

# Package ‘cgdsr’

June 26, 2019

**Type** Package

**Title** R-Based API for Accessing the MSKCC Cancer Genomics Data Server (CGDS)

**Version** 1.3.0

**Date** 2019-06-10

**Author** Anders Jacobsen <jacobsen@cbio.mskcc.org>, Augustin Luna <lunaa@cbio.mskcc.org>

**Maintainer** Benjamin Gross <cgdsr@cbioportal.org>

**Description** Provides a basic set of R functions for querying the Cancer Genomics Data Server (CGDS), hosted by the Computational Biology Center at Memorial-Sloan-Kettering Cancer Center (MSKCC) at <www.cbioportal.org>.

**License** LGPL-3

**LazyLoad** yes

**URL** <https://github.com/cBioPortal/cgdsr>

**Depends** R (>= 3.1)

**Imports** R.oo, R.methodsS3, httr

**Suggests** testthat

**NeedsCompilation** no

**Repository** CRAN

**RoxygenNote** 6.0.1

**Date/Publication** 2019-06-26 11:50:22 UTC

## R topics documented:

cgdsr-package . . . . .	2
cgdsr-CGDS . . . . .	3
cgdsr-getCancerStudies . . . . .	4
cgdsr-getCaseLists . . . . .	5
cgdsr-getClinicalData . . . . .	7
cgdsr-getGeneticProfiles . . . . .	8
cgdsr-getMutationData . . . . .	9

cgdsr-getProfileData . . . . .	11
cgdsr-plot . . . . .	13
cgdsr-processURL . . . . .	15
cgdsr-setPlotErrorMsg . . . . .	15
cgdsr-setVerbose . . . . .	16
cgdsr-test . . . . .	17

<b>Index</b>	<b>19</b>
--------------	-----------

---

cgdsr-package	<i>CGDS-R : a library for accessing data in the MSKCC Cancer Genomics Data Server (CGDS).</i>
---------------	---

---

## Description

The package provides a basic set of R functions for querying the Cancer Genomics Data Server (CGDS), hosted by the Computational Biology Center at Memorial-Sloan-Kettering Cancer Center (MSKCC). Read more about this service at the cBio Cancer Genomics Portal, <http://www.cbioportal.org/>.

## Details

Package:	cgdsr
Type:	Package
License:	GPL
LazyLoad:	yes

The Cancer Genomic Data Server (CGDS) web service interface provides direct programmatic access to all genomic data stored within the server. This package provides a basic set of R functions for querying the CGDS hosted by the Computational Biology Center at Memorial-Sloan-Kettering Cancer Center (MSKCC).

The library can issue the following types of queries:

1. *getCancerStudies()*: What cancer studies are hosted on the server? For example TCGA Glioblastoma or TCGA Ovarian cancer.
2. *getGeneticProfiles()*: What genetic profile types are available for cancer study X? For example mRNA expression or copy number alterations.
3. *getCaseLists()*: what case sets are available for cancer study X? For example all samples or only samples corresponding to a given cancer subtype.
4. *getProfileData()*: Retrieve slices of genomic data. For example, a client can retrieve all mutation data from PTEN and EGFR in TCGA glioblastoma.
5. *getClinicalData()*: Retrieve clinical data (e.g. patient survival time and age) for a given case list.

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[CGDS](#), [getCancerStudies](#), [getGeneticProfiles](#), [getCaseLists](#), [getProfileData](#), [getClinicalData](#).

**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

# Test the CGDS endpoint URL using a few simple API tests
test(mycgds)

# Get list of cancer studies at server
getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
mycancerstudy = getCancerStudies(mycgds)[2,1]
mycaselist = getCaseLists(mycgds,mycancerstudy)[1,1]

# Get available genetic profiles
mygeneticprofile = getGeneticProfiles(mycgds,mycancerstudy)[4,1]

# Get data slices for a specified list of genes, genetic profile and case list
getProfileData(mycgds,c('BRCA1','BRCA2'),mygeneticprofile,mycaselist)

# Get clinical data for the case list
myclinicaldata = getClinicalData(mycgds,mycaselist)
```

---

cgdsr-CGDS

*Construct a CGDS connection object*

---

**Description**

Creates a CGDS connection object from a CGDS endpoint URL. This object must be passed on to the methods which query the server.

**Usage**

```
CGDS(url,verbose=FALSE,ploterrormsg='',token=NULL)
```

**Arguments**

url	A CGDS URL (required).
verbose	A boolean variable specifying verbose output (default FALSE)
ploterrormsg	An optional message to display in plots if an error occurs (default ”)
token	An optional 'Authorization: Bearer' token to connect to cBioPortal instances that require authentication (default NULL)

**Value**

A CGDS connection object. This object must be passed on to the methods which query the server.

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[cgdsr.getCancerStudies](#), [getGeneticProfiles](#), [getCaseLists](#), [getProfileData](#)

**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

# Test the CGDS endpoint URL using a few simple API tests
test(mycgds)

# Get list of cancer studies at server
getCancerStudies(mycgds)
```

---

cgdsr-getCancerStudies

*Get available cancer studies available in CGDS*

---

**Description**

Queries the CGDS API and returns available cancer studies. Input is a CGDS object and output is a data.matrix with information regarding the different cancer studies.

**Usage**

```
## S3 method for class 'CGDS'
getCancerStudies(x, ...)
```

**Arguments**

x                    A CGDS object (required)  
...                   Not used.

**Value**

A data.frame with three columns:

1. *cancer\_study\_id*: unique ID used to identify the cancer study in subsequent interface calls. This is a human readable ID.
2. *name*: short name of the cancer type.
3. *description*: short description of the cancer type, describing the source of study.

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[cgdsr,CGDS,getGeneticProfiles,getCaseLists](#)

**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
mycancerstudy = getCancerStudies(mycgds)[2,1]
mycaselist = getCaseLists(mycgds,mycancerstudy)[1,1]
```

---

cgdsr-getCaseLists      *Get available case lists for a specific cancer study*

---

**Description**

Queries the CGDS API and returns available case lists for a specific cancer study.

**Usage**

```
## S3 method for class 'CGDS'
getCaseLists(x,cancerStudy,...)
```

## Arguments

x	A CGDS object (required)
cancerStudy	cancer study ID (required)
...	Not used.

## Details

Queries the CGDS API and returns available case lists for a specific cancer study. For example, a within a particular study, only some cases may have sequence data, and another subset of cases may have been sequenced and treated with a specific therapeutic protocol. Multiple case lists may be associated with each cancer study, and this method enables you to retrieve meta-data regarding all of these case lists.

## Value

A data.frame with five columns:

1. *case\_list\_id*: a unique ID used to identify the case list ID in subsequent interface calls. This is a human readable ID. For example, "gbm\_tcga\_all" identifies all cases profiles in the TCGA GBM study.
2. *case\_list\_name*: short name for the case list.
3. *case\_list\_description*: short description of the case list.
4. *cancer\_study\_id*: cancer study ID tied to this genetic profile. Will match the input cancer\_study\_id.
5. *case\_ids*: space delimited list of all case IDs that make up this case list.

## Author(s)

<jacobsen@cbio.mskcc.org>

## References

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

## See Also

[cgdsr](#), [CGDS](#), [getCancerStudies](#), [getGeneticProfiles](#), [getProfileData](#)

## Examples

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

# Get list of cancer studies at server
getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
mycancerstudy = getCancerStudies(mycgds)[2,1]
```

```
mycaselist = getCaseLists(mycgds,mycancerstudy)[1,1]

# Get available genetic profiles
mygeneticprofile = getGeneticProfiles(mycgds,mycancerstudy)[1,1]

# Get data slices for a specified list of genes, genetic profile and case list
getProfileData(mycgds,c('BRCA1','BRCA2'),mygeneticprofile,mycaselist)
```

---

cgdsr-getClinicalData *Get clinical data for cancer study*

---

## Description

Queries the CGDS API and returns clinical data for a given case list.

## Usage

```
## S3 method for class 'CGDS'
getClinicalData(x, caseList, cases, caseIdsKey, ...)
```

## Arguments

x	A CGDS object (required)
caseList	A case list ID
cases	A vector of case IDs
caseIdsKey	Only used by web portal.
...	Not used.

## Value

A data.frame with rows for each case, rownames corresponding to case IDs, and columns:

1. *overall\_survival\_months*: Overall survival, in months.
2. *overall\_survival\_status*: Overall survival status, usually indicated as "LIVING" or "DECEASED".
3. *disease\_free\_survival\_months*: Disease free survival, in months.
4. *disease\_free\_survival\_status*: Disease free survival status, usually indicated as "DiseaseFree" or "Recurred/Progressed".
5. *age\_at\_diagnosis*: Age at diagnosis.

## Author(s)

<jacobsen@cbio.mskcc.org>

## References

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[cgdsr](#), [CGDS](#), [getCaseLists](#)

**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
mycancerstudy = getCancerStudies(mycgds)[2,1]
mycaselist = getCaseLists(mycgds,mycancerstudy)[1,1]

# Get clinical data for caselist
getClinicalData(mycgds,mycaselist)
```

---

cgdsr-getGeneticProfiles

*Get available genetic data profiles for a specific cancer study*

---

**Description**

Queries the CGDS API and returns the available genetic profiles, e.g. mutation or copy number profiles, stored about a specific cancer study.

**Usage**

```
## S3 method for class 'CGDS'
getGeneticProfiles(x, cancerStudy, ...)
```

**Arguments**

x	A CGDS object (required)
cancerStudy	cancer study ID (required)
...	Not used.

**Value**

A data.frame with six columns:

1. *genetic\_profile\_id*: a unique ID used to identify the genetic profile ID in subsequent interface calls. This is a human readable ID. For example, "gbm\_tcga\_mutations" identifies the TCGA GBM mutation genetic profile.
2. *genetic\_profile\_name*: short profile name.
3. *genetic\_profile\_description*: short profile description.



4. *cancer\_study\_id*: cancer study ID tied to this genetic profile. Will match the input *cancer\_study\_id*.
5. *genetic\_alteration\_type*: indicates the profile type. Will be one of: MUTATION, MUTATION\_EXTENDED, COPY\_NUMBER\_ALTERATION, MRNA\_EXPRESSION.
6. *show\_profile\_in\_analysis\_tab*: a boolean flag used for internal purposes (you can safely ignore it).

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[cgdsr](#), [CGDS](#), [getCancerStudies](#), [getCaseLists](#), [getProfileData](#)

**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

# Get list of cancer studys at server
getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
mycancerstudy = getCancerStudies(mycgds)[2,1]
mycaselist = getCaseLists(mycgds,mycancerstudy)[1,1]

# Get available genetic profiles
mygeneticprofile = getGeneticProfiles(mycgds,mycancerstudy)[1,1]

# Get data slices for a specified list of genes, genetic profile and case list
getProfileData(mycgds,c('BRCA1','BRCA2'),mygeneticprofile,mycaselist)
```

---

cgdsr-getMutationData *Get mutation data for cancer study*

---

**Description**

Queries the CGDS API and returns mutation data for a given case set and list of genes.

**Usage**

```
## S3 method for class 'CGDS'
getMutationData(x, caseList, geneticProfile, genes, ...)
```

**Arguments**

x	A CGDS object (required)
caseList	A case list ID
geneticProfile	A genetic profile ID with mutation data
genes	A vector of query genes
...	Not used.

**Value**

A data.frame with rows for each sample/case, rownames corresponding to case IDs, and columns corresponding to:

1. *entrez\_gene\_id*: Entrez gene ID
2. *gene\_symbol*: HUGO gene symbol
3. *sequencing\_center*: Sequencer Center responsible for identifying this mutation.
4. *mutation\_status*: somatic or germline mutation status. all mutations returned will be of type somatic.
5. *age\_at\_diagnosis*: Age at diagnosis.
6. *mutation\_type*: mutation type, such as nonsense, missense, or frameshift\_ins.
7. *validation\_status*: validation status. Usually valid, invalid, or unknown.
8. *amino\_acid\_change*: amino acid change resulting from the mutation.
9. *functional\_impact\_score*: predicted functional impact score, as predicted by Mutation Assessor.
10. *xvar\_link*: Link to the Mutation Assessor web site.
11. *xvar\_link\_pdb*: Link to the Protein Data Bank (PDB) View within Mutation Assessor web site.
12. *xvar\_link\_msa*: Link the Multiple Sequence Alignment (MSA) view within the Mutation Assessor web site.
13. *chr*: chromosome where mutation occurs.
14. *start\_position*: start position of mutation.
15. *end\_position*: end position of mutation

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[cgdsr,CGDS](#)

## Examples

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
# Get Extended Mutation Data for EGFR and PTEN in TCGA GBM

getMutationData(mycgds,gbm_tcga_all,gbm_tcga_mutations,c('EGFR', 'PTEN'))
```

---

cgdsr-getProfileData *Retrieves genomic profile data for genes and genetic profiles.*

---

## Description

Queries the CGDS API and returns data based on gene(s), genetic profile(s), and a case list.

## Usage

```
## S3 method for class 'CGDS'
getProfileData(x,genes,geneticProfiles,caseList,cases,caseIdsKey,...)
```

## Arguments

x	A CGDS object (required)
genes	A vector of gene names or a String specifying a single gene (required)
geneticProfiles	A vector of genetic profile IDs or String specifying a single genetic profile (required)
caseList	A case list ID
cases	A vector of case IDs)
caseIdsKey	Only used by web portal.
...	Not used.

## Details

Only one list is allowed, specify either a list of genes or genetic profiles. The format of the output data.frame depends on if a single or a list of genes was specified in the arguments.

**Value**

When requesting one or multiple genes and a single genetic profile, the function returns a data.frame with genetic profile data in columns for each gene.

When requesting a single gene and multiple genetic profiles, the function returns a data.frame containing columns with data for each genetic profile.

Cases can be specified either through a case list ID, or a vector of case IDs.

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[cgdsr](#), [CGDS](#), [getCancerStudies](#), [getGeneticProfiles](#), [getCaseLists](#)

**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

# Get list of cancer studies at server
getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
mycancerstudy = getCancerStudies(mycgds)[2,1]
mycaselist = getCaseLists(mycgds,mycancerstudy)[1,1]

# Get available genetic profiles
mygeneticprofile = getGeneticProfiles(mycgds,mycancerstudy)[1,1]

# Get data slices for a specified list of genes, genetic profile and case list
getProfileData(mycgds,c('BRCA1', 'BRCA2'),mygeneticprofile,mycaselist)

# Get data slice for a single gene
getProfileData(mycgds,'HMGA2',mygeneticprofile,mycaselist)

# Get data slice for multiple genetic profiles and single gene
getProfileData(mycgds,'HMGA2',getGeneticProfiles(mycgds,mycancerstudy)[c(1,2),1],mycaselist)

# Get the same dataset from a vector of case IDs
cases = unlist(strsplit(getCaseLists(mycgds,mycancerstudy)[1,'case_ids'],' '))
getProfileData(mycgds,'HMGA2',getGeneticProfiles(mycgds,mycancerstudy)[c(1,2),1],cases=cases)
```

---

cgdsr-plot

*Generic plot function for CGDS API data.*


---

**Description**

Queries the CGDS API and plots data for specified genes and genetic profiles.

**Usage**

```
## S3 method for class 'CGDS'
plot(x,cancerStudy, genes, geneticProfiles,
     caseList, cases, caseIdsKey, skin, skin.normals, skin.col.gp, add.corr, legend.pos, ...)
```

**Arguments**

x	A CGDS object (required)
cancerStudy	cancer study ID (required)
genes	A vector of gene names or a String specifying a single gene (required)
geneticProfiles	A vector of genetic profile IDs or String specifying a single genetic profile (required)
caseList	A case list ID
cases	A vector of case IDs)
caseIdsKey	Only used by web portal.
skin	A string specifying which plotting layout skin to use (default is continuous data 'cont')
skin.normals	Specify a case list ID with normal samples, only some skins handle normal data.
skin.col.gp	Specify a vector of additional case list IDs to use for color coding of data points. Color coding is only handled by some skins.
add.corr	Computes correlation between the two data vectors. Specify correlation method ('pearson' or 'spearman') as argument.
legend.pos	Position of legend in plot (default is 'topright').
...	Not used.

**Details**

Queries the CGDS API and plots data for specified genes and genetic profiles.

The following combinations are allowed:

1. 1 gene and 1 genetic profile. Plots genetic profile data histogram for specified gene.
2. 2 genes and 1 genetic profile. Scatter plot of continuous genetic profile data for the two genes.
3. 3 1 gene and 2 genetic profiles. Scatterplot or boxplot relating two genetic profile datasets for single gene.

The function currently implements the following skins:

1. *cont*: This is the default skin. It treats all data as being continuous.
2. *disc*: Requires a single gene and a single genetic profile. The genetic profile data is handled as a discrete dataset and barplot is returned. being continuous.
3. *disc\_cont*: Requires two genetic profiles. The first dataset is handled as being discrete data, and the function generates a boxplot with distributions for each level of the discrete genetic profile.
4. *cna\_mrna\_mut*: This skin plots mRNA expression level as function of copy number status for a given gene. Data points are colored by mutation status if specified (*skin.col.gp*), and normal data points are included if specified (*skin.normals*).
5. *cna\_mrna\_mut*: This skin plots mRNA expression level as function of DNA methylation status for a given gene. Data points are colored by copy number and mutation status if specified (two element vector of copy number and mutation genetic profiles specified for *skin.col.gp*). Normal data points are included if specified (*skin.normals*).

### Author(s)

<jacobsen@cbio.mskcc.org>

### References

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

### See Also

[cgdsr](#), [CGDS](#), [getCancerStudies](#), [getGeneticProfiles](#), [getProfileData](#)

### Examples

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

# Get list of cancer studies at server
getCancerStudies(mycgds)

# Get available case lists (collection of samples) for a given cancer study
mycancerstudy = getCancerStudies(mycgds)[2,1]
mycaselist = getCaseLists(mycgds,mycancerstudy)[1,1]

# Get available genetic profiles
mygeneticprofile = getGeneticProfiles(mycgds,mycancerstudy)[4,1]

# histogram of genetic profile data for gene
plot(mycgds,mycancerstudy,'MDM2',mygeneticprofile,mycaselist)

# scatter plot of genetic profile data for two genes
plot(mycgds,mycancerstudy,c('MDM2','MDM4'),mygeneticprofile,mycaselist)

# See vignette for more details ...
```

---

cgdsr-processURL      *Internal methods for CGDS library.*

---

**Description**

These methods should not be invoked by the user.

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**

[cgdsr,CGDS](#)

---

cgdsr-setPlotErrorMsg      *Set custom plot error message*

---

**Description**

Sets custom plot error message.

**Usage**

```
## S3 method for class 'CGDS'  
setPlotErrorMsg(x, msg, ...)
```

**Arguments**

x	A CGDS object (required)
msg	A custom message (string)
...	Not used.

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**[cgdsr,CGDS](#)**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

getCancerStudies(mycgds)

# Set custom error plot message
setPlotErrorMsg(mycgds, 'My message ...')

getCancerStudies(mycgds)
```

---

`cgdsr-setVerbose`*Set verbose logging level for CGDS function calls*

---

**Description**

Sets verbose logging level for CGDS function calls.

**Usage**

```
## S3 method for class 'CGDS'
setVerbose(x, verbose, ...)
```

**Arguments**

<code>x</code>	A CGDS object (required)
<code>verbose</code>	Activate verbose logging (boolean)
<code>...</code>	Not used.

**Author(s)**

<jacobsen@cbio.mskcc.org>

**References**

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

**See Also**[cgdsr,CGDS](#)



## Examples

```
# Create CGDS object
mycgds = CGDS("http://www.cbioportal.org/")

getCancerStudies(mycgds)

# Activate verbose logging
setVerbose(mycgds, TRUE)

getCancerStudies(mycgds)
```

---

cgdsr-test

*Simple test suite for CGDS object.*

---

## Description

Queries the CGDS API and returns results of the tests.

## Usage

```
## S3 method for class 'CGDS'
test(x, ...)
```

## Arguments

x	A CGDS object.
...	Not used.

## Details

A set of simple tests are evaluated. The format of the returned output from the following queries are tested: "getCancerStudies()", "getCaseLists()", and "getGeneticProfiles()"

## Value

Test results in text format.

## Author(s)

<jacobsen@cbio.mskcc.org>

## References

cBio Cancer Genomics Portal: <http://www.cbioportal.org/>

## See Also

[cgdsr,CGDS](#)

**Examples**

```
# Create CGDS object
mycgds = CGDS("http://www.cbiportal.org/")

# Run tests
test(mycgds)
```

# Index

## \*Topic **package**

cgdsr-package, 2

CGDS, 3, 5, 6, 8–10, 12, 14–17

CGDS (cgdsr-CGDS), 3

cgdsr, 4–6, 8–10, 12, 14–17

cgdsr (cgdsr-package), 2

cgdsr-CGDS, 3

cgdsr-getCancerStudies, 4

cgdsr-getCaseLists, 5

cgdsr-getClinicalData, 7

cgdsr-getGeneticProfiles, 8

cgdsr-getMutationData, 9

cgdsr-getProfileData, 11

cgdsr-package, 2

cgdsr-plot, 13

cgdsr-processURL, 15

cgdsr-setPlotErrorMsg, 15

cgdsr-setVerbose, 16

cgdsr-test, 17

getCancerStudies, 3, 4, 6, 9, 12, 14

getCancerStudies

(cgdsr-getCancerStudies), 4

getCaseLists, 3–5, 8, 9, 12

getCaseLists (cgdsr-getCaseLists), 5

getClinicalData, 3

getClinicalData

(cgdsr-getClinicalData), 7

getGeneticProfiles, 3–6, 12, 14

getGeneticProfiles

(cgdsr-getGeneticProfiles), 8

getMutationData

(cgdsr-getMutationData), 9

getProfileData, 3, 4, 6, 9, 14

getProfileData (cgdsr-getProfileData),

11

plot (cgdsr-plot), 13

processURL (cgdsr-processURL), 15

setPlotErrorMsg

(cgdsr-setPlotErrorMsg), 15

setVerbose (cgdsr-setVerbose), 16

test (cgdsr-test), 17