

R version 4.4.0 (2024-04-24) -- "Puppy Cup"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.80 (8376) aarch64-apple-darwin20]

[History restored from /Users/alperkaragol/.Rapp.history]

```
> setwd("/Users/alperkaragol/Desktop/Consurf/")
> # Set the path to your folder
> folder_path <- "/Users/alperkaragol/Desktop/Consurf/"
>
> # List all files in the folder with the pattern "pathogenicity_scores_"
> file_names <- list.files(path = folder_path, pattern = "*\\.txt$", full.names = TRUE)
>
> # Initialize an empty list to store the medians
> parcor <- list()
> # Loop through each file
> for (file_name in file_names) {
+   # Extract the base name of the file
+   base_name <- basename(file_name)
+
+   # Extract the last three characters before the ".txt"
+   vector_name <- substr(base_name, nchar(base_name) - 5, nchar(base_name) - 4)
+
+   # Read the file into a vector
+   file_vector <- scan(file_name, sep = ",", what = numeric())
+
+   # Assign the vector to a variable with the extracted name
+   assign(vector_name, file_vector)
+
+   print(vector_name)
+ }
Read 74679 items
[1] "A"
Read 74679 items
[1] "C"
Read 74679 items
[1] "CS"
Read 74679 items
[1] "D"
Read 74679 items
[1] "E"
Read 74679 items
[1] "F"
Read 74679 items
[1] "G"
Read 74679 items
[1] "H"
Read 74679 items
[1] "I"
Read 74679 items
[1] "K"
Read 74679 items
[1] "L"
Read 74679 items
[1] "M"
Read 74679 items
[1] "N"
Read 74679 items
[1] "P"
Read 74679 items
[1] "Q"
```

```

Read 74679 items
[1] "R"
Read 74679 items
[1] "S"
Read 74679 items
[1] "T"
Read 74679 items
[1] "V"
Read 74679 items
[1] "W"
Read 74679 items
[1] "Y"
> cor.test(I, T, alternative = "two.sided", method = "spearman", exact=FALSE )

Spearman's rank correlation rho

data: I and T
S = 5.1912e+13, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.2521385

> cor.test(A, T, alternative = "two.sided", method = "spearman", exact=FALSE )

Spearman's rank correlation rho

data: A and T
S = 3.5986e+13, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.4815717

> cor.test(T, V, alternative = "two.sided", method = "spearman", exact=FALSE )

Spearman's rank correlation rho

data: T and V
S = 4.7492e+13, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.3158177

> library(ppcor)
Loading required package: MASS
> F_rank <- rank(F)
> Y_rank <- rank(Y)
> W_rank <- rank(W)
> I_rank <- rank(I)
> T_rank <- rank(T)
> M_rank <- rank(M)
> A_rank <- rank(A)
> V_rank <- rank(V)
> G_rank <- rank(G)
> S_rank <- rank(S)
> L_rank <- rank(L)
> C_rank <- rank(C)
> R_rank <- rank(R)
> Q_rank <- rank(Q)
> N_rank <- rank(N)
> H_rank <- rank(H)
>
> pcor.test(I_rank, T_rank, CS)
  estimate p.value statistic      n gp Method
1 0.1921442      0 53.50401 74679 1 pearson
> pcor.test(A_rank, T_rank, CS)
  estimate p.value statistic      n gp Method
1 0.3815911      0 112.8135 74679 1 pearson
> pcor.test(V_rank, T_rank, CS)
  estimate p.value statistic      n gp Method
1 0.2326689      0 65.37539 74679 1 pearson
> pcor.test(V_rank, T_rank, A_rank)
  estimate p.value statistic      n gp Method
1 0.182528      0 50.73151 74679 1 pearson
> pcor.test(I_rank, T_rank, A_rank)
  estimate p.value statistic      n gp Method

```

```

1 0.1835468      0  51.02452 74679  1 pearson
> pcor.test(A_rank, T_rank, I_rank)
  estimate p.value statistic      n gp Method
1 0.4554109      0  139.787 74679  1 pearson
> pcor.test(V_rank, T_rank, I_rank)
  estimate p.value statistic      n gp Method
1 0.2011892      0  56.12645 74679  1 pearson
> pcor.test(I_rank, T_rank, V_rank)
  estimate      p.value statistic      n gp Method
1 0.04392292 3.204824e-33  12.01437 74679  1 pearson
> pcor.test(A_rank, T_rank, V_rank)
  estimate p.value statistic      n gp Method
1 0.4186155      0  125.9626 74679  1 pearson
>
>
>
>

```