

# oligoClasses

November 11, 2009

## R topics documented:

affyPlatforms . . . . .	2
chromosome . . . . .	2
efsExample . . . . .	3
scqsExample . . . . .	3
sfsExample . . . . .	4
sqsExample . . . . .	4
DBPDInfo-class . . . . .	5
db . . . . .	5
exprs-methods . . . . .	6
FeatureSet-class . . . . .	6
genomeBuild . . . . .	7
getPlatformDesign . . . . .	7
mmindex . . . . .	8
kind . . . . .	9
length-methods . . . . .	9
manufacturer-methods . . . . .	9
oligoSnpSet-methods . . . . .	10
geometry . . . . .	10
platform-methods . . . . .	10
pmAllele . . . . .	11
pmFragmentLength-methods . . . . .	11
pmPosition . . . . .	11
pmStrand . . . . .	12
position . . . . .	12
probeNames . . . . .	13
QuantificationSet-class . . . . .	13
sampleNames-methods . . . . .	14
mmSequence . . . . .	14
calls . . . . .	15
SnpCallSetPlus-class . . . . .	15
SnpCnvQSet-class . . . . .	16
cnConfidence . . . . .	17
SnpLevelSet-class . . . . .	17
SnpQSet-class . . . . .	18
getA . . . . .	19
antisenseThetaA . . . . .	20
SnpCallSet-class . . . . .	21

**Index**[22](#)

---

<code>affyPlatforms</code>	<i>Available Affymetrix platforms for SNP arrays</i>
----------------------------	--

---

**Description**

Provides a listing of available Affymetrix platforms currently supported by the R package oligo

**Usage**

```
affyPlatforms ()
```

**Value**

A vector of class character.

**Author(s)**

R. Scharpf

**Examples**

```
affyPlatforms ()
```

---

<code>chromosome</code>	<i>Accessor for chromosome information</i>
-------------------------	--

---

**Description**

Returns chromosome information.

**Usage**

```
pmChr (object)
```

**Arguments**

`object` `TilingFeatureSet` or `SnpcallSet` object

**Details**

`chromosome ()` returns the chromosomal information for all probes and `pmChr ()` subsets the output to the PM probes only (if a `TilingFeatureSet` object).

**Value**

Vector with chromosome information.

---

`efsExample`*ExpressionFeatureSet Object*

---

**Description**

Example of ExpressionFeatureSet Object.

**Usage**

```
data(efsExample)
```

**Format**

Object belongs to ExpressionFeatureSet class.

**Examples**

```
data(efsExample)
class(efsExample)
```

---

`scqsExample`*SnpCnvQSet Example*

---

**Description**

Example of SnpCnvQSet object.

**Usage**

```
data(scqsExample)
```

**Format**

Object belongs to SnpCnvQSet class.

**Examples**

```
data(scqsExample)
class(scqsExample)
```

---

`sfsExample`*SnpFeatureSet Example*

---

**Description**

Example of SnpFeatureSet object.

**Usage**

```
data(sfsExample)
```

**Format**

Object belongs to SnpFeatureSet class

**Examples**

```
data(sfsExample)
class(sfsExample)
```

---

`sqsExample`*SnpQSet Example*

---

**Description**

Example of SnpQSet instance.

**Usage**

```
data(sqsExample)
```

**Format**

Belongs to SnpQSet class.

**Examples**

```
data(sqsExample)
class(sqsExample)
```

---

DBPDInfo-class      *Class "DBPDInfo"*

---

### Description

A class for Platform Design Information objects, stored using a database approach

### Objects from the Class

Objects can be created by calls of the form `new("DBPDInfo", ...)`.

### Slots

**getDb:** Object of class "function"

**tableInfo:** Object of class "data.frame"

**manufacturer:** Object of class "character"

**genomebuild:** Object of class "character"

**geometry:** Object of class "integer" with length 2 (rows x columns)

### Methods

No methods defined with class "DBPDInfo" in the signature.

---

db      *Get the connection to the SQLite Database*

---

### Description

This function will return the SQLite connection to the database associated to objects used in oligo.

### Usage

```
db(object)
```

### Arguments

object      Object of valid class. See methods.

### Value

SQLite connection.

### Methods

**object = "FeatureSet"** object of class FeatureSet

**object = "SnpCallSet"** object of class SnpCallSet

**object = "DBPDInfo"** object of class DBPDInfo

**object = "SnpLevelSet"** object of class SnpLevelSet

**Author(s)**

Benilton Carvalho

**Examples**

```
## db(object)
```

---

```
exprs-methods      Accessor for the 'exprs' slot
```

---

**Description**

Accessor for the 'exprs'/'se.exprs' slot of FeatureSet-like objects

**Methods**

**object = "ExpressionSet"** Expression matrix for objects of this class. Usually results of preprocessing algorithms, like RMA.

**object = "FeatureSet"** General container 'exprs' inherited from eSet

**object = "SnpSet"** General container 'exprs' inherited from eSet, not yet used.

---

```
FeatureSet-class   "FeatureSet" and "FeatureSet" Extensions
```

---

**Description**

Classes to store data from Expression/Exon/SNP/Tiling arrays at the feature level.

**Objects from the Class**

The FeatureSet class is VIRTUAL. Therefore users are not able to create instances of such class.

Objects for FeatureSet-like classes can be created by calls of the form: `new(CLASSNAME, assayData, manufacturer, platform, exprs, phenoData, featureData, experimentData, annotation, ...)`. But the preferred way is using parsers like [read.celfiles](#) and [read.xysfiles](#).

**Slots**

**manufacturer:** Object of class "character"

**assayData:** Object of class "AssayData"

**phenoData:** Object of class "AnnotatedDataFrame"

**featureData:** Object of class "AnnotatedDataFrame"

**experimentData:** Object of class "MIAME"

**annotation:** Object of class "character"

**.\_\_classVersion\_\_:** Object of class "Versions"

**Methods**

**show** signature(.Object = "FeatureSet"): show object contents

**Author(s)**

Benilton Carvalho

**See Also**

[eSet](#), [VersionedBiobase](#), [Versioned](#)

**Examples**

```
set.seed(1)
tmp <- 2^matrix(rnorm(100), ncol=4)
rownames(tmp) <- 1:25
colnames(tmp) <- paste("sample", 1:4, sep="")
efs <- new("ExpressionFeatureSet", exprs=tmp)
```

---

genomeBuild

*Genome Build Information*

---

**Description**

Returns the genome build information. This information comes from the annotation package and is given as an argument during the package creation process.

**Usage**

```
genomeBuild(object)
```

**Arguments**

object            PDInfo or FeatureSet object.

---

getPlatformDesign

*Retrieve Platform Design object*

---

**Description**

Retrieve platform design object.

**Usage**

```
getPlatformDesign(object)
getPD(object)
```

**Arguments**

object            FeatureSet object

**Details**

Retrieve platform design object.

**Value**

platformDesign or PDInfo object.

---

mmindex

*Accessors for PM, MM or background probes indices.*

---

**Description**

Extracts the indexes for PM, MM or background probes.

**Usage**

```
mmindex(object, ...)  
pmindex(object, ...)  
bgindex(object, ...)
```

**Arguments**

object	FeatureSet or DBPDInfo object
...	Extra arguments, not yet implemented

**Details**

The indices are ordered by 'fid', i.e. they follow the order that the probes appear in the CEL/XYS files.

**Value**

A vector of integers representing the rows of the intensity matrix that correspond to PM, MM or background probes.

**Examples**

```
## How pm() works  
## Not run:  
x <- read.celfiles(list.celfiles())  
pms0 <- pm(x)  
pmi <- pmindex(x)  
pms1 <- exprs(x)[pmi,]  
identical(pms0, pms1)  
## End(Not run)
```



---

kind	<i>Array type</i>
------	-------------------

---

**Description**

Retrieves the array type.

**Usage**

```
kind(object)
```

**Arguments**

object            FeatureSet or DBPDInfo object

**Value**

String: "Expression", "Exon", "SNP" or "Tiling"

**Examples**

```
if (require(pd.mapping50k.xba240)) {
  data(sfsExample)
  annotation(sfsExample) <- "pd.mapping50k.xba240"
  kind(sfsExample)
}
```

---

length-methods	<i>Number of samples for FeatureSet-like objects.</i>
----------------	---

---

**Description**

Number of samples for FeatureSet-like objects.

**Methods**

**x = "FeatureSet"** Number of samples

---

manufacturer-methods	<i>Manufacturer ID for FeatureSet-like objects.</i>
----------------------	---

---

**Description**

Manufacturer ID for FeatureSet-like and DBPDInfo-like objects.

**Methods**

**object = "FeatureSet"** Manufacturer ID

**object = "PDInfo"** Manufacturer ID

---

oligoSnpSet-methods

*Methods for oligoSnpSet class*

---

### **Description**

Methods for oligoSnpSet

---

geometry

*Array Geometry Information*

---

### **Description**

For a given array, `geometry` returns the physical geometry of it.

### **Usage**

```
geometry(object)
```

### **Arguments**

object      PDInfo object

### **Examples**

```
if (require(pd.mapping50k.xba240))  
  geometry(pd.mapping50k.xba240)
```

---

platform-methods

*Platform Information*

---

### **Description**

Platform Information

### **Methods**

**object = "FeatureSet"** platform information

---

pmAllele *Access the allele information for PM probes.*

---

### Description

Accessor to the allelic information for PM probes.

### Usage

pmAllele(object)

### Arguments

object SnpFeatureSet or PDInfo object.

---

pmFragmentLength-methods

*Information on Fragment Length*

---

### Description

This method will return the fragment length for PM probes.

### Methods

**object = "AffySNPPDInfo"** On `AffySNPPDInfo` objects, it will return the fragment length that contains the SNP in question.

---

pmPosition

*Accessor to position information*

---

### Description

pmPosition will return the genomic position for the (PM) probes.

### Usage

pmPosition(object)  
pmOffset(object)

### Arguments

object `AffySNPPDInfo`, `TilingFeatureSet` or `SnpCallSet` object

### Details

pmPosition will return genomic position for PM probes on a tiling array.

pmOffset will return the offset information for PM probes on SNP arrays.

---

pmStrand	<i>Accessor to the strand information</i>
----------	---

---

**Description**

Returns the strand information on SNP arrays for PM probes (0 - sense / 1 - antisense).

**Usage**

```
pmStrand(object)
```

**Arguments**

object	AffySNPPDInfo object
--------	----------------------

---

position	<i>Accessor to position information</i>
----------	---

---

**Description**

position will return the genomic position of a SNP.

**Usage**

```
position(object)
```

**Arguments**

object	object inheriting from SnpLevelSet
--------	------------------------------------

**Details**

position will return genomic position of a SNP (number of basepairs from the 5-prime chromosomal end)

**Value**

an integer

**Author(s)**

R. Scharpf

---

probeNames	<i>Accessor to feature names</i>
------------	----------------------------------

---

**Description**

Accessor to PM feature names.

**Usage**

```
probeNames(object, subset = NULL)
```

**Arguments**

object	FeatureSet or DBPDInfo
subset	not implemented yet.

---

QuantificationSet-class	<i>Class "QuantificationSet"</i>
-------------------------	----------------------------------

---

**Description**

A virtual class to store summarized measures.

**Objects from the Class**

A virtual Class: No objects may be created from it.

**Slots**

**assayData:** Object of class "AssayData"  
**phenoData:** Object of class "AnnotatedDataFrame"  
**featureData:** Object of class "AnnotatedDataFrame"  
**experimentData:** Object of class "MIAME"  
**annotation:** Object of class "character"  
**.\_\_classVersion\_\_:** Object of class "Versions"

**Extends**

Class [eSet](#), directly. Class [VersionedBiobase](#), by class "eSet", distance 2. Class [Versioned](#), by class "eSet", distance 3.

**Methods**

No methods defined with class "QuantificationSet" in the signature.

**See Also**

[SnpQSet](#), [SnpCnvQSet](#)

**Examples**

```
showClass("QuantificationSet")
```

---

```
sampleNames-methods
```

*Sample names for FeatureSet-like objects*

---

**Description**

Returns sample names for FeatureSet-like objects.

**Methods**

**object = "FeatureSet"** Sample names

---

```
mmSequence
```

*Probe Sequences*

---

**Description**

Accessor to the (PM/MM/background) probe sequences.

**Usage**

```
mmSequence(object)
pmSequence(object, ...)
bgSequence(object, ...)
```

**Arguments**

```
object      FeatureSet, AffySNPPDInfo or DBPDInfo object
...         additional arguments
```

**Value**

A DNASTringSet containing the PM/MM/background probe sequence associated to the array.

---

calls *Accessors for SnpCallSet*

---

### Description

Accessors and replacement methods for genotype calls and associated confidences.

### Usage

```
calls(object)
callsConfidence(object)
calls(object) <-value
callsConfidence(object) <-value
```

### Arguments

object	SnpCallSet object.
value	matrix

### Value

Matrix with R rows (SNP's) and C columns (Samples) with genotype calls (1 - AA; 2 - AB; 3 - BB) or confidences.

### See Also

[crlmm](#)

---

SnpCallSetPlus-class  
*Class "SnpCallSetPlus"*

---

### Description

Description for Genotyping

### Objects from the Class

Objects of this class are unions of Quantification and Calls.

### Slots

**assayData:** Object of class "AssayData"  
**phenoData:** Object of class "AnnotatedDataFrame"  
**featureData:** Object of class "AnnotatedDataFrame"  
**experimentData:** Object of class "MIAME"  
**annotation:** Object of class "character"  
**.\_\_classVersion\_\_:** Object of class "Versions"

**Extends**

Class "SnpQSet", directly. Class "SnpCnvQSet", directly. Class "SnpCallSet", directly. Class "QuantificationSet", by class "SnpQSet", distance 2. Class eSet, by class "SnpQSet", distance 3. Class VersionedBiobase, by class "SnpQSet", distance 4. Class Versioned, by class "SnpQSet", distance 5. Class "QuantificationSet", by class "SnpCnvQSet", distance 2. Class eSet, by class "SnpCnvQSet", distance 3. Class VersionedBiobase, by class "SnpCnvQSet", distance 4. Class Versioned, by class "SnpCnvQSet", distance 5. Class "SnpLevelSet", by class "SnpCallSet", distance 2. Class eSet, by class "SnpCallSet", distance 3. Class VersionedBiobase, by class "SnpCallSet", distance 4. Class Versioned, by class "SnpCallSet", distance 5.

**Methods**

```
db signature(object="SnpCallSetPlus")
```

**Examples**

```
showClass("SnpCallSetPlus")
```

---

SnpCnvQSet-class    *Class "SnpCnvQSet"*

---

**Description**

A class to store summarized measures from SnpCnvFeatureSet objects

**Objects from the Class**

Objects can be created by calls of the form `new("SnpCnvQSet", assayData, phenoData, featureData, experimentData, annotation, ...)`.

**Slots**

**assayData:** Object of class "AssayData"  
**phenoData:** Object of class "AnnotatedDataFrame"  
**featureData:** Object of class "AnnotatedDataFrame"  
**experimentData:** Object of class "MIAME"  
**annotation:** Object of class "character"  
**.\_\_classVersion\_\_:** Object of class "Versions"

**Extends**

Class QuantificationSet, directly. Class eSet, by class "QuantificationSet", distance 2. Class VersionedBiobase, by class "QuantificationSet", distance 3. Class Versioned, by class "QuantificationSet", distance 4.

**Methods**

No methods defined with class "SnpCnvQSet" in the signature.



**Examples**

```
showClass("SnpCnvQSet")
```

---

cnConfidence	<i>Accessors for SnpCopyNumberSet objects</i>
--------------	---

---

**Description**

Accessors and replacement methods for SnpCopyNumberSet objects.

**Usage**

```
copyNumber(object)
cnConfidence(object)
copyNumber(object) <-value
cnConfidence(object) <-value
```

**Arguments**

object	SnpCopyNumberSet object
value	matrix

**Value**

Matrix with R rows (SNP's) and C columns (Samples) containing copy number estimates and confidences.

---

SnpLevelSet-class *Class "SnpLevelSet"*

---

**Description**

Virtual class for SNP-level data

**Objects from the Class**

A virtual Class: No objects may be created from it.

**Slots**

**assayData:** The usual restrictions on assayData: contains matrices with equal dimensions, and with column number equal to nrow(phenoData). Class:[class.AssayData](#)

**phenoData:** See [class.eSet](#)

**featureData:** See [class.eSet](#)

**experimentData:** See [class.eSet](#)

**annotation:** See [class.eSet](#)

**.\_\_classVersion\_\_:** Object of class "Versions"

**Extends**

Class `eSet`, directly. Class `VersionedBiobase`, by class "eSet", distance 2. Class `Versioned`, by class "eSet", distance 3.

**Methods**

**calls** signature(object = "SnpLevelSet"): Accessor for matrix of genotype calls

**calls<-** signature(object = "SnpLevelSet", value = "matrix"): Replacement method for genotype calls.

**callsConfidence** signature(object = "SnpLevelSet"): Accessor for matrix of confidence scores for the genotype calls

**callsConfidence<-** signature(object = "SnpLevelSet", value = "matrix"): Replacement method

**chromosome** signature(object = "SnpLevelSet"): See also [chromosome](#)

**chromosome<-** signature(object = "SnpLevelSet", value = "character"): See also [chromosome](#)

**cnConfidence** signature(object = "SnpLevelSet"): Accessor for matrix of confidence scores for the copy number estimates.

**cnConfidence<-** signature(object = "SnpLevelSet", value = "matrix"): Replacement method for confidence scores of copy number estimates.

**combine** signature(x = "SnpLevelSet", y = "SnpLevelSet"): Useful for combining two objects with the same phenoData. For instance, combine an object created from the Hind 50k chip and an object created from the Xba 50k chip.

**copyNumber** signature(object = "SnpLevelSet"): Accessor for matrix of copy numbers.

**copyNumber<-** signature(object = "SnpLevelSet", value = "matrix"): Replacement method for copy number matrices

**db** signature(object = "SnpLevelSet"): See also [db](#)

**position** signature(object = "SnpLevelSet"): See also [position](#)

**Author(s)**

RS

**Examples**

```
showClass("SnpLevelSet")
```

---

SnpQSet-class

*Class "SnpQSet"*

---

**Description**

Class to store Snp (Quantifications) Summaries

### Objects from the Class

Objects can be created by calls of the form `new("SnpQSet", assayData, senseThetaA, senseThetaB, antisenseThetaA, antisenseThetaB, phenoData, featureData, experimentData, annotation)`.

### Slots

**assayData:** Object of class "AssayData"  
**phenoData:** Object of class "AnnotatedDataFrame"  
**featureData:** Object of class "AnnotatedDataFrame"  
**experimentData:** Object of class "MIAME"  
**annotation:** Object of class "character"  
**.\_\_\_classVersion\_\_:** Object of class "Versions"

### Extends

Class `eSet`, directly. Class `VersionedBiobase`, by class "eSet", distance 2. Class `Versioned`, by class "eSet", distance 3.

### Methods

No methods defined with class "SnpQSet" in the signature.

### See Also

[eSet](#), [VersionedBiobase](#), [Versioned](#)

---

getA

*Compute average log-intensities / log-ratios*

---

### Description

Methods to compute average log-intensities and log-ratios across alleles, within strand.

### Usage

```
getA(object)
getM(object)
```

### Arguments

object            SnpQSet, SnpCnvQSet or TilingFeatureSet2 object.

**Details**

For SNP data, SNPRMA summarizes the SNP information into 4 quantities (log<sub>2</sub>-scale):

antisenseThetaA antisense allele A

antisenseThetaB antisense allele B

senseThetaA sense allele A

senseThetaB sense allele B

The average log-intensities are given by:  $(\text{antisenseThetaA} + \text{antisenseThetaB}) / 2$  and  $(\text{senseThetaA} + \text{senseThetaB}) / 2$ .

The average log-ratios are given by:  $\text{antisenseThetaA} - \text{antisenseThetaB}$  and  $\text{senseThetaA} - \text{senseThetaB}$ .

For Tiling data, `getM` and `getA` return the log-ratio and average log-intensities computed across channels:  $M = \log_2(\text{channel1}) - \log_2(\text{channel2})$   $A = (\log_2(\text{channel1}) + \log_2(\text{channel2})) / 2$

**Value**

A 3-dimensional array (SNP's x Samples x Strand) with the requested measure, when the input SNP data (50K, 250K).

A 2-dimensional array (SNP's x Samples), when the input is from SNP 5.0 and SNP 6.0 arrays.

A 2-dimensional array if the input is from Tiling arrays.

**See Also**

[snprma](#)

---

antisenseThetaA      *Summarized Measures for SNP*

---

**Description**

SNPRMA summarizes the intensities for each SNP into 4 quantities. These measures are at the log<sub>2</sub> scale and are used to create log-ratios and log-averages, which are used by the genotype algorithm and copy number tool.

**Usage**

```
antisenseThetaA(object)
antisenseThetaB(object)
senseThetaA(object)
senseThetaB(object)
thetaA(object)
thetaB(object)
```

**Arguments**

object                      SnpQSet object

**Details**

For a given SNP, an Affymetrix SNP array usually contains multiple features for each combination allele x strand, on which the summaries are computed.

**Value**

A matrix with R rows (SNP's) x C columns (Samples).

**See Also**

[snprma](#), [crlmm](#)

---

SnpCallSet-class     *"SnpSet" Extensions*

---

**Description**

Class to store genotype calls / copy number estimates (and confidences) obtained via SNP arrays.

**Objects from the Class**

Objects can be created by calls of the form `new("SnpCallSet", assayData, featureData, phenoData, experimentData, annotation, calls, callsConfidence, ...)`.

**Slots**

**assayData:** Object of class "AssayData"  
**phenoData:** Object of class "AnnotatedDataFrame"  
**featureData:** Object of class "AnnotatedDataFrame"  
**experimentData:** Object of class "MIAME"  
**annotation:** Object of class "character"  
**.\_\_classVersion\_\_:** Object of class "Versions"

**Extends**

Class `eSet`, directly. Class `VersionedBiobase`, by class "eSet", distance 2. Class `Versioned`, by class "eSet", distance 3.

**Methods**

No methods defined with class "SnpCallSet" in the signature.

**See Also**

[eSet](#), [VersionedBiobase](#), [Versioned](#)

# Index

## \*Topic **classes**

- DBPDInfo-class, 4
- FeatureSet-class, 5
- QuantificationSet-class, 12
- SnpcallSet-class, 20
- SnpcallSetPlus-class, 14
- SnpcnvQSet-class, 15
- SnplevelSet-class, 16
- SnqSet-class, 17

## \*Topic **datasets**

- efsExample, 2
- scqsExample, 2
- sfsExample, 3
- sqsexample, 3

## \*Topic **list**

- affyPlatforms, 1

## \*Topic **manip**

- antisenseThetaA, 19
- calls, 14
- chromosome, 1
- cnConfidence, 16
- genomeBuild, 6
- geometry, 9
- getA, 18
- getPlatformDesign, 6
- kind, 8
- mmindex, 7
- mmSequence, 13
- pmAllele, 10
- pmPosition, 10
- pmStrand, 11
- position, 11
- probeNames, 12

## \*Topic **methods**

- db, 4
- exprs-methods, 5
- length-methods, 8
- manufacturer-methods, 8
- oligoSnpSet-methods, 9
- platform-methods, 9
- pmFragmentLength-methods, 10
- sampleNames-methods, 13
- SnpcallSet-class, 20

## \*Topic **misc**

- affyPlatforms, 1
- AffyExonPDInfo-class  
(DBPDInfo-class), 4
- AffyExpressionPDInfo-class  
(DBPDInfo-class), 4
- AffyGenePDInfo-class  
(DBPDInfo-class), 4
- affyPlatforms, 1
- AffySNPCNVPDInfo-class  
(DBPDInfo-class), 4
- AffySNPPDInfo-class  
(DBPDInfo-class), 4
- AffySTPDInfo-class  
(DBPDInfo-class), 4
- AffyTilingPDInfo-class  
(DBPDInfo-class), 4
- antisenseThetaA, 19
- antisenseThetaA, SnqSet-method  
(antisenseThetaA), 19
- antisenseThetaB  
(antisenseThetaA), 19
- antisenseThetaB, SnqSet-method  
(antisenseThetaA), 19
- bgindex (mmindex), 7
- bgindex, DBPDInfo-method  
(mmindex), 7
- bgindex, FeatureSet-method  
(mmindex), 7
- bgSequence (mmSequence), 13
- bgSequence, DBPDInfo-method  
(mmSequence), 13
- bgSequence, FeatureSet-method  
(mmSequence), 13
- calls, 14
- calls, oligoSnpSet-method  
(oligoSnpSet-methods), 9
- calls, SnpcallSet-method (calls),  
14
- calls, SnplevelSet-method  
(SnplevelSet-class), 16

- calls<- (*calls*), 14
- calls<-, oligoSnpSet, matrix-method (*oligoSnpSet-methods*), 9
- calls<-, SnpCallSet, matrix-method (*calls*), 14
- calls<-, SnpLevelSet, matrix-method (*SnpLevelSet-class*), 16
- callsConfidence (*calls*), 14
- callsConfidence, oligoSnpSet-method (*oligoSnpSet-methods*), 9
- callsConfidence, SnpCallSet-method (*calls*), 14
- callsConfidence, SnpLevelSet-method (*SnpLevelSet-class*), 16
- callsConfidence<- (*calls*), 14
- callsConfidence<-, oligoSnpSet, matrix-method (*oligoSnpSet-methods*), 9
- callsConfidence<-, SnpCallSet, matrix-method (*calls*), 14
- callsConfidence<-, SnpLevelSet, matrix-method (*SnpLevelSet-class*), 16
- chromosome, 1, 17
- chromosome, SnpLevelSet-method (*SnpLevelSet-class*), 16
- chromosome<- (*chromosome*), 1
- chromosome<-, AnnotatedDataFrame, character-method (*chromosome*), 1
- chromosome<-, SnpLevelSet, character-method (*SnpLevelSet-class*), 16
- class.AssayData, 16
- class.eSet, 16
- cnConfidence, 16
- cnConfidence, oligoSnpSet-method (*oligoSnpSet-methods*), 9
- cnConfidence, SnpCopyNumberSet-method (*cnConfidence*), 16
- cnConfidence, SnpLevelSet-method (*SnpLevelSet-class*), 16
- cnConfidence<- (*cnConfidence*), 16
- cnConfidence<-, oligoSnpSet, matrix-method (*oligoSnpSet-methods*), 9
- cnConfidence<-, SnpCopyNumberSet, matrix-method (*cnConfidence*), 16
- cnConfidence<-, SnpLevelSet, matrix-method (*SnpLevelSet-class*), 16
- coerce, SnpCallSetPlus, oligoSnpSet-method (*SnpCallSetPlus-class*), 14
- combine, SnpLevelSet, SnpLevelSet-method (*SnpLevelSet-class*), 16
- copyNumber (*cnConfidence*), 16
- copyNumber, oligoSnpSet-method (*oligoSnpSet-methods*), 9
- copyNumber, SnpCopyNumberSet-method (*cnConfidence*), 16
- copyNumber, SnpLevelSet-method (*SnpLevelSet-class*), 16
- copyNumber<- (*cnConfidence*), 16
- copyNumber<-, oligoSnpSet, matrix-method (*oligoSnpSet-methods*), 9
- copyNumber<-, SnpCopyNumberSet, matrix-method (*cnConfidence*), 16
- copyNumber<-, SnpLevelSet, matrix-method (*SnpLevelSet-class*), 16
- crlmm, 14, 20
- db, 4, 17
- db, DBPDInfo-method (*db*), 4
- db, FeatureSet-method (*db*), 4
- db, SnpCallSet-method (*db*), 4
- db, SnpCallSetPlus-method (*SnpCallSetPlus-class*), 14
- db, SnpCnvQSet-method (*db*), 4
- db, SnpLevelSet-method (*SnpLevelSet-class*), 16
- db, SnpQSet-method (*db*), 4
- db-methods (*db*), 4
- DBPDInfo-class, 4
- efsExample, 2
- eSet, 6, 12, 15, 17, 18, 20
- ExonFeatureSet-class (*FeatureSet-class*), 5
- ExpressionFeatureSet-class (*FeatureSet-class*), 5
- ExpressionPDInfo-class (*DBPDInfo-class*), 4
- exprs, FeatureSet-method (*exprs-methods*), 5
- exprs-methods, 5
- FeatureSet-class, 5
- GenFeatureSet-class (*FeatureSet-class*), 5
- genomeBuild, 6
- genomeBuild, DBPDInfo-method (*genomeBuild*), 6
- genomeBuild, FeatureSet-method (*genomeBuild*), 6
- geometry, 9
- geometry, DBPDInfo-method (*geometry*), 9
- getA, 18
- getA, SnpCnvQSet-method (*getA*), 18
- getA, SnpQSet-method (*getA*), 18

- getA, TilingFeatureSet2-method  
(*getA*), 18
- getM(*getA*), 18
- getM, SnpCnvQSet-method (*getA*), 18
- getM, SnpQSet-method (*getA*), 18
- getM, TilingFeatureSet2-method  
(*getA*), 18
- getPD (*getPlatformDesign*), 6
- getPlatformDesign, 6
- getPlatformDesign, FeatureSet-method  
(*getPlatformDesign*), 6
  
- initialize, DBPDInfo-method  
(*DBPDInfo-class*), 4
- initialize, FeatureSet-method  
(*FeatureSet-class*), 5
- initialize, oligoSnpSet-method  
(*oligoSnpSet-methods*), 9
- initialize, SnpCallSet-method  
(*SnpCallSet-class*), 20
- initialize, SnpCallSetPlus-method  
(*SnpCallSetPlus-class*), 14
- initialize, SnpCnvCallSetPlus-method  
(*SnpCallSetPlus-class*), 14
- initialize, SnpCopyNumberSet-method  
(*SnpCallSet-class*), 20
- initialize, SnpQSet-method  
(*SnpQSet-class*), 17
  
- kind, 8
- kind, AffyExonPDInfo-method  
(*kind*), 8
- kind, AffyExpressionPDInfo-method  
(*kind*), 8
- kind, AffyGenePDInfo-method  
(*kind*), 8
- kind, AffySNPCNVPDInfo-method  
(*kind*), 8
- kind, AffySNPPDInfo-method (*kind*),  
8
- kind, ExpressionPDInfo-method  
(*kind*), 8
- kind, FeatureSet-method (*kind*), 8
- kind, TilingPDInfo-method (*kind*), 8
  
- length, FeatureSet-method  
(*length-methods*), 8
- length-methods, 8
  
- manufacturer  
(*manufacturer-methods*), 8
- manufacturer, DBPDInfo-method  
(*manufacturer-methods*), 8
  
- manufacturer, FeatureSet-method  
(*manufacturer-methods*), 8
- manufacturer-methods, 8
- mmindex, 7
- mmindex, DBPDInfo-method  
(*mmindex*), 7
- mmindex, FeatureSet-method  
(*mmindex*), 7
- mmSequence, 13
- mmSequence, DBPDInfo-method  
(*mmSequence*), 13
- mmSequence, FeatureSet-method  
(*mmSequence*), 13
  
- NgsExpressionPDInfo-class  
(*DBPDInfo-class*), 4
- NgsTilingPDInfo-class  
(*DBPDInfo-class*), 4
  
- oligoSnpSet-class  
(*oligoSnpSet-methods*), 9
- oligoSnpSet-methods, 9
  
- platform (*platform-methods*), 9
- platform, FeatureSet-method  
(*platform-methods*), 9
- platform-methods, 9
- pmAllele, 10
- pmAllele, AffySNPPDInfo-method  
(*pmAllele*), 10
- pmAllele, SnpFeatureSet-method  
(*pmAllele*), 10
- pmChr (*chromosome*), 1
- pmChr, FeatureSet-method  
(*chromosome*), 1
- pmFragmentLength  
(*pmFragmentLength-methods*),  
10
- pmFragmentLength, AffySNPPDInfo-method  
(*pmFragmentLength-methods*),  
10
- pmFragmentLength-methods, 10
- pmindex (*mmindex*), 7
- pmindex, DBPDInfo-method  
(*mmindex*), 7
- pmindex, FeatureSet-method  
(*mmindex*), 7
- pmOffset (*pmPosition*), 10
- pmOffset, AffySNPPDInfo-method  
(*pmPosition*), 10
- pmPosition, 10
- pmPosition, ExpressionPDInfo-method  
(*pmPosition*), 10



- pmPosition, TilingFeatureSet-method  
(*pmPosition*), 10
- pmPosition, TilingPDInfo-method  
(*pmPosition*), 10
- pmSequence (*mmSequence*), 13
- pmSequence, AffySNPPDInfo-method  
(*mmSequence*), 13
- pmSequence, DBPDInfo-method  
(*mmSequence*), 13
- pmSequence, FeatureSet-method  
(*mmSequence*), 13
- pmStrand, 11
- pmStrand, AffySNPPDInfo-method  
(*pmStrand*), 11
- position, 11, 17
- position, SnpLevelSet-method  
(*SnpLevelSet-class*), 16
- probeNames, 12
- probeNames, DBPDInfo-method  
(*probeNames*), 12
- probeNames, FeatureSet-method  
(*probeNames*), 12
  
- QuantificationSet, 15
- QuantificationSet-class, 12
  
- read.celfiles, 5
- read.xysfiles, 5
  
- sampleNames, FeatureSet-method  
(*sampleNames-methods*), 13
- sampleNames-methods, 13
- scqsExample, 2
- se.exprs, FeatureSet-method  
(*exprs-methods*), 5
- senseThetaA (*antisenseThetaA*), 19
- senseThetaA, SnpQSet-method  
(*antisenseThetaA*), 19
- senseThetaB (*antisenseThetaA*), 19
- senseThetaB, SnpQSet-method  
(*antisenseThetaA*), 19
- sfsExample, 3
- show, DBPDInfo-method  
(*DBPDInfo-class*), 4
- show, FeatureSet-method  
(*FeatureSet-class*), 5
- SnpCallSet, 15
- SnpCallSet-class, 20
- SnpCallSet-methods  
(*SnpCallSet-class*), 20
- SnpCallSetPlus-class, 14
- SnpCnvCallSetPlus-class  
(*SnpCallSetPlus-class*), 14
  
- SnpCnvFeatureSet-class  
(*FeatureSet-class*), 5
- SNPCNVPDInfo-class  
(*DBPDInfo-class*), 4
- SnpCnvQSet, 12, 15
- SnpCnvQSet-class, 15
- SnpCopyNumberSet-class  
(*SnpCallSet-class*), 20
- SnpCopyNumberSet-methods  
(*SnpCallSet-class*), 20
- SnpFeatureSet-class  
(*FeatureSet-class*), 5
- SnpLevelSet, 15
- SnpLevelSet-class, 16
- SNPPDInfo-class (*DBPDInfo-class*),  
4
- SnpQSet, 12, 15
- SnpQSet-class, 17
- snprma, 19, 20
- sqsexample, 3
  
- thetaA (*antisenseThetaA*), 19
- thetaA, SnpCnvQSet-method  
(*antisenseThetaA*), 19
- thetaB (*antisenseThetaA*), 19
- thetaB, SnpCnvQSet-method  
(*antisenseThetaA*), 19
- TilingFeatureSet-class  
(*FeatureSet-class*), 5
- TilingPDInfo-class  
(*DBPDInfo-class*), 4
  
- Versioned, 6, 12, 15, 17, 18, 20
- VersionedBiobase, 6, 12, 15, 17, 18, 20