Package: CyFj11 (via r-universe)

November 25, 2026

Type Package

Title Parse FlowJo V11 Workspace Files and Export to FlowJo V10

Version 1.0.0

Description Provides functions to parse FlowJo v11 workspace files (.flowjo) and extract gating information, populations, and associated FCS files. Includes utilities for exporting data to FlowJo v10 workspace format. The package enables bidirectional data flow between FlowJo and R/Bioconductor, facilitating reproducible analysis workflows.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.5.0)

Imports jsonlite, dplyr, flowWorkspace, flowCore, readr, rlang, tibble, tidyr, xml2, methods, scales, magrittr, stats, utils

Suggests testthat (>= 3.0.0), Biobase, CytoML, FlowSOM, flowWorkspaceData, ggcyto, knitr, mockery, BiocStyle, qpdf, rmarkdown

VignetteBuilder knitr

biocViews FlowCytometry, DataImport, Infrastructure, Preprocessing

URL https://github.com/C3BI-pasteur-fr/CyFj11

BugReports https://github.com/C3BI-pasteur-fr/CyFj11/issues

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:57:05 UTC

RemoteUrl https://github.com/bioc/CyFj11

RemoteRef RELEASE_3_24

RemoteSha 4472bd9d5f4ea32ea69c416f31b362b73c43e7c3

Contents

export_flowjo10_workspace 2
fj11_schema_version 3
fj11_to_gatingset 4
get_sanitize_slashes 7
get_verbose 7
load_example_workspace 8
pretty_print_flowjo 8
read_flowjo11_workspace 9
set_sanitize_slashes 9
set_verbose 10

Index **11**

`export_flowjo10_workspace`

Export GatingSet to FlowJo v10 Workspace

Description

Export GatingSet to FlowJo v10 Workspace

Usage

```
export_flowjo10_workspace(
    gating_set,
    output_path,
    workspace_name = NULL,
    fcs_root = NULL,
    overwrite = FALSE
)
```

Arguments

`gating_set`	A GatingSet object to export. A named list of GatingSets, as returned by `fj11_to_gatingset`, is also accepted and merged with `flowWorkspace::merge_list_to_gs()` before export.
`output_path`	Path where the .xml file should be created
`workspace_name`	Optional name for the workspace
`fcs_root`	Optional base directory for FCS files. If provided, FCS files will be written there and URIs in the WSP will reference that location. If NULL, uses the actual paths found in the GatingSet.
`overwrite`	Logical. When `fcs_root` is supplied and FCS files already exist in that directory, `FALSE` (default) aborts with an error; `TRUE` silently replaces existing files.

Value

Logical indicating success

Examples

```
# Export a GatingSet to FlowJo v10 XML format
ws_path  <- system.file("extdata", "min_test.flowjo", package = "CyFj11")
fcs_path <- system.file("extdata", package = "CyFj11")
ws <- read_flowjo11_workspace(ws_path)
gs <- fj11_to_gatingset(ws, group_name = 1, path = fcs_path)
out_file <- tempfile(fileext = ".xml")
export_flowjo10_workspace(gs, out_file)
file.exists(out_file)
```

`fj11_schema_version` *Get the schema version of a FlowJo v11 workspace*

Description

Get the schema version of a FlowJo v11 workspace

Usage

```
fj11_schema_version(x)
```

Arguments

`x`	A `flowjo11_workspace` (from `read_flowjo11_workspace`), a character string such as `"4.0.0"`, a `numeric_version`, or `NULL`.

Value

A `numeric_version`, or `NULL` if the version is unknown.

Examples

```
ws <- read_flowjo11_workspace(system.file("extdata", "min_test.flowjo", package = "CyFj11"))
fj11_schema_version(ws)
fj11_schema_version("4.0.0") >= "4.0.0"
```

`fj11_to_gatingset` *Convert FlowJo v11 Workspace to GatingSet*

Description

Parses a FlowJo v11 workspace and creates a GatingSet object with gating hierarchy, compensation, and transformations. Supports biexponential (Biex) transformations which are mapped to logicle transforms.

Usage

```
fj11_to_gatingset(
    fj11_workspace,
    group_name = NULL,
    subset = list(),
    execute = TRUE,
    path = NULL,
    cytoset = NULL,
    backend_dir = tempdir(),
    backend = c("h5", "tile"),
    include_gates = TRUE,
    compensation = NULL,
    additional.keys = "$TOT",
    additional.sampleID = FALSE,
    keywords = character(),
    keyword.ignore.case = FALSE,
    channel.ignore.case = FALSE,
    extend_val = 0,
    extend_to = -4000,
    leaf.bool = TRUE,
    include_empty_tree = FALSE,
    correct_faulty_gate = 0,
    transform = TRUE,
    use_transformed_coords = transform,
    max_search_depth = 5,
    stop_on_multiple = TRUE,
    mc.cores = 1,
    strip_comp_prefix = TRUE,
    sanitize_slashes = TRUE,
    ...
)
```

Arguments

`fj11_workspace`	List returned by `read_flowjo11_workspace()` containing the complete workspace structure

`group_name` Character or numeric. Name or index of the sample group to import. If NULL, will list available groups interactively.

`subset` Numeric vector, character vector of FCS filenames, or expression to filter samples. See Details.

`execute` Logical. Should gates be executed immediately? Default TRUE. If FALSE, creates GatingSet structure without computing cell counts.

`path` Character. Root directory to search for FCS files. Required if FCS files are not in workspace location.

`cytoset` A cytoset object providing alternative data source (instead of FCS files). Useful for preprocessed data.

`backend_dir` Directory for storing h5 backend data. Default is tempdir().

`backend` Character. Storage backend, either "h5" or "tile". Default "h5".

`include_gates` Logical. Should gates be imported? Default TRUE.

`compensation` Compensation object, matrix, data.frame, or named list of these. Overrides workspace compensation. Names should match sample GUIDs.

`additional.keys` Character vector of FCS keywords to include in sample GUID (concatenated with filename). Default is "$TOT".

`additional.sampleID` Logical. Include FlowJo sample ID in GUID? Default FALSE.

`keywords` Character vector of keywords to extract as pData.

`keyword.ignore.case` Logical. Case-insensitive keyword matching? Default FALSE.

`channel.ignore.case` Logical. Case-insensitive channel matching? Default FALSE.

`extend_val` Numeric. Threshold for extending gate coordinates. Default 0.

`extend_to` Numeric. Value to extend gates to. Default -4000.

`leaf.bool` Logical. Compute leaf boolean gates? Default TRUE.

`include_empty_tree` Logical. Include samples without gates? Default FALSE.

`correct_faulty_gate` Numerical, zero = don't correct, otherwise use this value as transformation max value and Try to correct faulty gates instead of erroring? Default TRUE.

`transform` Logical. Apply transformations? Default TRUE.

`use_transformed_coords` Logical. Keep gate coordinates in FlowJo display (transformed) space rather than raw space? Default is the value of `transform`. Only relevant when `transform = TRUE`.

`max_search_depth` Integer. Maximum directory depth when searching for FCS files. Default 5.

`stop_on_multiple` Logical. Stop if multiple FCS files match a sample? When TRUE, stops on both missing files and multiple matches. When FALSE, only stops on missing files and continues with multiple matches. Default TRUE.

`mc.cores` Integer. Number of cores for parallel processing. Default 1.

`strip_comp_prefix`
Logical. Strip "Comp-" prefix from gate parameter names when adding populations? Default TRUE. Set to FALSE if compensation has already been applied and gate names should match the compensated parameter names.

`sanitize_slashes`
Logical. Replace "/" with "_" in parameter names? Default TRUE (matches flowCore behavior). Set to FALSE if you want to preserve "/" in marker names (e.g., "CD3/CD4" stays as-is).

`...` Additional arguments passed to FCS parser.

Details

The `subset` argument can be:

Numeric vector: Sample indices to import

Character vector: FCS filenames to import

Expression: Logical expression to filter pData (requires `keywords` argument)

FlowJo v11 stores data differently than v10:

populationDefinitions: Master gate definitions

desyncTable: Per-sample gate variations (tailored gates)

groups: Sample groupings

dataSources: FCS file references

Value

A named list of `GatingSet` objects, one per sample. Each element contains the full gating hierarchy, gates, compensation, and transformations for that sample. To obtain a single merged `GatingSet` for downstream analysis, call `flowWorkspace::merge_list_to_gs(gsList)` on the returned list.

Examples

```
ws_path  <- system.file("extdata", "min_test.flowjo", package = "CyFj11")
fcs_path <- system.file("extdata", package = "CyFj11")
ws       <- read_flowjo11_workspace(ws_path)
gsList   <- fj11_to_gatingset(ws, group_name = 1, path = fcs_path)
length(gsList)  # Number of GatingSets created
```

get_sanitize_slashes *Get slash sanitization mode*

Description

Check whether "/" characters in parameter names are being replaced with "_".

Usage

```
get_sanitize_slashes()
```

Value

Logical. TRUE if "/" is replaced with "_", FALSE if preserved

Examples

```
# Check current setting
is_sanitize_slashes <- get_sanitize_slashes()
```

get_verbose *Get verbose mode*

Description

Check whether verbose output is currently enabled.

Usage

```
get_verbose()
```

Value

Logical. TRUE if verbose output is enabled, FALSE otherwise

Examples

```
# Check current verbose setting
is_verbose <- get_verbose()
is_verbose  # Should be FALSE by default
```

load_example_workspace *Load Example FlowJo v11 Workspace*

Description

Loads the example FlowJo v11 workspace from inst/extdata/min_test.flowjo for use in tests and examples. This workspace contains a simple gating hierarchy (Ungated -> Lymphocytes -> CD8, CD3 subset) and references sample04.fcs.

Usage

```
load_example_workspace()
```

Value

A flowjo11_workspace object

Examples

```
ws <- load_example_workspace()
length(ws$samples)  # Number of samples
```

pretty_print_flowjo *Pretty Print FlowJo Workspace*

Description

This function takes a FlowJo v11 workspace file (.flowjo) and extracts the JSON content, then pretty prints it to a text file with "_pretty.json" suffix.

Usage

```
pretty_print_flowjo(flowjo_file)
```

Arguments

flowjo_file	Path to the FlowJo workspace file (.flowjo)

Value

Path to the created pretty printed JSON file

Examples

```
# Pretty print a FlowJo workspace to readable JSON
ws_path <- system.file("extdata", "min_test.flowjo", package = "CyFj11")
out_file <- pretty_print_flowjo(ws_path)
file.exists(out_file)  # Verify output was created
# Clean up
unlink(out_file)
```

`read_flowjo11_workspace` *Read FlowJo v11 Workspace*

Description

Main wrapper function to read and parse FlowJo v11 workspace files.

Usage

```
read_flowjo11_workspace(workspace_path)
```

Arguments

`workspace_path` Path to the FlowJo workspace file (.flowjo)

Value

Parsed workspace object containing manifest and JSON data

Examples

```
ws_path <- system.file("extdata", "min_test.flowjo", package = "CyFj11")
ws <- read_flowjo11_workspace(ws_path)
length(ws$samples)  # Number of samples in workspace
names(ws$groups)    # Group names
```

`set_sanitize_slashes` *Set parameter name slash sanitization behavior*

Description

Control whether "/" characters in parameter/marker names are automatically replaced with "*". This is important because flowCore sanitizes parameter names by replacing "/" with "*" when reading FCS files.

Usage

```
set_sanitize_slashes(v)
```

Arguments

v	Logical. TRUE to replace "/" with "_" (default), FALSE to preserve "/"

Details

By default, this is TRUE to match flowCore's behavior, ensuring that gate parameter names match the flowFrame column names. Set to FALSE if you want to preserve "/" in marker names (e.g., "CD3/CD4" stays as-is).

Value

Invisible NULL. Called for side effects.

Examples

```
# Preserve "/" in marker names
set_sanitize_slashes(FALSE)
# Use default behavior (replace "/" with "_")
set_sanitize_slashes(TRUE)
```

---

set_verbose *Set verbose mode*

Description

Enable or disable verbose output for debugging and monitoring package operations. When enabled, the package will print detailed information about its operations, including file processing, gate conversion, and data extraction steps.

Usage

```
set_verbose(v)
```

Arguments

v	Logical. TRUE to enable verbose output, FALSE to disable it

Value

Invisible NULL. Called for side effects.

Examples

```
set_verbose(TRUE)
set_verbose(FALSE)
```

Index

export_flowjo10_workspace, 2

fj11_schema_version, 3
fj11_to_gatingset, 2, 4

get_sanitize_slashes, 7
get_verbose, 7

load_example_workspace, 8

pretty_print_flowjo, 8

read_flowjo11_workspace, *3*, 9

set_sanitize_slashes, 9
set_verbose, 10