DOM_3	Páramo Las Domínguez	3234	03°44'17.1"	076°05'32.9"	Andean forest
BEL_1	Páramo Belmira	3254	06°38'43.8"	075°40'13.3"	Subpáramo
BEL_2	Páramo Belmira	2950	06°38'22.7"	075°39'57.3"	Andean forest
BEL_3	Páramo Belmira	2715	06°36'55.3"	075°39'51.1"	Andean forest

MA_1	Páramo Matarredonda	3678	04°33'31.5"	074°01'43.8"	Páramo
MA_2	Páramo Matarredonda	3381	04°33'10.3"	073°59'35.3"	Subpáramo

## Bioinformatic pipeline and read-filtering summary

This section provides supplementary details on sequencing read processing and quality filtering. The full bioinformatic pipeline (including DADA2 processing and replicate filtering) is available in the project repository. Here we report summary tables of read retention and replicate-filtering thresholds used for downstream analyses.

**Table S2:** Median read numbers per PCR replicate for environmental samples (root and soil samples) and controls (blank extractions, positive, PCR blank, and tag-jump controls). The read counts are grouped at each stage of the DADA2 pipeline. Blank extraction controls (not containing any biological sample) were performed alongside real sample DNA extractions per each extraction batch. PCR blank controls contained all PCR reagents but no DNA template. Tag-jump controls are PCR reactions containing no primers and no DNA template, used to detect tag-jumping (index hopping).

	Raw reads	Filtered	Merged - Non pooled	Merged - Pooled	Merged - Pseu- dopooled
<b>Root samples</b>	77,047	71,783	71,084	70,343	71,242
<b>Soil samples</b>	72,158	66,555	65,311	65,498	65,385
<b>Blank Extraction Controls</b>	2,844	2,652	2,618	2,566	2,625
<b>Positive Controls</b>	95,238	88,174	88,062	87,996	87,972
<b>PCR Blank Controls</b>	291	261	250	257	256
<b>Tag-jump Controls</b>	41	36	27	30	32

---

Warning: package 'knitr' was built under R version 4.4.3

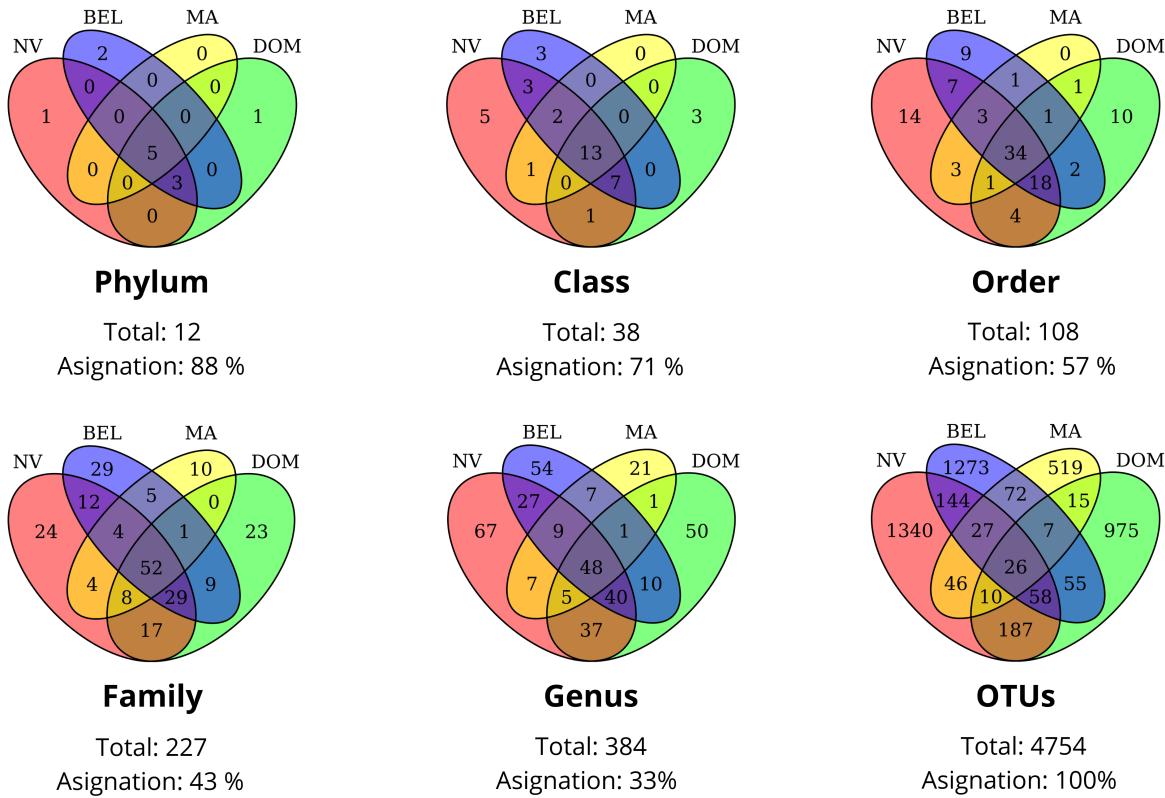
Table 3: Table S3. Number of OTUs and samples after applying different replicate control filtering thresholds. The table shows results for both soil and root samples combined and only root samples.

Filtering_level	OTU_count_soil+rSamples_soil+root	OTU_count_root	Samples_root	Samples_root_only
No filtering	18451	132	13225	120
OTU present in 2/4	7174	132	4386	120
OTU present in 3/4	4870	131	2966	120
OTU present in 4/4	2875	113	1904	103

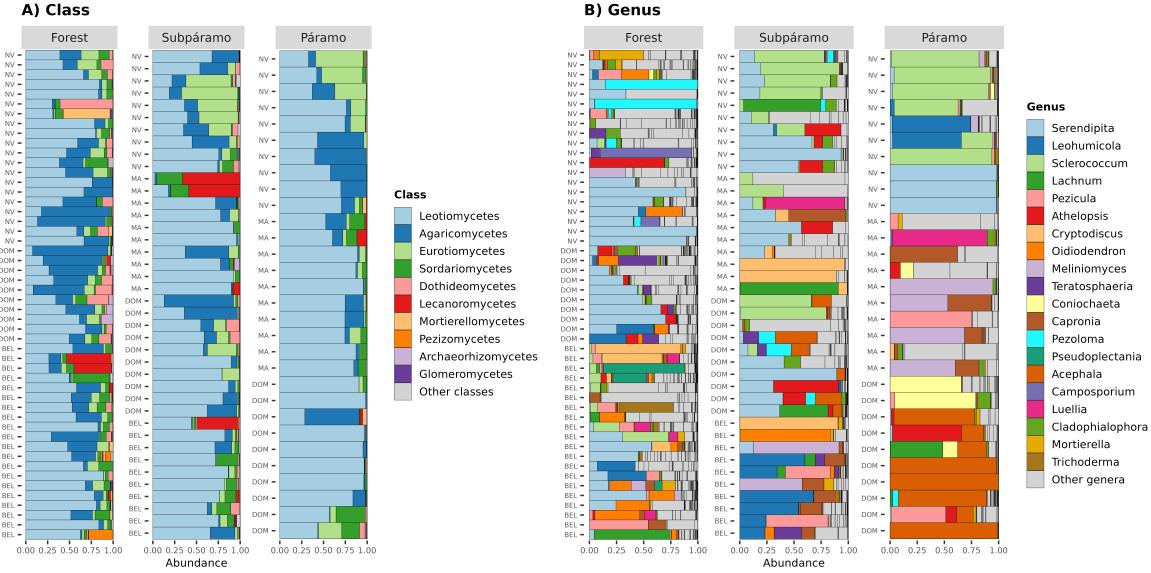
**Table S4:** Summary of the percentage of reads and OTUS with confident taxonomic assignment at each taxonomic rank, represented by habitat. Percentages were calculated from the filtered root-only dataset (soil reads removed).

Rank	Habitat	%_reads_assigned	%_otus_assigned
Phylum	Forest	88,1	88,1
Phylum	Subpáramo	91,6	88,6
Phylum	Páramo	92,9	89,3
Class	Forest	78,3	72,5
Class	Subpáramo	84,2	72,3
Class	Páramo	84,3	76,0
Order	Forest	68,3	59,2
Order	Subpáramo	74,2	57,1
Order	Páramo	78,2	60,9
Genus	Forest	32,8	34,2
Genus	Subpáramo	32,1	32,2
Genus	Páramo	31,4	32,7

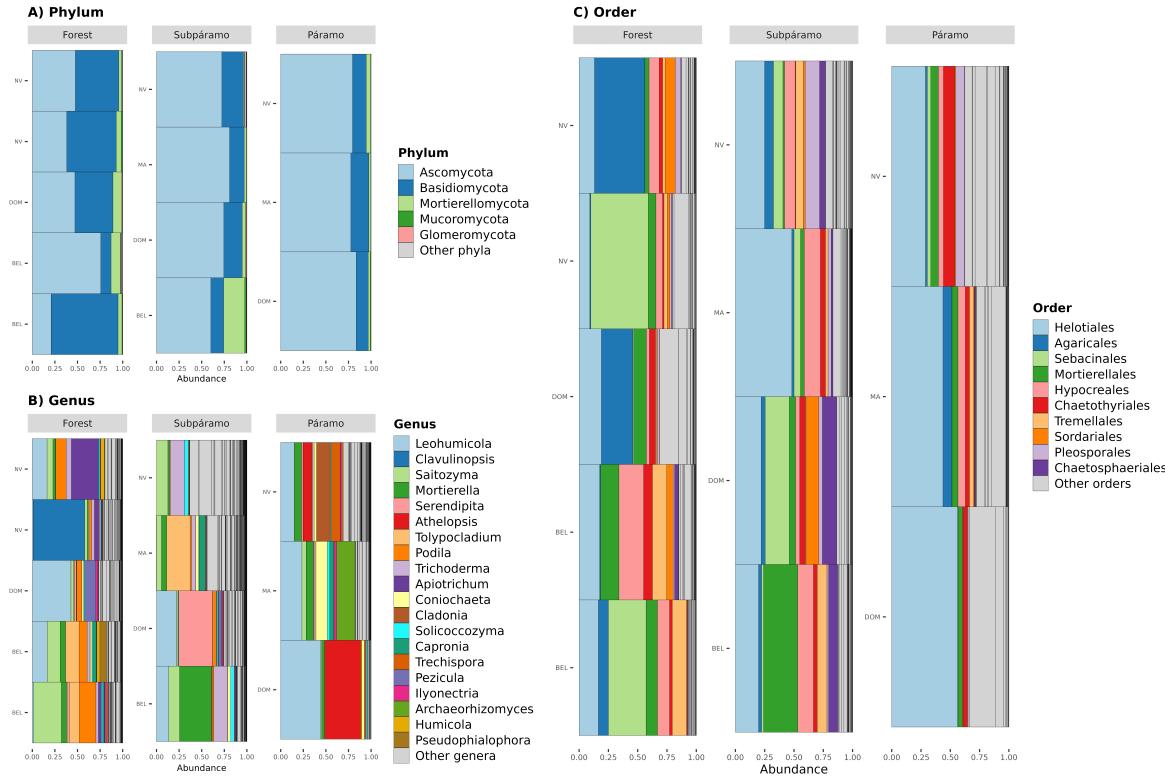
## Taxonomic composition of root and soil communities



**Figure S1:** Venn diagram of taxonomic assignments across sampling sites, in the root only dataset, but previous to soil sequences removal. Venn diagrams illustrate the overlap of OTU taxonomic assignments across four study sites: La Nevera (NV), Las Domínguez (DOM), Belmira (BEL), and Matarredonda (MA). Only OTUs with confirmed assignments at each taxonomic level are included.



**Figure S2:** Taxonomic composition in root samples along the elevational gradient. Relative read abundances are shown at two taxonomic ranks: A) Class and B) Genus. Each bar represents an individual root sample, grouped by habitat: Forest (2700-3300 m), Subpáramo (3300-3500 m), and Páramo (3800 m). Study sites are indicated on the y-axis. *Code: see “MicroViz code for taxonomic barplots” below.*



**Figure S3:** Taxonomic composition in soil samples, collected at each of the sampling locations. Relative read abundances are shown at three taxonomic ranks: A) Phylum, B) Genus, and C) Order. Each bar represents an individual root sample, grouped by habitat: Forest (2700-3300 m), Subpáramo (3300-3500 m), and Páramo (3800 m). Study sites are indicated on the y-axis. *Code: see “MicroViz code for taxonomic barplots” below.*

```
# MicroViz taxonomic barplots
# Code executed on remote server; plots saved as PNG and embedded above.

library(microViz)

filter_taxa_by_rank <- function(physeq_obj, rank_prefix = "o__", exclude_term = "Incertae_se
  taxa_assigned <- grepl(rank_prefix, tax_table(physeq_obj)[, "Order"])
  exclude_incertae_sedis <- !grepl(exclude_term, tax_table(physeq_obj)[, "Order"])
  taxa_filtered <- taxa_assigned & exclude_incertae_sedis
  physeq_filtered <- prune_taxa(taxa_filtered, physeq_obj)
  return(physeq_filtered)
}

ps_filtered <- filter_taxa_by_rank(alldat.N[[2]])
```

```

# Clean taxonomic prefixes like o__, g__, etc.
tax_table(ps_filtered) <- apply(
  tax_table(ps_filtered), 2, function(x) gsub("^[a-z]__", "", x)
)

# Recode habitat labels
sample_data(ps_filtered)$habitat <- dplyr::recode(
  as.character(sample_data(ps_filtered)$habitat),
  "forest"      = "Forest",
  "subparamo"   = "Subpáramo",
  "paramo"      = "Páramo",
  .default = NA_character_
)

# Order the factor levels
sample_data(ps_filtered)$habitat <- factor(
  sample_data(ps_filtered)$habitat,
  levels = c("Forest", "Subpáramo", "Páramo"),
  ordered = TRUE
)

ps_filtered <- ps_filtered %>% ps_arrange(site)

# within each site, order samples by Bray/OLO seriation at Order level ---
site_levels <- sample_data(ps_filtered) %>% as.data.frame() %>% pull(site) %>% unique()

samp_order <- unlist(lapply(site_levels, function(s) {
  ps_filtered %>%
    ps_filter(site == s) %>%
    ps_seriate(rank = "Order") %>%    # same tax_level you plot
    sample_names()                  # returns samples in seriated order
}))

# Barplot faceted by habitat, clean legend labels
p <- ps_filtered %>%
  comp_barplot(
    tax_level = "Order", n_taxa = 10, label = "site",
    bar_outline_colour = "grey5", facet_by = "habitat", sample_order = samp_order,
    merge_other = FALSE, other_name = "Other orders"
  ) +
  coord_flip() +

```

```

theme(
  legend.text = element_text(size = 6),
  legend.title = element_text(size = 7),
  legend.key.size = unit(0.3, "cm"),
  legend.spacing.x = unit(0.2, "cm"),
  plot.title = element_text(hjust = 0.5)
)

#Plot for phylum
phy <- ps_filtered %>%
  comp_barplot(
    tax_level = "Phylum", n_taxa = 5, label = "site",
    bar_outline_colour = "grey5", facet_by = "habitat", sample_order = samp_order,
    merge_other = FALSE, other_name = "Other phyla"
  ) +
  coord_flip() +
  theme(
    legend.text = element_text(size = 6),
    legend.title = element_text(size = 7),
    legend.key.size = unit(0.3, "cm"),
    legend.spacing.x = unit(0.2, "cm"),
    plot.title = element_text(hjust = 0.5)
  )

#Plot for class
class <- ps_filtered %>%
  comp_barplot(
    tax_level = "Class", n_taxa = 10, label = "site",
    bar_outline_colour = "grey5", facet_by = "habitat", sample_order = samp_order,
    merge_other = FALSE, other_name = "Other classes"
  ) +
  coord_flip() +
  theme(
    legend.text = element_text(size = 6),
    legend.title = element_text(size = 7),
    legend.key.size = unit(0.3, "cm"),
    legend.spacing.x = unit(0.2, "cm"),

```

```

    plot.title = element_text(hjust = 0.5)
  )

#Plot for genus
genus <- ps_filtered %>%
  comp_barplot(
    tax_level = "Genus", n_taxa = 20, label = "site",
    bar_outline_colour = "grey5", facet_by = "habitat", sample_order = samp_order,
    merge_other = FALSE, other_name = "Other genera"
  ) +
  coord_flip() +
  theme(
    legend.text = element_text(size = 6),
    legend.title = element_text(size = 7),
    legend.key.size = unit(0.3, "cm"),
    legend.spacing.x = unit(0.2, "cm"),
    plot.title = element_text(hjust = 0.5)
  )

```

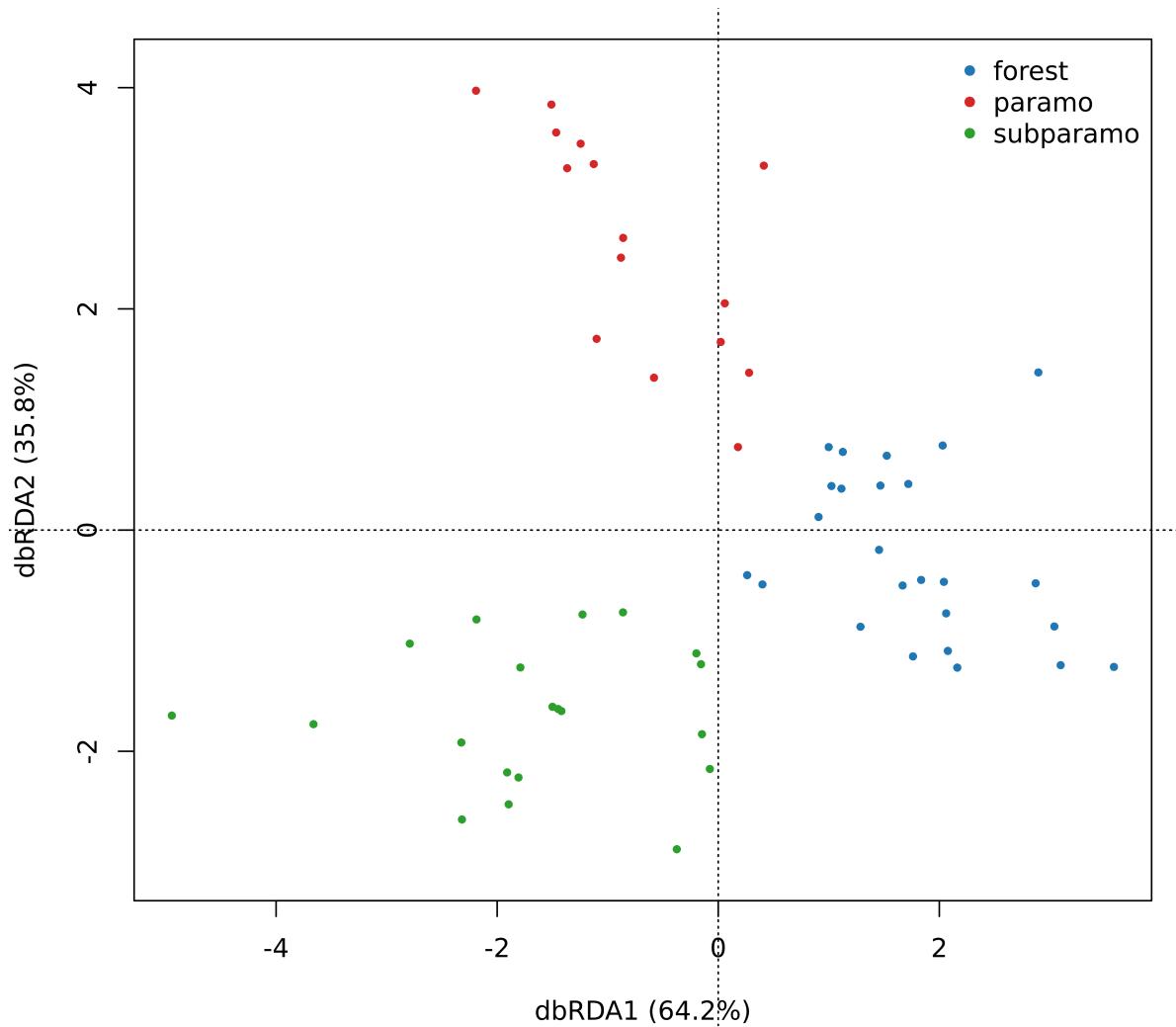
## Community composition analyses

This section provides detailed statistical analyses supporting the patterns of root-associated fungal community composition described in the main text. Several of these analyses form the basis of results reported in the main manuscript; here we document the full model specifications, test statistics, and complementary ordinations to ensure transparency and reproducibility.

We include distance-based redundancy analysis (dbRDA) based on robust Aitchison distances to assess habitat-associated compositional structure while conditioning on site. We further present PERMANOVA results from a balanced subset of samples (NV-DOM) used to test habitat and site effects under a controlled sampling design. In addition, we report a centroid-based turnover model quantifying the magnitude of community change across habitats using the full dataset.

### Multivariate ordination

Distance-based redundancy analysis (dbRDA) was conducted on robust Aitchison distances to test for habitat-associated compositional structure while conditioning on site.



**Figure S4:** Distance-based redundancy analysis (dbRDA) of root-associated fungal communities based on robust Aitchison distances. Points are colored by habitat; site was included as a conditioning variable. *Code: see “Code used to generate Figure S4 and Table S5” below.*

---

Table 5: Table S5a. dbRDA permutation tests (by term; 999 permutations).

Component	Df	Variance	F	Pr
habitat	2	22.718	1.408	0.001
Residual	54	435.501	NA	NA

Table 6: Table S5b. dbRDA permutation tests (by constrained axis; 999 permutations).

Axis	Df	Variance	F	Pr
CAP1	1	14.579	1.808	0.001
CAP2	1	8.139	1.009	0.470
Residual	54	435.501	NA	NA

```
#Running dbRDA with robust aitchison
library(phyloseq)
library(vegan)
library(knitr)

ps <- ps_individual #phyloseq object

X <- as(otu_table(ps), "matrix")
if (taxa_are_rows(ps)) X <- t(X)

sam <- data.frame(sample_data(ps))
sam$habitat <- factor(sam$habitat)
sam$site     <- factor(sam$site)

# dbRDA model: habitat constrained, conditioning on site
d_ait <- vegdist(otu_table(ps), binary=FALSE, method="robust.aitchison")
mod_ait <- capscale(d_ait ~ habitat + Condition(site), data = sam)

# Permutation tests (999 permutations)
set.seed(1)
a_terms   <- anova.cca(mod_ait, by = "terms", permutations = 999)
a_axis    <- anova.cca(mod_ait, by = "axis", permutations = 999)

a_terms
a_axis

tab_terms <- data.frame(
  Term = rownames(a_terms),
  Df = a_terms$Df,
  Variance = a_terms$Variance,
  F = a_terms$F,
  Pr = a_terms$`Pr(>F)`,
```

```

    row.names = NULL
)

tab_axis <- data.frame(
  Axis = rownames(a_axis),
  Df = a_axis$Df,
  Variance = a_axis$Variance,
  F = a_axis$F,
  Pr = a_axis$`Pr(>F)`,
  row.names = NULL
)

write.csv(tab_terms, "Table_S4_dbRDA_terms.csv", row.names = FALSE)
write.csv(tab_axis, "Table_S4_dbRDA_axes.csv", row.names = FALSE)

# Plotting Figure S4
eig <- mod_ait$CCA$eig
cap1_pct <- round(100 * eig[1] / sum(eig), 1)
cap2_pct <- round(100 * eig[2] / sum(eig), 1)

cols <- c(forest="steelblue", subparamo="seagreen3", paramo="firebrick")

png("FigS4_dbRDA.png", width=7.2, height=6.2, units="in", res=600)
par(mar=c(4.5,4.5,1,1))
plot(mod_ait, display="sites", type="n",
     xlab=paste0("dbRDA1 (", cap1_pct, "%)"),
     ylab=paste0("dbRDA2 (", cap2_pct, "%)"))
pts <- scores(mod_ait, display="sites")
points(pts, pch=16, cex=0.7, col=cols[as.character(sam$habitat)])
legend("topright", bty="n", legend=levels(sam$habitat),
       col=cols[levels(sam$habitat)], pch=16, pt.cex=0.9)
dev.off()

```

## PERMANOVA and dispersion analyses

Beta-dispersion tests indicated higher within-habitat variability in forests compared with sub-páramo and páramo communities ( $p < 0.001$ ), whereas dispersion did not differ among sites ( $p = 0.526$ ). Because the PERMANOVA was conducted on a balanced subset of samples, the significant effects of habitat, site, and their interaction are interpreted as reflecting genuine compositional differences rather than artifacts of unequal dispersion. *Code: see “Code used to*

generate Tables S6a-S6c” below.

Table 7: Table S6a. PERMANOVA (adonis2) results by term.

Source	df	SumOfSqs	R2	Pseudo.F	Pr..F.
site	1	742.078	0.056	1.786	0.001
habitat	2	1351.164	0.102	1.626	0.001
site:habitat	2	1241.269	0.093	1.493	0.001
Residual	24	9974.546	0.749	NA	NA
Total	29	13309.057	1.000	NA	NA

Table 8: Table S6b. Habitat beta-dispersion (distance to centroid). Permutation test p = 0.001.

habitat	avg_distance_to_centroid
forest	23.631
paramo	17.147
subparamo	17.794

Table 9: Table S6c. Site beta-dispersion permutation test.

test	permutations	F	p_value
betadisper_site	999	0.314	0.571

```
ps <- individual_ps #phyloseq obj

# Balanced subset: NV + DOM, remove NV_4
balanced_ps <- subset_samples(ps, site %in% c("DOM", "NV"))
balanced_ps <- subset_samples(balanced_ps, site_elevation != "NV_4")
balanced_ps <- prune_taxa(taxa_sums(balanced_ps) > 0, balanced_ps)

# Ensure samples are rows
X <- as(otu_table(balanced_ps), "matrix")
if (taxa_are_rows(balanced_ps)) X <- t(X)

sampledf <- data.frame(sample_data(balanced_ps))
sampledf$site      <- factor(sampledf$site)
sampledf$habitat <- factor(sampledf$habitat)
```

```

# Robust Aitchison distance
dists <- vegdist(X, method = "robust.aitchison")

# PERMANOVA (by terms)
set.seed(1)
perma <- adonis2(dists ~ site * habitat, by = "terms", data = sampledf, permutations = 999)

# Convert output to a clean CSV table
tab_perma <- data.frame(
  Source    = rownames(perma),
  df        = perma$Df,
  SumOfSqs = perma$SumOfSqs,
  R2        = perma$R2,
  `Pseudo-F` = perma$F,
  `Pr(>F)` = perma `$Pr(>F)` ,
  row.names = NULL
)

# Beta-dispersion (habitat & site)
bd_hab  <- betadisper(dists, sampledf$habitat)
bd_site <- betadisper(dists, sampledf$site)

set.seed(1)
pt_hab  <- permute(bd_hab, permutations = 999)
pt_site <- permute(bd_site, permutations = 999)

# Group dispersions
disp_hab <- data.frame(
  habitat = names(bd_hab$group.distances),
  avg_distance_to_centroid = as.numeric(bd_hab$group.distances),
  row.names = NULL
)

disp_site <- data.frame(
  site = names(bd_site$group.distances),
  avg_distance_to_centroid = as.numeric(bd_site$group.distances),
  row.names = NULL
)

test_hab <- data.frame(
  test = "betadisper_habitat",

```

```

permutations = 999,
F = unname(pt_hab$tab[1, "F"]),
p_value = unname(pt_hab$tab[1, "Pr(>F)"])
)

test_site <- data.frame(
  test = "betadisper_site",
  permutations = 999,
  F = unname(pt_site$tab[1, "F"]),
  p_value = unname(pt_site$tab[1, "Pr(>F)"])
)

# Output paths
out_dir <- "/data/lastexpansion/danieang/objects"
dir.create(out_dir, showWarnings = FALSE, recursive = TRUE)

write.csv(tab_perma, file.path(out_dir, "Table_S6_PERMANOVA.csv"), row.names = FALSE)
write.csv(disp_hab, file.path(out_dir, "Table_S6_dispersion_habitat.csv"), row.names = FALSE)
write.csv(disp_site, file.path(out_dir, "Table_S6_dispersion_site.csv"), row.names = FALSE)
write.csv(test_hab, file.path(out_dir, "Table_S6_betadisper_test_habitat.csv"), row.names = FALSE)
write.csv(test_site, file.path(out_dir, "Table_S6_betadisper_test_site.csv"), row.names = FALSE)

```

## Centroid-based turnover model

To quantify the magnitude of community turnover across habitats while accounting for spatial structure, we calculated the Aitchison distance of each sample to its site-specific centroid and modelled turnover as a function of habitat, site, and their interaction. Habitat had a strong effect on turnover magnitude, whereas the habitat  $\times$  site interaction was not significant, indicating consistent habitat-associated turnover across geographically isolated páramo regions (Table S7).

**Table S7:** Linear model results for centroid-based turnover.

Effect	df	Sum.of.Squares	Mean.Square	F.value	p.value
habitat	2	376.745	188.373	14.719	0.000
site	3	138.096	46.032	3.597	0.020
habitat:site	4	65.496	16.374	1.279	0.291
Residuals	50	639.914	12.798	NA	NA

```

library(phyloseq)
library(vegan)

ps <- individual_ps #phyloseq obj

# OTU table -> samples x taxa
X <- as(otu_table(ps), "matrix")
if (taxa_are_rows(ps)) X <- t(X)
X <- as.matrix(X)

# Robust CLR
rfy <- decostand(X, "rclr", MARGIN = 1)

# Metadata
meta <- data.frame(sample_data(ps), check.names = FALSE, stringsAsFactors = FALSE)
stopifnot(identical(rownames(rfy), rownames(meta)))

# Site centroids in CLR space
centroids <- rowsum(rfy, group = meta$site) / as.vector(table(meta$site))

# Euclidean distance to site centroid
euclid <- function(x, y) sqrt(sum((x - y)^2))

dist_centroid <- vapply(
  seq_len(nrow(rfy)),
  function(i) euclid(rfy[i, ], centroids[meta$site[i], ]),
  numeric(1)
)

# Model dataframe
df <- meta
df$dist_centroid <- dist_centroid
df$site <- factor(df$site)
df$habitat <- factor(df$habitat, levels = c("forest", "subparamo", "paramo"))

# Linear model (habitat-based)
mod_hab <- lm(dist_centroid ~ habitat * site, data = df)

# ANOVA table
a <- anova(mod_hab)

```

```

tab_s7 <- data.frame(
  Effect = rownames(a),
  df = a$Df,
  `Sum of Squares` = a$`Sum Sq`,
  `Mean Square` = a$`Mean Sq`,
  `F value` = a$`F value`,
  `p value` = a$`Pr(>F)`,
  row.names = NULL
)

# Output
out <- "/data/lastexpansion/danieang/objects/Table_S7_centroid_model.csv"
dir.create(dirname(out), showWarnings = FALSE, recursive = TRUE)
write.csv(tab_s7, out, row.names = FALSE)

```

### Abundance-based beta diversity partitioning

Bray–Curtis dissimilarities were partitioned into balanced variation and abundance gradients across taxonomic levels using the `betapart` package. Metrics were calculated from rarefied abundance data.

Table 11: Table S8. Bray–Curtis beta-diversity partitioning across taxonomic levels and sites.

Site	Taxonomic_Level	beta.BRAY.BAL	beta.BRAY.GRA	beta.BRAY
DOM	OTU	0.9748	0.0244	0.9993
DOM	Genus	0.9164	0.0780	0.9944
DOM	Order	0.7708	0.2166	0.9874
DOM	Class	0.4256	0.5418	0.9674
DOM	Phylum	0.2759	0.6859	0.9619
NV	OTU	0.9870	0.0124	0.9995
NV	Genus	0.9553	0.0415	0.9969
NV	Order	0.7846	0.2059	0.9905
NV	Class	0.4516	0.5238	0.9754
NV	Phylum	0.1470	0.8046	0.9516
BEL	OTU	0.9870	0.0125	0.9994
BEL	Genus	0.9371	0.0573	0.9945
BEL	Order	0.7916	0.1995	0.9911
BEL	Class	0.4611	0.5121	0.9733
BEL	Phylum	0.0258	0.9434	0.9692
MA	OTU	0.9431	0.0557	0.9989
MA	Genus	0.9305	0.0627	0.9932

Site	Taxonomic_Level	beta.BRAY.BAL	beta.BRAY.GRA	beta.BRAY
MA	Order	0.7966	0.1911	0.9877
MA	Class	0.5906	0.3750	0.9656
MA	Phylum	0.0000	0.9504	0.9505

### Beta-regression models

To quantify elevational trends in the relative abundance of dominant fungal lineages, we fitted beta-regression models to order-level relative read abundances. Models were fitted separately for each order using a logit link, with elevation as a continuous predictor and site included as a fixed effect. Relative abundances were calculated as proportions of total reads per sample and transformed using a Smithson–Verkuilen adjustment.

We focus on helotiales and sebacinales, the two most abundant orders, which also showed the only significant elevation responses. For both orders, fitted relationships and raw data are shown in Figure 2, while full model summaries are provided in Table S9. Elevation effects for the eight most abundant orders are summarized in Table S10.

Model diagnostics were examined to assess fit and identify potential violations of distributional assumptions. Diagnostic plots include Pearson residuals versus fitted values and elevation, quantile residual Q–Q plots, and leverage values (Figures S6–S7).

**Table S9:**

Table 12: Table S9a. Beta regression summary for helotiales (mean model coefficients and precision (phi)).

Model	Term	Estimate	SE	z	p
helotiales	(Intercept)	-3.2040	1.2559	-2.5512	0.0107
helotiales	elevation	0.0010	0.0004	2.3959	0.0166
helotiales	siteDOM	-0.7291	0.3651	-1.9969	0.0458
helotiales	siteMA	0.0050	0.3871	0.0130	0.9896
helotiales	siteNV	-0.8864	0.3346	-2.6494	0.0081
helotiales	(phi) (phi)	6.2698	1.0711	5.8534	0.0000

Table 13: Table S9b. Beta regression summary for sebacinales (mean model coefficients and precision (phi)).

Model	Term	Estimate	SE	z	p
sebacinales	(Intercept)	0.4169	1.3941	0.2990	0.7649
sebacinales	elevation	-0.0010	0.0005	-2.1320	0.0330

Model	Term	Estimate	SE	z	p
sebacinales	siteDOM	0.7196	0.4162	1.7289	0.0838
sebacinales	siteMA	0.3029	0.4603	0.6580	0.5105
sebacinales	siteNV	0.9779	0.3714	2.6333	0.0085
sebacinales	(phi) (phi)	10.1810	2.0241	5.0298	0.0000

---

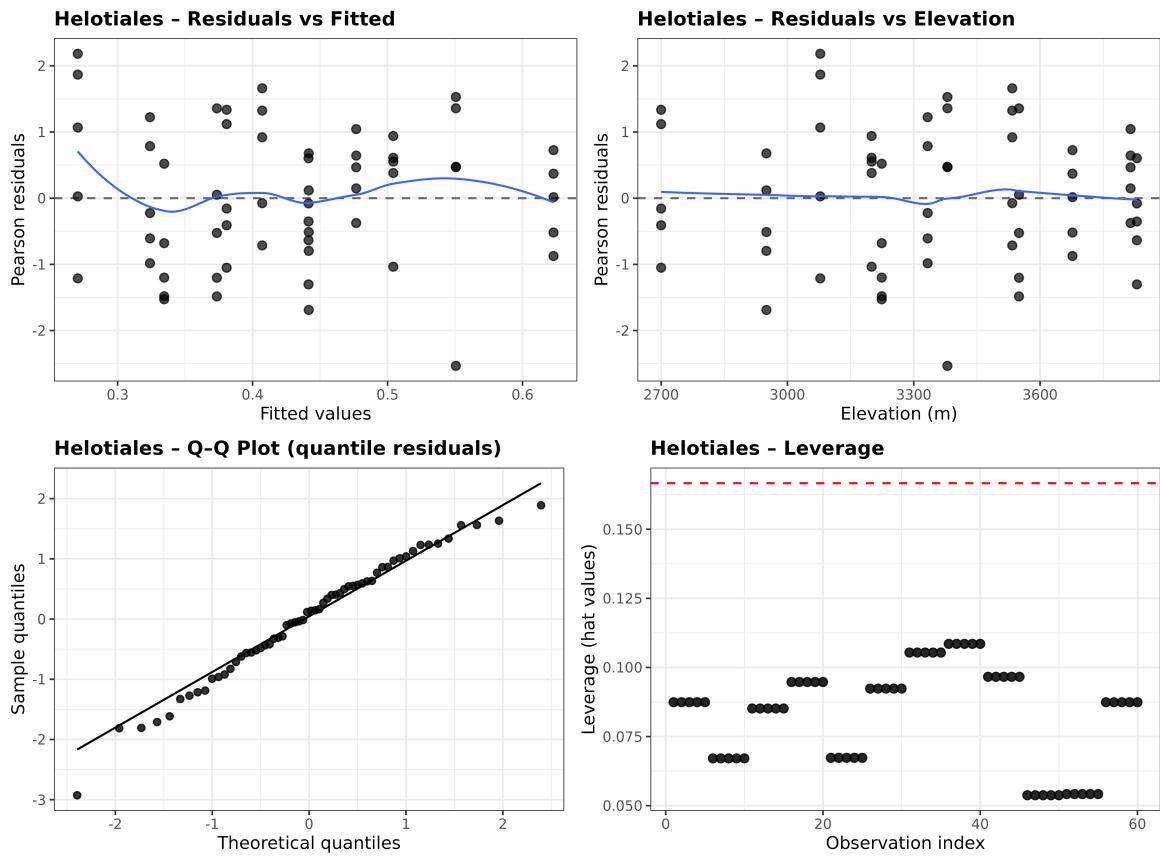
**Table S10:**

Table 14: Table S10. Elevation effects from beta regression models for the dominant orders (mean submodel), with pseudo-R<sup>2</sup>.

order	term	estimate	se	z	p_value	pseudo_R2
helotiales	elevation	1e-03	4e-04	2.3959	0.0166	0.2133
sebacinales	elevation	-1e-03	5e-04	-2.1320	0.0330	0.2261
chaetothyriales	elevation	1e-04	4e-04	0.2327	0.8160	0.0239
sclerococcales	elevation	7e-04	5e-04	1.3758	0.1689	0.1330
agaricales	elevation	-7e-04	4e-04	-1.7597	0.0785	0.0929
pleosporales	elevation	-4e-04	4e-04	-1.0668	0.2861	0.1528
hypocreales	elevation	-2e-04	3e-04	-0.7300	0.4654	0.2397
leotiales	elevation	-2e-04	3e-04	-0.6812	0.4957	0.2964

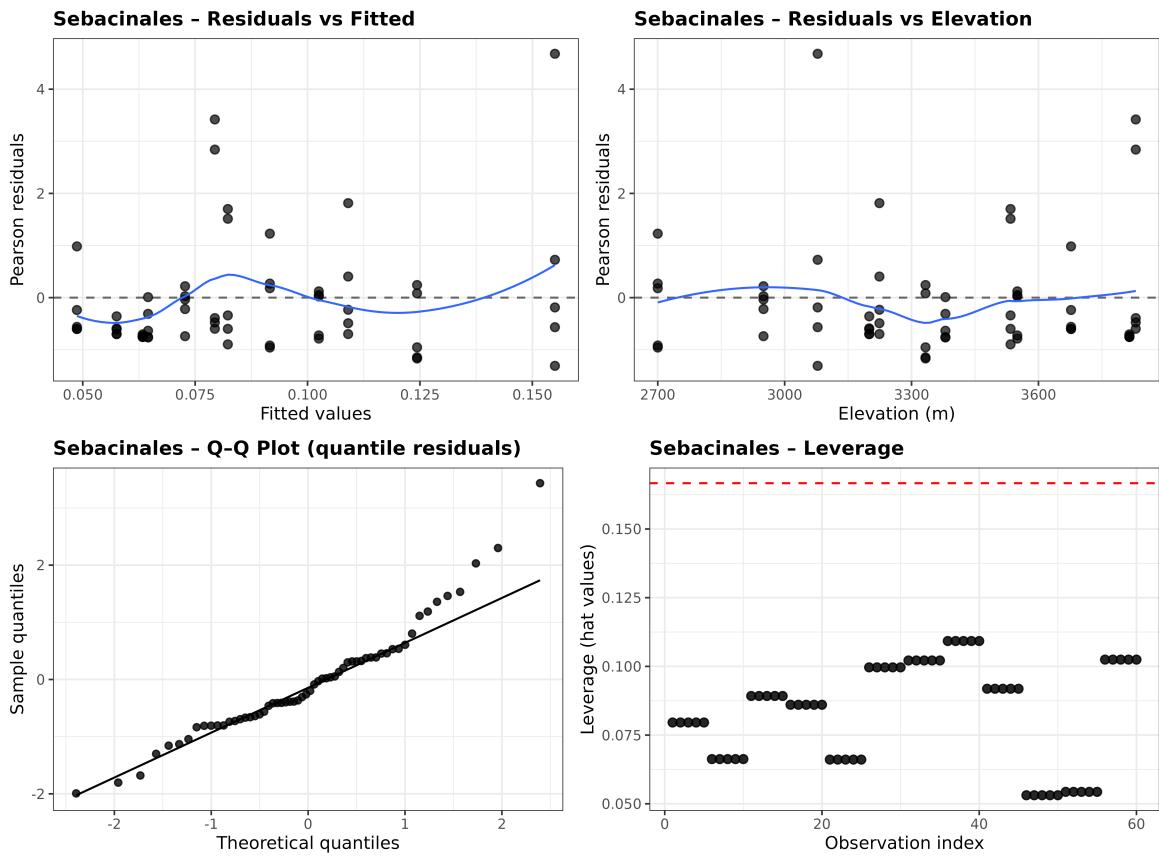
---

### Helotiales model diagnostics



**Figure S6:** Diagnostic plots for the beta-regression model of Helotiales. *Code: see “Beta-regression models” below.*

### Sebacinales model diagnostics



**Figure S7:** Diagnostic plots for the beta-regression model of Sebacinales. *Code: see “Beta-regression models” below.*

```
# This code fits beta-regression models for dominant fungal orders,
# generates Tables S9-S10, and exports Figures 2a and S6-S7 as PNG files.
```

```
library(phyloseq)
library(dplyr)
library(tidyr)
library(ggplot2)
library(betareg)
library(patchwork)

out_fig_dir <- "/data/lastexpansion/danieang/objects"
out_tab_dir <- "/data/lastexpansion/danieang/objects"
```

```

set.seed(1)

## ---- Helper: build beta-regression dataframe for an Order ----
make_beta_df_from_order <- function(ps, order_name) {

  otu <- as(otu_table(ps), "matrix")
  if (!taxa_are_rows(ps)) otu <- t(otu) # taxa x samples

  sd <- as.data.frame(sample_data(ps))
  sd <- sd[colnames(otu), , drop = FALSE] # force alignment

  tax <- as.data.frame(tax_table(ps), stringsAsFactors = FALSE)
  stopifnot("Order" %in% names(tax))

  order_vec <- tolower(as.character(tax$Order))
  order_vec[is.na(order_vec)] <- ""

  otu_ids <- rownames(tax)[grepl(tolower(order_name), order_vec, fixed = TRUE)]
  if (length(otu_ids) == 0) stop(paste("No OTUs found for", order_name))

  counts <- colSums(otu[otu_ids, , drop = FALSE])
  total <- colSums(otu)

  samps <- colnames(otu)

  df <- data.frame(
    sample      = samps,
    count       = as.numeric(counts),
    total       = as.numeric(total),
    prop_raw   = ifelse(total > 0, counts / total, NA_real_),
    site        = sd[, "site", drop = TRUE],
    elevation   = sd[, "elevation", drop = TRUE],
    stringsAsFactors = FALSE
  )

  # Smithson-Verkuilen adjustment to (0,1)
  n <- nrow(df)
  df$Proportion <- (df$prop_raw * (n - 1) + 0.5) / n

  df$site <- factor(df$site)
  df$elevation <- as.numeric(df$elevation)
}

```

```

    df
}

## ---- Helper: tidy coefficient table for Table S8 ----
tidy_betareg_mean <- function(mod, model_name) {
  sm <- summary(mod)
  co <- as.data.frame(sm$coefficients$mean)
  co$Term <- rownames(co)
  rownames(co) <- NULL
  names(co) <- c("Estimate", "SE", "z", "p", "Term")

  # move Term first + add model label
  co <- co %>%
    select(Term, Estimate, SE, z, p) %>%
    mutate(Model = model_name, .before = 1)

  # also add phi (precision) row, for completeness
  phi <- as.data.frame(sm$coefficients$precision)
  phi$Term <- rownames(phi); rownames(phi) <- NULL
  names(phi) <- c("Estimate", "SE", "z", "p", "Term")
  phi <- phi %>%
    select(Term, Estimate, SE, z, p) %>%
    mutate(Model = model_name, .before = 1) %>%
    mutate(Term = paste0(Term, " (phi)"))

  bind_rows(co, phi)
}

## ---- Helper: elevation-only summary across orders for Table S9 ----
extract_elev <- function(mod, order_name) {
  sm <- summary(mod)
  co <- sm$coefficients$mean
  if (!("elevation" %in% rownames(co))) return(NULL)
  data.frame(
    order = order_name,
    term = "elevation",
    estimate = co["elevation", "Estimate"],
    se = co["elevation", "Std. Error"],
    z = co["elevation", "z value"],
    p_value = co["elevation", "Pr(>|z|)"],
    pseudo_R2 = sm$pseudo.r.squared,
    stringsAsFactors = FALSE
}

```

```

    )

}

## ---- 1) Fit models ----
orders_target <- c(
  "helotiales",
  "sebacinales",
  "chaetothyriales",
  "sclerococcales",
  "agaricales",
  "pleosporales",
  "hypocreales",
  "leotiales"
)

df_list <- setNames(vector("list", length(orders_target)), orders_target)
mod_list <- setNames(vector("list", length(orders_target)), orders_target)

for (ord in orders_target) {
  message("Building DF + fitting: ", ord)
  df_list[[ord]] <- make_beta_df_from_order(ps, ord)
  mod_list[[ord]] <- betareg(Proportion ~ elevation + site, data = df_list[[ord]])
}

## ---- 2) Table S9: helotiales + sebacinales full model tables ----
tab_s9_helot <- tidy_betareg_mean(mod_list[["helotiales"]], "helotiales")
tab_s9_seba <- tidy_betareg_mean(mod_list[["sebacinales"]], "sebacinales")

write.csv(tab_s9_helot,
          file.path(out_tab_dir, "Table_S9_betareg_helotiales.csv"),
          row.names = FALSE)
write.csv(tab_s9_seba,
          file.path(out_tab_dir, "Table_S9_betareg_sebacinales.csv"),
          row.names = FALSE)

## ---- 3) Table S10: elevation effects across orders ----
tab_s10 <- bind_rows(lapply(names(mod_list), \((ord) extract_elev(mod_list[[ord]], ord)))
write.csv(tab_s10,
          file.path(out_tab_dir, "Table_S10_betareg_elevation_effects.csv"),
          row.names = FALSE)

```

```

## ---- 4) Beta regression plots for Helotiales + Sebacinales ----

# helper
inv_logit <- function(x) 1 / (1 + exp(-x))

plot_betareg_with_raw <- function(model, data, label_prefix = "helotiales") {

  # Ensure required columns exist
  stopifnot(all(c("site", "elevation", "prop_raw") %in% names(data)))

  data <- data[complete.cases(data[, c("site", "elevation", "prop_raw")]), , drop = FALSE]

  # Align site factor levels to model
  if (!is.null(model$xlevels$mean$site)) {
    data$site <- factor(as.character(data$site), levels = model$xlevels$mean$site)
  } else {
    data$site <- factor(data$site)
  }

  # Drop any rows that became NA due to level mismatch
  data <- data[!is.na(data$site), , drop = FALSE]

  one_level_site <- (nlevels(droplevels(data$site)) < 2)

  # Build prediction grid by site across observed elevation range
  newdat <- data %>%
    dplyr::select(site, elevation) %>%
    dplyr::distinct() %>%
    dplyr::group_by(site) %>%
    dplyr::summarize(
      elev_min = min(elevation, na.rm = TRUE),
      elev_max = max(elevation, na.rm = TRUE),
      .groups = "drop"
    ) %>%
    dplyr::rowwise() %>%
    dplyr::mutate(elevation = list(seq(elev_min, elev_max, length.out = 100))) %>%
    tidyverse::unnest(elevation) %>%
    dplyr::select(site, elevation) %>%
    dplyr::ungroup()

  # Re-apply model levels to newdat site
  if (!is.null(model$xlevels$mean$site)) {

```

```

newdat$site <- factor(as.character(newdat$site), levels = model$xlevels$mean$site)
} else {
  newdat$site <- factor(newdat$site)
}

# If we only have 1 site level, disable contrasts to prevent the error
if (one_level_site) {
  contrasts(newdat$site) <- NULL
}

# Design matrix for mean submodel
mean_terms <- stats::delete.response(stats::terms(model, model = "mean"))
X <- model.matrix(mean_terms, newdat)

beta <- coef(model, model = "mean")
V <- try(vcov(model, model = "mean"), silent = TRUE)
if (inherits(V, "try-error")) V <- vcov(model)

eta     <- as.vector(X %*% beta)
se_eta <- sqrt(rowSums((X %*% V) * X))

newdat$fit <- inv_logit(eta)
newdat$lwr <- inv_logit(eta - 1.96 * se_eta)
newdat$upr <- inv_logit(eta + 1.96 * se_eta)

# Plot
ggplot() +
  geom_point(
    data = data,
    aes(x = elevation, y = prop_raw, colour = site),
    size = 2, alpha = 0.7
  ) +
  geom_ribbon(
    data = newdat,
    aes(x = elevation, ymin = lwr, ymax = upr, fill = site, group = site),
    alpha = 0.15, colour = NA
  ) +
  geom_line(
    data = newdat,
    aes(x = elevation, y = fit, colour = site, group = site),
    linewidth = 0.8
  ) +

```

```

    scale_y_continuous("Relative abundance (raw proportion)", limits = c(0, 1)) +
    scale_x_continuous("Elevation (m)") +
    ggtitle(label_prefix) +
    theme_bw(base_size = 10) +
    theme(
      plot.title = element_text(face = "bold", size = 12, hjust = 0),
      legend.position = "bottom"
    )
  }

# Build plots
p_helot <- plot_betareg_with_raw(mod_list[["helotiales"]], df_list[["helotiales"]], "A) helotiales")
p_seba <- plot_betareg_with_raw(mod_list[["sebacinales"]], df_list[["sebacinales"]], "B) sebacinales")

# Combine and save
panel_beta <- (p_helot | p_seba) + patchwork::plot_layout(guides = "collect")

ggsave(
  filename = file.path(out_fig_dir, "Fig_2a_beta_regression_panel.png"),
  plot = panel_beta,
  width = 12, height = 5.5, dpi = 800
)

## ---- 5) Diagnostic plots for Helotiales + Sebacinales ----

plot_betareg_diagnostics <- function(model, data, label_prefix = "Helotiales") {

  # Ensure alignment (betareg expects the same row order it was fitted with)
  data <- data[complete.cases(data[, c("elevation", "site", "Proportion")]), , drop = FALSE]

  # residuals & fitted
  res_pear <- residuals(model, type = "pearson")
  res_quant <- residuals(model, type = "quantile") # approx N(0,1)
  fit       <- fitted(model)
  hatv     <- hatvalues(model)

  diagdf <- data.frame(
    fitted     = fit,
    res_pear   = res_pear,
    res_quant  = res_quant,
    elevation = data$elevation,
}

```

```

    site      = as.factor(data$site),
    hat       = hatv,
    idx       = seq_len(nrow(data))
  )

# (1) Residuals vs Fitted
p1 <- ggplot(diagdf, aes(fitted, res_pear)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey40") +
  geom_point(alpha = 0.7, size = 1.8) +
  geom_smooth(method = "loess", se = FALSE, linewidth = 0.5) +
  labs(x = "Fitted values", y = "Pearson residuals",
       title = paste(label_prefix, "- Residuals vs Fitted")) +
  theme_bw(base_size = 9)

# (2) Residuals vs Elevation
p2 <- ggplot(diagdf, aes(elevation, res_pear)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey40") +
  geom_point(alpha = 0.7, size = 1.8) +
  geom_smooth(method = "loess", se = FALSE, linewidth = 0.5) +
  labs(x = "Elevation (m)", y = "Pearson residuals",
       title = paste(label_prefix, "- Residuals vs Elevation")) +
  theme_bw(base_size = 9)

# (3) Q-Q plot of quantile residuals
p3 <- ggplot(diagdf, aes(sample = res_quant)) +
  stat_qq(size = 1.4, alpha = 0.8) +
  stat_qq_line() +
  labs(title = paste(label_prefix, "- Q-Q Plot (quantile residuals")),
       x = "Theoretical quantiles", y = "Sample quantiles") +
  theme_bw(base_size = 9)

# (4) Leverage
hat_thr <- 2 * mean(diagdf$hat, na.rm = TRUE)
p4 <- ggplot(diagdf, aes(idx, hat)) +
  geom_point(alpha = 0.85, size = 1.8) +
  geom_hline(yintercept = hat_thr, color = "red", linetype = "dashed") +
  labs(x = "Observation index", y = "Leverage (hat values)",
       title = paste(label_prefix, "- Leverage")) +
  theme_bw(base_size = 9)

(p1 | p2) / (p3 | p4) +
  patchwork::plot_annotation(title = paste(label_prefix, "model diagnostics")) &

```

```

    theme(plot.title = element_text(face = "bold", size = 10, hjust = 0))
}

# Use your existing objects from Steps 1-3
diag_helot <- plot_betareg_diagnostics(mod_list[["helotiales"]], df_list[["helotiales"]]),
diag_seba <- plot_betareg_diagnostics(mod_list[["sebacinales"]], df_list[["sebacinales"]]),

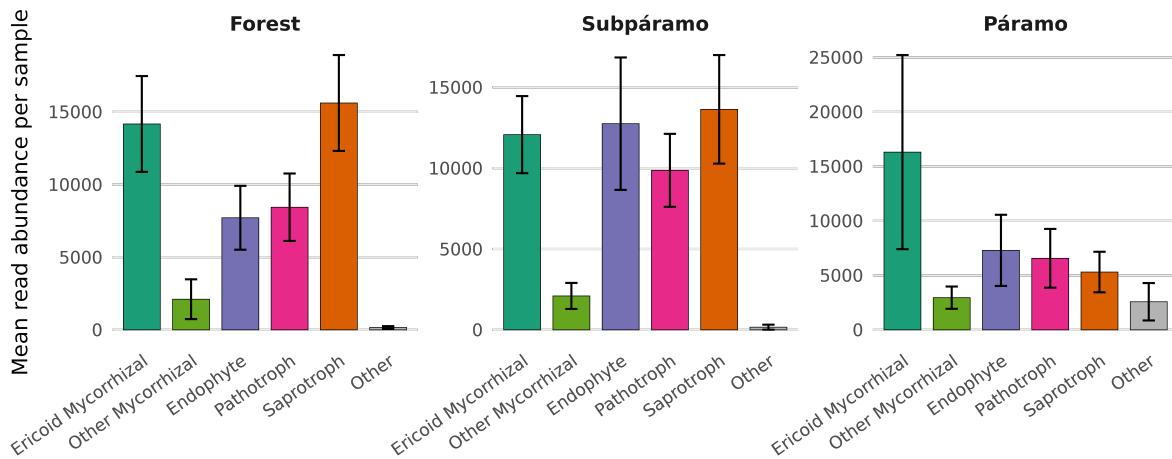
ggsave(
  filename = file.path(out_fig_dir, "Fig_S6_betareg_diagnostics_helotiales.png"),
  plot = diag_helot, width = 8.5, height = 6.5, dpi = 600
)

ggsave(
  filename = file.path(out_fig_dir, "Fig_S7_betareg_diagnostics_sebacinales.png"),
  plot = diag_seba, width = 8.5, height = 6.5, dpi = 600
)

```

## Functional annotation with FUNGuild

Functional guilds from root samples were assigned using FUNGuild based on the curated taxonomic strings exported from the phyloseq object. OTUs with a valid confidence ranking in the FUNGuild output were retained for downstream summaries. Guild labels were additionally collapsed into a primary-guild hierarchy (Ericoid mycorrhizal > other mycorrhizal > endophyte > pathotroph > saprotroph > other) to simplify interpretation. Figure S8 summarizes mean guild-associated read abundance per sample across habitats.



**Figure S8:** Mean relative read abundance per sample of fungal functional guilds across forest, subpáramo, and páramo habitats, based on FUNGuild annotations. Only root samples. *Code:*

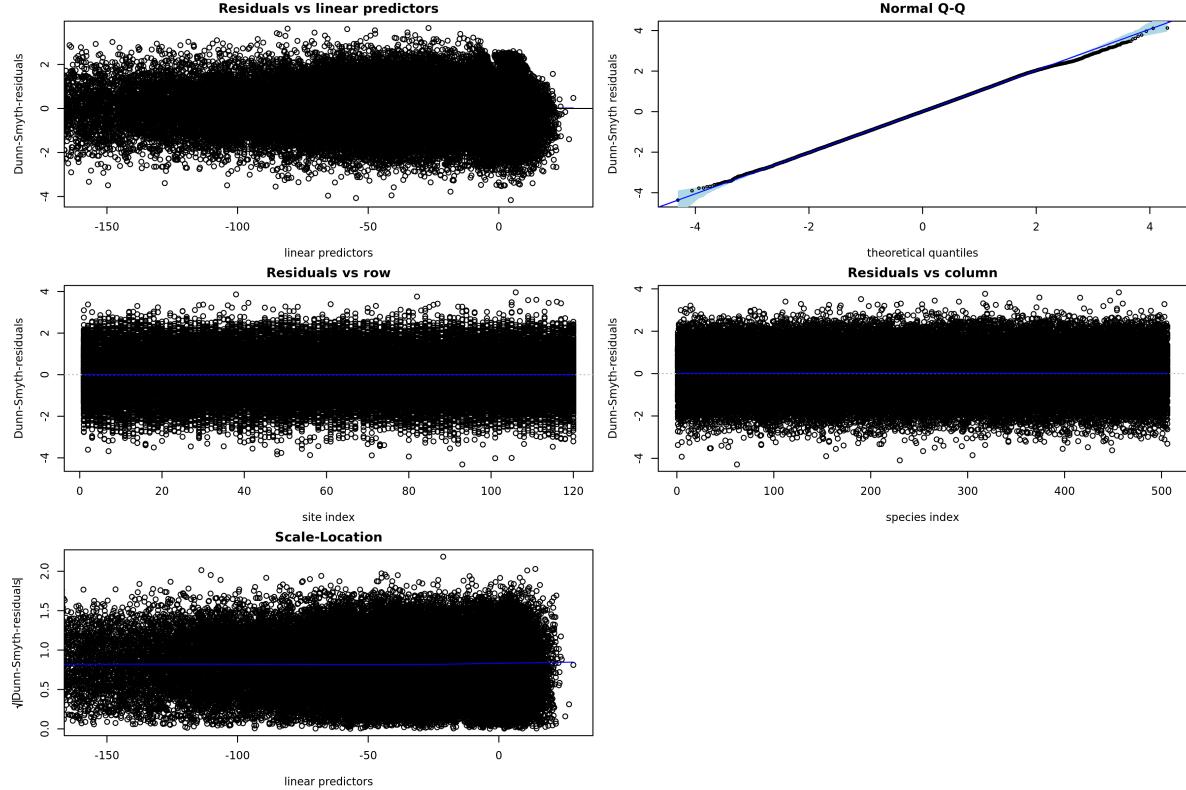
see “*FUNGuild functional annotation*” below.

```
# Functional guild assignment was performed using FUNGuild prior to figure generation.  
#  
# The full annotation workflow is available in the GitHub repository:  
#   scripts/funguild_script.R  
#  
# Input:  
#   - Taxonomically annotated ASV/OTU table  
#  
# Output:  
#   - Table of guild assignments per taxon  
#   - Mean read abundances per sample and habitat  
#  
# Figure S8 was generated from these outputs and is included above as a PNG.
```

## Generalized Linear Latent Variable Model (GLLVM)

We modelled OTU counts using a generalized linear latent variable model (GLLVM; negative binomial distribution, log link), with habitat (forest, subpáramo, páramo) as a fixed effect, log library size as an offset, and random intercepts for site and individual plants (Unique\_ID). One latent variable was included to account for residual correlation among taxa. The final model (log-likelihood =  $-37,616$ ; AIC = 80,295) provided the best balance between fit and parsimony. It substantially outperformed earlier versions with nested random effects (AIC = 186,832), and attempts to increase the number of latent variables to two worsened performance ( $\Delta\text{AIC} = 12,700$ ) and produced a singular information matrix, indicating over-parameterization. Simplifying the random-effects structure and filtering rare taxa also improved convergence and reduced overdispersion while retaining ecological signal.

Model fit was evaluated using Dunn–Smyth residual diagnostics (Fig. S9). Habitat coefficients ( ) were extracted for each OTU from the fixed-effects matrix. To obtain genus-level responses, OTUs were restricted to those assigned to a genus (excluding incertae sedis) and weighted by their total read abundance across samples. For each genus, we computed the abundance-weighted mean for páramo and subpáramo relative to forest. Uncertainty was summarized using a weighted bootstrap (R = 500) resampling OTUs within genera with probabilities proportional to total abundance; 2.5–97.5% quantiles were used as 95% confidence intervals. These summaries were used to generate the heatmap and barplot shown in Figure 4. The full OTU-level coefficient table is provided in Data\_S2\_GLLVM\_table.csv.



**Figure S9:** Diagnostic plots for the negative-binomial GLLVM (Dunn–Smyth residuals). Code: see “*GLLVM model fitting + coefficient summaries*” below.

For transparency, we provide a short preview of the genus-level GLLVM summaries below. The complete OTU- and genus-level coefficient table is available as Data S2 (Data\_S2\_GLLVM\_table.csv).

Table 15: Data S2 (preview). Selected columns from the GLLVM genus-level summary. The full table with all metrics is provided as a CSV file.

Order	Genus	n_OTUs	mean_betaaw_pareano_beta	pvalue_enriched_pareano_enriched_subparamo_w	
Coniochaetales	Coniochaeta	2	254.379	-25.536 2.04801e+115	-1.000000e+02
Helotiales	Acephala	1	78.144	69.862 8.663205e+35	2.190394e+32
Helotiales	Hyaloscypha	3	26.861	29.053 3.743955e+14	6.764441e+13
Helotiales	Pezicula	5	26.640	29.280 1.824961e+09	-9.894700e+01
Helotiales	Lachnum	3	-1.529	9.013 1.016873e+08	2.107609e+04
Sclerococcales	Sclerococcum	4	24.669	-8.648 2.210442e+06	-9.989400e+01
Mortierellales	Mortierella	5	-7.650	-34.063 1.011728e+04	-1.000000e+02
Helotiales	Gyoerffyella	3	2.739	0.188 1.021700e+01	-9.967900e+01

Order	Genus	n_OTUs	mean_beta_w	pareano_beta_subparamo	beta_pacanched_parcamo	enriched_parcamo	enriched_subparamo	w
Ostropales	Cryptodiscus	3	-15.238	12.611	-	6.019926e+04		
					1.947200e+01			
Helotiales	Pezoloma	2	-6.125	-4.096	-	-9.684900e+01		
					9.964100e+01			

```

# GLLVM analysis was executed on the server due to computational cost.
# The complete workflow (data filtering, model fitting, diagnostics, and genus-level summaries)
#   scripts/gllvm_analysis.R
#
# Key outputs saved:
#   - figures/Fig_S9_GLLVM_diagnostics.png
#   - tables/File_S2_GLLVM_table.csv
#   - objects/fit_nb_2.rds # fitted model object

```

## Diversity analyses with iNEXT3D

We quantified habitat-associated patterns in taxonomic diversity (TD) and phylogenetic diversity (PD) using iNEXT3D and Hill numbers ( $q = 0, 1, 2$ ). Diversity estimates were computed from incidence (presence/absence) matrices at the habitat level and evaluated under rarefaction/extrapolation with bootstrap confidence intervals ( $nboot = 500$ ). Phylogenetic diversity was estimated as meanPD using a pruned phylogeny matched to the observed taxa. Diversity curves are presented in the main text (Fig. 4), while asymptotic summaries are provided here (Tables S11–S12).

**Table S11:** Taxonomic diversity summary

Table 16: Table S11. Taxonomic diversity (TD) summary from iNEXT3D across habitats ( $q = 0, 1, 2$ ;  $nboot = 500$ ). Observed and asymptotic estimates are reported with standard errors, 95% confidence intervals, and sample coverage at observed effort and at double effort.

Assemblage	qTD	TD_obs	TD_asy	s.e.	qTD.LCL	qTD.UCL
Forest	Species richness	2521.000	4581.983	125.848	4335.326	4828.640
Forest	Shannon diversity	1810.823	2927.463	51.242	2827.030	3027.896
Forest	Simpson diversity	1158.528	1467.726	34.168	1400.758	1534.695
Páramo	Species richness	901.000	1945.702	95.214	1759.086	2132.317
Páramo	Shannon diversity	693.768	1316.325	43.456	1231.153	1401.497
Páramo	Simpson diversity	474.571	661.005	30.584	601.061	720.948
Subpáramo	Species richness	1533.000	3059.820	108.142	2847.866	3271.775

Assemblage	qTD	TD_obs	TD_asy	s.e.	qTD.LCL	qTD.UCL
Subpáramo	Shannon diversity	1225.295	2341.860	58.351	2227.494	2456.226
Subpáramo	Simpson diversity	846.718	1247.075	49.323	1150.403	1343.747

**Table S12:** Phylogenetic diversity summary

```
#| echo: false
library(knitr)

tab_s12 <- read.csv("tables/Table_S12_iNEXT3D_phylogenetic_diversity.csv")

kable(
  tab_s12,
  digits = 3,
  caption = "Table S12. Phylogenetic diversity (PD; meanPD) summary from iNEXT3D across habitats"
)
```

Table 17: Table S12. Phylogenetic diversity (PD; meanPD) summary from iNEXT3D across habitats ( $q = 0, 1, 2$ ;  $n_{boot} = 500$ ), including coverage and effective lineage estimates under rarefaction/extrapolation.

Habitat	qPD	PD_obs	PD_asy	s.e.	qPD.LCL	qPD.UCL	Reftime	Type
Forest	$q = 0$ PD	185.553	283.343	7.643	268.362	298.324	0.873	meanPD
Forest	$q = 1$ PD	96.045	115.843	1.427	113.045	118.641	0.873	meanPD
Forest	$q = 2$ PD	53.644	56.697	0.585	55.551	57.842	0.873	meanPD
Páramo	$q = 0$ PD	81.971	134.076	6.501	121.334	146.818	0.873	meanPD
Páramo	$q = 1$ PD	46.984	60.900	1.321	58.310	63.489	0.873	meanPD
Páramo	$q = 2$ PD	27.101	29.356	0.529	28.320	30.392	0.873	meanPD
Subpáramo	$q = 0$ PD	132.936	213.016	7.909	197.514	228.518	0.873	meanPD
Subpáramo	$q = 1$ PD	75.157	98.360	1.843	94.748	101.973	0.873	meanPD
Subpáramo	$q = 2$ PD	39.842	43.233	0.715	41.831	44.635	0.873	meanPD

```

library(phyloseq)
library(dplyr)
library(tidyr)
library(ggplot2)
library(iNEXT.3D)
library(ape)
library(knitr)
library(phangorn)

set.seed(1)

## ---- shared habitat labeling ----
labs_map <- c(forest = "Forest", subparamo = "Subpáramo", paramo = "Páramo")
lvl_order <- c("Forest", "Subpáramo", "Páramo")

relabel_inext <- function(x) {
  # iNEXT3D objects store multiple data.frames with "Assemblage"
  f <- function(df){
    if (!("Assemblage" %in% names(df))) return(df)
    df$Assemblage <- as.character(df$Assemblage)
    df$Assemblage <- labs_map[df$Assemblage]
    df$Assemblage <- factor(df$Assemblage, levels = lvl_order)
    df
  }
  x$TDInfo <- f(as.data.frame(x$TDInfo))
  x$TDAsyEst <- f(as.data.frame(x$TDAsyEst))
  if (!is.null(x$TDiNextEst)) {
    for (nm in names(x$TDiNextEst)) x$TDiNextEst[[nm]] <- f(as.data.frame(x$TDiNextEst[[nm]]))
  }
  x
}

## =====
## A) TAXONOMIC DIVERSITY (TD) ---- Table S10 + TD plot
## =====

ps_td <- tree_ps

stopifnot(inherits(ps_td, "phyloseq"))

```

```

sd <- as.data.frame(sample_data(ps_td))
stopifnot("habitat" %in% names(sd))

# build taxa x samples incidence matrix
OTU <- as(otu_table(ps_td), "matrix")
if (!taxa_are_rows(ps_td)) OTU <- t(OTU)    # taxa x samples

# split samples by habitat
inc_by_hab_td <- lapply(split(rownames(sd), sd$habitat), function(samps){
  M <- OTU[, colnames(OTU) %in% samps, drop = FALSE]
  M[M > 0] <- 1
  storage.mode(M) <- "numeric"
  M
})

# keep only expected habitats and non-empty
inc_by_hab_td <- inc_by_hab_td[names(inc_by_hab_td) %in% names(labs_map)]
inc_by_hab_td <- inc_by_hab_td[vapply(inc_by_hab_td, function(M) nrow(M) > 0 && ncol(M) > 0,
stopifnot(length(inc_by_hab_td) >= 2)

out_TD <- iNEXT3D(
  data      = inc_by_hab_td,
  diversity = "TD",
  q         = c(0,1,2),
  datatype  = "incidence_raw",
  nboot     = 500
)

# relabel habitat names + ordering for plotting/tables
out_TD <- relabel_inext(out_TD)

# ---- Figure (TD curves) ----
p_TD <- ggiNEXT3D(out_TD, type = 1, facet.var = "Order.q") +
  labs(x = "Sampling units", y = "Taxonomic diversity") +
  theme_minimal(base_size = 14) +
  theme(legend.title = element_blank())

ggsave(
  filename = file.path(out_fig_dir, "Fig_4B_iNEXT3D_TD.png"),
  plot = p_TD, width = 10, height = 6, dpi = 800, bg = "white"
)

```

```

# ---- Table S10 (TD summary)
tab_s10 <- as.data.frame(out_TD$TDAsyEst)

# Optional: make "Diversity_order" readable
tab_s10 <- tab_s10 %>%
  mutate(
    Diversity_order = case_when(
      qTD == 0 ~ "Species richness",
      qTD == 1 ~ "Shannon diversity",
      qTD == 2 ~ "Simpson diversity",
      TRUE ~ as.character(qTD)
    )
  ) %>%
  rename(
    Habitat = Assemblage
  )

write.csv(
  tab_s10,
  file.path(out_tab_dir, "Table_S11_iNEXT3D_taxonomic_diversity.csv"),
  row.names = FALSE
)

## =====
## B) PHYLOGENETIC DIVERSITY (PD) ---- Table S11 + PD plot
## =====

ps_pd <- tree_ps #phyloseq object with phy tree
stopifnot(inherits(ps_pd, "phyloseq"))
stopifnot(!is.null(phy_tree(ps_pd, errorIfNULL = FALSE)))

sd2 <- as.data.frame(sample_data(ps_pd))
stopifnot("habitat" %in% names(sd2))

# Fix mislabel if present
sd2$habitat <- as.character(sd2$habitat)
sd2$habitat[sd2$habitat == "pasture"] <- "forest"
sample_data(ps_pd)$habitat <- factor(sd2$habitat)

# build taxa x samples incidence
X <- as(otu_table(ps_pd), "matrix")

```

```

if (!taxa_are_rows(ps_pd)) X <- t(X) # taxa x samples

sd2 <- as.data.frame(sample_data(ps_pd))

inc_by_hab_pd <- lapply(split(rownames(sd2), sd2$habitat), function(samps){
  M <- X[, colnames(X) %in% samps, drop = FALSE]
  M[M > 0] <- 1
  storage.mode(M) <- "numeric"
  M
})

inc_by_hab_pd <- inc_by_hab_pd[names(inc_by_hab_pd) %in% names(labs_map)]
inc_by_hab_pd <- inc_by_hab_pd[vapply(inc_by_hab_pd, function(M) nrow(M) > 0 && ncol(M) > 0, logical(1))]

# Tree: prune to observed taxa in incidence matrices
tr <- phy_tree(ps_pd)
keep_tips <- intersect(tr$tip.label, unique(unlist(lapply(inc_by_hab_pd, rownames))))
stopifnot(length(keep_tips) >= 2)

tr2 <- ape::keep.tip(tr, keep_tips)
tr2$node.label <- NULL
if (!ape::is.rooted(tr2)) tr2 <- phangorn::midpoint(tr2)

# Align matrices to final tip set (same taxa order)
tipset <- tr2$tip.label
inc_by_hab_pd <- lapply(inc_by_hab_pd, function(M) M[rownames(M) %in% tipset, , drop = FALSE])
inc_by_hab_pd <- inc_by_hab_pd[vapply(inc_by_hab_pd, function(M) nrow(M) > 0 && ncol(M) > 0, logical(1))]

out_PD <- iNEXT3D(
  data      = inc_by_hab_pd,
  diversity = "PD",
  q         = c(0,1,2),
  datatype  = "incidence_raw",
  nboot     = 500,
  PDtree    = tr2,
  PDtype    = "meanPD"
)

# Plot
p_PD <- ggiNEXT3D(out_PD, type = 1, facet.var = "Order.q") +
  labs(x = "Sampling units", y = "Phylogenetic diversity (meanPD)") +
  theme_minimal(base_size = 14) +

```

```

theme(legend.title = element_blank())

ggsave(
  filename = file.path(out_fig_dir, "Fig_4A_iNEXT3D_PD.png"),
  plot = p_PD, width = 10, height = 6, dpi = 800, bg = "white"
)

# Table S11: PD summary lives in $PDAsyEst (same logic as TD)
tab_s11 <- as.data.frame(out_PD$PDAsyEst) %>%
  rename(Habitat = Assemblage)

# Relabel habitats to capitalized versions (same mapping)
tab_s11$Habitat <- as.character(tab_s11$Habitat)
tab_s11$Habitat <- labs_map[tab_s11$Habitat]
tab_s11$Habitat <- factor(tab_s11$Habitat, levels = lvl_order)

write.csv(
  tab_s11,
  file.path(out_tab_dir, "Table_S12_iNEXT3D_phylogenetic_diversity.csv"),
  row.names = FALSE
)

```

## Nearest Taxon Index (NTI) and Net Relatedness Index (NRI)

Phylogenetic community structure was quantified using Net Relatedness Index (NRI; based on MPD) and Nearest Taxon Index (NTI; based on MNTD), calculated under a taxa-label null model with 999 randomizations using both abundance-weighted (aw) and presence-absence (pa) versions. Habitat-level summaries are reported as median (IQR) (Table S13). Habitat and site effects on abundance-weighted metrics were evaluated with linear models (habitat + site; Table S14). Model diagnostics for the NTI\_aw model are available in Fig. S10.

**Table S13:** NTI and NRI across habitats.

Table 18: Table S13. Median (IQR) of NRI and NTI across habitats, calculated using abundance-weighted (aw) and presence-absence (pa) metrics. Positive values indicate phylogenetic clustering relative to null expectations.

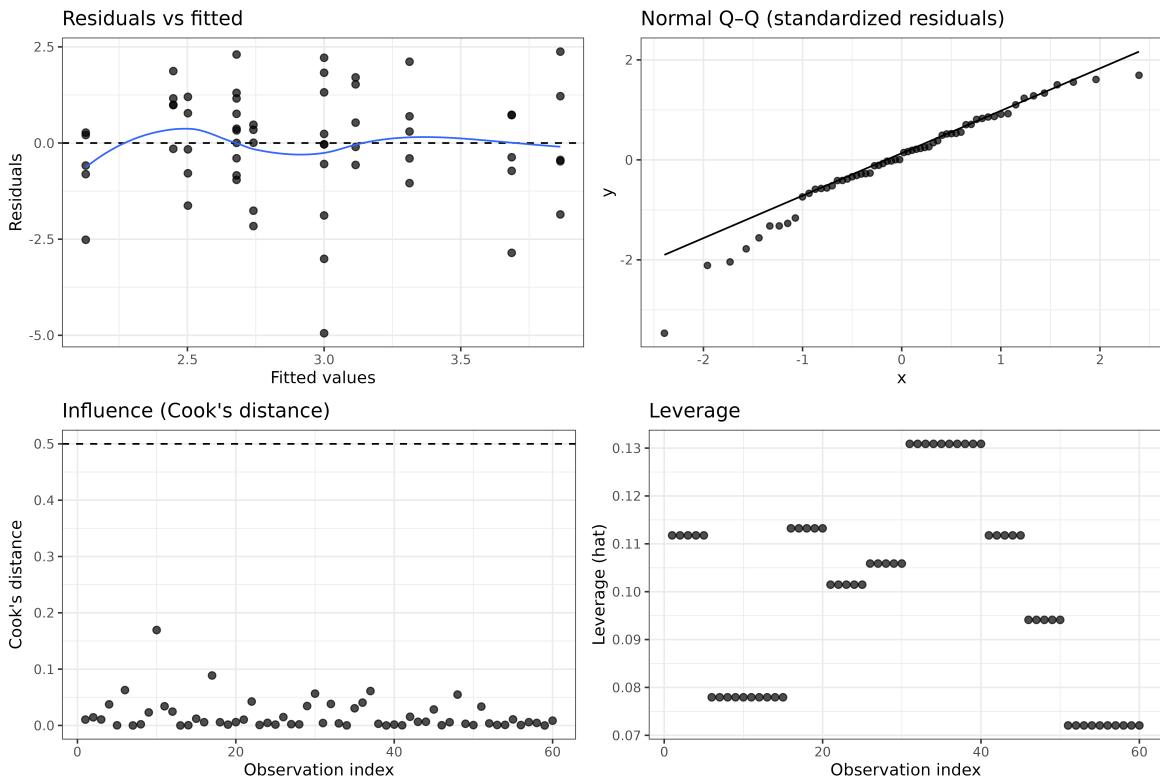
Habitat	n	NRI_aw	NTI_aw	NRI_pa	NTI_pa
forest	25	0.991 (3.02)	3.053 (1.70)	0.711 (2.19)	5.764 (2.13)
paramo	15	1.435 (1.85)	3.277 (1.62)	0.831 (2.81)	4.930 (2.06)
subparamo	20	1.675 (1.77)	2.831 (1.40)	0.395 (1.50)	4.876 (2.49)

**Table S14:** ANOVA summaries for NTI and NRI

Table 19: Table S14. ANOVA summaries for linear models of abundance-weighted NTI and NRI (habitat + site).

Response	Predictor	df	F	p
NTI_aw	habitat	2	0.5106	0.6030
NTI_aw	site	3	1.8164	0.1551
NTI_aw	Residuals	54	NA	NA
NRI_aw	habitat	2	1.2729	0.2883
NRI_aw	site	3	3.4206	0.0235
NRI_aw	Residuals	54	NA	NA

#### NTI\_aw model diagnostics



**Figure S10:** Diagnostic plots for the NTI\_aw linear model.

```
# Key outputs saved:
#   - figures/Fig_S10_NTI_aw_diagnostics.png
#   - tables/Table_S13_NRI_NTI_median_IQR.csv
```

```

#      - tables/Table_S14_NRI_NTI_ANOVA.csv

library(phyloseq)
library(picante)
library(ape)
library(phangorn)
library(dplyr)
library(ggplot2)
library(patchwork)

set.seed(1)

ps <- ps_tree #phyloseq object with tree

stopifnot(exists("ps"))
tr2 <- phy_tree(ps)
stopifnot(!is.null(tr2))

# Root tree if needed
if (!ape::is.rooted(tr2)) tr2 <- phangorn::midpoint(tr2)

## 1) samples x taxa abundance matrix
comm <- as(otu_table(ps), "matrix")
if (taxa_are_rows(ps)) comm <- t(comm)

## 2) align taxa with tree tips
keep_taxa <- intersect(colnames(comm), tr2$tip.label)
comm <- comm[, keep_taxa, drop = FALSE]
tr2 <- ape::keep.tip(tr2, keep_taxa)

## 3) optional: drop ultra-rare taxa
min_total_taxon_reads <- 10
keep_taxa2 <- colSums(comm) >= min_total_taxon_reads
comm <- comm[, keep_taxa2, drop = FALSE]
tr2 <- ape::keep.tip(tr2, colnames(comm))

## 4) metadata aligned to comm
md <- as(sample_data(ps), "data.frame")
md <- md[rownames(comm), , drop = FALSE]
md$habitat <- droplevels(factor(md$habitat))
md$site    <- droplevels(factor(md$site))

```

```

## 5) phylogenetic distances among taxa
dist_phy <- cophenetic(tr2)

## 6) null model tests
mpd_aw <- ses.mpd (comm, dist_phy, null.model = "taxa.labels",
                     runs = 999, abundance.weighted = TRUE)
mndt_aw <- ses.mndt(comm, dist_phy, null.model = "taxa.labels",
                     runs = 999, abundance.weighted = TRUE)

comm_pa <- 1 * (comm > 0)
mpd_pa <- ses.mpd (comm_pa, dist_phy, null.model = "taxa.labels",
                     runs = 999, abundance.weighted = FALSE)
mndt_pa <- ses.mndt(comm_pa, dist_phy, null.model = "taxa.labels",
                     runs = 999, abundance.weighted = FALSE)

## 7) per-sample output table
out <- data.frame(
  sample = rownames(comm),
  NRI_aw = -mpd_aw$mpd.obs.z,
  NTI_aw = -mndt_aw$mndt.obs.z,
  NRI_pa = -mpd_pa$mpd.obs.z,
  NTI_pa = -mndt_pa$mndt.obs.z,
  stringsAsFactors = FALSE
)
rownames(out) <- out$sample

# Drop any sample with NA (e.g., too few taxa after filtering)
out <- out[complete.cases(out[, c("NRI_aw","NTI_aw","NRI_pa","NTI_pa")]), ]
out <- cbind(out, md[rownames(out), c("habitat","site")], drop = FALSE)
out$habitat <- droplevels(factor(out$habitat))
out$site <- droplevels(factor(out$site))

## 8) Table S12: median (IQR) per habitat, formatted as "median (IQR)"
fmt_med_iqr <- function(x) sprintf("%.3f (%.2f)", median(x, na.rm=TRUE), IQR(x, na.rm=TRUE))

tab_s12 <- out %>%
  group_by(habitat) %>%
  summarise(
    n = n(),
    NRI_aw = fmt_med_iqr(NRI_aw),
    NTI_aw = fmt_med_iqr(NTI_aw),
    NRI_pa = fmt_med_iqr(NRI_pa),

```

```

NTI_pa = fmt_med_iqr(NTI_pa),
  .groups = "drop"
) %>%
  rename(Habitat = habitat)

write.csv(tab_s12, file.path(out_tab_dir, "Table_S13_NRI_NTI_median_IQR.csv"), row.names = FALSE)

## 9) Table S13: ANOVA summaries (NTI_aw and NRI_aw models)
m_NTI_aw <- lm(NTI_aw ~ habitat + site, data = out)
m_NRI_aw <- lm(NRI_aw ~ habitat + site, data = out)

a_NTI <- as.data.frame(anova(m_NTI_aw))
a_NRI <- as.data.frame(anova(m_NRI_aw))

tab_s13 <- bind_rows(
  data.frame(Response = "NTI_aw", Predictor = rownames(a_NTI), df = a_NTI$Df, F = a_NTI$`F value`, p = a_NTI$`Pr(> F)`),
  data.frame(Response = "NRI_aw", Predictor = rownames(a_NRI), df = a_NRI$Df, F = a_NRI$`F value`, p = a_NRI$`Pr(> F)`),
) %>%
  filter(Predictor %in% c("habitat", "site", "Residuals"))

write.csv(tab_s13, file.path(out_tab_dir, "Table_S14_NRI_NTI_ANOVA.csv"), row.names = FALSE)

## 10) Figure S11: diagnostics for NTI_aw model
diagdf <- data.frame(
  fitted = fitted(m_NTI_aw),
  resid = resid(m_NTI_aw),
  stdres = rstandard(m_NTI_aw),
  cooks = cooks.distance(m_NTI_aw),
  hat = hatvalues(m_NTI_aw)
)

p1 <- ggplot(diagdf, aes(fitted, resid)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point(alpha = 0.7, size = 1.8) +
  geom_smooth(method = "loess", se = FALSE, linewidth = 0.5) +
  labs(x = "Fitted values", y = "Residuals", title = "Residuals vs fitted") +
  theme_bw(base_size = 10)

p2 <- ggplot(diagdf, aes(sample = stdres)) +
  stat_qq(alpha = 0.7, size = 1.5) +
  stat_qq_line() +
  labs(title = "Normal Q-Q (standardized residuals)") +

```

```

theme_bw(base_size = 10)

p3 <- ggplot(diagdf, aes(seq_along(cooks), cooks)) +
  geom_point(alpha = 0.7, size = 1.8) +
  geom_hline(yintercept = 0.5, linetype = "dashed") +
  labs(x = "Observation index", y = "Cook's distance", title = "Influence (Cook's distance)") +
  theme_bw(base_size = 10)

p4 <- ggplot(diagdf, aes(seq_along(hat), hat)) +
  geom_point(alpha = 0.7, size = 1.8) +
  labs(x = "Observation index", y = "Leverage (hat)", title = "Leverage") +
  theme_bw(base_size = 10)

panel <- (p1 | p2) / (p3 | p4) + plot_annotation(title = "NTI_aw model diagnostics")

ggsave(
  filename = file.path(out_fig_dir, "Fig_S10_NTI_aw_diagnostics.png"),
  plot = panel,
  width = 10, height = 7, dpi = 600, bg = "white"
)

```

## Soil Physicochemical parameters

To provide environmental context for the elevational gradients sampled in this study, we quantified bulk soil physicochemical properties from one composite topsoil sample (0–15 cm) collected at each sampling location ( $n = 12$ ; one per location). Because soil chemistry was measured once per location, these data are interpreted descriptively and were not used for formal statistical inference among habitats or sites.

The full soil dataset, including all measured variables, is provided as Table S15 and is archived in the repository as “Data\_S3\_soil.csv”.

**Figure S11:** Correlation heatmap of bulk soil physicochemical variables measured at each sampling location ( $n = 12$ ). Colors indicate Pearson correlation among soil variables and elevation.