

Environmental filtering and habitat (mis)matching of riverine invertebrate communities

NRSA DisEQ-Analyses

Contents

Questions	3
Q1: How do filtering and habitat matching vary across ecoregions?	4
Filtering & Habitat Matching ANOVAs	4
Filtering ANOVA	4
Habitat Matching ANOVA	6
Vectors of Habitat Matching & Mismatch	8
T_{\max} Vector Component ANOVA	9
T_{\min} Vector Component ANOVA	11
pH Vector Component ANOVA	13
Conductivity Vector Component ANOVA	15
Response to Q1: How do filtering and habitat matching vary across ecoregions?	17
Q2: How does functional diversity vary across ecoregions?	19
Trait Diversity	19
Trait Diversity ANOVAs	21
Functional Richness (FRic)	22
Functional Evenness (FEve)	24
Functional Divergence (FDiv)	26
Functional Dispersion (FDis)	28
Response to Q2: How does functional diversity vary across ecoregions?	30
Trait Abundance Calculation	31
Q3: Which functional traits are linked to filtering and habitat matching?	35
DisEQ-by-Traits Relationships	35
Filtering-by-Traits BRTs	37
Habitat Matching-by-Traits BRTs	38
DisEQ-by-Trait Diversity ANCOVAs	39
DisEQ-by-FRic	40
DisEQ-by-FEve	44
DisEQ-by-FDiv	48
DisEQ-by-FDis	52
Response to Q3: Which functional traits are linked to filtering and habitat matching?	56
Q4: What are the environmental predictors of functional trait abundances?	61
Trait-by-Environment Relationships	61
Trait-by-Environment Data Management	61
Low Dispersal-by-Environment BRTs	66
High Dispersal-by-Environment BRTs	67
Nonflyer-by-Environment BRTs	68

Weak Flyer-by-Environment BRTs	69
Strong Flyer-by-Environment BRTs	70
Small Size-by-Environment BRTs	71
Medium Size-by-Environment BRTs	72
Large Size-by-Environment BRTs	73
Depositional-by-Environment BRTs	74
Depositional-Erosional-by-Environment BRTs	75
Erosional-by-Environment BRTs	76
Cold Water-by-Environment BRTs	77
Cool-Warm Water-by-Environment BRTs	78
Warm Water-by-Environment BRTs	79
Sensitive Tolerance-by-Environment BRTs	80
Intermediate Tolerance-by-Environment BRTs	81
Tolerant-by-Environment BRTs	82
Collector Gatherer-by-Environment BRTs	83
Collector Filterer-by-Environment BRTs	84
Herbivore-by-Environment BRTs	85
Predator-by-Environment BRTs	86
Response to Q4: What are the environmental predictors of functional trait abundances?	87
Supplementary Results	89
Environmental Filtering & Habitat Matching Model	89
Filtering & Habitat Matching Summaries	92
Trait Diversity Summaries	93
Community Composition	95
R Session Information	96

Questions

1. How do filtering and habitat matching vary across ecoregions?
2. How does functional diversity vary across ecoregions?
3. Which functional traits are linked to filtering and habitat matching?
4. What are the environmental predictors of functional trait abundances?

Q1: How do filtering and habitat matching vary across ecoregions?

Filtering & Habitat Matching ANOVAs

Filtering and habitat matching were compared by ecoregion to determine if filtering and mismatch vary spatially (**Question 1**). Filtering and mismatch were compared by ecoregion using a one-way ANOVA with Type III sums-of-squares [`Anova()` in the `car` package]; post-hoc Tukey's HSD tests [`HSD.test()` in the `agricolae` package] were used to identify differences among groups. ANOVA assumptions were inspected graphically using `check_model()` in the `performance` package. Effect sizes for the ANOVAs were calculated as η^2 using `eta_squared()` in the `effectsize` package.

Filtering ANOVA

```
filtering.aov <- lm(filtering.scaled ~ ecoregion, data = DisEQ.analysis.results)
```

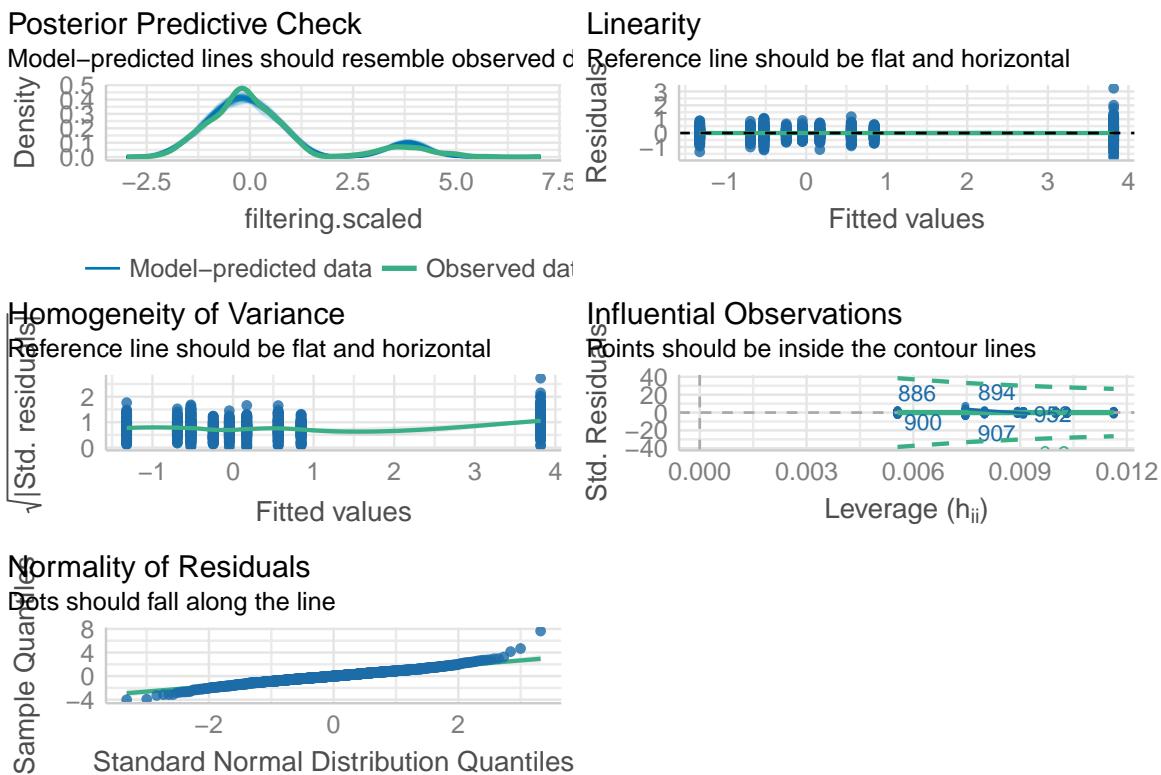


Figure 1: Diagnostic plots for the filtering ANOVA.

Table 1: Summary of the environmental filtering ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	6.060	1	32.011	0
ecoregion	2298.216	8	1517.496	0
Residuals	202.372	1069	NA	NA

Table 2: Effect size for ecoregion in the environmental filtering ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.919		0.95	0.913

Table 3: Tukey groups assigned to ecoregions differing in environmental filtering.

	Mean Filtering	Grouping
WMT	3.817	a
NAP	0.845	b
SPL	0.561	c
TPL	0.170	d
XER	-0.046	e
CPL	-0.246	f
SAP	-0.523	g
UMW	-0.692	g
NPL	-1.320	h

Habitat Matching ANOVA

```
mismatch.aov <- lm(mismatch.scaled ~ ecoregion, data = DisEQ.analysis.results)
```

```
check_model(mismatch.aov)
```

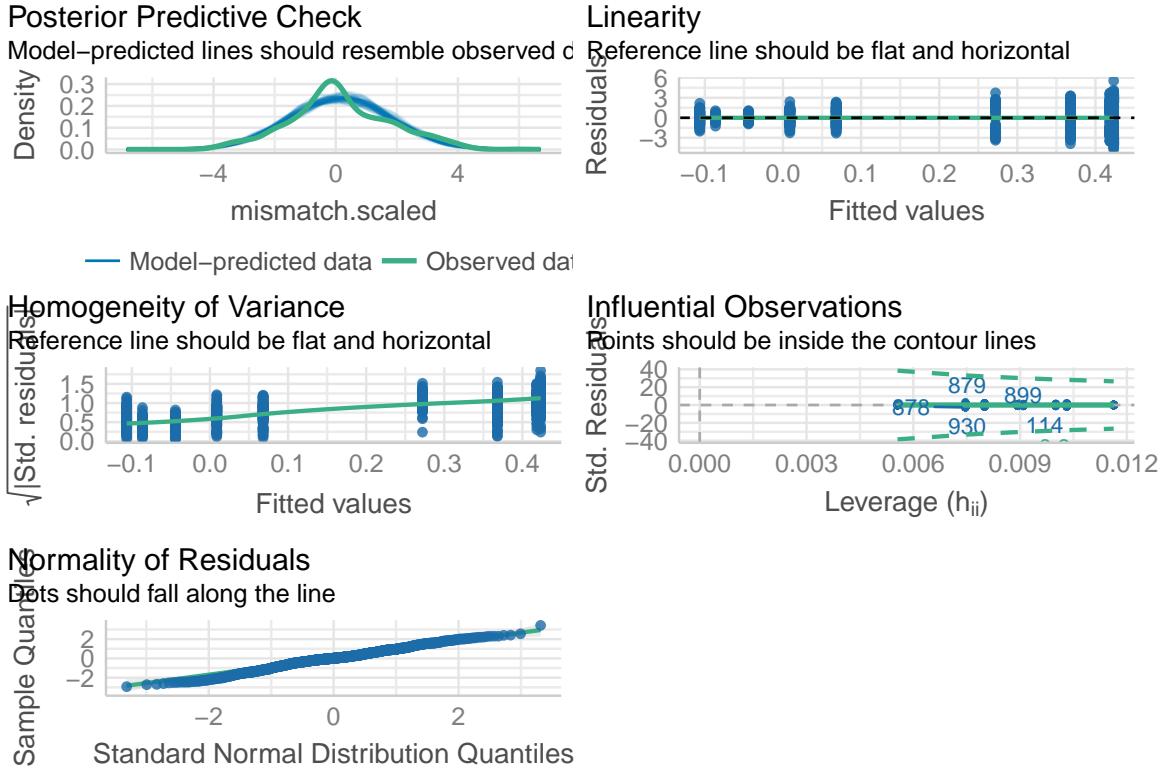


Figure 2: Diagnostic plots for the mismatch ANOVA.

Table 4: Summary of the habitat matching ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	0.463	1	0.173	0.677
ecoregion	48.649	8	2.277	0.020
Residuals	2854.846	1069	NA	NA

Table 5: Effect size for ecoregion in the habitat matching ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.017	0.95	0.001	1

Table 6: Tukey groups assigned to ecoregions differing in habitat matching.

	Mean Habitat Matching	Grouping
WMT	0.423	a
SPL	0.418	a
NPL	0.368	a
NAP	0.272	a
CPL	0.068	a
TPL	0.009	a
SAP	-0.044	a
UMW	-0.086	a
XER	-0.107	a

Vectors of Habitat Matching & Mismatch

Vector components of habitat matching and mismatch were compared by ecoregion to determine if components vary spatially (Addendum to **Question 1**). Vector components were compared by ecoregion using a one-way ANOVA with Type III sums-of-squares [`Anova()` in the `car` package]; post-hoc Tukey's HSD tests [`HSD.test()` in the `agricolae` package] were used to identify differences among groups. ANOVA assumptions were inspected graphically using `check_model()` in the `performance` package. Effect sizes for the ANOVAs were calculated as η^2 using `eta_squared()` in the `effectsize` package.

T_{max} Vector Component ANOVA

```
Tmax.DisEQ.vector.aov <- lm(Tmax.direction ~ ecoregion, data = final.DisEQ.data)
```

```
check_model(Tmax.DisEQ.vector.aov)
```

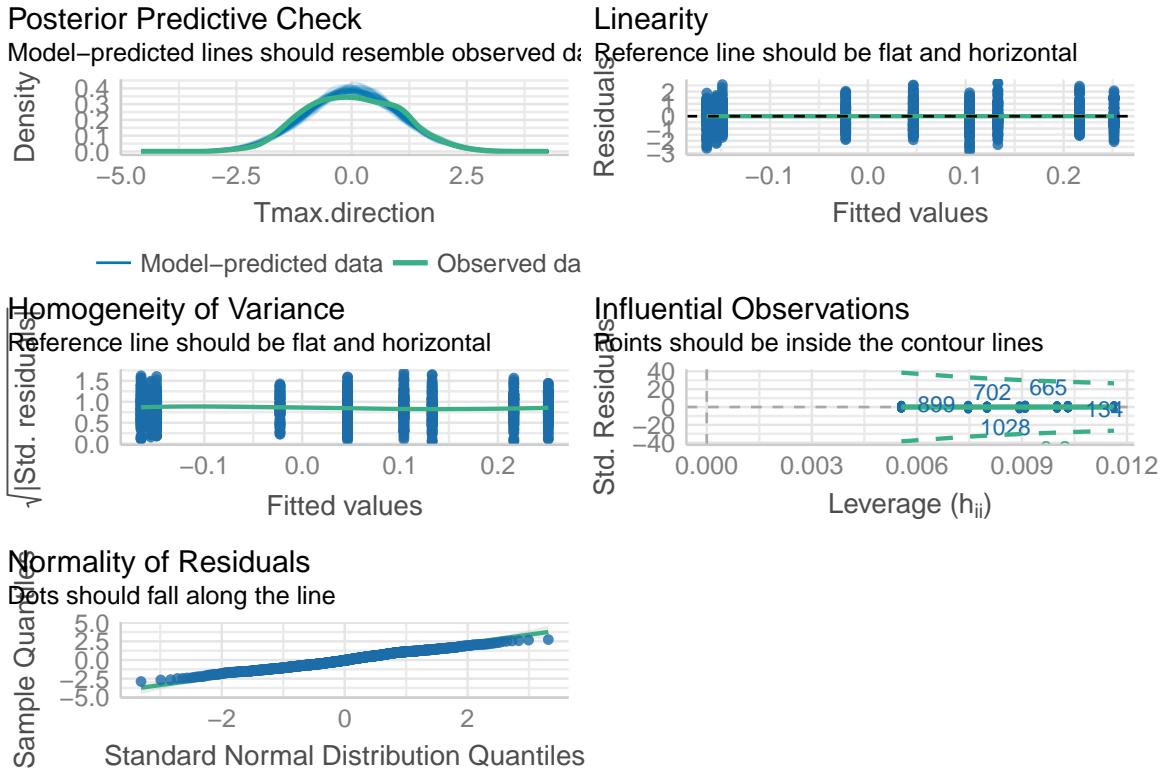


Figure 3: Diagnostic plots for the Tmax DisEQ Vector ANOVA.

Table 7: Summary of the Tmax DisEQ Vector ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	6.328	1	6.328	0.012
ecoregion	23.178	8	2.897	0.003
Residuals	1069.008	1069	NA	NA

Table 8: Effect size for ecoregion in the Tmax DisEQ Vector ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.0212221	0.95	0.004046	1

Table 9: Tukey groups assigned to ecoregions differing in Tmax DisEQ vector components.

	Mean Tmax	Grouping
CPL	0.252	a
NAP	0.216	a
TPL	0.133	a
WMT	0.104	a
SAP	0.046	a
UMW	-0.023	a
NPL	-0.149	a
SPL	-0.155	a
XER	-0.165	a

T_{min} Vector Component ANOVA

```
Tmin.DisEQ.vector.aov <- lm(Tmin.direction ~ ecoregion, data = final.DisEQ.data)

check_model(Tmin.DisEQ.vector.aov)
```

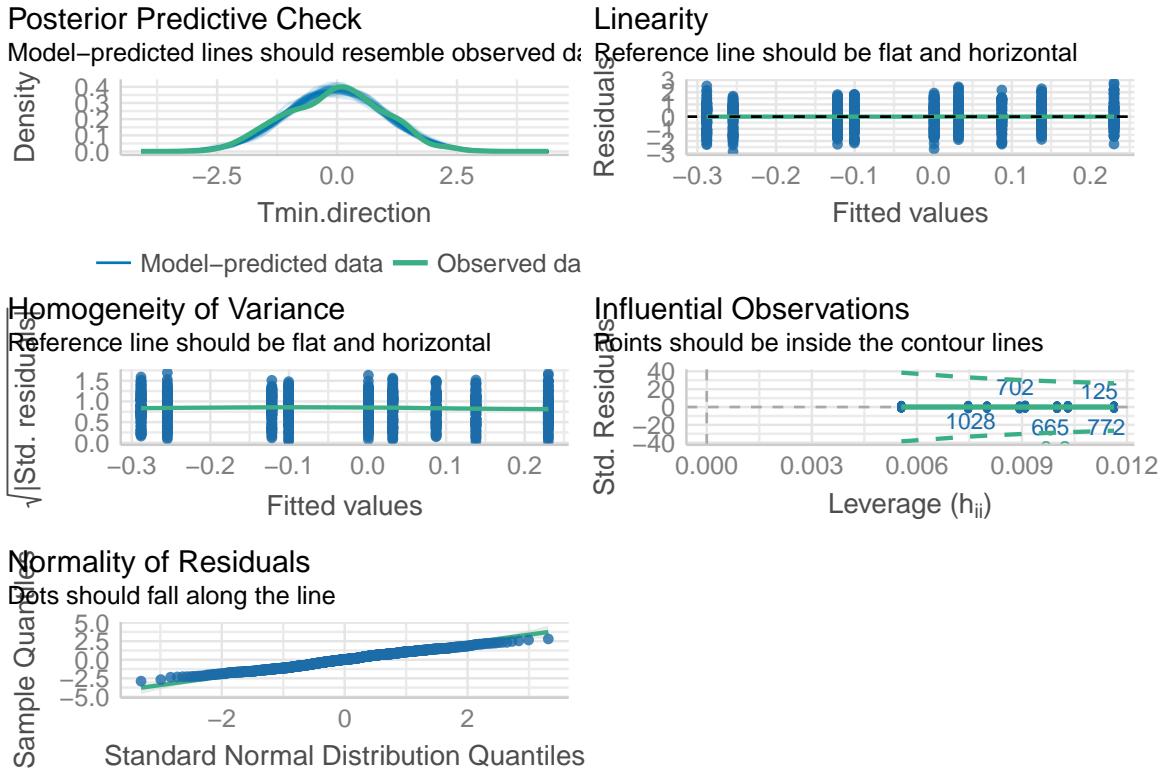


Figure 4: Diagnostic plots for the Tmin DisEQ Vector ANOVA.

Table 10: Summary of the Tmin DisEQ Vector ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	0.762	1	0.763	0.383
ecoregion	28.416	8	3.555	0.000
Residuals	1067.992	1069	NA	NA

Table 11: Effect size for ecoregion in the Tmin DisEQ Vector ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.026	0.95	0.007	1

Table 12: Tukey groups assigned to ecoregions differing in Tmin DisEQ vector components.

	Mean	Tmin	Grouping
TPL	0.230	a	
NAP	0.138	ab	
CPL	0.087	abc	
SAP	0.032	abc	
WMT	0.001	abc	
UMW	-0.100	abc	
SPL	-0.122	abc	
XER	-0.255	bc	
NPL	-0.288	c	

pH Vector Component ANOVA

```
pH.DisEQ.vector.aov <- lm(pH.direction ~ ecoregion, data = final.DisEQ.data)
```

```
check_model(pH.DisEQ.vector.aov)
```

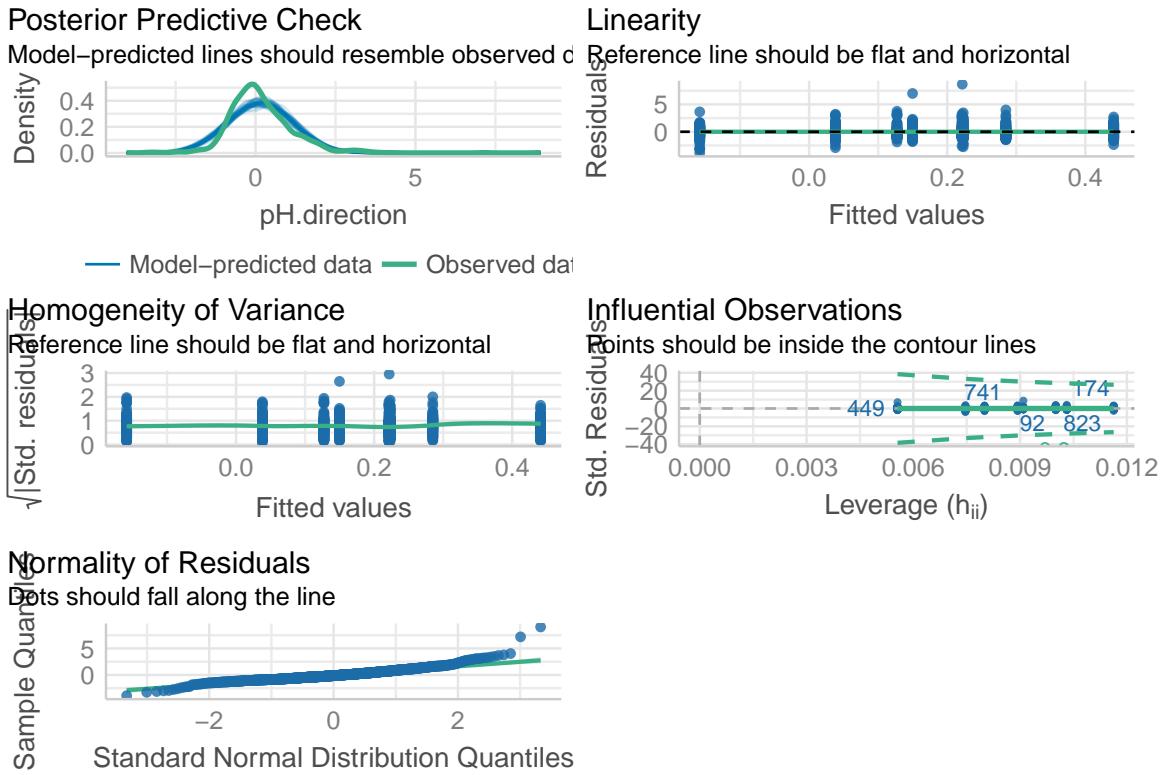


Figure 5: Diagnostic plots for the pH DisEQ Vector ANOVA.

Table 13: Summary of the pH DisEQ Vector ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	1.619	1	1.618	0.204
ecoregion	28.830	8	3.602	0.000
Residuals	1069.401	1069	NA	NA

Table 14: Effect size for ecoregion in the pH DisEQ Vector ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.026	0.95	0.007	1

Table 15: Tukey groups assigned to ecoregions differing in pH DisEQ vector components.

	Mean pH	Grouping
WMT	0.441	a
NAP	0.285	ab
XER	0.223	abc
TPL	0.222	abc
UMW	0.221	abc
SAP	0.150	abc
CPL	0.127	abc
SPL	0.038	bc
NPL	-0.158	c

Conductivity Vector Component ANOVA

```
cond.DisEQ.vector.aov <- lm(cond.direction ~ ecoregion, data = final.DisEQ.data)
```

```
check_model(cond.DisEQ.vector.aov)
```

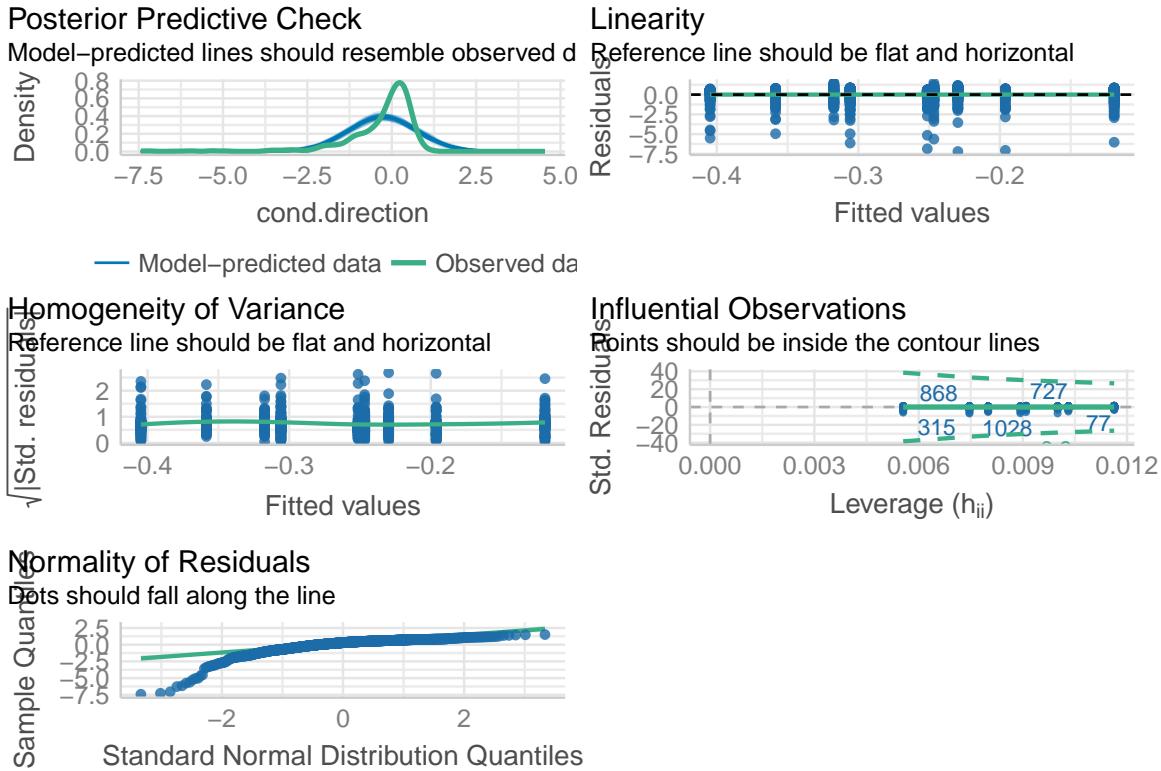


Figure 6: Diagnostic plots for the Conductivity DisEQ Vector ANOVA.

Table 16: Summary of the Conductivity DisEQ Vector ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	3.847	1	3.85	0.050
ecoregion	7.356	8	0.92	0.499
Residuals	1068.362	1069	NA	NA

Table 17: Effect size for ecoregion in the Conductivity DisEQ Vector ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.007		0.95	1

Table 18: Tukey groups assigned to ecoregions differing in Conductivity DisEQ vector components.

	Mean Conductivity	Grouping
WMT	-0.119	a
CPL	-0.196	a
NPL	-0.230	a
TPL	-0.247	a
XER	-0.251	a
SAP	-0.306	a
UMW	-0.317	a
NAP	-0.358	a
SPL	-0.405	a

Response to Q1: How do filtering and habitat matching vary across ecoregions?

Environmental filtering varied strongly by ecoregion ($F_{8,1069} = 1517.496, P < 0.001, \eta^2 = 0.919$). Environmental filtering (lower δ) was strongest in the NPL ecoregion, followed by the SAP and UMW ecoregions, while environmental permissiveness (higher δ) was highest in the WMT ecoregion followed by the NAP. Habitat matching also varied by ecoregion, but this effect was weak ($F_{8,1069} = 2.277, P = 0.020, \eta^2 = 0.017$); post-hoc Tukey's HSD did not identify any pairwise differences between ecoregions. Qualitatively, habitat matching (higher λ) was stronger in the WMT, SPL, and NPL ecoregions while niche matching (lower λ) was stronger in the UMW and XER ecoregions.

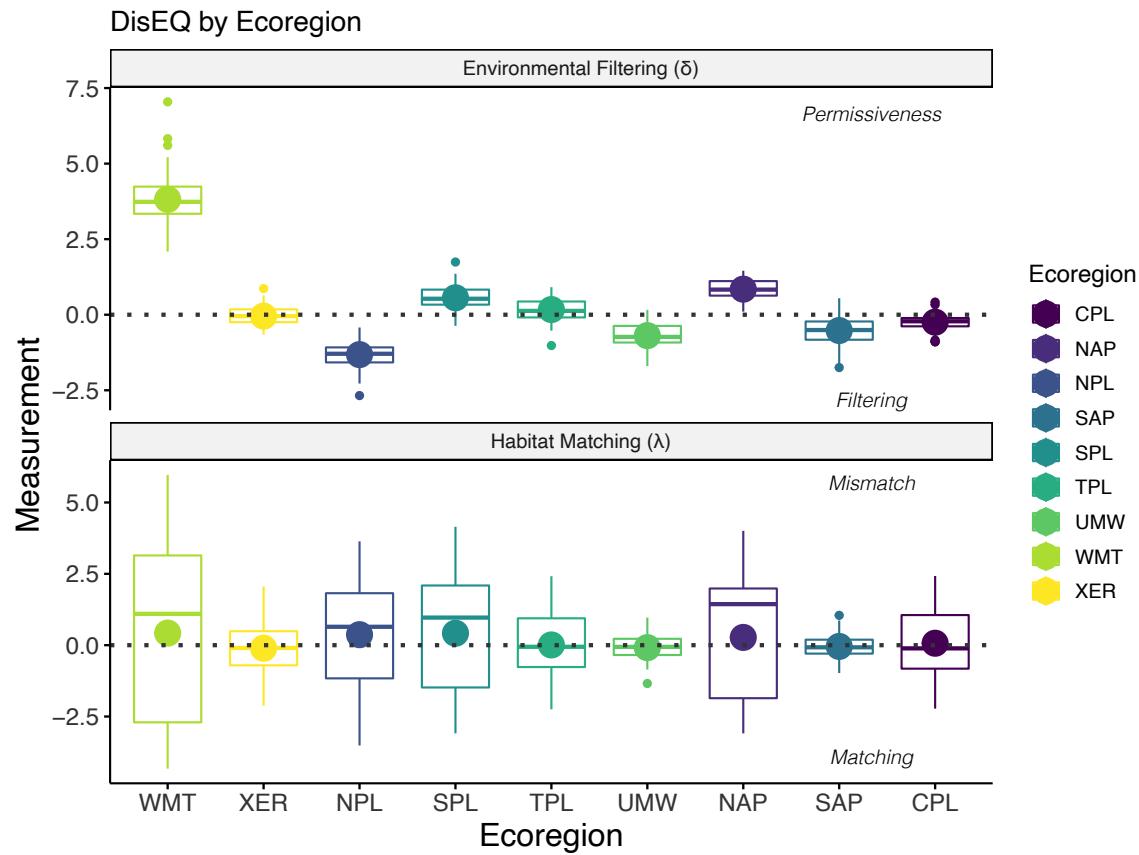


Figure 7: Plots of environmental filtering and habitat matching by ecoregion.

Vector components varied by ecoregion in response to habitat matching and mismatch. T_{\max} vector components varied by ecoregion ($F_{8,1069} = 2.897, P = 0.003, \eta^2 = 0.021$), but post-hoc tests did not identify any pairwise differences. Generally, T_{\max} pushed the CPL and NAP ecoregions into mismatch and the XER, SPL, and NPL ecoregions into habitat matching. T_{\min} vector components also varied by ecoregion ($F_{8,1069} = 3.555, P < 0.001, \eta^2 = 0.026$), driving mismatch in the TPL and habitat matching in the NPL. pH vector components also varied by ecoregion ($F_{8,1069} = 3.602, P < 0.001, \eta^2 = 0.026$), driving mismatch in the WMT and habitat matching in the NPL. There was no evidence that conductivity vector components varied by ecoregion ($F_{8,1069} = 0.920, P = 0.499, \eta^2 = 0.007$)

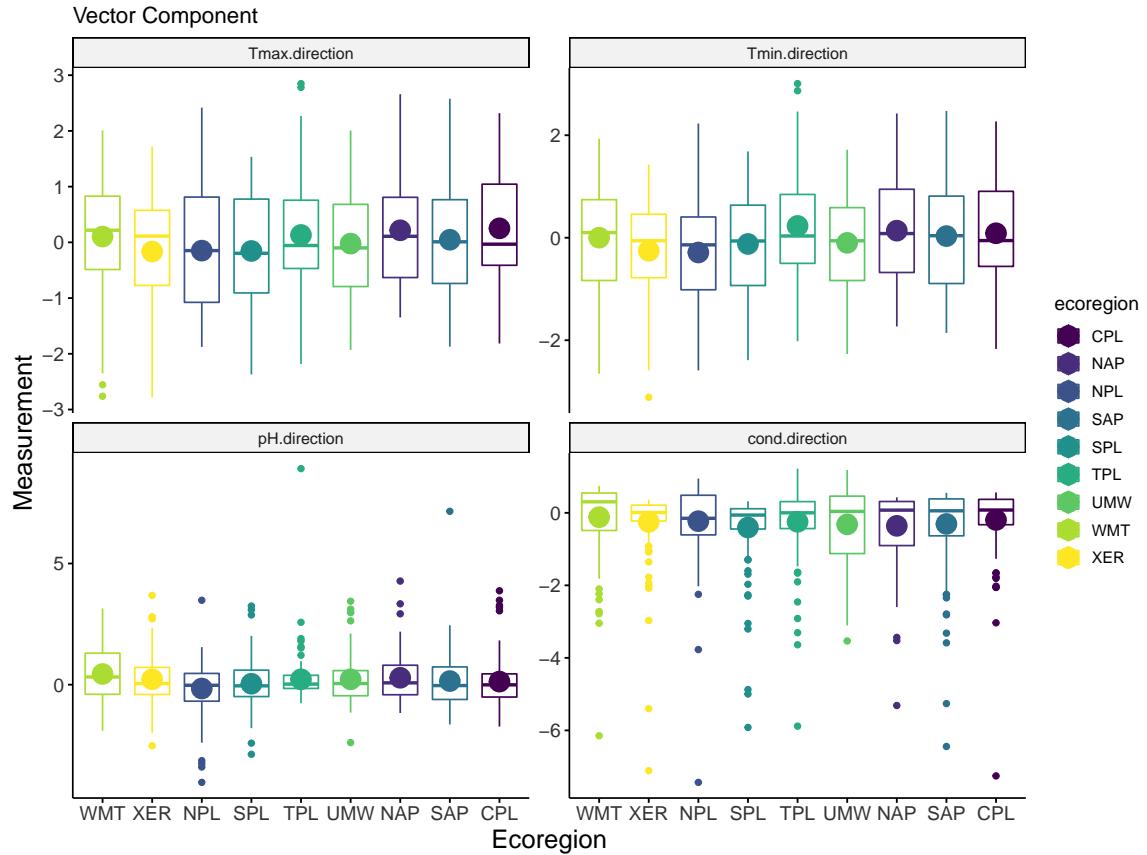


Figure 8: Plots of habitat matching and mismatch vector components by ecoregion.

Q2: How does functional diversity vary across ecoregions?

Trait Diversity

Trait diversity was quantified as functional richness (FRic), functional evenness (FEve), functional divergence (FDiv), and functional dispersion (FDis) for each community across all 9 ecoregions.

Definitions:

- FRic = portion of trait space occupied by the community
- FEve = measures the regularity of the distribution of traits within occupied trait space using a minimum spanning tree
- FDiv = proportion of trait space occupied by extreme trait values
- FDis = weighted mean distance of individual taxa in the community to the centroid of all species in multidimensional trait space, and simultaneously measures trait dissimilarity and evenness within the community.

```
community.matrix <- final.data[rowSums(final.data[, 67:142]) > 0, 67:142]

## Set a trait table for re-coded values
trait.table.recoded <- trait.table.categorical

## Set character mappings
dispersal.mapping <- c("low" = 0, "high" = 1)
flying.mapping <- c("none" = 0, "weak" = 1, "strong" = 2)
size.mapping <- c("small" = 0, "medium" = 1, "large" = 2)
rheophily.mapping <- c("depositional" = 0, "depositional_erosional" = 1, "erosional" = 2)
thermal.mapping <- c("cold" = 0, "cool_warm" = 1, "warm" = 2)
FFG.mapping <- c("CG" = 0, "CF" = 1, "HB" = 2, "PR" = 3)
tolerance.mapping <- c("sensitive" = 0, "medium" = 1, "tolerant" = 2)

## Recode categorical traits to quasi-ordinal
trait.table.recoded$dispersal <- dispersal.mapping[trait.table.recoded$dispersal]
trait.table.recoded$flying.strength <- flying.mapping[trait.table.recoded$flying.strength]
trait.table.recoded$size <- size.mapping[trait.table.recoded$size]
trait.table.recoded$rheophily <- rheophily.mapping[trait.table.recoded$rheophily]
trait.table.recoded$thermal.preference <- thermal.mapping[trait.table.recoded$thermal.preference]
trait.table.recoded$FFG <- FFG.mapping[trait.table.recoded$FFG]
trait.table.recoded$tolerance <- tolerance.mapping[trait.table.recoded$tolerance]

## Calculate functional trait diversity metrics
trait.diversity <- data.frame(dbFD(
  x = trait.table.recoded,
  a = community.matrix,
  m = "min",
  stand.FRIC = TRUE,
  calc.CWM = FALSE,
  corr = "none",
  messages = FALSE
))
)

## Add site UID to merge the trait diversity with the final data
trait.diversity$UID <- final.data[rowSums(final.data[, 67:142]) > 0, 1]

## Merge into final dataframe
```

```
final.DisEQ.data <- DisEQ.analysis.results %>%
  full_join(trait.diversity, by = "UID")
```

Trait Diversity ANOVAs

Trait diversity was compared by ecoregion to determine if trait diversity varied among ecoregions (**Question 2**). Each measure of trait diversity (functional richness = FRic, functional evenness = FEve, functional divergence = FDiv, and functional dispersion = FDis) was compared using the same ANOVA workflow for the filtering and habitat matching ANOVAs described above. Briefly, a one-way ANOVA with Type III sums-of-squares was performed on each trait diversity metric, with post-hoc Tukey's HSD tests to identify differences among groups. Assumptions were checked using `check_model` in the `performance` package, with effect sizes calculated as η^2 .

We calculated each trait diversity measure using the `dbFD()` function in the `FD` package.

References for functional traits and trait diversity metrics:

McGill, B., et al. 2006. Rebuilding community ecology from functional traits. *Trends in Ecology and Evolution* 21: 178-185.

Villéger, S., et al. 2008. New multidimensional functional diversity indices for a multifaceted framework in functional ecology. *Ecology* 89: 2290-2301.

Laliberté, E., and P. Legendre. 2010. A distance-based framework for measuring functional diversity from multiple traits. *Ecology* 91: 299-305.

Functional Richness (FRic)

```
FRic.aov <- lm(FRic ~ ecoregion, data = final.DisEQ.data)
```

```
check_model(FRic.aov)
```

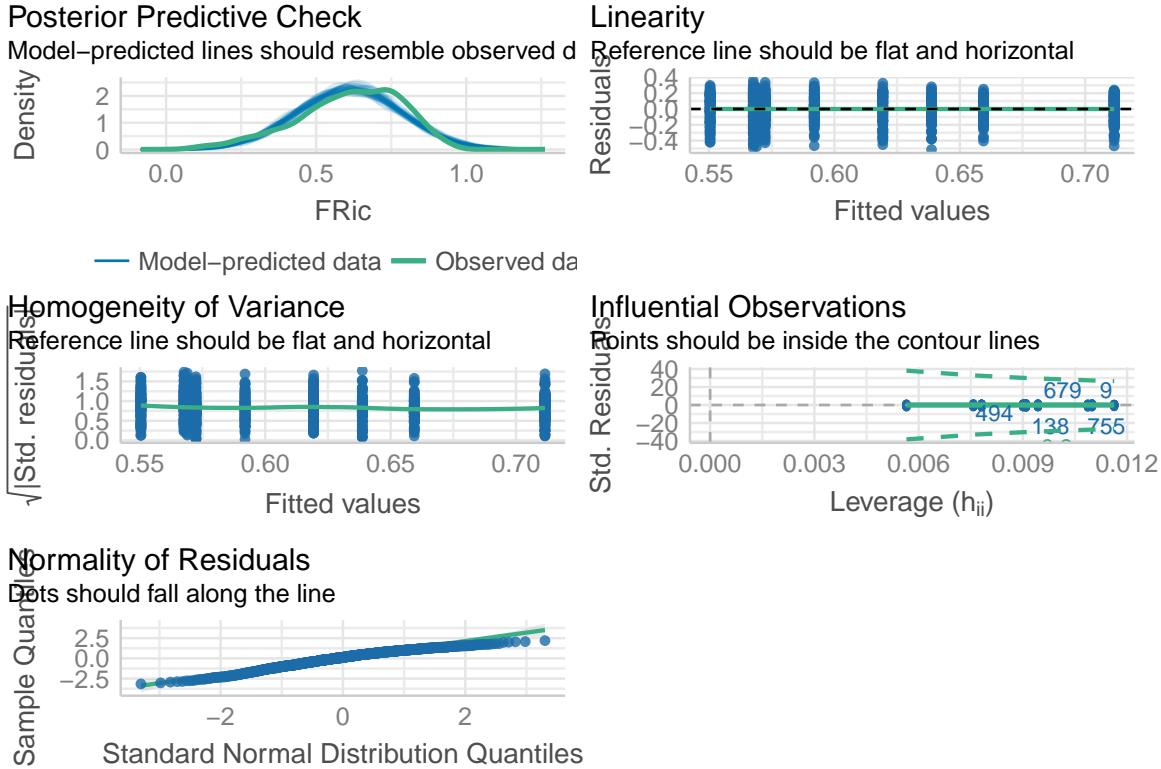


Figure 9: Diagnostic plots for the FRic ANOVA.

Table 19: Summary of the FRic ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	31.888	1	1188.788	0
ecoregion	3.138	8	14.621	0
Residuals	27.467	1024	NA	NA

Table 20: Effect size for ecoregion in the FRic ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.103	0.95	0.07	1

Table 21: Tukey groups assigned to ecoregions differing in FRic.

	Mean FRic	Grouping
SAP	0.712	a
UMW	0.659	ab
NAP	0.639	bc
TPL	0.619	bcd
CPL	0.592	bcd
WMT	0.572	cd
NPL	0.569	d
SPL	0.568	d
XER	0.550	d

Functional Evenness (FEve)

```
FEve.aov <- lm(FEve ~ ecoregion, data = final.DisEQ.data)
```

```
check_model(FEve.aov)
```

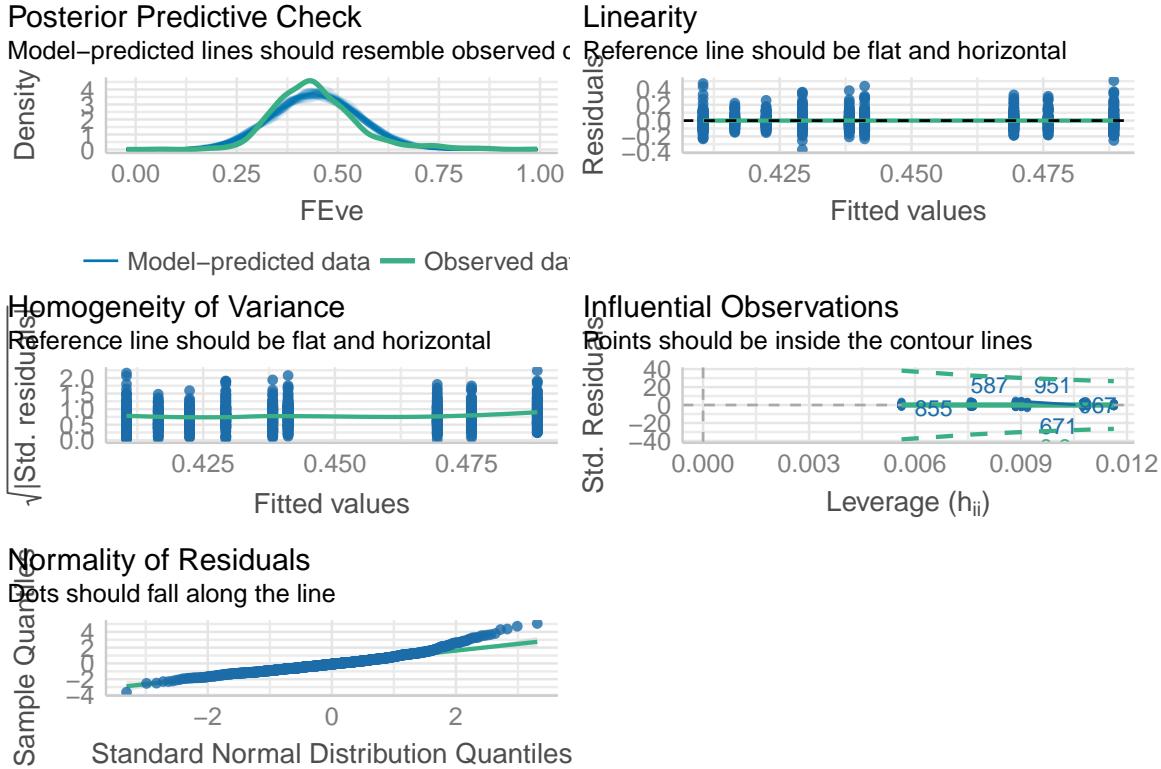


Figure 10: Diagnostic plots for the FEve ANOVA.

Table 22: Summary of the FEve ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	21.065	1	2047.910	0
ecoregion	0.666	8	8.089	0
Residuals	10.656	1036	NA	NA

Table 23: Effect size for ecoregion in the FEve ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.059	0.95	0.032	1

Table 24: Tukey groups assigned to ecoregions differing in FEve.

	Mean FEve	Grouping
SPL	0.488	a
CPL	0.476	ab
NPL	0.469	abc
WMT	0.441	bcd
SAP	0.438	bcd
TPL	0.429	cd
NAP	0.422	d
UMW	0.416	d
XER	0.410	d

Functional Divergence (FDiv)

```
FDiv.aov <- lm(FDiv ~ ecoregion, data = final.DisEQ.data)
```

```
check_model(FDiv.aov)
```

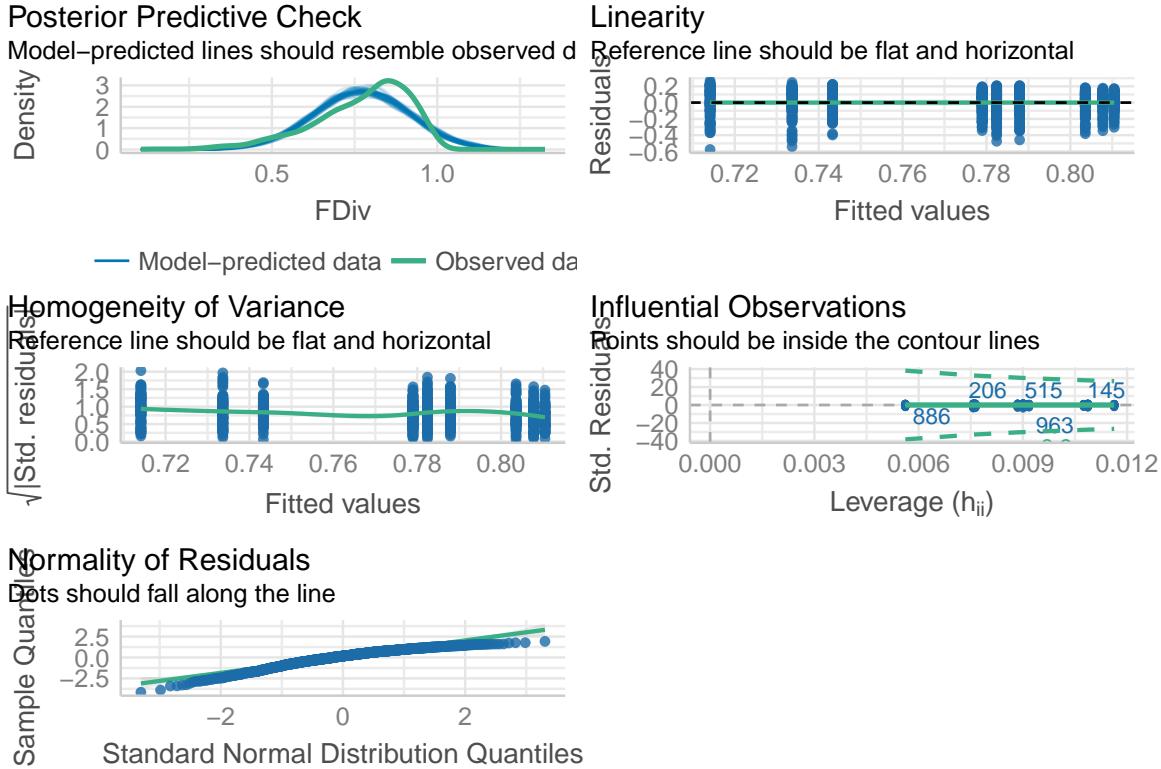


Figure 11: Diagnostic plots for the FDiv ANOVA.

Table 25: Summary of the FDiv ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	61.086	1	3105.245	0
ecoregion	1.049	8	6.667	0
Residuals	20.380	1036	NA	NA

Table 26: Effect size for ecoregion in the FDiv ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.049	0.95	0.024	1

Table 27: Tukey groups assigned to ecoregions differing in FDiv.

	Mean FDiv	Grouping
CPL	0.810	a
UMW	0.808	ab
TPL	0.804	ab
SPL	0.788	abc
NPL	0.782	abc
SAP	0.779	abc
NAP	0.743	bcd
WMT	0.734	cd
XER	0.714	d

Functional Dispersion (FDis)

```
FDis.aov <- lm(FDis ~ ecoregion, data = final.DisEQ.data)
```

```
check_model(FDis.aov)
```

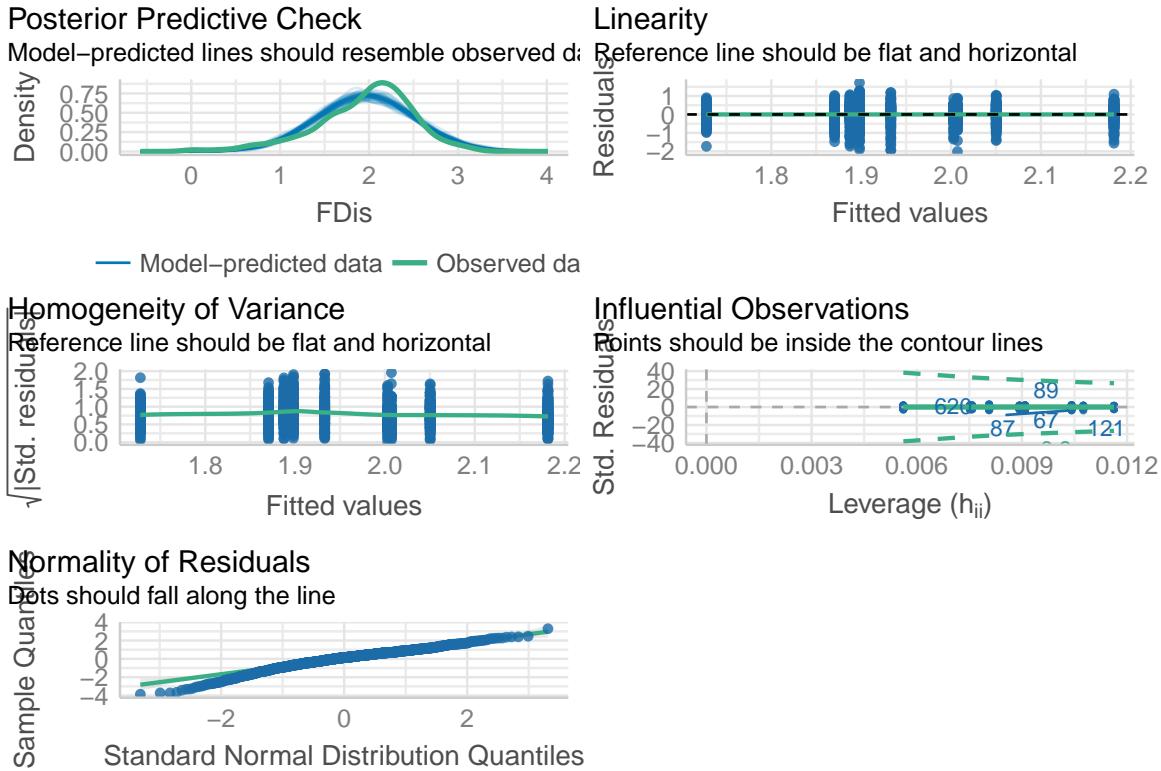


Figure 12: Diagnostic plots for the FDis ANOVA.

Table 28: Summary of the FDis ANOVA.

Term	Sums-of-Squares	df	F	P-value
(Intercept)	358.690	1	1293.416	0
ecoregion	18.384	8	8.286	0
Residuals	292.573	1055	NA	NA

Table 29: Effect size for ecoregion in the FDis ANOVA.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
ecoregion	0.059	0.95	0.033	1

Table 30: Tukey groups assigned to ecoregions differing in FDis.

	Mean FDis	Grouping
SAP	2.181	a
NPL	2.050	ab
NAP	2.007	ab
UMW	2.003	ab
CPL	1.933	bc
SPL	1.898	bc
TPL	1.888	bc
WMT	1.870	bc
XER	1.728	c

Response to Q2: How does functional diversity vary across ecoregions?

All four measures of functional trait diversity varied by ecoregion. Functional richness was highest in the SAP and lowest in the NPL, SPL, and XER ecoregions ($F_{8,1069} = 14.621, P < 0.001, \eta^2 = 0.103$), while FEve was highest in the SPL and lowest in the NAP, UMW, and XER ecoregions ($F_{8,1069} = 8.089, P < 0.001, \eta^2 = 0.059$). Functional divergence was highest in the CPL and lowest in the XER ecoregions ($F_{8,1069}=6.667, P < 0.001, \eta^2 = 0.049$), and FDis was highest in the SAP and lowest in the XER ecoregions ($F_{8,1069}=8.286, P < 0.001, \eta^2 = 0.059$).

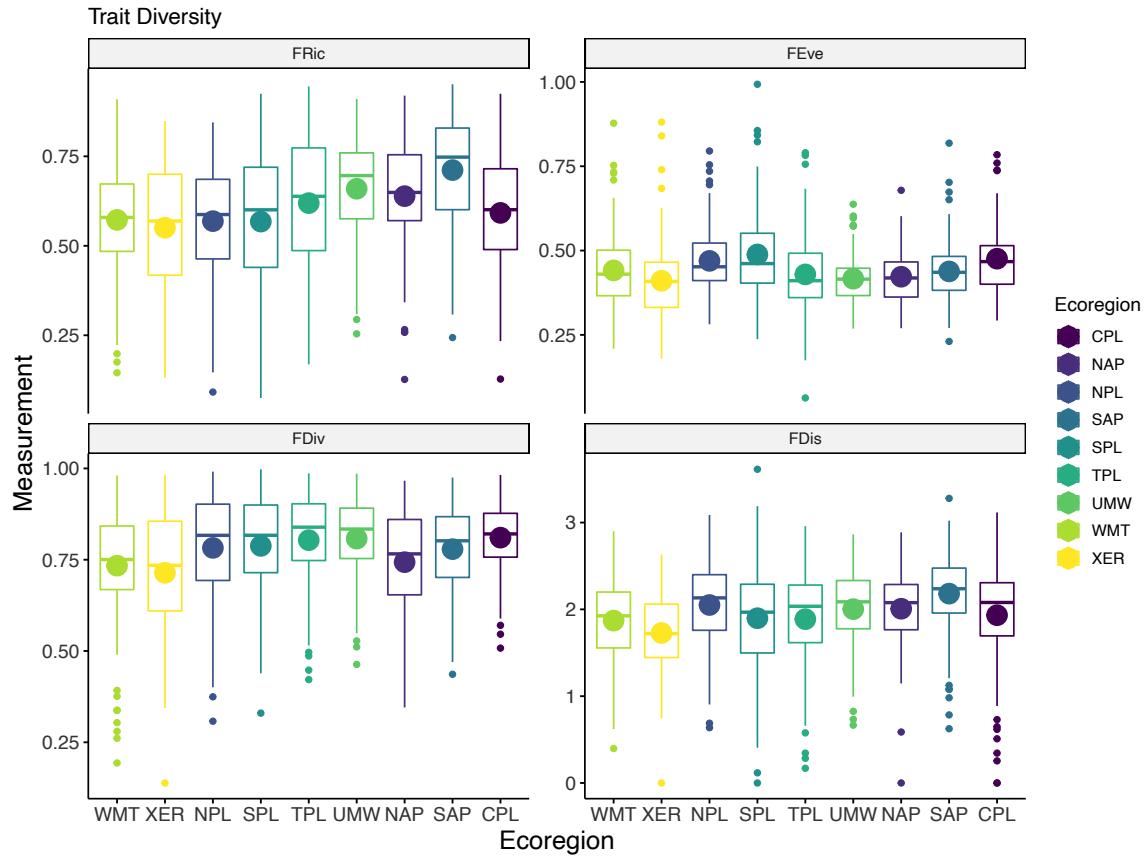


Figure 13: Plots of functional trait diversity by ecoregion.

Trait Abundance Calculation

```
## Subset taxa for each ecoregion (include UID for joining dataframes)
CPL.taxa <- CPL.data %>%
  select(c(1, 67:142))
NAP.taxa <- NAP.data %>%
  select(c(1, 67:142))
NPL.taxa <- NPL.data %>%
  select(c(1, 67:142))
SAP.taxa <- SAP.data %>%
  select(c(1, 67:142))
SPL.taxa <- SPL.data %>%
  select(c(1, 67:142))
TPL.taxa <- TPL.data %>%
  select(c(1, 67:142))
UMW.taxa <- UMW.data %>%
  select(c(1, 67:142))
WMT.taxa <- WMT.data %>%
  select(c(1, 67:142))
XER.taxa <- XER.data %>%
  select(c(1, 67:142))
```

```

## Function to calculate trait abundances for each site
trait_by_site_abundance <- function(j) {
  ## Load required packages
  require(dplyr)

  ## Set trait matrix
  trait.matrix <- trait.table.binary

  ## Sequentially calculate abundances for each trait for all sites in the supplied taxa matrix
  dispersal.low <- j * trait.matrix[, 2] [match(names(j), trait.matrix$taxon)][col(j)]
  dispersal.high <- j * trait.matrix[, 3] [match(names(j), trait.matrix$taxon)][col(j)]
  flying.none <- j * trait.matrix[, 4] [match(names(j), trait.matrix$taxon)][col(j)]
  flying.weak <- j * trait.matrix[, 5] [match(names(j), trait.matrix$taxon)][col(j)]
  flying.strong <- j * trait.matrix[, 6] [match(names(j), trait.matrix$taxon)][col(j)]
  size.small <- j * trait.matrix[, 7] [match(names(j), trait.matrix$taxon)][col(j)]
  size.medium <- j * trait.matrix[, 8] [match(names(j), trait.matrix$taxon)][col(j)]
  size.large <- j * trait.matrix[, 9] [match(names(j), trait.matrix$taxon)][col(j)]
  depositional <- j * trait.matrix[, 10] [match(names(j), trait.matrix$taxon)][col(j)]
  depositional.erosional <- j * trait.matrix[, 11] [match(names(j), trait.matrix$taxon)][col(j)]
  erosional <- j * trait.matrix[, 12] [match(names(j), trait.matrix$taxon)][col(j)]
  cold <- j * trait.matrix[, 13] [match(names(j), trait.matrix$taxon)][col(j)]
  cool.warm <- j * trait.matrix[, 14] [match(names(j), trait.matrix$taxon)][col(j)]
  warm <- j * trait.matrix[, 15] [match(names(j), trait.matrix$taxon)][col(j)]
  CG <- j * trait.matrix[, 16] [match(names(j), trait.matrix$taxon)][col(j)]
  CF <- j * trait.matrix[, 17] [match(names(j), trait.matrix$taxon)][col(j)]
  HB <- j * trait.matrix[, 18] [match(names(j), trait.matrix$taxon)][col(j)]
  PR <- j * trait.matrix[, 19] [match(names(j), trait.matrix$taxon)][col(j)]
  sensitive <- j * trait.matrix[, 20] [match(names(j), trait.matrix$taxon)][col(j)]
  intermediate <- j * trait.matrix[, 21] [match(names(j), trait.matrix$taxon)][col(j)]
  tolerant <- j * trait.matrix[, 22] [match(names(j), trait.matrix$taxon)][col(j)]

  ## Set list of traits; remove UID column
  trait.list <- list(
    dispersal.low[,-1], dispersal.high[,-1], flying.none[,-1], flying.weak[,-1],
    flying.strong[,-1], size.small[,-1], size.medium[,-1], size.large[,-1],
    depositional[,-1], depositional.erosional[,-1], erosional[,-1],
    cold[,-1], cool.warm[,-1], warm[,-1], CG[,-1], CF[,-1], HB[,-1],
    PR[,-1], sensitive[,-1], intermediate[,-1], tolerant[,-1]
  )
}

## Sum trait abundances by site
trait.abundances.bin.1 <- sapply(trait.list, FUN = rowSums, USE.NAMES = TRUE) %>%
  as_tibble()

## Set UID vector
UID.vector <- j[, 1] %>%
  as_tibble()

## Bind UID and trait abundances
trait.abundances.bin.2 <- bind_cols(UID.vector, trait.abundances.bin.1)

## Rename trait columns

```

```
colnames(trait.abundances.bin.2) [2:22] <- colnames(trait.matrix) [2:22]

## Rename UID column
colnames(trait.abundances.bin.2) [1] <- "UID"

## Set final trait abundance tibble
trait.abundances <- trait.abundances.bin.2 %>%
  as_tibble()
}
```

```

## List of taxa matrices for each ecoregion
taxa.matrix.list <- list(
  CPL.taxa, NAP.taxa, NPL.taxa,
  SAP.taxa, SPL.taxa, TPL.taxa,
  UMW.taxa, WMT.taxa, XER.taxa
)

## Calculate trait abundances for all sites
trait.abundance.list <- lapply(
  X = taxa.matrix.list,
  FUN = trait_by_site_abundance
)

## Rename each dataframe within the list
names(trait.abundance.list) <- c(
  "CPL.trait.abundance.data", "NAP.trait.abundance.data", "NPL.trait.abundance.data",
  "SAP.trait.abundance.data", "SPL.trait.abundance.data", "TPL.trait.abundance.data",
  "UMW.trait.abundance.data", "WMT.trait.abundance.data", "XER.trait.abundance.data"
)

## Export trait abundance data
list2env(trait.abundance.list, envir = .GlobalEnv)

## <environment: R_GlobalEnv>

```

Q3: Which functional traits are linked to filtering and habitat matching?

DisEQ-by-Traits Relationships

We used boosted regression trees (BRTs) to identify which functional traits were the best predictor of filtering and habitat matching. We fitted one set of BRTs with filtering as the response and one set of BRTs with habitat matching as the response, functional trait abundances were fitted as the predictor variables for both DisEQ-by-Trait BRTs. Our goal was to identify which functional traits were useful predictors of filtering and habitat matching, with separate BRTs for each ecoregion to allow for the relative influence of functional traits to vary by ecoregion.

DisEQ-by-Trait BRTs were: (1) fitted to a Gaussian error distribution, (2) fitted to 10,000 trees, (3) had a learning rate of 0.0001, (4) a minimum of 5 observations per terminal node, (5) had an interaction depth of 4, (6) had a bagging fraction of 50%, and (7) used ten-fold cross validation.

We fitted all BRTs using the `gbm()` function in the `gbm` package, with code parallelized using the `snow` package.

List of functional traits and levels:

- Dispersal ability = low, high
- Flying strength = none, weak, strong
- Body size = small, medium, large
- Rheophilic preference = depositional, depositional-erosional, erosional
- Thermal preference = cold water, cool-warm water, warm water
- Tolerance = sensitive, intermediate, tolerant
- Functional feeding group = CG, CF, HB, PR

Note: Shredders were not evaluated as they comprised less than 1% of all taxa

Traits were interpreted in three categories

- Dispersal = dispersal ability, flying strength, and body size
- Habitat = rheophilic preference, thermal preference, and tolerance
- Ecology = functional feeding group

References for BRTs:

- De'ath, G. 2007. Boosted trees for ecological modeling and prediction. *Ecology* 88: 243-251.
Elith, J., et al. 2008. A working guide to boosted regression trees. *Journal of Animal Ecology* 77: 802-813.

```

## Subset trait abundance and DisEQ data by ecoregion
CPL.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "CPL") %>%
  select(c(1:2, 4)) %>%
  left_join(CPL.trait.abundance.data, by = "UID")
NAP.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "NAP") %>%
  select(c(1:2, 4)) %>%
  left_join(NAP.trait.abundance.data, by = "UID")
NPL.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "NPL") %>%
  select(c(1:2, 4)) %>%
  left_join(NPL.trait.abundance.data, by = "UID")
SAP.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "SAP") %>%
  select(c(1:2, 4)) %>%
  left_join(SAP.trait.abundance.data, by = "UID")
SPL.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "SPL") %>%
  select(c(1:2, 4)) %>%
  left_join(SPL.trait.abundance.data, by = "UID")
TPL.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "TPL") %>%
  select(c(1:2, 4)) %>%
  left_join(TPL.trait.abundance.data, by = "UID")
UMW.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "UMW") %>%
  select(c(1:2, 4)) %>%
  left_join(UMW.trait.abundance.data, by = "UID")
WMT.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "WMT") %>%
  select(c(1:2, 4)) %>%
  left_join(WMT.trait.abundance.data, by = "UID")
XER.trait.DiseQ.data <- filter(final.DiseQ.data, ecoregion == "XER") %>%
  select(c(1:2, 4)) %>%
  left_join(XER.trait.abundance.data, by = "UID")

```

Filtering-by-Traits BRTs

```
## List of data for boosted regression analyses
filtering.by.trait.BRT.data.list <- list(
  CPL.filtering.trait.DisEQ.data <- CPL.trait.DisEQ.data %>% select(2, 4:24),
  NAP.filtering.trait.DisEQ.data <- NAP.trait.DisEQ.data %>% select(2, 4:24),
  NPL.filtering.trait.DisEQ.data <- NPL.trait.DisEQ.data %>% select(2, 4:24),
  SAP.filtering.trait.DisEQ.data <- SAP.trait.DisEQ.data %>% select(2, 4:24),
  SPL.filtering.trait.DisEQ.data <- SPL.trait.DisEQ.data %>% select(2, 4:24),
  TPL.filtering.trait.DisEQ.data <- TPL.trait.DisEQ.data %>% select(2, 4:24),
  UMW.filtering.trait.DisEQ.data <- UMW.trait.DisEQ.data %>% select(2, 4:24),
  WMT.filtering.trait.DisEQ.data <- WMT.trait.DisEQ.data %>% select(2, 4:24),
  XER.filtering.trait.DisEQ.data <- XER.trait.DisEQ.data %>% select(2, 4:24)
)

## Start cluster
cluster <- makeCluster(n.cores)

## Run the analysis for each ecoregion
filtering.by.trait.BRT.list <- parLapply(
  cluster,
  filtering.by.trait.BRT.data.list,
  fun = function(j) {
    require(gbm)

    ## Fit the BRT
    filtering.by.trait.BRT <- gbm(
      j[, 1] ~ .,
      distribution = "gaussian",
      data = j[, -1],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Habitat Matching-by-Traits BRTs

```
## List of data for boosted regression analyses
mismatch.by.trait.BRT.data.list <- list(
  CPL.mismatch.trait.DisEQ.data <- CPL.trait.DisEQ.data %>% select(3, 4:24),
  NAP.mismatch.trait.DisEQ.data <- NAP.trait.DisEQ.data %>% select(3, 4:24),
  NPL.mismatch.trait.DisEQ.data <- NPL.trait.DisEQ.data %>% select(3, 4:24),
  SAP.mismatch.trait.DisEQ.data <- SAP.trait.DisEQ.data %>% select(3, 4:24),
  SPL.mismatch.trait.DisEQ.data <- SPL.trait.DisEQ.data %>% select(3, 4:24),
  TPL.mismatch.trait.DisEQ.data <- TPL.trait.DisEQ.data %>% select(3, 4:24),
  UMW.mismatch.trait.DisEQ.data <- UMW.trait.DisEQ.data %>% select(3, 4:24),
  WMT.mismatch.trait.DisEQ.data <- WMT.trait.DisEQ.data %>% select(3, 4:24),
  XER.mismatch.trait.DisEQ.data <- XER.trait.DisEQ.data %>% select(3, 4:24)
)

## Start cluster
cluster <- makeCluster(n.cores)

## Run the analysis for the CUS and each ecoregion
mismatch.by.trait.BRT.list <- parLapply(
  cluster,
  mismatch.by.trait.BRT.data.list,
  fun = function(j) {
    require(gbm)

    ## Fit the BRT
    mismatch.by.trait.BRT <- gbm(
      j[, 1] ~ .,
      distribution = "gaussian",
      data = j[, -1],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

DisEQ-by-Trait Diversity ANCOVAs

We analyzed how the effects of functional trait diversity covary with ecoregion to affect environmental filtering and habitat matching using ANCOVAs. Four separate ANCOVAs were fitted for each response variable (i.e., environmental filtering and habitat matching). Model fits were assessed using `check_model()` in the `performance` package. Influence of predictors was estimated with Type III sums-of-squares using `Ancova()` in the `car` package, and effect sizes were estimated as partial η_P^2 using the `eta_squared()` function.

ANCOVAs were fitted as:

$$\text{DisEQ Metric} = \text{Intercept} + x_i + \text{Ecoregion} + (x_i \times \text{Ecoregion}) + e_i$$

where either environmental filtering or Habitat Matching was the response, x_i represented the main effect of the trait diversity metric (FRic, FEve, FDiv, or FDis), *Ecoregion* was the main effect of ecoregion, $(x_i \times \text{Ecoregion})$ was the interaction between trait diversity and ecoregion, and e_i was the residual error associated with the model.

DisEQ-by-FRic

ANCOVAs of environmental filtering or Habitat Matching against FRic, ecoregion, and the interaction.

```
filtering.FRic.ANCova <- lm(
  filtering.scaled ~ FRic * ecoregion,
  data = final.DisEQ.data
)
```

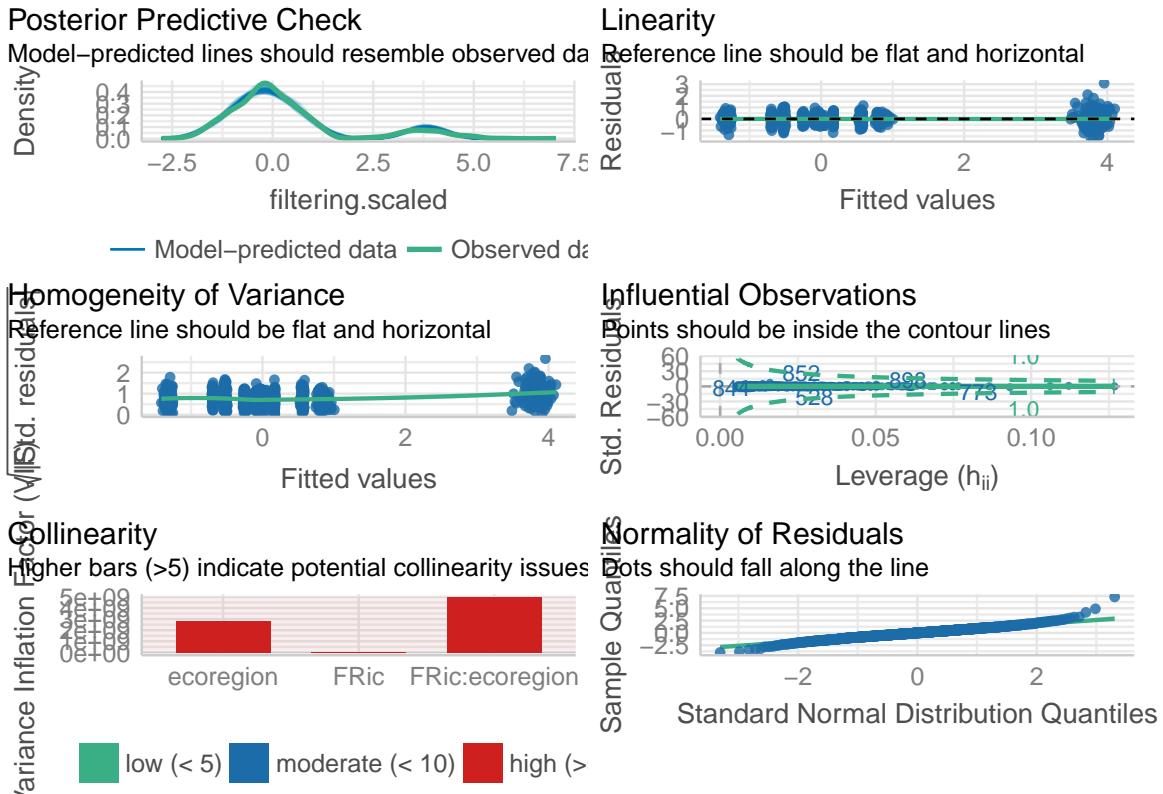


Figure 14: Diagnostic plots for the filtering and FRic ANCOVA.

Table 31: Summary of the environmental filtering by FRic ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	0.241	1	1.261	0.262
FRic	0.014	1	0.075	0.784
ecoregion	118.928	8	77.853	0.000
FRic:ecoregion	2.431	8	1.591	0.123
Residuals	193.813	1015	NA	NA

Table 32: Table of the effect sizes in the environmental filtering by FRic ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FRic	0.000	0.95	0.000	1
ecoregion	0.380	0.95	0.341	1
FRic:ecoregion	0.012	0.95	0.000	1

```

mismatch.FRic.ANCova <- lm(
  mismatch.scaled ~ FRic * ecoregion,
  data = final.DisEQ.data
)

```

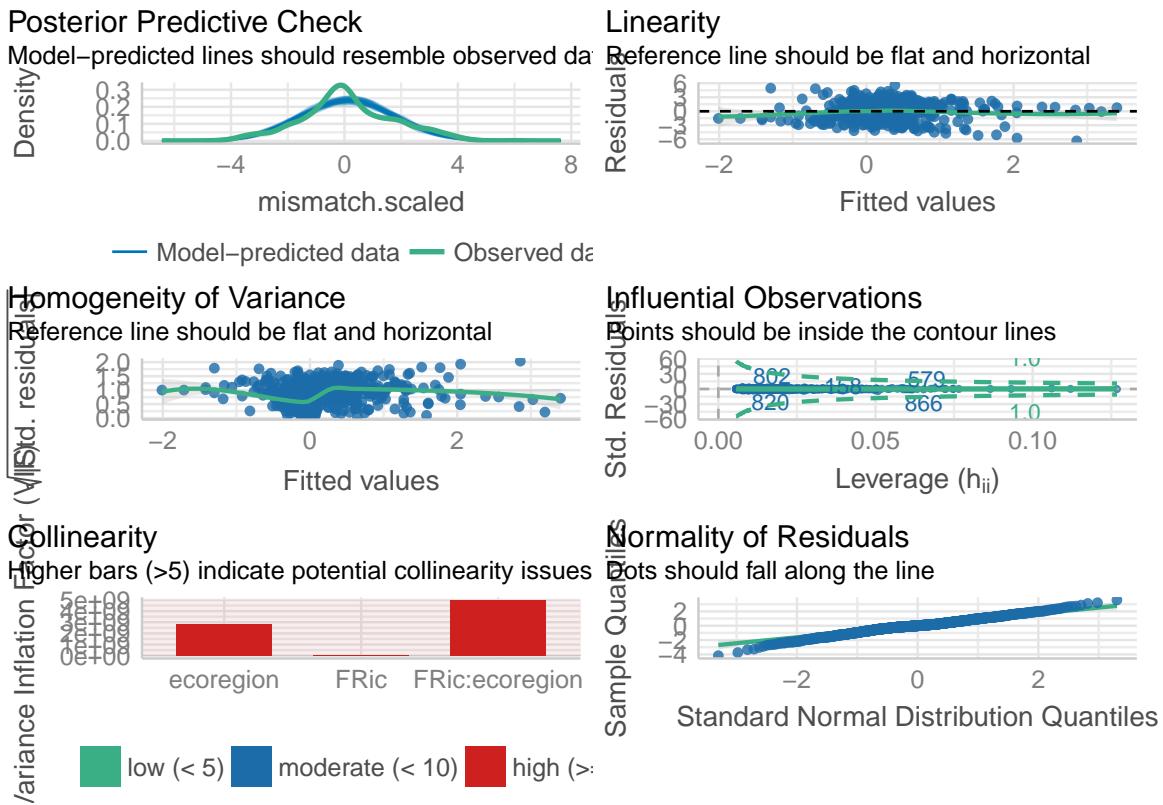


Figure 15: Diagnostic plots for the habitat matching and FRic ANCOVA.

Table 33: Summary of the habitat matching by FRic ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	5.385	1	2.188	0.139
FRic	5.842	1	2.373	0.124
ecoregion	155.866	8	7.915	0.000
FRic:ecoregion	145.489	8	7.388	0.000
Residuals	2498.434	1015	NA	NA

Table 34: Table of the effect sizes in the habitat matching by FRic ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FRic	0.002	0.95	0.000	1
ecoregion	0.059	0.95	0.032	1
FRic:ecoregion	0.055	0.95	0.029	1

DisEQ-by-FEve

ANCOVAs of environmental filtering or Habitat Matching against FEve, ecoregion, and the interaction.

```
filtering.FEve.ANCova <- lm(
  filtering.scaled ~ FEve * ecoregion,
  data = final.DiseQ.data
)
```

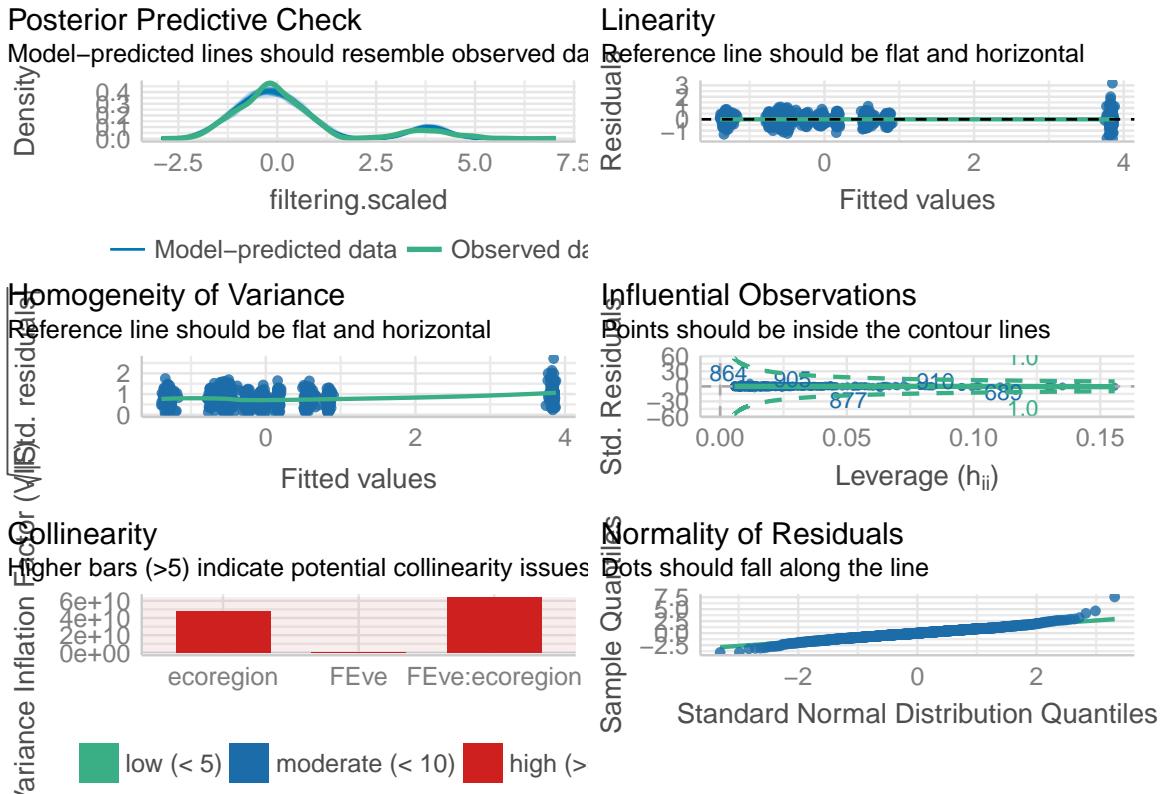


Figure 16: Diagnostic plots for the filtering and FEve ANCOVA.

Table 35: Summary of the environmental filtering by FEve ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	0.075	1	0.395	0.530
FEve	0.049	1	0.256	0.613
ecoregion	126.296	8	82.861	0.000
FEve:ecoregion	0.830	8	0.544	0.823
Residuals	195.668	1027	NA	NA

Table 36: Table of the effect sizes in the environmental filtering by FEve ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FEve	0.000	0.95	0.000	1
ecoregion	0.392	0.95	0.354	1
FEve:ecoregion	0.004	0.95	0.000	1

```

mismatch.FEve.ANCova <- lm(
  mismatch.scaled ~ FEve * ecoregion,
  data = final.DisEQ.data
)

```

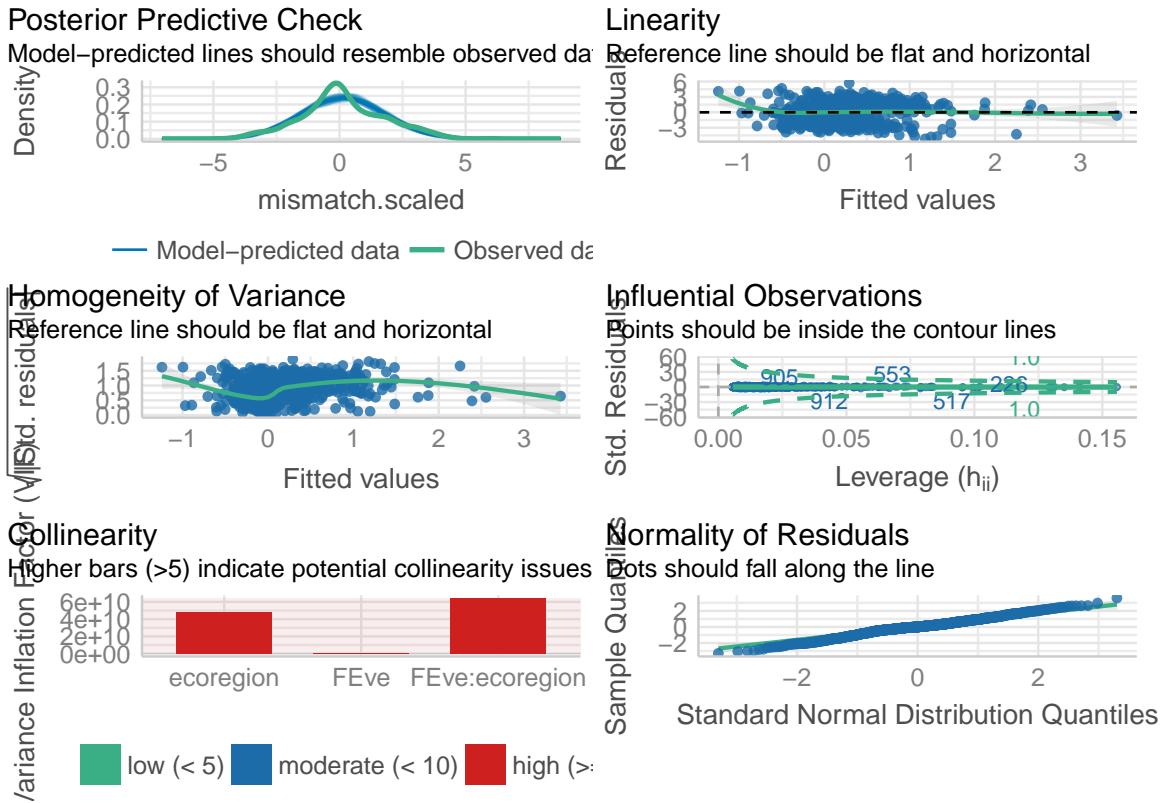


Figure 17: Diagnostic plots for the habitat matching and FEve ANCOVA.

Table 37: Summary of the habitat matching by FEve ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	10.170	1	4.022	0.045
FEve	10.812	1	4.277	0.039
ecoregion	66.267	8	3.276	0.001
FEve:ecoregion	87.722	8	4.337	0.000
Residuals	2596.543	1027	NA	NA

Table 38: Table of the effect sizes in the habitat matching by FEve ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FEve	0.004	0.95	0.000	1
ecoregion	0.025	0.95	0.006	1
FEve:ecoregion	0.033	0.95	0.012	1

DisEQ-by-FDiv

ANCOVAs of environmental filtering or Habitat Matching against FDiv, ecoregion, and the interaction.

```
filtering.FDiv.ANCova <- lm(
  filtering.scaled ~ FDiv * ecoregion,
  data = final.DisEQ.data
)
```

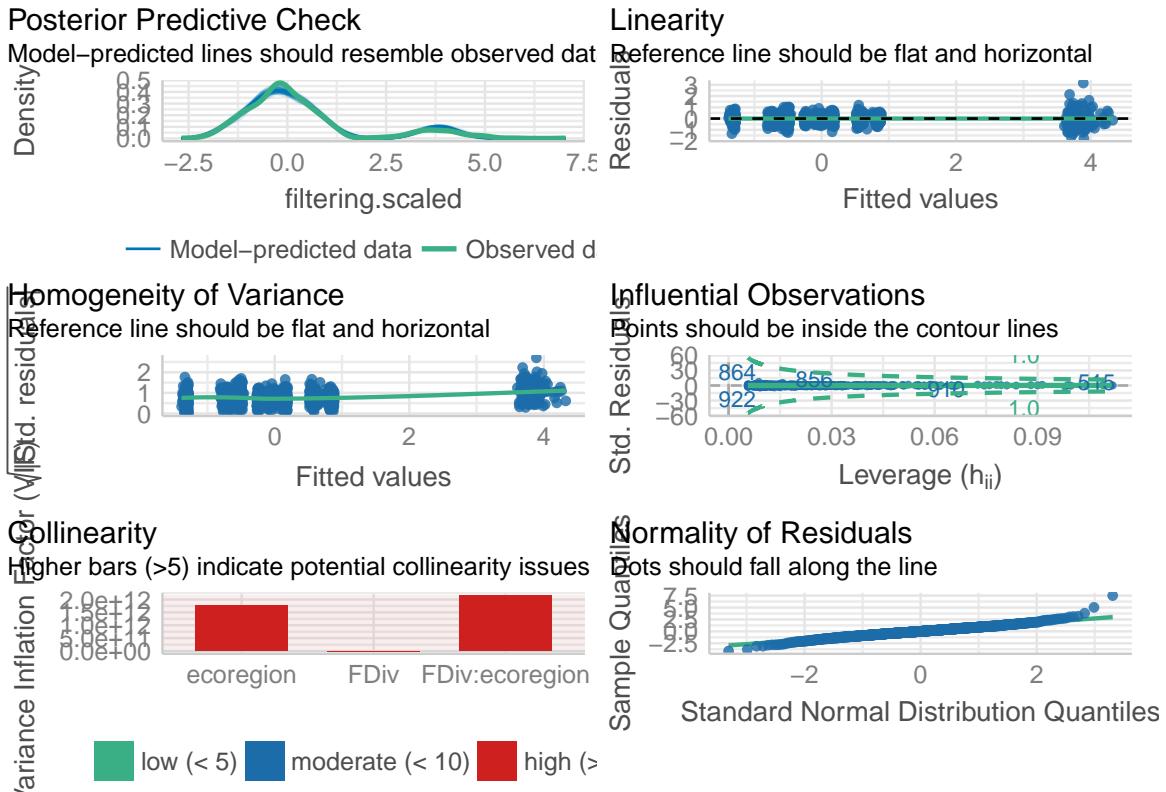


Figure 18: Diagnostic plots for the filtering and FDiv ANCOVA.

Table 39: Summary of the environmental filtering by FDiv ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	0.028	1	0.147	0.701
FDiv	0.014	1	0.076	0.783
ecoregion	116.864	8	77.745	0.000
FDiv:ecoregion	3.391	8	2.256	0.022
Residuals	192.968	1027	NA	NA

Table 40: Table of the effect sizes in the environmental filtering by FDiv ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FDiv	0.000	0.95	0.000	1
ecoregion	0.377	0.95	0.338	1
FDiv:ecoregion	0.017	0.95	0.001	1

```

mismatch.FDiv.ANCova <- lm(
  mismatch.scaled ~ FDiv * ecoregion,
  data = final.DisEQ.data
)

```

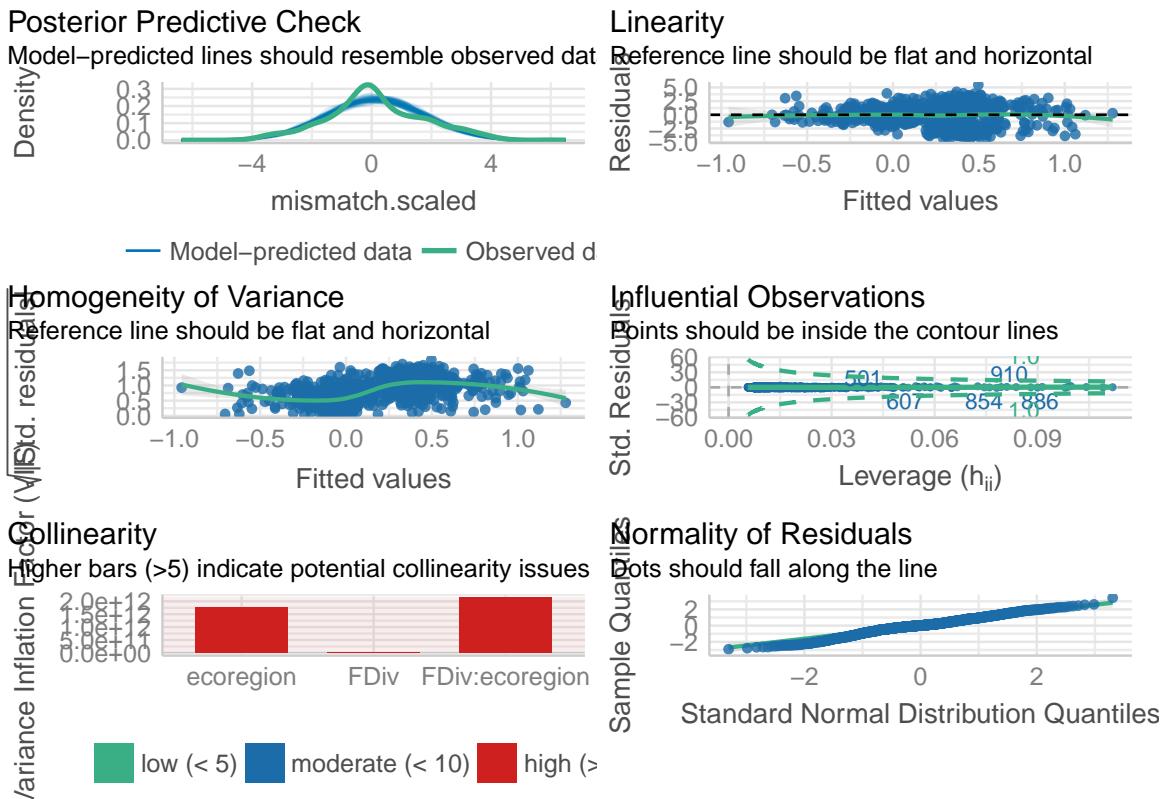


Figure 19: Diagnostic plots for the habitat matching and FDiv ANCOVA.

Table 41: Summary of the habitat matching by FDiv ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	15.725	1	6.051	0.014
FDiv	15.831	1	6.092	0.014
ecoregion	37.463	8	1.802	0.073
FDiv:ecoregion	48.264	8	2.322	0.018
Residuals	2668.753	1027	NA	NA

Table 42: Table of the effect sizes in the habitat matching by FDiv ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FDiv	0.006	0.95	0.001	1
ecoregion	0.014	0.95	0.000	1
FDiv:ecoregion	0.018	0.95	0.002	1

DisEQ-by-FDis

ANCOVAs of environmental filtering or Habitat Matching against FDis, ecoregion, and the interaction.

```
filtering.FDis.ANCova <- lm(
  filtering.scaled ~ FDis * ecoregion,
  data = final.DiseQ.data
)
```

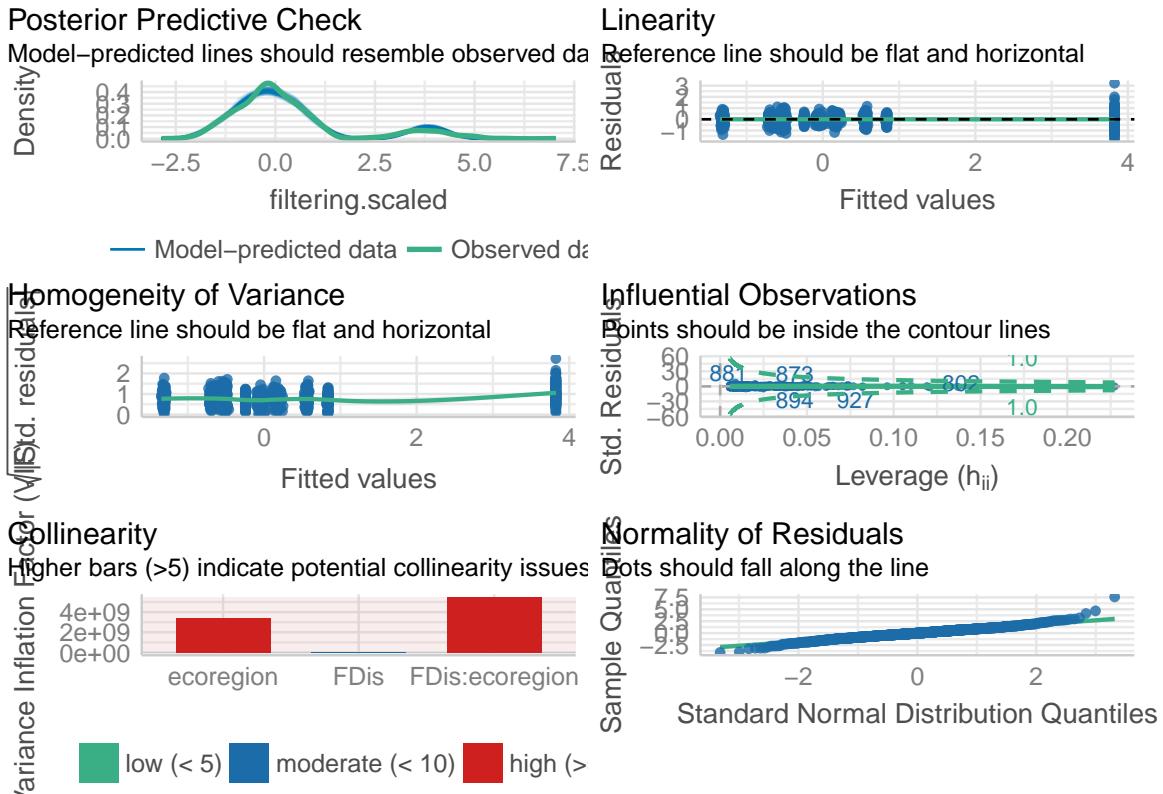


Figure 20: Diagnostic plots for the filtering and FDis ANCOVA.

Table 43: Summary of the environmental filtering by FDis ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	0.581	1	3.048	0.081
FDis	0.000	1	0.001	0.979
ecoregion	142.929	8	93.792	0.000
FDis:ecoregion	0.372	8	0.244	0.982
Residuals	199.250	1046	NA	NA

Table 44: Table of the effect sizes in the environmental filtering by FDis ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FDis	0.000	0.95	0.00	1
ecoregion	0.418	0.95	0.38	1
FDis:ecoregion	0.002	0.95	0.00	1

```

mismatch.FDis.ANCova <- lm(
  mismatch.scaled ~ FDis * ecoregion,
  data = final.DisEQ.data
)

```

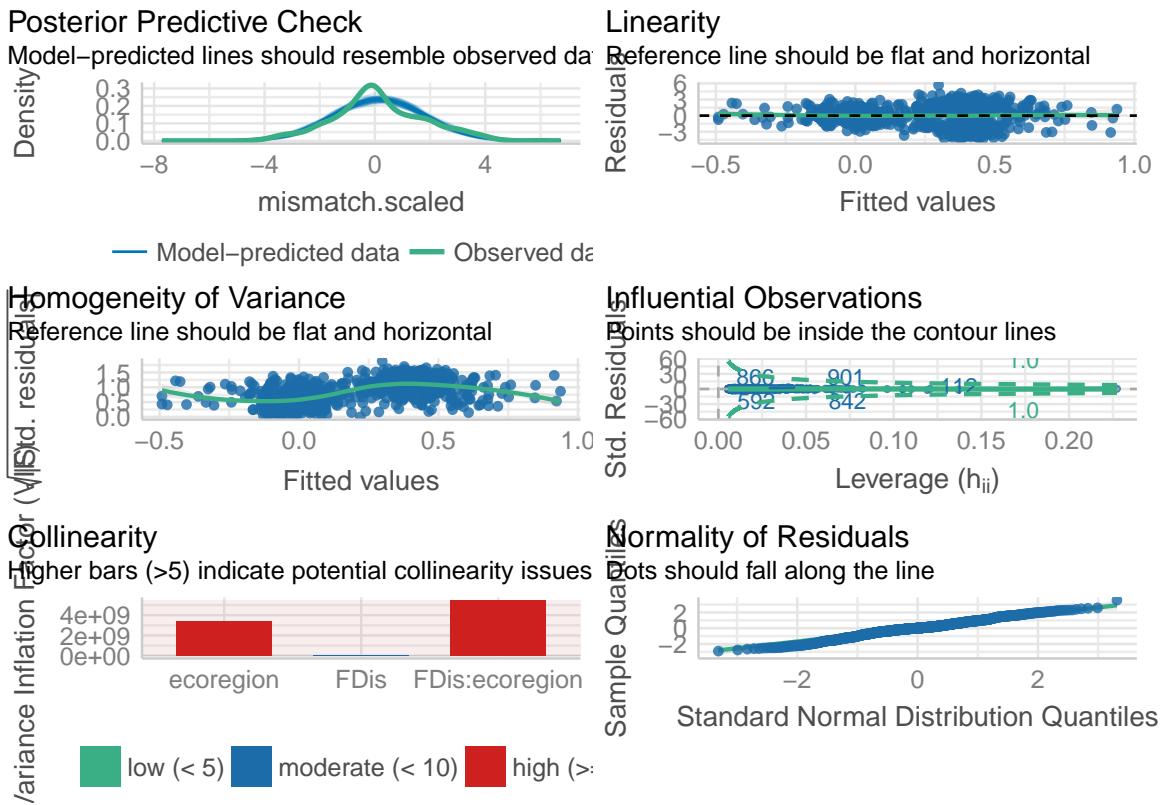


Figure 21: Diagnostic plots for the habitat matching and FDis ANCOVA.

Table 45: Summary of the habitat matching by FDis ANCOVA.

	Sums-of-Squares	df	F	P-value
(Intercept)	2.292	1	0.860	0.354
FDis	2.579	1	0.968	0.325
ecoregion	14.263	8	0.669	0.719
FDis:ecoregion	14.817	8	0.695	0.696
Residuals	2787.004	1046	NA	NA

Table 46: Table of the effect sizes in the habitat matching by FDis ANCOVA.

Parameter	Eta2_partial	CI	CI_low	CI_high
FDis	0.001	0.95	0	1
ecoregion	0.005	0.95	0	1
FDis:ecoregion	0.005	0.95	0	1

Response to Q3: Which functional traits are linked to filtering and habitat matching?

Functional trait predictors of environmental filtering and habitat matching varied by individual trait, trait category, and ecoregion. Environmental filtering was primarily influenced by habitat traits (7/9 ecoregions), with dispersal (4/9 ecoregions) and ecology (4/9 ecoregions) traits generally of secondary and tertiary influence, respectively. In contrast to environmental filtering, habitat matching was primarily influenced by both habitat (4/9 ecoregions) and ecology (4/9 ecoregions) traits. Habitat traits were also commonly the secondary influence on habitat matching (5/9 ecoregions), with both dispersal (4/9 ecoregions) and ecology (4/9 ecoregions) traits of tertiary importance.

Trait diversity was more frequently related to habitat matching than environmental filtering. Environmental filtering was only predicted by the interaction of FDiv and ecoregion ($F_{8,1027} = 2.256$, $P = 0.022$, $\eta_P^2 = 0.017$). Habitat matching was also influenced by interactions between trait diversity and ecoregion for FRic ($FRic \times Ecoregion$, $F_{8,1015} = 7.388$, $P < 0.001$, $\eta_P^2 = 0.055$), FEve ($FEve \times Ecoregion$, $F_{8,1027} = 4.337$, $P < 0.001$, $\eta_P^2 = 0.033$), and FDiv ($FDiv \times Ecoregion$, $F_{8,1027} = 2.322$, $P = 0.018$, $\eta_P^2 = 0.018$). There was no evidence for a main effect or interaction of FDis for either environmental filtering or Habitat Matching. Ecoregion consistently had the strongest effect on environmental filtering and habitat matching.

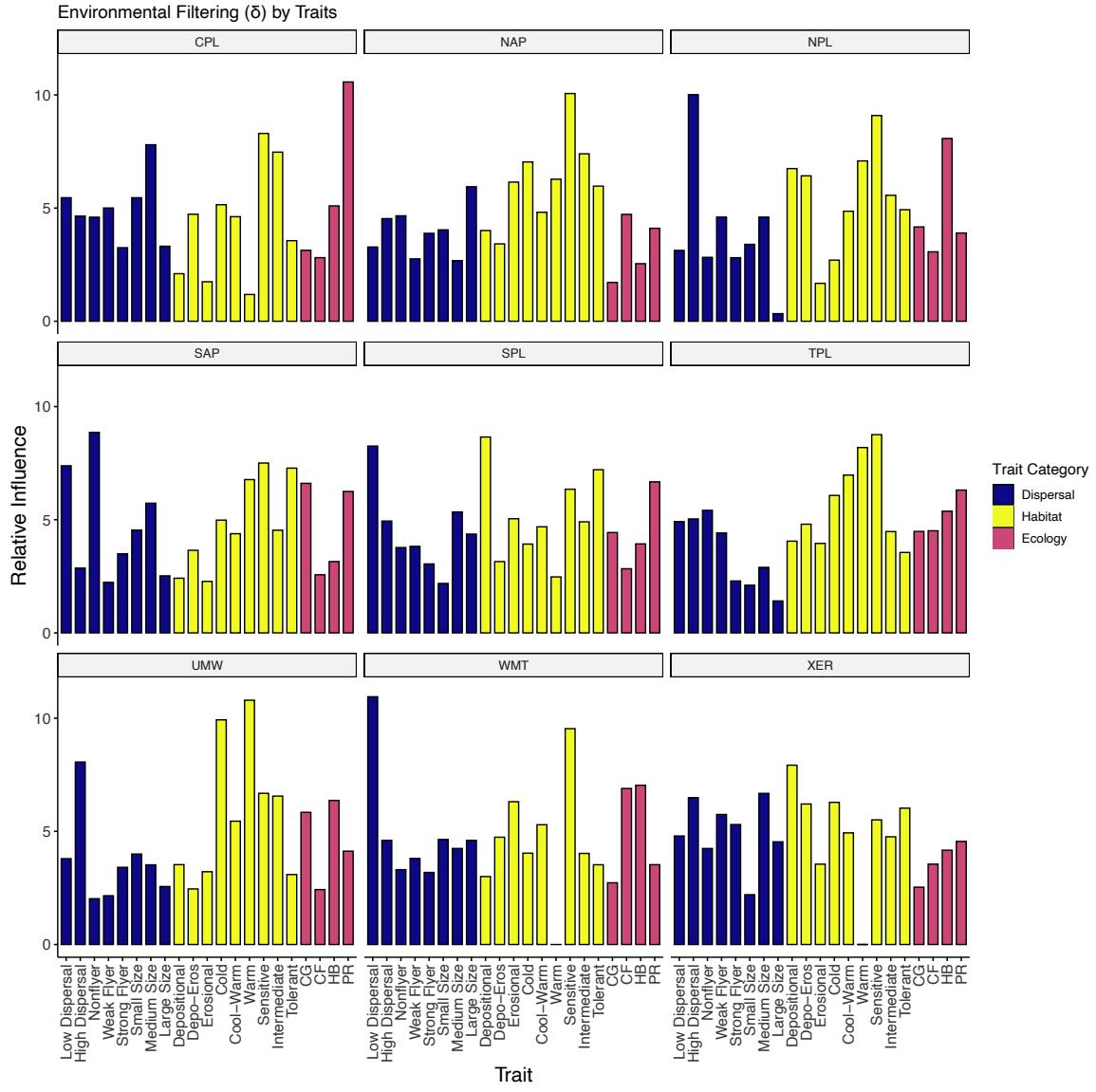


Figure 22: Facet plot of environmental filtering-by-trait relationships.

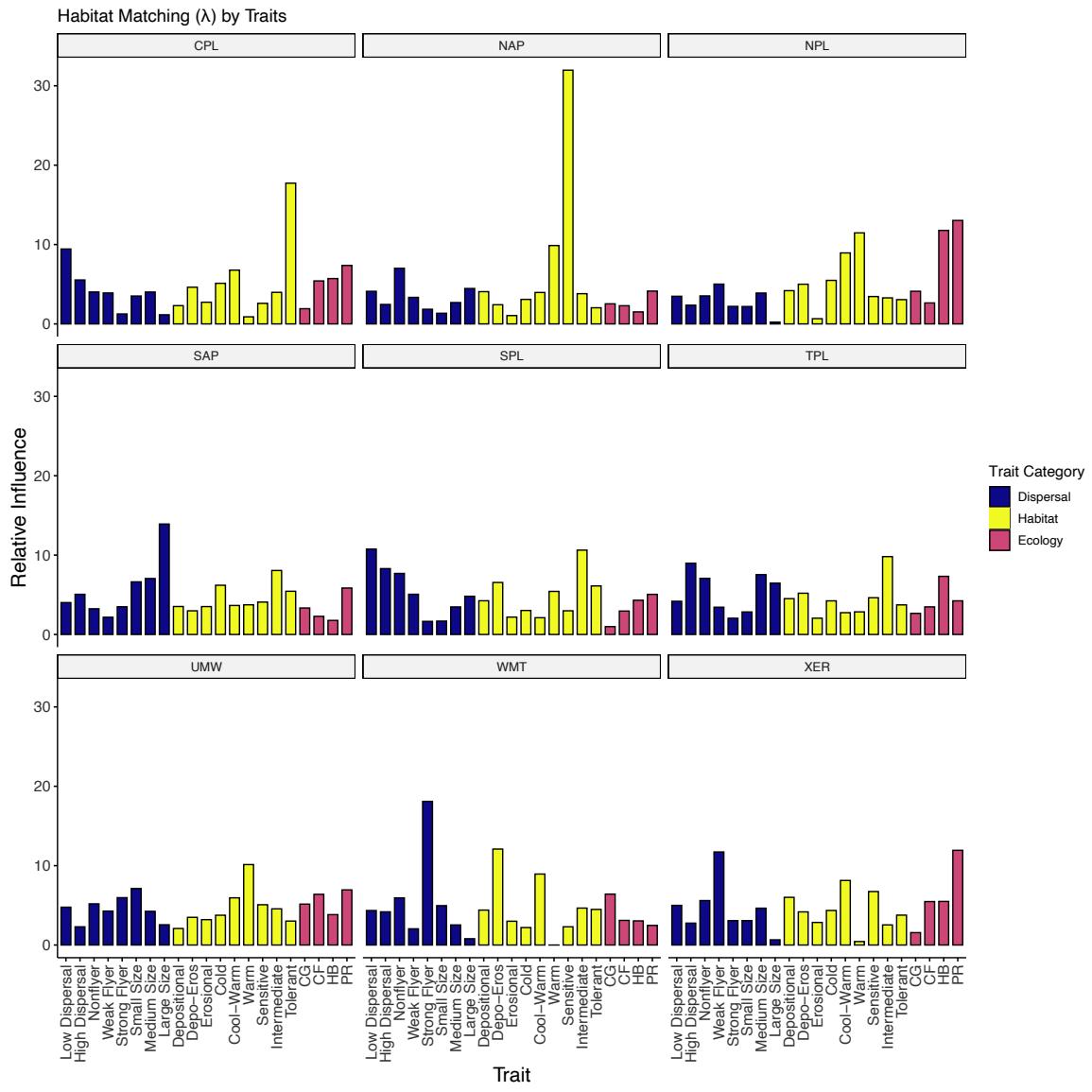


Figure 23: Facet plot of habitat matching-by-trait relationships.

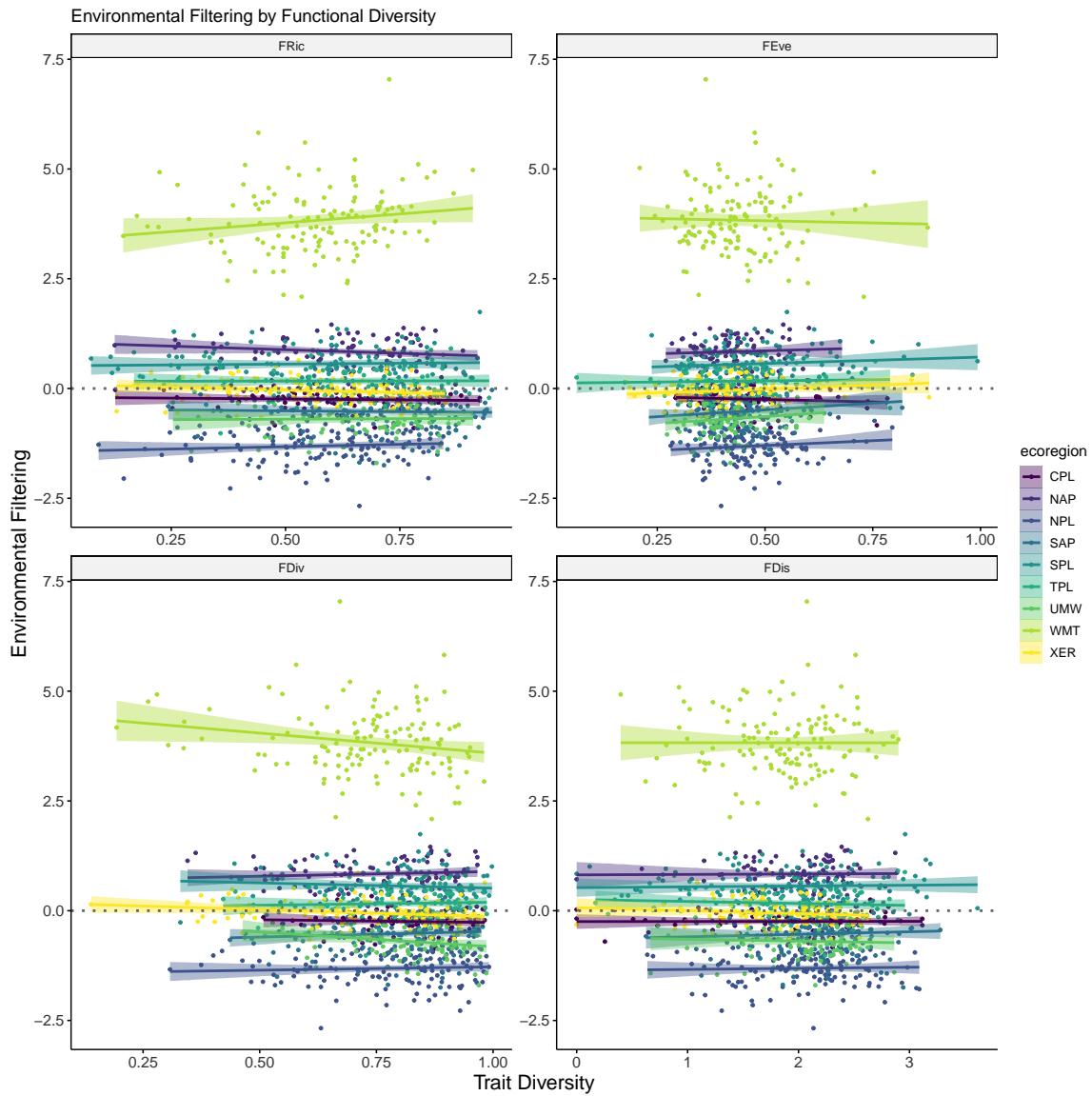


Figure 24: Regressions between environmental filtering and functional diversity.

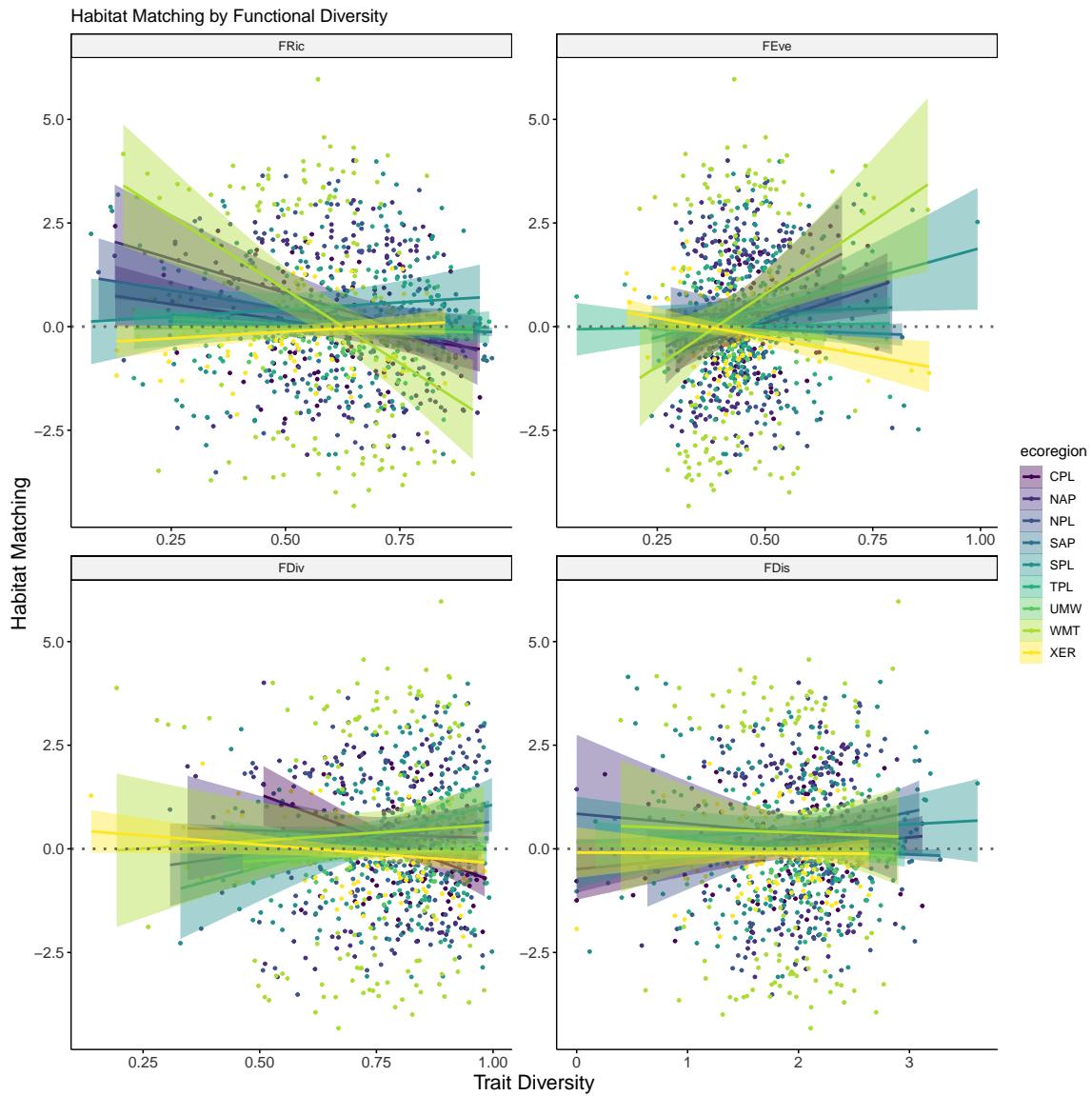


Figure 25: Regressions between habitat matching and functional diversity.

Q4: What are the environmental predictors of functional trait abundances?

Trait-by-Environment Relationships

We used boosted regression trees (BRTs) to identify which environmental variables were the best predictors of trait abundances. We fitted separate sets of BRTs for each of the 22 functional trait states (list of functional traits provided below), with each set of BRTs fitted with the trait as the response and a set of environmental variables as the predictors in the Trait-by-Environment BRTs. Environmental predictors variables were divided into three classes: (1) environmental, (2) landscape, and (3) network (list of variables and respective categories provided below). Our goal was to identify which variables were useful predictors of trait abundance, and then linking the Trait-by-Environment and DisEQ-by-Trait BRTs to better understand how (Trait-by-Environment) and why (DisEQ-by-Trait) filtering and mismatch vary.

Trait-by-Environment BRTs were fitted using the same parameters as the DisEQ-by-Trait BRTs: (1) fitted to a Poisson error distribution, (2) fitted to 10,000 trees, (3) had a learning rate of 0.0001, (4) a minimum of 5 observations per terminal node, (5) had an interaction depth of 4, (6) had a bagging fraction of 50%, and (7) used ten-fold cross validation.

We fitted all BRTs using the `gbm()` function in the `gbm` package, with code parallelized using the `snow` package.

List of functional traits and levels:

- Dispersal ability = low, high
- Flying strength = none, weak, strong
- Body size = small, medium, large
- Rheophilic preference = depositional, depositional-erosional, erosional
- Thermal preference = cold water, cool-warm water, warm water
- Tolerance = sensitive, intermediate, tolerant
- Functional feeding group = CG, CF, HB, PR

List of predictor variables of functional trait abundance and their respective categories:

- Environmental = total.N, total.P, DOC, LWD.reach, NAT.cover, ALG.cover, AQM.cover
- Landscape = pct.for, pct.ag, pct.urb, pct.ISC
- Network = site.lat, site.long, basin.area, mean.annual.flow, mean.basin.elevation, range.basin.elevation, site.centrality

Trait-by-Environment Data Management

```
## Trait abundance data
trait.abundance.data <- lapply(
  X = taxa.matrix.list,
  FUN = trait_by_site_abundance
) %>%
  bind_rows()

## Environment data
environment.data <- final.DisEQ.data %>%
  select(c(1, 12:30))
```

```

## Create a list of dataframes with trait abundance and environmental
## predictors for each trait

## Low dispersal
low.dispersal.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, dispersal.low) %>%
  inner_join(environment.data, by = "UID")

low.dispersal.environment.data <- split(
  low.dispersal.environment.data.bin.1, list(low.dispersal.environment.data.bin.1$ecoregion)
)

## High dispersal
high.dispersal.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, dispersal.high) %>%
  inner_join(environment.data, by = "UID")

high.dispersal.environment.data <- split(
  high.dispersal.environment.data.bin.1, list(high.dispersal.environment.data.bin.1$ecoregion)
)

## Nonflyer
nonflyer.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, flying.strength.none) %>%
  inner_join(environment.data, by = "UID")

nonflyer.environment.data <- split(
  nonflyer.environment.data.bin.1, list(nonflyer.environment.data.bin.1$ecoregion)
)

## Weak flyer
weak.flyer.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, flying.strength.weak) %>%
  inner_join(environment.data, by = "UID")

weak.flyer.environment.data <- split(
  weak.flyer.environment.data.bin.1, list(weak.flyer.environment.data.bin.1$ecoregion)
)

## Strong flyer
strong.flyer.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, flying.strength.strong) %>%
  inner_join(environment.data, by = "UID")

strong.flyer.environment.data <- split(
  strong.flyer.environment.data.bin.1, list(strong.flyer.environment.data.bin.1$ecoregion)
)

## Small size
small.size.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, size.small) %>%
  inner_join(environment.data, by = "UID")

```

```

small.size.environment.data <- split(
  small.size.environment.data.bin.1, list(small.size.environment.data.bin.1$ecoregion)
)

## Medium size
medium.size.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, size.medium) %>%
  inner_join(environment.data, by = "UID")

medium.size.environment.data <- split(
  medium.size.environment.data.bin.1, list(medium.size.environment.data.bin.1$ecoregion)
)

## Large size
large.size.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, size.large) %>%
  inner_join(environment.data, by = "UID")

large.size.environment.data <- split(
  large.size.environment.data.bin.1, list(large.size.environment.data.bin.1$ecoregion)
)

## Depositional
deposition.alternate.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, depositional) %>%
  inner_join(environment.data, by = "UID")

deposition.alternate.environment.data <- split(
  deposition.alternate.environment.data.bin.1, list(deposition.alternate.environment.data.bin.1$ecoregion)
)

## Depositional and erosional
deposition.erosional.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, depositional.erosional) %>%
  inner_join(environment.data, by = "UID")

deposition.erosional.environment.data <- split(
  deposition.erosional.environment.data.bin.1, list(deposition.erosional.environment.data.bin.1$ecoregion)
)

## Erosional
erosional.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, erosional) %>%
  inner_join(environment.data, by = "UID")

erosional.environment.data <- split(
  erosional.environment.data.bin.1, list(erosional.environment.data.bin.1$ecoregion)
)

## Cold water
cold.water.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, cold) %>%
  inner_join(environment.data, by = "UID")

```

```

cold.water.environment.data <- split(
  cold.water.environment.data.bin.1, list(cold.water.environment.data.bin.1$ecoregion)
)

## Cool-warm water
cool.warm.water.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, cool.warm) %>%
  inner_join(environment.data, by = "UID")

cool.warm.water.environment.data <- split(
  cool.warm.water.environment.data.bin.1, list(cool.warm.water.environment.data.bin.1$ecoregion)
)

## Warm water
warm.water.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, warm) %>%
  inner_join(environment.data, by = "UID")

warm.water.environment.data <- split(
  warm.water.environment.data.bin.1, list(warm.water.environment.data.bin.1$ecoregion)
)

## Sensitive tolerance
sensitive.tolerance.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, sensitive) %>%
  inner_join(environment.data, by = "UID")

sensitive.tolerance.environment.data <- split(
  sensitive.tolerance.environment.data.bin.1, list(sensitive.tolerance.environment.data.bin.1$ecoregion)
)

## Intermediate tolerance
intermediate.tolerance.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, medium) %>%
  inner_join(environment.data, by = "UID")

intermediate.tolerance.environment.data <- split(
  intermediate.tolerance.environment.data.bin.1, list(intermediate.tolerance.environment.data.bin.1$ecoregion)
)

## Tolerant
tolerant.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, tolerant) %>%
  inner_join(environment.data, by = "UID")

tolerant.environment.data <- split(
  tolerant.environment.data.bin.1, list(tolerant.environment.data.bin.1$ecoregion)
)

## Collector-gather
CG.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, CG) %>%
  inner_join(environment.data, by = "UID")

```

```

CG.environment.data <- split(
  CG.environment.data.bin.1, list(CG.environment.data.bin.1$ecoregion)
)

## Collector-filterer
CF.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, CF) %>%
  inner_join(environment.data, by = "UID")

CF.environment.data <- split(
  CF.environment.data.bin.1, list(CF.environment.data.bin.1$ecoregion)
)

## Herbivore
HB.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, HB) %>%
  inner_join(environment.data, by = "UID")

HB.environment.data <- split(
  HB.environment.data.bin.1, list(HB.environment.data.bin.1$ecoregion)
)

## Predator
PR.environment.data.bin.1 <- trait.abundance.data %>%
  select(UID, PR) %>%
  inner_join(environment.data, by = "UID")

PR.environment.data <- split(
  PR.environment.data.bin.1, list(PR.environment.data.bin.1$ecoregion)
)

```

Low Dispersal-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the low dispersal-by-environment BRT for each ecoregion
low.dispersal.by.environment.BRT.list <- parLapply(
  cluster,
  low.dispersal.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      dispersal.low ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

High Dispersal-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the high dispersal-by-environment BRT for each ecoregion
high.dispersal.by.environment.BRT.list <- parLapply(
  cluster,
  high.dispersal.environment.data,
  fun = function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      dispersal.high ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Nonflyer-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the nonflyer-by-environment BRT for each ecoregion
nonflyer.by.environment.BRT.list <- parLapply(
  cluster,
  nonflyer.environment.data,
  fun = function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      flying.strength.none ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Weak Flyer-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the weak flyer-by-environment BRT for each ecoregion
weak.flyer.by.environment.BRT.list <- parLapply(
  cluster,
  weak.flyer.environment.data,
  fun = function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      flying.strength.weak ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Strong Flyer-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the strong flyer-by-environment BRT for each ecoregion
strong.flyer.by.environment.BRT.list <- parLapply(
  cluster,
  strong.flyer.environment.data,
  fun = function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      flying.strength.strong ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Small Size-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the small size-by-environment BRT for each ecoregion
small.size.by.environment.BRT.list <- parLapply(
  cluster,
  small.size.environment.data,
  fun = function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      size.small ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Medium Size-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the medium size-by-environment BRT for each ecoregion
medium.size.by.environment.BRT.list <- parLapply(
  cluster,
  medium.size.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      size.medium ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Large Size-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the large size-by-environment BRT for each ecoregion
large.size.by.environment.BRT.list <- parLapply(
  cluster,
  large.size.environment.data,
  fun = function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      size.large ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Depositional-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the depositional-by-environment BRT for each ecoregion
depositional.by.environment.BRT.list <- parLapply(
  cluster,
  depositional.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      depositional ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Depositional-Erosional-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the depositional-erosional-by-environment BRT for each ecoregion
depositional.erosional.by.environment.BRT.list <- parLapply(
  cluster,
  depositional.erosional.environment.data,
  fun = function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      depositional.erosional ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Erosional-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the erosional-by-environment BRT for each ecoregion
erosional.by.environment.BRT.list <- parLapply(
  cluster,
  erosional.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      erosional ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Cold Water-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the cold water-by-environment BRT for each ecoregion
cold.water.by.environment.BRT.list <- parLapply(
  cluster,
  cold.water.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      cold ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Cool-Warm Water-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the cool-warm water-by-environment BRT for each ecoregion
cool.warm.water.by.environment.BRT.list <- parLapply(
  cluster,
  cool.warm.water.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      cool.warm ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Warm Water-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the warm water-by-environment BRT for each ecoregion
warm.water.by.environment.BRT.list <- parLapply(
  cluster,
  warm.water.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      warm ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Sensitive Tolerance-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the sensitive tolerance-by-environment BRT for each ecoregion
sensitive.tolerance.by.environment.BRT.list <- parLapply(
  cluster,
  sensitive.tolerance.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      sensitive ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Intermediate Tolerance-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the intermediate tolerance-by-environment BRT for each ecoregion
intermediate.tolerance.by.environment.BRT.list <- parLapply(
  cluster,
  intermediate.tolerance.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      medium ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Tolerant-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the tolerant-by-environment BRT for each ecoregion
tolerant.by.environment.BRT.list <- parLapply(
  cluster,
  tolerant.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      tolerant ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Collector Gatherer-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the CG-by-environment BRT for each ecoregion
CG.by.environment.BRT.list <- parLapply(
  cluster,
  CG.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      CG ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Collector Filterer-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the CF-by-environment BRT for each ecoregion
CF.by.environment.BRT.list <- parLapply(
  cluster,
  CF.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      CF ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Herbivore-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the HB-by-environment BRT for each ecoregion
HB.by.environment.BRT.list <- parLapply(
  cluster,
  HB.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      HB ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Predator-by-Environment BRTs

```
## Start cluster
cluster <- makeCluster(n.cores)

## Run the PR-by-environment BRT for each ecoregion
PR.by.environment.BRT.list <- parLapply(
  cluster,
  PR.environment.data,
  fun =  function(j) {
    require(gbm)

    ## BRT fitting
    trait.by.DisEQ.BRT <- gbm(
      PR ~ .,
      distribution = "poisson",
      data = j[, -c(1, 3)],
      n.trees = 10000,
      interaction.depth = 4,
      n.minobsinnode = 5,
      shrinkage = 0.0001,
      bag.fraction = 0.5,
      cv.folds = 10
    )
  }
)

## Stop cluster
stopCluster(cluster)
```

Response to Q4: What are the environmental predictors of functional trait abundances?

Predictors of functional traits varied by trait category and ecoregion (Figure 6). Dispersal traits were primarily influenced by network (39/72 trait-by-ecoregion combinations) and environmental (23/81 trait-by-ecoregion combinations) predictors, with environmental predictors also frequently of secondary influence (41/81 trait-by-ecoregion combinations) and landscape predictors of tertiary influence (39/81 trait-by-ecoregion combinations). Similarly, habitat traits were primarily influenced by network (48/81 trait-by-ecoregion combinations) and environmental (29/81 trait-by-ecoregion combinations) predictors; environmental predictors were commonly of secondary influence (37/81 trait-by-ecoregion combinations) and landscape predictors of tertiary influence (40/81 trait-by-ecoregion combinations). Ecology traits were primarily structured by environmental (17/36 trait-by-ecoregion combinations) and network (15/36 trait-by-ecoregion combinations) predictors. Each of environmental, landscape, and network predictors were frequently of secondary influence for ecology traits (environmental = 12/36, landscape = 10/36, network = 14/36 trait-by-ecoregion combinations), which was in contrast to dispersal and habitat traits where environmental predictors were most commonly of secondary influence; however, we again identified landscape predictors to most commonly be of tertiary importance (20/36 trait-by-ecoregion combinations).

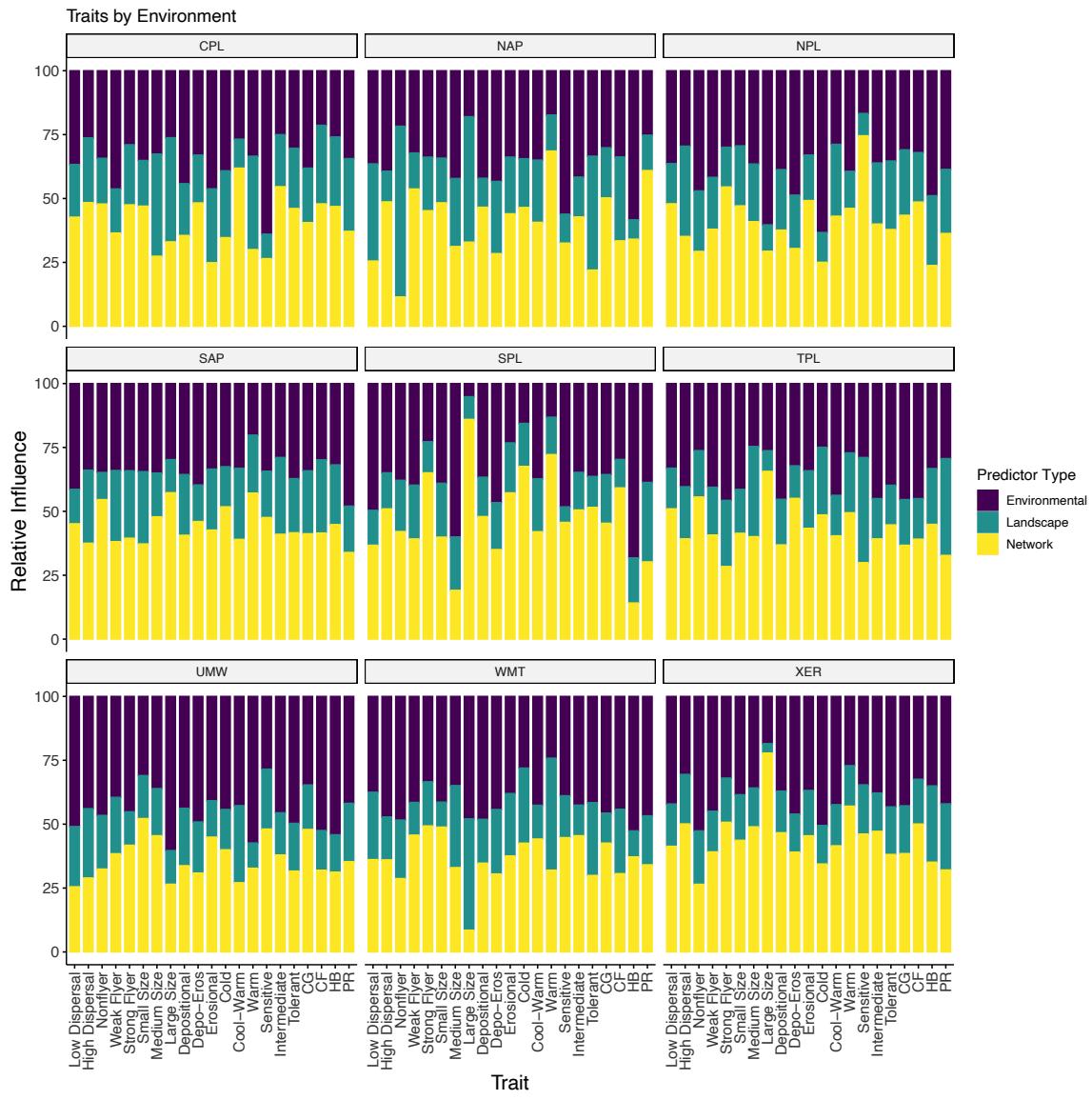


Figure 26: Facet plot of trait-by-environment relationships.

Supplementary Results

Environmental Filtering & Habitat Matching Model

We fitted a linear mixed-effects model to analyze the relationship between environmental filtering and habitat matching. The model was fitted as:

$$\text{Environmental Filtering} = \text{Intercept} + \text{Habitat Matching} + (1 | \text{Ecoregion}) + e$$

where environmental filtering was the response, habitat matching was the predict, ecoregion was a random intercept, and e was the residual error. Model fit was assessed using `check_model()` in the `performance` package. Influence of predictors was estimated with Type III sums-of-squares with Kenward-Roger denominator degrees of freedom, and effect sizes were estimated as η^2 using the `eta_squared()` function.

```
DisEQ.LMM <- lmer(
  filtering.scaled ~ mismatch.scaled + (1 | ecoregion),
  data = final.DisEQ.data,
  REML = TRUE
)
```

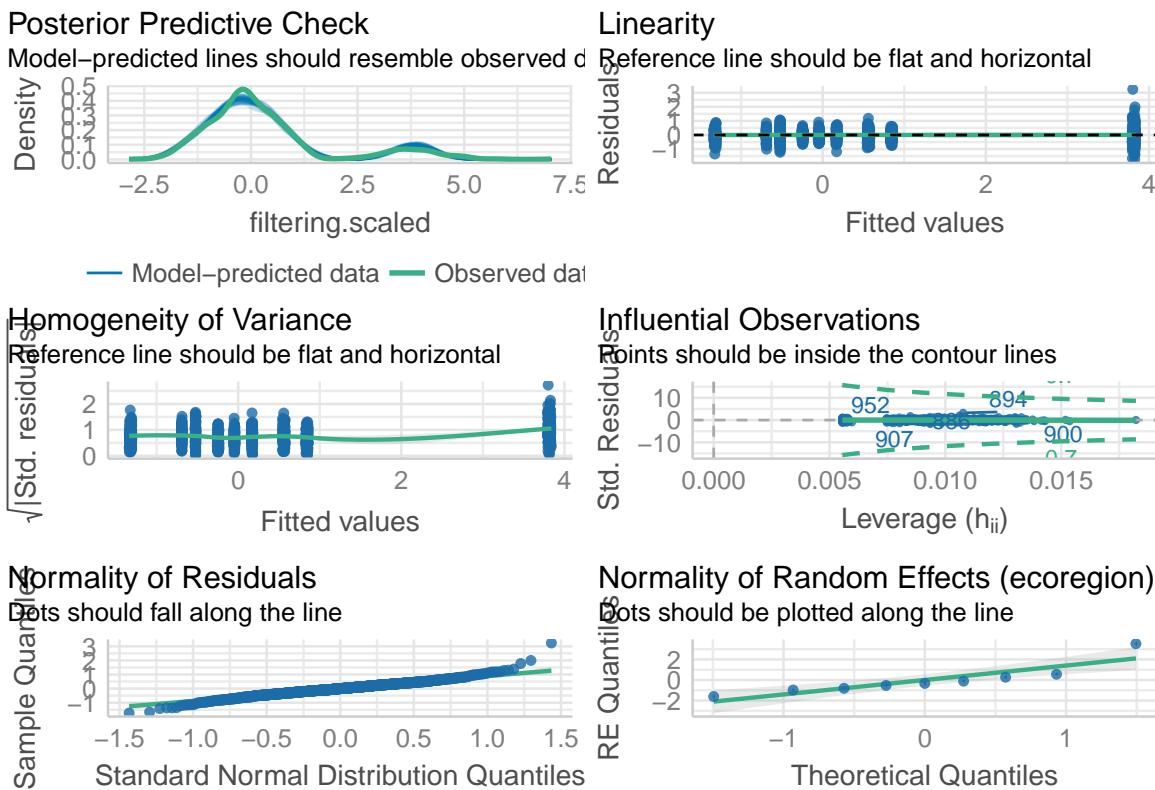


Figure 27: Diagnostic plots for the environmental filtering and habitat matching linear mixed-effects model.

```
## Fit an ANOVA with Type III sums-of-squares
DisEQ.LMM.anova <- anova(
  object = DisEQ.LMM,
  type = "III",
```

```

ddf = "Kenward-Roger"
)

```

Table 47: ANOVA table for the environmental filtering by habitat matching linear mixed-effects model.

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
mismatch.scaled	0.058	0.058	1	1068.026	0.307	0.58

Table 48: Table of coefficients for the environmental filtering by habitat matching linear mixed-effects model. Approximately 90% of the variation in the model was explained solely by ecoregion.

	Estimate	Std. Error	df	t value	Pr(> t)
(Intercept)	0.284	0.492	8.001	0.578	0.579
mismatch.scaled	0.005	0.008	1068.027	0.554	0.580

Table 49: Effect size for Habitat Matching in the environmental filtering by habitat matching linear mixed-effects model.

Term	η^2	Confidence Level	CI_{lower}	CI_{upper}
mismatch.scaled	0	0.95	0	1

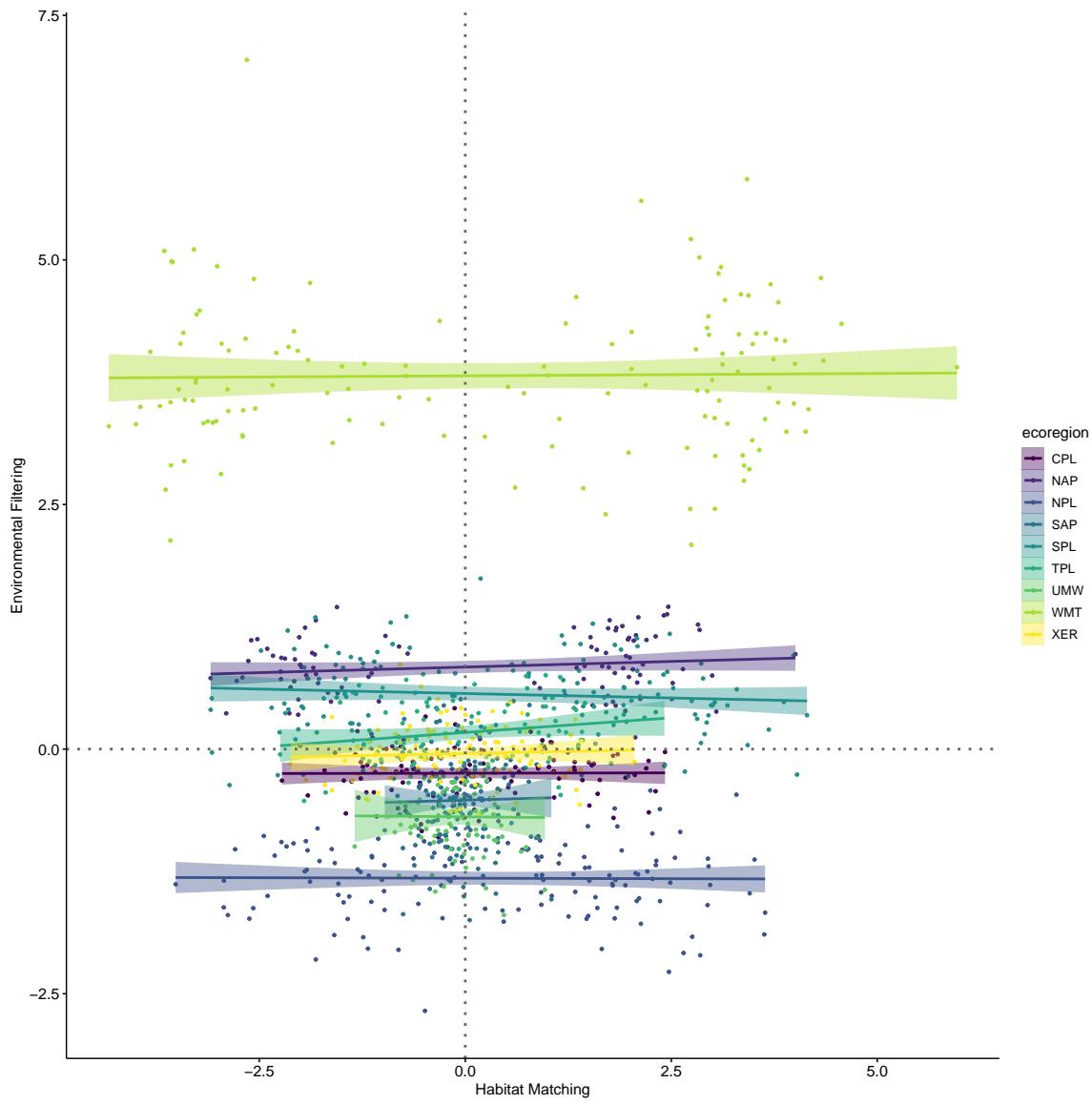


Figure 28: Plot of environmental filtering against habitat matching while controlling for the effect of ecoregion.

Filtering & Habitat Matching Summaries

Table 50: Numerical summary of environmental filtering by ecoregion.

ecoregion	N	Mean	SD
CPL	100	-0.246	0.256
NAP	97	0.845	0.308
NPL	134	-1.320	0.387
SAP	180	-0.523	0.451
SPL	125	0.561	0.392
TPL	110	0.170	0.363
UMW	86	-0.692	0.380
WMT	134	3.817	0.750
XER	112	-0.046	0.299

Table 51: Numerical summary of habitat matching by ecoregion.

ecoregion	N	Mean	SD
CPL	100	0.068	1.174
NAP	97	0.272	2.016
NPL	134	0.368	1.785
SAP	180	-0.044	0.380
SPL	125	0.418	2.044
TPL	110	0.009	0.996
UMW	86	-0.086	0.408
WMT	134	0.423	2.956
XER	112	-0.107	0.828

Trait Diversity Summaries

Table 52: Numerical summary of functional richness (FRic) by ecoregion.

ecoregion	N	Mean	SD
CPL	91	0.592	0.159
NAP	92	0.639	0.158
NPL	128	0.569	0.162
SAP	177	0.712	0.152
SPL	110	0.568	0.193
TPL	106	0.619	0.186
UMW	86	0.659	0.139
WMT	132	0.572	0.147
XER	111	0.550	0.176

Table 53: Numerical summary of functional evenness (FEve) by ecoregion.

ecoregion	N	Mean	SD
CPL	93	0.476	0.102
NAP	92	0.422	0.078
NPL	131	0.469	0.093
SAP	178	0.438	0.081
SPL	113	0.488	0.131
TPL	109	0.429	0.116
UMW	86	0.416	0.075
WMT	132	0.441	0.107
XER	111	0.410	0.117

Table 54: Numerical summary of functional divergence (FDiv) by ecoregion.

ecoregion	N	Mean	SD
CPL	93	0.810	0.099
NAP	92	0.743	0.141
NPL	131	0.782	0.154
SAP	178	0.779	0.121
SPL	113	0.788	0.145
TPL	109	0.804	0.129
UMW	86	0.808	0.120
WMT	132	0.734	0.156
XER	111	0.714	0.178

Table 55: Numerical summary of functional Dispersion (FDis) by ecoregion.

ecoregion	N	Mean	SD
CPL	96	1.933	0.644
NAP	93	2.007	0.450
NPL	132	2.050	0.493
SAP	178	2.181	0.438
SPL	124	1.898	0.672
TPL	110	1.888	0.609
UMW	86	2.003	0.439
WMT	133	1.870	0.499
XER	112	1.728	0.458

Community Composition

We evaluated differences in community composition using a permutational multivariate analysis of variance (PERMANOVA). We applied a Bray–Curtis dissimilarity index to a square-root transformed abundance matrix for each ecoregional metacommunity and then compared dissimilarities using a PERMANOVA with 10000 permutations. Results were illustrated using non-metric multi-dimensional scaling (NMDS); three dimensions were required to represent the communities in ordination space with acceptable stress (stress ≤ 0.20).

For additional reading on PERMANOVA, please see:

Anderson, M. J. 2001. A new method for non-parametric multivariate analysis of variance. *Austral Ecology* 26:32–46.

```
## Transform the community matrix (square-root)
BC.community.matrix <- sqrt(final.data[rowSums(final.data[, 67:142]) > 0, 67:142])

## Calculate the Bray-Curtis dissimilarity
BC.distance <- vegdist(
  BC.community.matrix,
  method = "bray",
  binary = FALSE
)

BC.NMDS <- monoMDS(
  BC.distance, k = 3, # 3 dimensions to reduce ordination stress
  model = "global",
  scaling = TRUE,
  maxit = 500
)

## PERMANOVA
BC.PERMANOVA <- adonis(
  community.matrix ~ ecoregion,
  data = ecoregion.vector.PERMANOVA,
  method = "bray",
  sqrt.dist = TRUE,
  permutations = 10000
)
```

Table 56: Results of the PERMANOVA.

	df	SS	MS	F	\$R^2\$	P-value
ecoregion	8	34.888	4.361	11.9	0.083	0
Residuals	1055	386.631	0.366	NA	0.917	NA
Total	1063	421.519	NA	NA	1.000	NA

R Session Information

Table 57: Packages required for data management and analyses.

Package	Loaded	Version	Date
ade4		1.7-18	2021-09-16
agricolae		1.3-5	2021-06-06
ape		5.6-2	2022-03-02
broom		0.7.12	2022-01-28
car		3.0-12	2021-11-06
carData		3.0-5	2022-01-06
dplyr		1.0.8	2022-02-08
effectsize		0.6.0.1	2022-01-26
FD		1.0-12	2014-08-19
forcats		0.5.1	2021-01-27
gbm		2.1.8	2020-07-15
geometry		0.4.5	2019-12-04
ggplot2		3.3.5	2021-06-25
kableExtra		1.3.4	2021-02-20
knitr		1.38	2022-03-25
lattice		0.20-45	2021-09-22
lme4		1.1-28	2022-02-05
lmerTest		3.1-3	2020-10-23
Matrix		1.4-1	2022-03-23
performance		0.9.0	2022-03-30
permute		0.9-7	2022-01-27
plyr		1.8.7	2022-03-24
purrr		0.3.4	2020-04-17
qqplotr		0.0.5	2021-04-23
readr		2.1.2	2022-01-30
snow		0.4-4	2021-10-27
stringr		1.4.0	2019-02-10
tibble		3.1.6	2021-11-07
tidyverse		1.2.0	2022-02-01
		1.3.1	2021-04-15
vegan		2.5-7	2020-11-28