Package 'tidycwl'

May 3, 2022

Type Package

Title Tidy Common Workflow Language Tools and Workflows

Version 1.0.7

Maintainer Soner Koc <soner.koc@sevenbridges.com>

Description The Common Workflow Language <https://www.commonwl.org/> is an open standard for describing data analysis workflows. This package takes the raw Common Workflow Language workflows encoded in JSON or 'YAML' and turns the workflow elements into tidy data frames or lists. A graph representation for the workflow can be constructed and visualized with the parsed workflow inputs, outputs, and steps. Users can embed the visualizations in their 'Shiny' applications, and export them as HTML files or static images.

License AGPL-3

SystemRequirements PhantomJS (https://phantomjs.org), pandoc (>= 1.12.3) - https://pandoc.org.

VignetteBuilder knitr

URL https://sbg.github.io/tidycwl/, https://github.com/sbg/tidycwl

BugReports https://github.com/sbg/tidycwl/issues

Encoding UTF-8

Imports jsonlite, yaml, dplyr, magrittr, visNetwork, htmlwidgets, webshot

Suggests knitr, rmarkdown, shiny

RoxygenNote 7.1.2

NeedsCompilation no

Author Soner Koc [aut, cre] (<https://orcid.org/0000-0002-0772-6600>), Jeffrey Grover [aut] (<https://orcid.org/0000-0001-6246-1767>), Nan Xiao [aut] (<https://orcid.org/0000-0002-0250-5673>), Dennis Dean [aut] (<https://orcid.org/0000-0002-7621-9717>), Seven Bridges Genomics [cph, fnd]

Repository CRAN

Date/Publication 2022-05-03 07:00:02 UTC

R topics documented:

`export_html` *Export the workflow plot as HTML*

Description

Export the workflow plot as HTML

Usage

```
export_html(g, file, ...)
```

Arguments

g	Plot rendered by visualize_graph.
file	File to save HTML into.
...	Additional parameters for visSave.

Value

HTML file path

Examples

```
file_html <- tempfile(fileext = ".html")
flow <- system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>% read_cwl_json()
if (rmarkdown::pandoc_available("1.12.3")) {
  get_graph(
    flow %>% parse_inputs(),
    flow %>% parse_outputs(),
    flow %>% parse_steps()
  ) %>%
    visualize_graph() %>%
    export_html(file_html)
}
```

---

export_image *Export the workflow plot as PNG, JPEG, or PDF files*

Description

Export the workflow plot as PNG, JPEG, or PDF files

Usage

```
export_image(file_html, file_image, ...)
```

Arguments

file_html	File path to the HTML exported by export_html.
file_image	File path to the output image. Should end with .png, .pdf, or .jpeg.
...	Additional parameters for webshot.

Value

Image file path

Note

This function uses webshot to take a screenshot for the rendered HTML of the graph. It requires PhantomJS installed in your system. You can use install_phantomjs to install it.

Examples

```
if (interactive()) {
  file_png <- tempfile(fileext = ".png")
 flow <- system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>% read_cwl_json()
  get_graph(
    flow %>% parse_inputs(),
    flow %>% parse_outputs(),
    flow %>% parse_steps()
  ) %>%
    visualize_graph() %>%
    export_html(tempfile(fileext = ".html")) %>%
    export_image(file_png, vwidth = 2000, vheight = 3000, selector = "div.vis-network")
}
```

`get_cwl_version` *Get CWL version*

Description

Get CWL version

Usage

```
get_cwl_version(x)
```

Arguments

x	CWL object

Value

CWL version number

Examples

```
system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  get_cwl_version()

system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  get_cwl_version()
```

get_edges *Get edges in a CWL workflow into a data frame*

Description

Get edges in a CWL workflow into a data frame

Usage

```
get_edges(outputs, steps)
```

Arguments

outputs	Parsed outputs
steps	Parsed steps

Value

Data frame containing edge information

Examples

```
# edges represented by a dictionary
flow <- system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>% read_cwl_json()
get_edges(
  flow %>% parse_outputs(),
  flow %>% parse_steps()
) %>% str()

# edges represented by a list
try(
 flow <- system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>% read_cwl_yaml()
)
try(
  get_edges(
    flow %>% parse_outputs(),
    flow %>% parse_steps()
  ) %>% str()
)
```

get_graph *Get the CWL workflow graph*

Description

Get the CWL workflow graph as a list of two data frames: a data frame of nodes and a data frame of edges.

Usage

```
get_graph(inputs, outputs, steps)
```

Arguments

inputs	Parsed inputs
outputs	Parsed outputs
steps	Parsed steps

Value

List of two data frames containing node and edge information

Examples

```
# sbg:draft2
flow <- system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>% read_cwl_json()
get_graph(
  flow %>% parse_inputs(),
  flow %>% parse_outputs(),
  flow %>% parse_steps()
) %>% str()

# v1.0
flow <- system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>% read_cwl_json()
get_graph(
  flow %>% parse_inputs(),
  flow %>% parse_outputs(),
  flow %>% parse_steps()
) %>% str()
```

get_inputs_id *Get ID for inputs*

Description

Get ID for inputs

Usage

```
get_inputs_id(inputs)
```

Arguments

inputs	Parsed inputs

Value

Vector of input IDs

Examples

```
# inputs represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_inputs() %>%
  get_inputs_id()

# inputs represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_inputs() %>%
  get_inputs_id()
```

get_inputs_label *Get label for inputs*

Description

Get label for inputs

Usage

```
get_inputs_label(inputs)
```

Arguments

inputs	Parsed inputs

Value

Vector of input labels

Examples

```
# inputs represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_inputs() %>%
  get_inputs_label()

# inputs represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_inputs() %>%
  get_inputs_label()
```

---

get_nodes *Get nodes in a CWL workflow into a data frame*

Description

Get nodes in a CWL workflow into a data frame

Usage

```
get_nodes(inputs, outputs, steps)
```

Arguments

inputs	Parsed inputs
outputs	Parsed outputs
steps	Parsed steps

Value

Data frame containing node information

Examples

```
flow <- system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>% read_cwl_json()
get_nodes(
  flow %>% parse_inputs(),
  flow %>% parse_outputs(),
  flow %>% parse_steps()
) %>% str()
```

get_outputs_id *Get ID for outputs*

Description

Get ID for outputs

Usage

```
get_outputs_id(outputs)
```

Arguments

outputs	Parsed outputs

Value

Vector of output IDs

Examples

```
# inputs represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_outputs() %>%
  get_outputs_id()

# inputs represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_outputs() %>%
  get_outputs_id()
```

get_outputs_label *Get label for outputs*

Description

Get label for outputs

Usage

```
get_outputs_label(outputs)
```

Arguments

outputs	Parsed outputs

Value

Vector of output labels

Examples

```
# inputs represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_outputs() %>%
  get_outputs_label()

# inputs represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_outputs() %>%
  get_outputs_label()
```

---

get_steps_doc *Get documentation/description for steps*

Description

Get documentation/description for steps

Usage

```
get_steps_doc(steps)
```

Arguments

steps	Steps object parsed by parse_steps

Value

Vector of step documentation/descriptions

Examples

```
# steps represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_steps() %>%
  get_steps_doc()

# steps represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_steps() %>%
  get_steps_doc()
```

get_steps_id *Get ID for steps*

Description

Get ID for steps

Usage

```
get_steps_id(steps)
```

Arguments

steps	Steps object parsed by parse_steps

Value

Vector of step IDs

Examples

```
# steps represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_steps() %>%
  get_steps_id()

# steps represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_steps() %>%
  get_steps_id()
```

get_steps_label *Get label for steps*

Description

Get label for steps

Usage

```
get_steps_label(steps)
```

Arguments

steps	Steps object parsed by parse_steps

Value

Vector of step labels

Examples

```
# steps represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_steps() %>%
  get_steps_label()

# steps represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_steps() %>%
  get_steps_label()
```

---

get_steps_revision *Get revision number for steps*

Description

Get revision number for steps

Usage

```
get_steps_revision(steps)
```

Arguments

steps	Steps object parsed by parse_steps

Value

Vector of step revision numbers

Examples

```
# steps represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_steps() %>%
  get_steps_revision()

# steps represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_steps() %>%
  get_steps_revision()
```

get_steps_version *Get toolkit version for steps*

Description

Get toolkit version for steps

Usage

```
get_steps_version(steps)
```

Arguments

steps	Steps object parsed by parse_steps

Value

Vector of step toolkit versions

Examples

```
# steps represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_steps() %>%
  get_steps_version()

# steps represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_steps() %>%
  get_steps_version()
```

is_cwl *Is this a CWL object?*

Description

Is this a CWL object?

Usage

```
is_cwl(x)
```

Arguments

x	any object

Value

Logical. TRUE if it is a CWL object, FALSE if not.

Examples

```
system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_cwl()
```

---

is_draft2 *Is this CWL draft2?*

Description

Is this CWL draft2?

Usage

```
is_draft2(x)
```

Arguments

x	CWL object

Value

Logical. TRUE if it is a CWL draft2 object, FALSE if not.

Examples

```
system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_draft2()

system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_draft2()
```

is_tool *Is this a CWL command line tool?*

Description

Is this a CWL command line tool?

Usage

```
is_tool(x)
```

Arguments

x	CWL object

Value

Logical. TRUE if it is a CWL command line tool (instead of a workflow), FALSE if not.

Examples

```
system.file("cwl/sbg/tool/bwa-mem.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_tool()

system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_tool()
```

is_v1.0 *Is this CWL v1.0?*

Description

Is this CWL v1.0?

Usage

```
is_v1.0(x)
```

Arguments

x	CWL object

Value

Logical. TRUE if it is a CWL v1.0 object, FALSE if not.

Examples

```
system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_v1.0()

system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_v1.0()
```

---

is_v1.1 *Is this CWL v1.1?*

Description

Is this CWL v1.1?

Usage

```
is_v1.1(x)
```

Arguments

x	CWL object

Value

Logical. TRUE if it is a CWL v1.1 object, FALSE if not.

Examples

```
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_v1.1()
```

---

is_workflow *Is this a CWL workflow?*

Description

Is this a CWL workflow?

Usage

```
is_workflow(x)
```

Arguments

x	CWL object

Value

Logical. TRUE if it is a CWL workflow (instead of a command line tool), FALSE if not.

Examples

```
system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_workflow()

system.file("cwl/sbg/tool/bwa-mem.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  is_workflow()
```

---

parse_cwl *Parse a CWL workflow*

Description

Parse a CWL workflow and return the metadata, inputs, outputs, and steps in a list.

Usage

```
parse_cwl(x)
```

Arguments

x	CWL object

Value

List of CWL metadata, inputs, outputs, and steps

Examples

```
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_cwl() %>%
  names()
```

parse_inputs *Parse the inputs of the CWL workflow into a data frame*

Description

Parse the inputs of the CWL workflow into a data frame

Usage

```
parse_inputs(x, simplify = TRUE)
```

Arguments

x	CWL object
simplify	Simplify the list as a data frame?

Value

List or data frame of inputs

Examples

```
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_inputs() %>%
  names()

system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_inputs() %>%
  names()
```

parse_meta *Parse the metadata in the CWL workflow*

Description

Parse the metadata in the CWL workflow

Usage

```
parse_meta(x)
```

Arguments

x	CWL object

Value

List of CWL metadata

Examples

```
system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  parse_meta()
```

`parse_outputs` *Parse the outputs of the CWL workflow into a data frame*

Description

Parse the outputs of the CWL workflow into a data frame

Usage

```
parse_outputs(x, simplify = TRUE)
```

Arguments

`x`	CWL object
`simplify`	Simplify the list as a data frame?

Value

List or data frame of outputs

Examples

```
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_outputs() %>%
  names()

system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_outputs() %>%
  names()
```

parse_steps *Parse the steps of the CWL workflow into a data frame*

Description

Parse the steps of the CWL workflow into a data frame

Usage

```
parse_steps(x)
```

Arguments

x	CWL object

Value

List or data frame of steps

Examples

```
# steps represented by a dictionary
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json() %>%
  parse_steps() %>%
  nrow()

# steps represented by a list
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml() %>%
  parse_steps() %>%
  length()
```

parse_type *Parse CWL content type*

Description

Parse CWL content type

Usage

```
parse_type(x)
```

Arguments

x	CWL object

Value

CWL content type (Workflow or CommandLineTool)

Examples

```
system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  parse_type()

system.file("cwl/sbg/tool/bwa-mem.json", package = "tidycwl") %>%
  read_cwl(format = "json") %>%
  parse_type()
```

`print.cwl` *Print CWL objects*

Description

Print a brief summary of the CWL object.

Usage

```
## S3 method for class 'cwl'
print(x, ...)
```

Arguments

`x`	An object of class `cwl`.
`...`	Additional parameters for `print` (not used).

Value

The input `cwl` object.

Examples

```
path <- system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl")
flow <- read_cwl(path, format = "json")
flow
```

read_cwl *Read a CWL file into a list*

Description

Read a CWL file into a list

Usage

```
read_cwl(file, format = c("json", "yaml"))
```

Arguments

file	A file path, character string, or connection.
format	CWL storage format. "json" or "yaml".

Value

List representation of the input CWL

Examples

```
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl(format = "json")

system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl(format = "yaml")
```

read_cwl_json *Read a CWL file (JSON format) into a list*

Description

Read a CWL file (JSON format) into a list

Usage

```
read_cwl_json(file)
```

Arguments

file	A file path, JSON string, or connection.

Value

List representation of the input CWL

Examples

```
system.file("cwl/sbg/workflow/rnaseq-salmon.json", package = "tidycwl") %>%
  read_cwl_json()
```

---

read_cwl_yaml *Read a CWL file (YAML format) into a list*

Description

Read a CWL file (YAML format) into a list

Usage

```
read_cwl_yaml(file)
```

Arguments

file	A file path, YAML string, or connection.

Value

List representation of the input CWL

Examples

```
system.file("cwl/sbg/workflow/rnaseq-salmon.cwl", package = "tidycwl") %>%
  read_cwl_yaml()
```

---

tidycwl_shiny *Shiny bindings for tidycwl*

Description

Output and renderer functions for using tidycwl within Shiny apps and interactive R Markdown documents.

Usage

```
cwl_output(outputId, width = "100%", height = "600px")

render_cwl(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like "100%", "600px", "auto") or a number, which will be coerced to a string and have "px" appended.
expr	An expression that generates a CWL graph
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

Value

An output or render function that enables the use of the widget within Shiny apps.

Examples

```
if (interactive()) {
  library("shiny")
  library("tidycwl")

  cwl_folder <- system.file("cwl/sbg/workflow/", package = "tidycwl")
  file_all <- list.files(cwl_folder)
  cwl_name <- file_all[which(tools::file_ext(file_all) == "json")]

  ui <- fluidPage(
    selectInput("cwl_file", "Select a CWL file:", cwl_name),
    cwl_output("cwl_plot", height = "800px")
  )

  server <- function(input, output, session) {
    output$cwl_plot <- render_cwl({
      flow <- paste0(cwl_folder, input$cwl_file) %>% read_cwl_json()
      get_graph(
        flow %>% parse_inputs(),
        flow %>% parse_outputs(),
        flow %>% parse_steps()
      ) %>% visualize_graph()
    })
  }

  shinyApp(ui, server)
}
```

---

visualize_graph *Visualize the CWL workflow*

Description

Visualize the CWL workflow

Usage

```
visualize_graph(
  g,
  hierarchical = TRUE,
  direction = "LR",
  separation = 300,
  palette = c("#C3C3C3", "#FF8F00", "#00AAA8"),
  width = "100%",
  height = 600
)
```

Arguments

g	Graph generated by get_graph.
hierarchical	Enable the hierarchical layout? Default is TRUE.
direction	Direction of the hierarchical layout. Options include "LR", "RL", "UD", and "DU" (up-down, down-up, left-right, right-left). Default is "LR".
separation	Level separation parameter from visHierarchicalLayout.
palette	Three-color palette for inputs, outputs, and steps.
width	Canvas width, see visNetwork. Default is "100%".
height	Canvas height, see visNetwork. Default is 600.

Value

A visNetwork output.

Examples

```
flow <- system.file("cwl/sbg/workflow/gatk4-wgs.json", package = "tidycwl") %>% read_cwl_json()
get_graph(
  flow %>% parse_inputs(),
  flow %>% parse_outputs(),
  flow %>% parse_steps()
) %>% visualize_graph()
```

Index