Tutorial 7: Calculation of distances between proteins

DFM & PL

2022-03-12

Contents

Finding proteins with similar profiles

We may find the proteins with profiles nearest to a given protein using the function "nearestProts". Distance is computed as the Euclidean distance between profiles. To use the function, we first use the R function `dist` to create a distance matrix for the proteins in a list of mean profiles, such as `protProfileNSA_AT5tmtMS2`. For clarity of presentation, we rename the embedded data sets to remove experiment-specific labels.

```
library(protlocassign)
data(protNSA_AT5tmtMS2)
data(totProtAT5)
protNSA <- protNSA_AT5tmtMS2
totProt <- totProtAT5
distUseNSA <- dist(protNSA[,1:9], method="euclidean")
```

Then select the protein names:

```
protsUse <- rownames(protNSA)
```

Finally, provide a protein name. Here, for the protein "CTSD", we find the 10 nearest proteins.

```
nearestProts(protName="CTSD", n.nearest=10,  distProts=distUseNSA, protNames=protsUse,
             profile=protNSA)
```

```
#>    protName euclidean distance
#> 1      CTSD          0.00000000
#> 2      HPSE          0.02076348
#> 3      NEU1          0.02640476
#> 4    TM7SF3          0.02657160
#> 5  STARD3NL          0.02851244
#> 6   SLC15A4          0.02886989
#> 7     DNAH6          0.02954456
#> 8      CTSZ          0.03200073
#> 9      LYZ2          0.03213164
#> 10    MFSD8          0.03256431
```

Instead of using normalized specific amounts, we may transform them to relative specific amounts:

```
protProfileLevelsRSA <- RSAfromNSA(NSA=protNSA[,1:9],
                                   NstartMaterialFractions=6, totProt=totProt)
distUseRSA <- dist(protProfileLevelsRSA, method="euclidean")
nearestProts(protName="CTSD", n.nearest=10,  distProts=distUseRSA, protNames=protsUse,
             profile=protProfileLevelsRSA)
```

```
#>         protName euclidean distance
#> 1           CTSD          0.0000000
#> 2          ASAH2          0.8280874
#> 3          DIRC2          0.8288559
#> 4          DNAH6          0.8450258
#> 5       STARD3NL          0.8488629
#> 6            GGH          0.8514247
#> 7          MANBA          0.8984153
#> 8  LOC100909630          0.9363477
#> 9        SLC38A6          0.9589476
#> 10         PLBD1          0.9844416
```

Note that if one wants to generate a table listing the distances between all protein pairs, one needs to convert the distUse or distUseRSA to a matrix. We show the first five rows and columns here:

```
distUseNSAmatrix <- as.matrix(distUseNSA)
distUseNSAmatrix[1:5,1:5]
```

```
#>                 2900026A02RIK      A1CF A930018M24RIK      AAAS AABR07001519.1
#> 2900026A02RIK       0.0000000 0.2049219     0.2751973 0.3312602       0.2577645
#> A1CF                0.2049219 0.0000000     0.4238232 0.2400116       0.3664503
#> A930018M24RIK       0.2751973 0.4238232     0.0000000 0.4943403       0.4076998
#> AAAS                0.3312602 0.2400116     0.4943403 0.0000000       0.4950206
#> AABR07001519.1      0.2577645 0.3664503     0.4076998 0.4950206       0.0000000
```

This matrix can be written to a local directory using standard procedures.

Reproducibility

```
print(utils::sessionInfo(), width=80)
```

```
#> R version 4.1.3 (2022-03-10)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 19044)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=English_United States.1252
#> [2] LC_CTYPE=English_United States.1252
#> [3] LC_MONETARY=English_United States.1252
#> [4] LC_NUMERIC=C
```

```
#> [5] LC_TIME=English_United States.1252
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] BiocParallel_1.28.3  outliers_0.14        plot.matrix_1.6.1
#> [4] pracma_2.3.8         protlocassign_0.99.1 lme4_1.1-28
#> [7] Matrix_1.4-0
#>
#> loaded via a namespace (and not attached):
#>  [1] Rcpp_1.0.8        lattice_0.20-45   snow_0.4-4        prettyunits_1.1.1
#>  [5] ps_1.6.0          rprojroot_2.0.2   digest_0.6.29     utf8_1.2.2
#>  [9] R6_2.5.1          evaluate_0.15     ggplot2_3.3.5     highr_0.9
#> [13] pillar_1.7.0      rlang_1.0.2       rstudioapi_0.13   minqa_1.2.4
#> [17] callr_3.7.0       nloptr_2.0.0      rmarkdown_2.13    desc_1.4.1
#> [21] devtools_2.4.3    splines_4.1.3     stringr_1.4.0     munsell_0.5.0
#> [25] tinytex_0.37      compiler_4.1.3    xfun_0.30         pkgconfig_2.0.3
#> [29] pkgbuild_1.3.1    htmltools_0.5.2   tibble_3.1.6      gridExtra_2.3
#> [33] BB_2019.10-1      quadprog_1.5-8    fansi_1.0.2       viridisLite_0.4.0
#> [37] crayon_1.5.0      withr_2.5.0       MASS_7.3-55       brio_1.1.3
#> [41] grid_4.1.3        nlme_3.1-155      gtable_0.3.0      lifecycle_1.0.1
#> [45] magrittr_2.0.2    scales_1.1.1      cli_3.2.0         stringi_1.7.6
#> [49] cachem_1.0.6      viridis_0.6.2     fs_1.5.2          remotes_2.4.2
#> [53] testthat_3.1.2    ellipsis_0.3.2    vctrs_0.3.8       boot_1.3-28
#> [57] tools_4.1.3       glue_1.6.2        purrr_0.3.4       processx_3.5.2
#> [61] pkgload_1.2.4     parallel_4.1.3    fastmap_1.1.0     yaml_2.3.5
#> [65] colorspace_2.0-3  sessioninfo_1.2.2 memoise_2.0.1     knitr_1.37
#> [69] usethis_2.1.5
```