

Key Data Visualizations for Extreme Flux Project

Code ▾

Loading in necessary packages.

Hide

```
library(tibble)
library(dplyr)
library(tidyr)
library(purrr)
library(ggplot2)
library(fastshap)
library(tidyverse)
library(ggbeeswarm)
library(cowplot)
library(EnvStats)
library(patchwork)
library(randomForestSRC)
library(gridExtra)
library(ggthemes)
library(cowplot)
library(devtools)
library(ggh4x)
library(grid)
library(mgcv)
library(latex2exp)
library(tidyverse)
library(broom)
library(maps)
library(reshape2)
```

```
G3;
Attaching package: 'reshape2'

g G3;The following object is masked from 'package:tidyr':

  smiths

g
```

Hide

```
source('./ggplot_theme_Publication-2.R')
print("All packages are loaded.")
```

```
[1] "All packages are loaded."
```

Loading in necessary data, including tower data and SHAP values.

[Hide](#)

```
# Loading in tower data
bindat_transition <- read.csv("./bindat_transition_west.csv")

# Loading in the random forest models
load("./Model_Results/transition_west_sink_results.R")
load("./Model_Results/transition_west_source_results.R")

# Loading in the SHAP analysis results performed on extreme sink classifications
sink_shap_P <- read.csv("./Model_Results/SHAP/sink_shap_Pwest.csv")
sink_shap_SW <- read.csv("./Model_Results/SHAP/sink_shap_SWwest.csv")
sink_shap_SWC <- read.csv("./Model_Results/SHAP/sink_shap_SWCwest.csv")
sink_shap_TA <- read.csv("./Model_Results/SHAP/sink_shap_TAwest.csv")
sink_shap_VPD <- read.csv("./Model_Results/SHAP/sink_shap_VPDwest.csv")

# Loading in the SHAP analysis results performed on extreme source classification
source_shap_TA <- read.csv("./Model_Results/SHAP/source_shap_TAwest.csv")
source_shap_P <- read.csv("./Model_Results/SHAP/source_shap_Pwest.csv")
source_shap_SW <- read.csv("./Model_Results/SHAP/source_shap_SWwest.csv")
source_shap_SWC <- read.csv("./Model_Results/SHAP/source_shap_SWCwest.csv")
source_shap_VPD <- read.csv("./Model_Results/SHAP/source_shap_VPDwest.csv")
print("All data is loaded.")
```

```
[1] "All data is loaded."
```

Calculating the interactions of variables in the random forest.

[Hide](#)

```
# Calculate variable interactions
interactions <- find.interaction(transition_sink_results, method="maxsubtree")
```

Method: maxsubtree

No. of variables: 45

Variables sorted by minimal depth?: TRUE

	SW	P_sd_30	P_30	P_365	SW_sd_30	SWC_30	TA	TA_30	SWC_7	SWC_365	SW_sd_365	P
_sd_365	TA_7	VPD_30	SW_7	lat	MAP							
SW	0.14	0.21	0.20	0.22	0.24	0.23	0.24	0.24	0.25	0.23	0.23	
0.24	0.25	0.24	0.25	0.33	0.38							
P_sd_30	0.23	0.15	0.26	0.24	0.26	0.24	0.25	0.24	0.28	0.25	0.25	
0.25	0.27	0.26	0.27	0.39	0.40							
P_30	0.24	0.25	0.17	0.25	0.27	0.25	0.25	0.24	0.27	0.24	0.26	
0.27	0.26	0.28	0.27	0.41	0.42							
P_365	0.25	0.26	0.25	0.18	0.26	0.26	0.27	0.26	0.28	0.27	0.27	
0.27	0.29	0.28	0.27	0.46	0.46							
SW_sd_30	0.25	0.28	0.26	0.26	0.18	0.28	0.29	0.29	0.29	0.28	0.28	
0.28	0.30	0.30	0.30	0.46	0.47							
SWC_30	0.27	0.29	0.28	0.27	0.29	0.20	0.31	0.29	0.31	0.28	0.29	
0.29	0.31	0.28	0.31	0.59	0.57							
TA	0.27	0.28	0.28	0.28	0.31	0.29	0.22	0.29	0.31	0.28	0.29	
0.27	0.32	0.30	0.30	0.53	0.59							
TA_30	0.30	0.27	0.28	0.30	0.29	0.29	0.29	0.22	0.30	0.29	0.29	
0.28	0.32	0.30	0.31	0.52	0.55							
SWC_7	0.32	0.30	0.31	0.31	0.31	0.32	0.30	0.31	0.22	0.28	0.32	
0.30	0.34	0.31	0.35	0.61	0.59							
SWC_365	0.27	0.28	0.27	0.27	0.29	0.28	0.28	0.27	0.28	0.23	0.28	
0.25	0.31	0.30	0.30	0.63	0.62							
SW_sd_365	0.29	0.30	0.29	0.31	0.30	0.31	0.29	0.29	0.31	0.28	0.23	
0.28	0.32	0.31	0.32	0.60	0.63							
P_sd_365	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.28	0.30	0.27	0.29	
0.23	0.32	0.30	0.31	0.59	0.60							
TA_7	0.31	0.30	0.31	0.32	0.32	0.32	0.31	0.33	0.32	0.32	0.33	
0.31	0.23	0.34	0.33	0.57	0.55							
VPD_30	0.32	0.30	0.31	0.32	0.32	0.32	0.31	0.33	0.33	0.32	0.33	
0.33	0.34	0.24	0.35	0.61	0.61							
SW_7	0.30	0.33	0.32	0.32	0.33	0.32	0.31	0.31	0.35	0.32	0.32	
0.31	0.32	0.34	0.24	0.58	0.59							
lat	0.37	0.39	0.37	0.39	0.38	0.36	0.38	0.38	0.39	0.35	0.39	
0.37	0.41	0.39	0.40	0.24	0.69							
MAP	0.38	0.38	0.39	0.37	0.40	0.39	0.40	0.39	0.38	0.38	0.39	
0.37	0.41	0.40	0.43	0.67	0.24							
TA_sd_365	0.29	0.28	0.30	0.29	0.29	0.29	0.31	0.30	0.30	0.28	0.29	
0.28	0.32	0.31	0.32	0.59	0.64							
SWC_1	0.36	0.35	0.36	0.34	0.37	0.37	0.35	0.37	0.37	0.34	0.34	
0.34	0.38	0.38	0.40	0.69	0.68							
SW_30	0.32	0.31	0.32	0.31	0.32	0.33	0.30	0.31	0.33	0.33	0.31	
0.31	0.34	0.33	0.35	0.57	0.59							
SWC_sd_30	0.33	0.34	0.33	0.35	0.35	0.33	0.35	0.34	0.35	0.32	0.33	
0.34	0.36	0.33	0.36	0.66	0.67							
VPD_365	0.32	0.31	0.31	0.30	0.32	0.33	0.34	0.32	0.33	0.29	0.31	
0.31	0.33	0.33	0.33	0.71	0.71							
TA_sd_365	SWC_1	SW_30	SWC_sd_30	VPD_365	TA_365	TA_sd_30	SW_365	TA_1	SW_1	VPD		

_sd_365	SWC_sd_7	VPD_7	SWC	SWC_sd_365								
SW		0.24	0.27	0.24	0.25	0.25	0.24	0.25	0.25	0.27	0.30	
0.25	0.27	0.26	0.27	0.25								
P_sd_30		0.27	0.29	0.27	0.28	0.26	0.26	0.27	0.27	0.29	0.29	
0.27	0.29	0.28	0.29	0.28								
P_30		0.26	0.30	0.26	0.29	0.27	0.27	0.27	0.26	0.29	0.30	
0.28	0.30	0.30	0.30	0.28								
P_365		0.26	0.30	0.28	0.26	0.28	0.27	0.29	0.27	0.32	0.30	
0.25	0.30	0.31	0.31	0.26								
SW_sd_30		0.28	0.31	0.29	0.30	0.30	0.29	0.29	0.29	0.33	0.32	
0.29	0.30	0.32	0.33	0.30								
SWC_30		0.27	0.33	0.29	0.30	0.30	0.28	0.31	0.29	0.34	0.33	
0.29	0.31	0.32	0.32	0.28								
TA		0.28	0.33	0.28	0.30	0.29	0.28	0.29	0.31	0.34	0.33	
0.30	0.30	0.32	0.32	0.29								
TA_30		0.29	0.33	0.31	0.30	0.29	0.28	0.31	0.30	0.33	0.36	
0.28	0.33	0.32	0.32	0.29								
SWC_7		0.30	0.37	0.33	0.33	0.31	0.32	0.33	0.31	0.37	0.37	
0.32	0.35	0.34	0.36	0.32								
SWC_365		0.28	0.32	0.29	0.29	0.28	0.26	0.29	0.28	0.33	0.34	
0.28	0.32	0.32	0.32	0.27								
SW_sd_365		0.28	0.35	0.30	0.30	0.30	0.30	0.32	0.30	0.33	0.34	
0.30	0.33	0.34	0.33	0.30								
P_sd_365		0.27	0.33	0.31	0.29	0.28	0.29	0.29	0.29	0.35	0.33	
0.29	0.31	0.34	0.34	0.29								
TA_7		0.32	0.35	0.33	0.33	0.34	0.31	0.33	0.32	0.36	0.37	
0.33	0.36	0.36	0.38	0.32								
VPD_30		0.31	0.36	0.33	0.31	0.33	0.32	0.32	0.33	0.36	0.40	
0.32	0.36	0.36	0.37	0.32								
SW_7		0.31	0.37	0.34	0.34	0.34	0.32	0.34	0.34	0.35	0.38	
0.31	0.34	0.37	0.35	0.33								
lat		0.37	0.44	0.39	0.39	0.39	0.38	0.40	0.36	0.44	0.43	
0.40	0.42	0.43	0.42	0.38								
MAP		0.38	0.44	0.41	0.40	0.40	0.39	0.41	0.39	0.46	0.46	
0.40	0.42	0.44	0.44	0.39								
TA_sd_365		0.25	0.33	0.31	0.30	0.31	0.29	0.31	0.29	0.34	0.36	
0.29	0.32	0.34	0.33	0.30								
SWC_1		0.37	0.25	0.37	0.36	0.36	0.36	0.38	0.35	0.41	0.41	
0.37	0.39	0.41	0.41	0.35								
SW_30		0.30	0.35	0.25	0.32	0.32	0.31	0.32	0.32	0.35	0.38	
0.32	0.35	0.35	0.36	0.31								
SWC_sd_30		0.33	0.38	0.35	0.25	0.34	0.34	0.36	0.34	0.39	0.39	
0.33	0.38	0.40	0.36	0.33								
VPD_365		0.28	0.36	0.33	0.34	0.25	0.31	0.34	0.31	0.37	0.39	
0.29	0.35	0.36	0.35	0.30								
	VPD	lon	MAT	VPD_sd_30	elev_meters	P_sd_7	P_7	TA_sd_7	P	VPD_sd_7	VPD_1	
SW_sd_7	P_1											
SW		0.26	0.38	0.37	0.27	0.38	0.30	0.30	0.30	0.37	0.30	0.30
0.31	0.45											
P_sd_30		0.30	0.43	0.44	0.29	0.43	0.34	0.35	0.31	0.46	0.32	0.33
0.31	0.49											
P_30		0.28	0.44	0.43	0.29	0.45	0.34	0.34	0.32	0.47	0.31	0.33

```

0.32 0.49
P_365      0.30 0.50 0.47      0.30      0.50      0.34 0.35      0.34 0.52      0.33 0.33
0.32 0.52
SW_sd_30   0.32 0.50 0.51      0.32      0.51      0.37 0.36      0.33 0.55      0.35 0.36
0.34 0.57
SWC_30     0.32 0.54 0.57      0.31      0.59      0.37 0.37      0.35 0.56      0.35 0.37
0.34 0.57
TA         0.32 0.56 0.59      0.31      0.59      0.34 0.36      0.34 0.49      0.33 0.35
0.34 0.52
TA_30      0.33 0.54 0.56      0.32      0.57      0.36 0.36      0.34 0.57      0.34 0.34
0.32 0.57
SWC_7      0.33 0.60 0.63      0.34      0.61      0.39 0.41      0.36 0.57      0.37 0.39
0.37 0.59
SWC_365    0.28 0.61 0.64      0.30      0.64      0.36 0.35      0.35 0.53      0.34 0.35
0.33 0.54
SW_sd_365  0.31 0.62 0.65      0.31      0.63      0.37 0.37      0.34 0.52      0.37 0.35
0.35 0.54
P_sd_365   0.30 0.62 0.61      0.32      0.61      0.37 0.37      0.36 0.56      0.34 0.35
0.34 0.56
TA_7       0.34 0.60 0.55      0.35      0.59      0.39 0.42      0.37 0.59      0.37 0.37
0.36 0.58
VPD_30     0.35 0.64 0.64      0.34      0.61      0.39 0.41      0.38 0.60      0.38 0.39
0.38 0.59
SW_7       0.34 0.58 0.59      0.36      0.59      0.40 0.40      0.35 0.59      0.39 0.38
0.37 0.60
lat        0.41 0.68 0.71      0.41      0.71      0.46 0.46      0.44 0.66      0.43 0.46
0.45 0.66
MAP        0.42 0.67 0.71      0.42      0.71      0.48 0.49      0.47 0.65      0.46 0.49
0.45 0.66
TA_sd_365  0.33 0.62 0.67      0.34      0.63      0.38 0.39      0.36 0.58      0.34 0.37
0.35 0.60
SWC_1      0.37 0.70 0.72      0.37      0.69      0.45 0.46      0.41 0.63      0.41 0.43
0.41 0.63
SW_30      0.35 0.59 0.61      0.34      0.60      0.39 0.41      0.36 0.63      0.37 0.38
0.38 0.60
SWC_sd_30  0.38 0.64 0.69      0.36      0.68      0.41 0.43      0.40 0.64      0.38 0.40
0.39 0.64
VPD_365    0.34 0.68 0.73      0.34      0.67      0.38 0.41      0.37 0.59      0.37 0.39
0.36 0.61
[ reached 'max' / getOption("max.print") -- omitted 23 rows ]

```

[Hide](#)

```
interactions_source <- find.interaction(transition_source_results, methods="maxsubtree")
```

Method: maxsubtree

No. of variables: 45

Variables sorted by minimal depth?: TRUE

			P_sd_30	P_30	TA	SWC_7	SWC_30	SWC_1	SWC	SW_sd_365	TA_365	lat	P_365	SWC_36
5	MAP	P_7	SWC_sd_7	TA_30	P_sd_7	VPD								
	P_sd_30		0.18	0.29	0.25	0.28	0.28	0.30	0.30	0.28	0.28	0.46	0.28	0.2
7	0.43	0.34	0.30	0.27	0.33	0.28								
	P_30		0.29	0.19	0.25	0.30	0.29	0.30	0.29	0.29	0.28	0.51	0.29	0.2
9	0.45	0.34	0.32	0.28	0.34	0.30								
	TA		0.25	0.24	0.19	0.24	0.25	0.26	0.25	0.25	0.24	0.48	0.23	0.2
2	0.47	0.29	0.28	0.22	0.30	0.28								
	SWC_7		0.31	0.31	0.25	0.20	0.32	0.34	0.34	0.32	0.31	0.57	0.30	0.3
0	0.55	0.38	0.34	0.29	0.36	0.31								
	SWC_30		0.31	0.32	0.26	0.33	0.21	0.35	0.32	0.31	0.31	0.61	0.31	0.3
1	0.58	0.40	0.34	0.30	0.40	0.32								
	SWC_1		0.32	0.32	0.27	0.34	0.32	0.21	0.35	0.32	0.33	0.60	0.31	0.3
2	0.56	0.42	0.34	0.32	0.40	0.31								
	SWC		0.31	0.32	0.27	0.33	0.32	0.33	0.21	0.32	0.33	0.61	0.32	0.3
0	0.58	0.38	0.34	0.30	0.39	0.32								
	SW_sd_365		0.32	0.32	0.27	0.33	0.31	0.33	0.35	0.22	0.32	0.64	0.31	0.2
8	0.64	0.37	0.32	0.31	0.37	0.32								
	TA_365		0.31	0.32	0.25	0.31	0.33	0.32	0.33	0.30	0.22	0.71	0.30	0.2
8	0.68	0.39	0.34	0.31	0.38	0.30								
	lat		0.42	0.43	0.38	0.42	0.43	0.46	0.44	0.44	0.41	0.22	0.41	0.3
9	0.69	0.49	0.46	0.42	0.49	0.42								
	P_365		0.28	0.29	0.26	0.32	0.30	0.30	0.33	0.29	0.29	0.57	0.22	0.2
8	0.55	0.36	0.32	0.29	0.36	0.30								
	SWC_365		0.30	0.30	0.25	0.31	0.31	0.32	0.32	0.30	0.27	0.69	0.29	0.2
3	0.70	0.38	0.30	0.30	0.37	0.28								
	MAP		0.35	0.37	0.32	0.38	0.39	0.41	0.37	0.38	0.39	0.69	0.34	0.3
7	0.23	0.44	0.39	0.38	0.45	0.39								
	P_7		0.36	0.36	0.34	0.40	0.37	0.41	0.40	0.39	0.38	0.61	0.37	0.3
8	0.60	0.24	0.41	0.37	0.45	0.38								
	SWC_sd_7		0.34	0.34	0.30	0.38	0.36	0.38	0.37	0.33	0.36	0.65	0.35	0.3
4	0.61	0.43	0.24	0.35	0.42	0.33								
	TA_30		0.28	0.29	0.25	0.31	0.32	0.34	0.32	0.30	0.31	0.57	0.28	0.2
8	0.56	0.35	0.31	0.24	0.35	0.32								
	P_sd_7		0.37	0.37	0.35	0.40	0.39	0.42	0.39	0.39	0.39	0.63	0.38	0.3
9	0.59	0.45	0.42	0.35	0.24	0.40								
	VPD		0.32	0.33	0.28	0.34	0.33	0.35	0.35	0.33	0.31	0.64	0.31	0.3
1	0.64	0.38	0.32	0.32	0.40	0.24								
	TA_1		0.32	0.35	0.27	0.35	0.35	0.35	0.33	0.32	0.33	0.64	0.32	0.3
3	0.64	0.41	0.36	0.32	0.42	0.32								
	VPD_sd_365		0.32	0.33	0.25	0.33	0.33	0.35	0.33	0.32	0.32	0.71	0.33	0.3
0	0.68	0.39	0.35	0.32	0.38	0.32								
	SWC_sd_30		0.34	0.37	0.30	0.37	0.35	0.38	0.38	0.34	0.36	0.68	0.35	0.3
4	0.65	0.42	0.37	0.34	0.41	0.34								
	SW_365		0.34	0.33	0.29	0.33	0.33	0.35	0.34	0.32	0.35	0.71	0.31	0.3
0	0.70	0.42	0.35	0.32	0.38	0.32								
	TA_1		VPD_sd_365	SWC_sd_30	SW_365	TA_7	VPD_365	VPD_30	lon	P_sd_365	MAT	SW_s		

d_30	SWC_sd_365	TA_sd_365	SW_7	SW_30							
P_sd_30	0.31	0.29	0.28	0.28	0.29	0.29	0.29	0.47	0.29	0.49	
0.29	0.30	0.28	0.29	0.29							
P_30	0.32	0.28	0.30	0.29	0.29	0.30	0.29	0.48	0.29	0.50	
0.30	0.30	0.28	0.31	0.30							
TA	0.30	0.25	0.27	0.25	0.27	0.24	0.24	0.47	0.24	0.49	
0.24	0.25	0.26	0.24	0.24							
SWC_7	0.34	0.31	0.31	0.30	0.33	0.32	0.30	0.52	0.29	0.57	
0.31	0.31	0.32	0.31	0.32							
SWC_30	0.33	0.31	0.33	0.31	0.34	0.33	0.34	0.59	0.32	0.60	
0.32	0.33	0.31	0.33	0.32							
SWC_1	0.35	0.33	0.32	0.32	0.34	0.33	0.33	0.56	0.31	0.57	
0.32	0.33	0.32	0.33	0.35							
SWC	0.33	0.32	0.32	0.32	0.35	0.32	0.32	0.58	0.31	0.61	
0.32	0.32	0.32	0.34	0.33							
SW_sd_365	0.33	0.31	0.33	0.30	0.34	0.32	0.32	0.63	0.30	0.65	
0.33	0.31	0.31	0.33	0.32							
TA_365	0.34	0.32	0.32	0.31	0.34	0.30	0.33	0.71	0.30	0.69	
0.32	0.31	0.32	0.33	0.32							
lat	0.45	0.39	0.45	0.42	0.46	0.42	0.43	0.70	0.42	0.69	
0.43	0.43	0.44	0.45	0.43							
P_365	0.32	0.28	0.31	0.30	0.30	0.30	0.29	0.59	0.29	0.57	
0.30	0.29	0.30	0.31	0.29							
SWC_365	0.31	0.29	0.31	0.29	0.34	0.29	0.31	0.69	0.29	0.71	
0.31	0.28	0.29	0.30	0.30							
MAP	0.42	0.38	0.39	0.39	0.38	0.38	0.38	0.67	0.34	0.68	
0.38	0.37	0.38	0.39	0.38							
P_7	0.39	0.38	0.39	0.38	0.39	0.39	0.39	0.62	0.37	0.61	
0.39	0.39	0.39	0.40	0.41							
SWC_sd_7	0.39	0.36	0.36	0.36	0.38	0.36	0.36	0.63	0.33	0.66	
0.36	0.34	0.36	0.37	0.37							
TA_30	0.32	0.30	0.30	0.31	0.33	0.32	0.32	0.57	0.29	0.56	
0.30	0.29	0.31	0.32	0.31							
P_sd_7	0.41	0.41	0.39	0.39	0.40	0.38	0.37	0.63	0.37	0.64	
0.40	0.39	0.39	0.41	0.40							
VPD	0.34	0.32	0.35	0.32	0.35	0.32	0.33	0.62	0.31	0.65	
0.34	0.34	0.33	0.31	0.32							
TA_1	0.24	0.34	0.35	0.33	0.36	0.34	0.34	0.64	0.33	0.65	
0.34	0.34	0.34	0.33	0.34							
VPD_sd_365	0.34	0.25	0.33	0.33	0.35	0.33	0.35	0.69	0.30	0.69	
0.33	0.32	0.30	0.35	0.32							
SWC_sd_30	0.37	0.35	0.25	0.36	0.39	0.38	0.35	0.68	0.34	0.67	
0.35	0.35	0.34	0.38	0.36							
SW_365	0.33	0.33	0.33	0.25	0.36	0.33	0.35	0.71	0.31	0.70	
0.33	0.33	0.33	0.34	0.32							
elev_meters	TA_sd_30	VPD_7	VPD_sd_30	VPD_1	SW_sd_7	SW	SW_1	TA_sd_7	VPD_sd_7	P	P_1
7	P	P_1									
P_sd_30	0.48	0.28	0.32	0.30	0.33	0.33	0.34	0.32	0.33	0.3	
5	0.49	0.52									
P_30	0.50	0.29	0.33	0.31	0.36	0.34	0.34	0.34	0.33	0.3	
5	0.49	0.53									
TA	0.48	0.26	0.28	0.26	0.32	0.27	0.30	0.31	0.31	0.3	

```

0 0.43 0.47
SWC_7          0.56      0.33 0.34      0.34 0.36      0.35 0.37 0.36      0.38      0.3
9 0.52 0.54
SWC_30         0.61      0.33 0.36      0.34 0.37      0.37 0.37 0.37      0.39      0.4
0 0.51 0.58
SWC_1          0.60      0.35 0.38      0.35 0.39      0.37 0.37 0.39      0.38      0.4
0 0.51 0.58
SWC           0.59      0.33 0.36      0.33 0.38      0.36 0.36 0.37      0.36      0.4
0 0.54 0.59
SW_sd_365      0.67      0.32 0.37      0.33 0.39      0.34 0.37 0.38      0.38      0.3
7 0.53 0.60
TA_365         0.70      0.33 0.37      0.33 0.35      0.34 0.36 0.37      0.37      0.3
8 0.55 0.61
lat            0.70      0.46 0.48      0.46 0.49      0.46 0.49 0.49      0.49      0.5
0 0.62 0.66
P_365          0.58      0.31 0.33      0.31 0.35      0.34 0.34 0.35      0.35      0.3
4 0.52 0.56
SWC_365        0.71      0.32 0.35      0.32 0.33      0.34 0.35 0.37      0.36      0.3
7 0.52 0.59
MAP            0.68      0.39 0.42      0.39 0.44      0.43 0.42 0.46      0.44      0.4
4 0.60 0.62
P_7            0.61      0.38 0.43      0.40 0.42      0.42 0.45 0.45      0.44      0.4
5 0.58 0.61
SWC_sd_7       0.64      0.36 0.40      0.35 0.42      0.39 0.39 0.41      0.40      0.4
3 0.56 0.62
TA_30          0.56      0.32 0.35      0.33 0.37      0.34 0.35 0.36      0.36      0.3
6 0.52 0.58
P_sd_7         0.65      0.39 0.43      0.39 0.45      0.43 0.46 0.45      0.43      0.4
3 0.59 0.58
VPD            0.68      0.34 0.37      0.35 0.38      0.36 0.37 0.38      0.38      0.3
9 0.55 0.62
TA_1           0.63      0.36 0.40      0.35 0.39      0.38 0.41 0.41      0.38      0.4
0 0.54 0.59
VPD_sd_365     0.70      0.35 0.37      0.34 0.36      0.35 0.40 0.39      0.38      0.3
8 0.58 0.59
SWC_sd_30      0.70      0.35 0.40      0.38 0.40      0.38 0.43 0.42      0.41      0.4
2 0.59 0.61
SW_365         0.74      0.33 0.39      0.35 0.39      0.36 0.38 0.40      0.40      0.4
2 0.57 0.64
[ reached 'max' /getOption("max.print") -- omitted 23 rows ]

```

Figure 1. Heatmaps showing variable importance (VIMP; diagonal) and maximum subtree interaction strengths (off-diagonal) for predictors in the (A) extreme sink and (B) extreme source random forest classifiers. Displayed variables are those ranked among the top 15 by either variable importance (VIMP) or average interaction score. Variables are ordered from left to right by decreasing VIMP; see Table 2 in ReadMe for a description of the variable names.

[Hide](#)


```

# Extract the interaction matrix
interaction_matrix <- as.matrix(interactions)
interaction_matrix_source <- as.matrix(interactions_source)

# If it's a matrix, convert it to long format for ggplot2
interaction_long <- melt(interaction_matrix)
interaction_long_source <- melt(interaction_matrix_source)

# Adding the VIMP onto the diagonals
# Extract interaction matrix and VIMP
# Select variables with top 15 VIMP
vimp_vals <- transition_sink_results$importance[,3]
vimp_vals_df <- data.frame(vimp_vals)
vimp_vals_df$var <- row.names(vimp_vals_df)
top_vimp_sink <- vimp_vals_df[order(vimp_vals_df$vimp_vals,decreasing=TRUE),][1:15,2]

vimp_vals_source <- transition_source_results$importance[,3]
vimp_vals_source_df <- data.frame(vimp_vals_source)
vimp_vals_source_df$var <- row.names(vimp_vals_source_df)
top_vimp_source <- vimp_vals_source_df[order(vimp_vals_source_df$vimp_vals_source,decreasing=TRUE),][1:15,2]

# Select variables with top 15 mean interaction values
mean_interaction_sink <- interaction_long %>%
  group_by(Var1) %>%
  summarize(mean_interaction = mean(value))
mean_interaction_sink <- data.frame(mean_interaction_sink)
top_interaction_sink <- mean_interaction_sink[order(mean_interaction_sink$mean_interaction,decreasing=TRUE),][1:15,1]
top_interaction_sink <- as.character(top_interaction_sink)

mean_interaction_source <- interaction_long_source %>%
  group_by(Var1) %>%
  summarize(mean_interaction = mean(value))
mean_interaction_source <- data.frame(mean_interaction_source)
top_interaction_source <- mean_interaction_source[order(mean_interaction_source$mean_interaction,decreasing=TRUE),][1:17,1]
top_interaction_source <- as.character(top_interaction_source)

# Combining the top VIMP and interaction variables
sink_var <- unique(c(top_vimp_sink,top_interaction_sink))
source_var <- unique(c(top_vimp_source,top_interaction_source))

# Build two datasets: one for off-diagonal, one for diagonal with VIMP
interaction_offdiag <- interaction_long %>%
  filter((Var1 != Var2) & (Var1 %in% sink_var) & (Var2 %in% sink_var))

interaction_offdiag$Var1 <- as.character(interaction_offdiag$Var1)
interaction_offdiag$Var2 <- as.character(interaction_offdiag$Var2)

# Changing the factor names to the new variable labels
interaction_offdiag$Var1 <- as.factor(interaction_offdiag$Var1)

```

```

interaction_offdiag$Var2 <- as.factor(interaction_offdiag$Var2)
new_levels <- c("elev","lat","lon","MAP","MAT","P","P.D",
               "P.M","P.Y","P.W","σ(P.M)","σ(P.Y)","σ(P.W)",
               "SW","SW.D","SW.W","σ(SW.M)","σ(SW.Y)","σ(SW.W)",
               "SWC.M","SWC.Y","σ(SWC.W)","TA.M","σ(TA.Y)",
               "σ(TA.W)","VPD.D","VPD.W","σ(VPD.Y)","σ(VPD.W)")
levels(interaction_offdiag$Var1) <- new_levels
levels(interaction_offdiag$Var2) <- new_levels

# Repeating for extreme sources
interaction_offdiag_source <- interaction_long_source %>%
  filter((Var1 != Var2) & (Var1 %in% source_var) & (Var2 %in% source_var))

interaction_offdiag_source$Var1 <- as.character(interaction_offdiag_source$Var1)
interaction_offdiag_source$Var2 <- as.character(interaction_offdiag_source$Var2)

interaction_offdiag_source$Var1 <- as.factor(interaction_offdiag_source$Var1)
interaction_offdiag_source$Var2 <- as.factor(interaction_offdiag_source$Var2)

new_levels_source <- c("elev","lat","lon","MAP","MAT","P","P.D",
                      "P.M","P.Y","P.W","σ(P.M)","σ(P.Y)","σ(P.W)",
                      "SW","SW.D","σ(SW.Y)","σ(SW.W)","SWC.D","SWC.M","SWC.Y","SWC.
W","TA","TA.Y",
                      "σ(TA.W)","VPD.D","VPD.W","σ(VPD.M)","σ(VPD.Y)",
                      "σ(VPD.W)")
levels(interaction_offdiag_source$Var1) <- new_levels_source
levels(interaction_offdiag_source$Var2) <- new_levels_source

interaction_diag <- data.frame(
  Var1 = names(vimp_vals),
  Var2 = names(vimp_vals),
  value = vimp_vals
)

interaction_diag_source <- data.frame(
  Var1 = names(vimp_vals_source),
  Var2 = names(vimp_vals_source),
  value = vimp_vals_source
)

# Filter for variable pairs in top vimp
interaction_diag <- interaction_diag %>%
  filter((Var1 %in% sink_var) & (Var2 %in% sink_var))

interaction_diag <- interaction_diag[order(interaction_diag$Var1),]
new_levels_diag <- c("elev","lat","lon","MAP","MAT","P","P.D",
                    "P.M","P.Y","P.W","σ(P.M)","σ(P.Y)","σ(P.W)",
                    "SW","SW.D","SW.W","σ(SW.M)","σ(SW.Y)","σ(SW.W)",
                    "SWC.M","SWC.Y","σ(SWC.W)", "TA.M","σ(TA.Y)","σ(TA.W)",
                    "VPD.D","VPD.W","σ(VPD.Y)","σ(VPD.W)")

interaction_diag$Var1 <- new_levels_diag

```

```

interaction_diag$Var2 <- new_levels_diag

interaction_diag_source <- interaction_diag_source %>%
  filter((Var1 %in% source_var) & (Var2 %in% source_var))

interaction_diag_source <- interaction_diag_source[order(interaction_diag_source$Var1),]
new_levels_source_diag <- c("elev","lat","lon","MAP","MAT","P","P.D",
                           "P.M","P.Y","P.W"," $\sigma$ (P.M)"," $\sigma$ (P.Y)"," $\sigma$ (P.W)",
                           "SW","SW.D"," $\sigma$ (SW.Y)"," $\sigma$ (SW.W)","SWC.D","SWC.M","SWC.Y","SW
C.W",
                           "TA","TA.Y"," $\sigma$ (TA.W)","VPD.D","VPD.W"," $\sigma$ (VPD.M)"," $\sigma$ (VPD.Y)",
                           " $\sigma$ (VPD.W)")

interaction_diag_source$Var1 <- new_levels_source_diag
interaction_diag_source$Var2 <- new_levels_source_diag

# Order by highest VIMP
interaction_diag <- interaction_diag[order(interaction_diag$value,decreasing=TRUE),]
interaction_diag_source <- interaction_diag_source[order(interaction_diag_source$value,
decreasing=TRUE),]

# Setting factor levels so they are the same between offdiag and diag
var_order <- unique(c(as.character(interaction_diag$Var1), as.character(interaction_diag
$Var2)))
var_order_source <- unique(c(as.character(interaction_diag_source$Var1), as.character(in
teraction_diag_source$Var2)))

interaction_offdiag$Var1 <- factor(interaction_offdiag$Var1, levels = var_order)
interaction_offdiag$Var2 <- factor(interaction_offdiag$Var2, levels = var_order)
interaction_diag$Var1 <- factor(interaction_diag$Var1, levels = var_order)
interaction_diag$Var2 <- factor(interaction_diag$Var2, levels = var_order)

interaction_offdiag_source$Var1 <- factor(interaction_offdiag_source$Var1, levels = var_
order_source)
interaction_offdiag_source$Var2 <- factor(interaction_offdiag_source$Var2, levels = var_
order_source)
interaction_diag_source$Var1 <- factor(interaction_diag_source$Var1, levels = var_order_
source)
interaction_diag_source$Var2 <- factor(interaction_diag_source$Var2, levels = var_order_
source)

# Now plot both, with separate fill scales #####
# Create diagonal labels
diag_labels <- interaction_diag %>%
  dplyr::mutate(
    x = as.numeric(Var2),
    y = as.numeric(Var1),
    label = as.character(Var1)
  )

# Plot the heatmap for extreme sink variables VIMP and Interactions

```

```

p1_main <- ggplot() +
  # Diagonal VIMP tiles
  geom_tile(
    data = interaction_diag,
    aes(x = Var2, y = Var1, fill = value),
    color="grey"
  ) +
  scale_fill_gradientn(
    colors = c("white", "#da1b28"),
    name = "VIMP",
    guide = "none",
    limits = c(0,0.1)
  ) +

  ggnewscale::new_scale_fill() +

  # Off-diagonal interactions
  geom_tile(
    data = interaction_offdiag %>% dplyr::filter(as.numeric(Var1) < as.numeric(Var2)),
    aes(x = Var2, y = Var1, fill = value),
    color="grey"
  ) +
  scale_fill_gradientn(
    colors = c("white", "#0f2f96"),
    values = scales::rescale(c(0.2, 1)),
    limits = c(0.2, 1),
    oob = scales::oob_squish,
    name = "Interaction\nStrength",
    guide = "none"
  ) +

  # Labels and styling
  labs(x = NULL, y = NULL) +
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
    panel.grid = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks = element_blank(),
    axis.text.x = element_text(angle = 50, hjust = 1, color = "black", size = 7),
    plot.margin = unit(c(10, 10, 10, 50), "pt")
  ) +
  coord_fixed(clip = "off") +
  geom_text(
    data = diag_labels,
    aes(x = x-1, y = y, label = label),
    hjust = 1, size = 3, color = "black"
  ) +
  annotate("text", x = 5, y = 25, label = "(A) Extreme Sink", size = 7, color = "black")

# Create custom legends with different colors
vimp_legend <- ggplot() +

```

```

geom_tile(aes(x = 1, y = seq(0, 1, length.out = 100), fill = seq(0, 1, length.out = 100))) +
scale_fill_gradientn(
  colors = c("white", "#da1b28"),
  name = "VIMP",
  guide = guide_colorbar(
    barheight = 10,
    title.position = "top"
  ),
  limits = c(0, 0.1)
) +
theme_void() +
theme(legend.position = "right",
      legend.text = element_text(size = 10),
      legend.title = element_text(size = 10))

interaction_legend <- ggplot() +
  geom_tile(aes(x = 1, y = seq(0.2, 1, length.out = 100), fill = seq(0.2, 1, length.out = 100))) +
  scale_fill_gradientn(
    colors = c("white", "#0f2f96"),
    name = "Interaction\nStrength",
    limits = c(0.2, 1),
    guide = guide_colorbar(
      barheight = 10,
      title.position = "top"
    )
  )

# Extract legends
vimp_leg <- cowplot::get_legend(vimp_legend)

```

G2; H2; Warning h in get_plot_component(plot, "guide-box") :
 Multiple components found; returning the first one. To return all, use `return_all = TRUE`. g

[Hide](#)

```
interaction_leg <- cowplot::get_legend(interaction_legend)
```

G2; H2; Warning h in get_plot_component(plot, "guide-box") :
 Multiple components found; returning the first one. To return all, use `return_all = TRUE`. g

[Hide](#)

```

# Combine plot with custom legends
p1 <- cowplot::plot_grid(
  p1_main,
  cowplot::plot_grid(vimp_leg, interaction_leg, ncol = 1,
                      rel_heights = c(1, 1),
                      align = "v",
                      vjust = 1),
  ncol = 2,
  rel_widths = c(3, .5)
)

##### Plotting for source model now #####
# Create diagonal labels
diag_labels_source <- interaction_diag_source %>%
  dplyr::mutate(
    x = as.numeric(Var2),
    y = as.numeric(Var1),
    label = as.character(Var1)
  )

##### Plotting for source model now #####
# Create diagonal labels
diag_labels_source <- interaction_diag_source %>%
  dplyr::mutate(
    x = as.numeric(Var2),
    y = as.numeric(Var1),
    label = as.character(Var1)
  )

# Plot
# Create the main plot without legends
p2_main <- ggplot() +
  # Diagonal VIMP tiles
  geom_tile(
    data = interaction_diag_source,
    aes(x = Var2, y = Var1, fill = value),
    color="grey"
  ) +
  scale_fill_gradientn(
    colors = c("white", "#da1b28"), # Plot uses red gradient
    name = "VIMP",
    guide = "none" # Hide this legend
  ) +

  ggnewscale::new_scale_fill() +

  # Off-diagonal interactions
  geom_tile(
    data = interaction_offdiag_source %>% dplyr::filter(as.numeric(Var1) < as.numeric(Var2)),
    aes(x = Var2, y = Var1, fill = value),

```

```

    color="grey"
  ) +
  scale_fill_gradientn(
    colors = c("white", "#0f2f96"), # Plot uses blue gradient
    values = scales::rescale(c(0.2, 1)),
    limits = c(0.2, 1),
    oob = scales::oob_squish,
    name = "Interaction\nStrength",
    guide = "none" # Hide this legend too
  ) +

# Labels and styling
labs(x = NULL, y = NULL) +
theme_minimal(base_size = 12) +
theme(
  plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
  panel.grid = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks = element_blank(),
  axis.text.x = element_text(angle = 50, hjust = 1, color = "black", size = 7),
  plot.margin = unit(c(10, 10, 10, 50), "pt")
) +
coord_fixed(clip = "off") +
geom_text(
  data = diag_labels_source,
  aes(x = x-1, y = y, label = label),
  hjust = 1, size = 3, color = "black"
) +
annotate("text", x = 5, y = 25, label = "(B) Extreme Source", size = 7, color = "black")

# Create custom legends with different colors
vimp_legend <- ggplot() +
  geom_tile(aes(x = 1, y = seq(0, 1, length.out = 100), fill = seq(0, 1, length.out = 100))) +
  scale_fill_gradientn(
    colors = c("white", "white"), # Different colors for legend
    name = ""
  ) +
  theme_void() +
  theme(legend.position = "right",
    legend.text = element_blank())

interaction_legend <- ggplot() +
  geom_tile(aes(x = 1, y = seq(0.2, 1, length.out = 100), fill = seq(0.2, 1, length.out = 100))) +
  scale_fill_gradientn(
    colors = c("white", "white"), # Different colors for legend
    name = "",
    limits = c(0.2, 1)
  ) +
  theme_void() +

```

```
theme(legend.position = "right",  
      legend.text = element_blank())
```

```
# Extract legends  
vimp_leg <- cowplot::get_legend(vimp_legend)
```

```
G2; H2; Warning h in get_plot_component(plot, "guide-box") :  
Multiple components found; returning the first one. To return all, use `return_all = TRUE`. g
```

[Hide](#)

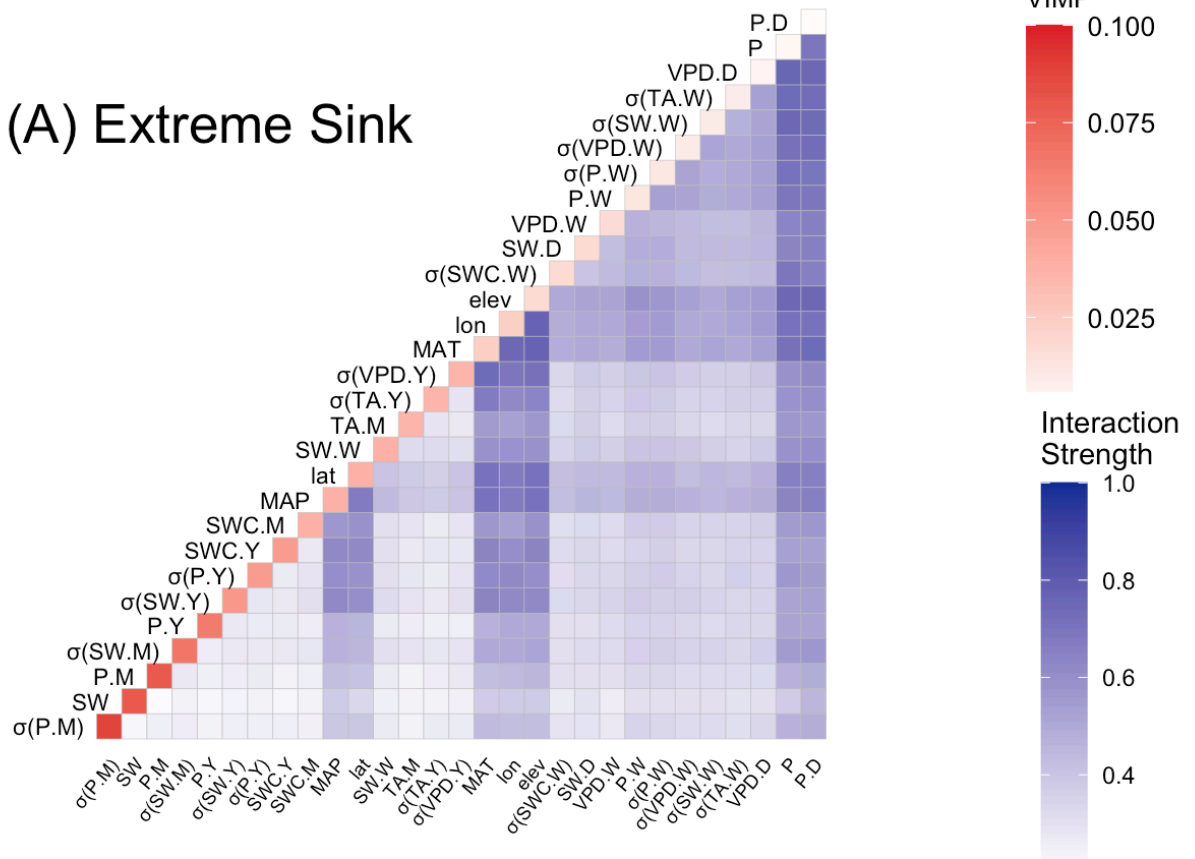
```
interaction_leg <- cowplot::get_legend(interaction_legend)
```

```
G2; H2; Warning h in get_plot_component(plot, "guide-box") :  
Multiple components found; returning the first one. To return all, use `return_all = TRUE`. g
```

[Hide](#)

```
# Combine plot with custom legends  
p2 <- cowplot::plot_grid(  
  p2_main,  
  cowplot::plot_grid(vimp_leg, interaction_leg, ncol = 1),  
  ncol = 2,  
  rel_widths = c(3, .5)  
)  
  
p1
```


(A) Extreme Sink



Hide

p2

(B) Extreme Source

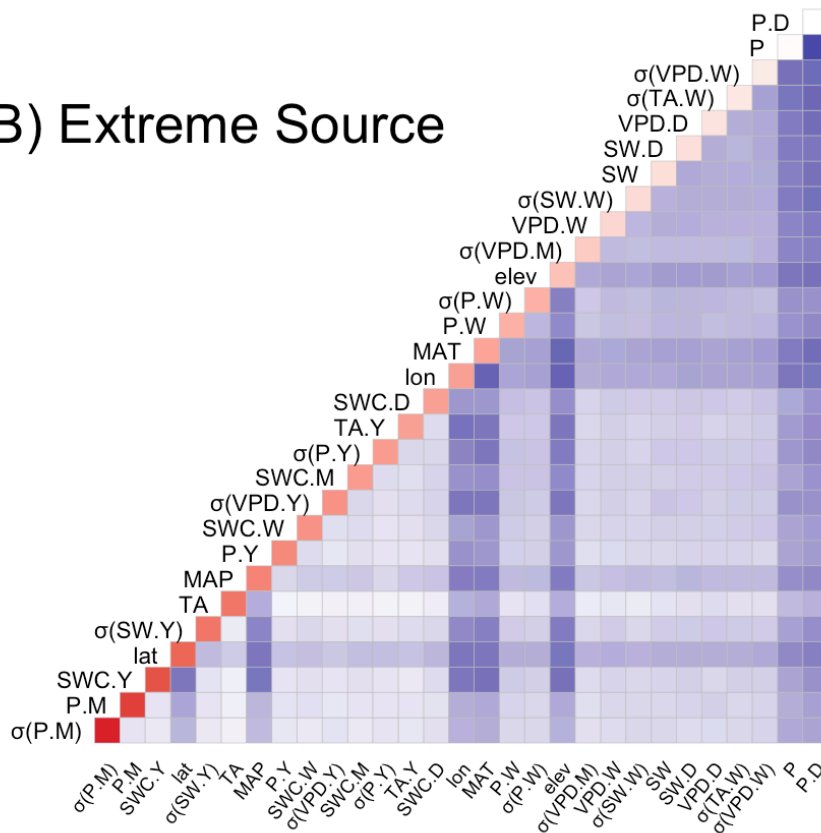


Figure 2. SHAP values for key temperature predictors of (A-B) extreme sinks and (C-D) extreme source classifications, selected based on top variable importance (VIMP; see Fig. 2), including (A, C) concurrent temperature, TA and (B, D) temperature over the past month, TA.M. Seasonality is represented by the color of each regression line, with green for spring months (March-May), yellow for summer months (June-August), orange for fall months (September-November), and blue for winter months (December-February). Quadratic linear regressions are fit within each MAT and season category, and R^2 values are shown for models with statistically significant F-statistics ($p < 0.05$). Linestyles are designated by mean annual temperature (MAT): dotted for $\text{MAT} < 6^\circ\text{C}$ (low), dashed for $6^\circ\text{C} \leq \text{MAT} < 14^\circ\text{C}$ (medium), and solid for $\text{MAT} > 14^\circ\text{C}$ (high).

[Hide](#)

```

# First we categorize the mean annual temperature (MAT) of each SHAP value as low, medium, or high
sink_shap_TA <- sink_shap_TA %>% mutate(
  MAT_cat = case_when(
    MAT < 6 ~ "Low",
    (MAT >= 6) & (MAT < 14) ~ "Medium",
    MAT > 14 ~ "High"
  )
)

source_shap_TA <- source_shap_TA %>% mutate(
  MAT_cat = case_when(
    MAT < 6 ~ "Low",
    (MAT >= 6) & (MAT < 14) ~ "Medium",
    MAT >= 14 ~ "High"
  )
)

sink_shap_TA$MAT_cat <- factor(sink_shap_TA$MAT_cat, levels = c("Low","Medium","High"))
source_shap_TA$MAT_cat <- factor(source_shap_TA$MAT_cat, levels = c("Low","Medium","High"))

# Merge SHAP values with the corresponding date of their observation
date_shapTA <- left_join(sink_shap_TA, bindat_transition, by=colnames(sink_shap_TA)[9:48])
date_shapTA_source <- left_join(source_shap_TA, bindat_transition, by=colnames(source_shap_TA)[9:48])

# Categorize the season that each SHAP value belongs using the dates
date_shapTA <- date_shapTA %>%
  mutate(season = case_when(
    Month %in% c(12,1,2) ~ "Winter",
    Month %in% c(3,4,5) ~ "Spring",
    Month %in% c(6,7,8) ~ "Summer",
    Month %in% c(9,10,11) ~ "Fall",
    TRUE ~ "Error"
  ))

date_shapTA_source <- date_shapTA_source %>%
  mutate(season = case_when(
    Month %in% c(12,1,2) ~ "Winter",
    Month %in% c(3,4,5) ~ "Spring",
    Month %in% c(6,7,8) ~ "Summer",
    Month %in% c(9,10,11) ~ "Fall",
    TRUE ~ "Error"
  ))

# Assign a color for each season
season_colors <- c(
  "Spring" = "#77DD77",
  "Summer" = "#FFD700",
  "Fall" = "#FF8C00",

```

```

"Winter" = "#4682B4"
)

# Set x and y limits for both sink and source plots
# This aligns the x and y axis for cross subplot comparisons in the figure
x_limits <- list(TA = c(min(min(sink_shap_TA$TA), min(source_shap_TA$TA)),
                        max(max(sink_shap_TA$TA), max(source_shap_TA$TA))),
                 TA_30 = c(min(min(sink_shap_TA$TA_30), min(source_shap_TA$TA_30)),
                           max(max(sink_shap_TA$TA_30), max(source_shap_TA$TA_30)))
)

y_limits <- list(TA = c(min(min(sink_shap_TA$shap[sink_shap_TA$var == "TA"]),
                        min(source_shap_TA$shap[source_shap_TA$var == "TA"])),
                 max(max(sink_shap_TA$shap[sink_shap_TA$var == "TA"]),
                     max(source_shap_TA$shap[source_shap_TA$var == "TA"]))),
                 TA_30 = c(min(min(sink_shap_TA$shap[sink_shap_TA$var == "TA_30"]),
                        min(source_shap_TA$shap[source_shap_TA$var == "TA_30"])),
                           max(max(sink_shap_TA$shap[sink_shap_TA$var == "TA_30"]),
                               max(source_shap_TA$shap[source_shap_TA$var == "TA_30"])))
)

# Plotting the SHAP values for concurrent and preceding month average temperature
# on extreme sinks. Applying quadratic regressions to each combination of MAT categories
# and season.
p1 <- date_shapTA %>%
  filter(var %in% c("TA", "TA_30")) %>%
  mutate(
    var_factor = factor(var, levels = c("TA", "TA_30"), labels = c("TA", "TA.M")),
    season = factor(season, levels = c("Spring", "Summer", "Fall", "Winter"))
  ) %>%
  ggplot(aes(x = value, y = shap, linetype = MAT_cat)) +
  geom_hline(yintercept = 0) +
  geom_point(color = "grey60", alpha = .2, size = 1) +
  geom_smooth(aes(color = season), method = "lm", formula = y ~ x + I(x^2), se = FALSE)
+
  scale_color_manual(values = season_colors) +
  scale_linetype_manual(values = c("Low" = "dotted", "Medium" = "dashed", "High" = "solid")) +
  labs(x = "Temperature (°C)", y = NULL, color = "Season", linetype = "MAT") +
  facet_wrap(~var_factor, nrow = 1, scales = "free") +
  faceted_pos_scales(
    x = list(scale_x_continuous(limits = x_limits$TA),
             scale_x_continuous(limits = x_limits$TA_30)),
    y = list(scale_y_continuous(limits = y_limits$TA),
             scale_y_continuous(limits = y_limits$TA_30))
  ) +
  theme_bw(base_size = 12) +
  theme(
    axis.text = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    axis.title.x = element_blank(),
    axis.text.x = element_text(size = 10),

```

```

strip.text = element_text(size = 10, face = "bold"),
legend.key = element_rect(color = "white", fill = "white"),
legend.text = element_text(size = 10),
legend.title = element_text(size = 10),
legend.position = "none",
plot.margin = margin(6, 20, 6, 20)
) +
coord_cartesian(clip = "off")

# Adding (A) and (B) labels
label_df <- tibble::tibble(
  var_factor = factor(c("TA", "TA.M"), levels = c("TA", "TA.M")),
  x = c(-Inf, -Inf),
  y = c( Inf,  Inf),
  lab = c("(A)", "(B)")
)

# Add the labels to the plot.
p1 <- p1 +
  geom_text(
    data = label_df,
    mapping = aes(x = x, y = y, label = lab),
    inherit.aes = FALSE,
    hjust = -0.1,
    vjust = 1.2,
    size = 5,
    fontface = "bold",
    color = "black"
  )

# Creating the same plot as above, but for extreme sources and adding a legend.
p2 <- date_shapTA_source %>%
  filter(var %in% c("TA","TA_30")) %>%
  mutate( var_factor = factor(var, levels = c("TA","TA_30"),
                                labels = c("TA", "TA.M")),
          season = factor(season, levels = c("Spring", "Summer", "Fall", "Winter")) ) %
>%
  ggplot(aes(x = value, y = shap, linetype = MAT_cat)) +
  geom_hline(yintercept = 0) + # Points stay grey
  geom_point(color = "grey60", alpha = .2, size = 1) +
  geom_smooth( aes(color = season), method = "lm", formula = y ~ x + I(x^2), se = FALSE
  ) +
  scale_color_manual(values = season_colors) +
  scale_linetype_manual(values = c( "Low" = "dotted", "Medium" = "dashed", "High" = "solid"
  id" )) +
  facet_wrap(~var_factor, nrow = 1, scales = "free") +
  labs(x = "Temperature (°C)", y = NULL, color = "Season", linetype = "MAT") +
  facetted_pos_scales( x = list( scale_x_continuous(limits = x_limits$TA),
                                scale_x_continuous(limits = x_limits$TA_30)),
                      y = list( scale_y_continuous(limits = y_limits$TA),
                                scale_y_continuous(limits = y_limits$TA_30))) +
  theme_bw(base_size = 12) +

```

```

theme(
  panel.grid.minor = element_blank(),
  legend.key.width = unit(24, "pt"),
  axis.text = element_text(size = 10),
  axis.title.y = element_text(size = 10),
  axis.title.x = element_text(size = 10),
  axis.text.x = element_text(size = 10),
  strip.text = element_text(size = 10, face = "bold"),
  legend.key = element_rect(color = "white", fill = "white"),
  legend.text = element_text(size = 10),
  legend.title = element_text(size = 10)
) +
guides(
  linetype = guide_legend(
    title = "MAT",
    override.aes = list(color = "black", size = 1.5)
  ),
  color = guide_legend(
    title = "Season",
    override.aes = list(size = 1.5)
  )
)

# Create the (C) and (D) labels.
label_df_p2 <- tibble(
  var_factor = factor(c("TA", "TA.M"), levels = c("TA", "TA.M")),
  x = c(-Inf, -Inf),
  y = c( Inf,  Inf),
  lab = c("(C)", "(D)")
)

# Add the labels to the second plot.
p2 <- p2 +
  coord_cartesian(clip = "off") +
  geom_text(
    data = label_df_p2,
    aes(x = x, y = y, label = lab),
    inherit.aes = FALSE,
    hjust = -0.1,
    vjust = 1.2,
    size = 5,
    fontface = "bold",
    color = "black"
  )

# Stack the two plots.
p1 / p2

```

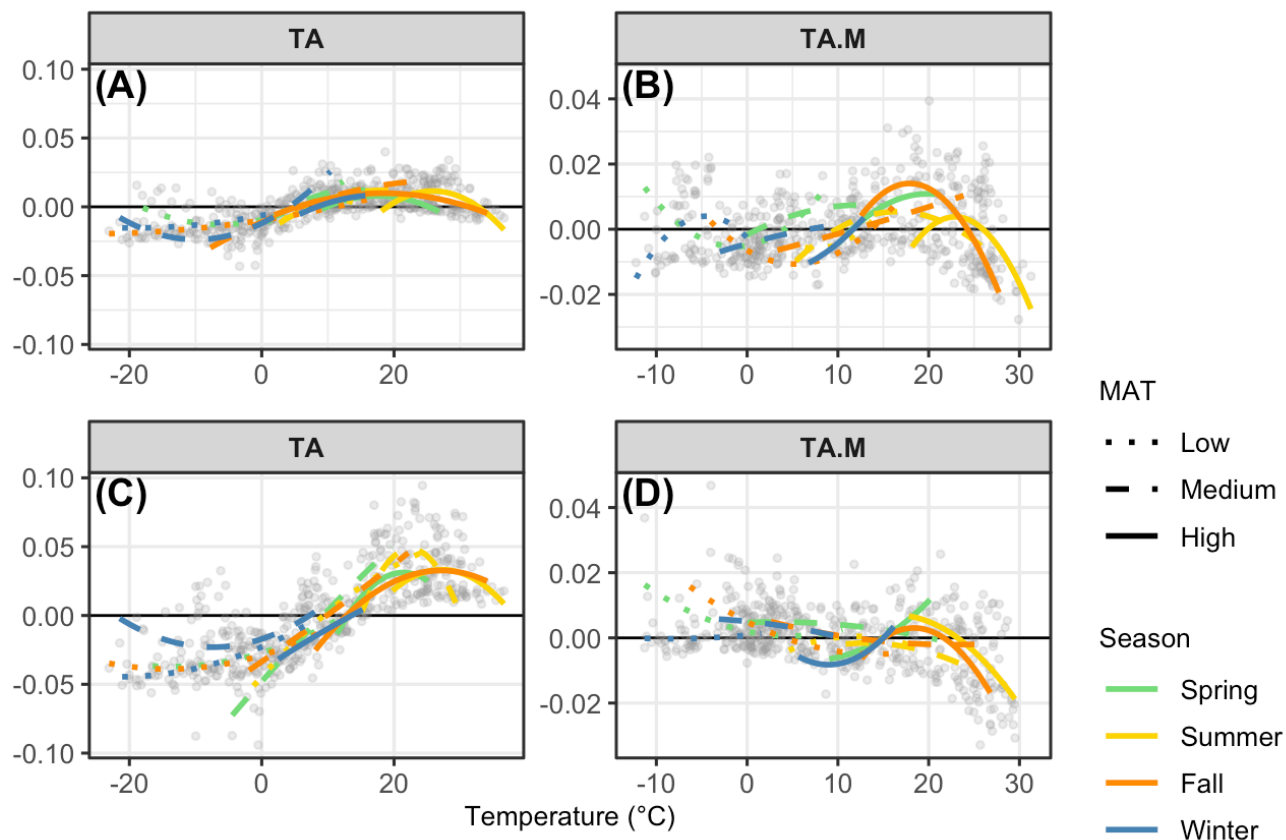


Figure 3. SHAP values for key precipitation predictors in the (A-D) extreme sink and (E-H) extreme source random forest classifications, selected based on top variable importance (VIMP; see Fig. 2), including precipitation over (A, E) the past month, P.M, and (C, G) the past year, P.Y, and variation in daily precipitation over (B, F) the past month, Δ (P.M) and (D, H) the past year, Δ (P.Y). Points are colored by mean annual precipitation (MAP): red for MAP < 40 cm (low), orange for 40 cm \leq MAP < 50 cm (medium), and blue for MAP > 50 cm (high). Linear regressions are fit within each MAP category, and R^2 values are shown for models with statistically significant slopes ($p < 0.05$).

[Hide](#)

```

# Categorizing each SHAP value by its associated mean annual precipitation for
# both sink and source RF
source_shap_P <- source_shap_P %>% mutate(
  MAP_cat = case_when(
    MAP < 40 ~ "Low",
    (MAP >= 40) & (MAP < 55) ~ "Medium",
    MAP >= 55 ~ "High"
  )
)

sink_shap_P <- sink_shap_P %>% mutate(
  MAP_cat = case_when(
    MAP < 40 ~ "Low",
    (MAP >= 40) & (MAP < 55) ~ "Medium",
    MAP >= 55 ~ "High"
  )
)

source_shap_P$MAP_cat <- factor(source_shap_P$MAP_cat, levels = c("Low","Medium","High"))
sink_shap_P$MAP_cat <- factor(sink_shap_P$MAP_cat, levels = c("Low", "Medium", "High"))

# Define colors for each MAP category
my_colors <- c(
  "Low" = "#D81B60",
  "Medium" = "#E69F00",
  "High" = "#1E90FF"
)

# Create x and y limits so that we may compare subplots across the figure
x_limits <- list(
  P_30 = c(min(min(sink_shap_P$P_30),min(source_shap_P$P_30)),
    max(max(sink_shap_P$P_30),max(source_shap_P$P_30))),
  P_sd_30 = c(min(min(sink_shap_P$P_sd_30),min(source_shap_P$P_sd_30)),
    max(max(sink_shap_P$P_sd_30),max(source_shap_P$P_sd_30))),
  P_365 = c(min(min(sink_shap_P$P_365),min(source_shap_P$P_365)),
    max(max(sink_shap_P$P_365),max(source_shap_P$P_365))),
  P_sd_365 = c(min(min(sink_shap_P$P_sd_365),min(source_shap_P$P_sd_365)),
    max(max(sink_shap_P$P_sd_365),max(source_shap_P$P_sd_365)))
)

y_limits <- data.frame(min = min(min(sink_shap_P$shap), min(source_shap_P$shap)),
  max = max(max(sink_shap_P$shap), max(source_shap_P$shap)))

# Creating the plot for the precipitation SHAP values on the extreme sink RF
# Starting by creating labels of R^2 that will be added to the figure
P30_low <- data.frame(var_factor = "P.M", x = 1.5, y = 0.08, label = "R^2 == .47")
P30_med <- data.frame(var_factor = "P.M", x = 1.5, y = .07, label = "R^2 == .48")
P30_high <- data.frame(var_factor = "P.M", x = 1.5, y = .06, label = "R^2 == .25")
Psd30_low <- data.frame(var_factor = "σ(P.M)", x = 20, y = .08, label = "R^2 == .48")
Psd30_med <- data.frame(var_factor = "σ(P.M)", x = 20, y = .07, label = "R^2 == .50")
Psd30_high <- data.frame(var_factor = "σ(P.M)", x = 20, y = .06, label = "R^2 == .22")

```



```

P365_low <- data.frame(var_factor = "P.Y", x = 3.25, y = .08, label = "R^2 == .43")
P365_med <- data.frame(var_factor = "P.Y", x = 3.25, y = .07, label = "R^2 == .17")
P365_high <- data.frame(var_factor = "P.Y", x = 3.25, y = .06, label = "R^2 == .07")
Psd365_low <- data.frame(var_factor = "σ(P.Y)", x = 7.2, y = .08, label = "R^2 == .43")
Psd365_med <- data.frame(var_factor = "σ(P.Y)", x = 7.2, y = .07, label = "R^2 == .03")
a <- data.frame(var_factor = "P.M", x = 7.5, y = -.03, label = "(A)")
b <- data.frame(var_factor = "σ(P.M)", x = 20, y = -.03, label = "(B)")
c <- data.frame(var_factor = "P.Y", x = 3.5, y = -.03, label = "(C)")
d <- data.frame(var_factor = "σ(P.Y)", x = 7, y = -.03, label = "(D)")

# Making var_factor for text an actual factor
levels_order <- c("P.M", "σ(P.M)", "P.Y", "σ(P.Y)")

# Apply to all annotation data frames
P30_low$var_factor <- factor("P.M", levels = levels_order)
P30_med$var_factor <- factor("P.M", levels = levels_order)
P30_high$var_factor <- factor("P.M", levels = levels_order)
Psd30_low$var_factor <- factor("σ(P.M)", levels = levels_order)
Psd30_med$var_factor <- factor("σ(P.M)", levels = levels_order)
Psd30_high$var_factor <- factor("σ(P.M)", levels = levels_order)
P365_low$var_factor <- factor("P.Y", levels = levels_order)
P365_med$var_factor <- factor("P.Y", levels = levels_order)
P365_high$var_factor <- factor("P.Y", levels = levels_order)
Psd365_low$var_factor <- factor("σ(P.Y)", levels = levels_order)
Psd365_med$var_factor <- factor("σ(P.Y)", levels = levels_order)
a$var_factor <- factor(a$var_factor, levels = levels_order)
b$var_factor <- factor(b$var_factor, levels = levels_order)
c$var_factor <- factor(c$var_factor, levels = levels_order)
d$var_factor <- factor(d$var_factor, levels = levels_order)

p1 <- sink_shap_P %>%
  filter(var %in% c("P_30", "P_sd_30", "P_365", "P_sd_365")) %>%
  mutate(var_factor = factor(var, levels = c("P_30", "P_sd_30", "P_365", "P_sd_365"),
    labels = c("P.M", "σ(P.M)", "P.Y", "σ(P.Y)"))) %>%
  ggplot(aes(x = value, y = shap, color = MAP_cat)) +
  geom_point(alpha=.5, size=1) +
  geom_smooth(method = "lm", formula = y ~ x) +
  facet_wrap(~var_factor, scales = "free_x", nrow = 1) +
  faceted_pos_scales(
    x = list(
      scale_x_continuous(limits = x_limits$P_30),
      scale_x_continuous(limits = x_limits$P_sd_30),
      scale_x_continuous(limits = x_limits$P_365),
      scale_x_continuous(limits = x_limits$P_sd_365)),
    ) +
  scale_y_continuous(limits = c(y_limits$min, y_limits$max),
    breaks = c(-0.05, -.04, -.02, 0, .02, .04, .06, 0.08),
    labels = c("-0.05", "-0.04", "-0.02", "0", ".02", ".04", ".06", "0.08"))
+
  scale_color_manual(values = my_colors) +

```

```

labs(x = "Precipitation (mm)", y = "Contribution to extreme sink (SHAP)") +
# Labels
scale_y_continuous(breaks = c(-0.05, -.04, -.02, 0, .02, .04, .06, 0.08),
                    labels = c("-.05", "-0.04", "-.02", "0", ".02", ".04", ".06", "0.08"))
+
geom_text(data = P30_low, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRUE) +
geom_text(data = P30_med, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#E69F00", fontface="bold", size=3, parse = TRUE) +
geom_text(data = P30_high, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#1E90FF", fontface="bold", size=3, parse = TRUE) +
geom_text(data = Psd30_low, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRUE) +
geom_text(data = Psd30_med, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#E69F00", fontface="bold", size=3, parse = TRUE) +
geom_text(data = Psd30_high, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#1E90FF", fontface="bold", size=3, parse = TRUE) +
geom_text(data = P365_low, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRUE) +
geom_text(data = P365_med, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#E69F00", fontface="bold", size=3, parse = TRUE) +
geom_text(data = P365_high, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#1E90FF", fontface="bold", size=3, parse = TRUE) +
geom_text(data = Psd365_low, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRUE) +
geom_text(data = Psd365_med, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "#E69F00", fontface="bold", size=3, parse = TRUE) +
geom_text(data = a, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
geom_text(data = b, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
geom_text(data = c, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
geom_text(data = d, aes(x = x, y = y, label = label, group = var_factor),
          inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
theme_bw() +
theme(
  axis.text = element_text(size = 8),
  axis.title.y = element_text(size = 7),
  axis.title.x = element_blank(),
  axis.text.x = element_text(size = 10),

```

```

    strip.text = element_text(size = 10, face = "bold"),
    legend.position = "none"
  ) +
  guides(color=guide_legend(override.aes=list(fill=NA),
    title = "MAP"))

# Applying the same plot to extreme source RF precipitation SHAP values
# Adding R^2 text to each
P30_low <- data.frame(var_factor = "P.M", x = 7.5, y = 0.08, label = "R^2 == .62")
P30_med <- data.frame(var_factor = "P.M", x = 7.5, y = .07, label = "R^2 == .40")
P30_high <- data.frame(var_factor = "P.M", x = 7.5, y = .06, label = "R^2 == .45")
Psd30_low <- data.frame(var_factor = "σ(P.M)", x = 20, y = .08, label = "R^2 == .59")
Psd30_med <- data.frame(var_factor = "σ(P.M)", x = 20, y = .07, label = "R^2 == .38")
Psd30_high <- data.frame(var_factor = "σ(P.M)", x = 20, y = .06, label = "R^2 == .40")
P365_low <- data.frame(var_factor = "P.Y", x = 3.25, y = .08, label = "R^2 == .54")
P365_med <- data.frame(var_factor = "P.Y", x = 3.25, y = .07, label = "R^2 == .16")
Psd365_low <- data.frame(var_factor = "σ(P.Y)", x = 6.5, y = .08, label = "R^2 == .41")
Psd365_high <- data.frame(var_factor = "σ(P.Y)", x = 6.5, y = .07, label = "R^2 == .09")
d <- data.frame(var = "P.M", x = 7.5, y = -.03, label = "(E)")
e <- data.frame(var = "σ(P.M)", x = 20, y = -.03, label = "(F)")
f <- data.frame(var = "P.Y", x = 3.5, y = -.03, label = "(G)")
h <- data.frame(var = "σ(P.Y)", x = 7, y = -.03, label = "(H)")

# Converting to a leveled factor
levels <- c("P.M", "σ(P.M)", "P.Y", "σ(P.Y)")
P30_low$var_factor <- factor(P30_low$var, levels = levels)
P30_med$var_factor <- factor(P30_med$var, levels = levels)
P30_high$var_factor <- factor(P30_high$var, levels = levels)
Psd30_low$var_factor <- factor(Psd30_low$var, levels = levels)
Psd30_med$var_factor <- factor(Psd30_med$var, levels = levels)
Psd30_high$var_factor <- factor(Psd30_high$var, levels = levels)
P365_low$var_factor <- factor(P365_low$var, levels = levels)
P365_med$var_factor <- factor(P365_med$var, levels = levels)
Psd365_low$var_factor <- factor(Psd365_low$var, levels = levels)
Psd365_high$var_factor <- factor(Psd365_high$var, levels = levels)
d$var_factor <- factor(d$var, levels = levels)
e$var_factor <- factor(e$var, levels = levels)
f$var_factor <- factor(f$var, levels = levels)
h$var_factor <- factor(h$var, levels = levels)

# Plot with custom colors
p2 <- source_shap_P %>%
  filter(var %in% c("P_30", "P_sd_30", "P_365", "P_sd_365")) %>%
  mutate(var_factor = factor(var, levels = c("P_30", "P_sd_30", "P_365", "P_sd_365"),
    labels = c("P.M", "σ(P.M)", "P.Y", "σ(P.Y)"))) %>%
  ggplot(aes(x = value, y = shap, color = MAP_cat)) +
  geom_point(alpha=.5, size=1) +
  geom_smooth(method = "lm", formula = y ~ x) +
  facet_wrap(~var_factor, scales = "free_x", nrow = 1) +
  facetted_pos_scales(
    x = list(

```

```

    scale_x_continuous(limits = x_limits$P_30),
    scale_x_continuous(limits = x_limits$P_sd_30),
    scale_x_continuous(limits = x_limits$P_365),
    scale_x_continuous(limits = x_limits$P_sd_365)),
  ) +
  scale_color_manual(values = my_colors) +
  labs(x = "Precipitation (mm)", y = "Contribution to extreme source (SHAP)", color = "M
AP") +
  scale_y_continuous(limits = c(y_limits$min, y_limits$max),
    breaks = c(-0.05, -.04, -.02, 0, .02, .04, .06, 0.08),
    labels = c("-0.05", "-0.04", "-.02", "0", ".02", ".04", ".06", "0.08"))
+
  geom_text(data = P30_low, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = P30_med, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#E69F00", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = P30_high, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#1E90FF", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = Psd30_low, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = Psd30_med, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#E69F00", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = Psd30_high, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#1E90FF", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = P365_low, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = P365_med, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#E69F00", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = Psd365_high, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#1E90FF", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = Psd365_low, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "#D81B60", fontface="bold", size=3, parse = TRU
E) +
  geom_text(data = d, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
  geom_text(data = e, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
  geom_text(data = f, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
  geom_text(data = h, aes(x = x, y = y, label = label, group = var_factor),
    inherit.aes = FALSE, color = "black", fontface = "bold", size = 3) +
  theme_bw() +
  theme(

```

```

axis.text = element_text(size = 8),
axis.title.y = element_text(size = 7),
axis.title.x = element_text(size = 10),
axis.text.x = element_text(size = 10),
strip.text = element_text(size = 10, face = "bold"),
legend.key = element_rect(color = "white", fill = "white"),
legend.text = element_text(size = 10),
legend.title = element_text(size = 10)
) +
guides(color=guide_legend(override.aes=list(fill=NA)))

# Finally, stacking the two plots
p1 / p2

```

Figure 4. SHAP values for the standard deviation of daily shortwave radiation over the previous year, $\sigma(\text{SW.Y})$, for (A) extreme sink and (B) extreme source classifications. Symbol color indicates whether the site belongs to a Mediterranean climate (red) or a non-Mediterranean climate (yellow). Quadratic regressions fit to all 14 sites are shown as solid lines. Dashed lines depict linear regressions fit to those four sites with a Mediterranean climate: US-Ton, US-Var, US-Me2, and US-Me6. Dotted lines depict linear regressions fit to all other 10 sites. R² values are shown for linear regressions with statistically significant slopes ($p < 0.05$).

[Hide](#)

```

# Grouping SHAP values by belonging to a Mediterranean climate or not
sink_shap_SW <- sink_shap_SW %>% mutate(
  lat_cat = case_when(
    lat < 38 ~ "Low",
    lat >= 38 ~ "High"
  )
)

source_shap_SW <- source_shap_SW %>% mutate(
  lat_cat = case_when(
    lat < 38 ~ "Low",
    lat >= 38 ~ "High"
  )
)

sink_shap_SW <- sink_shap_SW %>% mutate(
  elev_cat = case_when(
    elev_meters < 1300 ~ "Low",
    elev_meters >= 1300 ~ "High"
  )
)

source_shap_SW <- source_shap_SW %>% mutate(
  elev_cat = case_when(
    elev_meters < 1300 ~ "Low",
    elev_meters >= 1300 ~ "High"
  )
)

sink_shap_SW <- sink_shap_SW %>% mutate(
  med_cat = case_when(
    (lat_cat == "High") & (elev_meters <= 1253) ~ "Mediterranean",
    (lat_cat == "Low") | (elev_meters > 1253) ~ "Other"
  )
)

source_shap_SW <- source_shap_SW %>% mutate(
  med_cat = case_when(
    (lat_cat == "High") & (elev_meters <= 2153) ~ "Mediterranean",
    (lat_cat == "Low") | (elev_meters > 1253) ~ "Other"
  )
)

levels <- c("SW", "SW_sd_365")
levels2 <- c("SW_30", "SW_sd_365")
levels3 <- c("Mediterranean", "Other")

# Plotting SW SHAP values for extreme sink RF
p1 <- sink_shap_SW %>%
  filter(var == "SW_sd_365") %>%
  ggplot(aes(x = value, y = shap, color = med_cat)) +
  geom_point(size = 2) +

```

```

geom_smooth(method = "lm",
             formula = y ~ x + I(x^2),
             color = "black",
             lty = "solid",
             aes(group = 1),
             fill = "grey77") +
geom_smooth(data = . %>% filter(elev_meters %in% c(1253, 998, 177, 129)),
             method = "lm",
             formula = y ~ x,
             aes(group = 1),
             color = "black",
             lty = "dashed",
             fill = "grey77") +
geom_smooth(data = . %>% filter(!(elev_meters %in% c(1253, 998, 177, 129))),
             method = "lm",
             formula = y ~ x,
             aes(group = 1),
             color = "black",
             lty = "dotted",
             fill = "grey77") +
theme_bw() +
labs(x = NULL,
     y = "Contribution to extreme flux (SHAP)",
     color="Climate") +
theme(
  legend.position = "none",
  axis.text = element_text(size = 10),
  axis.title = element_text(size = 10)
) +
scale_y_continuous(limits = c(-.03, .06)) +
geom_text(aes(x = 80, y = .055, label = "(A) Extreme Sink"), size = 4, color = "black") +
geom_text(aes(x = 78, y = .0475, label = "R^2 == .176"), color = "#D81B60", parse = TRUE, size = 5) +
geom_text(aes(x = 78, y = .04, label = "R^2 == .081"), color = "#E69F00", parse = TRUE, size = 5) +
scale_color_manual(values = c("Mediterranean" = "#D81B60", "Other" = "#E69F00"))

p2 <- source_shap_SW %>%
  filter(var == "SW_sd_365") %>%
  ggplot(aes(x = value, y = shap, color = med_cat)) +
  geom_point(size = 2) +
  geom_smooth(method = "lm",
             formula = y ~ x + I(x^2),
             color = "black",
             lty = "solid",
             aes(group = 1),
             fill = "grey77") +
  geom_smooth(data = . %>% filter(elev_meters %in% c(1253, 998, 177, 129)),
             method = "lm",
             formula = y ~ x,
             aes(group = 1),

```

```

        color = "black",
        lty = "dashed",
        fill = "grey77") +
geom_smooth(data = . %>% filter(!(elev_meters %in% c(1253, 998, 177, 129))),
            method = "lm",
            formula = y ~ x,
            aes(group = 1),
            color = "black",
            lty = "dotted",
            fill = "grey77") +
theme_bw() +
labs(x = expression(paste("Daily variation in shortwave radiation over previous year
(W m-2,"))),
     y = "",
     color="Climate") +
scale_y_continuous(limits = c(-.03, .06)) +
theme(
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank(),
  legend.text = element_text(size = 10),
  legend.title = element_text(size = 10),
  axis.text.x = element_text(size = 10),
  axis.title.x = element_text(size = 10)
) +
geom_text(aes(x = 82, y = .055, label = "(B) Extreme Source"), size = 4, color = "black") +
geom_text(aes(x = 78, y = .0475, label = "R2 == .203"), color = "#D81B60", parse = TRUE, size = 5) +
geom_text(aes(x = 78, y = .04, label = "R2 == .082"), color = "#E69F00", parse = TRUE, size = 5) +
scale_color_manual(values = c("Mediterranean" = "#D81B60", "Other" = "#E69F00"))

p1 <- p1 + labs(x = NULL)
p2 <- p2 + labs(x = NULL)

(p1 | p2) +
  patchwork::plot_annotation(
    caption = "Daily variation in shortwave radiation over previous year (W m-2)",
    theme = theme(
      plot.caption = element_text(size = 10, hjust = 0.4, margin = margin(t = 20))
    )
  )

```