

Package ‘gpuR’

May 23, 2024

Type Package

Title GPU Functions for R Objects

Description Provides GPU enabled functions for 'R' objects in a simple and approachable manner. New 'gpu*' and 'vcl*' classes have been provided to wrap typical 'R' objects (e.g. vector, matrix), in both host and device spaces, to mirror typical 'R' syntax without the need to know 'OpenCL'.

Version 2.0.6

Date 2024-05-22

Maintainer Ruoyong Xu <ruoyong.xu@mail.utoronto.ca>

VignetteBuilder knitr

License GPL (>= 2)

Encoding UTF-8

Depends R (>= 3.0.2), methods, utils

Imports Rcpp (>= 0.12.15), RViennaCL, BH, RcppEigen

LinkingTo Rcpp (>= 0.12.15), RcppEigen (>= 0.3.3.4.0), RViennaCL (>= 1.7.1.7), BH

NeedsCompilation yes

Suggests testthat, knitr

URL <https://github.com/eborgnine/gpuR>

BugReports <https://github.com/cdeterman/gpuR/issues/new>

SystemRequirements C++11 (supporting at least std=c++0x), OpenCL shared library (provided by an SDK such as AMD/NVIDIA) and OpenCL headers including the C++ header file (provided by Khronos if not by SDK)

OS_type unix

RoxygenNote 7.3.1

Author Charles Determan Jr [aut, cph],
Patrick Brown [aut, ctb] (<<https://orcid.org/0000-0003-2541-3744>>),
Ruoyong Xu [cre, ctb] (<<https://orcid.org/0000-0003-3474-8183>>),
The Khronos Group Inc [cph] (Copyright holder of some included header files)

Repository CRAN

Date/Publication 2024-05-23 16:00:02 UTC

R topics documented:

gpuR-package	4
Arith,gpuVector,gpuVector-method	4
as.gpuMatrix	6
as.gpuVector	7
as.vclVector	8
assert_has_double	8
block	9
cgpuMatrix-class	10
chol.vclMatrix	11
colnames	12
colSums,gpuMatrix-method	13
colSums,vclMatrix-method	14
Compare,vector,gpuVector-method	14
cov,vclMatrix,missing,missing,missing-method	15
crossprod,gpuMatrix,missing-method	16
currentContext	17
currentDevice	18
currentPlatform	18
custom_opencl	19
cvclMatrix-class	19
deepcopy	20
det,vclMatrix-method	21
detectCPUs	22
detectGPUs	22
detectPlatforms	23
deviceHasDouble	23
deviceType	24
dgpuMatrix-class	24
dgpuVector-class	25
diag,vclMatrix-method	25
dim,vclMatrix-method	26
distance	27
dvcMatrix-class	28
dvcVector-class	28
eigen,gpuMatrix-method	29
fgpuMatrix-class	30
fgpuVector-class	30
fvclMatrix-class	31
fvclVector-class	31
gpuInfo	32
gpuMatrix	33
gpuMatrix-class	34

gpuVector	35
gpuVector-class	36
has_cpu_skip	36
has_double_skip	37
has_gpu_skip	37
has_multiple_double_skip	37
has_multiple_gpu_skip	38
identity_matrix	38
igpuMatrix-class	39
igpuVector-class	39
inplace	40
ivclMatrix-class	41
ivclVector-class	42
length,gpuVector-method	42
listContexts	43
log,gpuVector-method	44
Math,gpuVector-method	45
norm,vclMatrix,character-method	46
nrow,vclMatrix-method	47
permute	47
platformInfo	48
pmax	49
pocl_check	49
print,gpuMatrix	50
qr,gpuMatrix	50
qr.R,gpuQR-method	51
setContext	52
setup_opencl	52
set_device_context	53
slice	54
solve,vclMatrix,vclMatrix-method	55
Summary,gpuVector-method	55
svd	56
synchronize	57
t,vclMatrix-method	58
tcrossprod,gpuVector,gpuVector-method	58
typeof,gpuMatrix-method	60
vclMatrix	61
vclMatrix-class	62
vclVector	63
vclVector-class	64
zgpuMatrix-class	65
zvclMatrix-class	65
[,gpuMatrix,missing,missing,missing-method	66
%o%,gpuVector,gpuVector-method	69
%*%,gpuVector,gpuVector-method	69

 gpuR-package

GPU functions for R Objects

Description

This package was developed to provide simple to use R functions that leverage the power of GPU's but also retain a format familiar to the R user. There are a hand full of other R packages that provide some GPU functionality but nearly all rely on a CUDA backend thereby restricting the user to NVIDIA GPU hardware. In the spirit of being as broadly applicable as possible, this GPU code herein relies upon OpenCL via the ViennaCL library.

OpenCL, in contrast to CUDA, is open source and can be used across different graphics cards (e.g. NVIDIA, AMD, Intel). This package removes the complex code needed for GPU computing and provides easier to use functions to apply on R objects.

```

Package:    gpuR
Type:      Package
Version:   1.0.0
Date:      2015-03-31
License:   GPL-3
Copyright: (c) 2015 Charles E. Determan Jr.
URL:       https://github.com/cdeterman/gpuR
LazyLoad: yes
  
```

Note

There are other packages that also provide wrappers for OpenCL programming including **OpenCL** by Simon Urbanek and **ROpenCL** at Open Analytics by Willem Ligtenberg. Both of these packages provide the R user an interface to directly call OpenCL functions. This package, however, hides these functions so the user does not require any knowledge of OpenCL to begin using their GPU. The idea behind this package is to provide a means to begin using existing algorithms without the need to write extensive amounts of C/C++/OpenCL code.

Author(s)

Charles Determan <cdetermanjr@gmail.com>

Maintainer: Charles Determan <cdetermanjr@gmail.com>

 Arith, gpuVector, gpuVector-method

Arith methods

Description

Methods for the base Arith methods [S4groupGeneric](#)

Usage

```
## S4 method for signature 'gpuVector,gpuVector'  
Arith(e1, e2)  
  
## S4 method for signature 'numeric,gpuVector'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuVector,numeric'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuVector,missing'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuVector,gpuMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'vclMatrix,vclMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'vclMatrix,matrix'  
Arith(e1, e2)  
  
## S4 method for signature 'matrix,vclMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'vclMatrix,numeric'  
Arith(e1, e2)  
  
## S4 method for signature 'numeric,vclMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'vclMatrix,missing'  
Arith(e1, e2)  
  
## S4 method for signature 'vclMatrix,vclVector'  
Arith(e1, e2)  
  
## S4 method for signature 'vclVector,vclVector'  
Arith(e1, e2)  
  
## S4 method for signature 'numeric,vclVector'  
Arith(e1, e2)  
  
## S4 method for signature 'vclVector,numeric'  
Arith(e1, e2)  
  
## S4 method for signature 'vclVector,missing'  
Arith(e1, e2)
```

```
## S4 method for signature 'vclVector,vclMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuMatrix,gpuMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuMatrix,matrix'  
Arith(e1, e2)  
  
## S4 method for signature 'matrix,gpuMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuMatrix,numeric'  
Arith(e1, e2)  
  
## S4 method for signature 'numeric,gpuMatrix'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuMatrix,missing'  
Arith(e1, e2)  
  
## S4 method for signature 'gpuMatrix,gpuVector'  
Arith(e1, e2)
```

Arguments

e1	A gpuR object
e2	A gpuR object

Value

A gpuR object

Author(s)

Charles Determan Jr.

as.gpuMatrix	<i>Convert object to a gpuMatrix</i>
--------------	--------------------------------------

Description

Construct a gpuMatrix of a class that inherits from gpuMatrix

Usage

```
as.gpuMatrix(object, type)
```

Arguments

object	An object that is or can be converted to a matrix object
type	A character string specifying the type of gpuMatrix. Default is NULL where type is inherited from the source data type.

Value

A gpuMatrix object

Author(s)

Charles Determan Jr.

as.gpuVector	<i>Convert object to a gpuVector</i>
--------------	--------------------------------------

Description

Construct a gpuVector of a class that inherits from gpuVector

Usage

```
as.gpuVector(object, type)

## S4 method for signature 'vector'
as.gpuVector(object, type = NULL)
```

Arguments

object	An object that is or can be converted to a vector object
type	A character string specifying the type of gpuVector. Default is NULL where type is inherited from the source data type.

Value

A gpuVector object

Author(s)

Charles Determan Jr.

as.vclVector *Convert object to a vclVector*

Description

Construct a vclVector of a class that inherits from vclVector
stuff

Usage

```
as.vclVector(object, type = NULL, ...)
```

```
## S4 method for signature 'vector'
```

```
as.vclVector(object, type = NULL)
```

```
## S4 method for signature 'vclMatrix'
```

```
as.vclVector(object, type = NULL, shared = FALSE)
```

Arguments

object	An object that is or can be converted to a vector object
type	A character string specifying the type of vclVector. Default is NULL where type is inherited from the source data type.
...	Additional arguments to as.vclVector methods
shared	Logical indicating if memory should be shared with x

Value

A vclVector object
A vclVector object.

Author(s)

Charles Determan Jr.

assert_has_double *Does device have 'double' support?*

Description

Function to query if device (identified by index) supports double precision

Usage

```

assert_has_double(
    device_idx,
    context_idx,
    severity = getOption("assertive.severity", "stop")
)

```

Arguments

device_idx	An integer indicating which device to query
context_idx	An integer indicating which context to query
severity	How severe should the consequences of the assertion be?

Value

Returns nothing but throws an error if device does not support double precision

Author(s)

Charles Determan Jr.

See Also

[deviceHasDouble](#)

block

Matrix Blocks

Description

This doesn't create a copy, it provides a child class that points to a contiguous submatrix of a [gpuMatrix](#) or [vclMatrix](#). Non-contiguous blocks are currently not supported.

Usage

```

block(object, rowStart, rowEnd, colStart, colEnd)

## S4 method for signature 'vclMatrix,integer,integer,integer,integer'
block(object, rowStart, rowEnd, colStart, colEnd)

## S4 method for signature 'gpuMatrix,integer,integer,integer,integer'
block(object, rowStart, rowEnd, colStart, colEnd)

```

Arguments

object	A <code>gpuMatrix</code> or <code>vclMatrix</code> object
rowStart	An integer indicating the first row of block
rowEnd	An integer indicating the last row of block
colStart	An integer indicating the first column of block
colEnd	An integer indicating the last column of block

Details

This function allows a user to create a `gpuR` matrix object that references a continuous subset of columns and rows of another `gpuR` matrix object without a copy.

NOTE - this means that altering values in a matrix block object will alter values in the source matrix.

Value

A block of the input `gpuMatrixBlock` or `vclMatrixBlock` object.

Author(s)

Charles Determan Jr.

`cgpuMatrix-class` *cgpuMatrix Class*

Description

An complex float type matrix in the `S4 gpuMatrix` representation.

Value

If the `gpuMatrix` object is of type 'complex float', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a complex float matrix.

Author(s)

Charles Determan Jr.

See Also

[gpuMatrix-class](#), [igpuMatrix-class](#), [dgpuMatrix-class](#)

chol.vclMatrix	<i>Cholesky Decomposition of a gpuR matrix</i>
----------------	--

Description

Compute the Choleski factorization of a real symmetric positive-definite square matrix.

Usage

```
## S3 method for class 'vclMatrix'  
chol(x, ...)  
  
## S3 method for class 'gpuMatrix'  
chol(x, ...)
```

Arguments

x	A symmetric, positive-definite gpuR matrix object.
...	arguments to be passed to or from methods

Value

Default - the upper triangular factor of the Choleski decomposition, i.e. the matrix R such that $R'R = x$.

Note

This an S3 generic of [chol](#). The default continues to point to the default base function.

No pivoting is used.

The argument `upper` is additionally accepted representing a boolean which will indicate if the upper or lower (FALSE) triangle should be solved.

Author(s)

Charles Determan Jr.

See Also

[chol](#)

colnames	<i>Row and Column Names</i>
----------	-----------------------------

Description

Retrieve or set the row or column names of a gpuR matrix object

Usage

```
colnames(x, do.NULL, prefix)

## Default S3 method:
colnames(x, do.NULL = TRUE, prefix = "col")

## S3 method for class 'gpuMatrix'
colnames(x, ...)

## S4 replacement method for signature 'gpuMatrix'
colnames(x) <- value

## S3 method for class 'vclMatrix'
colnames(x, ...)

## S4 replacement method for signature 'vclMatrix'
colnames(x) <- value
```

Arguments

x	A gpuR matrix object
do.NULL	logical. If FALSE names are NULL, names are created. (not currently used)
prefix	for create names. (not currently used)
...	Additional arguments
value	A character vector to assign as row/column names

Value

A character vector of column names of the gpuMatrix object.

`colSums,gpuMatrix-method`*Row and Column Sums and Means of gpuMatrix*

Description

Row and column sums and of gpuMatrix objects

Usage

```
## S4 method for signature 'gpuMatrix'  
colSums(x)
```

```
## S4 method for signature 'gpuMatrix'  
rowSums(x)
```

```
## S4 method for signature 'gpuMatrix'  
colMeans(x)
```

```
## S4 method for signature 'gpuMatrix'  
rowMeans(x)
```

Arguments

x A gpuMatrix object

Value

A gpuvector containing the sum of each column in the gpuMatrix.

A gpuvector containing the sum of each row in the gpuMatrix.

A gpuvector containing the mean of each column in the gpuMatrix.

A gpuvector containing the mean of each row in the gpuMatrix.

Author(s)

Charles Determan Jr.

colSums, vclMatrix-method

Row and Column Sums and Means of vclMatrix

Description

Row and column sums and of vclMatrix objects

Usage

```
## S4 method for signature 'vclMatrix'  
colSums(x)
```

```
## S4 method for signature 'vclMatrix'  
rowSums(x)
```

```
## S4 method for signature 'vclMatrix'  
colMeans(x)
```

```
## S4 method for signature 'vclMatrix'  
rowMeans(x)
```

Arguments

x A vclMatrix object

Value

A vclVector object

Author(s)

Charles Determan Jr.

Compare, vector, gpuVector-method

Compare vector and gpuVector elements

Description

Methods for comparison operators

Usage

```
## S4 method for signature 'vector, gpuVector'
Compare(e1, e2)
```

```
## S4 method for signature 'gpuVector, vector'
Compare(e1, e2)
```

Arguments

```
e1          A vector/gpuVector object
e2          A vector/gpuVector object
```

Value

A logical vector

Author(s)

Charles Determan Jr.

```
cov, vclMatrix, missing, missing, missing-method
      Covariance (gpuR)
```

Description

Compute covariance values

Usage

```
## S4 method for signature 'vclMatrix, missing, missing, missing'
cov(x, y = NULL, use = NULL, method = "pearson")
```

```
## S4 method for signature 'vclMatrix, vclMatrix, missing, missing'
cov(x, y = NULL, use = NULL, method = "pearson")
```

```
## S4 method for signature 'vclMatrix, missing, missing, character'
cov(x, y = NULL, use = NULL, method = "pearson")
```

```
## S4 method for signature 'vclMatrix, vclMatrix, missing, character'
cov(x, y = NULL, use = NULL, method = "pearson")
```

```
## S4 method for signature 'gpuMatrix, missing, missing, missing'
cov(x, y = NULL, use = NULL, method = "pearson")
```

```
## S4 method for signature 'gpuMatrix, gpuMatrix, missing, missing'
cov(x, y = NULL, use = NULL, method = "pearson")
```

```
## S4 method for signature 'gpuMatrix,missing,missing,character'
cov(x, y = NULL, use = NULL, method = "pearson")
```

```
## S4 method for signature 'gpuMatrix,gpuMatrix,missing,character'
cov(x, y = NULL, use = NULL, method = "pearson")
```

Arguments

x	A gpuR object
y	A gpuR object
use	Not used
method	Character string indicating with covariance to be computed.

Value

A gpuMatrix/vclMatrix containing the symmetric covariance values.

Author(s)

Charles Determan Jr.

```
crossprod, gpuMatrix, missing-method
      gpuMatrix Crossproduct
```

Description

Return the matrix cross-product of two conformable matrices using a GPU. This is equivalent to `t(x)` or `x` device and host is required.

Usage

```
## S4 method for signature 'gpuMatrix,missing'
crossprod(x, y)
```

```
## S4 method for signature 'gpuMatrix,gpuMatrix'
crossprod(x, y)
```

```
## S4 method for signature 'gpuMatrix,matrix'
crossprod(x, y)
```

```
## S4 method for signature 'matrix,gpuMatrix'
crossprod(x, y)
```

```
## S4 method for signature 'gpuMatrix,missing'
tcrossprod(x, y)
```



```
## S4 method for signature 'gpuMatrix,gpuMatrix'  
tcrossprod(x, y)  
  
## S4 method for signature 'matrix,gpuMatrix'  
tcrossprod(x, y)  
  
## S4 method for signature 'gpuMatrix,matrix'  
tcrossprod(x, y)
```

Arguments

x	A gpuMatrix
y	A gpuMatrix

Value

A gpuMatrix

Author(s)

Charles Determan Jr.

currentContext	<i>Current Context</i>
----------------	------------------------

Description

Get current context index

Usage

```
currentContext()
```

Value

An integer reflecting the context listed in [listContexts](#)

currentDevice	<i>Current Device Information</i>
---------------	-----------------------------------

Description

Check current device information

Usage

```
currentDevice()
```

Value

list containing

device	Character string of device name
device_index	Integer identifying device
device_type	Character string identifying device type (e.g. gpu)

currentPlatform	<i>Return Current Platform</i>
-----------------	--------------------------------

Description

Find out which platform is currently in use

Usage

```
currentPlatform()
```

Value

platform	Name of the current platform
platform_index	Index of current platform

See Also

[detectPlatforms](#)

custom_opengl	<i>Custom OpenCL Kernels</i>
---------------	------------------------------

Description

Compile a custom function using a user provided OpenCL kernel

Usage

```
custom_opengl(kernel, cl_args, type)
```

Arguments

kernel	path to OpenCL kernel file
cl_args	A data.frame that contains argument definitions. Provided by setup_opengl
type	The precision on which the kernel is compiled. Options include "int", "float", and "double"

Value

This function does not return a value directly, but it compiles the provided OpenCL kernel and performs necessary operations for using it.

cvclMatrix-class	<i>cvclMatrix Class</i>
------------------	-------------------------

Description

An complex float type matrix in the S4 vclMatrix representation.

Value

If the vclMatrix object is of type 'complex float', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a complex float type matrix

Author(s)

Charles Determan Jr.

See Also

[vclMatrix-class](#), [ivclMatrix-class](#), [fvclMatrix-class](#)

deepcopy	<i>Copy a "gpuR" object</i>
----------	-----------------------------

Description

This is needed to make a duplicate of a gpuR object

Usage

```
deepcopy(object, ...)  
  
## S4 method for signature 'gpuVector'  
deepcopy(object)  
  
## S4 method for signature 'vclMatrix'  
deepcopy(object, source = FALSE)  
  
## S4 method for signature 'vclVector'  
deepcopy(object)  
  
## S4 method for signature 'gpuMatrix'  
deepcopy(object)
```

Arguments

object	A gpuR object
...	Additional arguments
source	A boolean indicating if source matrix should be copied (only relevant for 'block' and 'slice' objects).

Details

This is needed to make a duplicate of a gpuR object (i.e. [gpuMatrix](#), [gpuVector](#), [vclMatrix](#), [vclVector](#) because the traditional syntax would only copy the pointer of the object.

Value

A gpuR object
A deep copy of the input `gpuVector` object.
A deep copy of the input `vclMatrix` object.
A deep copy of the input `vclVector` object.
A deep copy of the input `gpuMatrix` object.

Author(s)

Charles Determan Jr.

See Also

[block](#)

det, vclMatrix-method *Calculate Determinant of a Matrix on GPU*

Description

det calculates the determinant of a matrix.

Usage

```
## S4 method for signature 'vclMatrix'  
det(x)
```

```
## S4 method for signature 'gpuMatrix'  
det(x)
```

Arguments

x A gpuR matrix object

Value

The determinant of x

Note

This function uses an LU decomposition and the det function is simply a wrapper returning the determinant product

Author(s)

Charles Determan Jr.

detectCPUs	<i>Detect Available OpenCL enabled CPUs</i>
------------	---

Description

Find out how many CPUs available

Usage

```
detectCPUs(platform_idx = NULL)
```

Arguments

platform_idx An integer value indicating which platform to query. If NULL it will iterate over all platforms and sum results

Value

An integer representing the number of available CPUs

See Also

[detectPlatforms](#) [detectGPUs](#)

detectGPUs	<i>Detect Available GPUs</i>
------------	------------------------------

Description

Find out how many GPUs available

Usage

```
detectGPUs(platform_idx = NULL)
```

Arguments

platform_idx An integer value indicating which platform to query. If NULL it will iterate over all platforms and sum results

Value

An integer representing the number of available GPUs

See Also

[detectPlatforms](#)

detectPlatforms	<i>Detect Number of Platforms</i>
-----------------	-----------------------------------

Description

Find out how many OpenCL enabled platforms are available.

Usage

```
detectPlatforms()
```

Value

An integer value representing the number of platforms available.

See Also

[detectGPUs](#)

deviceHasDouble	<i>Check GPU double precision support</i>
-----------------	---

Description

This function checks the GPU device extensions for the variable `cl_khr_fp64` which means the device supports double precision.

Usage

```
deviceHasDouble(  
    gpu_idx = currentDevice()$device_index,  
    context_idx = currentContext()  
)
```

Arguments

<code>gpu_idx</code>	An integer value indicating which gpu to query.
<code>context_idx</code>	An integer value indicating which context to query.

Value

A boolean designating whether the device supports double precision

See Also

[gpuInfo](#)

deviceType	<i>Check device type</i>
------------	--------------------------

Description

Check what type a device is given platform and device indices

Usage

```
deviceType(device_idx = NULL, context_idx = currentContext())
```

Arguments

device_idx	An integer value indicating which device to query.
context_idx	An integer value indicating which context to query.

Value

A character string indicating the device type

dgpuMatrix-class	<i>dgpuMatrix Class</i>
------------------	-------------------------

Description

A double type matrix in the S4 `gpuMatrix` representation.

Value

If the `gpuMatrix` object is of type 'double', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a double type matrix

Author(s)

Charles Determan Jr.

See Also

[gpuMatrix-class](#), [igpuMatrix-class](#), [fgpuMatrix-class](#)

dgpuVector-class *dgpuVector Class*

Description

An double vector in the S4 gpuVector representation.

Value

If the gpuVector object is of type 'double', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a double typed vector

Author(s)

Charles Determan Jr.

See Also

[gpuVector-class](#)

diag,vclMatrix-method *gpuR Matrix Diagonals*

Description

Extract or replace the diagonal of a matrix

Usage

```
## S4 method for signature 'vclMatrix'  
diag(x)  
  
## S4 replacement method for signature 'vclMatrix,vclVector'  
diag(x) <- value  
  
## S4 method for signature 'gpuMatrix'  
diag(x)  
  
## S4 replacement method for signature 'gpuMatrix,gpuVector'  
diag(x) <- value
```

Arguments

x A gpuR matrix object
value A vector object (gpuR)

Value

A gpuRvector object of the matrix diagonal of x. The replacement form returns nothing as it replaces the diagonal of x.

Note

If an identity matrix is desired, please see [identity_matrix](#).

Author(s)

Charles Determan Jr.

See Also

[identity_matrix](#)

dim,vclMatrix-method *gpuMatrix/vclMatrix dim method*

Description

Retrieve dimension of object

Usage

```
## S4 method for signature 'vclMatrix'  
dim(x)  
  
## S4 method for signature 'gpuMatrix'  
dim(x)
```

Arguments

x A gpuMatrix/vclMatrix object

Value

A numeric vector of length 2: the number of rows and columns in the gpuR object 'x'.

Author(s)

Charles Determan Jr.

Description

This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix.

Usage

```
distance(x, y, method = "euclidean")

## S4 method for signature 'vclMatrix'
dist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)

## S4 method for signature 'vclMatrix,vclMatrix'
distance(x, y, method = "euclidean")

## S4 method for signature 'gpuMatrix'
dist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)

## S4 method for signature 'gpuMatrix,gpuMatrix'
distance(x, y, method = "euclidean")
```

Arguments

x	A gpuMatrix or vclMatrix object
y	A gpuMatrix or vclMatrix object
method	the distance measure to be used. This must be one of "euclidean" or "sqEuclidean".
diag	logical value indicating whether the diagonal of the distance matrix should be printed
upper	logical value indicating whether the upper triangle of the distance matrix
p	The power of the Minkowski distance (not currently used)

Value

a gpuMatrix/vclMatrix containing the pairwise distances between rows of 'x' and 'y', based on the specified method.

`dvclMatrix-class` *dvclMatrix Class*

Description

An integer type matrix in the S4 `vclMatrix` representation.

Value

If the `vclMatrix` object is of type 'double', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a double type matrix

Author(s)

Charles Determan Jr.

See Also

[vclMatrix-class](#), [ivclMatrix-class](#), [fvclMatrix-class](#)

`dvclVector-class` *dvclVector Class*

Description

An double vector in the S4 `vclVector` representation.

Value

If the `vclVector` object is of type 'double', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a double typed vector

Author(s)

Charles Determan Jr.

See Also

[vclVector-class](#)

eigen,gpuMatrix-method

gpuMatrix Eigen Decomposition

Description

Computes the eigenvalues and eigenvectors for gpuMatrix objects.

Usage

```
## S4 method for signature 'gpuMatrix'
eigen(x, symmetric, only.values = FALSE, EISPACK = FALSE)

## S4 method for signature 'vclMatrix'
eigen(x, symmetric, only.values = FALSE, EISPACK = FALSE)
```

Arguments

x	A gpuMatrix object
symmetric	logical indication if matrix is assumed to be symmetric. If not specified or FALSE, the matrix is inspected for symmetry
only.values	if TRUE, returns only eigenvalues (internals still currently calculate both regardless)
EISPACK	logical. Defunct and ignored

Details

This function currently implements the qr_method function from the ViennaCL library. As such, non-symmetric matrices are not supported given that OpenCL does not have a 'complex' data type. Neither the eigenvalues nor the eigenvectors are sorted as done in the base R eigen method.

Value

values	A gpuVector containing the unsorted eigenvalues of x.
vectors	A gpuMatrix containing the unsorted eigenvectors of x

Note

The sign's may be different on some of the eigenvector elements. As noted in the base eigen documentation:

Recall that the eigenvectors are only defined up to a constant: even when the length is specified they are still only defined up to a scalar of modulus one (the sign for real matrices).

Therefore, although the signs may be different, the results are functionally equivalent

fgpuMatrix-class *fgpuMatrix Class*

Description

A float type matrix in the S4 `gpuMatrix` representation.

Value

If the `gpuMatrix` object is of type 'float', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a float matrix.

Author(s)

Charles Determan Jr.

See Also

[gpuMatrix-class](#), [igpuMatrix-class](#), [dgpuMatrix-class](#)

fgpuVector-class *fgpuVector Class*

Description

An float vector in the S4 `gpuVector` representation.

Value

If the `gpuVector` object is of type 'float', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a float typed vector

Author(s)

Charles Determan Jr.

See Also

[gpuVector-class](#)

fvclMatrix-class *fvclMatrix Class*

Description

An integer type matrix in the S4 `vclMatrix` representation.

Value

If the `vclMatrix` object is of type 'float', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a float matrix.

Author(s)

Charles Determan Jr.

See Also

[vclMatrix-class](#), [ivclMatrix-class](#), [dvclMatrix-class](#)

fvclVector-class *fvclVector Class*

Description

An float vector in the S4 `vclVector` representation.

Value

If the `vclVector` object is of type 'float', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a float typed vector

Author(s)

Charles Determan Jr.

See Also

[vclVector-class](#)

gpuInfo *Device Information*

Description

Get basic information about selected device (e.g. GPU)

Usage

```
gpuInfo(device_idx = NULL, context_idx = currentContext())
```

```
cpuInfo(device_idx = NULL, context_idx = currentContext())
```

Arguments

device_idx An integer value indicating which device to query.
context_idx An integer value indicating which context to query.

Value

deviceName Device Name
deviceVendor Device Vendor
numberOfCores Number of Computing Units (which execute the work groups)
maxWorkGroupSize
 Maximum number of work items per group
maxWorkItemDim Number of dimensions
maxWorkItemSizes
 Maximum number of works items per dimension
deviceMemory Global amount of memory (bytes)
clockFreq Maximum configured clock frequency of the device in MHz
localMem Maximum amount of local memory for each work group (bytes)
maxAllocatableMem
 Maximum amount of memory in a single piece (bytes)
available Whether the device is available
deviceExtensions
 OpenCL device extensions available
double_support Logical value if double type supported

Author(s)

Charles Determan Jr.

See Also

[detectPlatforms](#) [detectGPUs](#) [detectCPUs](#) [cpuInfo](#)

gpuMatrix	<i>Construct a gpuMatrix</i>
-----------	------------------------------

Description

Construct a gpuMatrix of a class that inherits from gpuMatrix

Usage

```
gpuMatrix(data = NA, nrow = NA, ncol = NA, type = NULL, ...)  
  
## S4 method for signature 'matrix'  
gpuMatrix(data, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'missing'  
gpuMatrix(data, nrow = NA, ncol = NA, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'numeric'  
gpuMatrix(data, nrow, ncol, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'integer'  
gpuMatrix(data, nrow, ncol, type = NULL, ctx_id = NULL)
```

Arguments

data	An object that is or can be converted to a matrix object
nrow	An integer specifying the number of rows
ncol	An integer specifying the number of columns
type	A character string specifying the type of gpuMatrix. Default is NULL where type is inherited from the source data type.
...	Additional method to pass to gpuMatrix methods
ctx_id	An integer specifying the object's context

Value

A gpuMatrix object

Author(s)

Charles Determan Jr.

`gpuMatrix-class`*gpuMatrix Class*

Description

This is the 'mother' class for all `gpuMatrix` objects. It is essentially a wrapper for a basic R matrix (possibly to be improved). All other `gpuMatrix` classes inherit from this class but there are no current circumstances where this class is used directly.

There are multiple child classes that correspond to the particular data type contained. These include `igpuMatrix`, `fgpuMatrix`, and `dgpuMatrix` corresponding to integer, float, and double data types respectively.

Value

An object of class 'gpuMatrix' with the specified slots.

Slots

Common to all `gpuMatrix` objects in the package

`address`: Pointer to data matrix

`.context_index`: Integer index of OpenCL contexts

`.platform_index`: Integer index of OpenCL platforms

`.platform`: Name of OpenCL platform

`.device_index`: Integer index of active device

`.device`: Name of active device

Note

R does not contain a native float type. As such, the matrix data within a `fgpuMatrix-class` will be represented as double but downcast when any `gpuMatrix` methods are used.

May also remove the type slot

Author(s)

Charles Determan Jr.

See Also

[igpuMatrix-class](#), [fgpuMatrix-class](#), [dgpuMatrix-class](#)

gpuVector	<i>Construct a gpuVector</i>
-----------	------------------------------

Description

Construct a gpuVector of a class that inherits from gpuVector

Usage

```
gpuVector(data, length, type = NULL, ...)  
  
## S4 method for signature 'vector,missing'  
gpuVector(data, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'missingOrNULL,ANY'  
gpuVector(data, length, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'numeric,numericOrInt'  
gpuVector(data, length, type = NULL, ctx_id = NULL)
```

Arguments

data	An object that is or can be converted to a vector
length	A non-negative integer specifying the desired length.
type	A character string specifying the type of gpuVector. Default is NULL where type is inherited from the source data type.
...	Additional method to pass to gpuVector methods
ctx_id	An integer specifying the object's context

Value

A gpuVector object

Author(s)

Charles Determan Jr.

gpuVector-class *gpuVector Class*

Description

This is the 'mother' class for all gpuVector objects. All other gpuVector classes inherit from this class but there are no current circumstances where this class is used directly.

There are multiple child classes that correspond to the particular data type contained. These include igpuVector.

Value

An object of class 'gpuVector' with the specified slots.

Slots

Common to all vclMatrix objects in the package

address: Pointer to data matrix
 .context_index: Integer index of OpenCL contexts
 .platform_index: Integer index of OpenCL platforms
 .platform: Name of OpenCL platform
 .device_index: Integer index of active device
 .device: Name of active device

Author(s)

Charles Determan Jr.

See Also

[igpuVector-class](#)

has_cpu_skip *Skip test for CPUs*

Description

Function to skip test that tests if no valid CPU's are detected

Usage

```
has_cpu_skip()
```

Value

Returns nothing but gives a message if no CPU is available.

has_double_skip	<i>Skip test for GPU double precision</i>
-----------------	---

Description

Function to skip test that tests if the detected GPU doesn't support double precision

Usage

```
has_double_skip()
```

Value

Returns nothing but gives a message if GPU doesn't support double precision.

has_gpu_skip	<i>Skip test for GPUs</i>
--------------	---------------------------

Description

Function to skip test that tests if no valid GPU's are detected

Usage

```
has_gpu_skip()
```

Value

Returns nothing but gives a message if no GPUs available.

has_multiple_double_skip	<i>Skip test for multiple GPUs with double precision</i>
--------------------------	--

Description

Function to skip test that tests if their aren't multiple detected GPU with double precision

Usage

```
has_multiple_double_skip()
```

Value

Returns nothing but gives a message if there are less than 2 GPUs with double precision.

has_multiple_gpu_skip *Skip test in less than 2 GPUs*

Description

Function to skip test that tests if less than 2 valid GPU's are detected

Usage

```
has_multiple_gpu_skip()
```

Value

Returns nothing but gives a message if only one GPU is available.

identity_matrix *Identity Matrix on Device*

Description

Creates an identity matrix directly on the current device (e.g. GPU)

Usage

```
identity_matrix(x, type = NULL)
```

Arguments

x	A numeric value indicating the order of the identity matrix
type	A character string specifying the type of gpuMatrix. Default is derived from <code>getOption("gpuR.default.type")</code> .

Value

A `vclMatrix` object

Note

This function was only created for `vclMatrix` objects as the copy from CPU to `gpuMatrix` is trivial using the base `diag` function.

Author(s)

Charles Determan Jr.

igpuMatrix-class	<i>igpuMatrix Class</i>
------------------	-------------------------

Description

An integer type matrix in the S4 `gpuMatrix` representation.

Value

If the `gpuMatrix` object is of type 'integer', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a integer typed matrix

Author(s)

Charles Determan Jr.

See Also

[gpuMatrix-class](#), [igpuMatrix-class](#), [dgpuMatrix-class](#)

igpuVector-class	<i>igpuVector Class</i>
------------------	-------------------------

Description

An integer vector in the S4 `gpuVector` representation.

Value

If the `gpuVector` object is of type 'integer', returns TRUE, if not, returns an error message.

Slots

address: An integer vector object

Author(s)

Charles Determan Jr.

See Also

[gpuVector-class](#)

`inplace`*Inplace Function Wrapper*

Description

Applies the provided function in-place on the first object passed

Usage

```
inplace(f, x, y)

## S4 method for signature ``function`,vclMatrix,vclMatrix'
inplace(f, x, y)

## S4 method for signature ``function`,vclMatrix,missing'
inplace(f, x, y)

## S4 method for signature ``function`,numeric,vclMatrix'
inplace(f, x, y)

## S4 method for signature ``function`,vclMatrix,numeric'
inplace(f, x, y)

## S4 method for signature ``function`,gpuMatrix,gpuMatrix'
inplace(f, x, y)

## S4 method for signature ``function`,gpuMatrix,missing'
inplace(f, x, y)

## S4 method for signature ``function`,numeric,gpuMatrix'
inplace(f, x, y)

## S4 method for signature ``function`,gpuMatrix,numeric'
inplace(f, x, y)

## S4 method for signature ``function`,vclVector,vclVector'
inplace(f, x, y)

## S4 method for signature ``function`,vclVector,missing'
inplace(f, x, y)

## S4 method for signature ``function`,vclVector,numeric'
inplace(f, x, y)

## S4 method for signature ``function`,numeric,vclVector'
inplace(f, x, y)
```



```
## S4 method for signature ``function`,gpuVector,gpuVector'  
inplace(f, x, y)  
  
## S4 method for signature ``function`,gpuVector,missing'  
inplace(f, x, y)  
  
## S4 method for signature ``function`,gpuVector,numeric'  
inplace(f, x, y)  
  
## S4 method for signature ``function`,numeric,gpuVector'  
inplace(f, x, y)
```

Arguments

f	A function
x	A gpuR object
y	A gpuR object

Value

No return, result applied in-place

Author(s)

Charles Determan Jr.

ivclMatrix-class	<i>ivclMatrix Class</i>
------------------	-------------------------

Description

An integer type matrix in the S4 `vclMatrix` representation.

Value

If the `vclMatrix` object is of type 'integer', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a integer typed matrix

Author(s)

Charles Determan Jr.

See Also

[vclMatrix-class](#), [ivclMatrix-class](#), [dvclMatrix-class](#)

`ivclVector-class` *ivclVector Class*

Description

An integer vector in the S4 `vclVector` representation.

Value

If the `vclVector` object is of type 'integer', returns TRUE, if not, returns an error message.

Slots

`address`: An integer vector object

Author(s)

Charles Determan Jr.

See Also

[vclVector-class](#)

`length, gpuVector-method`
gpuMatrix/vclMatrix length method

Description

Retrieve number of elements in object

Usage

```
## S4 method for signature 'gpuVector'
length(x)
```

```
## S4 method for signature 'vclMatrix'
length(x)
```

```
## S4 method for signature 'vclVector'
length(x)
```

```
## S4 method for signature 'gpuMatrix'
length(x)
```

Arguments

x A gpuMatrix/vclMatrix object

Value

Length of the gpuVector object x.

The total number of elements in the vclMatrix object 'x'.

The length of the vclVector based on its data type.

A numeric value

Author(s)

Charles Determan Jr.

listContexts	<i>Available OpenCL Contexts</i>
--------------	----------------------------------

Description

Provide a data.frame of available OpenCL contexts and associated information.

Usage

```
listContexts()
```

Value

data.frame containing the following fields

context	Integer identifying context
platform	Character string listing OpenCL platform
platform_index	Integer identifying platform
device	Character string listing device name
device_index	Integer identifying device
device_type	Character string labeling device (e.g. gpu)

log, gpuVector-method *gpuR Logarithms and Exponentials*

Description

log computes logarithms, by default natural logarithms and log10 computes common (i.e. base 10) logarithms. The general form log(x, base) computes logarithms with base base.

exp computes the exponential function.

Usage

```
## S4 method for signature 'gpuVector'  
log(x, base = NULL)  
  
## S4 method for signature 'vclMatrix'  
log(x, base = NULL)  
  
## S4 method for signature 'vclVector'  
log(x, base = NULL)  
  
## S4 method for signature 'gpuMatrix'  
log(x, base = NULL)
```

Arguments

x	A gpuR object
base	A positive number (complex not currently supported by OpenCL): the base with respect to which logarithms are computed. Defaults to the natural log.

Value

A gpuVector object with the element-wise natural logarithm of the elements of the input gpuVector object x.

A vclMatrix object.

A vclVector object.

A gpuR object of the same class as x

Math, gpuVector-method *gpuR Math methods*

Description

Methods for the base Math methods [S4groupGeneric](#)

Usage

```
## S4 method for signature 'gpuVector'  
Math(x)
```

```
## S4 method for signature 'vclMatrix'  
Math(x)
```

```
## S4 method for signature 'vclVector'  
Math(x)
```

```
## S4 method for signature 'gpuMatrix'  
Math(x)
```

Arguments

x A gpuR object

Details

Currently implemented methods include:

- "sin", "cos", "tan", "asin", "acos", "atan", "sinh", "cosh", "tanh", "log10", "exp", "abs", "sqrt", "sign"

Value

A gpuVector object resulting from the element-wise mathematical operation performed on the input gpuVector object x.

A vclMatrix object.

A vclVector object.

A gpuMatrix object.

Author(s)

Charles Determan Jr.

norm, vclMatrix, character-method

Compute the Norm of a Matrix

Description

Computes a matrix norm of x . The norm can be the one $\| \cdot \|_1$ norm, the infinity $\| \cdot \|_\infty$ norm, the Frobenius $\| \cdot \|_F$ norm, the maximum modulus $\| \cdot \|_M$ among elements of a matrix, or the “spectral” or “2”-norm, as determined by the value of `type`.

Usage

```
## S4 method for signature 'vclMatrix,character'  
norm(x, type)
```

```
## S4 method for signature 'vclMatrix,missing'  
norm(x, type)
```

```
## S4 method for signature 'gpuMatrix,character'  
norm(x, type)
```

```
## S4 method for signature 'gpuMatrix,missing'  
norm(x, type)
```

```
## S4 method for signature 'ANY,missing'  
norm(x, type)
```

```
## S4 method for signature 'ANY,character'  
norm(x, type)
```

Arguments

<code>x</code>	A gpuR matrix object
<code>type</code>	character string, specifying the type of matrix norm to be computed.

Value

The matrix norm, a non-negative number

Author(s)

Charles Determan Jr.

See Also

[norm](#)

nrow,vclMatrix-method *The Number of Rows/Columns of a gpuR matrix*

Description

nrow and ncol return the number of rows or columns present in x respectively.

Usage

```
## S4 method for signature 'vclMatrix'  
nrow(x)  
  
## S4 method for signature 'vclMatrix'  
ncol(x)  
  
## S4 method for signature 'gpuMatrix'  
nrow(x)  
  
## S4 method for signature 'gpuMatrix'  
ncol(x)
```

Arguments

x A gpuMatrix/vclMatrix object

Value

An integer.

An integer of length 1

Author(s)

Charles Determan Jr.

permute *Permuting functions for gpuR objects*

Description

Generate a permutation of row or column indices

Usage

```
permute(x, MARGIN, order)
```

Arguments

x	A gpuR matrix object
MARGIN	dimension over which the ordering should be applied, 1 indicates rows, 2 indicates columns
order	An integer vector indicating order of rows to assign

Value

A gpuR object

Author(s)

Charles Determan Jr.

platformInfo	<i>OpenCL Platform Information</i>
--------------	------------------------------------

Description

Get basic information about the OpenCL platform

Usage

```
platformInfo(platform_idx = 1L)
```

Arguments

platform_idx	An integer value to specify which platform to check
--------------	---

Value

platformName	Platform Name
platformVendor	Platform Vendor
platformVersion	Platform OpenCL Version
platformExtensions	Available platform extensions

Author(s)

Charles Determan Jr.

pmax *Parallel Maxima and Minima*

Description

pmax and pmin take one or more vectors as arguments and return a single vector giving the 'parallel' maxima (or minima) of the argument vectors

Usage

```
pmax(...)  
pmin(...)  
  
## S3 method for class 'vclVector'  
pmin(..., na.rm = FALSE)
```

Arguments

... a vclVector object
na.rm a logical indicating whether missing values should be removed.

Value

A vclMatrix object.

See Also

[pmax](#) [pmin](#)

pocl_check *POCL Version Check*

Description

Versions of POCL up to 0.15-pre have a bug which results in values being returned when NA values should be (e.g. fractional powers of negative values)

Usage

```
pocl_check()
```

Value

Returns nothing but gives a message if the POCL version is too old.

print.gpuMatrix *S3 print for gpuMatrix objects*

Description

prints a gpuMatrix object that is truncated to fit the screen

Usage

```
## S3 method for class 'gpuMatrix'  
print(x, ..., n = NULL, width = NULL)
```

Arguments

x	A gpuMatrix object
...	Additional arguments to print
n	Number of rows to display
width	Number of columns to display

Value

This function prints a truncated summary of a gpuMatrix object 'x' to fit the screen

qr.gpuMatrix *The QR Decomposition of a gpuR matrix*

Description

qr computes the QR decomposition of a gpuR matrix

Usage

```
## S3 method for class 'gpuMatrix'  
qr(x, ..., inplace = FALSE)  
  
## S3 method for class 'vclMatrix'  
qr(x, ..., inplace = FALSE)
```

Arguments

x	A gpuR matrix
...	further arguments passed to or from other methods
inplace	Logical indicating if operations performed inplace

Value

A list containing the QR decomposition of the matrix of class `gpuQR`. The returned value is a list with the following components:

- `qr` a matrix with the same dimensions as `x`.
- `betas` vector of numeric values containing additional information of `qr` for extracting Q and R matrices.

Note

This is an S3 generic of `qr`. The default continues to point to the default base function.

Furthermore, the list returned does not contain the exact same elements as `qr`. The matrix storage format applied herein doesn't match the base compact form. The method also doesn't return `qraux`, `rank`, or `pivot` but instead returns `betas`.

Author(s)

Charles Determan Jr.

See Also

[qr](#)

<code>qr.R, gpuQR-method</code>	<i>Reconstruct the Q or R Matrices from a gpuQR Object</i>
---------------------------------	--

Description

Returns the components of the QR decomposition.

Usage

```
## S4 method for signature 'gpuQR'
qr.R(qr, complete = FALSE)
```

```
## S4 method for signature 'gpuQR'
qr.Q(qr, complete = FALSE)
```

Arguments

<code>qr</code>	gpuQR object
<code>complete</code>	not currently used

Value

`qr.Q` returns all of Q, `qr.R` returns all of R

Author(s)

Charles Determan Jr.

See Also

[qr.R](#), [qr.Q](#)

setContext	<i>Set Context</i>
------------	--------------------

Description

Change the current context used by default

Usage

```
setContext(id = 1L)
```

Arguments

`id` Integer identifying which context to set

Value

It does not return anything. This function is designed to set the current context to the specified context ID.

See Also

[listContexts](#)

setup_opengl	<i>Setup OpenCL Arguments</i>
--------------	-------------------------------

Description

Generates a data.frame of argument definitions for use in [custom_opengl](#)

Usage

```
setup_opengl(objects, intents, queues, kernel_maps = NULL)
```

Arguments

objects	character vector of gpuR objects to be passed
intents	character vector specifying 'intent' of gpuR objects. options include "IN", "OUT", "INOUT"
queues	list of character vectors reflecting equal length to "objects" where each element reflects a kernel function defined in an OpenCL kernel file.
kernel_maps	The corresponding arguments names in the provided OpenCL kernel corresponds to the gpuR objects passed and contains a character vector of which kernels the object will be enqueued.

Value

A data.frame with columns:

object: The name of the gpuR object.

intents: The intent of the object, specified as 'IN', 'OUT', or 'INOUT'.

queues: A character vector reflecting equal length to objects, where each element reflects a kernel function defined in an OpenCL kernel file.

map: The corresponding arguments names in the provided OpenCL kernel corresponding to the gpuR objects passed.

set_device_context	<i>Set Context for Specific Device Type</i>
--------------------	---

Description

This function find the first context that contains a device of the specified type.

Usage

```
set_device_context(type)
```

Arguments

type	A character vector specifying device type
------	---

Value

An integer indicating previous context index

slice

Vector Slices

Description

This doesn't create a copy, it provides a child class that points to a contiguous subvector of a [gpuVector](#) or [vclVector](#). Non-contiguous slices are currently not supported.

Usage

```
slice(object, start, end)

## S4 method for signature 'gpuVector,integer,integer'
slice(object, start, end)

## S4 method for signature 'vclVector,integer,integer'
slice(object, start, end)
```

Arguments

object	A gpuVector or vclVector object
start	An integer indicating the start of slice
end	An integer indicating the end of slice

Details

This function allows a user to create a [gpuR](#) vector object that references a continuous subset of columns and rows of another [gpuR](#) vector object without a copy.

NOTE - this means that altering values in a vector slice object will alter values in the source vector.

Value

A [gpuVectorSlice](#) or [vclVectorSlice](#) object.

Author(s)

Charles Determan Jr.

 solve, vclMatrix, vclMatrix-method

Solve a System of Equations for gpuR objects

Description

This function solves the equation $a \%*\% x = b$ for x , where b can be either a vector or a matrix.

Usage

```
## S4 method for signature 'vclMatrix,vclMatrix'
solve(a, b, ...)
```

```
## S4 method for signature 'vclMatrix,missing'
solve(a, b, ...)
```

```
## S4 method for signature 'gpuMatrix,gpuMatrix'
solve(a, b, ...)
```

```
## S4 method for signature 'gpuMatrix,missing'
solve(a, b, ...)
```

Arguments

a	A gpuR object
b	A gpuR object
...	further arguments passed to or from other methods

Value

A gpuR object

Author(s)

Charles Determan Jr.

 Summary, gpuVector-method

gpuR Summary methods

Description

Methods for the base Summary methods [S4groupGeneric](#)

Usage

```
## S4 method for signature 'gpuVector'
Summary(x, ..., na.rm = FALSE)

## S4 method for signature 'vclMatrix'
Summary(x, ..., na.rm = FALSE)

## S4 method for signature 'vclVector'
Summary(x, ..., na.rm = FALSE)

## S4 method for signature 'gpuMatrix'
Summary(x, ..., na.rm = FALSE)
```

Arguments

x	A gpuR object
...	Additional arguments passed to method (not currently used)
na.rm	a logical indicating whether missing values should be removed (not currently used)

Value

For min or max, a length-one vector

svd	<i>Singular Value Decomposition of a gpuR matrix</i>
-----	--

Description

Compute the singular-value decomposition of a gpuR matrix

Usage

```
svd(x, nu, nv, LINPACK)

## S3 method for class 'vclMatrix'
svd(x, nu, nv, LINPACK)

## S3 method for class 'gpuMatrix'
svd(x, nu, nv, LINPACK)
```

Arguments

x	A gpuR matrix
nu	ignored
nv	ignored
LINPACK	ignored

Value

The SVD decomposition of the matrix. The returned value is a list with the following components:

- `d` a vector containing the singular values of `x`
- `u` a matrix whose columns contain the left singular vectors of `x`.
- `v` a matrix whose columns contain the right singular vectors of `x`.

Note

This is an S3 generic of `svd`. The default continues to point to the default base function.

Author(s)

Charles Determan Jr.

See Also

[svd](#)

synchronize

Synchronize Device Execution

Description

This pauses execution until the processing is complete on the device (CPU/GPU/etc.). This is important especially for benchmarking applications.

Usage

```
synchronize()
```

Value

This does not return anything.

Author(s)

Charles Determan Jr.

Examples

```
## Not run:
  mat <- vclMatrix(rnorm(500^2), ncol = 500, nrow = 500)
  system.time({mat %*% mat})
  system.time({mat %*% mat; synchronize()})

## End(Not run)
```

t, vclMatrix-method gpuR *matrix transpose*

Description

Given a gpuR matrix *x*, *t* returns the transpose of *x*

Usage

```
## S4 method for signature 'vclMatrix'
t(x)

## S4 method for signature 'gpuMatrix'
t(x)
```

Arguments

x A gpuR matrix

Value

The transpose of the input vclMatrix object.
A gpuR matrix

Author(s)

Charles Determan Jr.

tcrossprod, gpuVector, gpuVector-method
vclMatrix Crossproduct

Description

Return the matrix cross-product of two conformable matrices using a GPU. This is equivalent to *t(x)* or *x* device and host is required.

Usage

```
## S4 method for signature 'gpuVector, gpuVector'
tcrossprod(x, y)

## S4 method for signature 'gpuVector, missing'
tcrossprod(x, y)
```

```
## S4 method for signature 'vclMatrix,missing'  
crossprod(x, y)  
  
## S4 method for signature 'vclMatrix,vclMatrix'  
crossprod(x, y)  
  
## S4 method for signature 'vclMatrix,matrix'  
crossprod(x, y)  
  
## S4 method for signature 'matrix,vclMatrix'  
crossprod(x, y)  
  
## S4 method for signature 'vclMatrix,vclVector'  
crossprod(x, y)  
  
## S4 method for signature 'vclVector,vclMatrix'  
crossprod(x, y)  
  
## S4 method for signature 'vclMatrix,missing'  
tcrossprod(x, y)  
  
## S4 method for signature 'vclMatrix,vclMatrix'  
tcrossprod(x, y)  
  
## S4 method for signature 'matrix,vclMatrix'  
tcrossprod(x, y)  
  
## S4 method for signature 'vclMatrix,matrix'  
tcrossprod(x, y)  
  
## S4 method for signature 'vclMatrix,vclVector'  
tcrossprod(x, y)  
  
## S4 method for signature 'vclVector,vclMatrix'  
tcrossprod(x, y)  
  
## S4 method for signature 'vclVector,vclVector'  
tcrossprod(x, y)  
  
## S4 method for signature 'vclVector,missing'  
tcrossprod(x, y)
```

Arguments

x	A vclMatrix object.
y	A vclMatrix object.

Value

A vclMatrix object of the transpose of the outer product of the two objects x and y.

Author(s)

Charles Determan Jr.

typeof,gpuMatrix-method

Get gpuR object type

Description

typeof determines the type (i.e. storage mode) of a gpuR object

Usage

```
## S4 method for signature 'gpuMatrix'  
typeof(x)
```

```
## S4 method for signature 'gpuVector'  
typeof(x)
```

```
## S4 method for signature 'vclMatrix'  
typeof(x)
```

```
## S4 method for signature 'vclVector'  
typeof(x)
```

Arguments

x A gpuR object

Value

A character string indicating the type of the gpuR object.

Author(s)

Charles Determan Jr.

`vclMatrix`*Construct a vclMatrix*

Description

Construct a `vclMatrix` of a class that inherits from `vclMatrix`. This class points to memory directly on the GPU to avoid the cost of data transfer between host and device.

Usage

```
vclMatrix(data = NA, nrow = NA, ncol = NA, type = NULL, ...)  
  
## S4 method for signature 'matrix'  
vclMatrix(data, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'missing'  
vclMatrix(data, nrow = NA, ncol = NA, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'numeric'  
vclMatrix(data, nrow, ncol, type = NULL, ctx_id = NULL)  
  
## S4 method for signature 'integer'  
vclMatrix(data, nrow, ncol, type = NULL, ctx_id = NULL)
```

Arguments

<code>data</code>	An object that is or can be converted to a matrix object
<code>nrow</code>	An integer specifying the number of rows
<code>ncol</code>	An integer specifying the number of columns
<code>type</code>	A character string specifying the type of <code>vclMatrix</code> . Default is <code>NULL</code> where type is inherited from the source data type.
<code>...</code>	Additional method to pass to <code>vclMatrix</code> methods
<code>ctx_id</code>	An integer specifying the object's context

Value

A `vclMatrix` object

Author(s)

Charles Determan Jr.

`vclMatrix-class`*vclMatrix Class*

Description

This is the 'mother' class for all `vclMatrix` objects. These objects are pointers to `viennacl` matrices directly on the GPU. This will avoid the overhead of passing data back and forth between the host and device.

As such, any changes made to normal R 'copies' (e.g. `A <- B`) will be propagated to the parent object.

There are multiple child classes that correspond to the particular data type contained. These include `ivclMatrix`, `fvclMatrix`, and `dvclMatrix` corresponding to integer, float, and double data types respectively.

Value

An object of class 'vclMatrix' with the specified slots.

Slots

Common to all `vclMatrix` objects in the package

`address`: Pointer to data matrix

`.context_index`: Integer index of OpenCL contexts

`.platform_index`: Integer index of OpenCL platforms

`.platform`: Name of OpenCL platform

`.device_index`: Integer index of active device

`.device`: Name of active device

Note

R does not contain a native float type. As such, the matrix data within a `fvclMatrix-class` will be represented as double but downcast when any `vclMatrix` methods are used.

May also remove the type slot

Author(s)

Charles Determan Jr.

See Also

[ivclMatrix-class](#), [fvclMatrix-class](#), [dvclMatrix-class](#)

vclVector	<i>Construct a vclVector</i>
-----------	------------------------------

Description

Construct a vclVector of a class that inherits from vclVector. This class points to memory directly on the GPU to avoid the cost of data transfer between host and device.

Usage

```
vclVector(data, length, type = NULL, ...)

## S4 method for signature 'vector,missing'
vclVector(data, length, type = NULL, ctx_id = NULL)

## S4 method for signature 'missing,ANY'
vclVector(data, length, type = NULL, ctx_id = NULL)

## S4 method for signature 'numeric,numericOrInt'
vclVector(data, length, type = NULL, ctx_id = NULL)

## S4 method for signature 'vclMatrix,missing'
vclVector(
  data,
  length = NULL,
  type = NULL,
  ctx_id = NULL,
  col = NULL,
  row = NULL
)
```

Arguments

<code>data</code>	An object that is or can be converted to a vector
<code>length</code>	A non-negative integer specifying the desired length.
<code>type</code>	A character string specifying the type of vclVector. Default is NULL where type is inherited from the source data type.
<code>...</code>	Additional method to pass to vclVector methods
<code>ctx_id</code>	An integer specifying the object's context
<code>col</code>	index of column to extract from vclMatrix
<code>row</code>	index of row to extract from vclMatrix

Value

A vclVector object

Author(s)

Charles Determan Jr.

vclVector-class *vclVector Class*

Description

This is the 'mother' class for all vclVector objects. All other vclVector classes inherit from this class but there are no current circumstances where this class is used directly.

There are multiple child classes that correspond to the particular data type contained. These include ivclVector.

Value

An object of class 'vclVector' with the specified slots.

Slots

Common to all vclMatrix objects in the package

address: Pointer to data matrix

.context_index: Integer index of OpenCL contexts

.platform_index: Integer index of OpenCL platforms

.platform: Name of OpenCL platform

.device_index: Integer index of active device

.device: Name of active device

Author(s)

Charles Determan Jr.

See Also

[ivclVector-class](#)

zgpuMatrix-class	<i>zgpuMatrix Class</i>
------------------	-------------------------

Description

An complex double type matrix in the S4 `gpuMatrix` representation.

Value

If the `gpuMatrix` object is of type 'complex double', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a complex double matrix.

Author(s)

Charles Determan Jr.

See Also

[gpuMatrix-class](#), [igpuMatrix-class](#), [dgpuMatrix-class](#)

zvclMatrix-class	<i>zvclMatrix Class</i>
------------------	-------------------------

Description

An complex double type matrix in the S4 `vclMatrix` representation.

Value

If the `vclMatrix` object is of type 'complex double', returns TRUE, if not, returns an error message.

Slots

address: Pointer to a complex double type matrix

Author(s)

Charles Determan Jr.

See Also

[vclMatrix-class](#), [ivclMatrix-class](#), [fvclMatrix-class](#)

```
[,gpuMatrix,missing,missing,missing-method
  Extract gpuR object elements
```

Description

Operators to extract or replace elements

Usage

```
## S4 method for signature 'gpuMatrix,missing,missing,missing'
x[i, j, drop]

## S4 method for signature 'gpuMatrix,missing,numeric,missing'
x[i, j, drop]

## S4 method for signature 'gpuMatrix,numeric,missing,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'gpuMatrix,numeric,numeric,missing'
x[i, j, drop]

## S4 replacement method for signature 'gpuMatrix,numeric,missing,numeric'
x[i, j, ...] <- value

## S4 replacement method for signature 'igpuMatrix,numeric,missing,integer'
x[i, j] <- value

## S4 replacement method for signature 'gpuMatrix,missing,numeric,numeric'
x[i, j] <- value

## S4 replacement method for signature 'igpuMatrix,missing,numeric,integer'
x[i, j] <- value

## S4 replacement method for signature 'gpuMatrix,numeric,numeric,numeric'
x[i, j] <- value

## S4 replacement method for signature 'igpuMatrix,numeric,numeric,integer'
x[i, j] <- value

## S4 method for signature 'gpuVector,missing,missing,missing'
x[i, j, drop]

## S4 method for signature 'gpuVector,numeric,missing,missing'
x[i, j, drop]

## S4 replacement method for signature 'gpuVector,numeric,missing,numeric'
```

```
x[i, j] <- value

## S4 replacement method for signature 'gpuVector,numeric,missing,integer'
x[i, j] <- value

## S4 method for signature 'vclMatrix,missing,missing,missing'
x[i, j, drop]

## S4 method for signature 'vclMatrix,missing,numeric,missing'
x[i, j, drop]

## S4 method for signature 'vclMatrix,numeric,missing,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'vclMatrix,numeric,numeric,missing'
x[i, j, drop]

## S4 replacement method for signature 'vclMatrix,missing,numeric,numeric'
x[i, j] <- value

## S4 replacement method for signature 'ivclMatrix,missing,numeric,integer'
x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,numeric,missing,numeric'
x[i, j, ...] <- value

## S4 replacement method for signature 'ivclMatrix,numeric,missing,integer'
x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,numeric,numeric,numeric'
x[i, j] <- value

## S4 replacement method for signature 'ivclMatrix,numeric,numeric,integer'
x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,missing,missing,matrix'
x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,missing,missing,vclMatrix'
x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,missing,numeric,vclMatrix'
x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,missing,missing,numeric'
x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,missing,missing,vclVector'
```

```

x[i, j] <- value

## S4 replacement method for signature 'vclMatrix,missing,numeric,vclVector'
x[i, j] <- value

## S4 method for signature 'vclVector,missing,missing,missing'
x[i, j, drop]

## S4 method for signature 'vclVector,numeric,missing,missing'
x[i, j, drop]

## S4 replacement method for signature 'vclVector,numeric,missing,numeric'
x[i, j] <- value

## S4 replacement method for signature 'ivclVector,numeric,missing,integer'
x[i, j] <- value

## S4 replacement method for signature 'vclVector,logical,missing,numeric'
x[i, j] <- value

## S4 replacement method for signature 'vclVector,missing,missing,numeric'
x[i, j] <- value

## S4 replacement method for signature 'vclVector,missing,missing,vclVector'
x[i, j] <- value

## S4 replacement method for signature 'vclVector,numeric,missing,vclVector'
x[i, j] <- value

## S4 replacement method for signature 'vclVector,missing,missing,vclMatrix'
x[i, j] <- value

## S4 replacement method for signature 'vclVector,numeric,missing,vclMatrix'
x[i, j] <- value

```

Arguments

x	A gpuR object
i	indices specifying rows
j	indices specifying columns
drop	missing
...	Additional arguments
value	data of similar type to be added to gpuMatrix object

Value

A gpuR object of the same type as the input x, containing the elements specified by the indices i and j.

Author(s)

Charles Determan Jr.

%%,gpuVector,gpuVector-method
Outer Product

Description

The outer product of two gpuR vector objects

Usage

```
## S4 method for signature 'gpuVector,gpuVector'  
X %% Y  
  
## S4 method for signature 'vclVector,vclVector'  
X %% Y
```

Arguments

X	A gpuR object
Y	A gpuR object

Value

The outer product of the two gpuVector objects X and Y.

Author(s)

Charles Determan Jr.

%%,gpuVector,gpuVector-method
Methods for gpu/vcl Vector

Description

Multiply two gpuR objects, if they are conformable. If both are vectors of the same length, it will return the inner product (as a matrix).

Usage

```
## S4 method for signature 'gpuVector,gpuVector'  
x %*% y  
  
## S4 method for signature 'gpuVector,gpuMatrix'  
x %*% y  
  
## S4 method for signature 'vclMatrix,vclMatrix'  
x %*% y  
  
## S4 method for signature 'vclMatrix,vclVector'  
x %*% y  
  
## S4 method for signature 'vclMatrix,matrix'  
x %*% y  
  
## S4 method for signature 'matrix,vclMatrix'  
x %*% y  
  
## S4 method for signature 'vclVector,vclVector'  
x %*% y  
  
## S4 method for signature 'vclVector,vclMatrix'  
x %*% y  
  
## S4 method for signature 'gpuMatrix,gpuMatrix'  
x %*% y  
  
## S4 method for signature 'gpuMatrix,gpuVector'  
x %*% y  
  
## S4 method for signature 'gpuMatrix,matrix'  
x %*% y  
  
## S4 method for signature 'matrix,gpuMatrix'  
x %*% y
```

Arguments

x	A gpuR object
y	A gpuR object

Value

The inner product of the two `gpuVector` objects `x` and `y`.

The result of matrix-vector multiplication between the `gpuMatrix` `y` and the `gpuVector` `x`.

The result of multiplying the `vclMatrix` `'x'` and the `vclVector` `'y'`.

A `vclMatrix` object, the result of multiplying the `vclMatrix` 'x' and the regular R matrix 'y'.

A `vclMatrix` object, the result of multiplying the regular R matrix x and the `vclMatrix` y.

A `gpuMatrix` object which is the result of multiplying the two `gpuMatrix` objects 'x' and 'y'.

Author(s)

Charles Determan Jr.

Index

[,gpuMatrix,missing,missing,missing-method, 66 ([,gpuMatrix,missing,missing,missing-method), 66

[,gpuMatrix,missing,numeric,missing-method 66 [<-,gpuMatrix,numeric,numeric,numeric-method ([,gpuMatrix,missing,missing,missing-method), 66

[,gpuMatrix,numeric,missing,missing-method 66 [<-,gpuVector,numeric,missing,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,gpuMatrix,numeric,numeric,missing-method 66 [<-,gpuVector,numeric,missing,numeric-method ([,gpuMatrix,missing,missing,missing-method), 66

[,gpuVector,missing,missing,missing-method 66 [<-,igpuMatrix,missing,numeric,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,gpuVector,numeric,missing,missing-method 66 [<-,igpuMatrix,numeric,missing,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,vclMatrix,missing,missing,missing-method 66 [<-,igpuMatrix,numeric,numeric,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,vclMatrix,missing,numeric,missing-method 66 [<-,ivclMatrix,missing,numeric,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,vclMatrix,numeric,missing,missing-method 66 [<-,ivclMatrix,numeric,missing,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,vclMatrix,numeric,numeric,missing-method 66 [<-,ivclMatrix,numeric,numeric,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,vclVector,missing,missing,missing-method 66 [<-,ivclVector,numeric,missing,integer-method ([,gpuMatrix,missing,missing,missing-method), 66

[,vclVector,numeric,missing,missing-method 66 [<-,vclMatrix,missing,missing,matrix-method ([,gpuMatrix,missing,missing,missing-method), 66

[<-,gpuMatrix,missing,numeric,numeric-method 66 [<-,vclMatrix,missing,missing,numeric-method ([,gpuMatrix,missing,missing,missing-method), 66

[<-,gpuMatrix,numeric,missing,numeric-method 66 [<-,vclMatrix,missing,missing,vclMatrix-method

([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclMatrix, missing, missing, vclVector-method %**, gpuVector, gpuMatrix-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclMatrix, missing, numeric, numeric-method %**, matrix, gpuMatrix-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclMatrix, missing, numeric, vclMatrix-method %**, matrix, vclMatrix-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclMatrix, missing, numeric, vclVector-method %**, vclMatrix, matrix-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclMatrix, numeric, missing, numeric-method %**, vclMatrix, vclMatrix-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclMatrix, numeric, numeric, numeric-method %**, vclMatrix, vclVector-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclVector, logical, missing, numeric-method %**, vclVector, vclMatrix-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclVector, missing, missing, numeric-method %**, vclVector, vclVector-method
 ([, gpuMatrix, missing, missing, missing-method), (%**, gpuVector, gpuVector-method),
 66 69
 [<-, vclVector, missing, missing, vclMatrix-method %o%, vclVector, vclVector-method
 ([, gpuMatrix, missing, missing, missing-method), (%o%, gpuVector, gpuVector-method),
 66 69
 [<-, vclVector, missing, missing, vclVector-method %**, gpuVector, gpuVector-method, 69
 ([, gpuMatrix, missing, missing, missing-method), %o%, gpuVector, gpuVector-method, 69
 66
 [<-, vclVector, numeric, missing, numeric-method Arith, gpuMatrix, gpuMatrix-method
 ([, gpuMatrix, missing, missing, missing-method), (Arith, gpuVector, gpuVector-method),
 66 4
 [<-, vclVector, numeric, missing, vclMatrix-method Arith, gpuMatrix, gpuVector-method
 ([, gpuMatrix, missing, missing, missing-method), (Arith, gpuVector, gpuVector-method),
 66 4
 [<-, vclVector, numeric, missing, vclVector-method Arith, gpuMatrix, matrix-method
 ([, gpuMatrix, missing, missing, missing-method), (Arith, gpuVector, gpuVector-method),
 66 4
 %**, gpuMatrix, gpuMatrix-method Arith, gpuMatrix, missing-method
 (%**, gpuVector, gpuVector-method), (Arith, gpuVector, gpuVector-method),
 69 4
 %**, gpuMatrix, gpuVector-method Arith, gpuMatrix, numeric-method
 (%**, gpuVector, gpuVector-method), (Arith, gpuVector, gpuVector-method),
 69 4
 %**, gpuMatrix, matrix-method Arith, gpuVector, gpuMatrix-method
 (Arith, gpuVector, gpuVector-method),

- 4
- Arith,gpuVector,gpuVector-method, 4
- Arith,gpuVector,missing-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,gpuVector,numeric-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,matrix,gpuMatrix-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,matrix,vclMatrix-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,numeric,gpuMatrix-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,numeric,gpuVector-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,numeric,vclMatrix-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,numeric,vclVector-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclMatrix,matrix-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclMatrix,missing-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclMatrix,numeric-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclMatrix,vclMatrix-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclMatrix,vclVector-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclVector,missing-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclVector,numeric-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclVector,vclMatrix-method
(Arith,gpuVector,gpuVector-method),
4
- Arith,vclVector,vclVector-method
(Arith,gpuVector,gpuVector-method),
4
- Arith-gpuR-method
(Arith,gpuVector,gpuVector-method),
4
- Arith-gpuVector-gpuVector-method
(Arith,gpuVector,gpuVector-method),
4
- Arith-gpuVector-missing-method
(Arith,gpuVector,gpuVector-method),
4
- Arith-gpuVector-numeric-method
(Arith,gpuVector,gpuVector-method),
4
- Arith-numeric-gpuVector-method
(Arith,gpuVector,gpuVector-method),
4
- as.gpuMatrix, 6
- as.gpuVector, 7
- as.gpuVector,vector(as.gpuVector), 7
- as.gpuVector,vector-method
(as.gpuVector), 7
- as.vclVector, 8
- as.vclVector,vclMatrix(as.vclVector), 8
- as.vclVector,vclMatrix-method
(as.vclVector), 8
- as.vclVector,vector-method
(as.vclVector), 8
- assert_has_double, 8
- block, 9, 21
- block,gpuMatrix,integer,integer,integer,integer-method
(block), 9
- block,vclMatrix,integer,integer,integer,integer-method
(block), 9
- cgpuMatrix-class, 10
- chol, 11
- chol.gpuMatrix(chol.vclMatrix), 11
- chol.vclMatrix, 11
- colMeans,gpuMatrix-method
(colSums,gpuMatrix-method), 13
- colMeans,vclMatrix-method
(colSums,vclMatrix-method), 14
- colnames, 12

- colnames<- ,gpuMatrix-method (colnames),
12
- colnames<- ,vclMatrix-method (colnames),
12
- colSums,gpuMatrix
(colSums,gpuMatrix-method), 13
- colSums,gpuMatrix-method, 13
- colSums,vclMatrix
(colSums,vclMatrix-method), 14
- colSums,vclMatrix-method, 14
- Compare,gpuVector ,vector-method
(Compare,vector,gpuVector-method),
14
- Compare,vector,gpuVector-method, 14
- Compare-gpuVector-vector
(Compare,vector,gpuVector-method),
14
- Compare-vector-gpuVector
(Compare,vector,gpuVector-method),
14
- cov,gpuMatrix,gpuMatrix,missing,character-method
(cov,vclMatrix,missing,missing,missing-method),
15
- cov,gpuMatrix,gpuMatrix,missing,missing-method
(cov,vclMatrix,missing,missing,missing-method),
15
- cov,gpuMatrix,missing,missing,character-method
(cov,vclMatrix,missing,missing,missing-method),
15
- cov,gpuMatrix,missing,missing,missing-method
(cov,vclMatrix,missing,missing,missing-method),
15
- cov,vclMatrix,missing,missing,character-method
(cov,vclMatrix,missing,missing,missing-method),
15
- cov,vclMatrix,missing,missing,missing-method,
15
- cov,vclMatrix,vclMatrix,missing,character-method
(cov,vclMatrix,missing,missing,missing-method),
15
- cov,vclMatrix,vclMatrix,missing,missing-method
(cov,vclMatrix,missing,missing,missing-method),
15
- cpuInfo, 32
- cpuInfo (gpuInfo), 32
- crossprod,gpuMatrix
(crossprod,gpuMatrix,missing-method),
16
- crossprod,gpuMatrix,gpuMatrix-method
(crossprod,gpuMatrix,missing-method),
16
- crossprod,gpuMatrix,matrix-method
(crossprod,gpuMatrix,missing-method),
16
- crossprod,gpuMatrix,missing-method, 16
- crossprod,matrix,gpuMatrix-method
(crossprod,gpuMatrix,missing-method),
16
- crossprod,matrix,vclMatrix-method
(tcrossprod,gpuVector ,gpuVector-method),
58
- crossprod,vclMatrix
(tcrossprod,gpuVector ,gpuVector-method),
58
- crossprod,vclMatrix,matrix-method
(tcrossprod,gpuVector ,gpuVector-method),
58
- crossprod,vclMatrix,missing-method
(tcrossprod,gpuVector ,gpuVector-method),
58
- crossprod,vclMatrix,vclMatrix-method
(tcrossprod,gpuVector ,gpuVector-method),
58
- crossprod,vclMatrix,vclVector-method
(tcrossprod,gpuVector ,gpuVector-method),
58
- crossprod,vclVector,vclMatrix-method
(tcrossprod,gpuVector ,gpuVector-method),
58
- currentContext, 17
- currentDevice, 18
- currentPlatform, 18
- custom_opengl, 19, 52
- cvclMatrix-class, 19
- deepcopy, 20
- deepcopy,gpuMatrix-method (deepcopy), 20
- deepcopy,gpuVector-method (deepcopy), 20
- deepcopy,vclMatrix-method (deepcopy), 20
- deepcopy,vclVector-method (deepcopy), 20
- det,gpuMatrix-method
(det,vclMatrix-method), 21
- det,vclMatrix (det,vclMatrix-method), 21
- det,vclMatrix-method, 21
- detectCPUs, 22, 32
- detectGPUs, 22, 22, 23, 32
- detectPlatforms, 18, 22, 23, 32

- deviceHasDouble, [9](#), [23](#)
- deviceType, [24](#)
- dgpuMatrix-class, [24](#)
- dgpuVector-class, [25](#)
- diag, [38](#)
- diag, gpuMatrix (diag, vclMatrix-method), [25](#)
- diag, gpuMatrix-method (diag, vclMatrix-method), [25](#)
- diag, vclMatrix (diag, vclMatrix-method), [25](#)
- diag, vclMatrix-method, [25](#)
- diag<- , gpuMatrix, gpuVector (diag, vclMatrix-method), [25](#)
- diag<- , gpuMatrix, gpuVector-method (diag, vclMatrix-method), [25](#)
- diag<- , vclMatrix, vclVector (diag, vclMatrix-method), [25](#)
- diag<- , vclMatrix, vclVector-method (diag, vclMatrix-method), [25](#)
- dim, gpuMatrix-method (dim, vclMatrix-method), [26](#)
- dim, vclMatrix-method, [26](#)
- dim-gpuMatrix (dim, vclMatrix-method), [26](#)
- dim-vclMatrix (dim, vclMatrix-method), [26](#)
- dist, gpuMatrix (distance), [27](#)
- dist, gpuMatrix-method (distance), [27](#)
- dist, vclMatrix (distance), [27](#)
- dist, vclMatrix-method (distance), [27](#)
- distance, [27](#)
- distance, gpuMatrix (distance), [27](#)
- distance, gpuMatrix, gpuMatrix-method (distance), [27](#)
- distance, vclMatrix (distance), [27](#)
- distance, vclMatrix, vclMatrix-method (distance), [27](#)
- dvclMatrix-class, [28](#)
- dvclVector-class, [28](#)
- eigen, gpuMatrix-method, [29](#)
- eigen, vclMatrix (eigen, gpuMatrix-method), [29](#)
- eigen, vclMatrix-method (eigen, gpuMatrix-method), [29](#)
- fgpuMatrix-class, [30](#)
- fgpuVector-class, [30](#)
- fvclMatrix-class, [31](#)
- fvclVector-class, [31](#)
- gpuInfo, [23](#), [32](#)
- gpuMatrix, [9](#), [20](#), [33](#)
- gpuMatrix, integer (gpuMatrix), [33](#)
- gpuMatrix, integer-method (gpuMatrix), [33](#)
- gpuMatrix, matrix (gpuMatrix), [33](#)
- gpuMatrix, matrix-method (gpuMatrix), [33](#)
- gpuMatrix, missing (gpuMatrix), [33](#)
- gpuMatrix, missing-method (gpuMatrix), [33](#)
- gpuMatrix, numeric (gpuMatrix), [33](#)
- gpuMatrix, numeric-method (gpuMatrix), [33](#)
- gpuMatrix-class, [34](#)
- gpuR (gpuR-package), [4](#)
- gpuR-package, [4](#)
- gpuVector, [20](#), [35](#), [54](#)
- gpuVector, missingOrNULL (gpuVector), [35](#)
- gpuVector, missingOrNULL, ANY-method (gpuVector), [35](#)
- gpuVector, numeric, numericOrInt-method (gpuVector), [35](#)
- gpuVector, vector (gpuVector), [35](#)
- gpuVector, vector, missing-method (gpuVector), [35](#)
- gpuVector-class, [36](#)
- has_cpu_skip, [36](#)
- has_double_skip, [37](#)
- has_gpu_skip, [37](#)
- has_multiple_double_skip, [37](#)
- has_multiple_gpu_skip, [38](#)
- identity_matrix, [26](#), [38](#)
- igpuMatrix-class, [39](#)
- igpuVector-class, [39](#)
- inplace, [40](#)
- inplace, function, gpuMatrix, gpuMatrix-method (inplace), [40](#)
- inplace, function, gpuMatrix, missing-method (inplace), [40](#)
- inplace, function, gpuMatrix, numeric-method (inplace), [40](#)
- inplace, function, gpuVector, gpuVector-method (inplace), [40](#)
- inplace, function, gpuVector, missing-method (inplace), [40](#)
- inplace, function, gpuVector, numeric-method (inplace), [40](#)
- inplace, function, numeric, gpuMatrix-method (inplace), [40](#)

- inplace, function, numeric, gpuVector-method
(inplace), 40
- inplace, function, numeric, vclMatrix-method
(inplace), 40
- inplace, function, numeric, vclVector-method
(inplace), 40
- inplace, function, vclMatrix, missing-method
(inplace), 40
- inplace, function, vclMatrix, numeric-method
(inplace), 40
- inplace, function, vclMatrix, vclMatrix-method
(inplace), 40
- inplace, function, vclVector, missing-method
(inplace), 40
- inplace, function, vclVector, numeric-method
(inplace), 40
- inplace, function, vclVector, vclVector-method
(inplace), 40
- ivclMatrix-class, 41
- ivclVector-class, 42

- length, gpuMatrix-method
(length, gpuVector-method), 42
- length, gpuVector-method, 42
- length, vclMatrix-method
(length, gpuVector-method), 42
- length, vclVector-method
(length, gpuVector-method), 42
- length-gpuMatrix
(length, gpuVector-method), 42
- length-vclMatrix
(length, gpuVector-method), 42
- listContexts, 17, 43, 52
- log, gpuMatrix-method
(log, gpuVector-method), 44
- log, gpuVector-method, 44
- log, vclMatrix-method
(log, gpuVector-method), 44
- log, vclVector-method
(log, gpuVector-method), 44
- log-gpuR-method (log, gpuVector-method),
44

- Math, gpuMatrix-method
(Math, gpuVector-method), 45
- Math, gpuVector-method, 45
- Math, vclMatrix-method
(Math, gpuVector-method), 45

- Math, vclVector-method
(Math, gpuVector-method), 45
- Math-gpuR-method
(Math, gpuVector-method), 45

- ncol, gpuMatrix-method
(nrow, vclMatrix-method), 47
- ncol, vclMatrix-method
(nrow, vclMatrix-method), 47

- norm, 46
- norm, ANY, character-method
(norm, vclMatrix, character-method),
46
- norm, ANY, missing-method
(norm, vclMatrix, character-method),
46
- norm, gpuMatrix, character-method
(norm, vclMatrix, character-method),
46
- norm, gpuMatrix, missing-method
(norm, vclMatrix, character-method),
46
- norm, vclMatrix, character-method, 46
- norm, vclMatrix, missing-method
(norm, vclMatrix, character-method),
46

- nrow, gpuMatrix-method
(nrow, vclMatrix-method), 47
- nrow, vclMatrix-method, 47

- permute, 47
- platformInfo, 48
- pmax, 49, 49
- pmin, 49
- pmin (pmax), 49
- pocl_check, 49
- print.gpuMatrix, 50

- qr, 51
- qr.gpuMatrix, 50
- qr.gpuR (qr.gpuMatrix), 50
- qr.Q, 52
- qr.Q, gpuQR-method (qr.R, gpuQR-method),
51
- qr.R, 52
- qr.R, gpuQR-method, 51
- qr.vclMatrix (qr.gpuMatrix), 50

- rowMeans, gpuMatrix-method
(colSums, gpuMatrix-method), 13

- rowMeans, vclMatrix-method
 - (colSums, vclMatrix-method), 14
- rowSums, gpuMatrix
 - (colSums, gpuMatrix-method), 13
- rowSums, gpuMatrix-method
 - (colSums, gpuMatrix-method), 13
- rowSums, vclMatrix
 - (colSums, vclMatrix-method), 14
- rowSums, vclMatrix-method
 - (colSums, vclMatrix-method), 14

- S4groupGeneric, 4, 45, 55
- set_device_context, 53
- setContext, 52
- setup_opencl, 19, 52
- slice, 54
- slice, gpuVector, integer, integer-method
 - (slice), 54
- slice, vclVector, integer, integer-method
 - (slice), 54
- solve, gpuMatrix, gpuMatrix-method
 - (solve, vclMatrix, vclMatrix-method), 55
- solve, gpuMatrix, missing-method
 - (solve, vclMatrix, vclMatrix-method), 55
- solve, vclMatrix, missing-method
 - (solve, vclMatrix, vclMatrix-method), 55
- solve, vclMatrix, vclMatrix-method, 55
- Summary, gpuMatrix-method
 - (Summary, gpuVector-method), 55
- Summary, gpuVector-method, 55
- Summary, vclMatrix-method
 - (Summary, gpuVector-method), 55
- Summary, vclVector-method
 - (Summary, gpuVector-method), 55
- Summary-gpuR-method
 - (Summary, gpuVector-method), 55
- svd, 56, 57
- synchronize, 57

- t, gpuMatrix (t, vclMatrix-method), 58
- t, gpuMatrix-method
 - (t, vclMatrix-method), 58
- t, vclMatrix (t, vclMatrix-method), 58
- t, vclMatrix-method, 58
- tcrossprod, gpuMatrix, gpuMatrix-method
 - (crossprod, gpuMatrix, missing-method), 16
- tcrossprod, gpuMatrix, matrix-method
 - (crossprod, gpuMatrix, missing-method), 16
- tcrossprod, gpuMatrix, missing-method
 - (crossprod, gpuMatrix, missing-method), 16
- tcrossprod, gpuVector, gpuVector-method, 58
- tcrossprod, gpuVector, missing-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, matrix, gpuMatrix-method
 - (crossprod, gpuMatrix, missing-method), 16
- tcrossprod, matrix, vclMatrix-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, vclMatrix, matrix-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, vclMatrix, missing-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, vclMatrix, vclMatrix-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, vclMatrix, vclVector-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, vclVector, missing-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, vclVector, vclMatrix-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- tcrossprod, vclVector, vclVector-method
 - (tcrossprod, gpuVector, gpuVector-method), 58
- typeof, gpuMatrix-method, 60
- typeof, gpuVector-method
 - (typeof, gpuMatrix-method), 60
- typeof, vclMatrix-method
 - (typeof, gpuMatrix-method), 60
- typeof, vclVector-method
 - (typeof, gpuMatrix-method), 60

- vclMatrix, 9, 20, 61
- vclMatrix, integer (vclMatrix), 61

vclMatrix, integer-method (vclMatrix), 61
vclMatrix, matrix (vclMatrix), 61
vclMatrix, matrix-method (vclMatrix), 61
vclMatrix, missing (vclMatrix), 61
vclMatrix, missing-method (vclMatrix), 61
vclMatrix, numeric (vclMatrix), 61
vclMatrix, numeric-method (vclMatrix), 61
vclMatrix, vector (vclMatrix), 61
vclMatrix-class, 62
vclVector, 20, 54, 63
vclVector, missing (vclVector), 63
vclVector, missing, ANY-method
 (vclVector), 63
vclVector, numeric, numericOrInt-method
 (vclVector), 63
vclVector, vclMatrix, missing-method
 (vclVector), 63
vclVector, vector (vclVector), 63
vclVector, vector, missing-method
 (vclVector), 63
vclVector-class, 64

zgpuMatrix-class, 65
zvclMatrix-class, 65