Package: CRUX (via r-universe)

September 12, 2024

Title Easily explore patterns of somatic variation in cancer using 'CRUX'

Version 0.0.0.9000

Description Shiny app for exploring somatic variation in cancer. Powered by maftools.

License GPL-3 + file LICENSE

Imports dplyr (>= 1.0.0), config (>= 0.3), golem, data.table, cowplot, forcats, ggplot2, rlang, shinyWidgets, shinyjs, tibble, tidyr, shiny (>= 1.5.0), DT (>= 0.16), magrittr (>= 2.0.1), purrr (>= 0.3.4), assertthat (>= 0.2.1), maftools (>= 2.6.05), bsplus (>= 0.1.2), shinyBS (>= 0.61), shinycssloaders (>= 1.0.0), utilitybeltassertions, utilitybeltgg, colourpicker, UpSetR, ggvenn, shinydashboard, somaticflags (>= 0.0.0.9000), mutaliskRutils (>= 0.0.0.9000), PCAWGmutations, methods, TCGAgistic (>= 0.0.0.9000), assertions, cli, htmltools, curl, shinybusy

Remotes selkamand/utilitybeltassertions, selkamand/utilitybeltgg, ebailey78/shinyBS, CCICB/somaticflags, CCICB/PCAWGmutations, selkamand/mutaliskRutils, CCICB/TCGAgistic, PoisonAlien/maftools@72208f2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

URL https://github.com/CCICB/CRUX

BugReports https://github.com/CCICB/CRUX/issues

Depends R (>= 3.5)

Suggests mclust, pheatmap, R.utils, testthat, usethis

Collate 'CRUX-package.R' 'GLOBALS.R' 'app_config.R' 'fct_external_tools_load_dataframe.R' 'app_server.R' 'app_ui.R' 'class_maf_data_pool.R' 'class_maf_dataset_wrapper.R' 'data.R' 'data_loading_utils.R' 'data_wrapping_pcawg.R'

'data_wrapping_tcga.R' 'data_wrapping_user_imports.R'
'fct_TCGAmutations_load_with_typed_metadata.R'
'fct_centered_down_arrow.R' 'fct_download_maf.R'
'fct_filetype_guesser.R' 'fct_gistic.R'
'fct_maftools_extending_functions.R'
'fct_maftools_fix_clinical_data_types.R' 'golem_utils_server.R'
'golem_utils_ui.R' 'helper_functions.R' 'mod_cnv.R'
'mod_compare_cohorts.R' 'mod_data_import.R'
'mod_data_import_maf_path_to_maf.R' 'mod_data_import_step2.R'
'mod_data_pool_table.R' 'mod_download_datatable.R'
'mod_download_maf.R' 'mod_download_plot.R' 'mod_druggability.R'
'mod_edit_dataset.R' 'mod_enrichment.R' 'mod_exclude_samples.R'
'mod_expression_analysis.R' 'mod_expression_import.R'
'mod_external_tools.R' 'mod_help.R' 'mod_home.R'
'mod_import_clinical_featurefile.R' 'mod_import_maf.R'
'mod_introduce_app.R' 'mod_is_tcga_checkbox.R'
'mod_load_tcga.R' 'mod_merge.R'
'mod_mod_select_maf_datasets_wrapper.R'
'mod_mutational_signatures.R' 'mod_pan_cohort_statistics.R'
'mod_plot_apobec_diff.R' 'mod_plot_cobarplot.R'
'mod_plot_complete_pathway.R' 'mod_plot_cooncoplots.R'
'mod_plot_druginteractions.R' 'mod_plot_gistic_genome.R'
'mod_plot_gistic_oncoplot.R' 'mod_plot_heterogeneity.R'
'mod_plot_lollipop.R' 'mod_plot_lollipop2.R'
'mod_plot_mutational_signatures.R'
'mod_plot_oncogenic_pathways.R' 'mod_plot_oncoplots.R'
'mod_plot_pfam_domains.R' 'mod_plot_rainfall.R'
'mod_plot_rainforest.R' 'mod_plot_somatic_interactions.R'
'mod_plot_summary.R' 'mod_plot_survival.R'
'mod_plot_tcgacompare.R' 'mod_plot_titv.R' 'mod_plot_tsne.R'
'mod_privacy.R' 'mod_render_clinical_data_table.R'
'mod_render_downloadable_dataframe.R'
'mod_sample_level_analysis.R'
'mod_select_column_from_dataframe.R'
'mod_select_dataset_from_maf_data_pool_pickerinput_and_return_maf_dataset_wrapper.R'
'mod_select_genes.R' 'mod_select_maf_column.R'
'mod_select_maf_column_value.R'
'mod_select_tumor_sample_barcode_from_maf.R'
'mod_select_value_from_vector.R' 'mod_shinyfile_import.R'
'mod_shinyfiles_get_clinical_featurefile_path.R'
'mod_shinyfiles_get_maf_path.R' 'mod_single_cohort_summary.R'
'mod_split_maf_by_clinical_feature.R' 'mod_survival_analysis.R'
'mod_tabulate_maf.R' 'mod_utilities.R' 'mod_utility_subset.R'
'run_app.R' 'update_starting_maf_data_pool.R' 'utils-pipe.R'
'utils_bool_to_colour.R' 'utils_error_messages.R'
'utils_installation.R' 'utils_links.R' 'utils_spacer.R'
'utils_text_is_valid_input.R'
'utils_validating_clinical_feature_files.R'

'visualise_data_pool.R'

Repository https://ccicb.r-universe.dev

RemoteUrl https://github.com/CCICB/CRUX

RemoteRef HEAD

RemoteSha a75a0718d07e81444f4be2a7750d9e9e30e45816

Contents

`annovar_guess_reference_table` *Guess reference table used for annovar annotation*

Description

Reads the first line of a file and confirms that it is an ANNOVAR file, then returns 'ensGene' or 'refGene' base on whether refseq or ensemble reference tables were for gene-based annotations

Usage

```
annovar_guess_reference_table(path)
```

Arguments

`path`	path to file

Value

one of MAF, ANNOVAR, VCF, OTHER

`assert_rnaseq_df_is_formatted_correctly` *Check RNAseq Dataframe*

Description

Runs a bunch of assertions on the supplied rnaseq dataframe to determine whether it is valid. Will error if it is not.

Usage

```
assert_rnaseq_df_is_formatted_correctly(rnaseq_df)
```

Arguments

`rnaseq_df`	Dataframe containing at least three columns, named "Tumor_Sample_Barcode", "Hugo_Symbol" and "TPM". May optionally include columns named "Fold_Change" and "Transcript" (dataframe)

Value

Nothing, run for its side effects

`assert_that_class_is_maf_dataset_wrapper` *Assert that class is maf_data_pool*

Description

Assert that class is maf_data_pool

Usage

```
assert_that_class_is_maf_dataset_wrapper(object)
```

Arguments

`object`	some object whose class you want to assert (anything)

Value

(invisible) TRUE if assertion succeeds, Throws an error if it doesn’t

See Also

Other class_assertions: `assert_that_class_is_maf_data_pool()`

`assert_that_class_is_maf_data_pool` *Assert that class is maf_data_pool*

Description

Assert that class is maf_data_pool object

Usage

```
assert_that_class_is_maf_data_pool(object)
```

Arguments

`object`	some object whose class you want to assert (anything)

Value

(invisible) TRUE if assertion succeeds, Throws an error if it doesn't

See Also

Other class_assertions: `assert_that_class_is_maf_dataset_wrapper()`

`badgify` *Turn Text into A Badge*

Description

Turn Text into A Badge

Usage

```
badgify(
  vec,
 fontweight = c("bold", "bolder", "normal", "inherit", "initial", "lighter", "unset"),
 fontsize = c("xx-small", "x-small", "small", "medium", "large", "unset", "inherit"),
  color_background = "#d2d6de",
  color_text = "#444"
)
```

Arguments

`vec`	a vector of terms to make into a badge
`fontweight`	fontweight of badge text
`fontsize`	fontsize of badge text
`color_background`	background colour. Can be a single value or a vector the same length as vec.
`color_text`	text colour. Can be a single value or a vector the same length as vec.

Value

string with html encoding for the badge

`bool_to_colour` *Map logical to colour*

Description

Hex codes

Usage

```
bool_to_colour(
  boolean,
  colour_if_true = "#69C776",
  colour_if_false = "#ED7676"
)
```

Arguments

`boolean`	TRUE/FALSE (boolean)
`colour_if_true`	colour to return if true (string)
`colour_if_false`	colour to return if false (string)

Value

colour code (string)

Examples

```
CRUX:::bool_to_colour(1==1)
```

class_is_maf_dataset_wrapper

Check if class is a maf_dataset_wrapper

Description

Check if class is a maf_dataset_wrapper object

Usage

```
class_is_maf_dataset_wrapper(object)
```

Arguments

`object` some object whose class you want to check

Value

TRUE if class of object is maf_dataset_wrapper, FALSE if not. (logical)

Examples

```
CRUX:::class_is_maf_dataset_wrapper("Hi")
```

class_is_maf_data_pool

Check if class is a maf_data_pool

Description

Check if class is a maf_data_pool object

Usage

```
class_is_maf_data_pool(object)
```

Arguments

`object` some object whose class you want to check

Value

TRUE if class of object is maf_data_pool, FALSE if not. (logical)

Examples

```
CRUX:::class_is_maf_data_pool("Hi")
CRUX:::class_is_maf_data_pool(CRUX:::new_maf_data_pool())
```

conditionalUI *Conditional UI*

Description

This function takes some UI element and a compiletime-evaluated expression and returns the UI element ONLY if the condition is true

Usage

```
conditionalUI(expression, ui_element)
```

Arguments

expression	compile-time evaluated expression (no server/reactive variables)
ui_element	the UI element to display if expression = TRUE

Value

if expression==TRUE: taglist wrapping passed UI element. If expression == FALSE, empty taglist

Examples

```
# In UI:
somevariable=TRUE
shinyWidgets::panel(
  heading="constitutivepanel",
  CRUX:::conditionalUI(
    somevariable,
    shinyWidgets::panel(somevariable, heading="ConditionalPanel")
  ),

  shiny::h1("constitutive title"),
  shiny::p("constitutive paragraph")
)
```

create_mac_binary *#' Install dependencies #' #' @export #' install_dependencies <- function() BiocManager::install("PoisonAlien/TCGAmutations") #BiocManager::install("maftools")*

Description

#' Install dependencies #' #' @export #' install_dependencies <- function() BiocManager::install("PoisonAlien/TCGAmutati
#BiocManager::install("maftools")

Usage

```
create_mac_binary()
```

`download_maf` *Download MAF*

Description

Downloads a MAF file (inc. silent mutations). Does NOT download clinical datafile.

Usage

```
download_maf(maf, file)
```

Arguments

`maf`	a loaded maf object
`file`	the path to save maf to

Value

NOTHING. Run for its side effects

Examples

```
## Not run: CRUX:::download_maf(maftools::tcgaLoad("GBM"), "tcga_gbm.maf")
```

`external_tools_add_tool_to_dataframe` *external_tools_add_tool_to_dataframe*

Description

Appends tool metadata on the end of a dataframe

Usage

```
external_tools_add_tool_to_dataframe(
  external_tools_df = dplyr::tibble(),
  tool_name,
  tool_id,
  tool_group,
  tool_class,
  tool_description,
  instructions = "No instructions available yet. You're on your own buddy",
  platform = "Web App",
  website,
  doi,
  requires_maf_export = TRUE,
  requires_gene_selection = FALSE,
  maf_conversion_function = NA,
  extension = "tsv"
)
```

Arguments

`external_tools_df`
the dataframe to add tool metadata to. By default, will create and return a new dataframe

`tool_name` name of tool (string)

`tool_id` id of tool (string)

`tool_group` research group that built/maintains the tool (string)

`tool_class` class of tool. Usually ’Positive Selection’, ’Variant Interpretation’ (string)

`tool_description`
brief description of tool (string)

`instructions` brief description of how to use the tool. Any HTML tags in the string will be correctly resolved. (string)

`platform` what platform do we access the tool from. Examples include web, desktop app, cli (string)

`website` url of tool (string)

`doi` publicatoin doi (string)

`requires_maf_export`
does the tool require a maf to be exported in some other form (flag)

`requires_gene_selection`
does user need to select a specific gene for export to work? (bool)

`maf_conversion_function`
only relevent if requires_maf_export == true. A function that takes a maf_dataset_wrapper object (first argument), a filepath (second argument) and, if requires_gene_selection == TRUE, a gene name (third argument) writes a file to that filepath. The idea is that said file can then be used as input to the specified tool.

`extension` what type of file is written by maf_conversion_function. default is ’tsv’. Used to appropriately name exported file (string)

Value

dataframe containing external_tool_metadata

`external_tools_convert_maf_to_bbglab` *Conversion Function: MAF to BBGLab Cancer Genome Interpreter Input*

Description

Convets MAF object to BBGLab Cancer Genome Interpreter Input (Genomic tabular format) then writes to a file. Works as a maf_conversion_function. Please don't change arguments

Usage

```
external_tools_convert_maf_to_bbglab(maf_dataset_wrapper, filepath)
```

Arguments

`maf_dataset_wrapper`	object to add to the data pool (maf_dataset_wrapper)
`filepath`	(string)

Value

Run for side effects (writes bbglab datamframe to file)

`external_tools_convert_maf_to_bbglab_return_dataframe` *Conversion Function: MAF to BBGLab Cancer Genome Interpreter Input*

Description

Convets MAF object to BBGLab Cancer Genome Interpreter Input (Genomic tabular format) (a tsv).

Usage

```
external_tools_convert_maf_to_bbglab_return_dataframe(maf)
```

Arguments

`maf`	(MAF)

Value

bbglab compatible dataframe

external_tools_convert_maf_to_cbioportal_mutation_mapper_return_dataframe

MAF + Gene –> cBioPortal mutation_mapper_input

Description

Takes a MAF and converts variants hitting a particular gene to the form the cBioPortal Mutation Mapper uses as input.

Usage

```
external_tools_convert_maf_to_cbioportal_mutation_mapper_return_dataframe(
  maf,
  gene_hugo_symbol
)
```

Arguments

maf	a maftools maf object (maf)
gene_hugo_symbol	hugo symbol representing the gene to export (string)

Details

You can find the tool at: https://www.cbioportal.org/mutation_mapper

Value

dataframe in a form usable as input at https://www.cbioportal.org/mutation_mapper (data.frame)

external_tools_convert_maf_to_oncodrive_return_dataframe

Conversion Function: MAF to BBGLab Cancer Genome Interpreter Input

Description

Convets MAF object to BBGLab Oncodrive mutation format.

Usage

```
external_tools_convert_maf_to_oncodrive_return_dataframe(maf)
```

Arguments

maf	(MAF)

Value

data.frame of appropriate format to supply to oncodrive

`external_tools_convert_maf_to_vanilla_vcf_return_dataframe`

Maf to VCF

Description

Convert a maf to a vanilla vcf (single sample)

Usage

```
external_tools_convert_maf_to_vanilla_vcf_return_dataframe(maf_df)
```

Arguments

`maf_df` either a maf object or a dataframe created using maftools_get_all_data

Value

data.frame with VCF columns

`external_tools_get_property_by_tool_name`

external_tools_get_property_by_tool_name

Description

external_tools_get_property_by_tool_name

Usage

```
external_tools_get_property_by_tool_name(
  tool_name,
  property_to_retrieve,
  external_tools_df = external_tool_metadata
)
```

Arguments

`tool_name` name of tool

`property_to_retrieve`
which property to retrieve. see details for options. (string)

`external_tools_df`
the dataframe to add tool metadata to. By default, will use global variable: GLOBAL_external_tools_dataframe

Details

Options for property_to_retrieve

- tool_name
- tool_id
- tool_group
- tool_class
- tool_description
- instructions
- website
- doi
- requires_maf_export
- requires_gene_selection
- maf_conversion_function
- extension

See `?external_tools_add_tool_to_dataframe` for more info on these properties

run `external_tool_metadata` to see the built in tool metadata dataframe

Value

value of retrieved property (string)

Examples

```
CRUX:::external_tools_get_property_by_tool_name(tool_name = "OncodriveFML", "website")
```

---

external_tools_load_all_tools

Load tool metadata into global variable

Description

Loads metadata for all tools, returning a dataframe. If any of the constitutent functions are changed, run `external_tools_update_builtin_dataset`

Usage

```
external_tools_load_all_tools()
```

Value

external_tools_df with metadata of tool appeneded (data.frame). see `?external_tools_add_tool_to_dataframe` for more info on the columnss in this data.frame

Examples

```
CRUX:::external_tools_load_all_tools()
```

external_tools_load_bbglab_cgi

Load tool metadata into environment. Returns

Description

Appends metadata for the "Cancer Genome Interpreter (cgi) tool onto external_tools_df

Usage

```
external_tools_load_bbglab_cgi(external_tools_df = data.frame())
```

Arguments

`external_tools_df`
: the dataframe to add tool metadata to. By default, will create and return a new dataframe

Value

external_tools_df with metadata of tool appeneded (data.frame)

Examples

```
CRUX:::external_tools_load_bbglab_cgi()
```

external_tools_load_bbglab_oncodrive_clustl

Load tool metadata into global variable

Description

Appends metadata for the OncodriveCLUSTL tool onto external_tools_df

Usage

```
external_tools_load_bbglab_oncodrive_clustl(external_tools_df = data.frame())
```

Arguments

`external_tools_df`
: the dataframe to add tool metadata to. By default, will create and return a new dataframe

Value

external_tools_df with metadata of tool appeneded (data.frame)

Examples

```
CRUX:::external_tools_load_bbglab_cgi()
```

`external_tools_load_bbglab_oncodrive_fml`

Load tool metadata into global variable

Description

Loads metadata for the OncodriveFML tool into a global variable **GLOBAL_external_tools_dataframe**

Usage

```
external_tools_load_bbglab_oncodrive_fml(external_tools_df = data.frame())
```

Arguments

`external_tools_df`
: the dataframe to add tool metadata to. By default, will create and return a new dataframe

Value

external_tools_df with metadata of tool appeneded (data.frame)

Examples

```
CRUX:::external_tools_load_bbglab_cgi()
```

`external_tools_load_proteinpaint`

Load tool metadata into global variable

Description

Appends metadata for the OncodriveCLUSTL tool onto external_tools_df

Usage

```
external_tools_load_proteinpaint(external_tools_df = data.frame())
```

Arguments

`external_tools_df`	the dataframe to add tool metadata to. By default, will create and return a new dataframe

Value

external_tools_df with metadata of tool appeneded (data.frame)

`external_tools_update_builtin_dataset` *Update builtin external_tools dataset*

Description

Takes the output of external_tools_load_all_tools and saves it to CRUX/data as the dataset: **external_tool_metadata**. This saved dataset is what is used by the app. If you make change to any external_tools_load_... function, you must rerun this function.

Usage

```
external_tools_update_builtin_dataset()
```

`get_rendered_plot_height` *Get Width/Height of a rendered element*

Description

Get Width/Height of a rendered element

Usage

```
get_rendered_plot_height(output_id, session, return_inches = F, dpi = 70)
```

Arguments

`output_id`	"ID of the plot/element who’s width/height you want to know"
`session`	"Current Session. Used to get namespacing and clientData information"
`return_inches`	"Should we return width/height in pixels or inches"
`dpi`	"dpi used to convert pixels to inches"

Value

"Width/Height in pixels/inches"

get_rendered_plot_width

Get Width/Height of a rendered element

Description

Get Width/Height of a rendered element

Usage

```
get_rendered_plot_width(output_id, session, return_inches = F, dpi = 70)
```

Arguments

`output_id`	"ID of the plot/element who's width/height you want to know"
`session`	"Current Session. Used to get namespacing and clientData information"
`return_inches`	"Should we return width/height in pixels or inches"
`dpi`	"dpi used to convert pixels to inches"

Value

"Width/Height in pixels/inches"

get_tcga_mafs

Get TCGA MAFs

Description

Get TCGA MAFs

Usage

```
get_tcga_mafs(study_name)
```

Arguments

`study_name`	Abbreviation of TCGA dataset to install. To see options, run maftools::tcgaAvailable()

Value

A single MAF object.

`gistic_tabulate_amp_or_del_genes_file` *Tabulate Gistic Gene Files*

Description

Parses a gistic path_to_amp_or_del_genes_file

Usage

```
gistic_tabulate_amp_or_del_genes_file(path_to_amp_or_del_genes_file)
```

Arguments

`path_to_amp_or_del_genes_file`	path to either del_genes.conf_XX.txt OR amp_genes.conf_90.txt. These files can be produced using gistic (string)

Value

a dataframe containing cytobands, q values, peak locations and which genes are in each peak

`guess_genomic_mutation_filetype` *Guesses mutation filetype*

Description

Reads the first line of a file and attempt to determine the filetype based on the header

Usage

```
guess_genomic_mutation_filetype(path)
```

Arguments

`path`	path to file

Value

one of MAF, ANNOVAR, VCF, OTHER

html_alert *HTML_alert*

Description

HTML_alert

Usage

```
html_alert(text, status = "warning")
```

Arguments

`text`	Alert text (string)
`status`	Boostrap status. One of primary,secondary,success,danger,warning,info,light,dark (string)

Value

html-flagged text (string")

icon_down_arrow *icon_down_arrow*

Description

icon_down_arrow

Usage

```
icon_down_arrow(fontsize = "40px", alignment = "center", break_after = FALSE)
```

Arguments

`fontsize`	a valid css fontsize, e.g. 60px (string)
`alignment`	One of 'left', 'right', 'center' or 'inherit'(string)
`break_after`	insert a linebreak after icon (flag)

Value

centered down arrow fluidrow()

imports *My function*

Description

This is a description of my function.

Usage

```
imports()
```

is_valid_clinicalfeaturefile *Is a clinical_feature_file valid for a given maf*

Description

Runs read.maf with a given clinicalData and maf object pair and returns TRUE ONLY if the clinicalData file is appropriate for the given MAF. If maf object is not valid, this function will also return FALSE. By using clinicalData=NULL the function allows testing validity of a MAF file in Isolation.

Usage

```
is_valid_clinicalfeaturefile(clinicalData, maf)
```

Arguments

clinicalData	Clinical data associated with each sample/Tumor_Sample_Barcode in MAF. Could be a text file or a data.frame. Default NULL.
maf	tab delimited MAF file. File can also be gz compressed. Required. Alternatively, you can also provide already read MAF file as a dataframe.

Details

If clinical feature file has duplicated Tumor_Sample_Barcodes, only the first entry is considered and the file is still 'valid'.

Value

True / False (boolean)

is_valid_clinicalfeaturefile_return_error

Is a clinical_feature_file valid for a given maf

Description

Runs read.maf with a given clinicalData and maf object pair and returns the relevant error messages If maf object is not valid, this function will also return the appropriate error message. By using clinicalData=NULL the function allows testing validity of a MAF file in isolation

Usage

```
is_valid_clinicalfeaturefile_return_error(clinicalData, maf)
```

Arguments

clinicalData	Clinical data associated with each sample/Tumor_Sample_Barcode in MAF. Could be a text file or a data.frame. Default NULL.
maf	tab delimited MAF file. File can also be gz compressed. Required. Alternatively, you can also provide already read MAF file as a dataframe.

Details

If clinical feature file has duplicated Tumor_Sample_Barcodes, only the first entry is considered and the file is still ’valid’.

Value

Metadata and MAF are valid if they are. The relevant string if they are not (character)

link

Link

Description

Link

Usage

```
link(url, newtab = TRUE, text)
```

Arguments

url	URL
newtab	shoule the link be opened in a new tab? (flag)
text	display text for link

Value

list() with a shiny.tag class. Can convert to HTML string via as.character()

---

maftools_add_brca_subtype

add BRCA subtype clinical feature

Description

Checks if maf is BRCA and contains cols required to classify as Triple Negative / Not Triple Negative. If so, adds a 'subtype' column describing whether its triple negative or not triple negative

Usage

```
maftools_add_brca_subtype(maf)
```

Arguments

`maf`	maftools maf object

Value

maf object

---

maftools_add_clinical_data

Add clindata to MAF

Description

Adds 'extra' clinical metadata to a maf object with existing metadata

Usage

```
maftools_add_clinical_data(maf, clindata_path)
```

Arguments

`maf`	an existing MAF object as produced by maftools::read.maf (MAF)
`clindata_path`	Eithera path to a csv/tsv that contains sample level metadata. Must include a

Value

maf object with

maftools_chrom_23_and_24_to_X_and_Y

Convert chromosomes 23 and 24 to x and y in maf object. This will convert

Description

Takes a maf object and returns that same object but converts and chromosomes named 23 or 24 (or chr23 / chr24) to X and Y.

Usage

```
maftools_chrom_23_and_24_to_X_and_Y(maf)
```

Arguments

`maf`	A MAF object (MAF)

Value

a MAF object with chr23/chr24 converted to "X" & "Y"

maftools_clinical_data_get_levels

maftools_clinical_data_get_levels

Description

maftools_clinical_data_get_levels

Usage

```
maftools_clinical_data_get_levels(maf, clinical_feature)
```

Arguments

`maf`	MAF object (MAF)
`clinical_feature`	Name of a clinical feature (string)

Value

Number of distinct levels of a clinical feature

maftools_clinical_data_visually_summarise *Plot Clinical Data Plot a particular column of clinical DATA.*

Description

Plot Clinical Data Plot a particular column of clinical DATA.

Usage

```
maftools_clinical_data_visually_summarise(
  maf,
  clinical_feature = "Tumor_Sample_Barcode",
  threshold = NULL,
  selected_items = NULL,
  distance_from_bar = 2
)
```

Arguments

`maf`	MAF object (MAF)
`clinical_feature`	Name of a clinical feature (string)
`threshold`	threshold (number)
`selected_items`	selectd items
`distance_from_bar`	how far should text be from the bar (number)

Value

ggplot / grob

maftools_clinical_feature_description *Get a data.frame describing each clinical feature of a MAF*

Description

Get a data.frame describing each clinical feature of a MAF

Usage

```
maftools_clinical_feature_description(
  maf,
  checkmark = c("none", "oncoplottable")
)
```

Arguments

`maf` a maftools maf object

`checkmark` should a pass/fail checkmark icon be added before the annotations name. (none = no pass/fail checkmark; oncoplottable = whether feature can be added to an oncoplot)

Details

Rule for pass/fail checks: To be 'oncoplottable' there must be <= 100 non-missing levels OR the feature must be numeric

Value

a data.frame with 3 columns. annotation (feature name); type (whether feature is numeric/categorical); levels (number of distinct, non-na values); content (html string with names + badges with additional information)

`maftools_cluster_samples` *maftools_cluster_samples*

Description

Identify and visualise clusters of cancer samples based on somatic mutation data (gene-level differences visualised, not variant-level)

Approach involves:

1. Selecting a geneset of interest (by default uses genes mutated in the most samples).
2. Calculating **1-jaccard** distance between samples based on which genes of the genesets are mutated.
3. Running heirarchical clustering algorithm using **pheatmap**.
4. Visualise resulting heatmap with user-selected annotations.

Usage

```
maftools_cluster_samples(
  maf,
  custom_genelist_to_cluster_by = NULL,
  number_of_genes = 50,
  genes_to_annotate = NULL,
  annotate_most_altered_genes = TRUE,
  topn_genes = 5,
  metadata_columns = NULL,
  include_silent_mutations = FALSE,
  show_rownames = FALSE,
  show_colnames = FALSE,
  annotation_legend = TRUE,
  fontsize = 10
)
```

Arguments

`maf` maf object from maftools package (maf)

`custom_genelist_to_cluster_by`
names of genes to base clustering on. If unsure what genes to use. By default, the top 50 genes ranked by how manys samples they are mutated in will be used (character vector)

`number_of_genes`
number of genes to cluster based on. Chooses genes which are mutated in the most samples (only matters if not supplying **custom_genelist_to_cluster_by**) (integer)

`genes_to_annotate`
a custom list of genes to plot as a pseudo oncoplot (string)

`annotate_most_altered_genes`
automatically annotate with mutational status of genes mutated in the most samples (this is always based on coding/splice site mutation) (bool)

`topn_genes` if annotate_most_altered_genes is true, how many genes to automatically visualise (integer)

`metadata_columns`
name of metadata columns to annotate heatmap based on (character)

`include_silent_mutations`
consider a gene mutated even if the only mutations present are silent (bool)

`show_rownames` show sample names on rows (bool)

`show_colnames` show sample names on columns (bool)

`annotation_legend`
show annotation legend (flag)

`fontsize` font size (number)

Value

pheatmap object

Examples

```
maf <- maftools::tcgaLoad("GBM", source = "Firehose")
maftools_cluster_samples(maf)
```

maftools_extract_geneset_by_altered_samples

Extract geneset

Description

Create a genelist containing the genes mutated in the most samples in a maf. If topn > total number of mutated genes then all mutated genes are returned.

Usage

```
maftools_extract_geneset_by_altered_samples(maf, topn = 100)
```

Arguments

`maf`	tab delimited MAF file. File can also be gz compressed. Required. Alternatively, you can also provide already read MAF file as a dataframe.
`topn`	How many genes to include in the gene-set.

Value

The names of genes mutated in the most samples (character vector).

Examples

```
CRUX:::maftools_extract_geneset_by_altered_samples(maftools::tcgaLoad("GBM"), topn=50)
```

maftools_fix_clinical_data_types

Type MAF clinical data

Description

Returns a MAF with correctly typed clinical data. By default, many maftools operations (e.g. subsetting) results in a loss of clinical data type (everything gets cast as characters). This function can be run on any MAF to resolve this issue.

Usage

```
maftools_fix_clinical_data_types(maf)
```

Arguments

`maf`	a maf object (MAF)

Value

MAF object with correctly typed clinical.data dataframe (MAF)

Examples

```
maf = maftools::tcgaLoad(study = "GBM", source = "Firehose")
str(maftools::getClinicalData(maf))
new_maf = CRUX:::maftools_fix_clinical_data_types(maf)
str(maftools::getClinicalData(new_maf))
```

`maftools_get_all_data` *Get MAF data*

Description

Input a maf object. Returns table containing ALL MAF data. Nonsynonymous AND synonymous

Usage

```
maftools_get_all_data(maf, include_silent_mutations = T)
```

Arguments

`maf` maf object

`include_silent_mutations`
include silent mutations (flag)

Value

a data.frame in MAF form where each variant has a separate row (data.frame)

Examples

```
CRUX:::maftools_get_all_data(maftools::tcgaLoad("GBM"))
```

maftools_get_transcript_refseq

Get Longest Transcript Refseq ID

Description

Returns the transcript id from the 'Transcript_ID' column of a maf_df. If no Transcript_ID column is found, will return the longest refseq transcript for each gene name.

Usage

```
maftools_get_transcript_refseq(maf_df)
```

Arguments

maf_df	(HGNC ID of gene)

Value

vector of the same length as maf_df containing transcript refseq ids (character)

maftools_number_of_samples

Number of samples in a MAF

Description

Number of samples in a MAF

Usage

```
maftools_number_of_samples(maf)
```

Arguments

maf	@inherit maftools::read.maf

Value

int

Examples

```
CRUX:::maftools_number_of_samples(maftools::tcgaLoad("GBM"))
```

maftools_plot_rainfall

Maftools rainfallPlot Wrapper

Description

Wraps maftools::rainfallPlot. The original function plots a rainfall graph, prints the predicted kataegis coords as a df, and writes it to a file. This function does all of this EXCEPT it hides the printed df, reads the df from the file and returns it. It also deletes the created file.

Usage

```
maftools_plot_rainfall(
  maf,
  tsb,
  detectChangePoints = TRUE,
  ref.build = "hg19",
  pointSize = 0.4,
  fontSize = 1.2
)
```

Arguments

maf	an MAF object generated by read.maf. Required.
tsb	specify sample names (Tumor_Sample_Barcodes) (string)
detectChangePoints	If TRUE, detectes genomic change points where potential kataegis are formed. Results are written to an output tab delimted file.
ref.build	Reference build for chromosome sizes. Can be hg18, hg19 or hg38. Default hg19.
pointSize	Default 0.8.
fontSize	Default 12.

Value

predicted kataegis sites (dataframe)

maftools_remove_dubious_genes *Filter dubious genes from MAF*

Description

Filter out genes likely to appear in many analyses from your MAF (e.g. TTN & Olfactory receptors)

Usage

```
maftools_remove_dubious_genes(maf, genelist = somaticflags::somaticflags)
```

Arguments

`maf`	a MAF object
`genelist`	a character vector containing HUGO Symbols to remove from MAF

Value

MAF object

Examples

```
CRUX:::maftools_remove_dubious_genes(maftools::tcgaLoad("gbm"), "TTN")
```

maf_data_pool_add_dataset *Add to MAF Data Pool*

Description

Add a maf_dataset_wrapper object to the maf_data_pool

Usage

```
maf_data_pool_add_dataset(maf_dataset_wrapper, maf_data_pool)
```

Arguments

`maf_dataset_wrapper`	object to add to the data pool (maf_dataset_wrapper)
`maf_data_pool`	the data pool to add the dataset wrapper to (maf_data_pool)

Value

a new maf_data_pool object with the additional objects appended (maf_data_pool)

maf_data_pool_add_rnaseq

Add RNAseq – maf_data_pool version

Description

Add RNA data to a dataset_wrapper within a maf_data_pool.

Usage

```
maf_data_pool_add_rnaseq(maf_data_pool, unique_name, rnaseq_path)
```

Arguments

`maf_data_pool`	see ?new_maf_data_pool for details (maf_data_pool)
`unique_name`	Unique name of the maf_data_wrapper. see ?new_maf_data_wrapper for details (string)
`rnaseq_path`	Path to rnaseq file (string)

Details

How this works: We read in the file at rnaseq_path as data frame, check if it looks like we expect. If so, the **filepath** is saved to the maf_dataset_wrapper (NOT THE DATAFRAME!). This is so we don't have to keep the RNA data in memory. When we go to export or visualise, we'll just reload it using **maf_data_wrapper_get_rnaseq_df**, which returns a dataframe (or NULL if none is found).

Also note that no cohort subsetting will affect this RNA file, as all we have is the filepath. This is not a problem, since the functions that export / visualise the data simply load it into memory, then check the clinical datafile to see which samples we need to export. If they've been subset out of the clinical datafile, the user probably wont want to export RNA data from these samples ... my export function are aware of this.

Also note, that this file wont be subset

Value

maf_data_pool with updated maf_data_pool_add_rnaseq

`maf_data_pool_get_all_nonfunction_property_names` *Get Colnames for maf_data_pool to data.frame conversion*

Description

Gets the names of all maf_dataframe_wrapper properties that don't hold functions. This information is useful when converting to a data.frame which can't hold functions

Usage

```
maf_data_pool_get_all_nonfunction_property_names(maf_data_pool)
```

Arguments

`maf_data_pool`	data pool of interest (maf_data_pool)

Details

This function works by only looking at the first element in the list, which means if one day we change the constructor such that not all wrappers have the same set of properties, we may see unexpected behaviour. Since it is extremely unlikely this will change, I won't make this more robust for now. If I am wrong ... sorry future me :(

Value

a vector of list_within_list 'properties' that are not functions (character)

`maf_data_pool_get_data_wrapper_from_unique_name` *Retrieve dataset wrapper from data pool using unique_name*

Description

Will throw error if unique_name is not found

Usage

```
maf_data_pool_get_data_wrapper_from_unique_name(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	The data pool of interest (maf_data_pool)
`unique_name`	unique_name of the maf_dataset_wrapper fetch (string).

Value

specified maf_dataset_wrapper

`maf_data_pool_get_index_from_unique_name` *Retrieve dataset wrapper from data pool using unique_name*

Description

Will throw error if unique_name is not found

Usage

```
maf_data_pool_get_index_from_unique_name(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	The data pool of interest (maf_data_pool)
`unique_name`	unique_name of the maf_dataset_wrapper you're interested in(string).

Value

index of the specified maf_dataset_wrapper within the maf_data_pool

`maf_data_pool_get_unique_names` *Get Unique Names from Maf Data Pool*

Description

Extract 'unique_name' from each maf_dataset_wrapper within a maf_data_pool. **NOTE:** 'unique_name' is a property of maf_dataset_wrapper, however since these wrappers are created independently from each other (no check for uniqueness at the time of creation) they may not actually be unique. Please check this using maf_data_pool_validate_unique_names

Usage

```
maf_data_pool_get_unique_names(maf_data_pool)
```

Arguments

`maf_data_pool`	A data pool to mine unique_names from (maf_data_pool)

Value

a vector listing the 'unique_name' of each maf_dataset_wrapper in the data pool (character)

maf_data_pool_load_data

Load a dataset wrapper from data pool using unique_name

Description

use maf_data_pool_robust_load instead. It wraps this function Will throw error if unique_name is not found or status is anything other than 'not_loaded'.

Usage

```
maf_data_pool_load_data(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	The data pool of interest (maf_data_pool)
`unique_name`	unique_name of the maf_dataset_wrapper you want to load (string).

Value

maf_data_pool object with the wrapped dataset now loaded (maf_data_pool)

maf_data_pool_make_name_unique

Make a name unique (within datapool)

Description

Checks if a name is unique in a given data pool. If not, the function will append 15 letters randomly to the name until it finds a unique name, or surpasses `max_number_of_attempts`. Will throw an error if it can't find a unique name

Usage

```
maf_data_pool_make_name_unique(
  maf_data_pool,
  name,
  max_number_of_attempts = 50
)
```

Arguments

`maf_data_pool`	The data pool of interest (maf_data_pool)
`name`	name to make unique (string)
`max_number_of_attempts`	max number of attempts (whole number)

Value

a unique name (string)

maf_data_pool_robust_load

Load MAF objects

Description

This function allows user to pass a maf data pool and a unique name of the dataset of interest. It differs from maf_data_pool_load_data in that maf_data_pool_load_data will throw an error if the status of the loaded object is anything other than "not_loaded". This function will take any valid status and try to get the data loaded. It decides if the data actually needs loading, and if so, loads it.

It is designed to run before you run maf_data_pool_unique_name_to_maf / maf_data_pool_get_data_wrapper_from_unique_

Usage

```
maf_data_pool_robust_load(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	The data pool of interest (maf_data_pool)
`unique_name`	unique_name of the maf_dataset_wrapper you want to load (string).

Value

maf_data_pool with the specified datawrapper loaded (maf_data_pool)

maf_data_pool_to_dataframe

MAF Data Pool to Dataframe

Description

Converts a data pool to a data.frame form.

Usage

```
maf_data_pool_to_dataframe(maf_data_pool)
```

Arguments

`maf_data_pool`	the data pool to convert to a dataframe (maf_data_pool)

Value

a data.frame containing all properties of maf_data_pool except for those containing functions (data.frame)

maf_data_pool_unique_name_is_available

Check availability of a name in the data pool

Description

Checks if a particular ’unique_name’ is already being used by an object in the some data pool.

Usage

```
maf_data_pool_unique_name_is_available(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	The data pool of interest (maf_data_pool)
`unique_name`	some string you want to check is not currently used as the ’unique_name’ of any object in the datapool (string).

Value

TRUE/FALSE depending on whether the unique_name is available (logical)

maf_data_pool_unique_name_to_maf_nonreactive

maf_data_pool_unique_name_to_maf_nonreactive

Description

Takes a unique name and returns the maf object associated with said entry in the maf_data_pool. Involves loading the dataset but does NOT actually update the maf_data_pool.

Usage

```
maf_data_pool_unique_name_to_maf_nonreactive(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	the data pool to add the dataset wrapper to (maf_data_pool)
`unique_name`	unique name (string)

Value

return loaded data (maf object, nonreactive)

`maf_data_pool_unique_name_to_maf_reactive` *maf_data_pool_unique_name_to_maf_reactive*

Description

Loads the relevant MAF if possible, appplies changes to maf_data_pool reactive then returns the loaded maf

Usage

```
maf_data_pool_unique_name_to_maf_reactive(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	the reactiveVal we want will search for our dataset of interest, and is also the object we apply any changes to. (reactiveVal)
`unique_name`	unique_name of the maf_dataset_wrapper fetch (string).

Value

maf object (maf)

`maf_data_pool_unload_data` *Unload dataset wrapper from data pool using unique_name*

Description

Will throw error if unique_name is not found

Usage

```
maf_data_pool_unload_data(maf_data_pool, unique_name)
```

Arguments

`maf_data_pool`	The data pool of interest (maf_data_pool)
`unique_name`	unique_name of the maf_dataset_wrapper you want to unload (string).

Value

maf_data_pool object with the wrapped dataset now unloaded (maf_data_pool)

maf_data_set_wrapper_load_data

Load data into memory

Description

Uses the functions in maf_dataset_wrapper to load data into the loaded_data element and update status.

Usage

```
maf_data_set_wrapper_load_data(maf_dataset_wrapper)
```

Arguments

`maf_dataset_wrapper`
the wrapper of the dataset you want to load into memory (maf_dataset_wrapper)

Details

It is only the maf object that is loaded. RNA data stays sitting in a file until it is retrieved via maf_dataset_wrapper_get_rnaseq

Value

a copy of the original wrapper with status and loaded data updated. (maf_dataset_wrapper)

See Also

Other data_set_wrapper_loading: `maf_data_set_wrapper_unload_data()`

Examples

```
#Generate wrapper
my_data <- CRUX:::tcga_dataset_to_maf_dataset_wrapper(
  maf_data_pool = CRUX:::new_maf_data_pool(),
  tcga_study_abbreviation = "ACC"
)

#Load data
my_data <- CRUX:::maf_data_set_wrapper_load_data(my_data)

#Access loaded data
print(my_data$loaded_data)

#Unload when finished
my_data <- CRUX:::maf_data_set_wrapper_unload_data(my_data)
```

maf_data_set_wrapper_unload_data

Unload data from memory

Description

Uses the functions in maf_dataset_wrapper to load data into the loaded_data element and update status.

Usage

```
maf_data_set_wrapper_unload_data(maf_dataset_wrapper)
```

Arguments

`maf_dataset_wrapper`
the wrapper of the dataset you want to unload into memory (maf_dataset_wrapper)

Value

a copy of the original wrapper with status and loaded data elements updated. Status is changed from "ready" => "not_loaded". loaded_data is changed from maf_object to NA. (maf_dataset_wrapper)

See Also

Other data_set_wrapper_loading: `maf_data_set_wrapper_load_data()`

Examples

```
#Generate wrapper
my_data <- CRUX:::tcga_dataset_to_maf_dataset_wrapper(
  maf_data_pool = CRUX:::new_maf_data_pool(),
  tcga_study_abbreviation = "ACC"
)

#Load data
my_data <- CRUX:::maf_data_set_wrapper_load_data(my_data)

#Access loaded data
print(my_data$loaded_data)

#Unload when finished
my_data <- CRUX:::maf_data_set_wrapper_unload_data(my_data)
```

maf_data_wrapper_add_rnaseq

Add RNAseq slot to maf_dataset_wrapper

Description

Add rnaseq data to maf_dataset_wrapper object

Usage

```
maf_data_wrapper_add_rnaseq(maf_dataset_wrapper, rnaseq_path)
```

Arguments

maf_dataset_wrapper	any maf object (maf_dataset_wrapper)
rnaseq_path	path to rnaseq dataset (string)

Value

the input maf_dataset_wrapper with rnaseq_path in rnaseq_filepath slot

maf_data_wrapper_get_rnaseq_data_for_samples_with_mutation_data

Get Expression Data from maf

Description

Similar tomaf_data_wrapper_get_rnaseq_df but only returns expression data for samples that have corresponding mutation data

Usage

```
maf_data_wrapper_get_rnaseq_data_for_samples_with_mutation_data(
  maf_dataset_wrapper
)
```

Arguments

maf_dataset_wrapper	a maf_dataset wrapper. See ?new_maf_dataset_wrapper for details.

Value

RNAseq data if present or NULL if no RNA data has been associated with it yet. Will only return expression data for samples that have corresponding mutation data (dataframe)

Examples

```
# Prepare Data
## Not run:
rna_path = system.file("example_data/blca_rnaseq.tsv", package = "CRUX")
maf_data_wrapper = CRUX:::tcga_dataset_to_maf_dataset_wrapper(CRUX:::new_maf_data_pool(), "BLCA")

# Add RNA data
maf_data_wrapper_with_RNA = CRUX:::maf_data_wrapper_add_rnaseq(
  maf_data_wrapper,
  rnaseq_path = rna_path
)

# Retrieve RNA data for samples with mutation data
CRUX:::maf_data_wrapper_get_rnaseq_data_for_samples_with_mutation_data(maf_data_wrapper_with_RNA)

## End(Not run)
```

---

maf_data_wrapper_get_rnaseq_df

Get rnaseq data

Description

You probaly want maf_data_wrapper_get_rnaseq_data_for_samples_with_mutation_data instead.

Usage

```
maf_data_wrapper_get_rnaseq_df(maf_dataset_wrapper)
```

Arguments

`maf_dataset_wrapper`
a maf_dataset wrapper. See ?new_maf_dataset_wrapper for details.

Details

This function takes a maf_dataset_wrapper and retrieves ALL rnaseq data present (inc. for samples that would have been filtered out by subsetting operations).

Value

RNAseq data if present or NULL if no RNA data has been associated with it yet (dataframe)

`maf_data_wrapper_has_rnaseq_data` *Maf Dataset has RNAseq data*

Description

Maf Dataset has RNAseq data

Usage

```
maf_data_wrapper_has_rnaseq_data(maf_dataset_wrapper)
```

Arguments

`maf_dataset_wrapper`
a maf dataset wrapper. See ?new_maf_dataset_wrapper

Value

TRUE if RNAseq data is present, False if not (Boolean)

`moduleDownloadPlotServer` *Saves Plots as SVG.*

Description

Saves Plots as SVG.

Usage

```
moduleDownloadPlotServer(
  id,
  session_parent,
  plotOutputId,
  plotting_function,
  default_filename = "download"
)
```

Arguments

`id` Module ID. Must be identical to moduleDownloadPlotUI

`session_parent` Session object of the calling module (used to extract plot details)

`plotOutputId` "ID of the rendered plot"

`plotting_function`
"function that when run with no arguments will create the plot. Can make by wrapping the plot call in its own function. e.g. for plot(mtcars) you could do: plotting_function <- reactive (function() plot(mtcars)) then pass ’plotting_function()’"

`default_filename`
default basename of downloaded file (string; non-reactive)

`moduleDownloadPlotUI` *Creates a shinyWidgets::downloadbttn*

Description

Creates a shinyWidgets::downloadbttn

Usage

```
moduleDownloadPlotUI(
  id,
  circle = FALSE,
  label = "Download",
  style = "unite",
  color = "default",
  size = "default",
  status = "default",
  icon = NULL,
  tooltip_placement = "right",
  tooltip_text = "",
  right = FALSE,
  up = FALSE,
  width = "200px",
  margin = "10px",
  inline = FALSE,
  ...
)
```

Arguments

`id` Module ID

`circle` Logical. Use a circle button

`label` The label that should appear on the button.

`style`	Style of the button, to choose between simple, bordered, minimal, stretch, jelly, gradient, fill, material-circle, material-flat, pill, float, unite.
`color`	Color of the button : default, primary, warning, danger, success, royal.
`size`	Size of the button : xs,sm, md, lg.
`status`	Add a class to the buttons, you can use Bootstrap status like 'info', 'primary', 'danger', 'warning' or 'success'. Or use an arbitrary strings to add a custom class, e.g. : with `status = 'myClass'`, buttons will have class `btn-myClass`.
`icon`	An icon to appear on the button.
`tooltip_placement`	Where the tooltip should appear relative to its target (top, bottom, left, or right). Defaults to "right" (string)
`tooltip_text`	Tooltip text (string)
`right`	Logical. The dropdown menu starts on the right.
`up`	Logical. Display the dropdown menu above.
`width`	Width of the dropdown menu content.
`margin`	Value of the dropdown margin-right and margin-left menu content.
`inline`	use an inline (`span()`) or block container (`div()`) for the output.
`...`	List of tag to be displayed into the dropdown menu.

Value

Nothing. Function run for its side effects

`moduleReadMafServer` *Read MAF server function*

Description

Read MAF server function

Usage

```
moduleReadMafServer(id)
```

Arguments

`id`	Internal parameters for shiny.

Value

named list containing 'maf' (type maf), 'cohortName' (type string), 'clinicalData' (type df), 'clinicalDataSupplied' (bool),

`moduleReadMafUI` *Read MAF UI function*

Description

Read MAF UI function

Usage

```
moduleReadMafUI(
  id,
  panel_name,
  cohort_name,
  label_cohort_name = "name",
  label_maf = "maf",
  label_clinical_data = "clinical features"
)
```

Arguments

`id` links ui and server components of module

`panel_name` "panel name (required)"

`cohort_name` "panel name (required)"

`label_cohort_name` "name of the cohort (required)"

`label_maf` label of maf input field ["maf"]

`label_clinical_data` label of clinicalData input field ["clinical data"]

`moduleSubsetByClinicalFeaturesServer` *moduleSubsetByClinicalFeaturesServer*

Description

moduleSubsetByClinicalFeaturesServer

Usage

```
moduleSubsetByClinicalFeaturesServer(id, maf, clinicalData)
```

Arguments

`id`	Internal parameters for shiny.
`maf`	reactive maf object
`clinicalData`	clinical data object (reactive)

Value

named list: 'maf1' (type maf), 'maf2' (type string),

`mod_data_import_maf_path_to_maf_server`

mod_data_import_step1 Server Functions

Description

mod_data_import_step1 Server Functions

Usage

```
mod_data_import_maf_path_to_maf_server(id, maf_path, clinicalData)
```

Arguments

`id`	Internal parameters for shiny.
`maf_path`	Path to maf file (string) (reactive)
`clinicalData`	Clinical data associated with each sample/Tumor_Sample_Barcode in MAF. Could be a text file or a data.frame. Default NULL.

Value

maf object if read was successful. NULL if it was not (MAF) (reactive)

`mod_data_import_maf_path_to_maf_ui`

mod_data_import_step1 UI Function

Description

GUI for taking a filepath returning a maf object. Includes a panel containing the MAF file summary or an error message if the supplied path does not point to a valid maf.

Usage

```
mod_data_import_maf_path_to_maf_ui(id)
```

Arguments

`id` Internal parameters for shiny.

`mod_data_import_step2_server`

data_import_step_2 Server Functions

Description

Returns a list of user-filled metadata. Do **NOT** evaluate any of the properties before checking all_valid == TRUE (list) (reactive).

Usage

```
mod_data_import_step2_server(
  id,
  default_data_source = reactive(NULL),
  default_display_name = reactive(NULL),
  default_short_name = reactive(NULL),
  default_description = reactive(NULL)
)
```

Arguments

`id`	Internal parameters for shiny.
`default_data_source`	default data source value
`default_display_name`	default display name value
`default_short_name`	default short name value
`default_description`	default description value

Details

The list itself is reactive, so to access elements do:

```
metadata <- mod_data_import_step2_server(id="my_id")

# In a reactive context:
observe({
    metadata()$all_valid
    metadata()$display_name
    metadata()$short_name
    metadata()$data_source
    metadata()$description
})
```

Value

named list with elements:

1. all_valid (bool)
2. display_name (string)
3. short_name (string)
4. data_source (string)
5. description (string)

---

mod_external_tools_server

external_tools Server Functions

Description

external_tools Server Functions

Usage

```
mod_external_tools_server(id, maf_data_pool)
```

Arguments

`id`	Internal parameters for shiny.
`maf_data_pool`	the data pool to add the dataset wrapper to (maf_data_pool)

---

mod_import_clinical_featurefile_server

import_clinical_featurefile Server Functions

Description

Imports a clinical feature file.

Usage

```
mod_import_clinical_featurefile_server(id, maf_path)
```

Arguments

`id`	Internal parameters for shiny.
`maf_path`	path to maf file (reactive string)

Value

the object with metadata from a clinical feature

mod_import_clinical_featurefile_ui

import_clinical_featurefile UI Function

Description

A shiny Module.

Usage

```
mod_import_clinical_featurefile_ui(id)
```

Arguments

id	Internal parameters for shiny.

mod_plot_apobec_diff_server

plot_apobec_diff Server Functions

Description

plot_apobec_diff Server Functions

Usage

```
mod_plot_apobec_diff_server(id, maf, tnm)
```

Arguments

id	Internal parameters for shiny.
maf	an MAF object used to generate the matrix
tnm	output generated by trinucleotideMatrix

mod_plot_lollipop_ui *plotforest: Wraps around maftools::forestPlot to more explicitly handle pval VS fdr option and allow both to be passed while onl using the value set by bool 'threshold_on_fdr'*

Description

plotforest: Wraps around maftools::forestPlot to more explicitly handle pval VS fdr option and allow both to be passed while onl using the value set by bool 'threshold_on_fdr'

Usage

```
mod_plot_lollipop_ui(id)
```

Arguments

`id`	Internal parameters for shiny.

Examples

```
plotforest <- function(
    mafCompareRes,
    pVal = 0.05,
    fdr = 0.05,
    threshold_on_fdr,
    color=NULL,
    geneFontSize = 1.2,
    lineWidth = 2.2,
    titleSize = 1.2
 ){
  message("running")
  if (threshold_on_fdr)
    maftools::forestPlot(
      mafCompareRes = mafCompareRes,
      fdr = fdr,
      color = color,
      geneFontSize = geneFontSize,
      lineWidth = lineWidth,
      titleSize = titleSize
     )
  else{
    maftools::forestPlot(
      mafCompareRes = mafCompareRes,
      pVal = pVal,color = color,
      geneFontSize = geneFontSize,
      lineWidth = lineWidth,
      titleSize = titleSize
    )
   }
}
```

`mod_plot_somatic_interactions_server` *somatic_interactions Server Functions*

Description

somatic_interactions Server Functions

Usage

```
mod_plot_somatic_interactions_server(id, maf)
```

Arguments

`id`	Internal parameters for shiny.
`maf`	a maf object (maf)

`mod_plot_survival_server` *plot_survival Server Functions*

Description

plot_survival Server Functions

Usage

```
mod_plot_survival_server(id, maf, geneset, time, status, or, is_tcga)
```

Arguments

`id`	Internal parameters for shiny.
`maf`	MAF object, usually produced by maftools::read_maf (reactive MAF object)
`geneset`	character vector where each element is the hugo_symbol of a gene. (reactive character vector)
`time`	name of column in clinical data describing time to event (reactive string)
`status`	name of column in clinical data describing event status (reactive string)
`or`	should samples be classified as mutant if ANY gene in geneset is mutated (default is ALL must be mutated) (reactive flag)
`is_tcga`	is sample from TCGA? (flag)

mod_render_clinical_data_table_server

Title

Description

Title

Usage

```
mod_render_clinical_data_table_server(id, maf)
```

Arguments

`id`	Internal parameters for shiny.
`maf`	maf object (reactive)

Value

Nothing. Run for its side effects

mod_render_clinical_data_table_ui

render_clinical_data_table UI Function

Description

A shiny Module. Renders clinica data attached to a maf object in the form of a datatable

Usage

```
mod_render_clinical_data_table_ui(id)
```

Arguments

`id`	Internal parameters for shiny.

mod_render_downloadabledataframe_server

Title

Description

Title

Usage

```
mod_render_downloadabledataframe_server(
  id,
  tabular_data_object,
  basename,
  rownames = FALSE,
  colnames = TRUE,
  filter = "top",
  message_if_tabular_data_is_null = "Please select valid mutalisk files"
)
```

Arguments

`id`	shiny paramater
`tabular_data_object`	tabular data object (usually data.frame or data.table)
`basename`	name of downloaded file (flag)
`rownames`	download with rownames (flag)
`colnames`	download with colnames (flag)
`filter`	Position of filter search box: one of 'top', 'bottom' or 'none' (string)
`message_if_tabular_data_is_null`	message if tabular data is null (string)

mod_select_maf_dataset_wrapper_server

Select Dataset, Return maf_dataset_wrapper

Description

wraps mod_select_dataset_from_maf_data_pool_pickerinput_server. Instead of simply returning a unique_name, this function will:

Usage

```
mod_select_maf_dataset_wrapper_server(id, maf_data_pool, label = "Dataset")
```

Arguments

`id` Internal parameters for shiny.

`maf_data_pool` the data pool to add the dataset wrapper to (maf_data_pool)

`label` label (string)

1. Load the specified dataset into memory if required.
2. Update maf_data_pool (a reactiveVal) to indicate the dataset has been loaded.
3. Return the relevant maf_dataset_wrapper. See **Accessing Properties** section for details.

Value

maf_dataset_wrapper.

Accessing Properties (Quick Reference)

MAF object:	`maf_dataset_wrapper()$loaded_data`
unique name:	`maf_dataset_wrapper()$unique_name`
short name:	`maf_dataset_wrapper()$short_name`
full name:	`maf_dataset_wrapper()$display_name`
source:	`maf_dataset_wrapper()$name_of_data_source`

See ?new_maf_dataset_wrapper for the full list of properties

`mod_shinyfiles_get_clinical_featurefile_path_server` *Get Path to Clinical Feature File*

Description

Server code for getting path to clinical feature file using shinyfiles button. No QC done here. Make sure file is valid when creating MAF object from maf + clinical feature file

Usage

```
mod_shinyfiles_get_clinical_featurefile_path_server(id)
```

Arguments

`id` Internal parameters for shiny.

Value

chosen filepath (string) (reactive). If multiple = TRUE, it returns a character vector instead of a string. When no file is selected, returns character(0)

`mod_shinyfiles_get_clinical_featurefile_path_ui` *shinyfiles_get_clinical_featurefile_path UI Function*

Description

A shiny Module.

Usage

```
mod_shinyfiles_get_clinical_featurefile_path_ui(id)
```

Arguments

`id`	Internal parameters for shiny.

`mod_shinyfiles_get_maf_path_server` *shinyfiles_get_maf_path Server Functions*

Description

shinyfiles_get_maf_path Server Functions

Usage

```
mod_shinyfiles_get_maf_path_server(id)
```

Arguments

`id`	Internal parameters for shiny.

Value

chosen filepath (string) (reactive). If multiple = TRUE, it returns a character vector instead of a string. When no file is selected, returns character(0)

mod_shinyfile_import_server

shinyfile_import Server Functions

Description

shinyfile_import Server Functions

Usage

```
mod_shinyfile_import_server(id)
```

Arguments

`id`	Internal parameters for shiny.

Value

chosen filepath (string) (reactive). If multiple = TRUE, it returns a character vector instead of a string. When no file is selected, returns character(0)

mod_shinyfile_import_ui

shinyfile_import UI Function

Description

Adds a shinyFilesButton

Usage

```
mod_shinyfile_import_ui(
  id,
  title,
  label,
  multiple = FALSE,
  tooltip_text = "",
  tooltip_placement = "right",
  trigger = "hover"
)
```

Arguments

`id`	Internal parameters for shiny.
`title`	The content of the tooltip.
`label`	Display label for the control, or `NULL` for no label.
`multiple`	Whether the user should be allowed to select and upload multiple files at once. **Does not work on older browsers, including Internet Explorer 9 and earlier.**
`tooltip_text`	The content of the tooltip (string / HTML)
`tooltip_placement`	Where the tooltip should appear relative to its target (top, bottom, left, or right). Defaults to "right"
`trigger`	What action should cause the tooltip to appear? (hover, focus, click, or manual). Defaults to "hover".

`mod_single_cohort_summary_tables_and_plots_server` *wrapper for running multiple mod_single_cohort_summary_tables_server modules*

Description

wrapper for running multiple mod_single_cohort_summary_tables_server modules

Usage

```
mod_single_cohort_summary_tables_and_plots_server(id, maf, cohortName)
```

Arguments

`id`	Internal parameters for shiny.
`maf`	maf object (reactive)
`cohortName`	cohort Name (string, reactive)

`mod_single_cohort_summary_tables_and_plots_ui` *wrapper for running multiple mod_single_cohort_summary_tables_ui modules*

Description

wrapper for running multiple mod_single_cohort_summary_tables_ui modules

Usage

```
mod_single_cohort_summary_tables_and_plots_ui(id)
```

Arguments

`id`	Internal parameters for shiny.

new_maf_dataset_wrapper *Constructor Objects of Class:* **maf_dataset_wrapper**

Description

maf_dataset_wrapper objects store the details of each dataset including the functions to download/load the data and its current status

Usage

```
new_maf_dataset_wrapper(
  maf_data_pool,
  display_name,
  short_name,
  unique_name,
  start_status,
  data_description,
  is_dataset_downloadable,
  function_to_download_data = function() {
     return(NA)
 },
  is_dataset_loadable = TRUE,
  function_to_load_data,
  name_of_data_source = "unknown",
  local_path_to_data = "",
  clinical_data = NA,
  datatype_of_stored_object = "",
  derived_from = NA,
  loaded_data = NA,
  rnaseq_filepath = NA,
  number_of_samples = NA
)
```

Arguments

`maf_data_pool`	A maf_data_pool object. Used to check supplied 'unique_name' is actually going to be unique. If not, characaters are appended to make it truly unique in the context of the supplied dataframe. (maf_data_pool)
`display_name`	Name that the end-user will see (string)
`short_name`	Abbreviated dataset name (string)
`unique_name`	Some unique identifier. (string)

`start_status` one of: "not_downloaded","not_loaded", "ready" (string)

`data_description`
a description of the dataset (string)

`is_dataset_downloadable`
is the dataset downloadable? The alternative is if its packaged in a compressed form with the tool.

`function_to_download_data`
a function that when run will download the required data return the path to which it was downloaded if successful. Will return NA if it failed (function that returns string or NA). If is_dataset_downloadable is false, this is ignored (just use default)

`is_dataset_loadable`
is dataset loadable (flag)

`function_to_load_data`
a function that takes 'local_path_to_data' as its argument and returns the loaded maf object when complete. The function needs to take a single argument, but it doesn't actually have to use it. For example, when loading tcga data using the TCGAmutations package, this function could be **function(filepath)maftools::tcgaLoad**(). This function completely ignores the filepath argument but you NEED to include it anyway (function)

`name_of_data_source`
name of the data source. For Example "USER" or "TCGA" or "PCAWG" (string)

`local_path_to_data`
a path to data which will be configured based on function_to_download_data. If this option is configured ahead of time (string)

`clinical_data` the clinical data of the object

`datatype_of_stored_object`
type of stored data object. Not used for anything right now, just interesting metadata. examples are *.Rds or *.mafs (string)

`derived_from` the maf_dataset_wrapper_object from which the new object was derived. If the dataset was obtained directly from an online source, leave as NA (maf_dataset_wrapper or NA)

`loaded_data` the loaded R object (MAF or NA)

`rnaseq_filepath`
the path to the rnaseq data (string)

`number_of_samples`
the number of samples in the cohort (integer)

Value

an object of type maf_dataset_wrapper (maf_dataset_wrapper)

All Properties

- display_name

- short_name
- unique_name
- status
- data_description
- download_data
- load_data
- name_of_data_source
- local_path_to_data
- datatype_of_stored_object
- loaded_data
- derived_from
- rnaseq_filepath
- number_of_samples

`new_maf_data_pool` *Constructor Objects of Class:* **maf_data_pool**

Description

The maf_data_pool class is simply a list of maf_dataset_wrapper objects.

Usage

```
new_maf_data_pool()
```

Value

object of class maf_data_pool (maf_data_pool)

Examples

```
maf_data_pool <- CRUX:::new_maf_data_pool()
```

`parse_number` *A function that parses numbers from strings*

Description

A function that parses numbers from strings

Usage

```
parse_number(x)
```

Arguments

`x`	character vector you want to parse numbers from

`pcawg_datasets_to_data_pool` *Add all PCAWG datasets to a data pool*

Description

Add all PCAWG datasets to a data pool

Usage

```
pcawg_datasets_to_data_pool(maf_data_pool)
```

Arguments

`maf_data_pool`	the datapool to add pcawg study too (maf_data_pool)

Value

data pool with all pcawg datasets added as maf_dataset_wrappers (maf_data_pool)

Examples

```
CRUX:::pcawg_dataset_to_data_pool(CRUX:::new_maf_data_pool(), "Bone-Cart")
```

pcawg_dataset_to_data_pool

Add PCAWG dataset to a data pool

Description

Add PCAWG dataset to a data pool

Usage

```
pcawg_dataset_to_data_pool(maf_data_pool, pcawg_study_abbreviation)
```

Arguments

`maf_data_pool` the datapool to add pcawg study too (maf_data_pool)

`pcawg_study_abbreviation`
Name of PCAWG study. Use pcawg_available to see options (string)

Value

returns the data pool with the specified dataset added (maf_data_pool)

Examples

```
CRUX:::pcawg_dataset_to_data_pool(CRUX:::new_maf_data_pool(), "Bone-Cart")
```

pcawg_dataset_to_maf_dataset_wrapper

Create dataset wrapper from pcawg dataset

Description

Create dataset wrapper from pcawg dataset

Usage

```
pcawg_dataset_to_maf_dataset_wrapper(maf_data_pool, pcawg_study_abbreviation)
```

Arguments

`maf_data_pool` the data pool to add the dataset wrapper to (maf_data_pool)

`pcawg_study_abbreviation`
Name of PCAWG study. Use pcawg_available to see options (string)

Value

maf_dataset_wrapper object

Examples

```
CRUX:::pcawg_dataset_to_maf_dataset_wrapper(CRUX:::new_maf_data_pool(), "Bone-Cart")
```

`plotforest` *plotforest: Wraps around maftools::forestPlot to more explicitly handle pval VS fdr option and allow both to be passed while onl using the value set by bool 'threshold_on_fdr'*

Description

plotforest: Wraps around maftools::forestPlot to more explicitly handle pval VS fdr option and allow both to be passed while onl using the value set by bool 'threshold_on_fdr'

Usage

```
plotforest(
  mafCompareRes,
  pVal = 0.05,
  fdr = 0.05,
  threshold_on_fdr,
  color = NULL,
  geneFontSize = 1.2,
  lineWidth = 2.2,
  titleSize = 1.2
)
```

Arguments

`mafCompareRes`	results from `mafCompare`
`pVal`	p-value threshold. Default 0.05.
`fdr`	fdr threshold. Default NULL. If provided uses adjusted pvalues (fdr).
`threshold_on_fdr`	should we threshold on FDR (TRUE) or on pValue (false) (flag)
`color`	vector of two colors for the lines. Default 'maroon' and 'royalblue'
`geneFontSize`	Font size for gene symbols. Default 0.8
`lineWidth`	line width for CI bars. Default 1
`titleSize`	font size for titles. Default 1.2

read_maf_flexible *Read MAF from ANNOVAR OR MAF input*

Description

Automatically detects filetype from

Usage

```
read_maf_flexible(
  path_mutations,
  refBuild = NULL,
  path_clindata = NULL,
  filetype = c("AUTO", "ANNOVAR", "MAF")
)
```

Arguments

path_mutations	path to mutation file (string)
refBuild	which reference version is being used
path_clindata	path to clinical annotation file (string)
filetype	what type of filetype do we expect (string). If "AUTO" will automatically guess the filetype using [guess_genomic_mutation_filetype()]

Value

one of MAF, ANNOVAR, VCF, OTHER

read_rnaseq_file *Read RNAseq file*

Description

Read RNAseq file

Usage

```
read_rnaseq_file(rnaseq_file)
```

Arguments

rnaseq_file	(string)

Value

Dataframe containing at least three columns, named "Tumor_Sample_Barcode", "Hugo_Symbol" and "TPM". May optionally include columns named "Fold_Change" and "Transcript" (dataframe)

`run_app` *Run the Shiny Application*

Description

Run the Shiny Application

Usage

```
run_app(options = list(launch.browser = TRUE), ...)
```

Arguments

`options` A list of options to be passed to shinyApp. Common options include launch.browser & port. You must always set ALL options you care about in one go. E.g. if you use options=list(port=3838) we cannot guarantee the launch.browser option will remain true. If multiple options are important to you, force-set them all, e.g. **options(list(launch.browser=TRUE, port = 3838))**

`...` A series of options to be used inside the app.

`shinyInput` *Programmatically create a Shiny input*

Description

Programmatically create a Shiny input

Usage

```
shinyInput(FUN, n, id, label, ...)
```

Arguments

`FUN` function to create the input

`n` number of inputs to be created

`id` ID prefix for each input

`label` label of shiny object (string)

`...` additaional arguments passed to FUN

TCGAmutations_load_with_typed_metadata

Load TCGA data

Description

Load TCGA data with correctly typed clinical data. By default, maftools::tcgaLoad makes all Firehose dataset clinical features character vectors

Usage

```
TCGAmutations_load_with_typed_metadata(study, source = "Firehose")
```

Arguments

`study`	Study names to load. Use `tcgaAvailable` to see available options.
`source`	Source for MAF files. Can be `MC3` or `Firehose`. Default `MC3`. Argument may be abbreviated (M or F)

Value

MAF object with correctly typed clinical.data dataframe (MAF)

tcga_datasets_to_data_pool

Adds all TCGA cohort to data pool

Description

Takes a maf_data_poolobject and adds maf_dataset_wrappers for all tcga datasets available

Usage

```
tcga_datasets_to_data_pool(maf_data_pool, source = "Firehose")
```

Arguments

`maf_data_pool`	the data pool to add the dataset wrapper to (maf_data_pool)
`source`	'MC3' or 'Firehose'. Source of TCGA data to use. See ?maftools::tcgaLoad for details (string)

Value

returns a data pool object with extra dataset added (maf_data_pool)

See Also

Other DataToWrapper: `tcga_dataset_to_data_pool()`, `tcga_dataset_to_maf_dataset_wrapper()`, `user_data_filepath_to_class_maf_dataset_wrapper()`, `user_to_dataset_to_data_pool()`

Examples

```
CRUX:::tcga_datasets_to_data_pool(CRUX:::new_maf_data_pool(), source = "Firehose")
```

`tcga_dataset_to_data_pool` *Add TCGA cohort to data pool*

Description

Takes a TCGA study abbreviation, creates a tcga_dataset_to_maf_dataset_wrapper object for the relevant dataset and adds the wrapper to the maf_data_pool

Usage

```
tcga_dataset_to_data_pool(
  tcga_study_abbreviation,
  maf_data_pool,
  source = "Firehose"
)
```

Arguments

`tcga_study_abbreviation`	a TCGA cohort abbreviation (see maftools::tcgaAvailable()) (string)
`maf_data_pool`	the data pool to add the dataset wrapper to (maf_data_pool)
`source`	'MC3' or 'Firehose'. Source of TCGA data to use. See ?maftools::tcgaLoad for details (string)

Value

returns a data pool object with extra dataset added (maf_data_pool)

See Also

Other DataToWrapper: `tcga_dataset_to_maf_dataset_wrapper()`, `tcga_datasets_to_data_pool()`, `user_data_filepath_to_class_maf_dataset_wrapper()`, `user_to_dataset_to_data_pool()`

Examples

```
CRUX:::tcga_dataset_to_data_pool("ACC", CRUX:::new_maf_data_pool())
```

tcga_dataset_to_maf_dataset_wrapper

TCGA to maf_dataset_wrapper

Description

Takes a TCGA cohort abbreviation

Usage

```
tcga_dataset_to_maf_dataset_wrapper(
  maf_data_pool,
  tcga_study_abbreviation,
  source = "Firehose"
)
```

Arguments

maf_data_pool	the data pool to add the dataset wrapper to (maf_data_pool)
tcga_study_abbreviation	a TCGA cohort abbreviation (see maftools::tcgaAvailable()) (string)
source	'MC3' or 'Firehose'. Source of TCGA data to use. See ?maftools::tcgaLoad for details (string)

Value

functions and values associated with specified TCGA cohort (maf_dataset_wrapper)

See Also

Other DataToWrapper: tcga_dataset_to_data_pool(), tcga_datasets_to_data_pool(), user_data_filepath_to_c
user_to_dataset_to_data_pool()

Examples

```
ACC_maf_dataset_wrapper <- CRUX:::tcga_dataset_to_maf_dataset_wrapper(
  CRUX:::new_maf_data_pool(),
  "ACC"
)
```

text_is_non_zero_string

Input is non-empty string

Description

must be in server

Usage

```
text_is_non_zero_string(text, test_custom_function = NULL)
```

Arguments

text	input text to assert is a string (string)
test_custom_function	a function which takes the value of the textInput and returns TRUE/FALSE depending on whether it meets expectations. If supplied, textInput value must be a non-emptystring AND pass the custom function's test(function)

Value

True if valid text is found in. FALSE if it is not (flag)

user_data_filepath_to_class_maf_dataset_wrapper

Title

Description

Title

Usage

```
user_data_filepath_to_class_maf_dataset_wrapper(
  filepath,
  clinicalData = NA,
  maf_data_pool,
  display_name,
  short_name,
  description = "User specified file",
  data_source = "USER",
  loaded_data = NA
)
```

Arguments

`filepath` tab delimited MAF file. File can also be gz compressed. Required. Alternatively, you can also provide already read MAF file as a dataframe.

`clinicalData` Clinical data associated with each sample/Tumor_Sample_Barcode in MAF. Could be a text file or a data.frame. Default NULL.

`maf_data_pool` A maf_data_pool object. Used to check supplied 'unique_name' is actually going to be unique. If not, characaters are appended to make it truly unique in the context of the supplied dataframe. (maf_data_pool)

`display_name` Name that the end-user will see (string)

`short_name` Abbreviated dataset name (string)

`description` description (string)

`data_source` data source (string)

`loaded_data` the loaded R object (MAF or NA)

Value

maf_dataset_wrapper

See Also

Other DataToWrapper: `tcga_dataset_to_data_pool()`, `tcga_dataset_to_maf_dataset_wrapper()`, `tcga_datasets_to_data_pool()`, `user_to_dataset_to_data_pool()`

`user_to_dataset_to_data_pool` *Add User Data to data pool*

Description

Takes a filepath to user dataset + some medata and creates a maf_dataset_wrapper object, then adds the wrapper to the maf_data_pool

Usage

```
user_to_dataset_to_data_pool(
  maf_data_pool,
  filepath,
  clinicalData = NA,
  display_name,
  short_name,
  description = "User specified file",
  data_source = "USER",
  loaded_data = NA
)
```

Arguments

`maf_data_pool`	the data pool to add the dataset wrapper to (maf_data_pool)
`filepath`	path to maf file
`clinicalData`	Clinical data associated with each sample/Tumor_Sample_Barcode in MAF. Could be a text file or a data.frame. Default NULL.
`display_name`	Name that the end-user will see (string)
`short_name`	Abbreviated dataset name (string)
`description`	dataset description
`data_source`	data source
`loaded_data`	the loaded R object (MAF or NA)

Value

returns a data pool object with extra dataset added (maf_data_pool)

See Also

Other DataToWrapper: `tcga_dataset_to_data_pool()`, `tcga_dataset_to_maf_dataset_wrapper()`, `tcga_datasets_to_data_pool()`, `user_data_filepath_to_class_maf_dataset_wrapper()`

Index