Mar 29, 2022

idpr Workflow

PLOS One ✓

William McFadden[1,2], Judith Yanowitz[1,3]

[1]Magee-Womens Research Institute; [2]Emory University*;

[3]Dept of OBGYN and Reproductive Sciences, University of Pittsburgh School of Medicine

William McFadden: *current address;

2

dx.doi.org/10.17504/protocols.io.b58gq9tw

PLOS ONE Lab Protocols Magee-Womens Research Institute

yanowitzjl

This protocol details about idpr workflow.

idprWorkflow.Rmd idprWorkflow.pdf

DOI

dx.doi.org/10.17504/protocols.io.b58gq9tw

https://doi.org/10.1371/journal.pone.0266929

William McFadden, Judith Yanowitz 2022. idpr Workflow. **protocols.io**
https://dx.doi.org/10.17504/protocols.io.b58gq9tw

Judith Yanowitz
Grant ID: R01GM127569

protocol

McFadden WM, Yanowitz JL (2022) idpr: A package for profiling and analyzing Intrinsically Disordered Proteins in R. PLOS ONE 17(4): e0266929. https://doi.org/10.1371/journal.pone.0266929

idpr Workflow, alpha-Synuclein Figures, p53 Figures

protocol ,

Mar 11, 2022

Mar 29, 2022

Mar 11, 2022 renuka.s

Mar 21, 2022 yanowitzjl

59368

References

Paper Citations

Erdős, G., & Dosztányi, Z. (2020). Analyzing protein disorder with IUPred2A. Current Protocols in Bioinformatics. https://doi.org/10.1002/cpbi.99

Mészáros, B., Erdős, G., & Dosztányi, Z. (2018). IUPred2A: context-dependent prediction of protein disorder as a function of redox state and protein binding. Nucleic acids research. https://doi.org/10.1093/nar/gky384

Soudy M, Anwar AM, Ahmed EA, Osama A, Ezzeldin S, Mahgoub S, Magdeldin S (2020). UniprotR: Retrieving and visualizing protein sequence and functional information from Universal Protein Resource (UniProt knowledgebase).. Journal of proteomics. https://doi.org/10.1016/j.jprot.2019.103613

UniProt Consortium (2014). UniProt: a hub for protein information. Nucleic acids research. https://doi.org/10.1093/nar/gku989

R / Package Citations

```
citation()

##
## To cite R in publications use:
##
## R Core Team (2021). R: A language and environment for
statistical
## computing. R Foundation for Statistical Computing, Vienna,
Austria.
## URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
## title = {R: A Language and Environment for Statistical
Computing},
## author = {{R Core Team}},
## organization = {R Foundation for Statistical Computing},
## address = {Vienna, Austria},
## year = {2021},
## url = {https://www.R-project.org/},
## }
##
## We have invested a lot of time and effort in creating R,
please cite it
## when using it for data analysis. See also
'citation("pkgname")' for
## citing R packages.

citation("idpr")

##
## To cite package 'idpr' in publications use:
##
## William M. McFadden and Judith L. Yanowitz (2021). idpr:
Profiling
## and Analyzing Intrinsically Disordered Proteins in R. R
package
## version 1.2.0.
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
## title = {idpr: Profiling and Analyzing Intrinsically
Disordered Proteins in R},
## author = {William M. McFadden and Judith L. Yanowitz},
## year = {2021},
## note = {R package version 1.2.0},
## }

citation("Biostrings")

##
## To cite package ’Biostrings’ in publications use:
##
## H. Pagès, P. Aboyoun, R. Gentleman and S. DebRoy (2021).
Biostrings:
## Efficient manipulation of biological strings. R package
version
## 2.60.1. https://bioconductor.org/packages/Biostrings
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
## title = {Biostrings: Efficient manipulation of biological
strings},
## author = {H. Pagès and P. Aboyoun and R. Gentleman and S.
DebRoy},
## year = {2021},
## note = {R package version 2.60.1},
## url = {https://bioconductor.org/packages/Biostrings},
## }
##
## ATTENTION: This citation information has been auto-generated
from the
## package DESCRIPTION file and may need manual editing, see
## ’help("citation")’.

citation("ggplot2")
##
## To cite ggplot2 in publications, please use:
##
## H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
## Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
## @Book{,
## author = {Hadley Wickham},
## title = {ggplot2: Elegant Graphics for Data Analysis},
## publisher = {Springer-Verlag New York},
## year = {2016},
## isbn = {978-3-319-24277-4},
```

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

```
## url = {https://ggplot2.tidyverse.org},
## }

citation("UniprotR")

##
## To cite UniprotR in publications use:
##
## Soudy, M., Anwar, A.M., Ahmed, E.A., Osama, A., Ezzeldin, S.,
## Mahgoub, S. and Magdeldin, S., 2020. UniprotR: Retrieving and
## visualizing protein sequence and functional information from
## Universal Protein Resource (UniProt knowledgebase). Journal
of
## Proteomics, 213, p.103613.
##
## A BibTeX entry for LaTeX users is
##
## @Article{,
## title = {UniprotR: Retrieving and visualizing protein
sequence and functional information from Universal ## author =
{Mohamed Soudy and Ali Mostafa Anwar and Eman Ali Ahmed and Aya
Osama and Shahd Ezzeldin ## journal = {Journal of Proteomics},
## volume = {213},
## pages = {103613},
## year = {2020},
## issn = {1874-3919},
## doi = {10.1016/j.jprot.2019.103613},
## url =
{https://www.sciencedirect.com/science/article/pii/S187439191930
3859},
## }
```

Additional Information:
Session Info

```
sessionInfo()
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:
/Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRb
las.dylib
## LAPACK:
/Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRl
apack.dylib
```

```
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-
8/en_US.UTF-8
##
## attached base packages:
## [1] stats4 parallel stats graphics grDevices utils datasets
## [8] methods base
##
## other attached packages:
## [1] Biostrings_2.60.1 GenomeInfoDb_1.28.1 XVector_0.32.0
## [4] IRanges_2.26.0 S4Vectors_0.30.0 BiocGenerics_0.38.0
## [7] ggplot2_3.3.5 idpr_1.2.0
##
## loaded via a namespace (and not attached):
## [1] colorspace_2.0-2 seqinr_4.2-8
## [3] ggsignif_0.6.2 ellipsis_0.3.2
## [5] rio_0.5.27 qdapRegex_0.7.2
## [7] GenomicRanges_1.44.0 fs_1.5.0
## [9] farver_2.1.0 ggpubr_0.4.0
## [11] alakazam_1.1.0 fansi_0.5.0
## [13] lubridate_1.7.10 xml2_1.3.2
## [15] knitr_1.33 ade4_1.7-17
## [17] jsonlite_1.7.2 Rsamtools_2.8.0
## [19] broom_0.7.9 dbplyr_2.1.1
## [21] data.tree_1.0.0 readr_2.0.1
## [23] compiler_4.1.0 httr_1.4.2
## [25] backports_1.2.1 assertthat_0.2.1
## [27] Matrix_1.3-4 lazyeval_0.2.2
## [29] htmltools_0.5.1.1 prettyunits_1.1.1
## [31] tools_4.1.0 igraph_1.2.6
## [33] gtable_0.3.0 glue_1.4.2
## [35] GenomeInfoDbData_1.2.6 dplyr_1.0.7
## [37] Rcpp_1.0.7 carData_3.0-4
## [39] Biobase_2.52.0 cellranger_1.1.0
## [41] vctrs_0.3.8 ape_5.5
## [43] nlme_3.1-152 xfun_0.25
## [45] stringr_1.4.0 networkD3_0.4
## [47] openxlsx_4.2.4 rvest_1.0.1
## [49] lifecycle_1.0.0 rstatix_0.7.0
## [51] zlibbioc_1.38.0 MASS_7.3-54
## [53] scales_1.1.1 airr_1.3.0
## [55] hms_1.1.0 MatrixGenerics_1.4.1
## [57] SummarizedExperiment_1.22.0 tidyverse_1.3.1
## [59] gprofiler2_0.2.0 yaml_2.2.1
## [61] curl_4.3.2 gridExtra_2.3
## [63] stringi_1.7.3 highr_0.9
## [65] zip_2.2.0 BiocParallel_1.26.1
```

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

```
## [67] rlang_0.4.11 pkgconfig_2.0.3
## [69] matrixStats_0.60.0 bitops_1.0-7
## [71] evaluate_0.14 lattice_0.20-44
## [73] purrr_0.3.4 UniprotR_2.0.8
## [75] labeling_0.4.2 GenomicAlignments_1.28.0
## [77] htmlwidgets_1.5.3 tidyselect_1.1.1
## [79] plyr_1.8.6 magrittr_2.0.1
## [81] R6_2.5.0 magick_2.7.2
## [83] generics_0.1.0 DelayedArray_0.18.0
## [85] DBI_1.1.1 pillar_1.6.2
## [87] haven_2.4.3 foreign_0.8-81
## [89] withr_2.4.2 abind_1.4-5
## [91] RCurl_1.98-1.4 tibble_3.1.3
## [93] modelr_0.1.8 crayon_1.4.1
## [95] car_3.0-11 utf8_1.2.2
## [97] plotly_4.9.4.1 tzdb_0.1.2
## [99] rmarkdown_2.10 progress_1.2.2
## [101] grid_4.1.0 readxl_1.3.1
## [103] data.table_1.14.0 forcats_0.5.1
## [105] reprex_2.0.1 digest_0.6.27
## [107] tidyr_1.1.3 munsell_0.5.0
## [109] viridisLite_0.4.0
```

Runtime

```
#--- End Runtime
end_time <- Sys.time()
time_diff <- end_time - start_time
time_diff

## Time difference of 41.00444 secs
```

Installing idpr: Downloading the Current Release

1

idpr is published in Bioconductor where the stable, released version of the package can be downloaded. The development version, which may be unstable, is published on GitHub.
The package can be installed from Bioconductor with the following line of code. This requires the BiocManager package to be installed.

```
if(!'BiocManager' %in% installed.packages()) {
    install.packages("BiocManager")
}
if(!'idpr' %in% installed.packages()) {
    BiocManager::install("idpr")
}
```

1.1 The **UniprotR** package is used within this workflow to fetch the amino acid sequences for the proteins analyzed. **idpr** contains multiple ways to read in sequences, including from .fasta files *via* **Biostrings**. To avoid distributing additional files, we are utilizing UniprotR to fetch sequences from the **UniProt** database.To run this workflow please install **UniprotR**. **UniprotR** is not a dependency of **idpr**, though this workflow exemplifies how the packages can be used together

1.2

```
if(!'UniprotR' %in% installed.packages()) {
    install.packages("UniprotR")
}
```

Installing idpr: Downloading the Development Version

2 The most recent version of the package can be installed with the following line of code. This requires the devtools package to be installed.

```
if(!'devtools' %in% installed.packages()) {
    install.packages("devtools")
}
if(!'idpr' %in% installed.packages()) {
    devtools::install_github("wmm27/idpr")
}
```

Installing idpr: Loading idpr

3 Once installed, idpr can be loaded with the ‘library’ function.

```
library(idpr)
```

3.1 To test the package is loaded, the **idpr** function ‘netCharge’ will be used to determine the charge of Glutamic Acid (E) at pH 8. Since pH » pKa, the charge of E should be near -1.

```
netCharge("E",
          pH = 8,
          includeTermini = FALSE)
## [1] -0.9997418
```

alpha-Synuclein Figures: Fetching the amino acid sequence

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

4 First, I will use the UniprotR package to get the alpha-synuclein amino acid sequence from the UniProtID.
For alpha-Synuclein, the ID is P37840.

```
## Please wait we are processing your accessions ...
```

Retrieved Sequence:

```
print(a_syn_seq)
## [1]
"MDVFMKGLSKAKEGVVAAAEKTKQGVAEAAGKTKEGVLYVGSKTKEGVVHGVATVAEKTKEQVTNVGG
AVVTGVTAVAQKTVEGAGSIAAATGFVKKDQLGKNEEGAPQEGILEDMPVDPDNEAYEM
```

alpha-Synuclein Figures: Generating the idprofile for alpha-Synuclein

5 To get the ‘idprofile’, a simple function is needed with the sequence and Uniprot ID specified. This generates
all plots in figure 1.

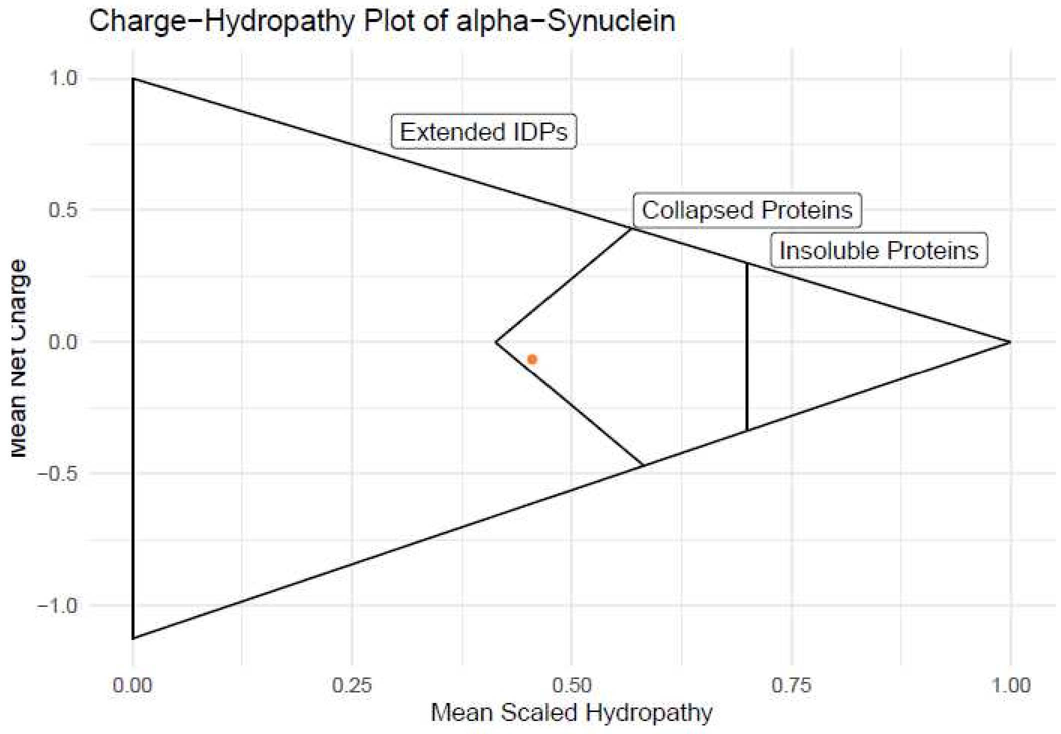
```
idprofile(sequence = a_syn_seq,
          uniprotAccession = "P37840",
          proteinName = "alpha-Synuclein")
```

alpha-Synuclein Figures: Generating the idprofile for alpha-Synuclein

5.1

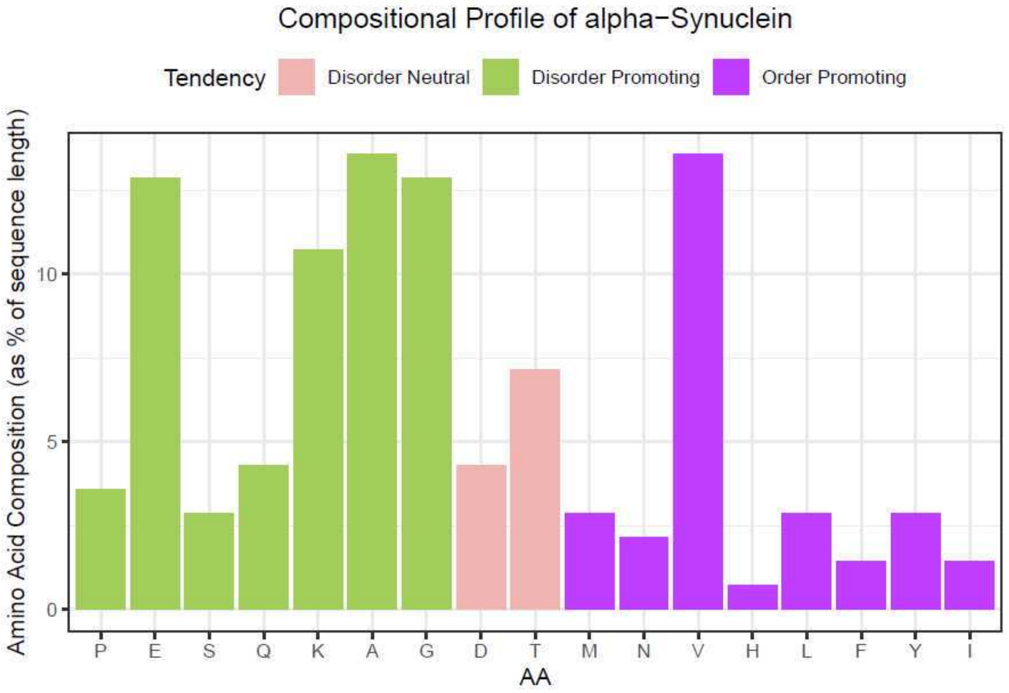
```
## [[1]]
```

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw



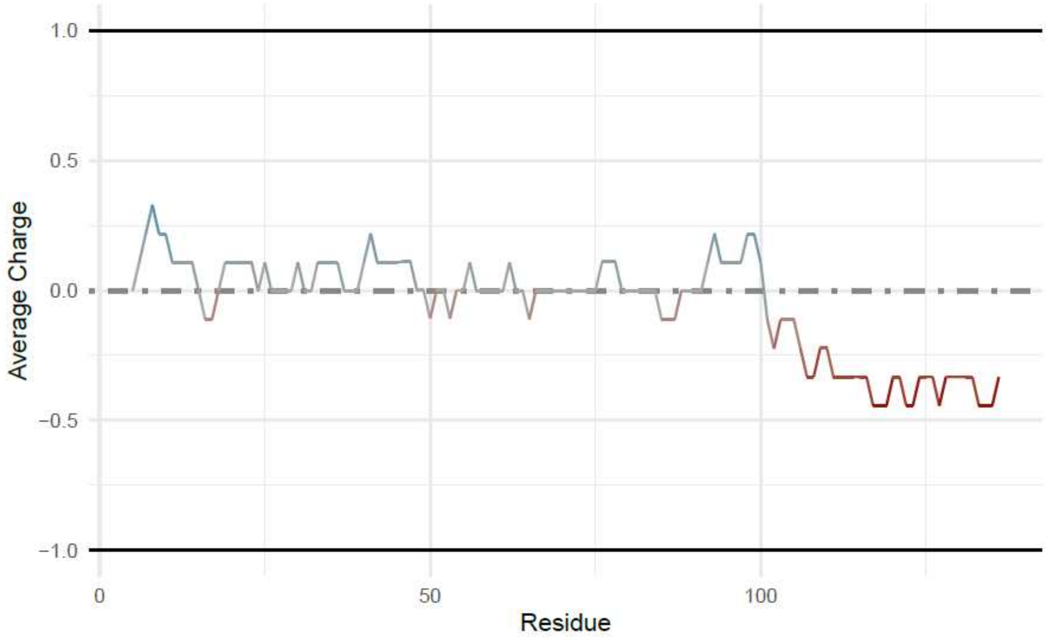
5.2

```
##
## [[2]]
```



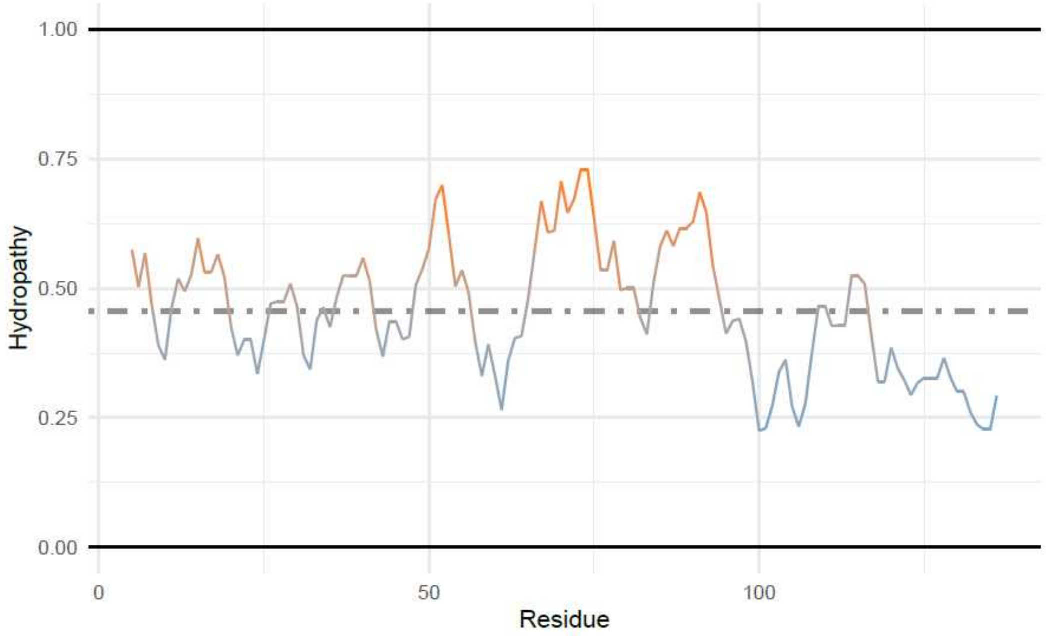
5.3 Calculation of Local Charge in alpha-Synuclein.

Window Size = 9 ; Net Charge = −9.048



```
##
## [[3]]
```

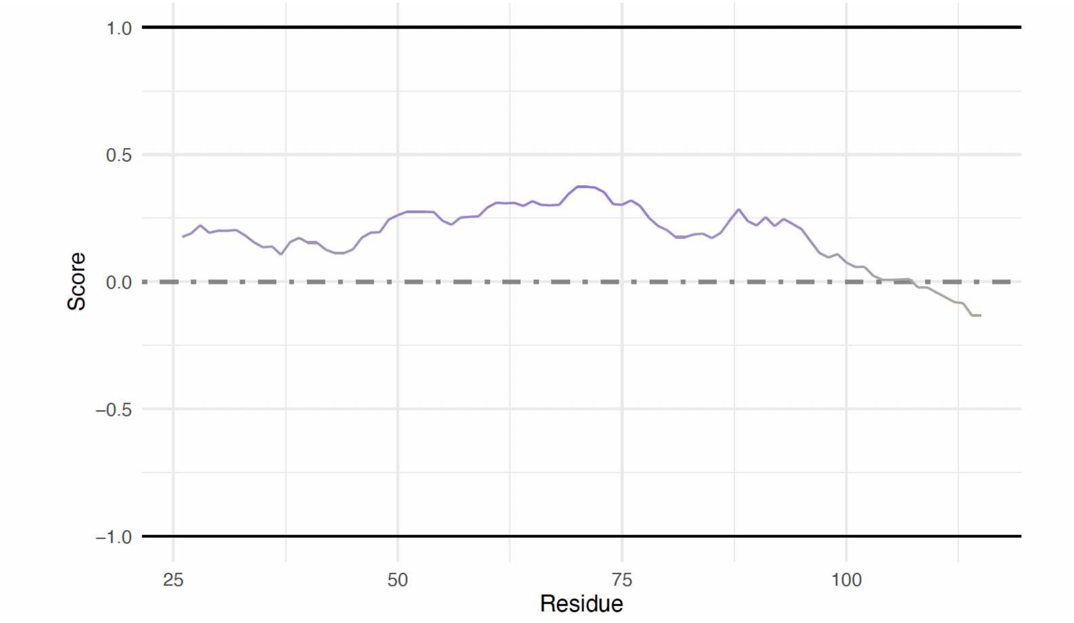
5.4 Measurement of Scaled Hydropathy in alpha−Synuclein.
Window Size = 9 ; Average Scaled Hydropathy = 0.455



```
##
## [[4]]
```

5.5

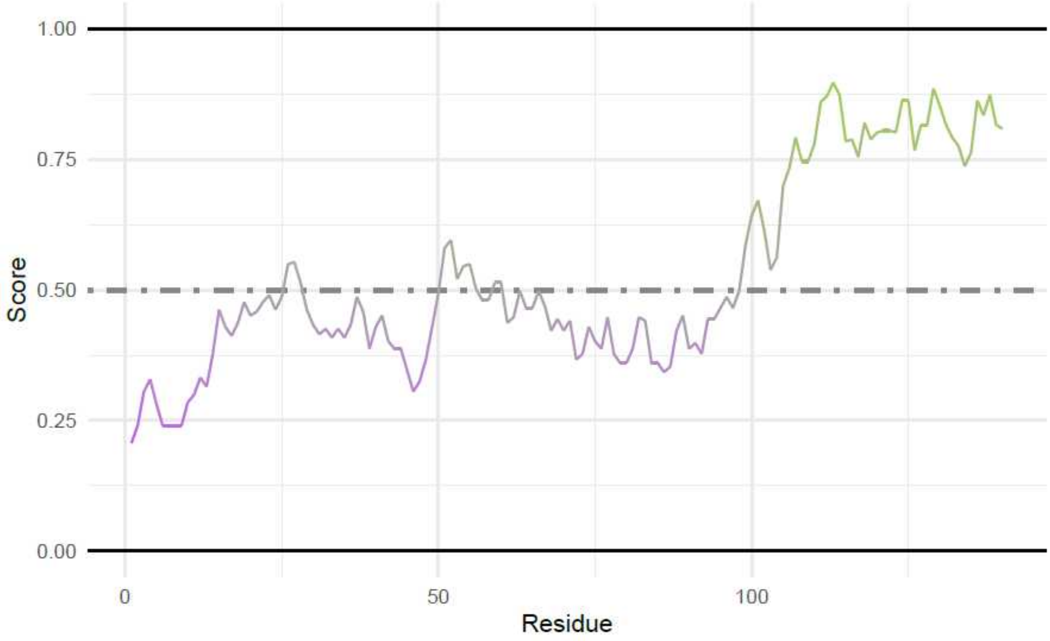
FoldIndex Prediction of Intrinsic Disorder in alpha-Synuclein



```
##
## [[5]]
```

5.6

Prediction of Intrinsic Disorder in alpha-Synuclein.
By IUPred2A long



Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

```
##
## [[6]]
```

alpha-Synuclein Figures: Generating Supplemental Figures for alpha-Synuclein

6 The following code generates plots for supplemental figure 1.

Generating Supplemental Figures for alpha-Synuclein: Charge-Hydropathy plot of protein domains

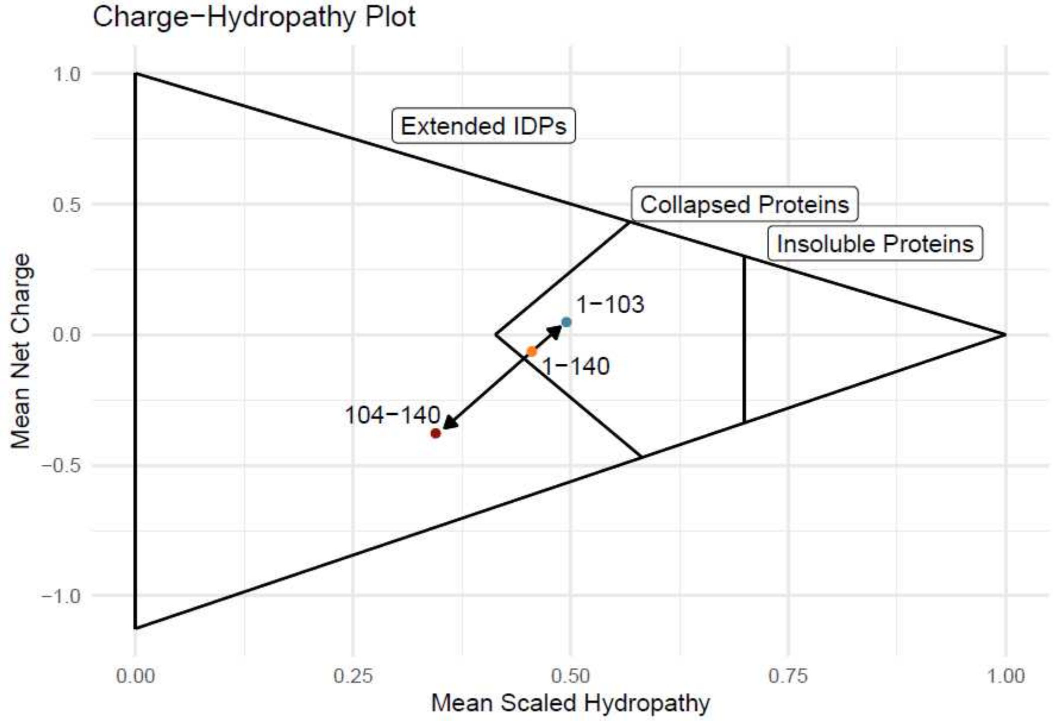
6.1 To add multiple points to the charge hydropathy plot, first the sequence will be split into the N-term (residues
1-103) and C-term (residues 104-140). To do this, I will use the 'AAString' function from **Biostrings**. For
the split sequences and the full length sequence, the average net charge and the mean scaled hydropathy are
calculated and put into a data frame. These coordinates will be used for adding **ggplot2** annotations. Since
idpr depends on both of these packages, they should already be installed.

```
# --- Load packages
library(ggplot2)
library(Biostrings)
```

A	B	C	D
R	**H**	**Name**	**Name_Expression**
0.047803	0.495049	1-103	aSyn [1-103]
-0.37783	0.34473	104-140	aSyn [104-140]
-0.06463	0.455321	1-140	aSyn [1-140]

6.2 Then, the ggplot is made and annotations are added. See **ggplot2** for annotation options.

```
# --- Make the base plot
a_syn_split_plot <- chargeHydropathyPlot(a_syn_seq_split)

# --- Add arrows to plot using ggplot2 geom_segment()
function.
    # Arrows start at aSyn [1-140] and point to domains.
a_syn_split_plot <- a_syn_split_plot +
    geom_segment(aes(x = 0.4553214, y = -0.06462863, xend
= 0.488, yend = 0.03),
                  arrow = arrow(length = unit(0.2, "cm"),
                                type = "closed"))+
    geom_segment(aes(x = 0.4553214, y = -0.06462863, xend
= 0.355, yend = -0.358),
                  arrow = arrow(length = unit(0.2, "cm"),
                                type = "closed"))
# --- Add labels to points with ggplot2 functions
a_syn_split_plot <- a_syn_split_plot +
    geom_text(data = RH_DF,
               aes(x = H,
                   y = R,
                   label = Name),
                nudge_x = c(0.05, -0.05, 0.05),
                nudge_y = c(0.07, 0.070, -0.055)
    )
# --- Adds colored points to plot. Adds on top of
geom_segment.
a_syn_split_plot <- a_syn_split_plot +
    geom_point(data = RH_DF,
               aes(x = H,
                   y = R),
               color = c("#348AA7", "#92140C",
"chocolate1"))
plot(a_syn_split_plot)
```



Generating Supplemental Figures for alpha-Synuclein: Sequence Map showing Familial Mutations

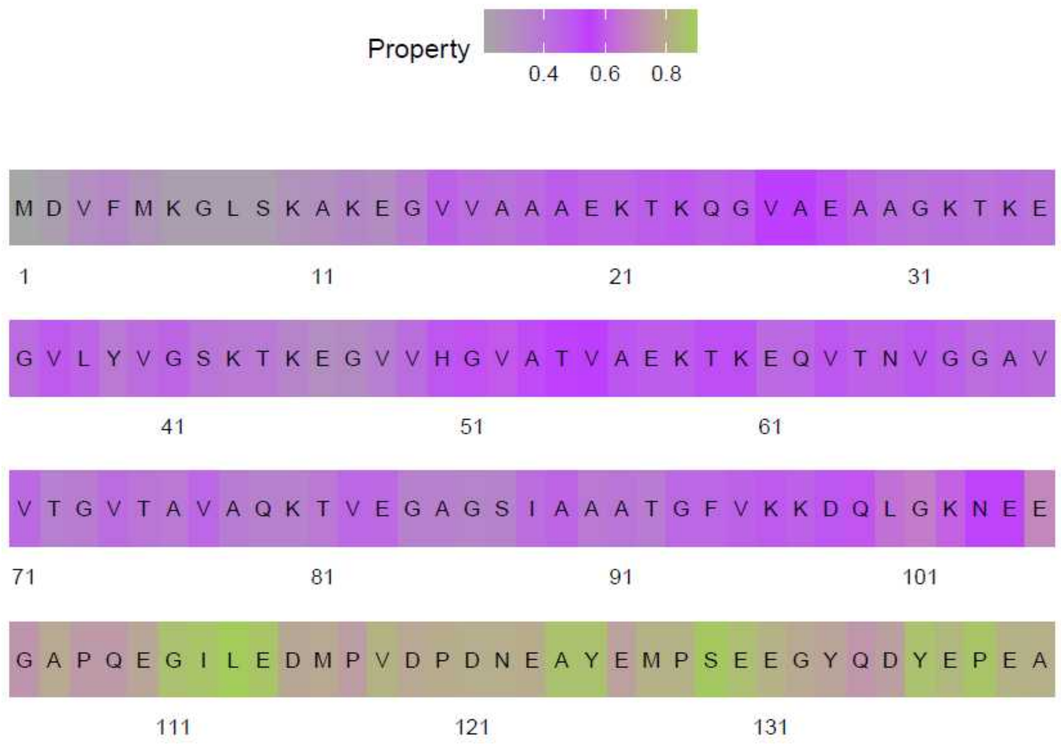
7 Several point mutations in the alpha-Synuclein NTD have been identified that are linked to familial parkinsons disease. These are annotated here in the context of intrinsic disorder predictions from IUPred2. Functions from **ggplot2** are needed for annotations, and therefore this package must be attached if not already.

7.1 First, the data is retrieved from IUPred2. Setting plotResults = FALSE returns a data frame for custom
plotting.

A	B	C
Position	**AA**	**IUPred2**
1	M	0.206376
2	D	0.239899
3	V	0.30533
4	F	0.328603
5	M	0.281712
6	K	0.239899

7.2 Then, a sequence map is created with the IUPred results. Column 2 (a_syn_iupred_df$AA) is a character
vector of individual, single-letter amino acids. Column 3 (a_syn_iupred_df$IUPred2) is a numeric vector
of IUPred2 scores.

```
iupred_map <-
    sequenceMap(sequence = a_syn_iupred_df$AA,
            property = a_syn_iupred_df$IUPred2,
            nbResidues = 35,
            customColors = c("darkolivegreen3", "grey65",
"darkorchid1"))
# --- Plot the unedited, unannotated sequenceMap
plot(iupred_map)
```



7.3 For adding annotations to a sequence map, you can get the position within the plot using the **idpr** function 'sequenceMapCoordinates'. This helps guide or identify the coordinates for **ggplot2** annotations.

A	B	C	D
Position	**AA**	**row**	**col**
1	M	4	1
2	D	4	2
3	V	4	3
4	F	4	4
5	M	4	5
6	K	4	6

7.4 Additionally, several annotations are added and the plot themes are edited. See code for all annotations.

7.5 Adding the labels for Familial Mutations and '*' to add above the mutated residues. These positions are determined by 'sequenceMapCoordinates' and values are added to row (y) value to move annotations above

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

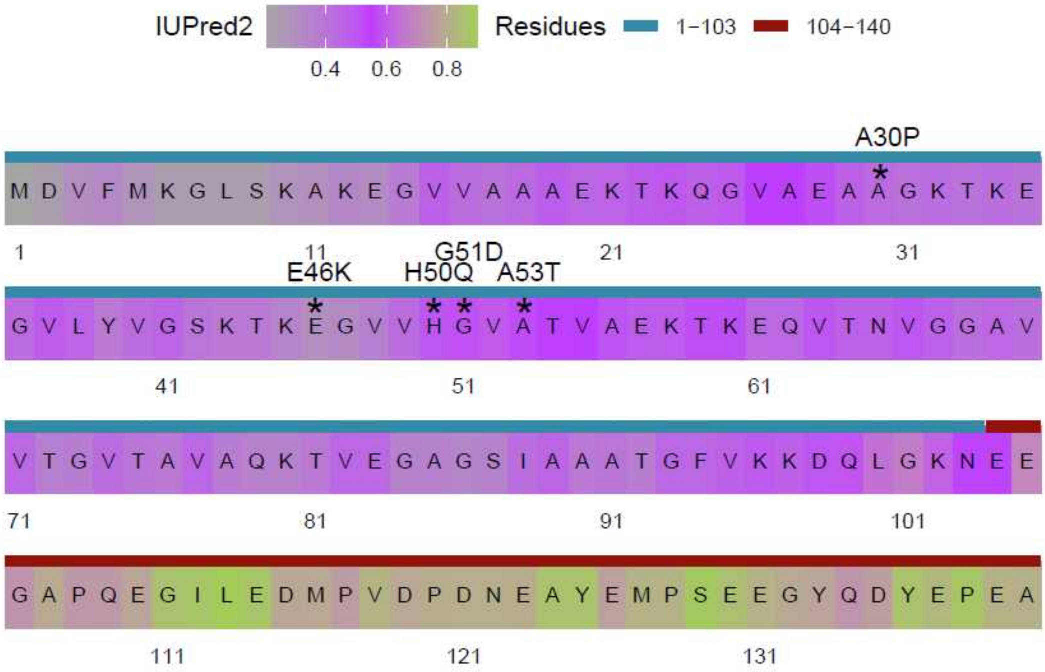
letters. To center residues, the column (x) values were adjusted by 0.5 or 0.35.

```
iupred_map <- iupred_map +
    annotate("text",
             x = c(15.5, 18.5, 30.5, 11.5, 16.5),
             y = c(3.15, 3.15, 4.15, 3.15, 3.3),
             label = c("H50Q", "A53T", "A30P", "E46K",
"G51D")) +
annotate("text",
         x = c(15.35, 18.35, 30.25, 11.35, 16.35),
         y = c(2.825, 2.825, 3.825, 2.825, 2.825),
         label = rep("*", 5),
         size = 7)
```

7.6 Finally, the annotated sequence map is plotted.

```
plot(iupred_map)
```

7.7 Sequence Map of IUPred2 Predictions for aSyn



p53 Figures: Fetching the amino acid sequence

8 First, I will use the UniprotR package to get the p53 amino acid sequence from the UniProtID. For p53,
the ID is P04637.

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

```
## Please wait we are processing your accessions ...
```

8.1 Retrieved Sequence:

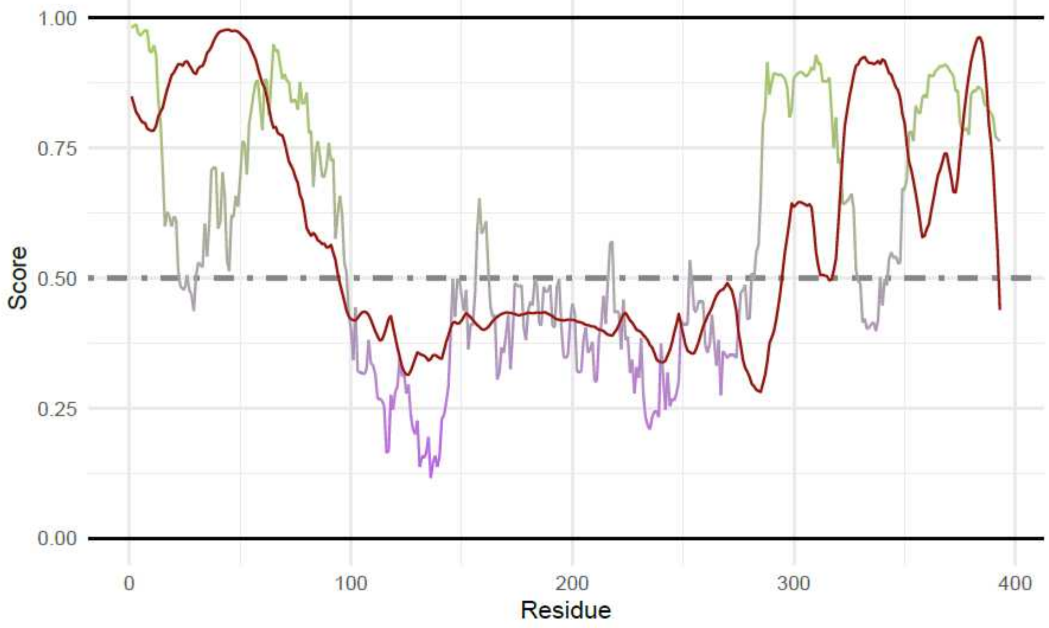
```
print(p53_seq)

## [1]
"MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTED
PGPDEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLH
SGTAKSVTCTYS
```

p53 Figures: Fetching IUPred2A

9 To retrieve the IUPred2 long and ANCHOR2 scores, the p53 uniprot is used.

```
iupredAnchor(uniprotAccession = "P04637",
             proteinName = "p53")
```

9.1 Prediction of Intrinsic Disorder in p53.
By IUPred2A long and ANCHOR2

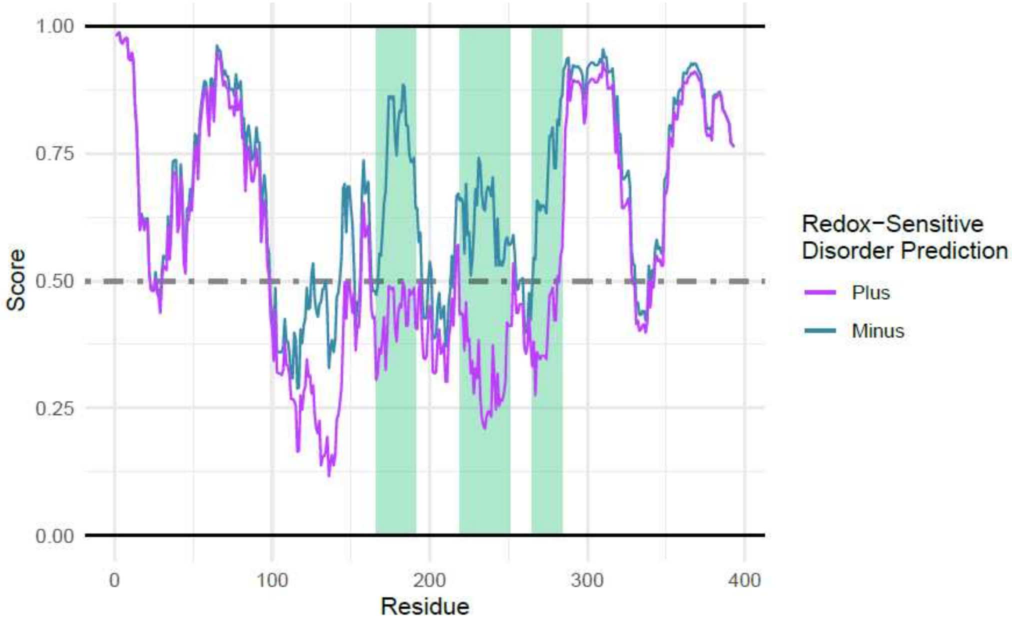


p53 Figures: Fetching IUPred2 Redox

10 To retrieve the IUPred2 with redox predictions, the p53 uniprot is used.

```
iupredRedox(uniprotAccession = "P04637",
            proteinName = "p53")
```

10.1 Prediction of Intrinsic Disorder in p53
By IUPred2 long|Based on Environmental Redox State



p53 Figures: Making the sequenceMap of p53’s sequence structural tendency

11 First, the characteristic to map in the plot must be calculated. Here, the tendency for each residue to favor ordered/disordered structures is determined

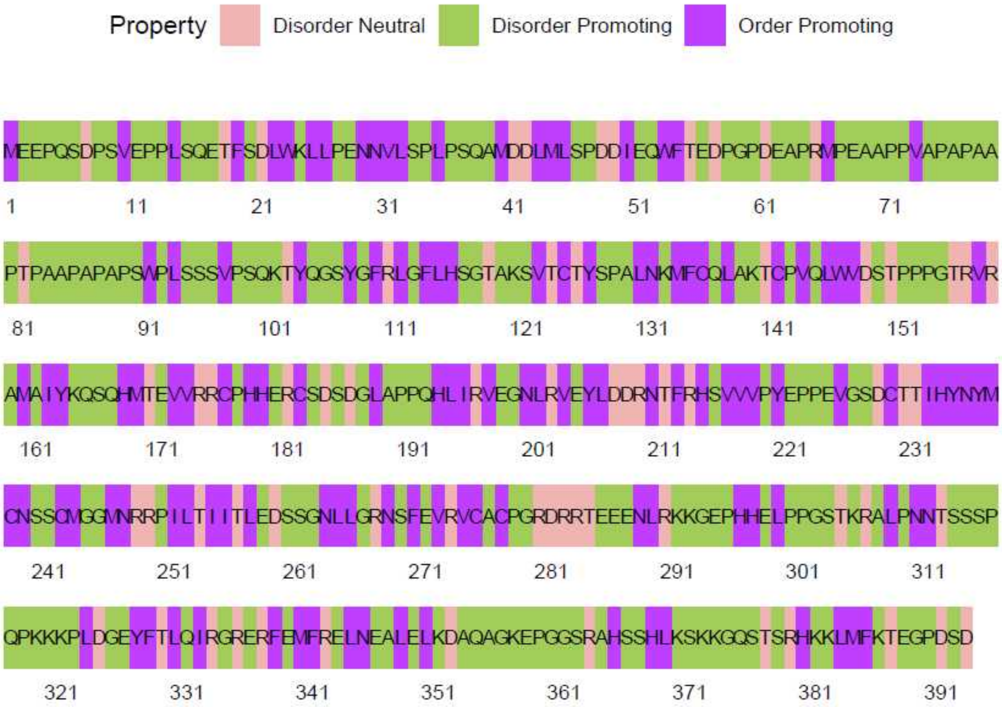
```
tendency_DF <- structuralTendency(p53_seq)
knitr::kable(head(tendency_DF))
```

A	B	C
Position	**AA**	**Tendency**
1	M	Order Promoting
2	E	Disorder Promoting
3	E	Disorder Promoting
4	P	Disorder Promoting
5	Q	Disorder Promoting
6	S	Disorder Promoting

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

11.1 Then, the sequenceMap is made. Since p53 is a long sequence, the nbResidues are increased in the sequenceMap
for easier viewing.

```
tendency_map <-
    sequenceMap(sequence = tendency_DF$AA,
            property = tendency_DF$Tendency,
            nbResidues = 79,
            customColors = c("#F0B5B3", "darkolivegreen3",
"darkorchid1")
            )
plot(tendency_map) #Return the unedited map
```



11.2 To get the coordinates for ggplot annotations, the ‘sequenceMapCoordinates’ function can assist. Since the
default has been changed for nbResidues, from 30 to 79, this must change in the coordinates function to
properly calculate the position of each residue within the sequenceMap.

```
p53_coords <- sequenceMapCoordinates(p53_seq,
                         nbResidues = 79)
knitr::kable(head(p53_coords)) #Top of results to show
example
```

A	B	C	D
Position	**AA**	**row**	**col**
1	M	5	1
2	E	5	2
3	E	5	3
4	P	5	4
5	Q	5	5
6	S	5	6

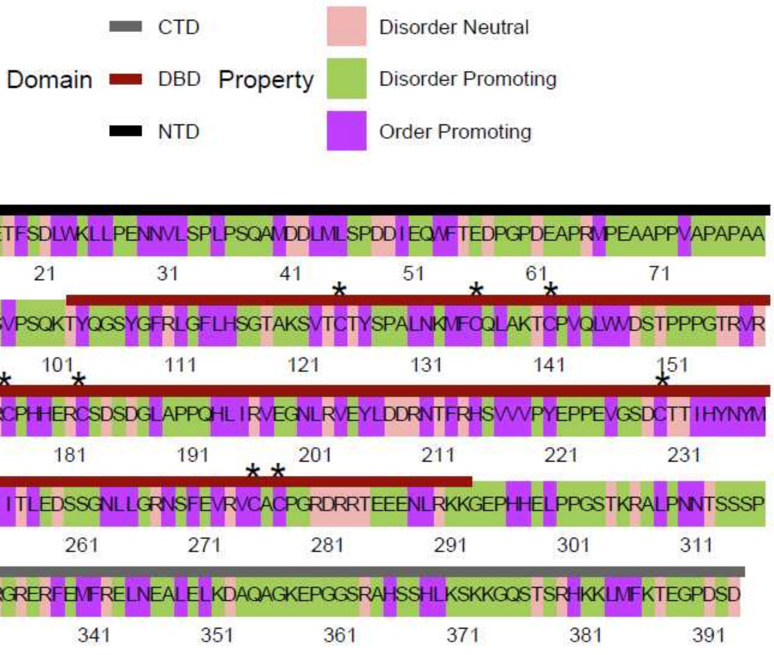
11.3 Additional annotations are made, see the code and the example from Fig. S1B on working with sequenceMap
and annotations.

11.4 After annotations and titles have been added, the plot can be generated.

```
plot(tendency_map)
```

p53 Figures: Making the sequenceMap of p53's sequence structural tendency

11.5 Sequence Map of Residue Tendency for p53.



p53 Figures: Generating the idprofile for p53

12 To get the 'idprofile', a simple function is needed with the sequence and Uniprot ID specified. This generates

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

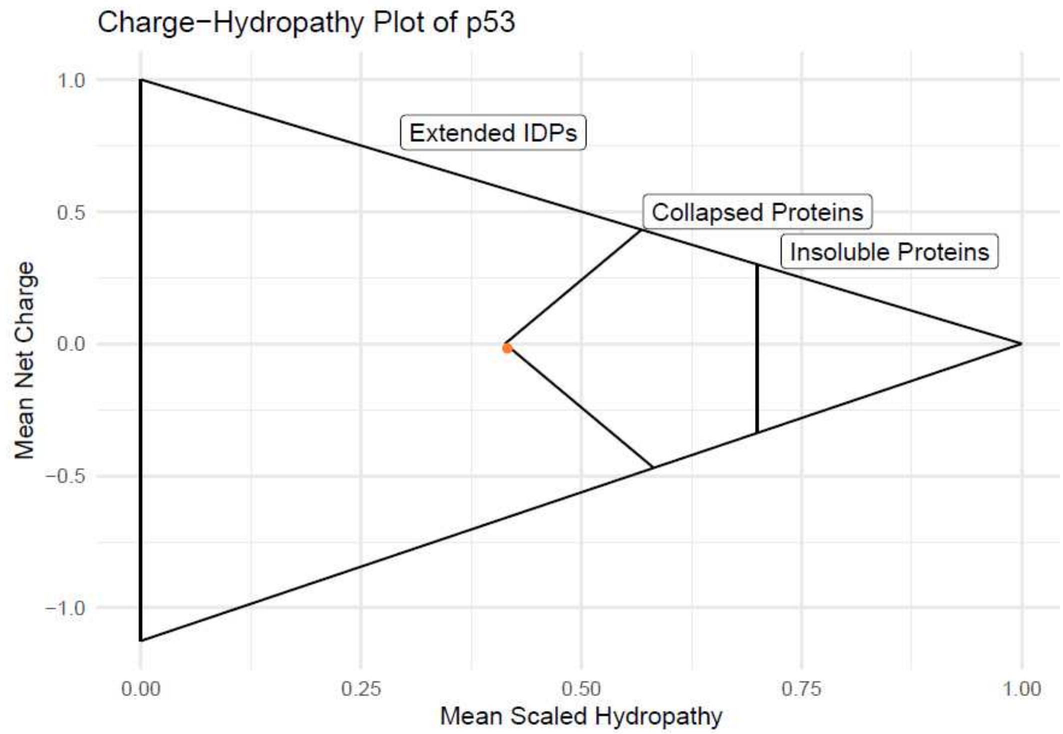
all plots in supplementary figure 2.

```
idprofile(sequence = p53_seq,
          uniprotAccession = "P04637",
          proteinName = "p53") #Specifying proteinName automatically
names plot
```

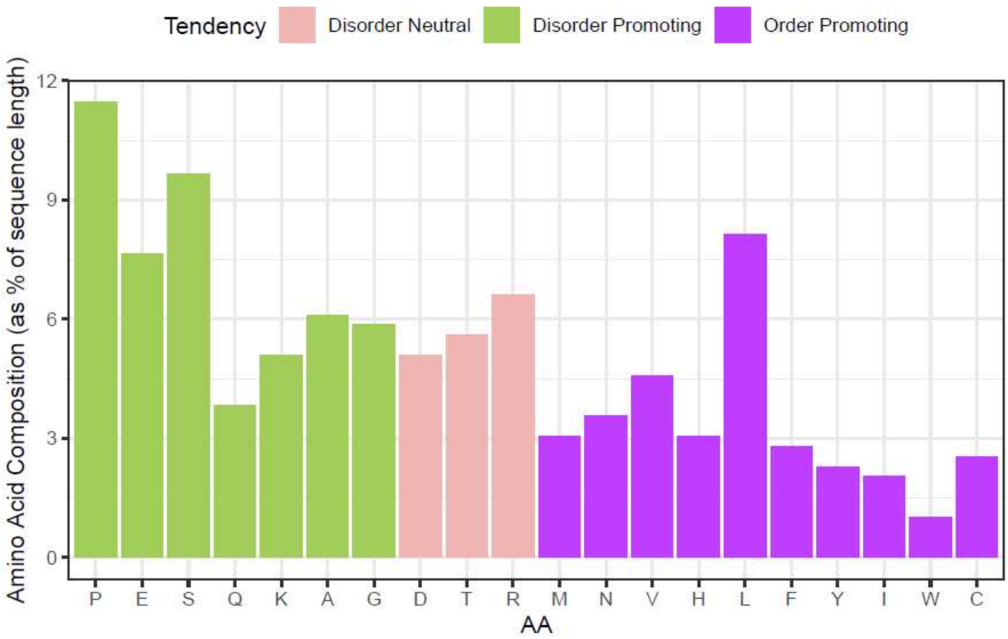
p53 Figures: Generating the idprofile for p53

12.1

```
## [[1]]
```

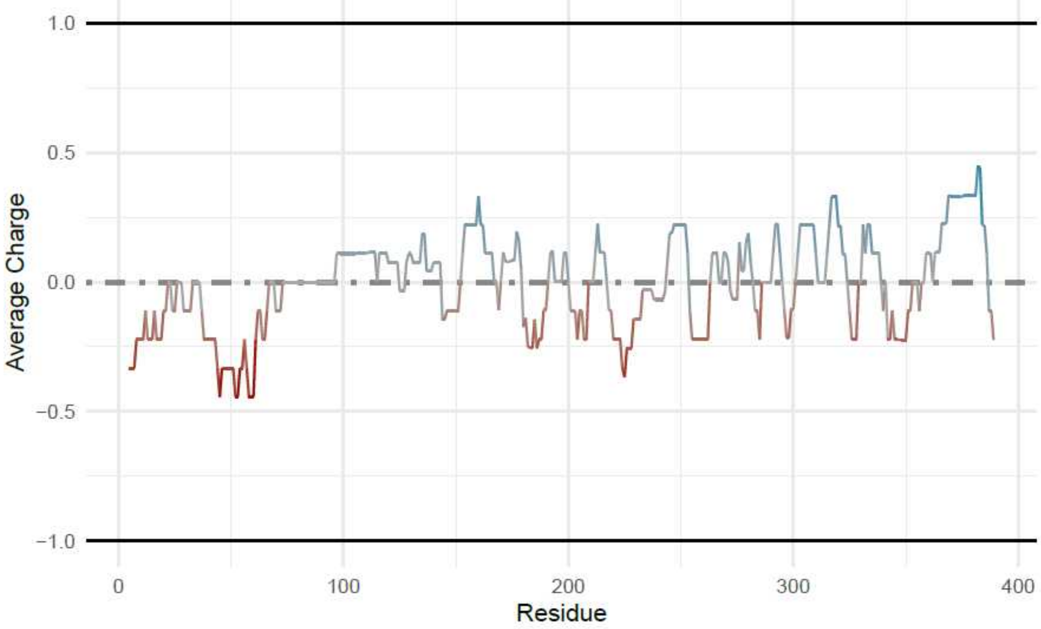


12.2 Compositional Profile of p53



```
##
## [[2]]
```

12.3 Calculation of Local Charge in p53
Window Size = 9 ; Net Charge = −5.774

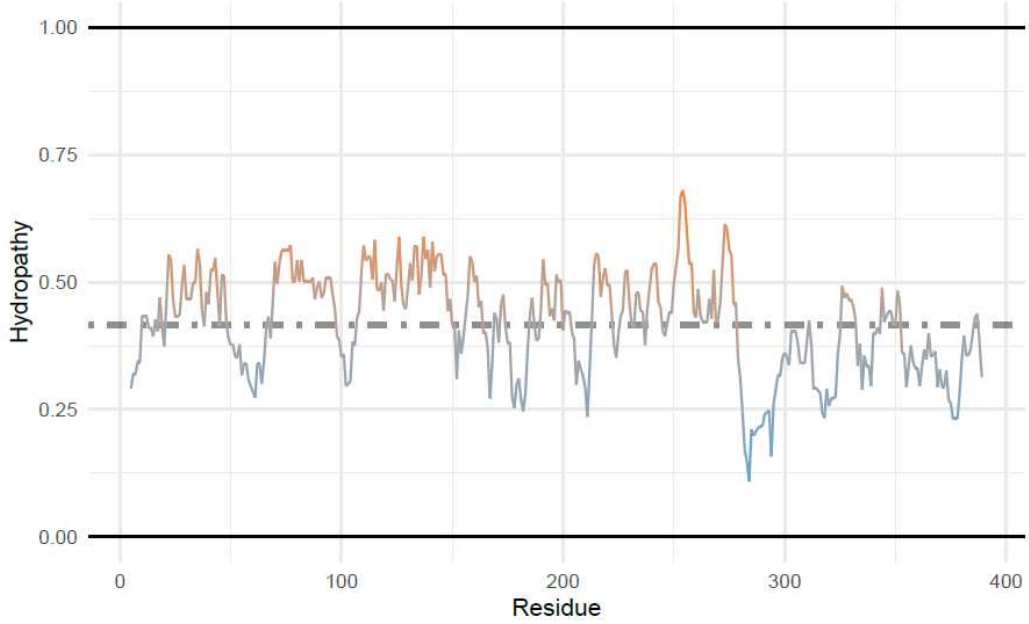


```
##
```

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

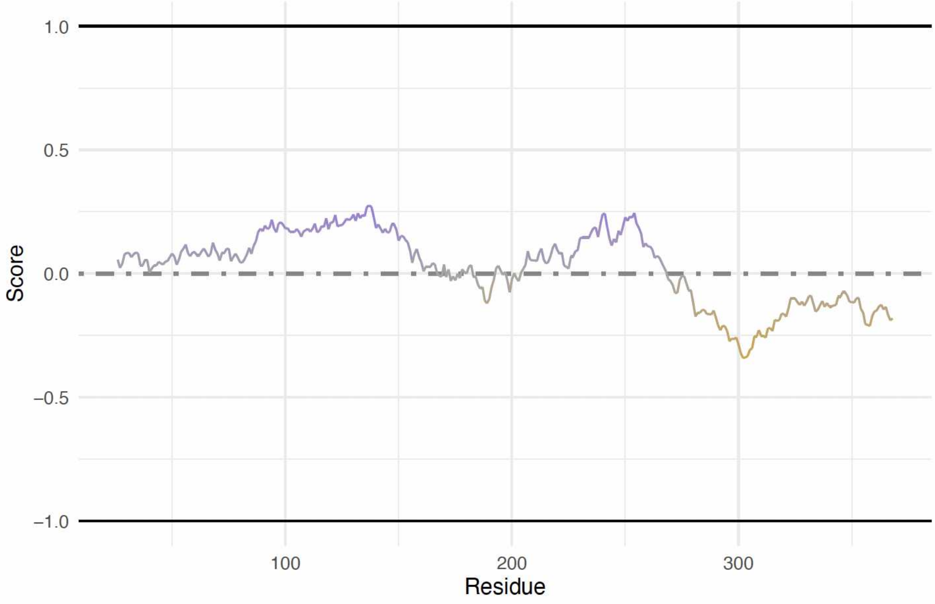
```
## [[3]]
```

12.4 Measurement of Scaled Hydropathy in p53
Window Size = 9 ; Average Scaled Hydropathy = 0.416



```
##
## [[4]]
```

12.5 FoldIndex Prediction of Intrinsic Disorder in p53

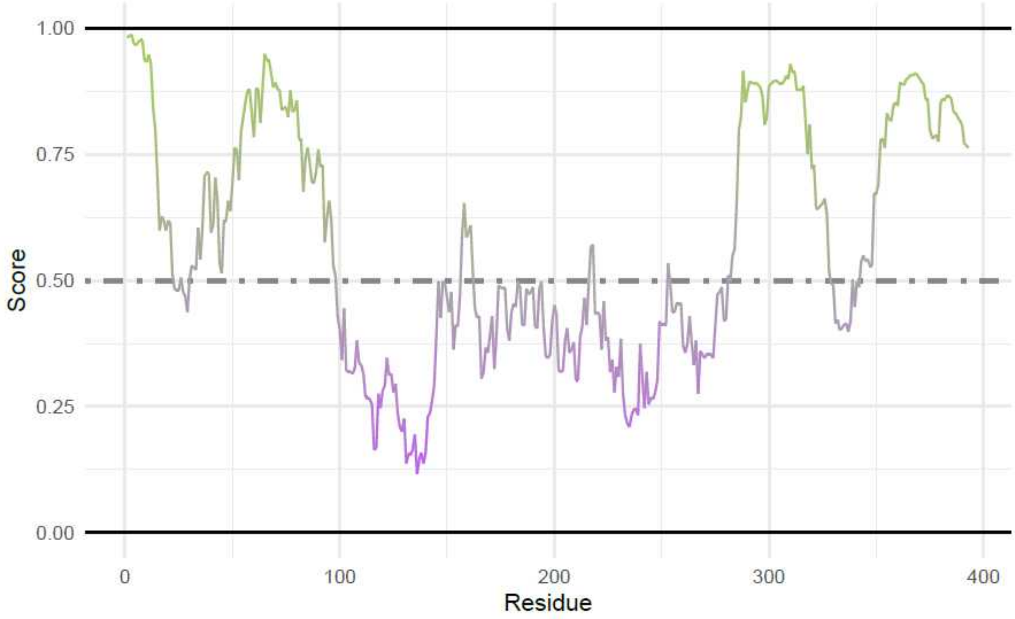


Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw

```
##
## [[5]]
```

12.6

Prediction of Intrinsic Disorder in p53
By IUPred2A long



```
##
## [[6]]
```

Citation: William McFadden, Judith Yanowitz idpr Workflow https://dx.doi.org/10.17504/protocols.io.b58gq9tw