

# Package ‘gpuR’

August 29, 2016

**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** 1.1.2

**Date** 2016-04-05

**Author** Charles Determan Jr.

**Maintainer** Charles Determan Jr. <[cdetermanjr@gmail.com](mailto:cdetermanjr@gmail.com)>

**VignetteBuilder** knitr

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.11.5), assertive, methods

**LinkingTo** Rcpp, RcppEigen, RViennaCL (>= 1.7.1-1), BH

**NeedsCompilation** yes

**Suggests** testthat, knitr

**URL** <http://github.com/cdeterman/gpuR>

**BugReports** <http://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)

**RoxygenNote** 5.0.1

**Repository** CRAN

**Date/Publication** 2016-04-05 17:40:29

**R topics documented:**

gpuR-package . . . . .	3
Arith,gpuMatrix,gpuMatrix-method . . . . .	4
as.gpuMatrix . . . . .	5
as.gpuVector . . . . .	6
block . . . . .	7
colSums,gpuMatrix,missing,missing-method . . . . .	8
colSums,vclMatrix,missing,missing-method . . . . .	8
Compare,vector,gpuVector-method . . . . .	9
cov,gpuMatrix,missing,missing,missing-method . . . . .	10
crossprod,gpuMatrix,missing-method . . . . .	11
crossprod,vclMatrix,missing-method . . . . .	11
currentContext . . . . .	12
currentDevice . . . . .	13
currentPlatform . . . . .	13
deepcopy . . . . .	14
detectCPUs . . . . .	15
detectGPUs . . . . .	15
detectPlatforms . . . . .	16
deviceHasDouble . . . . .	16
dgpuMatrix-class . . . . .	17
dgpuVector-class . . . . .	17
dim,gpuMatrix-method . . . . .	18
distance . . . . .	18
dvclMatrix-class . . . . .	19
dvclVector-class . . . . .	20
eigen,gpuMatrix-method . . . . .	20
fgpuMatrix-class . . . . .	21
fgpuVector-class . . . . .	22
fvclMatrix-class . . . . .	22
fvclVector-class . . . . .	23
gpuInfo . . . . .	23
gpuMatrix . . . . .	24
gpuMatrix-class . . . . .	25
gpuVector . . . . .	26
gpuVector-class . . . . .	26
has_cpu_skip . . . . .	27
has_double_skip . . . . .	27
has_gpu_skip . . . . .	28
igpuMatrix-class . . . . .	28
igpuVector-class . . . . .	28
ivclMatrix-class . . . . .	29
ivclVector-class . . . . .	29
length,gpuVector-method . . . . .	30
listContexts . . . . .	30
log,gpuMatrix-method . . . . .	31
Math,gpuMatrix-method . . . . .	31

nrow,gpuMatrix-method . . . . .	32
platformInfo . . . . .	33
print.gpuMatrix . . . . .	34
setContext . . . . .	34
slice . . . . .	35
Summary,gpuMatrix-method . . . . .	36
typeof,gpuMatrix-method . . . . .	36
vclMatrix . . . . .	37
vclMatrix-class . . . . .	38
vclVector . . . . .	39
vclVector-class . . . . .	40
[,gpuMatrix,missing,missing,missing-method . . . . .	40
%o%,gpuVector,gpuVector-method . . . . .	43
%*%,gpuMatrix,gpuMatrix-method . . . . .	43
<b>Index</b>	<b>45</b>

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 handfull 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:      http://www.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,gpuMatrix,gpuMatrix-method  
*Arith methods*

---

### Description

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

### Usage

```
## S4 method for signature 'gpuMatrix,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 '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 'vclMatrix,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 '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)
```

**Arguments**

e1	A gpuR object
e2	A gpuR object

**Value**

A gpuR object

**Author(s)**

Charles Determan Jr.

---

as.gpuMatrix

*Convert object to a gpuMatrix*


---

**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 gpuMatrix. Default is NULL where type is inherited from the source data type.

**Value**

A gpuVector object

**Author(s)**

Charles Determan Jr.

---

block	<i>Matrix Blocks</i>
-------	----------------------

---

**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 'gpuMatrix,integer,integer,integer,integer'
block(object, rowStart,
      rowEnd, colStart, colEnd)

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

**Arguments**

object	A <a href="#">gpuMatrix</a> or <a href="#">vclMatrix</a> 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 [gpuMatrixBlock](#) or [vclMatrixBlock](#) object

**Author(s)**

Charles Determan Jr.

---

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

---

**Description**

Row and column sums and of gpuMatrix objects

**Usage**

```
## S4 method for signature 'gpuMatrix,missing,missing'  
colSums(x, na.rm, dims)
```

```
## S4 method for signature 'gpuMatrix,missing,missing'  
rowSums(x, na.rm, dims)
```

```
## S4 method for signature 'gpuMatrix,missing,missing'  
colMeans(x, na.rm, dims)
```

```
## S4 method for signature 'gpuMatrix,missing,missing'  
rowMeans(x, na.rm, dims)
```

**Arguments**

x	A gpuMatrix object
na.rm	Not currently used
dims	Not currently used

**Value**

A gpuVector object

**Author(s)**

Charles Determan Jr.

---

colSums,vclMatrix,missing,missing-method  
*Row and Column Sums and Means of vclMatrix*

---

**Description**

Row and column sums and of vclMatrix objects



**Usage**

```
## S4 method for signature 'vclMatrix,missing,missing'  
colSums(x, na.rm, dims)
```

```
## S4 method for signature 'vclMatrix,missing,missing'  
rowSums(x, na.rm, dims)
```

```
## S4 method for signature 'vclMatrix,missing,missing'  
colMeans(x, na.rm, dims)
```

```
## S4 method for signature 'vclMatrix,missing,missing'  
rowMeans(x, na.rm, dims)
```

**Arguments**

x	A vclMatrix object
na.rm	Not currently used
dims	Not currently used

**Value**

A gpuVector 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,gpuMatrix,missing,missing,missing-method  
*Covariance (gpuR)*

---

**Description**

Compute covariance values

**Usage**

```
## S4 method for signature 'gpuMatrix,missing,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 'vclMatrix,missing,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")
```

**Arguments**

x	A gpuR object
y	Not used
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,missing'  
tcrossprod(x, y)  
  
## S4 method for signature 'gpuMatrix,gpuMatrix'  
tcrossprod(x, y)
```

### Arguments

x	A gpuMatrix
y	A gpuMatrix

### Value

A gpuMatrix

### Author(s)

Charles Determan Jr.

---

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

**Arguments**

x	A vclMatrix
y	A vclMatrix

**Value**

A vclMatrix

**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](#)

---

`deepcopy`*Copy a "gpuR" object*

---

**Description**

This is needed to make a duplicate of a gpuR object

**Usage**

```
deepcopy(object)
```

```
## S4 method for signature 'gpuMatrix'  
deepcopy(object)
```

```
## S4 method for signature 'gpuVector'  
deepcopy(object)
```

```
## S4 method for signature 'vclMatrix'  
deepcopy(object)
```

```
## S4 method for signature 'vclVector'  
deepcopy(object)
```

**Arguments**

`object`            A gpuR object

**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

**Author(s)**

Charles Determan Jr.

**See Also**

[block](#)

---

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

---

**Description**

Find out how many CPUs available

**Usage**

```
detectCPUs(platform_idx = 1L)
```

**Arguments**

platform\_idx    An integer value indicating which platform to query.

**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 = 1L)
```

**Arguments**

platform\_idx    An integer value indicating which platform to query.

**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(platform_idx = 1L, gpu_idx = 1L)
```

**Arguments**

<code>platform_idx</code>	An integer value indicating which platform to query.
<code>gpu_idx</code>	An integer value indicating which gpu to query.

**Value**

A boolean designating whether the device supports double precision

**See Also**

[gpuInfo](#)



---

`dgpuMatrix-class`      *dgpuMatrix Class*

---

**Description**

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

**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.

**Slots**

`address`: Pointer to a double typed vector

**Author(s)**

Charles Determan Jr.

**See Also**

[gpuVector-class](#)

---

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

---

### Description

Retrieve dimension of object

### Usage

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

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

### Arguments

x                    A gpuMatrix/vclMatrix object

### Value

A length 2 vector of the number of rows and columns respectively.

### Author(s)

Charles Determan Jr.

---

distance                    *GPU Distance Matrix Computations*

---

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

```
## S4 method for signature 'gpuMatrix,gpuMatrix'
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")
```

### 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 corresponding distances

---

dvclMatrix-class	<i>dvclMatrix Class</i>
------------------	-------------------------

---

### Description

An integer type matrix in the S4 vclMatrix representation.

### 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 `vc1Vector` representation.

**Slots**

`address`: Pointer to a double typed vector

**Author(s)**

Charles Determan Jr.

**See Also**

[vc1Vector-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 'vc1Matrix'
eigen(x, symmetric, only.values = FALSE,
      EISPACK = FALSE)
```

**Arguments**

<code>x</code>	A <code>gpuMatrix</code> object
<code>symmetric</code>	logical indication if matrix is assumed to be symmetric. If not specified or <code>FALSE</code> , the matrix is inspected for symmetry
<code>only.values</code>	if <code>TRUE</code> , returns only eigenvalues (internals still currently calculate both regardless)
<code>EISPACK</code>	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 <code>gpuVector</code> containing the unsorted eigenvalues of <code>x</code> .
vectors	A <code>gpuMatrix</code> containing the unsorted eigenvectors of <code>x</code>

**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	<i>fgpuMatrix Class</i>
------------------	-------------------------

---

**Description**

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

**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.

**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.

**Slots**

address: Pointer to a float matrix.

**Author(s)**

Charles Determan Jr.

**See Also**

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

---

fvclVector-class	<i>fvclVector Class</i>
------------------	-------------------------

---

**Description**

An float vector in the S4 vclVector representation.

**Slots**

address: Pointer to a float typed vector

**Author(s)**

Charles Determan Jr.

**See Also**

[vclVector-class](#)

---

gpuInfo	<i>Device Information</i>
---------	---------------------------

---

**Description**

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

**Usage**

```
gpuInfo(platform_idx = 1L, device_idx = 1L)
```

```
cpuInfo(platform_idx = 1L, device_idx = 1L)
```

**Arguments**

platform\_idx An integer value indicating which platform to query.

device\_idx An integer value indicating which device 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

**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)
```

```
## S4 method for signature 'missing'
gpuMatrix(data, nrow = NA, ncol = NA, type = NULL)
```

```
## S4 method for signature 'numeric'
gpuMatrix(data, nrow, ncol, type = NULL)
```

```
## S4 method for signature 'integer'
gpuMatrix(data, nrow, ncol, type = 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



**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.

**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

---

*Construct a gpuVector*


---

**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)

## S4 method for signature 'missingOrNULL,ANY'
gpuVector(data, length, type = 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

**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.

**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	<i>Skip test for CPUs</i>
--------------	---------------------------

---

**Description**

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

**Usage**

has\_cpu\_skip()

---

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()

---

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()

---

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

---

**Description**

An integer type matrix in the S4 gpuMatrix representation.

**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.

**Slots**

address: An integer vector object

**Author(s)**

Charles Determan Jr.

**See Also**

[gpuVector-class](#)

---

ivclMatrix-class      *ivclMatrix Class*

---

**Description**

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

**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.

**Slots**

address: An integer vector object

**Author(s)**

Charles Determan Jr.

**See Also**

[vclVector-class](#)

---

length, gpuVector-method  
*Length of gpuVector*

---

**Description**

Get the length of a gpuR vector object

**Usage**

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

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

**Arguments**

x                    A gpuVector/vclVector object

**Value**

A numeric value

---

listContexts            *Available OpenCL Contexts*

---

**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.gpuMatrix-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 'gpuMatrix'
log(x, base = NULL)
```

```
## 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)
```

### 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 gpuR object of the same class as x

---

Math.gpuMatrix-method *gpuR Math methods*

---

### Description

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

**Usage**

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

**Arguments**

x                    A gpuR object

**Details**

Currently implemented methods include:

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

**Value**

A gpuR object

**Author(s)**

Charles Determan Jr.

---

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



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

### Arguments

x                    A gpuMatrix/vclMatrix object

### Value

An integer of length 1

### 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.

---

print.gpuMatrix	<i>S3 print for gpuMatrix objects</i>
-----------------	---------------------------------------

---

**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

---

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

---

**Description**

Change the current context used by default

**Usage**

```
setContext(id)
```

**Arguments**

id	Integer identifying which context to set
----	--

**See Also**

[listContexts](#)

---

slice	<i>Vector Slices</i>
-------	----------------------

---

### 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 <a href="#">gpuVector</a> or <a href="#">vclVector</a> 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.

---

Summary,gpuMatrix-method

*gpuR Summary methods*

---

### Description

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

### Usage

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

```
## 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)
```

### 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

---

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

**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)  
  
## S4 method for signature 'missing'  
vclMatrix(data, nrow = NA, ncol = NA, type = NULL)  
  
## S4 method for signature 'numeric'  
vclMatrix(data, nrow, ncol, type = NULL)  
  
## S4 method for signature 'integer'  
vclMatrix(data, nrow, ncol, type = 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 vclMatrix. Default is NULL where type is inherited from the source data type.
...	Additional method to pass to vclMatrix methods

**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.

**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

*Construct a vclVector*


---

**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)
```

```
## S4 method for signature 'missing,ANY'
vclVector(data, length, type = 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 vclVector. Default is NULL where type is inherited from the source data type.
...	Additional method to pass to vclVector methods

**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.

### 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](#)

---

[,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]  
  
## 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]

## 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 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

```

**Arguments**

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

**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

**Author(s)**

Charles Determan Jr.

---

%%,gpuMatrix,gpuMatrix-method  
*Matrix Multiplication*

---

**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 'gpuMatrix,gpuMatrix'
x %% y
```

```
## S4 method for signature 'gpuVector,gpuVector'
x %% y
```

```
## S4 method for signature 'vclMatrix,vclMatrix'
x %% y
```

```
## S4 method for signature 'vclVector,vclVector'
x %% y
```

**Arguments**

x	A gpuR object
y	A gpuR object

**Author(s)**

Charles Determan Jr.

# Index

[,gpuMatrix,missing,missing,missing-method, ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,gpuMatrix,missing,numeric,missing-method [<-,gpuMatrix,numeric,numeric,numeric-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,gpuMatrix,numeric,missing,missing-method [<-,gpuVector,numeric,missing,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,gpuMatrix,numeric,numeric,missing-method [<-,gpuVector,numeric,missing,numeric-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,gpuVector,missing,missing,missing-method [<-,igpuMatrix,missing,numeric,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,gpuVector,numeric,missing,missing-method [<-,igpuMatrix,numeric,missing,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,vclMatrix,missing,missing,missing-method [<-,igpuMatrix,numeric,numeric,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,vclMatrix,missing,numeric,missing-method [<-,ivclMatrix,missing,numeric,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,vclMatrix,numeric,missing,missing-method [<-,ivclMatrix,numeric,missing,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,vclMatrix,numeric,numeric,missing-method [<-,ivclMatrix,numeric,numeric,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,vclVector,missing,missing,missing-method [<-,ivclVector,numeric,missing,integer-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[,vclVector,numeric,missing,missing-method [<-,vclMatrix,missing,numeric,numeric-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

[<-,gpuMatrix,missing,numeric,numeric-method [<-,vclMatrix,numeric,missing,numeric-method  
([,gpuMatrix,missing,missing,missing-method), ([,gpuMatrix,missing,missing,missing-method),  
40 40

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

- [, gpuMatrix, missing, missing, missing-method, [Arith](#)), [c1Matrix](#), [missing-method](#)  
[40](#) (Arith, gpuMatrix, gpuMatrix-method),
- [<- , vclVector, numeric, missing, numeric-method [4](#)  
([, gpuMatrix, missing, missing, missing-method, [Arith](#)), [c1Matrix](#), [numeric-method](#)  
[40](#) (Arith, gpuMatrix, gpuMatrix-method),
- %\*%, gpuVector, gpuVector-method  
(%\*%, gpuMatrix, gpuMatrix-method), [Arith](#), [vclMatrix](#), [vclMatrix-method](#)  
[43](#) (Arith, gpuMatrix, gpuMatrix-method),
- %\*%, vclMatrix, vclMatrix-method  
(%\*%, gpuMatrix, gpuMatrix-method), [Arith](#), [vclVector](#), [missing-method](#)  
[43](#) (Arith, gpuMatrix, gpuMatrix-method),
- %\*%, vclVector, vclVector-method  
(%\*%, gpuMatrix, gpuMatrix-method), [Arith](#), [vclVector](#), [numeric-method](#)  
[43](#) (Arith, gpuMatrix, gpuMatrix-method),
- %\*%-gpuR-method (gpuMatrix), [24](#)  
[4](#)
- %O%, vclVector, vclVector-method  
(%O%, gpuVector, gpuVector-method), [Arith](#), [vclVector](#), [vclVector-method](#)  
[43](#) (Arith, gpuMatrix, gpuMatrix-method),
- %\*%, gpuMatrix, gpuMatrix-method, [43](#)  
[Arith-gpuR-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- %O%, gpuVector, gpuVector-method, [43](#)  
[Arith-gpuVector-gpuVector-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, gpuMatrix, gpuMatrix-method, [4](#)  
[Arith-gpuVector-missing-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, gpuMatrix, missing-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, gpuMatrix, numeric-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, gpuVector, gpuVector-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, gpuVector, missing-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, gpuVector, numeric-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, numeric, gpuMatrix-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, numeric, gpuVector-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, numeric, vclMatrix-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- Arith, numeric, vclVector-method  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- [Arith-gpuR-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- [Arith-gpuVector-gpuVector-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- [Arith-gpuVector-missing-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- [Arith-gpuVector-numeric-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- [Arith-numeric-gpuVector-method](#)  
(Arith, gpuMatrix, gpuMatrix-method),  
[4](#)
- as.gpuMatrix, [5](#)
- as.gpuVector, [6](#)
- as.gpuVector, vector (as.gpuVector), [6](#)
- as.gpuVector, vector-method  
(as.gpuVector), [6](#)
- block, [7](#), [14](#)
- block, gpuMatrix, integer, integer, integer, integer-method  
(block), [7](#)
- block, vclMatrix, integer, integer, integer, integer-method  
(block), [7](#)
- colMeans, gpuMatrix, missing, missing-method  
(colSums, gpuMatrix, missing, missing-method),  
[8](#)

colMeans, vclMatrix, missing, missing-method (crossprod, vclMatrix, missing-method),  
 (colSums, vclMatrix, missing, missing-method), 11  
 8  
 colSums, gpuMatrix currentContext, 12  
 (colSums, gpuMatrix, missing, missing-method), 13  
 8 currentPlatform, 13  
 colSums, gpuMatrix, missing, missing-method, deepcopy, 14  
 8 deepcopy, gpuMatrix-method (deepcopy), 14  
 colSums, vclMatrix deepcopy, gpuVector-method (deepcopy), 14  
 (colSums, vclMatrix, missing, missing-method), 14  
 8 deepcopy, vclMatrix-method (deepcopy), 14  
 colSums, vclMatrix, missing, missing-method, deepcopy, vclVector-method (deepcopy), 14  
 8  
 Compare, gpuVector, vector-method detectCPUs, 15, 24  
 (Compare, vector, gpuVector-method), detectGPUs, 15, 15, 16, 24  
 9 detectPlatforms, 13, 15, 16, 24  
 Compare, vector, gpuVector-method, 9 deviceHasDouble, 16  
 Compare-gpuVector-vector dgpuMatrix-class, 17  
 (Compare, vector, gpuVector-method), dgpuVector-class, 17  
 9 dim, gpuMatrix-method, 18  
 Compare-vector-gpuVector dim, vclMatrix-method  
 (Compare, vector, gpuVector-method), (dim, gpuMatrix-method), 18  
 9 dim-gpuMatrix (dim, gpuMatrix-method), 18  
 cov, gpuMatrix, missing, missing, character-method dim-vclMatrix (dim, gpuMatrix-method), 18  
 (cov, gpuMatrix, missing, missing, missing-method), dist, gpuMatrix (distance), 18  
 10 dist, gpuMatrix-method (distance), 18  
 cov, gpuMatrix, missing, missing, missing-method, dist, vclMatrix (distance), 18  
 10 dist, vclMatrix-method (distance), 18  
 cov, vclMatrix, missing, missing, character-method distance, 18  
 (cov, gpuMatrix, missing, missing, missing-method), distance, gpuMatrix (distance), 18  
 10 distance, gpuMatrix, gpuMatrix-method  
 cov, vclMatrix, missing, missing, missing-method (distance), 18  
 (cov, gpuMatrix, missing, missing, missing-method), distance, vclMatrix (distance), 18  
 10 distance, vclMatrix, vclMatrix-method  
 (cov, gpuMatrix, missing, missing, missing-method), (distance), 18  
 10  
 cpuInfo, 24  
 cpuInfo (gpuInfo), 23  
 crossprod, gpuMatrix  
 (crossprod, gpuMatrix, missing-method),  
 11  
 crossprod, gpuMatrix, gpuMatrix-method  
 (crossprod, gpuMatrix, missing-method),  
 11  
 crossprod, gpuMatrix, missing-method, 11  
 crossprod, vclMatrix  
 (crossprod, vclMatrix, missing-method),  
 11  
 crossprod, vclMatrix, missing-method, 11  
 crossprod, vclMatrix, vclMatrix-method  
 (crossprod, vclMatrix, missing-method),  
 11  
 crossprod, vclMatrix, missing-method, 11  
 crossprod, vclMatrix, vclMatrix-method  
 (crossprod, vclMatrix, missing-method),  
 11  
 currentContext, 12  
 currentDevice, 13  
 currentPlatform, 13  
 deepcopy, 14  
 deepcopy, gpuMatrix-method (deepcopy), 14  
 deepcopy, gpuVector-method (deepcopy), 14  
 deepcopy, vclMatrix-method (deepcopy), 14  
 deepcopy, vclVector-method (deepcopy), 14  
 detectCPUs, 15, 24  
 detectGPUs, 15, 15, 16, 24  
 detectPlatforms, 13, 15, 16, 24  
 deviceHasDouble, 16  
 dgpuMatrix-class, 17  
 dgpuVector-class, 17  
 dim, gpuMatrix-method, 18  
 dim, vclMatrix-method  
 (dim, gpuMatrix-method), 18  
 dim-gpuMatrix (dim, gpuMatrix-method), 18  
 dim-vclMatrix (dim, gpuMatrix-method), 18  
 dist, gpuMatrix (distance), 18  
 dist, gpuMatrix-method (distance), 18  
 dist, vclMatrix (distance), 18  
 dist, vclMatrix-method (distance), 18  
 distance, 18  
 distance, gpuMatrix (distance), 18  
 distance, gpuMatrix, gpuMatrix-method  
 (distance), 18  
 distance, vclMatrix (distance), 18  
 distance, vclMatrix, vclMatrix-method  
 (distance), 18  
 dvclMatrix-class, 19  
 dvclVector-class, 20  
 eigen, gpuMatrix-method, 20  
 eigen, vclMatrix  
 (eigen, gpuMatrix-method), 20  
 eigen, vclMatrix-method  
 (eigen, gpuMatrix-method), 20  
 fgpuMatrix-class, 21  
 fgpuVector-class, 22  
 fvclMatrix-class, 22  
 fvclVector-class, 23  
 gpuInfo, 16, 23  
 gpuMatrix, 7, 14, 24

- gpuMatrix, integer (gpuMatrix), 24
- gpuMatrix, integer-method (gpuMatrix), 24
- gpuMatrix, matrix (gpuMatrix), 24
- gpuMatrix, matrix-method (gpuMatrix), 24
- gpuMatrix, missing (gpuMatrix), 24
- gpuMatrix, missing-method (gpuMatrix), 24
- gpuMatrix, numeric (gpuMatrix), 24
- gpuMatrix, numeric-method (gpuMatrix), 24
- gpuMatrix-class, 25
- gpuR (gpuR-package), 3
- gpuR-package, 3
- gpuVector, 14, 26, 35
- gpuVector, missingOrNULL (gpuVector), 26
- gpuVector, missingOrNULL, ANY-method (gpuVector), 26
- gpuVector, vector (gpuVector), 26
- gpuVector, vector, missing-method (gpuVector), 26
- gpuVector-class, 26
- has\_cpu\_skip, 27
- has\_double\_skip, 27
- has\_gpu\_skip, 28
- igpuMatrix-class, 28
- igpuVector-class, 28
- ivclMatrix-class, 29
- ivclVector-class, 29
- length, gpuVector-method, 30
- length, vclVector-method (length, gpuVector-method), 30
- listContexts, 12, 30, 34
- log, gpuMatrix-method, 31
- log, gpuVector-method (log, gpuMatrix-method), 31
- log, vclMatrix-method (log, gpuMatrix-method), 31
- log, vclVector-method (log, gpuMatrix-method), 31
- log-gpuR-method (log, gpuMatrix-method), 31
- Math, gpuMatrix-method, 31
- Math, gpuVector-method (Math, gpuMatrix-method), 31
- Math, vclMatrix-method (Math, gpuMatrix-method), 31
- Math, vclVector-method (Math, gpuMatrix-method), 31
- Math-gpuR-method (Math, gpuMatrix-method), 31
- ncol, gpuMatrix-method (nrow, gpuMatrix-method), 32
- ncol, vclMatrix-method (nrow, gpuMatrix-method), 32
- nrow, gpuMatrix-method, 32
- nrow, vclMatrix-method (nrow, gpuMatrix-method), 32
- platformInfo, 33
- print.gpuMatrix, 34
- rowMeans, gpuMatrix, missing, missing-method (colSums, gpuMatrix, missing, missing-method), 8
- rowMeans, vclMatrix, missing, missing-method (colSums, vclMatrix, missing, missing-method), 8
- rowSums, gpuMatrix, missing, missing-method (colSums, gpuMatrix, missing, missing-method), 8
- rowSums, vclMatrix, missing, missing-method (colSums, vclMatrix, missing, missing-method), 8
- S4groupGeneric, 4, 31, 36
- setContext, 34
- slice, 35
- slice, gpuVector, integer, integer-method (slice), 35
- slice, vclVector, integer, integer-method (slice), 35
- Summary, gpuMatrix-method, 36
- Summary, gpuVector-method (Summary, gpuMatrix-method), 36
- Summary, vclMatrix-method (Summary, gpuMatrix-method), 36
- Summary, vclVector-method (Summary, gpuMatrix-method), 36
- Summary-gpuR-method (Summary, gpuMatrix-method), 36
- tcrossprod, gpuMatrix, gpuMatrix-method (crossprod, gpuMatrix, missing-method), 11
- tcrossprod, gpuMatrix, missing-method (crossprod, gpuMatrix, missing-method), 11



tcrossprod, vclMatrix, missing-method  
(crossprod, vclMatrix, missing-method),  
11

tcrossprod, vclMatrix, vclMatrix-method  
(crossprod, vclMatrix, missing-method),  
11

typeof, gpuMatrix-method, 36

typeof, gpuVector-method  
(typeof, gpuMatrix-method), 36

typeof, vclMatrix-method  
(typeof, gpuMatrix-method), 36

typeof, vclVector-method  
(typeof, gpuMatrix-method), 36

  

vclMatrix, 7, 14, 37

vclMatrix, integer (vclMatrix), 37

vclMatrix, integer-method (vclMatrix), 37

vclMatrix, matrix (vclMatrix), 37

vclMatrix, matrix-method (vclMatrix), 37

vclMatrix, missing (vclMatrix), 37

vclMatrix, missing-method (vclMatrix), 37

vclMatrix, numeric-method (vclMatrix), 37

vclMatrix, vector (vclMatrix), 37

vclMatrix-class, 38

vclVector, 14, 35, 39

vclVector, missing (vclVector), 39

vclVector, missing, ANY-method  
(vclVector), 39

vclVector, vector (vclVector), 39

vclVector, vector, missing-method  
(vclVector), 39

vclVector-class, 40