

Using inSilicoDb 2.0

Quentin De Clerck*

April 15, 2014

1 Functions from inSilicoDb 2.0

This new version of the package provides all the functionality of previous inSilicoDb packages developed by *Jonatan Taminau*. The functions did not change, therefore we refer to the description of previous versions.

1.1 Access to InSilico MySafe

One of the new features of this package is the possibility for users to access their private data stored on InSilico MySafe. This feature is implemented by the functions:

InSilicoLogin(login, password) Logs the user in for the given login and password. There is no secure way to login to a webservice in R. Therefore the password given to the function has to be the md5 hash of the real password. After login in, the user can use the normal functions as described earlier with their private data.

InSilicoLogout() Logs the currently user out of the InSilico DB webservice.

getInSilicoUserDetails() Returns informations (id, name, email) about the user that is currently logged in.

1.2 Check the accessibility of data

Some helper functions were added to provide information about the availability of data: `getDatasetInfo` and `getPlatformList`.

The purpose of `getDatasetInfo` is two-fold. First, it will return the default values of all optional parameters and the title of the study. Second, it will return the availability of the requested dataset for specified parameters. It returns an error if the data is not available for download.

*qdeclerc@vub.ac.be

```

> library("inSilicoDb");
> eset = getDatasetInfo("GSE781", "GPL96");
> print(eset);

$dataset
[1] "GSE781"

$platform
[1] "GPL96"

$format
[1] "CURESET"

$features
[1] "PROBE"

$norm
[1] "ORIGINAL"

$scuration
[1] 4864

$type
[1] "RNA"

$title
[1] "Normal and Renal Cell Carcinoma Kidney Tissue, Human"

$state
[1] "finished"

$path
[1] "GSE781GPL96_RNA_ORIGINALPROBE_4864.RData"

> ## We check the availability of following normalizations for series GSE781 on platform GPL
> norms = c("FRMA", "ORIGINAL")
> output = sapply(norms, function(n) {
+   tryCatch({
+     eset <- getDatasetInfo("GSE781", "GPL97", norm = n);
+     eset$norm;
+   }, error = function(e) {
+     "Unavailable"
+   });
+ });
> print(output);

      FRMA      ORIGINAL

```

```
"Unavailable"      "ORIGINAL"
```

```
>
```

```
> # We can thus conclude that the series GSE781 on platform GPL97 does not support FRMA
```

The `getPlatformList` function returns all the platforms supported by the package. It is also possible to query the availability of the normalizations using this function.

```
> platforms = getPlatformList();  
> print(platforms);
```

```
[1] "GPL96"      "GPL570"      "GPL571"      "GPL3921"     "GPL4685"     "GPL97"  
[7] "GPL1355"     "GPL1261"     "GPL8321"     "GPL91"       "GPL8300"     "GPL85"  
[13] "GPL6947"     "GPL10558"
```

```
> FRMAplatforms = getPlatformList(norm = "FRMA");  
> print(FRMAplatforms);
```

```
[1] "GPL96"      "GPL570"      "GPL3921"     "GPL1261"     "GPL8321"
```

```
>
```

1.3 Curated Data

What changed in the functionality with respect to previous package is the better integration of the curated data. The user can now decide if he wants curated data or not by specifying the format. There are two possible formats: ESET and CURESET. By default the format is CURESET, this means with curated data. It is also possible to specify the curation wanted for the requested expression set.

```
> # without curated data  
> eset = getDataset("GSE4635", "GPL96", format = "ESET");  
> # with curated data  
> cureset = getDataset("GSE4635", "GPL96", format = "CURESET", curation = 9016);  
> print(phenoData(eset));
```

```
An object of class 'AnnotatedDataFrame': none
```

```
> print(phenoData(cureset));
```

```
An object of class 'AnnotatedDataFrame'
```

```
Measurements: GSM104076 GSM104072 ... GSM104075 (8 total)  
varLabels: description title source_name characteristics  
varMetadata: labelDescription
```

It is possible to get information about the default curation (id, time, curator) of a particular dataset by calling the `getDefaultCuration` function. Using the `getCurationInfo` function returns a complete overview of all the possible curations in more details.

```
> default = getDefaultCuration("GSE4635");
> print(default);
```

```
$id
[1] "14926"
```

```
$date
[1] "2011-12-13 15:31:44"
```

```
$curator
[1] "Virginie de Schaetzen"
```

```
> getCurationInfo("GSE4635");
>
```

2 SCAN and UPC normalizations

Two normalizations were added in to the package. It is now possible to request data in Single Channel Array Normalization (SCAN) [1] or Universal exPression Codes (UPC) [2]. SCAN is a possible alternative to fRMA and UPC outputs the probabilities of the probes/genes being expressed.

```
> scan = getDataset("GSE7670", "GPL96", norm = "SCAN", features = "gene");
> # example of values in SCAN normalization
> print(exprs(scan)[1:10, 1:5]);
```

	GSM185811	GSM185812	GSM185813	GSM185814	GSM185815
A1CF	-0.01439515	-0.10936067	-0.06770128	-0.01455117	-0.03543565
A2M	3.68428200	3.32630598	3.38711443	2.91544304	3.41579740
A4GALT	-0.09633088	-0.19550026	0.03164752	-0.01982268	-0.00476726
A4GNT	-0.07375380	-0.20114400	-0.11864749	-0.32669177	-0.22525964
AAAS	0.13653901	0.36010374	0.41218797	0.44505474	0.37398835
AACS	0.60109007	0.54875470	0.67716693	0.68475120	0.53789689
AADAC	0.40490456	-0.00592472	0.08551812	-0.14304748	0.40841383
AAGAB	0.30125869	0.22032590	0.41817506	0.21135862	0.30659907
AAK1	0.98813595	0.70454191	1.18395840	1.05741664	0.82605538
AAMDC	1.02744103	0.51491219	0.85858505	0.36821593	0.67096700

```
> upc = getDataset("GSE7670", "GPL96", norm = "UPC", features = "gene");
> # example of values in UPC normalization
> print(exprs(upc)[1:10, 1:5]);
```

	GSM185811	GSM185812	GSM185813	GSM185814	GSM185815
A1CF	0.1103261	0.1304934	0.1352884	0.1248155	0.1463560
A2M	0.9999999	0.9955987	0.9999984	0.9938093	0.9995542
A4GALT	0.1486228	0.2074570	0.2278594	0.2008301	0.2474129
A4GNT	0.1113161	0.1235891	0.1398627	0.1048917	0.1315567
AAAS	0.1595510	0.2630525	0.3107978	0.2711880	0.2844417
AACS	0.3954489	0.3481779	0.4754562	0.4109214	0.3527393
AADAC	0.3156781	0.1507835	0.1682749	0.1330868	0.3022356
AAGAB	0.2812513	0.2377095	0.3612420	0.2230756	0.3126640
AAK1	0.6221820	0.5102196	0.7267850	0.6358627	0.5533430
AAMDC	0.7073455	0.4501845	0.7045530	0.3448396	0.5920237

3 Session Info

```
> sessionInfo()
```

```
R version 3.1.0 RC (2014-04-02 r65358)
Platform: i386-w64-mingw32/i386 (32-bit)
```

```
locale:
```

```
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252
```

```
attached base packages:
```

```
[1] parallel stats graphics grDevices utils datasets methods
[8] base
```

```
other attached packages:
```

```
[1] limma_3.20.1 inSilicoDb_2.0.1 RCurl_1.95-4.1
[4] bitops_1.0-6 Biobase_2.24.0 BiocGenerics_0.10.0
[7] rjson_0.2.13
```

```
loaded via a namespace (and not attached):
```

```
[1] tools_3.1.0
```

References

- [1] Stephen R Piccolo, Ying Sun, Joshua D Campbell, Marc E Lenburg, Andrea H Bild, and W Evan Johnson. A single-sample microarray normalization method to facilitate personalized-medicine workflows. *Genomics*, 100(6):337–344, 2012.

- [2] Stephen R Piccolo, Michelle R Withers, Owen E Francis, Andrea H Bild, and W Evan Johnson. Multiplatform single-sample estimates of transcriptional activation. *Proceedings of the National Academy of Sciences*, 110(44):17778–17783, 2013.