The *proteinProfiles* package

Julian Gehring

June 20, 2024

1 Introduction

1.1 Motivation and method

In current high-throughput proteomics, it is feasible to assess the abundance of a large number of proteins in one measurement. In case these measurements correspond to different time points, it is often of interest to identify groups of proteins showing similar time courses.

The *proteinProfiles* package offers the functionality to

1. Define protein groups of interest based on matching text annotation.
2. Compute similarity (distance) measures of time courses for a set of proteins.
3. Assess the significance of the similarity in terms of p-values in relation to randomly permuted sets.

A detailed use case for this method is described in **(author?)** [1].

1.2 About the package

To use the functions and the data described in this document, you have to load the package first:

```
> library(proteinProfiles)
```

If you have not installed the package so far, you can do this in the same way as for any other bioconductor package (see also http://bioconductor.org/install/ for details):

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("proteinProfiles")
```

You can get more information about the package in general and specific function (e.g. the *profileDistance* function) with:

```
> vignette(package="proteinProfiles")
> vignette("proteinProfiles")
> ?profileDistance
```

2 Data import and structure

For illustrating a typical workflow, we will use an example data set which mimics the data used in **(author?)** [1].

```
> data(ips_sample)
> ls()
```

```
[1] "annotation" "ratios"
```

For the analysis, you need first the abundance measurements for the proteins over time. These can be absolute or relative values, and can optionally include replicates. The data is stored as a numeric matrix, with rows corresponding to proteins and columns to time points/replicates.

```
> head(ratios)
```

```
      T3R1   T3R2   T6R1   T6R2   T9R1   T9R2  T12R1  T12R2
46   1.770 -1.847 -1.462  1.066 -0.024 -0.110 -0.037  1.476
148 -0.974 -1.387  0.549 -0.440 -0.278 -0.253  0.337  0.332
169  0.177  0.111 -0.047  0.212  0.008 -0.086 -0.180 -0.229
262 -0.688 -0.822 -0.014  0.074 -0.108 -0.037  0.117  0.111
303  0.857  0.985  0.190 -0.090  0.211 -0.319 -0.269 -0.019
386 -0.418 -0.556 -0.493  0.127 -0.070 -0.644  0.440  0.793
     T15R1  T15R2
46   0.449 -0.088
148  0.379  0.265
169 -0.030 -0.146
262  0.407  0.291
303 -1.010 -1.106
386  0.927  1.067
```

Further, you have to provide a data frame with annotation data associated with proteins. This can include multiple annotation columns, as shown in the example data set.

```
> colnames(annotation)
```

```
[1] "Uniprot"            "Protein.Short.Name"
[3] "Gene.Name"          "Protein.Name"
[5] "GOCC"               "GOBP"
[7] "GOMF"               "KEGG"
```

The matching of the annotation to the measurements relies on a custom identifier which is stored as row names in both *ratios* and *annotation*.

3 Removing features with missing data

Not for all data points the measurement was successful and hence contains missing data (*NA*). Since computing the distances of profiles with several data points missing may be unreliable, you can optionally remove protein measurements with the fraction of missing data points exceeding a user-defined threshold. A threshold of e.g. 0.3 will remove all features with more than 30% of the data points missing.

```
> ratios_filtered <- filterFeatures(ratios, 0.3, verbose=TRUE)

[1] "Before filtering: 247"
[1] "After filtering: 230"
```

4 Defining protein group of interest based on annotation

Based on the annotation provided in the original data set, a group of proteins of interest can be obtained. The `grepAnnotation` function matches substrings (regular expressions) against a column of the annotation object and returns the matching protein identifiers. Here, we search for all protein names starting with the string *"28S"*. For details, read the documentation of the *grep* function.

```
> names(annotation)

[1] "Uniprot"            "Protein.Short.Name"
[3] "Gene.Name"          "Protein.Name"
[5] "GOCC"               "GOBP"
[7] "GOMF"               "KEGG"

> index_28S <- grepAnnotation(annotation, pattern="^28S", column="Protein.Name")
> index_28S

 [1] "1356" "1507" "1723" "2298" "2385" "3469" "4650" "4900"
 [9] "5476" "5652" "6249" "6624" "6754" "6804" "6846" "6883"
[17] "6891" "7004" "7010" "7087" "7106" "7415" "7471"
```

We can also use other columns of the annotation. Here, we search for all proteins associated with the term *"Ribosome"* in the annotation column, taken from the KEGG database.

```
> index_ribosome <- grepAnnotation(annotation, "Ribosome", "KEGG")
> index_ribosome
```

```
 [1] "46"   "169"  "497"  "578"  "664"  "1295" "1300" "1324"
 [9] "1358" "1372" "1390" "1510" "1532" "1533" "1551" "1588"
[17] "1850" "1987" "2088" "2481" "2714" "3050" "3051" "3059"
[25] "3103" "3148" "3248" "3249" "3268" "3330" "3332" "3333"
[33] "3334" "3380" "3515" "3530" "3534" "3537" "3550" "3555"
[41] "3559" "3560" "3561" "3562" "3563" "3581" "3583" "3588"
[49] "3593" "3594" "3599" "3601" "3602" "3603" "3605" "3631"
[57] "3633" "3637" "3704" "3760" "3782" "3885" "4561" "5254"
[65] "5259" "5260" "5261" "5263" "5265" "6442" "6698" "6913"
[73] "7129"
```

5 Computing profile distances and assessing significance

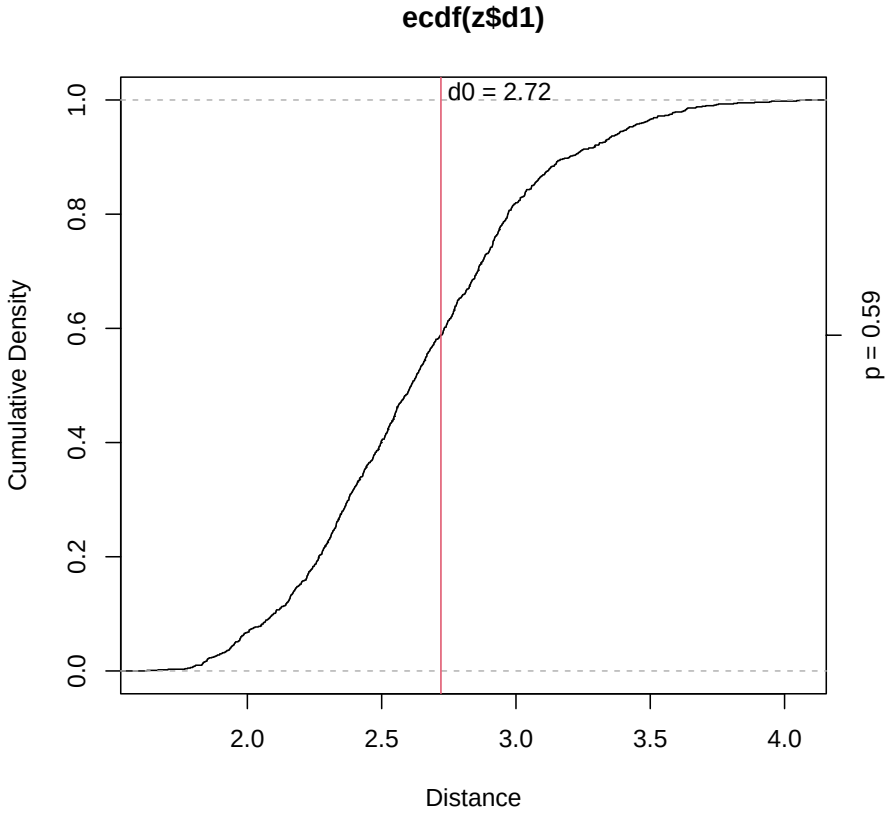
The `profileDistance` function constitutes the core part of the analysis.

1. It computes the mean euclidean distance d_0 of the profiles for the proteins of interest defined by *index*. This distance is shown as a red vertical line in the plot.

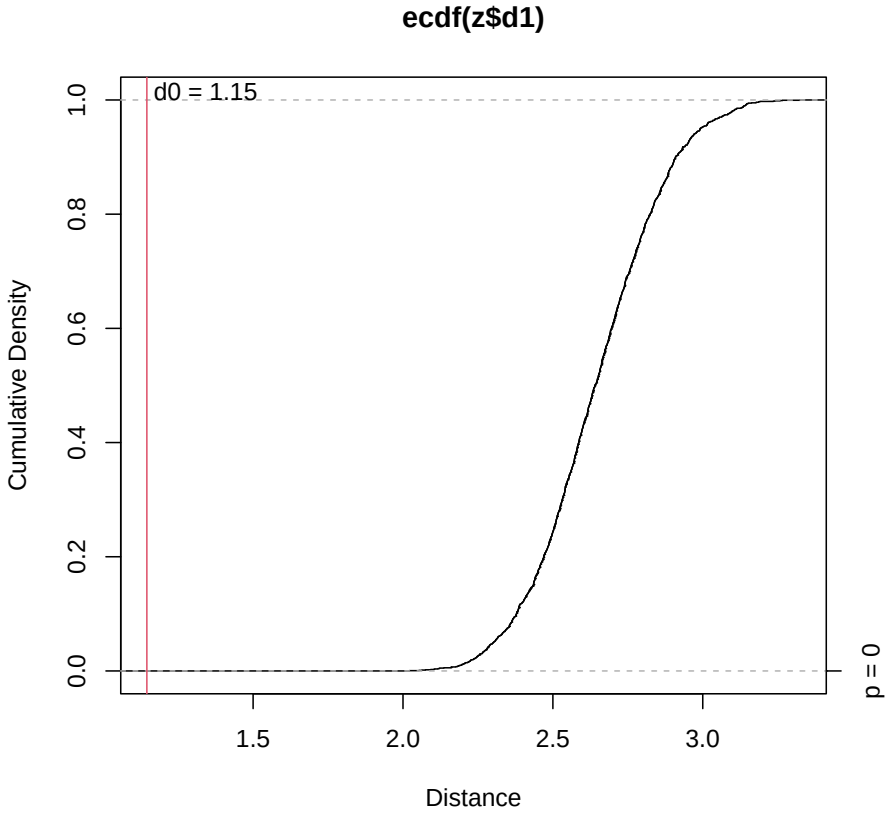
2. It performs step (1) for a number *nSample* of randomly selected groups with the same size as our group of interest. The distances are shown as a cumulative distribution in the plot.

3. Based on the results of step (1) and (2), a p-value p given by the cumulative density at d_0 (which is equivalent to the area under the probability density in the range $[-\infty, d_0]$) is computed. It indicates the probability of observing a group of proteins by chance with profiles having the same or a smaller distance as our group of interest.

```
> z1 <- profileDistance(ratios, index_28S)
> z1$d0

[1] 2.720685

> z1$p.value

[1] 0.588

> plotProfileDistance(z1)
```




```
> z2 <- profileDistance(ratios, index_ribosome, nSample=2000)
> plotProfileDistance(z2)
```


ecdf(z$d1)
d0 = 1.15
Cumulative Density
0.0
0.2
0.4
0.6
0.8
1.0
p = 0
1.5
2.0
2.5
3.0
Distance

References

[1] Jenny Hansson, Mahmoud Reza Rafiee, Sonja Reiland, Jose M. Polo, Julian Gehring, Satoshi Okawa, Wolfgang Huber, Konrad Hochedlinger, and Jeroen Krijgsveld. Highly coordinated proteome dynamics during reprogramming of somatic cells to pluripotency. *Cell Reports*, 2(6):1579–1592, December 2012. 1, 2

Session Info

- R version 4.4.0 (2024-04-24), `x86_64-pc-linux-gnu`
- Running under: `Ubuntu 24.04 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3`
- LAPACK: `/usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.26.so` ; LAPACK version3.12.0
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: proteinProfiles 1.45.0
- Loaded via a namespace (and not attached): buildtools 1.0.0, compiler 4.4.0, knitr 1.47, maketools 1.3.0, sys 3.4.2, tools 4.4.0, xfun 0.44