

# Package ‘panoply’

October 3, 2017

**Version** 0.95

**Date** 2017-5-23

**Title** Personalized Cancer Report

**Description** Personalized Cancer Report linking driver mutations, outlying gene expression events, with druggable targets.

**Depends** R (>= 3.2.3), xtable, MASS, circlize, gage, Rgraphviz, RColorBrewer

**License** GPL (>= 2)

**Suggests** R.rsp

**VignetteBuilder** R.rsp

**URL** <http://bioinformaticstools.mayo.edu/research/panoply/>

## R topics documented:

clinPanBC . . . . .	1
clinPanCO . . . . .	2
cnaPanCO . . . . .	3
cnaPanTNBC . . . . .	3
dgidbPan . . . . .	4
dgiSets . . . . .	4
gcinfoPan . . . . .	5
gcPanCO . . . . .	5
gcPanTNBC . . . . .	6
genelistPan . . . . .	6
panCircos . . . . .	7
panDrugGraph . . . . .	8
panDrugSets . . . . .	9
panGeneGraph . . . . .	11
panGeneSets . . . . .	12
variantPanTNBC . . . . .	14

---

 clinPanBC

*Clinical Data for Breast Cancer Subjects*


---

**Description**

Clinical Data of Breast Cancer patients to be used in Panoply

**Usage**

```
data("clinPanBC")
```

**Format**

A data frame with 6 subjects. One non-responder and 5 matched responders

PatientID character, patient identifier

Clinical.Molecular.Subtype character, Clinical Molecular Subtype Triple Negative

Breast.Nodal.CR.all character, Complete Response yes no

T.stage character, T stage T2

N.stage character, N stage N0

Percentage.tumor.cells.baseline.internal integer, Percentage Tumor Cells

Age.Group ordered factor, Age in 10 year categories 60-69 < 70+

**Examples**

```
data(clinPanBC)
str(clinPanBC)
```

---

 clinPanCO

*Clinical Data for TCGA Colon Cancer Patients*


---

**Description**

Clinical Data of TCGA Colon Cancer patients to be used in Panoply

**Usage**

```
data("clinPanCO")
```

**Format**

A data frame with 9 subjects. 5 cases (deceased) and 4 matched controls (alive)

bcr\_patient\_barcode character, patient identifier, TCGA barcode

bcr\_patient\_uuid character, TCGA uuid

patient\_id character, simplified patient identifier

days\_to\_death character, days to death

days\_to\_last\_followup character, days to last followup

race character, race, WHITE  
 vital\_status character, vital status Alive Dead  
 gender character, gender FEMALE MALE  
 kras\_mutation\_found character, kras mutation present YES  
 surv character, alive recur  
 CN.tumorpct numeric, Tumor Percentage

### Examples

```
data(clinPanCO)
str(clinPanCO)
```

---

cnaPanCO	<i>DNA CNV (germline) and CNA (tumor) data for the TCGA Colon Cancer Subjects</i>
----------	---

---

### Description

Per-Gene DNA Copy Number Variation (germline) and Copy Number Alteration (tumor) data for the TCGA Colon Cancer Subjects

### Usage

```
data("cnaPanCO")
```

### Format

A data frame with 6 subjects. One non-responder and 5 matched responders

CHROM chromosome of gene  
 START gene start position  
 STOP gene stop position  
 Gene.Symbol gene symbol, NCBI  
 TCGA-DM-A0XD patient identifier

### Examples

```
data(cnaPanCO)
str(cnaPanCO)
```

---

`cnaPanTNBC`*DNA CNV (germline) and CNA (tumor) data for TCGA samples*

---

**Description**

Per-Gene DNA Copy Number Variation (germline) and Copy Number Alteration (tumor) data for TCGA samples

**Usage**

```
data("cnaPanTNBC")
```

**Format**

A data frame with 77 subjects.

CHROM chromosome of gene

START gene start position

STOP gene stop position

geneid Gene ID

Gene.Symbol gene symbol, NCBI

TCGA-A2-A04T patient identifier

**Examples**

```
data(cnaPanBC)
str(cnaPanBC)
```

---

`dgidbPan`*Druggable Genome Interaction Database for Cancer*

---

**Description**

Druggable Genome Interaction Database annotation for Cancer Drugs and their targets

**Usage**

```
data("dgidbPan")
```

**Format**

Gene Gene Symbol

Drugs All cancer drugs that target the gene

---

`dgiSets`*Druggable Genome Interaction Database for Cancer*

---

**Description**

Druggable Genome Interaction Database annotation for Cancer Drugs and their targets

**Usage**

```
data("dgiSets")
```

**Format**

The adjacency matrix `dgi.adj` has drugs (rows) and a 1 marked where the drug targets a gene (columns). The same information is stored as a list of gene names for each drug in `dgi.gs`. The data.frame `dgiDrug` has each drug in a row, with genes they target, and sources for which drug database contained it in these columns:

`Drug` Cancer drugs

`GeneID` Gene Symbols of genes targeted by the drug

`Synonym` Drug synonyms

`DGIdb` Logical, curated by DGI-db?

`DrugBankCurate` Logical, curated by Drug Bank list?

---

`gcinfoPan`*Gene information data for 7574 genes*

---

**Description**

Gene information Data for 7574 genes

**Usage**

```
data("gcinfoPan")
```

**Format**

A data frame with gene information

`CHROM` Chromosome

`START` Gene start position

`STOP` Gene stop position

`Gene.Sybmol` Gene

`CodingLength` Gene coding length

**Examples**

```
data(gcinfoPan)
str(gcinfoPan)
```

---

gcPanCO	<i>Dataset with cqn-normalized RNA gene count data for TCGA colon cancer</i>
---------	--

---

**Description**

Dataset with cqn-normalized RNA gene count data for TCGA colon cancer

**Usage**

```
data("gcPanCO")
```

**Format**

A data frame with genes (rows) and patients (columns).

TCGA-DM-A0XD normalized gene count data for patient TCGA-DM-A0XD

**Examples**

```
data(gcPanCO)
str(gcPanCO)
```

---

gcPanTNBC	<i>Dataset with cqn-normalized RNA gene count data for TCGA samples</i>
-----------	---

---

**Description**

Dataset with cqn-normalized RNA gene count data for TCGA samples

**Usage**

```
data("gcPanTNBC")
```

**Format**

A data frame with 5490 genes (rows) and 79 patients (columns).

**Examples**

```
data(gcPanTNBC)
str(gcPanTNBC)
```

---

genelistPan	<i>Curated Cancer Gene list for panoply</i>
-------------	---

---

**Description**

Curated Cancer Gene list for panoply

**Usage**

```
data("genelistPan")
```

**Format**

A data frame with gene information

**Examples**

```
data(genelistPan)
str(genelistPan)
```

---

panCircos	<i>plot method for panPath/panDruggable objects</i>
-----------	---

---

**Description**

Create Circos Plot of copy number, RNA expression events, germline and somatic variants. By default, plots DNA mutation (variant/cna) and RNA expression events and shows connections between events in top pathways with a driver gene event. Allows either germline/somatic variants, cna, or all three.

**Usage**

```
panCircos(panGene, panDrug, caseids, controlids, variant = NULL,
          cna = NULL, gcount, gcinfo, controls = FALSE, tumorpct = 0.5,
          tailPct = 0.05, tailEnd = "upper", minTargets = 1, minPathPct = 0.05,
          minPathSize = 8, minPathways = 1, nperm = 500, ...)
```

**Arguments**

panGene	A data.frame of drug test results from panGeneSets
panDrug	A data.frame of results from panDrugSets
caseids	identifiers of subjects that will match to variant column PatientID, and column name of cna and gcount.
controlids	identifiers of subjects in row names of variant and column names of gcount who are controls
variant	data.frame for somatic and/or germline variants. Must contain columns CHROM, POS, Gene.Symbol, SampleType (Germline, Tumor) and PatientID that must match the caseid and controlid

cna	data.frame with log2 of copy number aberrations. Required columns CHROM, START, STOP, Gene.Symbol, and columns named to match caseid and controlid
gcount	Normalized gene expression counts for patients (columns) at gene symbols (rows). Gene symbols are the row names and should match dataset gcinfoPan gene symbols.
controls	TRUE/FALSE to get Cont.GenesPath from ppath object rather than Case.
tumorpct	Vector of approximate tumor percentage for each patient samples given in the order they appear in cases and controls. If a single value is passed, it is used for all patients. Copy number mutation events are called based on their approximate tumor percentage, transformed to the log2-ratio copy number call.
tailPct	percentile used to define the most extreme RNA expression events as cancer events per patient; only used when eventOnly=TRUE. Values 0 < tailPct < .5 allowed.
tailEnd	perform differential expression tests that are one-sided ("upper" or "lower"), or two-sided ("both"). Also used with tailPct to include both, upper, or lower expression events to be used as cancer events.
minTargets	Retain drug results for drugs that target at least minTargets genes in the cancer and network genes.
minPathPct	Of reactome pathways affected by gene networks targeted by the drug, only count those with total druggable by the drug in the pathway divided by the total pathway size >= minPathPct.
minPathSize	Of reactome pathways affected by gene networks targeted by the drug, only count the pathways that have total genes >= minPathSize.
minPathways	Filter drug results to those that target genes in at least minPathways, after accounting for minPathPct and minPathSize.
nperm	Number of simulations to perform for evaluating significance of meta drug tests

### Details

circos plot of the driver gene events in outer bands, and gene expression events in the inner bands, colored blue if they are over- or under-expressed, and if one of the key drugs target them. Across the center circle, we show genes that are connected as the top cancer drivers connected to their druggable in-network genes.

### Value

Nothing is returned

---

panDrugGraph                      *Drug Network Graph*

---

### Description

Graph that shows the connection between the drugs, genes and pathways for the drugs that target the most genes

### Usage

```
panDrugGraph(panDrug, usePathIDs = TRUE, ndrugs=8)
```



**Arguments**

panDrug	A data.frame of drug test results from panDrugSets
usePathIDs	If TRUE the shorter reactome pathway IDs will be used in the graph instead of the complete name
ndrugs	Minimum number of Drugs to include in the graph

**Details**

Graph where Drugs=Red Ellipse, Genes=Blue Ellipse, Pathways=Green Rectangles

**Value**

Nothing is returned

**Examples**

```

data(clinPanTNBC)
data(genelistPan)
data(cnaPanTNBC)
data(gcPanTNBC)
data(gcinfoPan)
data(variantPanTNBC)
data(dgidbPan)
data(dgiSets)
data(reactome)
patient <- "TCGA-B6-A0IK"
match.index <- which(clinPanTNBC$Vital.Status=="Alive" & clinPanTNBC$PatientID!=patient &
ptmatch <- clinPanTNBC[match.index,"PatientID"]
drivGenes <- panGeneSets(caseid=patient, controlid=ptmatch, eventOnly=TRUE, variant=variantPanTNBC,
cna=cnaPanTNBC, gcount=gcPanTNBC, tumorpct=0.3, tailEnd="upper", tailPct=0.1)
set.seed(1000)
drugResults <- panDrugSets(drivGenes, caseids=patient,
controlids=ptmatch, gcount=gcPanTNBC,nperm=200, tailEnd="upper")
drugResults<- drugResults[!is.na(drugResults$Pathways) & !is.na(drugResults$N.Cancer.Gene) & !is.na(drugResults$Drug)]
panDrugGraph(drugResults)

```

---

panDrugSets

*Tests on drugs for their effectiveness in targeting cancer genes in cases*


---

**Description**

Tests on drugs for their effectiveness in targeting activated cancer genes in cases against their matched controls.

**Usage**

```

panGeneSets(panGene, caseids, controlids, gcount, minTargets = 1,
minPathPct = 0.05, minPathSize = 8, minPathways = 1, pathSource = "reactome",
nperm = 1000, tailEnd = "both")

```

**Arguments**

panGene	A data.frame of drug test results from panGeneSets
caseids	identifiers of subjects case subjects that will match to variant column PatientID, and column name of cna and gcount.
controlids	identifiers of subjects in row names of variant and column names of gcount who are controls
gcount	Normalized gene expression counts for patients (columns) at gene symbols (rows). Gene symbols are the row names and should match dataset gcinfoPan gene symbols.
minTargets	Retain drug results for drugs that target at least minTargets genes in the cancer and network genes.
minPathPct	Of reactome pathways affected by gene networks targeted by the drug, only count those with total druggable by the drug in the pathway divided by the total pathway size $\geq$ minPathPct.
minPathSize	Of reactome pathways affected by gene networks targeted by the drug, only count the pathways that have total genes $\geq$ minPathSize.
minPathways	Filter drug results to those that target genes in at least minPathways, after accounting for minPathPct and minPathSize.
pathSource	Source of pathway gene sets. For now, only Reactome included.
tailEnd	For Drug Network tests, perform differential expression tests that are one-sided ("upper" or "lower"), or two-sided ("both"). Also used with tailPct to include both, upper, or lower expression events to be used as cancer events.
nperm	Number of simulations to perform for evaluating significance of meta drug tests

**Details**

If tailEnd is upper, then testing for drugs that target over-expressed genes and gene networks in case(s) versus controls.

**Value**

A data.frame with the following columns

- Drug: Drug Name
- N.Cancer.Genes: number of cancer genes that are targeted by the drug
- Cancer.Genes: cancer genes that are targeted by the drug
- N.Network.Genes: number of network genes targeted by the drug
- Network.Genes: network genes targeted by the drug
- N.Pathways: number of pathways with genes targeted by the drug (meeting minPathPct and minPathSize settings)
- Pathways: pathway names of pathways with a gene targeted by the drug
- Network: Number of genes directly targeted by the drug
- Network.pval: p-value comparing gene expression between caseids and controlids patients in the genes directly targeted by the drug
- ZSim: Meta Z-statistic of combined gene networks for any gene targeted by the drug
- ZSim.pval: p-value for Meta ZSim statistic, based on multiplying  $s_i \sim N(0,1)$  by gene network test Z-statistics for  $i=1..nperm$

- **Z.stat**: Meta Z-statistic of combined gene networks for any gene targeted by the drug, weighted by number of genes targeted by the drug divided by gene network size
- **ZPerm.pval**: p-value for Meta Z.stat statistic, based on multiplying  $s_i \sim N(0,1)$  by gene network test Z-statistics for  $i=1..nperm$
- **PScore**: A score used to sort the drug test results, which is a sum of the  $-\log_{10}(pval)$  of `Network.pval` and `ZSim.pval`

## Examples

```

data(clinPanTNBC)
data(genelistPan)
data(cnaPanTNBC)
data(gcPanTNBC)
data(gcinfoPan)
data(variantPanTNBC)
data(dgidbPan)
data(dgiSets)
data(reactome)
patient <- "TCGA-B6-A0IK"
match.index <- which(clinPanTNBC$Vital.Status=="Alive" & clinPanTNBC$PatientID!=patient &
ptmatch <- clinPanTNBC[match.index, "PatientID"]
drivGenes <- panGeneSets(caseid=patient, controlid=ptmatch, eventOnly=TRUE, variant=variantPanTNBC,
cna=cnaPanTNBC, gcount=gcPanTNBC, tumorpct=0.3, tailEnd="upper", tailPct=0.1)
set.seed(1000)
drugResults <- panDrugSets(drivGenes, caseids=patient,
controlids=ptmatch, gcount=gcPanTNBC, nperm=200, tailEnd="upper")

drugResults[1:10, !grepl("^Pathways", names(drugResults))]

```

---

panGeneGraph

*Creates reactome network graph for Genes connected to top Drugs*

---

## Description

Uses Reactome data to create graph that shows the connection between Driver Genes and Connected Genes for the top Drugs in panDrug

## Usage

```
panGeneGraph(panGene, panDrug, minTargets = 1, minPathways = 1, ndrugs = 4, ndrivers = 20, ...)
```

## Arguments

panGene	A data.frame of drug test results from panGeneSets
panDrug	A data.frame of drug test results from panDrugSets
minTargets	Retain drug results for drugs that target at least minTargets genes in the cancer and network genes.
minPathways	Filter drug results to those that target genes in at least minPathways, after accounting for minPathPct and minPathSize.
ndrugs	Number of Drugs to use to find Cancer Driver and Network genes in graph
ndrivers	Number of unique Cancer Drive genes in graph

**Details**

A graph that shows the Cancer Driver Network for the top Drugs in pan Drug. Circles=Druggable, Ellipse=Expressed Druggable Drivers, Red=Cancer Driver Genes, Blue=Network Genes

**Value**

nothing is returned

**Examples**

```
data(clinPanTNBC)
data(genelistPan)
data(cnaPanTNBC)
data(gcPanTNBC)
data(gcinfoPan)
data(variantPanTNBC)
data(dgidbPan)
data(dgiSets)
data(reactome)
patient <- "TCGA-B6-A0IK"
match.index <- which(clinPanTNBC$Vital.Status=="Alive" & clinPanTNBC$PatientID!=patient &
ptmatch <- clinPanTNBC[match.index,"PatientID"]
drivGenes <- panGeneSets(caseid=patient, controlid=ptmatch, eventOnly=TRUE, variant=variantPanTNBC,
cna=cnaPanTNBC, gcount=gcPanTNBC, tumorpct=0.3, tailEnd="upper", tailPct=0.1)
set.seed(1000)
drugResults <- panDrugSets(drivGenes, caseids=patient,
controlids=ptmatch, gcount=gcPanTNBC,nperm=200, tailEnd="upper")

panGeneGraph(drivGenes,drugResults)
```

---

panGeneSets

*Test gene networks for a set of cases versus a set of controls*


---

**Description**

Test differential gene expression in gene networks for a set of cases versus a set of controls, with gene networks defined by reactome directed graph for cancer genes, and genomic events determined from variant, cna, or gcount.

**Usage**

```
panGeneSets(caseids, controlids, variant = NULL, cna = NULL,
gcount, tumorpct = 0.5, tailPct = 0.1, tailEnd =
"both", eventOnly = FALSE, gageCompare =
ifelse(length(caseids) > 1, "as.group", "unpaired"))
drivDNA(ids, variant = NULL, cna = NULL, tumorpct = 0.5)
outrRNA(ids, gcount, tailPct = 0.1, tailEnd = "both")
```

**Arguments**

caseids	identifiers of subjects case subjects that will match to variant column PatientID, and column name of cna and gcount.
controlids	identifiers of subjects in row names of variant and column names of gcount who are controls
ids	identifiers of subjects in row names of variant and column names of gcount and cna matrices
variant	data.frame for somatic and/or germline variants. Must contain columns CHROM, POS, Gene.Symbol, SampleType (Germline, Tumor) and PatientID that must match the caseid and controlid
cna	data.frame with log2 of copy number aberrations. Required columns CHROM, START, STOP, Gene.Symbol, and columns named to match caseid and controlid
gcount	Normalized gene expression counts for patients (columns) at gene symbols (rows). Gene symbols are the row names and should match dataset gcinfoPan gene symbols.
tumorpct	Vector of approximate tumor percentage for each patient samples given in the order they appear in cases and controls. If a single value is passed, it is used for all patients. Copy number mutation events are called based on their approximate tumor percentage, transformed to the log2-ratio copy number call.
tailPct	percentile used to define the most extreme RNA expression events as cancer events per patient; only used when eventOnly=TRUE. Values 0 < tailPct < .5 allowed.
tailEnd	perform differential expression tests that are one-sided ("upper" or "lower"), or two-sided ("both"). Also used with tailPct to include both, upper, or lower expression events to be used as cancer events.
gageCompare	Character string telling the gage package, which performs the gene network tests, to perform tests of 1-vs-M ("unpaired") or N-vs-M ("as.group")
eventOnly	Logical, test only gene networks that have a cancer event (variant, cna, and rna-expression outside tailPct)).

**Details**

If eventOnly=FALSE, then test all gene networks for differential expression between cases and controls. Otherwise, cancer events are determined per gene from variant (any variant in the gene for the case(s)), cna (copy gain or loss for case(s) given tumorpct), and gcount (outlying expression from tailPct and tailEnd settings) and only those gene networks with a cancer event are tested.

**Value**

A data.frame with the following columns

- Cancer.Gene: genes whose networks were tested
- Network: total connections of the cancer gene to other genes via reactome adjacency matrix
- Network.pval: p-value comparing gene expression between caseids and controlids patients
- Network.mean: Z-statistic of differential expression between caseids and controlids
- Druggable: Is the cancer gene druggable? 1=TRUE
- Total.Druggable: Number of the connected network genes that are druggable
- NetGenes: Network genes that are over- or under-expressed
- NetDrugs: Drug names that would target the genes in Total.Druggable
- Drugs: Drug names that would target the cancer gene

**Examples**

```

data(clinPanTNBC)
data(genelistPan)
data(cnaPanTNBC)
data(gcPanTNBC)
data(gcinfoPan)
data(variantPanTNBC)
data(dgidbPan)
data(dgiSets)
data(reactome)

patient <- "TCGA-B6-A0IK"
match.index <- which(clinPanTNBC$Vital.Status=="Alive" & clinPanTNBC$PatientID!=patient &
ptmatch <- clinPanTNBC[match.index, "PatientID"]
drivGenes <- panGeneSets(caseid=patient, controlid=ptmatch, eventOnly=TRUE, variant=variantPanTNBC,
cna=cnaPanTNBC, gcount=gcPanTNBC, tumorpct=0.3, tailEnd="upper", tailPct=0.1)
drivGenes[1:10, 1:6]

```

---

variantPanTNBC

*Dataset with somatic and germline DNA variants for TCGA patients*


---

**Description**

Dataset with somatic and germline DNA variants for TCGA patients patients

**Usage**

```
data("variantPanTNBC")
```

**Format**

A data frame with the following columns:

Gene.Symbol Gene Symbol  
CHROM Chromosome  
POS Start Position  
Stop Stop Position  
Ref Reference Allele  
Alt Alternate Allele  
Effect Effect  
PatientID Patient ID  
V1B.refDP Germline Reference allele depth  
V1B.altDP Germline Alternate allele depth  
V1T.refDP Tumor Reference allele depth  
V1T.altDP Tumor Alternate allele depth  
AD AD  
GT Genotype  
SampleType Sample Type Blood Tissue

**Details**

Non-synonymous somatic variants, and high-impact germline variants.

**Examples**

```
data(variantPanTNBC)  
str(variantPanTNBC)
```

# Index

## \*Topic **datasets**

- clinPanBC, [1](#)
- clinPanCO, [2](#)
- cnaPanCO, [3](#)
- cnaPanTNBC, [3](#)
- dgidbPan, [4](#)
- dgiSets, [4](#)
- gcinfoPan, [5](#)
- gcPanCO, [5](#)
- gcPanTNBC, [6](#)
- genelistPan, [6](#)
- variantPanTNBC, [14](#)

- clinPanBC, [1](#)
- clinPanCO, [2](#)
- cnaPanCO, [3](#)
- cnaPanTNBC, [3](#)

- dgidbPan, [4](#)
- dgiSets, [4](#)
- drivDNA (*panGeneSets*), [12](#)

- gcinfoPan, [5](#)
- gcPanCO, [5](#)
- gcPanTNBC, [6](#)
- genelistPan, [6](#)

- outRNA (*panGeneSets*), [12](#)

- panCircos, [7](#)
- panConnect (*panGeneSets*), [12](#)
- panDrugGraph, [8](#)
- panDrugSets, [9](#)
- panGeneDruggable (*panGeneSets*), [12](#)
- panGeneGraph, [11](#)
- panGeneSets, [12](#)

- variantPanTNBC, [14](#)