

pbcmc: Permutation-Based Confidence for Molecular Classification

Cristóbal Fresno
CONICET

Universidad Católica de Córdoba

Germán A González
CONICET

Universidad Católica de Córdoba

Andrea S Llera
CONICET

Fundación Instituto Leloir

Elmer A Fernández
CONICET

Universidad Católica de Córdoba

Abstract

The **pbcmc** package characterizes uncertainty assessment on gene expression classifiers, a. k. a. molecular signatures, based on a permutation test. In order to achieve this goal, synthetic simulated subjects are obtained by permutations of gene labels. Then, each synthetic subject is tested against the corresponding subtype classifier to build the null distribution. Thus, classification confidence measurement reports can be provided for each subject, to assist physician therapy choice. At present, it is only available for PAM50 implementation in **genefu** package but, it can easily be extend to other molecular signatures.

Keywords: PAM50, single subject classifier, clinical outcome, breast cancer subtype.

1. Introduction

Gene expression-based classifiers, known as molecular signatures (MS), are gaining increasing attention in oncology and market. The MS can be defined as a set of coordinately expressed genes and an algorithm that use these data to predict disease subtypes, response to therapy, disease risk and clinical outcome (?). Particularly in breast cancer market there exists many MS such as PAM50 (??), Prosigna® (?), Oncotype DX® (?) and MammaPrint® (?). In essence, most MS try to provide patient subtype classification or risk prediction which has been associated with distant metastasis free survival (DMFS) or relapse free survival (RFS). Consequently, they are intended to be used to support therapy choice. However, several authors have shown that data processing steps, technology, as well as population variability have an effect on measured gene expression and could bias subtype/risk subject assignment (??????). These effects suggest that, from a statistical point of view, MS are not robust for subject classification. In particular, there is no control over type I and II like classification error and subjects could potentially be assigned to a wrong subtype/risk class. Indeed, the lack of certainty in class assignation could lead to a misleading therapy affecting subject outcome. Hence, the development of methods for significance or certainty on MS class assignation is crucial in order to assist physicians' decision making (??).

1.1. PAM50 Molecular Signature

The well-known breast cancer (BC) MS, PAM50 (??), is based on the comparison between the patient gene profiles (PGP) of 50 expressed genes, against five intrinsic genes profiles (IGP) representing: Basal, Her2-enriched, Luminal A, Luminal B and Normal-like subtypes using Spearman’s ρ correlation. Then, the subject will be assigned to the i – th subtype according to (??):

$$\arg \max_{i \in IGP} \rho(PGP, IGP_i) \quad (1)$$

Particularly, patients are assigned to the i – th subtype which maximize $\rho(PGP, IGP_i)$, even if correlation is weak or there are similar/tight IGP correlations. The latter case has been addressed by ?, where they have excluded subjects with a correlation difference between Luminal A and B of less than 0.1, considering them ambiguous pattern, as a way to control a kind of type II error. However, type I error control is still a debt.

1.2. genefu library

At present, the freely available PAM50 **genefu** algorithm implementation (?), offers a kind of subtype probability, $P(IGP_i)$, calculated as (??):

$$P(IGP_i) = \frac{\rho(PGP, IGP_i)}{\sum_i \rho(PGP, IGP_i)} \quad \forall \rho(PGP, IGP_i) > 0 \quad (2)$$

However, this probability does not take part in the classification rule. Even worse, a very low ρ (weak relationship) could reach a high subtype probability, for instance, if all the others ρ ’s are close to zero or are even negative.

1.3. Alternative proposal

In order to overcome these drawbacks, here a simple and reliable single subject classifier to control type I and II errors is proposed. Moreover, it provides a statistical significance on subtype assignation based on a gene label permutation test. Briefly, it evaluates if the observed ρ of each IGP can be achieved by chance, regarding subject observed MS expressed gene values. In addition, we propose a user-friendly subtype assignation panel to support physicians’ decision making, enhancing PAM50 or commercial reports currently available in the market. The method is presented for PAM50 but can easily be extended to other MS algorithms.

2. Methods

The Permutation Based Confidence for Molecular Classification (**pbcmc**) package, estimates the statistical significance of ρ for each IGP. In other words, we want to see whether the observed ρ can be obtained by chance. In order to perform this task, the ρ null distribution for each IGP ($\rho_{H0_{IGP}}$) is obtained by evaluating β permutations of the SGP gene expression. Then, the observed (un-permuted) IGP correlations ($\rho_{u_{IGP}}$) are compared against their own

$\rho_{H0_{IGP}}$ in order to evaluate whether $H_0 : \rho_{u_{IGP}} \in \rho_{H0_{IGP}}$ versus $H_1 : \rho_{u_{IGP}} \notin \rho_{H0_{IGP}}$ according to the p-values (p_{IGP}) calculated as in (??):

$$p_{IGP} = \frac{\sum_i^\beta I(\rho_i, \rho_u)}{\beta}; \quad I(a, b) = \begin{cases} 1 & \text{if } a > b \\ 0 & \text{if } a \leq b \end{cases} \quad (3)$$

where IGP stands for Basal, Her2-Enriched, Luminal A, Luminal B and Normal-like. The resulting five p_{IGP} 's are adjusted to control multiple comparisons using False Discovery Rate, FDR (?). Then, assuming an acceptable type I error, α , the hypothesis test for all IGPs could result in:

- i. No significant ρ_u for any IGP, i. e., all adjusted $p_{IGP} > \alpha$.
- ii. A unique significant ρ_u .
- iii. Multiple significant ρ_u .

For the first case, the subject cannot reliably be assigned to any IGP (not assigned - **NA**). In the second case, it is assigned (**A**) to the trustworthy current PAM50 subtype. In order to overcome the ambiguity of case iii., a correlation difference threshold of 0.1 between the top ones ($\rho_{u_{IGP_1}} > \rho_{u_{IGP_2}} > \dots > 0$) was established, similarly as in ? for Luminal subtypes. Then, if $(\rho_{u_{IGP_1}} - \rho_{u_{IGP_2}}) > 0.1$ the subject is assigned as in ii., otherwise it is considered as an ambiguous (**Amb**) subject.

3. Implementation

The S4 class hierarchy of the **pbmc** package is based on the implementation of an abstract **MolecularPermutationClassifier** class, which can potentially be used for any MS as depicted in Figure ??. Basically, it has been developed as an organized data processing framework for its heirs. The latter are supposed to implement the respective responsibilities. Once a heir object is implemented the user can:

loadBCDataset: Load one of the example breast cancer dataset available at Bioconductor.

At present, it is possible to load `breastCancerXXX` where XXX can be “upp” (?), “nki” (?), “vdx” (?), “mainz” (?), “transbig” (?) or “unt” (?).

filtrate: Remove, from the `exprs` matrix, subjects not required by the classification algorithm.

classify: Generate subject classification according to the heir's implementation (PAM50, etc.).

permute: Obtains subject classification based on the null ρ_{H0} distribution by means of β permutation simulations.

subtype: Obtain the new classification using the permutation results.

subjectReport: Create a friendly report to assist physician treatment decision making.

databaseReport: Create a pdf with all `subjectReports`, if a database is available.

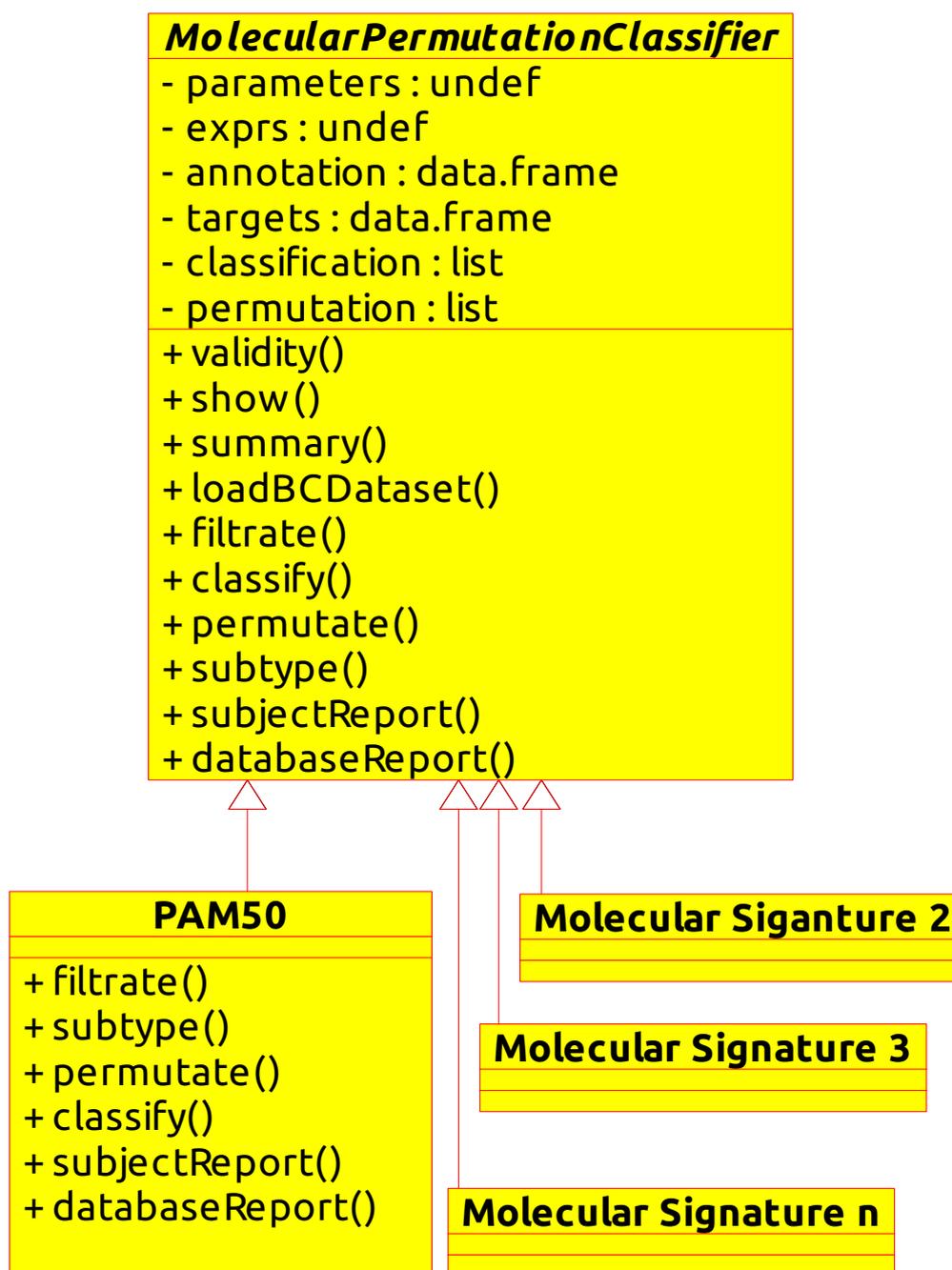


Diagram: class diagram Page 1

Figure 1: Package hierarchy. `MolecularPermutationClassifier` is the main S4 abstract class and `PAM50` implements it for the Molecular Signature (MS) of `??`. The other classes represent user-defined implementations of other MS. Note that complete operation signature have been omitted for simplicity.

At present, the only available heir is PAM50 based on **genefu** library (?). But, it can easily be extended to other MS such as Prosigna® (?) or others, just implementing `filtrate`, `classify`, `permutate`, `subtype`, `subjectReport` and `databaseReport` functions.

3.1. Computational Requirements

The **pbcmc** uses permutation approach as described in section ???. In this context, a parallelization approach by sample has been implemented using **BiocParallel** for the recommended parameters i.e., “`nPerm=10000`”, “`pCutoff=0.01`”, “`where="fdr"`” and “`corCutoff=0.1`”. Hence, it can work in any platform providing the appropriate “`BPPARAM=bpparam()`” option according to the operating system e.g. `SnowParam()` for Windows or `MulticoreParam()` in Linux. It worth to mention that this feature will not be used when working with a single sample.

Computational requirement for memory (Table ??) and time execution (Table ?? and Fig. ??) tested in a Kubuntu 16.04.1 LTS xenial machine, with an Intel(R) Core(TM) i7-4790 CPU 3.60GHz (8 cores), 32 GB DDR3 1866 MHz using the complete **breastCancerNKI** example for recommended parameters were:

Table 1: Memory requirements for **pbcmc** package using **breastCancerNKI**

Process performed	Total object memory size
Raw data (24481×337)	72.9 Mb
Filtered data (57×337)	263 Kb
Classify with <code>std="median"</code> option	324.3 Kb
Permutate with <code>keep=FALSE</code> option	416.3 Kb
Permutate with <code>keep=TRUE</code> option	129.2 Mb
Temporary <code>MulticoreParam(workers=7)</code>	2GB of free memory

It is worth to mention that extra memory requirements for **BiocParallel** will only be used while computing the permutations and released afterwards. For “`nki`” dataset using `MulticoreParam(workers=7)` (Table ??) it requires of 2GB of RAM memory. If additional workers are included, if available, it would require additional free memory.

Table 2: Time execution requirements for **pbcmc** package using **breastCancerNKI**

Type	Time(seconds)			Cores	Speed up
	User	System	Elapsed		
Single sample	2.440	0.072	2.515	1	1.00
	2.440	0.072	2.515	7	
NKI	0.576	0.132	253.011	7	3.35
	0.624	0.152	276.524	6	3.07
	0.780	0.388	285.937	5	2.97
	0.348	0.276	303.965	4	2.79
	0.416	0.204	313.045	3	2.71
	0.488	0.144	449.414	2	1.89
	830.980	17.924	848.911	1	1.00

The time execution (Table ?? and Fig. ??) shows that the algorithm takes advantage of the sample parallel implementation where fore two and three cores almost achieves the theoretical

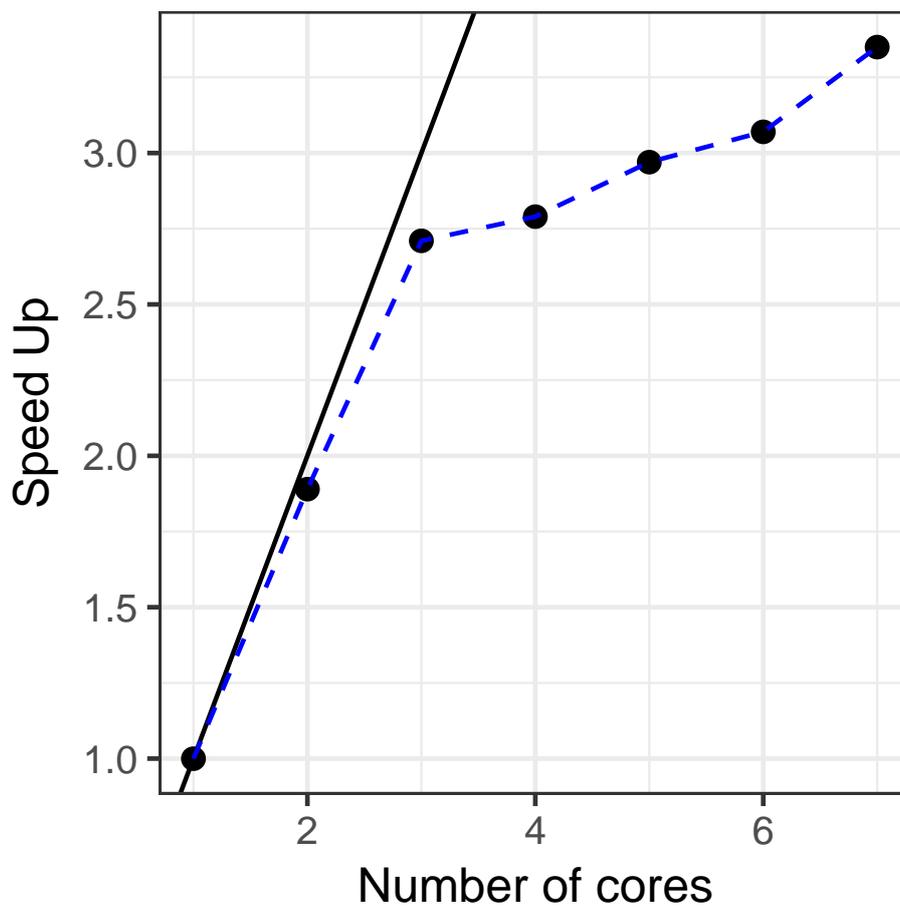


Figure 2: Time execution speed up for **breastCancerNKI** dataset.

identity speed up line. Nevertheless, the extra memory requirement provides a time reduction as depicted in both Table ?? and Fig. ??.

4. Examples

4.1. Using Bioconductor's breastCancerXXX dataset

In order to work with PAM50 MS, the user must load a Bioconductor's breastCancerXXX dataset where XXX stands for UPP, NKI, VDX, TRANSBIG, MAINZ or UNT. For example we can load NKI database (?), provided that the require library is installed, using the following code:

```
R> library("pbcmc")
R> library("BiocParallel")
R> object<-loadBCDataset(Class=PAM50, libname="nki", verbose=TRUE)
R> object
```

A PAM50 molecular permutation classifier object

Dimensions:

	nrow	ncol
exprs	24481	337
annotation	24481	10
targets	337	21

The object is a PAM50 instance, which contains the `exprs` matrix with gene expression values, associated `annotation` and clinical data in `targets` data.frame. On the other hand, the user can use PAM50's constructor to create an object with his/her own data or convert a **limma** MAList object into PAM50 using `as.PAM50(MAList_object)` function. In the first case, the user will only need:

- The **M** gene expression object, i. e., genes in rows and samples in columns.
- The `annotation` data.frame which must include the compulsory fields: "probe", "NCBI.gene.symbol" and "EntrezGene.ID".
- The `targets` data.frame which is an optional slot. If it is provided, it should contain as many rows as samples present in the **M** gene expression object. The idea is to include as many columns as clinical or experimental data these samples have available. For "nki" example, there are 21 columns.

4.2. Using any microarray R data package

The same example of section ?? can be built with **pbcmc** directly loading the package and extracting the data into the **M**, `annotation` and `targets` (optional) slots required by PAM50 object. Just for simplicity we will work with the first five samples but, it also works for a single sample.

```
R> library("breastCancerNKI")
R> data("nki")
```

```
R> ##The expression
R> M<-exprs(nki)[, 1:5, drop=FALSE]
R> head(M)
```

	NKI_4	NKI_6	NKI_7	NKI_8	NKI_9
Contig45645_RC	-0.215	0.071	0.182	-0.343	-0.134
Contig44916_RC	-0.207	0.055	0.077	0.302	0.051
D25272	-0.158	-0.010	0.059	0.169	-0.007
J00129	-0.819	-0.391	-0.624	-0.528	-0.811
Contig29982_RC	-0.267	-0.310	-0.120	-0.447	-0.536
Contig26811	0.229	0.157	0.120	0.283	-0.112

```
R> ##The annotation
R> genes<-fData(nki)[, c("probe", "NCBI.gene.symbol", "EntrezGene.ID")]
R> head(genes)
```

	probe	NCBI.gene.symbol	EntrezGene.ID
Contig45645_RC	Contig45645_RC	GREM2	64388
Contig44916_RC	Contig44916_RC	SUHW2	140883
D25272	D25272	<NA>	NA
J00129	J00129	FGB	2244
Contig29982_RC	Contig29982_RC	SCARA5	286133
Contig26811	Contig26811	<NA>	NA

```
R> ##Additional information (optional)
R> targets<-pData(nki)[1:5, ,drop=FALSE]
R> head(targets)
```

	samplename	dataset	series	id	filename	size	age	er	grade	pgr	
NKI_4	NKI_4	NKI	NKI	4	NA	2.0	41	1	3	NA	
NKI_6	NKI_6	NKI	NKI	6	NA	1.3	49	1	2	NA	
NKI_7	NKI_7	NKI	NKI	7	NA	2.0	46	0	1	NA	
NKI_8	NKI_8	NKI	NKI	8	NA	2.8	48	0	3	NA	
NKI_9	NKI_9	NKI	NKI	9	NA	1.5	48	1	3	NA	
	her2	brca.mutation	e.dmfs	t.dmfs	node	t.rfs	e.rfs	treatment			
NKI_4	NA	0	0	4747	0	4747	0	0			
NKI_6	NA	0	0	4075	0	4075	0	0			
NKI_7	NA	0	0	3703	0	3703	0	0			
NKI_8	NA	0	0	3215	0	3215	0	0			
NKI_9	NA	0	0	3760	0	3760	0	0			
	tissue	t.os	e.os								
NKI_4	1	4744	0								
NKI_6	1	4072	0								
NKI_7	1	3700	0								
NKI_8	1	3213	0								
NKI_9	1	3757	0								

Now we are ready to follow the workflow of section ??, i. e.:

1. **filtrate** the genes to keep only the required 50 genes by PAM50.
2. **classify** the sample/s using **genefu** implementation of PAM50 algorithm with the desired gene normalization: none, scale, robust or median. For single samples use none but, we recommend “median” for population approaches.
3. **permutate** the gene labels to build the null distribution and generate the uncertainty estimation proposed in **pbcmc**.

and further explore the obtained results with `summary`, `subjectReport` and `databaseReport`.

4.3. Using PAM50 centroids as proof of concept

For example, we could use **genefu**'s PAM50 centroids to check if our implementation solves the proof of concept, where we *a priori* know the true class of each subject:

```
R> M<-pam50$centroids
R> genes<-pam50$centroids.map
R> names(genes)<-c("probe", "NCBI.gene.symbol", "EntrezGene.ID")
R> object<-PAM50(exprs=M, annotation=genes)
R> object
```

A PAM50 molecular permutation classifier object

Dimensions:

	nrow	ncol
exprs	50	5
annotation	50	3
targets	0	0

Note that for the above output, the `targets` slot is empty, i. e., `nrow=0` and `ncol=0`. In addition, only the expression of the fifty genes and five IGP is available, with its corresponding `annotation` over the three compulsory fields. It is always a good idea to explore the slots content, to see whether they have been correctly loaded:

```
R> head(exprs(object))      ##The gene expression values for each subject
```

	Basal	Her2	LumA	LumB	Normal
ACTR3B	0.7183	-0.4817	0.009981	-0.1906	0.4657
ANLN	0.5374	0.2669	-0.579246	0.0988	-0.8369
BAG1	-0.5745	-0.4761	0.758221	-0.4055	0.3166
BCL2	-0.1188	-0.1579	0.287487	-0.4413	0.5340
BIRC5	0.3005	0.4057	-0.881434	0.6039	-0.8766
BLVRA	-0.6427	0.3353	0.042042	0.6912	-0.1634

```
R> head(annotation(object)) ##The compulsory annotation fields
```

	probe	NCBI.gene.symbol	EntrezGene.ID
ACTR3B	ACTR3B	ACTR3B	57180
ANLN	ANLN	ANLN	54443
BAG1	BAG1	BAG1	573
BCL2	BCL2	BCL2	596
BIRC5	BIRC5	BIRC5	332
BLVRA	BLVRA	BLVRA	644

```
R> head(targets(object)) ##The clinical data, if available.
```

```
data frame with 0 columns and 0 rows
```

Just as we expected, the five centroids are loaded in `exprs` slot, with their corresponding “probe”, “NCBI.gene.symbol” and “EntrezGene.ID” number in the `annotation` slot and no available data for the `targets`. Now, the user is ready to work with the data following the workflow suggested in section ?? (filtrate, classify and permutate):

```
R> object<-filtrate(object, verbose=TRUE)
R> object<-classify(object, std="none", verbose=TRUE)
R> object<-permutate(object, nPerm=10000, pCutoff=0.01, where="fdr",
+   corCutoff=0.1, keep=TRUE, seed=1234567890, verbose=TRUE,
+   BPPARAM=bpparam())
```

```
|
|
|
|=====| 25%
|
|=====| 50%
|
|=====| 75%
|
|=====| 100%
```

The intention of `filtrate` function is to keep only the genes that will take place in the classification. In this example, it will not produce any change on the original `exprs` slot, given the fact that only the required fifty genes are present. But, if a complete microarray would have been present, then, probes that do not code for IGP will be removed. In addition, probes that code for the same gene (repeated or with similar annotation) will be treated as described in standardization (`std`) parameter.

Once genes are `filtrated`, the user can `classify` them using the original PAM50 algorithm. However, here we propose to obtain subtype assignment confidence using at least $\beta = 10.000$ permutations over the SGP, using a type I error $\alpha = 0.01$ on the adjusted p-values (“`fdr`”) and a correlation difference threshold of `corCutoff=0.1`. As a matter of fact, this process is

computationally intensive, so we can take advantage of all the available computes cores using **BioParallel** package (`BPPARAM=bpparam()`) as we just did (?). In addition, the user can track the permutation progress bar by including `verbose=TRUE` option. If we now take a look at the object:

```
R> object
```

```
A PAM50 molecular permutation classifier object
```

```
Dimensions:
```

```
      nrow ncol
exprs    50   5
annotation 50   3
targets    0   0
```

```
Classification:
```

```
      nrow ncol
probability  5   5
correlation  5   5
```

```
$subtype
```

```
 Basal  Her2  LumA  LumB Normal
      1    1    1    1    1
```

```
Permutations test ran with following parameters:
```

```
Permutations=10000, fdr<0.01, corCutoff>0.1, keep=TRUE
```

```
Permutation:
```

```
correlation available: TRUE
```

```
      nrow ncol
pvalues  5   5
fdr      5   5
subtype  5   5
```

we can see that it has been updated. First, the `classification` slot contains two datasets: one with the subtype *probability*, $P(IGP_i)$, as described in section ?? and the *correlation* of each subject with the five IGP. The `$subtype` item shows a frequency table of the possible IGP with the used subjects. In addition, the used `permutation parameters` are shown with the dimension of *pvalues*, *fdr* and *subtypes*. Note that in this case `keep=TRUE` option was used so, the simulated correlation null distribution data points ($\rho_{H0_{IGPs}}$) are available.

In this example we have used **genefu**'s PAM50 centroids, thus, only one subject is present (1) for each IGP cell in the `object` output. This result is also confirmed by the ones in the diagonal of `summary(object)` matrix between the original *Subtype* and the *Classes* found by the **pbcmc** package. Moreover, this toy example only shows assigned subjects (**A**) to the original PAM50 subtypes, whereas not assigned (**NA**) marginal row/column contains only zeros (0). If ambiguous (**AMB**) subjects would have been found, *Classes* column will have included additional rows with the classes in dispute (e. g., "LumA, Normal" or "Her2, LumB", etc.).

```
R> summary(object)
```

Classes	Subtype					
	Basal	Her2	LumA	LumB	Normal	Not Assigned
Basal	1	0	0	0	0	0
Her2	0	1	0	0	0	0
LumA	0	0	1	0	0	0
LumB	0	0	0	1	0	0
Normal	0	0	0	0	1	0
Not Assigned	0	0	0	0	0	0

Finally, we can inspect the report of a single subject to see how the MS classification went (Figure ??), in order to suggest an appropriate therapy for the physician:

```
R> subjectReport(object, subject=1)
```

The report of Figure ?? is a `grid.arrange` object which basically consists of three main parts:

tableGrobs: A summary table which contains the following fields,

\$Summary: Subject name and subtype obtained by PAM50.Subtype or the proposed methodology (`Permuted.Subtype`).

\$Fields: For the five PAM50 subtypes,

- Correlation: The correlation of the i – th PAM50 centroid with the observed subject exprs, $\rho(PGP, IGP_i)$.
- p-value: Permutation p-value obtained using the simulation data, p_{IGP} .
- FDR: Adjusted p-value using False Discovery Rate (?).

facet_wrap: Two rows to display `ggplot2` (?) scatter plots of subject exprs versus PAM50 centroids (??) and a linear regression fit (in blue). If the subject has an unique subtype, then the graph is colored in red. In addition, if simulated permutations were run with `keep=TRUE` option, then the null distribution boxplots are plotted with the corresponding observed un-permuted correlations as big round dots.

textGrobs:The permutation `parameter` slot used in the simulation.

The `pbcmc` also includes the ability to get a `pdf` report for the complete database calling `databaseReport` function. In this context, the first page is a global summary of the database, i.e., a `summary` contingency table of the permuted test classes against the original PAM50 subtypes results. The following pages are the respective `subjectReport` outputs as the one shown in Figure ??.

	Summary	Field	Basal	Her2	LumA	LumB	Normal
<i>Subject</i>	Basal	Correlation	1	-0.146	-0.649	-0.360	0.118
<i>PAM50.Subtype</i>	Basal	p-value	0	0.851	1.000	0.994	0.204
<i>Permuted.Subtype</i>	Basal	FDR	0	1.000	1.000	1.000	0.511

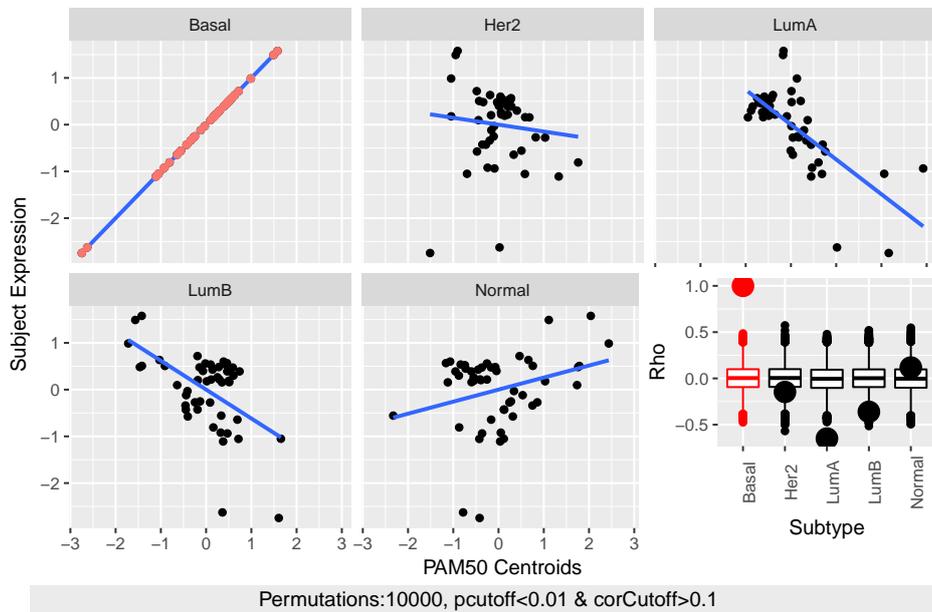


Figure 3: PAM50 permutation subject report for **genefu**'s "Basal" intrinsic gene profile (IGP). The top table summarizes the results for the i -th subject, i. e., the correlation, p-value and false discovery rate (fdr) obtained for each IGP. In addition, scatter plots of the observed subject gene profiles against the IGP with the linear regression line (in blue). Red color indicates the assigned subtype. Finally, a boxplot for each IGP null permuted correlation distribution and big dots to represent the un-permuted observed correlations.

5. Conclusion

The **pbcmc** package characterizes uncertainty assessment on gene expression classifiers, a. k. a. molecular signatures, based on a permutation test. In order to achieve this goal, synthetic simulated subjects are obtained by permutations of gene labels. Then, each synthetic subject is tested against the corresponding subtype classifier to build the null distribution. Thus, classification confidence measurement report can be provided for each subject, to assist physician therapy choice. At present, it is only available for PAM50 implementation in **genefu** package but, it can easily be extend to other molecular signatures.

Acknowledgements

Funding: This work was supported by Universidad Católica de Córdoba (PIP 800-201304-00047-CC to E.A.F.), Argentina and National Council of Scientific and Technical Research (CONICET), Argentina.

Session Info

```
R> sessionInfo()
```

```
R version 3.3.2 (2016-10-31)
```

```
Platform: x86_64-apple-darwin13.4.0 (64-bit)
```

```
Running under: OS X Mavericks 10.9.5
```

```
locale:
```

```
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
attached base packages:
```

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

```
other attached packages:
```

```
[1] breastCancerNKI_1.12.0 BiocParallel_1.8.1
[3] pbcmc_1.2.2 genefu_2.6.0
[5] AIMS_1.6.0 Biobase_2.34.0
[7] BiocGenerics_0.20.0 e1071_1.6-7
[9] iC10_1.1.3 iC10TrainingData_1.0.1
[11] pamr_1.55 cluster_2.0.5
[13] biomaRt_2.30.0 limma_3.30.6
[15] mclust_5.2 survcomp_1.24.0
[17] proclim_1.5.7 survival_2.40-1
[19] ggplot2_2.2.0
```

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

```
[1] Rcpp_0.12.8 plyr_1.8.4 class_7.3-14
```

[4]	bitops_1.0-6	tools_3.3.2	digest_0.6.10
[7]	RSQLite_1.1	memoise_1.0.0	tibble_1.2
[10]	gtable_0.2.0	lattice_0.20-34	Matrix_1.2-7.1
[13]	DBI_0.5-1	amap_0.8-14	bootstrap_2015.2
[16]	gridExtra_2.2.1	stringr_1.1.0	SuppDists_1.1-9.4
[19]	S4Vectors_0.12.1	IRanges_2.8.1	cowplot_0.7.0
[22]	stats4_3.3.2	grid_3.3.2	AnnotationDbi_1.36.0
[25]	XML_3.98-1.5	lava_1.4.5	rmeta_2.16
[28]	magrittr_1.5	reshape2_1.4.2	scales_0.4.1
[31]	survivalROC_1.0.3	splines_3.3.2	assertthat_0.1
[34]	colorspace_1.3-1	labeling_0.3	KernSmooth_2.23-15
[37]	stringi_1.1.2	RCurl_1.95-4.8	lazyeval_0.2.0
[40]	munsell_0.4.3		

Affiliation:

Cristóbal Fresno, Germán A González & Elmer A Fernández
Bioscience Data Mining Group
Facultad de Ingeniería
Universidad Católica de Córdoba - CONICET
X5016DHK Córdoba, Argentina
E-mail: cfresno@bdmg.com.ar, ggonzalez@bdmg.com.ar,
efernandez@bdmg.com.ar
URL: <http://www.bdmg.com.ar/>

Andrea S Llera
Laboratorio de Terapia Molecular y Celular
Fundación Instituto Leloir - CONICET
C1405BWE Ciudad Autónoma de Buenos Aires, Argentina
E-mail: allera@leloir.org.ar
URL: <http://www.leloir.org.ar/podhajcer/>