

# Package ‘pdInfoBuilder’

September 24, 2012

**Title** Platform Design Information Package Builder

**Description** Builds platform design information packages.

These consist of a SQLite database containing feature-level data such as x, y position on chip and featureSet ID. The database also incorporates featureSet-level annotation data. The products of this packages are used by the oligo pkg.

**Version** 1.20.0

**Author** Seth Falcon, Benilton Carvalho with contributions by Vince Carey, Matt Setles and Kristof de Beuf

**Maintainer** Benilton Carvalho <Benilton.Carvalho@cancer.org.uk>

**LazyLoad** yes

**Depends** R (>= 2.11.0), methods, Biobase (>= 2.5.5), RSQLite (>= 0.7-1), affxparser (>= 1.7.3), oligo (>= 1.9.19)

**Imports** Biostrings (>= 2.13.50), IRanges (>= 1.3.89)

**License** Artistic-2.0

**Collate** AllClasses.R AllGenerics.R initialize-methods.R utils.R  
schema.R initDb.R initDb.snp6.R pmmmBlockToMat.R loaders.R  
loaders.snp6.R makePdInfoPackage-methods.R chipName-methods.R  
getGeometry-methods.R pdBuilderV2TiledRegion.R  
pdBuilderV2ExonTranscription.R pdBuilderV2Gene.R  
pdBuilderV2AffyTiling.R pdBuilderV2NgsExpression.R  
pdBuilderV2AffyExpressionHT.R pdBuilderV2AffySNP.R pdBuilderV2AffySNPCNV.R

**biocViews** Annotation, Infrastructure

## R topics documented:

AffyExpressionPDInfoPkgSeed-class	2
AffySNPCNVPDInfoPkgSeed-class	3
AffySNPCNVPDInfoPkgSeed2-class	4
AffySNPPDInfoPkgSeed-class	5
AffySNPPDInfoPkgSeed2-class	6
AffySTPDInfoPkgSeed-class	7
AffyTilingPDInfoPkgSeed-class	8
chipName	9

getGeometry . . . . .	10
makePdInfoPackage . . . . .	10
NgsExpressionPDInfoPkgSeed-class . . . . .	11
NgsTilingPDInfoPkgSeed-class . . . . .	12
NimbleGenPDInfoPkgSeed-class . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

AffyExpressionPDInfoPkgSeed-class  
*Class "AffyExpressionPDInfoPkgSeed"*

---

## Description

PD Info Package Seed for Affymetrix Expression Arrays

## Objects from the Class

Objects can be created by calls of the form `new("AffyExpressionPDInfoPkgSeed", cdfFile, csvAnnoFile, tabSeqFile)`

## Slots

**cdfFile:** CDF filename  
**celFile:** CEL filename  
**tabSeqFile:** TAB sequence file  
**chipName:** Name of the chip or platform  
**manufacturer:** chip/platform manufacturer  
**url:** chip URL  
**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the biocViews field of the DESCRIPTION file

## Methods

**chipName** chipName  
**getGeometry** initialize  
**makePdInfoPackage** package creator

## Examples

```
showClass("AffyExpressionPDInfoPkgSeed")
```

---

 AffySNPCNVDPInfoPkgSeed-class

*Class "AffySNPCNVDPInfoPkgSeed"*


---

## Description

This class represents Platform Design (PD) packages for Affymetrix genomewide (SNP 5.0 and SNP 6.0) arrays.

## Objects from the Class

Objects can be created by calls of the form `new("AffySNPCNVDPInfoPkgSeed", cdfFile, csvAnnoFile, csvSeqFile)`.

## Slots

`cdfFile`: Path to the CDF file for this.

`csvAnnoFile`: Path to the Affymetrix CSV annotation for the SNP probes.

`csvSeqFile`: Path to the (SNP) probe sequence file.

`csvAnnoFileCnv`: Path to the Affymetrix CSV annotation for the CNV probes.

`csvSeqFileCnv`: Path to the (CNV) probe sequence file.

`splineParamFile`: Path to the spline parameters file used to compute the predicted accuracy of the the genotype calls. Used internally in `.predictAccuracy`.

`cr1mmInfoFile`: Path to is data file containing regions data used by the `cr1mm` function.

`referenceDistFile`: Path to a reference distribution file used in the normalization step. This is the reference used in `snprma`.

`chipName`: Name of the chip or platform

`manufacturer`: chip/platform manufacturer

`url`: chip URL

`genomebuild`: The genome build this platform is based upon.

`organism`: organism for chip.

`species`: species for chip.

`version`: A character vector giving the version number of the package.

`license`: The license of the package

`author`: Author of the package

`email`: An email address to use in the Maintainer field

`biocViews`: Character data for the biocViews field of the DESCRIPTION file

## Methods

**chipName** signature(object = "AffySNPCNVDPInfoPkgSeed"): ...

**getGeometry** signature(object = "AffySNPCNVDPInfoPkgSeed"): ...

**makePdInfoPackage** signature(object = "AffySNPCNVDPInfoPkgSeed"): ...

**Notes**

**\*IMPORTANT\*** Users are strongly advised to download Affymetrix SNP packages from BioConductor.

The files used for slots `splineParamFile`, `cr1mmInfoFile`, and `referenceDistFile` are generated by the Bioconductor project for each chip/platform and are hosted in our svn data repository at [https://hedgehog.fhcrc.org/bioc-data/trunk/annotation/parms\\_store](https://hedgehog.fhcrc.org/bioc-data/trunk/annotation/parms_store). When `makePdInfoPackage` is run, these files are simply copied to the `inst/extdata` directory of the generated package.

**Author(s)**

Benilton Carvalho

**Examples**

```
showClass("AffySNPCNVDPInfoPkgSeed")
```

---

```
AffySNPCNVDPInfoPkgSeed2-class
      Class "AffySNPCNVDPInfoPkgSeed2"
```

---

**Description**

A generic annotation package builder for Affymetrix SNP/CNV arrays. This is a simplified version of the annotation package and `cr1mm` will **\*NOT\*** work for them.

**Objects from the Class**

Objects can be created by calls of the form `new("AffySNPCNVDPInfoPkgSeed2", csvAnnoFileCnv, csvSeqFileCnv,`

**Slots**

`cdfFile`: Path to the CDF file for this.  
`csvAnnoFile`: Path to the Affymetrix CSV annotation for the SNP probes.  
`csvSeqFile`: Path to the (SNP) probe sequence file.  
`csvAnnoFileCnv`: Path to the Affymetrix CSV annotation for the CNV probes.  
`csvSeqFileCnv`: Path to the (CNV) probe sequence file.  
`chipName`: Name of the chip or platform  
`manufacturer`: chip/platform manufacturer  
`url`: chip URL  
`genomebuild`: The genome build this platform is based upon.  
`organism`: organism for chip.  
`species`: species for chip.  
`version`: A character vector giving the version number of the package.  
`license`: The license of the package  
`author`: Author of the package  
`email`: An email address to use in the Maintainer field  
`biocViews`: Character data for the `biocViews` field of the DESCRIPTION file

**Methods**

**chipName** signature(object = "AffySNPCNVPDInfoPkgSeed2"): ...  
**makePdInfoPackage** signature(object = "AffySNPCNVPDInfoPkgSeed2"): ...

**Note**

This is a simplified annotation package. CRLMM won't work for these objects.  
 The user may need to rename the columns or even add column names to the annotation and sequence files. In case problems are found, column names are suggested.

**Author(s)**

Benilton Carvalho

**Examples**

```
showClass("AffySNPCNVPDInfoPkgSeed2")
```

---

```
AffySNPPDInfoPkgSeed-class
      Class "AffySNPPDInfoPkgSeed"
```

---

**Description**

This class represents Platform Design (PD) packages for Affymetrix mapping (SNP chip) arrays.

**Objects from the Class**

Objects can be created by calls of the form `new("AffySNPPDInfoPkgSeed", splineParamFile, crlmmInfoFile, re`

**Slots**

**splineParamFile:** Spline parameters file used to compute the predicted accuracy of the genotype calls.  
**crlmmInfoFile:** Data file containing regions data used by the `crlmm` function.  
**referenceDistFile:** Reference distribution file used in the normalization step by `snprma`.  
**cdfFile:** CDF file for the design.  
**csvAnnoFile:** Affymetrix CSV Annotation file.  
**csvSeqFile:** Affymetrix Probe Sequence file.  
**chipName:** Name of the chip or platform  
**manufacturer:** chip/platform manufacturer  
**url:** chip URL  
**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the `biocViews` field of the DESCRIPTION file

**Methods**

**chipName** signature(object = "AffySNPPDInfoPkgSeed"): ...  
**getGeometry** signature(object = "AffySNPPDInfoPkgSeed"): ...  
**makePdInfoPackage** signature(object = "AffySNPPDInfoPkgSeed"): ...

**Note**

**\*IMPORTANT\*** The user is strongly advised to download Affymetrix SNP packages from BioConductor.

The files used for slots `splineParamFile`, `crLmmInfoFile`, and `referenceDistFile` are generated by the Bioconductor project for each chip/platform and are hosted in our svn data repository at [https://hedgehog.fhcrc.org/bioc-data/trunk/annotation/parms\\_store](https://hedgehog.fhcrc.org/bioc-data/trunk/annotation/parms_store). When `makePdInfoPackage` is run, these files are simply copied to the `inst/extdata` directory of the generated package.

**Examples**

```
showClass("AffySNPPDInfoPkgSeed")

cdfFile <- "Mapping250K_Nsp.cdf"
csvAnno <- "Mapping250K_Nsp_annot.csv"
csvSeq <- "Mapping250K_Nsp_probe_tab"

spline <- "pd.mapping250k.nsp.spline.params.rda"
refd <- "pd.mapping250k.nspRef.rda"
crLmmInf <- "pd.mapping250k.nspCrLmmInfo.rda"

pkg <- new("AffySNPPDInfoPkgSeed",
          version="0.1.5",
          author="A. U. Thor", email="au@thor.net",
          biocViews="AnnotationData",
          genomebuild="NCBI Build 35, May 2004",
          cdfFile=cdfFile, csvAnnoFile=csvAnno, csvSeqFile=csvSeq,
          splineParamFile=spline, crLmmInfoFile=crLmmInf,
          referenceDistFile=refd)

showMethods(classes=class(pkg))
```

---

AffySNPPDInfoPkgSeed2-class

*Class "AffySNPPDInfoPkgSeed2"*

---

**Description**

A generic annotation package builder for Affymetrix SNP arrays. This is a simplified version of the annotation package and `crLmm` will *not* work for them.

**Objects from the Class**

Objects can be created by calls of the form `new("AffySNPPDInfoPkgSeed2", cdfFile, csvAnnoFile, csvSeqFile,`

**Slots**

**axiom:** Logical flag for experimental build of annotation packages for Axiom arrays.  
**cdfFile:** CDF file for the design.  
**csvAnnoFile:** Affymetrix CSV Annotation file.  
**csvSeqFile:** Affymetrix Probe Sequence file.  
**chipName:** Name of the chip or platform  
**manufacturer:** chip/platform manufacturer  
**url:** chip URL  
**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the biocViews field of the DESCRIPTION file

**Methods**

**chipName** signature(object = "AffySNPPDIInfoPkgSeed2"): ...

**Note**

This is a simplified annotation package. CRLMM won't work for these objects.

The user may need to rename the columns or even add column names to the annotation and sequence files. In case problems are found, column names are suggested.

**Examples**

```
showClass("AffySNPPDIInfoPkgSeed2")
```

---

AffySTPDIInfoPkgSeed-class

*Class "AffySTPDIInfoPkgSeed" for the Sense Target gene-level array*

---

**Description**

container for parameters related to pdmapping package construction for ST type arrays

**Objects from the Class**

Objects can be created by calls of the form `new("AffySTPDIInfoPkgSeed", pgfFile, clfFile, probeFile, transF`

**Slots**

**pgfFile:** Object of class "ScalarCharacter" path to pgf  
**clfFile:** Object of class "ScalarCharacter" path to clf  
**probeFile:** Object of class "ScalarCharacter", path to probe sequence file (Optional)  
**transFile:** Object of class "ScalarCharacter", path to trans file (Optional)  
**chipName:** Name of the chip or platform  
**manufacturer:** chip/platform manufacturer  
**url:** chip URL  
**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the biocViews field of the DESCRIPTION file

**Methods**

**chipName** signature(object = "AffySTPDInfoPkgSeed"): ...  
**getGeometry** signature(object = "AffySTPDInfoPkgSeed"): ...  
**makePdInfoPackage** signature(object = "AffySTPDInfoPkgSeed"): ...

**Author(s)**

B. Carvalho

**Examples**

```
showClass("AffySTPDInfoPkgSeed")
```

---

AffyTilingPDInfoPkgSeed-class  
*Class "AffyTilingPDInfoPkgSeed"*

---

**Description**

PD Info Package Seed for Affymetrix Tiling Arrays

**Objects from the Class**

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



**Slots**

**bpmmapFile:** BMAP File - provided by Affymetrix  
**celFile:** CEL File - provided by Affymetrix  
**chipName:** Name of the chip or platform  
**manufacturer:** chip/platform manufacturer  
**url:** chip URL  
**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the biocViews field of the DESCRIPTION file

**Methods**

**makePdInfoPackage** signature(object = "AffyTilingPDInfoPkgSeed"): ...  
**chipName** signature(object = "AffyTilingPDInfoPkgSeed"): ...

**Examples**

```
showClass("AffyTilingPDInfoPkgSeed")
```

---

chipName	<i>Return an Official Chip/Platform Name</i>
----------	--

---

**Description**

This generic function returns an official or standard chip/platform name.

**Usage**

```
chipName(object)
```

**Arguments**

object	See showMethods("chipName"), but generally object will be a subclass of PkgSeed.
--------	--

**Details**

The idea is that the input files can be used to determine a standard name for each platform. For example, the method for AffySNPPDInfoPkgSeed objects reads the header of the CDF file to extract a name.

**Value**

A character vector of length one giving a standard name for the platform.

**Author(s)**

Seth Falcon

---

getGeometry	<i>Return the Chip/Platform geometry</i>
-------------	--

---

**Description**

This generic function returns the geometry for a chip/platform.

**Usage**

```
getGeometry(object)
```

**Arguments**

object            See showMethods("getGeometry"), but generally object will be a subclass of PkgSeed.

**Details**

The idea is that the input files can be used to determine the geometry for each platform. For example, the method for AffySNPPDInfoPkgSeed objects reads the header of the CDF file to extract the geometry.

**Value**

A list with two elements nrows and ncols

**Author(s)**

Matt Settles

---

makePdInfoPackage	<i>Create a Platform Design Info Package</i>
-------------------	--

---

**Description**

This generic function create a platform design info package based on the parameters contained in object which will generally be an instance of a subclass of PkgSeed. The result is a new directory on the filesystem containing the source for the generated pdInfo package.

**Usage**

```
makePdInfoPackage(object, destDir, batch_size = 10000, quiet = FALSE, unlink = FALSE)
```

**Arguments**

object	See showMethods("makePdInfoPackage") to see available methods.
destDir	Path where the resulting pdInfo package source directory will be written.
batch_size	An integer controlling the size of batches processed when reading the flatfiles and loading the DB. In general, larger values of batch_size will use more memory and less time (unless you exceed physical memory, in which case more time will be used as well).
quiet	A logical value. When TRUE, diagnostic and status messages are not printed.
unlink	A logical value. If 'TRUE', and 'destDir' already contains a file or directory with the name 'pkgname', try to unlink (remove) it.

**Details**

In general, creating the SQLite database will be a time and memory intensive task.

**Value**

This function is called for its side-effect of producing a pdInfo source package directory.

**Author(s)**

Seth Falcon

**Examples**

```

cdfFile <- "Mapping250K_Nsp.cdf"
csvAnno <- "Mapping250K_Nsp_annot.csv"
csvSeq <- "Mapping250K_Nsp_probe_tab"

## Not run:
pkg <- new("AffySNPPDInfoPkgSeed",
          version="0.1.5",
          author="A.U. Thor", email="au@thor.net",
          biocViews="AnnotationData",
          genomebuild="NCBI Build 35, May 2004",
          cdfFile=cdfFile, csvAnnoFile=csvAnno, csvSeqFile=csvSeq)

makePdInfoPackage(pkg, destDir=".")

## End(Not run)

```

---

NgsExpressionPDInfoPkgSeed-class

*Class "NgsExpressionPDInfoPkgSeed"*

---

**Description**

PDInfo package Seed for NimbleGen Expression arrays

**Objects from the Class**

Objects can be created by calls of the form new("NgsExpressionPDInfoPkgSeed", ndfFile, pairFile, xysFile,

**Slots**

**ndfFile:** NDF (NimbleGen Design) file  
**xysFile:** XYS File - used as template  
**chipName:** Name of the chip or platform  
**manufacturer:** chip/platform manufacturer  
**url:** chip URL  
**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the biocViews field of the DESCRIPTION file

**Methods**

**makePdInfoPackage** signature(.Object = "NgsExpressionPDInfoPkgSeed"): ...  
**chipName** signature(object = "NimbleGenPDInfoPkgSeed"): ...  
**getGeometry** signature(.Object = "NimbleGenPDInfoPkgSeed"): ...

**Examples**

```
showClass("NgsExpressionPDInfoPkgSeed")
```

---

```

NgsTilingPDInfoPkgSeed-class
      Class "NgsTilingPDInfoPkgSeed"
  
```

---

**Description**

PDInfo package Seed for NimbleGen Tiling arrays

**Objects from the Class**

Objects can be created by calls of the form `new("NgsTilingPDInfoPkgSeed", ndfFile, xysFile, pairFile, posF`

**Slots**

**ndfFile:** NDF (NimbleGen Design) file  
**xysFile:** XYS File - used as template  
**posFile:** POS (Positions) file  
**chipName:** Name of the chip or platform  
**manufacturer:** chip/platform manufacturer  
**url:** chip URL

**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the biocViews field of the DESCRIPTION file

### Methods

**makePdInfoPackage** signature(.Object = "NgsTilingPDInfoPkgSeed"): ...  
**chipName** signature(object = "NimbleGenPDInfoPkgSeed"): ...  
**getGeometry** signature(object = "NimbleGenPDInfoPkgSeed"): ...

### Examples

```
showClass("NgsTilingPDInfoPkgSeed")
```

---

```
NimbleGenPDInfoPkgSeed-class
      Class "NimbleGenPDInfoPkgSeed"
```

---

### Description

PDInfo package Seed for all NimbleGen arrays

### Objects from the Class

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

### Slots

**manufacturer:** Manufacturer = NimbleGen  
**chipName:** Name of the chip or platform  
**url:** chip URL  
**genomebuild:** The genome build this platform is based upon.  
**organism:** organism for chip.  
**species:** species for chip.  
**version:** A character vector giving the version number of the package.  
**license:** The license of the package  
**author:** Author of the package  
**email:** An email address to use in the Maintainer field  
**biocViews:** Character data for the biocViews field of the DESCRIPTION file

**Methods**

**chipName** signature(object = "NimbleGenPDInfoPkgSeed"): ...

**getGeometry** signature(object = "NimbleGenPDInfoPkgSeed"): ...

**Examples**

```
showClass("NimbleGenPDInfoPkgSeed")
```

# Index

## \*Topic **classes**

AffyExpressionPDInfoPkgSeed-class, [2](#)  
AffySNPCNVDPInfoPkgSeed-class, [3](#)  
AffySNPCNVDPInfoPkgSeed2-class, [4](#)  
AffySNPPDInfoPkgSeed-class, [5](#)  
AffySNPPDInfoPkgSeed2-class, [6](#)  
AffySTPDInfoPkgSeed-class, [7](#)  
AffyTilingPDInfoPkgSeed-class, [8](#)  
NgsExpressionPDInfoPkgSeed-class, [11](#)  
NgsTilingPDInfoPkgSeed-class, [12](#)  
NimbleGenPDInfoPkgSeed-class, [13](#)

## \*Topic **methods**

chipName, [9](#)  
getGeometry, [10](#)  
makePdInfoPackage, [10](#)

AffyExonPDInfoPkgSeed-class  
(AffySTPDInfoPkgSeed-class), [7](#)  
AffyExpressionPDInfoPkgSeed-class, [2](#)  
AffyGenePDInfoPkgSeed-class  
(AffySTPDInfoPkgSeed-class), [7](#)  
AffySNPCNVDPInfoPkgSeed-class, [3](#)  
AffySNPCNVDPInfoPkgSeed2-class, [4](#)  
AffySNPPDInfoPkgSeed-class, [5](#)  
AffySNPPDInfoPkgSeed2-class, [6](#)  
AffySTPDInfoPkgSeed-class, [7](#)  
AffyTilingPDInfoPkgSeed-class, [8](#)

chipName, [9](#)  
chipName, AffyExpressionPDInfoPkgSeed-method  
(chipName), [9](#)  
chipName, AffySNPCNVDPInfoPkgSeed-method  
(chipName), [9](#)  
chipName, AffySNPCNVDPInfoPkgSeed2-method  
(chipName), [9](#)  
chipName, AffySNPPDInfoPkgSeed-method  
(chipName), [9](#)  
chipName, AffySNPPDInfoPkgSeed2-method  
(chipName), [9](#)  
chipName, AffySTPDInfoPkgSeed-method  
(chipName), [9](#)

chipName, AffyTilingPDInfoPkgSeed-method  
(chipName), [9](#)

chipName, NimbleGenPDInfoPkgSeed-method  
(chipName), [9](#)

getGeometry, [10](#)

getGeometry, AffyExpressionPDInfoPkgSeed-method  
(getGeometry), [10](#)

getGeometry, AffySNPCNVDPInfoPkgSeed-method  
(getGeometry), [10](#)

getGeometry, AffySNPPDInfoPkgSeed-method  
(getGeometry), [10](#)

getGeometry, AffySTPDInfoPkgSeed-method  
(getGeometry), [10](#)

getGeometry, AffyTilingPDInfoPkgSeed-method  
(getGeometry), [10](#)

getGeometry, NimbleGenPDInfoPkgSeed-method  
(getGeometry), [10](#)

makePdInfoPackage, [10](#)

makePdInfoPackage, AffyExonPDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffyExpressionPDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffyGenePDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffySNPCNVDPInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffySNPCNVDPInfoPkgSeed2-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffySNPPDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffySNPPDInfoPkgSeed2-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffySTPDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, AffyTilingPDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, NgsExpressionPDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

makePdInfoPackage, NgsTilingPDInfoPkgSeed-method  
(makePdInfoPackage), [10](#)

NgsExpressionPDInfoPkgSeed-class, [11](#)

NgstilingPDInfoPkgSeed-class, [12](#)

NimbleGenPDInfoPkgSeed-class, [13](#)