

Package: nirs4alldatasets (via r-universe)

October 6, 2026

Title Acquire Curated NIRS Reference Datasets ('nirs4all-datasets')

Version 0.3.11

Description 'R' binding over the 'nirs4all-datasets' C 'ABI' ('n4ds_*'): resolve a dataset id from the distributable catalog index into a neutral descriptor plus version-pinned download contract, fetch the canonical 'Parquet' ('Dataverse' / 'Zenodo' / 'figshare') with SHA-256 verification into a local cache, and re-verify a cached directory offline. JSON crosses the stable C 'ABI'; analysis of the data is left to the host.

License AGPL-3

Copyright 2026 Gregory Beurier (CIRAD); see inst/COPYRIGHTS

Encoding UTF-8

NeedsCompilation yes

SystemRequirements Cargo (Rust toolchain >= 1.88), rustc (>= 1.88)

RoxygenNote 7.3.2

Config/pak/sysreqs libclang-dev

Repository <https://gbeurier.r-universe.dev>

Date/Publication 2026-10-06 14:50:59 UTC

RemoteUrl <https://github.com/GBeurier/nirs4all-datasets>

RemoteRef v0.3.11

RemoteSha 68c53f7909c0ca0f45cb5ea24679d7929c6c51d4

RemoteSubdir bindings/r/nirs4alldatasets

Contents

n4ds_abi_version	2
n4ds_fetch	2
n4ds_prepare_raw	3
n4ds_resolve	3
n4ds_retrieve_raw	4
n4ds_verify_cached	4

n4ds_abi_version	<i>The C ABI version string.</i>
------------------	----------------------------------

Description

The C ABI version string.

Usage

n4ds_abi_version()

Value

The C ABI version as a string.

n4ds_fetch	<i>Download + SHA-256-verify a resolved dataset into the cache (JSON status string).</i>
------------	--

Description

Download + SHA-256-verify a resolved dataset into the cache (JSON status string).

Usage

n4ds_fetch(resolved_json, opts_json = "")

Arguments

- resolved_jsonA resolved contract (from n4ds_resolve) as JSON.
- opts_jsonOptions as JSON: {cache_dir?, token?, instance?, timeout_secs?}; "" for the defaults.

Value

The fetch status as a JSON string (dir, files:[{name, relpath, path, status}]).

n4ds_prepare_raw	<i>Prepare Retrieved Raw Resources</i>
------------------	--

Description

Prepare already-retrieved raw resources with the Rust nirs4all reader stack.

Usage

```
n4ds_prepare_raw(request_json, opts_json = "")
```

Arguments

request_json	A retrieval request as JSON: {dataset_id, route}.
opts_json	Options as JSON: {cache_dir?}; "" for defaults.

Value

The preparation status as a JSON string: {dir, ok, route_id, resources}.

n4ds_resolve	<i>Resolve a dataset id against an index JSON into its download contract (JSON string).</i>
--------------	---

Description

Resolve a dataset id against an index JSON into its download contract (JSON string).

Usage

```
n4ds_resolve(index_json, dataset_id)
```

Arguments

index_json	The distributable catalog index as a JSON string.
dataset_id	The dataset id to resolve.

Value

The resolved download contract as a JSON string (id, tier, instance, doi, dataset_version, files, origins, retrieval, descriptor).

n4ds_retrieve_raw	<i>Retrieve raw origin resources into the cache (JSON status string).</i>
-------------------	---

Description

Retrieve raw origin resources into the cache (JSON status string).

Usage

```
n4ds_retrieve_raw(request_json, opts_json = "")
```

Arguments

request_json	A retrieval request as JSON: {dataset_id, route}.
opts_json	Options as JSON: {cache_dir?, timeout_secs?, max_total_bytes?}; "" for the defaults.

Value

The raw retrieval status as a JSON string (dir, ok, verified, route_id, resources).

n4ds_verify_cached	<i>Re-verify a cached dataset directory against the contract's SHA-256s (JSON report).</i>
--------------------	--

Description

Re-verify a cached dataset directory against the contract's SHA-256s (JSON report).

Usage

```
n4ds_verify_cached(resolved_json, dir)
```

Arguments

resolved_json	A resolved contract as JSON.
dir	The cached dataset directory.

Value

The verification report as a JSON string (dir, ok, files:[{name, relpath, status}]).

Index

n4ds_abi_version, [2](#)
n4ds_fetch, [2](#)
n4ds_prepare_raw, [3](#)
n4ds_resolve, [3](#)
n4ds_retrieve_raw, [4](#)
n4ds_verify_cached, [4](#)