Package 'ProteinGymR'

April 23, 2026

Title Programmatic access to ProteinGym datasets in R/Bioconductor

Version 1.5.1

Description The ProteinGymR package provides analysis-ready data resources from ProteinGym, generated by Notin et al., 2023, as well as built-in functionality to visualize the data. ProteinGym comprises a collection of benchmarks for evaluating the performance of models predicting the effect of point mutations. This package provides access to 1. deep mutational scanning (DMS) scores from 217 assays measuring the impact of all possible amino acid substitutions across 186 proteins, 2. model performance metrics and prediction scores from 79 variant prediction models in the zero-shot setting and 12 models in the semi-supervised setting.

License Artistic-2.0

URL https://github.com/ccb-hms/ProteinGymR

BugReports https://github.com/ccb-hms/ProteinGymR/issues

Depends R (>= 4.4.0)

Imports ExperimentHub, AnnotationHub, dplyr, purrr, queryup, spdl, tidyr, tidyselect, stringr, lifecycle, rlang, htmltools

Suggests tibble, BiocStyle, knitr, testthat (>= 3.0.0), ComplexHeatmap, circlize, pals, ggplot2 (>= 3.5.0), grDevices, ggExtra, bio3d, r3dmol, forcats, ggdist (>= 3.3.0)

VignetteBuilder knitr

Encoding UTF-8

biocViews ExperimentData, ExperimentHub, PackageTypeData, Homo_sapiens_Data, ReproducibleResearch, CellCulture, SequencingData, Proteome

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/ProteinGymR

git_branch devel

git_last_commit f4c9a24

git_last_commit_date 2026-04-07

Repository Bioconductor 3.24

Date/Publication 2026-04-23

Author Tram Nguyen [aut, cre] (ORCID: <https://orcid.org/0000-0003-4809-6227>),
Pascal Notin [aut],
Aaron Kollasch [aut],
Debora Marks [aut],
Ludwig Geistlinger [aut]

Maintainer Tram Nguyen <tram22.nguyen@gmail.com>

Contents

am_scores 2
available_models 3
dms_corr_plot 4
dms_substitutions 6
model_corr_plot 7
plot_dms_heatmap 8
plot_structure 9
plot_zeroshot_heatmap 11
supervised_substitutions 13
zeroshot_DMS_metrics 15

Index **17**

`am_scores` *AlphaMissense scores for ProteinGym variants*

Description

AlphaMissense scores for ProteinGym variants

Usage

```
am_scores(metadata = FALSE)
```

Arguments

`metadata`	Logical, whether only experiment metadata should be returned. Default behavior is to return processed data with metadata included.

Details

`am_scores()` loads in the AlphaMissense pathogencity scores for substitutions matching those in the ProteinGym DMS assays. The table is taken from the AlphaMissense Supplementary Data by Cheng et al. 2023. See reference for details.

The columns contain:

`DMS_id`: Character, ProteinGym assay identifier.

`Uniprot_ID`: Character, UniProt accession identifier.

`variant_id`: Character, variant identifier string matching ProteinGym. Protein position in the middle, and the reference and mutant amino acid residues to the left and right of the position, respectively.

`AlphaMissense`: Numeric, AlphaMissense pathogenicity score.

Value

Returns a `data.frame()`.

References

Cheng et al. (2023) Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science* 391, eadg7492. DOI:10.1126/science.adg7492.

Examples

```
data <- am_scores()
data_meta <- am_scores(metadata = TRUE)
```

`available_models` *Benchmark Variant Effect Prediction Models*

Description

`benchmark_models()` plots one of the five model performance metrics ("AUC", "MCC", "NDCG", "Spearman", "Top_recall") for up to 5 user-specified variant effect prediction tools listed in `available_models()`. See reference for more details about the metrics and models.

Usage

```
available_models()

supervised_available_models()

benchmark_models(
  metric = c("AUC", "MCC", "NDCG", "Spearman", "Top_recall"),
  models = available_models()
)
```

Arguments

`metric`	`character()` a model performance metric to benchmark ("AUC", "MCC", "NDCG", "Spearman", "Top_recall").
`models`	`character()` a character vector of up to five variant effect prediction models to compare. Valid models can be seen with `available_models()`.

Value

`benchmark_models()` returns a `ggplot` object visualizing a chosen model performance metric across several variant effect prediction models, ordered by highest to lowest mean performance score.

References

Notin, P., Kollasch, A., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., & Marks, D. (2023). ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in Neural Information Processing Systems* (Vol. 36, pp. 64331-64379). Curran Associates, Inc.

Examples

```
# Currently support models
available_models()

benchmark_models(metric = "Spearman", models = c("Site_Independent",
"DeepSequence_single", "ESM2_15B", "GEMME", "CARP_640M"))

benchmark_models(models = "GEMME")
```

`dms_corr_plot` *Integrate ProteinGym DMS and Model Prediction Scores*

Description

`dms_corr_plot()` runs a Spearman correlation between ProteinGym deep mutational scanning (DMS) assay scores and predicted model scores. Returns a ggplot object for visualization.

Usage

```
dms_corr_plot(
  uniprotId,
  alphamissense_table = lifecycle::deprecated(),
  dms_table,
  model = "AlphaMissense"
)
```

Arguments

`uniprotId`	`character()` a valid UniProt accession identifier.
`alphamissense_table`	deprecated as of ProteinGymR 1.2.0. To access AlphaMissense model predictions, get `model` argument to "AlphaMissense".
`dms_table`	a table containing deep mutational scanning (DMS) assay scores for mutations. The default table loads substitutions from ProteinGym. Alternatively, a user-defined `tibble::tbl_df` or `data.frame` can be supplied.
`model`	`character()` a model to plot. To view the possible zero-shot and semi-supervised models available in ProteinGym v1.2 run `ProteinGymR::available_models()` or `ProteinGymR::supervised_available_models()`, respectively, or set `model = "AlphaMissens` to access AlphaMissense predictions. If no `model` argument is specified, the default loads in the supplemental table from the AlphaMissense paper.

Details

For `dms_corr_plot()`, `model_table` columns must include:

- `UniProt_id`: UniProt accession identifier.
- `mutant`: Mutant identifier string matching the `dms_table` format. Protein position in the middle, and the reference and mutant amino acid residues to the left and right of the position, respectively.
- `{{model}}`: Predicted model scores. Set this column name as the model name.

`dms_table` columns must include:

- `UniProt_id`: UniProt accession identifier.
- `mutant`: Mutant identifier string matching `model_table` variants. Specifically, the set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').
- `DMS_score`: Experimental measurement in the DMS assay. Higher values indicate higher fitness of the mutated protein.

Value

`dms_corr_plot()` returns a `ggplot` object visualizing the Spearman correlation between experimental DMS scores and predicted model scores and prints the r and p-value of the analysis to console.

References

Cheng et al., Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science* 381, eadg7492. DOI:10.1126/science.adg7492.

Notin, P., Kollasch, A., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., & Marks, D. (2023). ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in Neural Information Processing Systems* (Vol. 36, pp. 64331-64379). Curran Associates, Inc.

Examples

```
# Use defaults. Only requires uniprotId
dms_corr_plot(uniprotId = "Q9NV35")

dms_corr_plot(
    uniprotId = "P04637",
    model = "Kermut"
)
```

dms_substitutions *ProteinGym Deep Mutational Scanning (DMS) Scores for Substitutions*

Description

ProteinGym Deep Mutational Scanning (DMS) Scores for Substitutions

Usage

```
dms_substitutions(metadata = FALSE)
```

Arguments

metadata	Logical, whether only experiment metadata should be returned. Default behavior is to return processed data with metadata included.

Details

`dms_substitutions()` loads in ProteinGym deep mutational scanning assays (DMS) scores for substitutions in 217 studies. The data is provided by Notin et. al 2023. See reference for details.

Each assay includes 6 columns:

`UniProt_id`: Character, UniProt accession identifier.

`DMS_id`: Character, ProteinGym assay identifier.

`mutant`: Character, set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').

`mutated_sequence`: Character, full amino acid sequence for the mutated protein.

`DMS_score`: Numeric, experimental measurement in the DMS assay. Higher values indicate higher fitness of the mutated protein.

`DMS_score_bin`: Factor, indicates whether the DMS_score is above the fitness cutoff (1 is fit, 0 is not fit).

Value

Returns a `list()` object of 217 individual assays.

References

Notin, P., Kollasch, A., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., & Marks, D. (2023). ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), Advances in Neural Information Processing Systems (Vol. 36, pp. 64331-64379). Curran Associates, Inc.

Examples

```
data <- dms_substitutions()
data_meta <- dms_substitutions(metadata = TRUE)
```

model_corr_plot *Compare Prediction Scores for ProteinGym Models*

Description

model_corr_plot() runs a Spearman correlation between predicted model scores for two models in ProteinGym v1.2. Returns a ggplot object for visualization.

Usage

```
model_corr_plot(uniprotId, model1 = "AlphaMissense", model2 = "GEMME")
```

Arguments

uniprotId	character() a valid UniProt accession identifier.
model1	character() first model to plot.
model2	character() second model to plot.

Details

For model_corr_plot():

model1 and model2 must be valid models. To view the possible zero-shot and semi-supervised models available in ProteinGym v1.2 run ProteinGymR::available_models() or ProteinGymR::supervised_available_
respectively, or set the model to 'AlphaMissense' for AlphaMissense predictions. If no models are specified, the default loads in AlphaMissense and GEMME.

Value

model_corr_plot() returns a ggplot object visualizing the Spearman correlation between the predicted scores generated by two models in ProteinGym and prints the p-value of the analysis to the console.

References

Cheng et al., Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science* 381, eadg7492. DOI:10.1126/science.adg7492.

Notin, P., Kollasch, A., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., & Marks, D. (2023). ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in Neural Information Processing Systems* (Vol. 36, pp. 64331-64379). Curran Associates, Inc.

Examples

```
# Use defaults. Only requires uniprotId
model_corr_plot(uniprotId = "Q9NV35")

model_corr_plot(
    uniprotId = "P04637",
    model1 = "Kermut",
    model2 = "EVE_single"
```

```
)
```

`plot_dms_heatmap` *Visualize DMS Scores Along a Protein*

Description

`plot_dms_heatmap()` plots DMS scores for amino acid substitutions along a protein in a defined DMS assay.

Usage

```
plot_dms_heatmap(
  assay_name,
  dms_data,
  start_pos = NULL,
  end_pos = NULL,
  exact_coord = FALSE,
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  color_scheme,
  ...
)
```

Arguments

`assay_name`	`character()` a valid DMS assay name. For the full list of available assays, run `names()` on the list object loaded with `ProteinGymR::dms_substitutions()`. Alternatively, the name of a user-defined DMS assay.
`dms_data`	`list()` object of DMS assays loaded with `ProteinGymR::dms_substitutions()` if not specified by user. Alternatively, a user-defined list of DMS assays with names corresponding to `assay_name` param.
`start_pos`	`integer()` first amino acid position to plot. If missing, default start is at the first position along the protein where DMS scores are available.
`end_pos`	`integer()` last amino acid position to plot. If missing, default end is at the last position along the protein where DMS scores are available.
`exact_coord`	`logical()` TRUE will plot the precise `start_pos` and `end_pos` coordinates defined. By default, `exact_coord` is set to FALSE, plotting only amino acid positions with available data in the chosen assay.
`cluster_rows`	`logical()` defaults to FALSE. See argument details in ComplexHeatmap::Heatmap.
`cluster_columns`	`logical()` defaults to FALSE. See argument details in ComplexHeatmap::Heatmap.
`color_scheme`	`character()` defaults to blue, white, and red to represent positive, neutral, negative scores. Set argument equal to "EVE" to use the color scheme consistent with the popEVE portal.
`...`	additional arguments passed to internal plotting functions.

Details

For `plot_dms_heatmap()`, `dms_data` must be a `list()` object with set names for each assay element matching `assay_name` parameter.

Each assay in the `dms_data()` must include the following columns:

- `mutant`: Mutant identifier string matching. Specifically, the set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').
- `DMS_score`: Experimental measurement in the DMS assay. Higher values indicate higher fitness of the mutated protein.

Value

Returns a ComplexHeatmap::Heatmap plot of DMS scores for each position along a protein in a chosen DMS assay. The x-axis shows amino acid positions where a DMS mutation exist, and the y-axis represents possible amino acid residues, ordered by default based on the physiochemical groupings. Higher and lower DMS scores indicate a more positive or negative fitness effect after the mutation, respectively.

Examples

```
dms_data <- dms_substitutions()

plot_dms_heatmap(assay_name = "A0A192B1T2_9HIV1_Haddox_2018",
    start_pos = 10,
    end_pos = 80)

plot_dms_heatmap(assay_name = "A0A192B1T2_9HIV1_Haddox_2018",
    start_pos = 10,
    end_pos = 80,
    exact_coord = TRUE,
    color_scheme = "EVE")

plot_dms_heatmap(assay_name = "A0A192B1T2_9HIV1_Haddox_2018",
    start_pos = 50,
    end_pos = 100,
    cluster_rows = TRUE)
```

---

plot_structure *Visualize DMS and Model Scores on 3D Protein Structures*

Description

`plot_structure()` plots DMS or model scores for amino acid substitutions on a 3D protein structure for a chosen assay.

Usage

```
plot_structure(
  assay_name,
  pdb_file,
  data_scores = "DMS",
  dms_data = NULL,
  start_pos = NULL,
  end_pos = NULL,
  full_structure = FALSE,
  aggregate_fun = mean,
  color_scheme = NULL
)
```

Arguments

assay_name	character() a valid DMS assay name. For the full list of available assays, run names() on the list object loaded with ProteinGymR::dms_substitutions(). Alternatively, the name of a
pdb_file	string() defaults to corresonding PDB FilePath on ExperimentHub. Alternatively, a file path to a user-defined PDB file.
data_scores	character() specify whether DMS, zero-shot, or supervised model prediction scores should be displayed scores. Pass either "DMS" for experimental scores, or alternatively, a model name from available_models() for zero-shot or supervised_available_models() for semi-supervised models options. Defaults to DMS.
dms_data	list() object of DMS assays loaded with ProteinGymR::dms_substitutions(). Alternatively, a user-defined list of DMS assays with names corresponding to assay_name param.
start_pos	integer() first amino acid position to plot. If missing, default start is the first position along the protein in the PDB file.
end_pos	integer() last amino acid position to plot. If missing, default end is the last position along the protein in the PDB file.
full_structure	logical() defaults to FALSE and will only plot protein regions where there is DMS data available in the assay. If start_pos and end_pos coordinates are specified, plotting is restricted to this defined region. Setting full_structure() to TRUE will display full protein structure in the PBD file, and grey out regions where no DMS data is available.
aggregate_fun	method for aggregating DMS scores for each residue. For example, give min, max, or var to return the minimum, maximum, or variance of scores for each position, respectively. aggregate_fun can also take in a user-defined function with a numeric vector as input. By default, the mean DMS score across mutations at each position is calculated.
color_scheme	character() defaults to blue, white, and red to represent positive, neutral, negative scores. Set argument equal to "EVE" to use the color scheme consistent with the popEVE portal.

Details

By default, plot_structure() plots the mean DMS values of all amino acid residues, summarized for a protein position. If a model is chosen instead for data_scores argument, a helper function is

invoked which normalizes the model prediction scores using a rank-based normal quantile transformation. The result is a set of normalized scores that preserve the rank order of the models scores, while standardizing the distribution. Transformed values typically fall between -3 and 3. This normalization ensures the scores are approximately standard normally distributed (mean = 0, SD = 1), allowing comparisons across models.

For `plot_structure()`, `dms_data` must be a `list()` object with set names for each assay element matching `assay_name` parameter.

Each assay in the `dms_data()` must include the following columns:

- `mutant`: Mutant identifier string matching. Specifically, the set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').
- `DMS_score`: Experimental measurement in the DMS assay. Higher values indicate higher fitness of the mutated protein.

Each PBD table in `pdb_file` must include the following columns:

Value

`plot_structure()` returns a `r3dmol::r3dmol` object of DMS scores for each position along a protein in a chosen DMS assay. The x-axis shows amino acid positions where a DMS mutation exist, and the y-axis represents possible amino acid residues, ordered by default based on the physiochemical groupings. Higher and lower DMS scores indicate a more positive or negative fitness effect after the mutation, respectively.

Examples

```
plot_structure(assay_name = "C6KNH7_9INFA_Lee_2018",
   start_pos = 20,
   end_pos = 50,
   full_structure = FALSE,
   aggregate_fun = max)

plot_structure(assay_name = "C6KNH7_9INFA_Lee_2018",
   start_pos = 20,
   end_pos = 50,
   data_scores = "GEMME")

plot_structure(assay_name = "ACE2_HUMAN_Chan_2020",
    data_scores = "Kermut",
    color_scheme = "EVE")
```

---

plot_zeroshot_heatmap *Visualize Zero Shot Scores Along a Protein*

Description

`plot_zeroshot_heatmap()` plots predicted model scores under the zero-shot model for amino acid substitutions along a protein in a defined DMS assay.

Usage

```
plot_zeroshot_heatmap(
  assay_name,
  model_data,
  model = NULL,
  start_pos = NULL,
  end_pos = NULL,
  exact_coord = FALSE,
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  color_scheme,
  ...
)
```

Arguments

assay_name	character() a valid assay name. For the full list of available assays, run names() on the list object loaded with ProteinGymR::zeroshot_substitutions(). Alternatively, the name of a user-defined DMS assay.
model_data	list() object of zero-shot assays loaded with ProteinGymR::zeroshot_substitutions(). Alternatively, a user-defined list of assays with names corresponding to assay_name param.
model	character() one of the 79 zero-shot models to plot. To view the list of models, runs ProteinGymR::available_models().
start_pos	integer() first amino acid position to plot. If missing, default start is at the first position along the protein where zero shot scores are available.
end_pos	integer() last amino acid position to plot. If missing, default end is at the last position along the protein where zero shot scores are available.
exact_coord	logical() TRUE will plot the precise start_pos and end_pos coordinates defined. By default, exact_coord is set to FALSE, plotting only amino acid positions with available data in the chosen assay.
cluster_rows	logical() defaults to FALSE. See argument details in ComplexHeatmap::Heatmap.
cluster_columns	logical() defaults to FALSE. See argument details in ComplexHeatmap::Heatmap.
color_scheme	character() defaults to blue, white, and red to represent positive, neutral, negative scores. Set argument equal to "EVE" to use the color scheme consistent with the popEVE portal.
...	additional arguments passed to internal plotting functions.

Details

For plot_zeroshot_heatmap(), model_data must be a list() object with set names for each assay element matching assay_name parameter.

Each assay in the model_data() must include the following columns:

- mutant: Mutant identifier string matching. Specifically, the set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').
- A column with predicted model scores for each substitution. Column name should match model argument string.

Value

Returns a ComplexHeatmap::Heatmap plot of zero-shot model scores for each position along a protein in a chosen DMS substitution assay. The x-axis shows amino acid positions where a DMS mutation exist, and the y-axis represents possible amino acid residues, ordered by default based on the physiochemical groupings. Higher and lower DMS scores indicate a more positive or negative fitness effect after the mutation, respectively.

Examples

```
available_models()

plot_zeroshot_heatmap(assay_name = "A0A192B1T2_9HIV1_Haddox_2018",
    model = "GEMME",
    start_pos = 600,
    end_pos = 700,
    color_scheme = "EVE")

plot_zeroshot_heatmap(assay_name = "SRC_HUMAN_Nguyen_2022",
    model = "CARP_38M")
```

---

supervised_substitutions

Load Semi-Supervised Model Predictions for Substitutions in 217 Assays

---

Description

Load Semi-Supervised Model Predictions for Substitutions in 217 Assays

Load Semi-Supervised Model Summary Metrics

Usage

```
supervised_substitutions(
  metadata = FALSE,
  fold_scheme = c("contiguous", "modulo", "random")
)

supervised_metrics(metadata = FALSE)
```

Arguments

metadata	Logical, whether only experiment metadata should be returned. Default behavior is to return processed data with metadata included.
fold_scheme	Character, which validation folding scheme to load. Options include: "contiguous", "modulo", or "random". Default behavior loads "contiguous". For more information about the different folding schemes, refer to the original publication.

Details

`supervised_substitutions()` loads prediction scores outputted by semi-supervised models run on the 217 DMS substitution assays.

For raw model predictions, each assay includes 18 columns:

`UniProt_id`: Character, UniProt accession identifier.

`DMS_id`: Character, ProteinGym assay identifier.

`mutant`: Character, set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').

`mutated_sequence`: Character, full amino acid sequence for the mutated protein.

`DMS_score`: Numeric, experimental measurement in the DMS assay. Higher values indicate higher fitness of the mutated protein.

`DMS_score_bin`: Factor, indicates whether the DMS_score is above the fitness cutoff (1 is fit, 0 is not fit).

`Columns 7:18`: Respective semi-supervised model name.

`supervised_metrics()` loads in model performance summary metrics ("Spearman" and "MSE") from semi-supervised models in ProteinGymR run on 217 DMS substitution assays.

A metric summary table with 7 columns:

`UniProt_id`: Character, UniProt accession identifier.

`DMS_id`: Character, ProteinGym assay identifier.

`mutant`: Character, set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').

`model_name`: Character, semi-supervised model used.

`fold_variable_name`: Character, the folding scheme used.

`Spearman`: Numeric, Spearman performance metric.

`MSE`: Numeric, MSE the Spearman performance metric.

Value

Returns a `list()` object of 217 individual assays.

Returns a `data.frame()` with 7 columns.

References

Notin, P., Kollasch, A., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., & Marks, D. (2023). ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), Advances in Neural Information Processing Systems (Vol. 36, pp. 64331-64379). Curran Associates, Inc.

Examples

```
data <- supervised_substitutions()
data_random <- supervised_substitutions(fold_scheme = "random")
meta <- supervised_substitutions(metadata = TRUE)

data <- supervised_metrics()
meta <- supervised_metrics(metadata = TRUE)
```

---

zeroshot_DMS_metrics *Load Zero-shot Model Predictions and Metrics for Substitutions in 217 DMS Assays*

Description

Load Zero-shot Model Predictions and Metrics for Substitutions in 217 DMS Assays

Usage

```
zeroshot_DMS_metrics(metadata = FALSE)

zeroshot_substitutions(metadata = FALSE)
```

Arguments

metadata	Logical, whether only experiment metadata should be returned. Default behavior is to return processed data with metadata included.

Details

`zeroshot_DMS_metrics()` loads in the model performance metrics ("AUC", "MCC", "NDCG", "Spearman", "Top_recall") calculated on the DMS substitutions in the zero-shot setting for 79 models updated in ProteinGym v1.2.

Each data.frame contains the following columns:

`DMS_ID`: Character, Assay name for the DMS study.

`Columns 2:80`: Numeric, Corresponding to the average performance score of each of the 79 models tested.

`Number_of_Mutants`: Numeric, Number of protein mutants evaluated.

`Selection_Type`: Character, Protein function grouping.

`UniProt_ID`: Character, UniProt protein entry name identifier.

`DMS_score`: Numeric, experimental measurement in the DMS assay. Higher values indicate higher fitness of the mutated protein.

`DMS_score_bin`: Factor, indicates whether the DMS_score is above the fitness cutoff (1 is fit, 0 is not fit).

`MSA_Neff_L_category`: Multiple sequence alignment category.

`Taxon`: Taxon group.

`zeroshot_substitutions()` loads prediction scores outputted by models in the zero-shot setting evaluated on the 217 DMS substitution assays. To examine all model options, run `available_models()`.

For raw model predictions, each data.frame includes 85 columns:

`UniProt_id`**:** Character, UniProt accession identifier.

`DMS_id`**:** Character, ProteinGym assay identifier.

`mutant`**:** Character, set of substitutions to apply on the reference sequence to obtain the mutated sequence (e.g., A1P:D2N implies the amino acid 'A' at position 1 should be replaced by 'P', and 'D' at position 2 should be replaced by 'N').

`mutated_sequence`**:** Character, full amino acid sequence for the mutated protein.

`DMS_score`**:** Numeric, experimental measurement in the DMS assay. Higher values indicate higher fitness of the mutated protein.

`DMS_score_bin`**:** Factor, indicates whether the DMS_score is above the fitness cutoff (1 is fit, 0 is not fit).

`Columns 7:85`**:** Respective zero-shot model name.

Value

Returns a `list()` object with 5 `data.frame()` corresponding to a model metric table.

Returns a `list()` object of 217 individual assays.

References

Notin, P., Kollasch, A., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., & Marks, D. (2023). ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), Advances in Neural Information Processing Systems (Vol. 36, pp. 64331-64379). Curran Associates, Inc.

Examples

```
data <- zeroshot_DMS_metrics()
data_meta <- zeroshot_DMS_metrics(metadata = TRUE)

data <- zeroshot_substitutions()
data_meta <- zeroshot_substitutions(metadata = TRUE)
```

Index

am_scores, 2
available_models, 3

benchmark_models *(*available_models*)*, 3

ComplexHeatmap::Heatmap, *8, 9*, *12, 13*

data.frame, *4*
data.frame(), *3*, *14*, *16*
dms_corr_plot, 4
dms_substitutions, 6

list(), *6*, *14*, *16*

max, *10*
min, *10*
model_corr_plot, 7

plot_dms_heatmap, 8
plot_structure, 9
plot_zeroshot_heatmap, 11

r3dmol::r3dmol, *11*

supervised_available_models
        *(*available_models*)*, 3
supervised_metrics
        *(*supervised_substitutions*)*, 13
supervised_substitutions, 13

tibble::tbl_df, *4*

var, *10*

zeroshot_DMS_metrics, 15
zeroshot_substitutions
        *(*zeroshot_DMS_metrics*)*, 15