Package: PhysioOpenSim (via r-universe)

May 16, 2026

Title Native OpenSim C++ Integration for PhysioExperiment

Version 0.1.0

Description Provides a native OpenSim C++ bridge for R via Rcpp. The package exposes OpenSim model-level operations to support biomechanics workflows in the Physio ecosystem. OpenSim linkage is optional at build time; when OpenSim is not detected, the package installs with informative runtime errors for OpenSim-dependent calls.

Depends R (>= 4.2)

Imports Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.2.0), knitr, rmarkdown

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

SystemRequirements C++17, OpenSim (optional)

Config/testthat/edition 3

Repository https://x-biosignal.r-universe.dev

Date/Publication 2026-03-16 11:31:42 UTC

RemoteUrl https://github.com/x-biosignal/PhysioOpenSim

RemoteRef HEAD

RemoteSha 6271a3d7e1bff982f693f5a713f9def888f9b994

Contents

`opensimAvailable` *Check Whether Native OpenSim Support Is Available*

Description

Returns whether this package was compiled with OpenSim headers and libraries linked into the native code.

Usage

```
opensimAvailable()
```

Value

`TRUE` if OpenSim native support is available, otherwise `FALSE`.

Examples

```
opensimAvailable()
```

`opensimBuildConfig` *Show Build Configuration for OpenSim Native Bridge*

Description

Show Build Configuration for OpenSim Native Bridge

Usage

```
opensimBuildConfig()
```

Value

A list containing compilation status and build metadata.

Examples

```
cfg <- opensimBuildConfig()
cfg$package
cfg$opensim_enabled
```

`opensimCLIAvailable` *Check Whether OpenSim CLI Is Available*

Description

Check Whether OpenSim CLI Is Available

Usage

```
opensimCLIAvailable(cli = NULL)
```

Arguments

`cli`	Optional command or absolute path.

Value

`TRUE` if CLI command is resolvable, otherwise `FALSE`.

Examples

```
opensimCLIAvailable()
```

opensimCLIPath *Resolve OpenSim CLI Command Path*

Description

Resolves the command path used to invoke OpenSim CLI.

Usage

```
opensimCLIPath(cli = NULL)
```

Arguments

cli	Optional command or absolute path. Defaults to `OPENSIM_CLI` environment variable, then `"opensim-cmd"`.

Value

A character scalar path to executable command.

Examples

```
## Not run:
opensimCLIPath()
opensimCLIPath("/usr/local/bin/opensim-cmd")

## End(Not run)
```

opensimFinalizeConnections *Finalize OpenSim Model Connections*

Description

Finalize OpenSim Model Connections

Usage

```
opensimFinalizeConnections(model)
```

Arguments

model	A `PhysioOpenSimModel` external pointer.

Value

The input `model` handle (invisibly).

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimFinalizeConnections(model)

## End(Not run)
```

`opensimLoadModel` *Load an OpenSim Model as a Native External Pointer*

Description

Load an OpenSim Model as a Native External Pointer

Usage

```
opensimLoadModel(path)
```

Arguments

`path`	Path to an OpenSim model file (`.osim`).

Value

A `PhysioOpenSimModel` external pointer handle.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelName(model)

## End(Not run)
```

`opensimModelComponents` *List Core OpenSim Model Component Names*

Description

List Core OpenSim Model Component Names

Usage

```
opensimModelComponents(model)
```

Arguments

model	A PhysioOpenSimModel external pointer.

Value

Named list with vectors for body, joint, marker, muscle, and coordinate names.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
comps <- opensimModelComponents(model)
comps$bodies
comps$muscles

## End(Not run)
```

---

opensimModelInitialize

Initialize OpenSim Model System State

Description

Initialize OpenSim Model System State

Usage

```
opensimModelInitialize(model)
```

Arguments

model	A PhysioOpenSimModel external pointer.

Value

The input model handle (invisibly).

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelInitialize(model)
opensimModelIsInitialized(model)  # TRUE

## End(Not run)
```

opensimModelIsInitialized *Check Whether an OpenSim Model Handle Is Initialized*

Description

Check Whether an OpenSim Model Handle Is Initialized

Usage

```
opensimModelIsInitialized(model)
```

Arguments

model	A PhysioOpenSimModel external pointer.

Value

TRUE when model state has been initialized by OpenSim.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelIsInitialized(model)

## End(Not run)
```

opensimModelName *Get OpenSim Model Name*

Description

Get OpenSim Model Name

Usage

```
opensimModelName(model)
```

Arguments

model	A PhysioOpenSimModel external pointer.

Value

Character scalar model name.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimModelName(model)

## End(Not run)
```

---

opensimModelSummary *Summarize an OpenSim Model via Native C++ API*

Description

Loads an `.osim` model (or uses a loaded model handle) and returns structural summary information.

Usage

```
opensimModelSummary(path)
```

Arguments

`path`	Path to an OpenSim model file (`.osim`) or a `PhysioOpenSimModel` external pointer.

Value

A named list with model metadata (`model_name`, `n_bodies`, `n_joints`, `n_markers`, `n_muscles`, `n_coordinates`, `total_mass`, `initialized`).

Examples

```
## Not run:
info <- opensimModelSummary("gait2392.osim")
info$model_name
info$n_muscles

# Also works with a loaded model handle
model <- opensimLoadModel("gait2392.osim")
opensimModelSummary(model)

## End(Not run)
```

`opensimRunAnalyze` *Run OpenSim Analyze Tool*

Description

Run OpenSim Analyze Tool

Usage

```
opensimRunAnalyze(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

`setup_file`	Path to OpenSim tool setup XML.
`workdir`	Optional working directory.
`cli`	Optional command or absolute path (CLI mode only).
`extra_args`	Optional extra CLI args appended after setup file.
`timeout_sec`	Timeout in seconds for CLI execution (0 disables timeout).
`fail_on_error`	If `TRUE`, stop on non-zero exit status.
`execution`	Execution backend: `"auto"` (default), `"native"`, `"cli"`.

Value

See `opensimRunTool()`.

Examples

```
## Not run:
result <- opensimRunAnalyze("analyze_setup.xml")
result$status

## End(Not run)
```

opensimRunCMC *Run OpenSim Computed Muscle Control Tool*

Description

Run OpenSim Computed Muscle Control Tool

Usage

```
opensimRunCMC(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See opensimRunTool().

Examples

```
## Not run:
result <- opensimRunCMC("cmc_setup.xml")
result$status

## End(Not run)
```

opensimRunID *Run OpenSim Inverse Dynamics Tool*

Description

Run OpenSim Inverse Dynamics Tool

Usage

```
opensimRunID(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See opensimRunTool().

Examples

```
## Not run:
result <- opensimRunID("id_setup.xml")
result$status

## End(Not run)
```

`opensimRunIK` *Run OpenSim Inverse Kinematics Tool*

Description

Run OpenSim Inverse Kinematics Tool

Usage

```
opensimRunIK(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

`setup_file`	Path to OpenSim tool setup XML.
`workdir`	Optional working directory.
`cli`	Optional command or absolute path (CLI mode only).
`extra_args`	Optional extra CLI args appended after setup file.
`timeout_sec`	Timeout in seconds for CLI execution (0 disables timeout).
`fail_on_error`	If `TRUE`, stop on non-zero exit status.
`execution`	Execution backend: `"auto"` (default), `"native"`, `"cli"`.

Value

See `opensimRunTool()`.

Examples

```
## Not run:
result <- opensimRunIK("ik_setup.xml")
result$status

## End(Not run)
```

opensimRunRRA *Run OpenSim Residual Reduction Algorithm Tool*

Description

Run OpenSim Residual Reduction Algorithm Tool

Usage

```
opensimRunRRA(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See opensimRunTool().

Examples

```
## Not run:
result <- opensimRunRRA("rra_setup.xml")
result$status

## End(Not run)
```

opensimRunSO *Run OpenSim Static Optimization Tool*

Description

Run OpenSim Static Optimization Tool

Usage

```
opensimRunSO(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".

Value

See opensimRunTool().

Examples

```
## Not run:
result <- opensimRunSO("so_setup.xml")
result$status

## End(Not run)
```

opensimRunTool *Run an OpenSim Tool Setup XML*

Description

Executes a setup XML through native OpenSim bindings (when available) or through OpenSim CLI.

Usage

```
opensimRunTool(
  setup_file,
  workdir = NULL,
  cli = NULL,
  extra_args = character(),
  timeout_sec = 0L,
  fail_on_error = TRUE,
  execution = c("auto", "native", "cli"),
  tool_type = c("tool", "ik", "id", "so", "analyze", "cmc", "rra")
)
```

Arguments

setup_file	Path to OpenSim tool setup XML.
workdir	Optional working directory.
cli	Optional command or absolute path (CLI mode only).
extra_args	Optional extra CLI args appended after setup file.
timeout_sec	Timeout in seconds for CLI execution (0 disables timeout).
fail_on_error	If TRUE, stop on non-zero exit status.
execution	Execution backend: "auto" (default), "native", "cli".
tool_type	Tool flavor token: "tool" (generic), "ik", "id", or "so", "analyze", "cmc", "rra".

Value

A named list with command metadata and logs.

Examples

```
## Not run:
result <- opensimRunTool("ik_setup.xml")
result$status
result$stdout

# Force CLI backend with extra arguments
opensimRunTool("ik_setup.xml", execution = "cli",
```

```
                extra_args = "--visualize")

## End(Not run)
```

`opensimSaveModel` *Save OpenSim Model to File*

Description

Save OpenSim Model to File

Usage

```
opensimSaveModel(model, output_file)
```

Arguments

`model`	A `PhysioOpenSimModel` external pointer.
`output_file`	Output path for model XML.

Value

Normalized output path.

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimSetModelName(model, "modified")
opensimSaveModel(model, "gait2392_modified.osim")

## End(Not run)
```

`opensimSetModelName` *Set OpenSim Model Name*

Description

Set OpenSim Model Name

Usage

```
opensimSetModelName(model, name)
```

Arguments

`model`	A `PhysioOpenSimModel` external pointer.
`name`	New model name.

Value

The input `model` handle (invisibly).

Examples

```
## Not run:
model <- opensimLoadModel("gait2392.osim")
opensimSetModelName(model, "my_model")
opensimModelName(model)  # "my_model"

## End(Not run)
```

`opensimWriteAnalyzeSetupFromTemplate` *Write Analyze Setup XML from Template*

Description

Convenience wrapper around `opensimWriteToolSetupFromTemplate()` for common AnalyzeTool tags.

Usage

```
opensimWriteAnalyzeSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  coordinates_file = NULL,
  external_loads_file = NULL,
  time_range = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

`template_file`	Path to Analyze template XML.
`output_file`	Path to output Analyze setup XML.
`model_file`	Path to `.osim` model.
`coordinates_file`	Optional path to coordinates/motion file.

`external_loads_file`
: Optional path to external loads XML.

`time_range`
: Optional numeric length-2 vector (`c(start, end)`).

`results_directory`
: Optional output directory.

`extra_fields`
: Optional named list of additional XML tag replacements.

`strict`
: Passed to `opensimWriteToolSetupFromTemplate()`.

Value

See `opensimWriteToolSetupFromTemplate()`.

Examples

```
## Not run:
opensimWriteAnalyzeSetupFromTemplate(
  template_file = "analyze_template.xml",
  output_file = "analyze_setup.xml",
  model_file = "gait2392.osim",
  coordinates_file = "ik_output.mot"
)

## End(Not run)
```

---

opensimWriteCMCSetupFromTemplate

Write CMC Setup XML from Template

Description

Convenience wrapper around `opensimWriteToolSetupFromTemplate()` for common CMCTool tags.

Usage

```
opensimWriteCMCSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  desired_kinematics_file = NULL,
  external_loads_file = NULL,
  time_range = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

`template_file`	Path to CMC template XML.
`output_file`	Path to output CMC setup XML.
`model_file`	Path to `.osim` model.
`desired_kinematics_file`	Optional path to desired kinematics file.
`external_loads_file`	Optional path to external loads XML.
`time_range`	Optional numeric length-2 vector (`c(start, end)`).
`results_directory`	Optional output directory.
`extra_fields`	Optional named list of additional XML tag replacements.
`strict`	Passed to `opensimWriteToolSetupFromTemplate()`.

Value

See `opensimWriteToolSetupFromTemplate()`.

Examples

```
## Not run:
opensimWriteCMCSetupFromTemplate(
  template_file = "cmc_template.xml",
  output_file = "cmc_setup.xml",
  model_file = "gait2392.osim",
  desired_kinematics_file = "ik_output.mot"
)

## End(Not run)
```

`opensimWriteIDSetupFromTemplate` *Write ID Setup XML from Template*

Description

Convenience wrapper around `opensimWriteToolSetupFromTemplate()` for common Inverse Dynamics tags.

Usage

```
opensimWriteIDSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  coordinates_file,
  output_gen_force_file,
  external_loads_file = NULL,
  time_range = NULL,
  lowpass_cutoff_frequency_for_coordinates = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

`template_file`	Path to ID template XML.
`output_file`	Path to output ID setup XML.
`model_file`	Path to `.osim` model.
`coordinates_file`	Path to coordinates/motion file (typically `.mot`).
`output_gen_force_file`	Path to generalized force output file (`.sto`).
`external_loads_file`	Optional path to external loads XML.
`time_range`	Optional numeric length-2 vector (`c(start, end)`).
`lowpass_cutoff_frequency_for_coordinates`	Optional numeric cutoff.
`results_directory`	Optional output directory.
`extra_fields`	Optional named list of additional XML tag replacements.
`strict`	Passed to `opensimWriteToolSetupFromTemplate()`.

Value

See `opensimWriteToolSetupFromTemplate()`.

Examples

```
## Not run:
opensimWriteIDSetupFromTemplate(
  template_file = "id_template.xml",
  output_file = "id_setup.xml",
  model_file = "gait2392.osim",
  coordinates_file = "ik_output.mot",
  output_gen_force_file = "id_output.sto",
```

```
  time_range = c(0.5, 2.0)
)

## End(Not run)
```

`opensimWriteIKSetupFromTemplate` *Write IK Setup XML from Template*

Description

Convenience wrapper around `opensimWriteToolSetupFromTemplate()` for common Inverse Kinematics tags.

Usage

```
opensimWriteIKSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  marker_file,
  output_motion_file,
  time_range = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

`template_file`	Path to IK template XML.
`output_file`	Path to output IK setup XML.
`model_file`	Path to `.osim` model.
`marker_file`	Path to marker trajectory file (typically `.trc`).
`output_motion_file`	Path to IK output motion file (`.mot`).
`time_range`	Optional numeric length-2 vector (`c(start, end)`).
`results_directory`	Optional output directory.
`extra_fields`	Optional named list of additional XML tag replacements.
`strict`	Passed to `opensimWriteToolSetupFromTemplate()`.

Value

See `opensimWriteToolSetupFromTemplate()`.

Examples

```
## Not run:
opensimWriteIKSetupFromTemplate(
  template_file = "ik_template.xml",
  output_file = "ik_setup.xml",
  model_file = "gait2392.osim",
  marker_file = "walking.trc",
  output_motion_file = "ik_output.mot",
  time_range = c(0.5, 2.0)
)

## End(Not run)
```

---

opensimWriteRRASetupFromTemplate

Write RRA Setup XML from Template

Description

Convenience wrapper around `opensimWriteToolSetupFromTemplate()` for common RRATool tags.

Usage

```
opensimWriteRRASetupFromTemplate(
  template_file,
  output_file,
  model_file,
  desired_kinematics_file = NULL,
  external_loads_file = NULL,
  output_model_file = NULL,
  time_range = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

`template_file`	Path to RRA template XML.
`output_file`	Path to output RRA setup XML.
`model_file`	Path to `.osim` model.
`desired_kinematics_file`	Optional path to desired kinematics file.
`external_loads_file`	Optional path to external loads XML.

`output_model_file`
: Optional path to adjusted model output (`.osim`).

`time_range`
: Optional numeric length-2 vector (`c(start, end)`).

`results_directory`
: Optional output directory.

`extra_fields`
: Optional named list of additional XML tag replacements.

`strict`
: Passed to `opensimWriteToolSetupFromTemplate()`.

Value

See `opensimWriteToolSetupFromTemplate()`.

Examples

```
## Not run:
opensimWriteRRASetupFromTemplate(
  template_file = "rra_template.xml",
  output_file = "rra_setup.xml",
  model_file = "gait2392.osim",
  desired_kinematics_file = "ik_output.mot",
  output_model_file = "gait2392_rra.osim"
)

## End(Not run)
```

---

opensimWriteSOSetupFromTemplate

Write SO Setup XML from Template

Description

Convenience wrapper around `opensimWriteToolSetupFromTemplate()` for common Static Optimization tags.

Usage

```
opensimWriteSOSetupFromTemplate(
  template_file,
  output_file,
  model_file,
  coordinates_file,
  external_loads_file = NULL,
  time_range = NULL,
  results_directory = NULL,
  extra_fields = list(),
  strict = TRUE
)
```

Arguments

`template_file`	Path to SO template XML.
`output_file`	Path to output SO setup XML.
`model_file`	Path to `.osim` model.
`coordinates_file`	Path to coordinates/motion file (typically `.mot`).
`external_loads_file`	Optional path to external loads XML.
`time_range`	Optional numeric length-2 vector (`c(start, end)`).
`results_directory`	Optional output directory.
`extra_fields`	Optional named list of additional XML tag replacements.
`strict`	Passed to `opensimWriteToolSetupFromTemplate()`.

Value

See `opensimWriteToolSetupFromTemplate()`.

Examples

```
## Not run:
opensimWriteSOSetupFromTemplate(
  template_file = "so_template.xml",
  output_file = "so_setup.xml",
  model_file = "gait2392.osim",
  coordinates_file = "ik_output.mot",
  time_range = c(0.5, 2.0)
)

## End(Not run)
```

`opensimWriteToolSetupFromTemplate` *Write OpenSim Tool Setup XML from Template*

Description

Replaces tag values in an existing OpenSim setup XML template. This function is tool-agnostic and can be used for IK/ID/SO/RRA/CMC templates.

Usage

```
opensimWriteToolSetupFromTemplate(
  template_file,
  output_file,
  fields,
  strict = TRUE
)
```

Arguments

`template_file`	Path to template XML.
`output_file`	Path to output XML.
`fields`	Named list of replacement values keyed by XML tag name.
`strict`	If `TRUE`, error when a tag in `fields` is missing in template.

Value

A named list with `output_file`, `applied_tags`, and `missing_tags`.

Examples

```
# Create a minimal template in a temp file
tpl <- tempfile(fileext = ".xml")
writeLines(c(
  "<OpenSimDocument>",
  "  <model_file>Unassigned</model_file>",
  "  <time_range>0 1</time_range>",
  "</OpenSimDocument>"
), tpl)

out <- tempfile(fileext = ".xml")
result <- opensimWriteToolSetupFromTemplate(
  template_file = tpl,
  output_file = out,
  fields = list(model_file = "my_model.osim", time_range = "0.5 2.0")
)
result$applied_tags
readLines(result$output_file)
```

Index