Package 'natcpp'

May 9, 2026

Type Package

Title Fast C++ Primitives for the 'NeuroAnatomy Toolbox'

Version 0.2

Description Fast functions implemented in C++ via 'Rcpp' to support the 'NeuroAnatomy Toolbox' ('nat') ecosystem. These functions provide large speed-ups for basic manipulation of neuronal skeletons over pure R functions found in the 'nat' package. The expectation is that end users will not use this package directly, but instead the 'nat' package will automatically use routines from this package when it is available to enable large performance gains.

License GPL (>= 3)

URL https://github.com/natverse/natcpp, https://natverse.org/natcpp/

BugReports https://github.com/natverse/natcpp/issues

Imports Rcpp (>= 1.0.6)

Suggests spelling, testthat (>= 3.0.0)

LinkingTo Rcpp

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.2

NeedsCompilation yes

Author Gregory Jefferis [aut, cre] (ORCID: <https://orcid.org/0000-0002-0587-9355>)

Maintainer Gregory Jefferis <jefferis@gmail.com>

Repository CRAN

Date/Publication 2025-10-14 21:50:02 UTC

Contents

`c_coords21dindex` *Convert physical coordinates to 1d indices into image array*

Description

Convert physical coordinates to 1d indices into image array

Usage

```
c_coords21dindex(xyz, origin, voxdims, dims, clamp = FALSE)
```

Arguments

`xyz`	Nx3 matrix or data.frame of physical coordinates
`origin`	Numeric: 3d coordinates of the origin
`voxdims`	Numeric: 3 numbers describing the voxel dimensions
`dims`	Integer dimensions of the 3d image array
`clamp`	Logical: whether or not to clamp values within the pixel boundaries of the image.

Value

Nx3 integer matrix of pixel coordinates

c_EdgeListFromSegList *Turn a segment list into an edgelist suitable for constructing an ngraph*

Description

Turn a segment list into an edgelist suitable for constructing an ngraph

Usage

```
c_EdgeListFromSegList(L)
```

Arguments

L	a list containing integer vectors from as.seglist

Details

It is up to the caller to generate the seglist. Note that isolated points will be dropped since they have no edges.

Value

An integer matrix of N rows and 2 columns

Examples

```
## Not run:
library(nat)
# make a neuron with multiple subtrees
n=prune_vertices(Cell07PNs[[1]], 48L)
# Must use flatten=T if including all subtrees
sl=as.seglist(n, all = TRUE, flatten = TRUE)
c_EdgeListFromSegList(sl)

## End(Not run)
```

c_ijkpos *Convert physical coordinates to pixel coordinates*

Description

Convert physical coordinates to pixel coordinates

Usage

```
c_ijkpos(xyz, origin, voxdims, dims, clamp = FALSE)
```

Arguments

xyz	Nx3 matrix of physical coordinates
origin	Numeric: 3d coordinates of the origin
voxdims	Numeric: 3 numbers describing the voxel dimensions
dims	Integer dimensions of the 3d image array
clamp	Logical: whether or not to clamp values within the pixel boundaries of the image.

Value

Nx3 integer matrix of pixel coordinates

---

c_listlengths *A simple function to compute the lengths of the elements of an R list*

Description

A simple function to compute the lengths of the elements of an R list

Usage

```
c_listlengths(L)
```

Arguments

L	a list

Details

This is equivalent to the `base::lengths` however it it much faster for long lists (and somewhat slower for short ones).

Value

An integer vector containing the length of each element of `L`

c_ListofMatrixRows *Convert a matrix into list of row vectors*

Description

Convert a matrix into list of row vectors

Usage

```
c_ListofMatrixRows(object)
```

Arguments

object	An integer, numeric, character or logical matrix of N rows and M columns

Details

Typically this will be for 3D coordinates but there are no limits on row length.

Value

a list containing N vectors of length M corresponding to the rows of object.

Examples

```
## Not run:
library(nat)
xyz=xyzmatrix(Cell07PNs)
mat2list = function(m) {
um=unname(m)
lapply(1:nrow(um), function(i) um[i,])
}
bench::mark(rcpp=c_ListofMatrixRows(xyz), r=mat2list(xyz))

## End(Not run)
```

c_seglengths *Compute summed segment lengths or total cable*

Description

c_seglengths computes the summed segment length equivalent to nat::seglengths(sumsegment = T)

c_total_cable computes the summed total cable for a whole neuron. It's intended use is the nat::summary.neuron function.

Usage

```
c_seglengths(sl, x, y, z)

c_total_cable(sl, x, y, z)
```

Arguments

sl	A seglist with 1-indices into vectors x,y,z
x, y, z	Numeric vectors with 3D coordinate data (which could be columns from a data frame)

---

c_sub2ind *Find 1D index given n-dimensional indices*

Description

Find 1D index given n-dimensional indices

Usage

```
c_sub2ind(dims, indices)
```

Arguments

dims	Integer dimensions of the array (usually 3d)
indices	Nx3 integer matrix of pixel coordinates

Value

numeric vector of linear indices into the array

---

c_topntail *Find the first and last elements of all vectors in a list*

Description

c_topntail returns an 2xN matrix containing the start and end of each of the vectors in the input list. Length 0 vectors are ignored, while length 1 vectors are duplicated

For c_topntail_list, a list of the same length as L having the same elements when their length is <=2 or the first and last elements when length>2.

Usage

```
c_topntail(L)

c_topntail_list(L)
```

Arguments

L	a list containing integer vectors, typically a `seglist`

Value

For `c_topntail` an integer `matrix`. For `c_topntail_list` a `list`.

Index