Creating a PharmacoSet object

Petr Smirnov[1], Zhaleh Safikhani[1,2], and Benjamin Haibe-Kains[1,2,3]

[1]Princess Margaret Cancer Centre, University Health Network, Toronto Canada
[2]Department of Medical Biophysics, University of Toronto, Toronto Canada
[3]Department of Computer Science, University of Toronto, Toronto Canada

October 15, 2019

Contents

1 Intro

The PharmacoSet class is structured to contain several different types of genomic data, as well as drug dose response data. It is built upon base R data types and the ExpressionSet class. However, a PharmacoSet object requires the presence of specific annotations for the data to be able to function correctly with the function provided in PharmacoGx. The structure allows users to then interrogate the data on the basis of cells and drugs, removing the need for the user to deal with single experiments. We do not recommend creating PharmacoSet objects by the end users, rather we recommend contacting us to have the objects created for your dataset of interest. However, for completeness and power users the process and structure is documented below.

2 PharamcoSet Structure

The base PharmacoSet structure is as follows:

- @ `annotation`:
  - $ `name`: Acronym of the pharmacogenomic dataset.
  - $ `dateCreated`: When the object was created.
  - $ `sessionInfo`: Software environment used to create the object.
  - $ `call`: Set of parameters used to create the object.
- @ `datasetType`: Either 'sensitivity', 'perturbation', or 'both'
- @ `cell`: data frame annotating all cell lines investigated in the study.
- @ `drug`: data frame annotating all the drugs investigated in the study.
- @ `molecularProfiles`: List of `ExpressionSet` objects containing the molecular profiles of the cell lines, such as mutations, gene expressions, or copy number variations.
- @ `sensitivity`:
  - $ `n`: Number of experiments for each cell line treated with a given drug
  - $ `info`: Metadata for each pharmacological experiment.
  - $ `raw`: All cell viability measurements at each drug concentration from the drug dose-response curves.
  - $ `profiles`: Drug sensitivity values summarizing each dose-response curve (IC_{50}, AUC, etc.)
- @ `perturbation`:
  - $ `n`: Number of experiments for each cell line perturbed by a given drug, for each molecular data type

$ `info`: 'The metadata for the perturbation experiments is available for each molecular type by calling the appropriate info function'

@ `curation`: list of data frames containing the mapping between original drug, cell line, and tissue names to standardized identifiers.

2.1 annotation

The `annotation` slot contains information pertaining to the PharmacoSet as a whole, including the full information on what packages were loaded into R when the PharmacoSet was created, as well as the date and call to the constructor. The only data that must be entered upon creation of the object is the name of the PharmacoSet, which is passed to the constructor.

2.2 datasetType

The `datasetType` slot is a character string which signifies whether the PharmacoSet contains drug dose `sensitivity` data, genomic `perturbation` data, or `both`. This slot is necessary for the subsetting and intersecting function to be aware of what data types they should be looking for.

2.3 cell

This slot contains a `data.frame` which contains information about all the cells profiles across all the data types in the PharamcoSet, including both perturbation and sensitivity experiments. It is crucial for the rownames of each entry in the data frame to be the **unique cell identifier** used across all the datatypes for each cell type. The content of this data frame will vary based on what information each dataset provides. One of the columns in this data frame must be `tissueid`, predictably

2.4 drug

This slot contains a `data.frame` which contains information about all the drugs profiled across all the data types in the PharmacoSet, including both perturbation and sensitivity experiments. Similar to the `cell` slot, it is crucial for the rownames of each entry in the data frame to be the **unique compound identifier** used across all the datatypes for each compound. Once again, the content in this data frame will vary based on information provided by each dataset.

2.5 molecularProfiles

The molecular profiles slot of a PharmacoSet object contains all the molecular expression data profiled in the dataset. Each type of data is kept in a separate ExpressionSet object, and they are all stored as a list. However, to insure coordination between data types, there are specific annotations that must be included in the ExpressionSet object. The annotations also differ slightly in the cases of `perturbation` and `sensitivity` datasets.

First of all, each ExpressionSet object must be labelled with the type of data included within the object. Currently only `mutation`, `fusion`, and `rna` molecular data types are recognized and handled differently by summarizing and signature generating functions. This labelling is done by calling, for example:

```
> Biobase::annotation(eset) <- "rna"
```

The phenoData in each expression set object also needs to contain specific columns labelling each experiment. For sensitivity type datasets, this has to include:

`cellid`: This column contains a cell identifier that matches **exactly** the rownames of the cell slot of the PSet.
`batchid`: This column contains and id of the batch of each experiment. If such data is not available then it is customary to fill with NA, but this column must be present.

Additionally, for the perturbation type datasets, there are several more columns of metadata that must be included to enable the modelling of differential gene expression. These are:

`drugid`: The identifier of the drug used in each experiment when a compound was applied. For controls, it should be left as NA. These must match **exactly** to the rownames of the drug slot in the PharmacoSet.
`duration`: If available, the length of the experiment. Should be 0 for controls.
`concentration`: For experiments where compounds were applied, the concentration. Should be 0 for controls.
`xptype`: A label of either `perturbation`, of `control` respectively.

Otherwise each ExpressionSet object should be constructed as specified by the Biobase package.

Once these ExpressionSet objects are created, they should be added to a list, with the name of each object in the list being descriptive of the type of data contained.

2.6 sensitivity

This is a list containing all the data pertaining to drug dose response experiments, usually only present in `sensitivity` or `both` type datasets. This includes the names the following names slots in the list:

`info`. Metadata for each pharmacological experiment stored in a `data.frame`. Each row of this data.frame should be labelled with a unique experiment identifier. This data.frame must also contain specific columns:

`cellid`: Contains a cell identifier that matches **exactly** the rownames of the cell slot of the PSet.
`drugid`: The identifier of the drug used in each experiment when a compound was applied. These must match **exactly** to the rownames of the drug slot in the PharmacoSet.

`raw`: This is a 3-D `array` of raw drug dose response data. The first dimension is the experiments, with the names of each row in this dimension labelled exactly as the experiment ids used in the metadata data.frame `info` above. The second dimension is the doses, with each c̈olumn l̈abelled as `doses#`, where the number is the column number. This is as long as there are different number of doses for each experiment. The third dimension is always fixed at 2, with names `Dose`, and `Viability`. `Dose` contains the actual dose used (usually in micro-molar), while viability contains the cell viability measurement at that dose. Therefore, for each experiment there are two vectors, one of doses and one of viabilities at each dose.

`profiles` This is a data.frame containing down the rows, with **exactly** matching rownames, the experiments as labelled in the `info` data.frame, and each column being a summary of the drug dose sensitivity, such as `auc_published` or `ic50_published`, as published with the data.

`n` This is a `matrix` containing cellids matching exactly the rownames of the cell slot of the PSet down the rows, and drugids matching exactly the rownames of the drug slot in the columns. This data.frame summaries how

many experiments are in the data for each pair, allowing quick identification of what pairs were tested together. Note that this does not need to be generated before constructing the object, it is generated by the constructor.

2.7 perturbation

This slot is fully filled by the constructor, so nothing needs to be created for it.

`n` This is a 3-D `array` containing cellids matching exactly the rownames of the cell slot of the PSet down the rows, and drugids matching exactly the rownames of the drug slot in the columns, and the third dimension being the names of the different molecular profile types. This array summarizes how many perturbation experiments are in each molecular data type for each pair, allowing quick identification of what pairs were tested together.

`info`: This is always exactly the string referring to each data type separately: 'The metadata for the perturbation experiments is available for each molecular type by calling the appropriate info function'

2.8 curation

This slot contains three data.frames, one for drugs, tissues, and cells. Each contains two columns, the first with the unique identifier that is used between all PharmacoSet objects for this drug, and the second with the identifier used within the dataset. At this time there is no set method to find the unique identifiers between all PharmacoSets for arbitrary drugs, as much labour intensive curation has gone into carefully matching identifiers between datasets. However, these tables are used only by the intersecting function, and all other functions will function without this data, therefore for PharmacoSet objects created by users, we recommend just filling both columns with the identifiers used in the datasets, to allow for at least some matching to be done.

3 Creating the PharmacoSet

Once all the data is in the proper format, it can be passed to a the constructor function as below:

```
> PharmacoSet(name,
+             molecularProfiles=list(),
```

```
+             cell=data.frame(),
+             drug=data.frame(),
+             sensitivityInfo=data.frame(),
+             sensitivityRaw=array(dim=c(0,0,0)),
+             sensitivityProfiles=matrix(),
+             curationDrug=data.frame(),
+             curationCell=data.frame(),
+             curationTissue=data.frame(),
+             datasetType=c("sensitivity", "perturbation", "both"),
+             verify = TRUE)
```

Here, `name` is the name of the PharmacoSet as described in annotations above, molecularProfiles is the list of ExpressionSet objects described above, cell and drug are the data.frames in the cell and drug slots. The sensitivityInfo, sensitivityRaw, sensitivityProfiles are the elements of the sensitivity slot, and curationDrug, curationCell, and curationTissue the contents of the curation slot. Finally, datasetType is a string to signify which type of data is included in the datset, and verify is a flag for running the verification of the pSet after its construction. Any data detailed above but missing from this constructor call will be created by the constructor, and the default empty values are as in the call above. Notice that none of the values should be set as NA, they should be ommited if the data is not being provided.