# Package ‘gpuR’

January 5, 2017

**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** 1.2.1

**Date** 2016-12-30

**Author** Charles Determan Jr.

**Maintainer** Charles Determan Jr. <cdetermanjr@gmail.com>

**VignetteBuilder** knitr

**License** GPL (>= 2)

**Depends** R (>= 3.0.2), methods, utils

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

**LinkingTo** Rcpp, RcppEigen, RViennaCL (>= 1.7.1.4), 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** 5.0.1.9000

**Repository** CRAN

**Date/Publication** 2017-01-05 19:03:11

# R topics documented:

## 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,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 '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)

## 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)
```

## 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 *Convert object to a gpuVector*

### 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.

---

## as.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

```
as.vclVector(data, shared, ...)

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)
```

### Arguments

| | |
|---|---|
| `data` | An object that is or can be converted to a `vector` |
| `shared` | Logical indicating if memory should be shared with `x` |
| `...` | Additional method to pass to vclVector methods |
| `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. |
| `ctx_id` | An integer specifying the object's context |

### Value

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(plat_idx, device_idx,
  severity = getOption("assertive.severity", "stop"))
```

### Arguments

| | |
|---|---|
| plat_idx | An integer indicating which platform to query |
| device_idx | An integer indicating which device 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 gpuMatrix or vclMatrix 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 continous 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.

---

# chol.vclMatrix *Cholesky Decomposition of a gpuR matrix*

---

## 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` *Row and Column Names*

---

### 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 |

## 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,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,missing,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,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` *Current Context*

### Description

Get current context index

### Usage

```
currentContext()
```

### Value

An integer reflecting the context listed in listContexts

## `currentDevice` *Current Device Information*

### 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` *Return Current Platform*

### 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 '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

### Author(s)

Charles Determan Jr.

### See Also

`block`

## `detectCPUs` *Detect Available OpenCL enabled CPUs*

### 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` *Detect Available GPUs*

### 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` *Detect Number of Platforms*

### 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` *Check GPU double precision support*

### 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

| | |
|---|---|
| `platform_idx` | An integer value indicating which platform to query. |
| `gpu_idx` | An integer value indicating which gpu to query. |

### Value

A boolean designating whether the device supports double precision

### See Also

gpuInfo

## `deviceType` *Check device type*

### Description

Check what type a device is given platform and device indices

### Usage

```
deviceType(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

A character string indicating the device type

## 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`

# 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 gpuR vector 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 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 '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 corresponding distances

## `dvclMatrix-class` *dvclMatrix Class*

### 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 `vclVector` representation.

### 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

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 *fvclVector Class*

### 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 *Device Information*

### Description

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

### Usage

```
gpuInfo(platform_idx = NULL, device_idx = NULL)

cpuInfo(platform_idx = NULL, device_idx = NULL)
```

### 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 |
| `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` *Construct a gpuMatrix*

### 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.

### 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, ctx_id = NULL)

## S4 method for signature 'missingOrNULL,ANY'
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`.

### 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 testthat tests if no valid CPU's are detected

### Usage

```
has_cpu_skip()
```

## `has_double_skip` *Skip test for GPU double precision*

### Description

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

### Usage

```
has_double_skip()
```

## `has_gpu_skip` *Skip test for GPUs*

### Description

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

### Usage

```
has_gpu_skip()
```

## `has_multiple_double_skip` *Skip test for multiple GPUs with double precision*

### Description

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

### Usage

```
has_multiple_double_skip()
```

## `has_multiple_gpu_skip` *Skip test in less than 2 GPUs*

### Description

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

### Usage

```
has_multiple_gpu_skip()
```

## 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 `getOption("gpuR.default.type")`. |

### 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 *igpuMatrix Class*

### 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` *igpuVector Class*

### 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 *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

A numeric value

### Author(s)

Charles Determan Jr.

## 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,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 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"

### Value

A gpuR object

### Author(s)

Charles Determan Jr.

## `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 of length 1

### Author(s)

Charles Determan Jr.

## permute *Permuting functions for* gpuR *objects*

### Description

Generate a perumutation of row or column indices

### Usage

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

### Arguments

| | |
|---|---|
| x | A gpuR matrix object |
| MARGIN | dimenstion 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 *OpenCL Platform Information*

### 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` | Avaiable platform extensions |

### Author(s)

Charles Determan Jr.

## `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 |

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

### Description

`qr` computs 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 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

## qr.R,gpuQR-method *Reconstruct the Q or R Matrices form a gpuQR Object*

### 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

| | |
|---|---|
| `qr` | `gpuQR` object |
| `complete` | 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` *Set Context*

### Description

Change the current context used by default

### Usage

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

### Arguments

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

### See Also

listContexts

# 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 continous 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 funciton 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 *Singular Value Decomposition of a gpuR matrix*

### 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 an S3 generic of svd. The default continues to point to the default base function.

## Author(s)

Charles Determan Jr.

## See Also

svd

---

# `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

A `gpuR` matrix

## 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 |

### 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

| | |
|---|---|
| `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 |
| `ctx_id` | 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 propogated 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-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

---

`[,gpuVector,missing,missing,missing-method`
*Extract gpuR object elements*

---

## Description

Operators to extract or replace elements

## Usage

```
## 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 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,missing,missing,numeric'
x[i, j] <- value

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

## 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
```

### 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 |
| ... | Additional arguments |

### Author(s)

Charles Determan Jr.

## %o%,gpuVector,gpuVector-method

*Outer Product*

### Description

The outer product of two gpuR vector objects

### Usage

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

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

### Arguments

| | |
|---|---|
| X | A gpuR object |
| Y | A gpuR object |

### Author(s)

Charles Determan Jr.

---

## %*%,gpuVector,gpuVector-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 'gpuVector,gpuVector'
x %*% y

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

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

## S4 method for signature 'gpuMatrix,gpuMatrix'
x %*% y
```

### Arguments

| | |
|---|---|
| x | A gpuR object |
| y | A gpuR object |

### Author(s)

Charles Determan Jr.

# Index