# Explained classification of high-dimensional tabular data

*UC3M 2020*

## Introduction

AI, machine learning, big data, ... sometimes become buzzwords in an attempt by the press to depict the technology as something revolutionary or even mystical. Often companies use the same words for marketing purposes, to present themselves as highly sophisticated when sometimes it's not even clear if they are achieving anything really meaningful to improve their services through AI. Despite the need to keep the expectations down to earth, machine learning is already providing tangible results in a wide range of problems, it's becoming increasingly important and a norm in many industries. Private investment is accelerating and governments around the world are also pouring money into what is considered a strategic sector. The momentum is there and it seems unlikely to stop.

One of the main strong points of machine learning is its potential to add value in almost every known domain. Wherever there is data, machine learning can step in and help to take better decision, from improving the personal assistance of your mobile to helping in the prevention of diseases. This is why data is sometimes referred to as the "new electricity", the prospect of a future society that will greatly depend on consuming data through AI technology in every aspect of its everyday life. This potential feature of ubiquity of machine learning highlights the importance of understanding how decisions are taken, specially in applications involving sensitive areas such as health, privacy or social fairness, where trust needs to be built before the technology is fully adopted. In that spirit, the EU introduced recently what is commonly known as the "right to explanation" in the GDPR regulation:

*(71) The data subject should have the right not to be subject to a decision, which may include a measure, evaluating personal aspects relating to him or her which is based solely on automated processing and which produces legal effects concerning him or her or similarly significantly affects him or her, such as automatic refusal of an online credit application or e-recruiting practices without any human intervention. [1]*

Being able to explain the decisions of ML models is not just a matter of complying with regulation, it can also be an engineering tool to detect faults that otherwise would go unnoticed. For instance, lack of diversity in the data (or just not enough data) could make overfitting detection a hard task. The examination of the decisions could help to identify the problem in a qualitative way (e.g. is your image classifier focusing on the object related to the predicted label, or is the decision actually based on something in the background that happens to be correlated to the label in the data). Lack of generalization could also be exploited in adversarial attacks, a concern in fields like cybersecurity or automated driving. It's clear that the more interpretable the model is, the easier these problems can be avoided.

### Motivation

When it comes to choosing and tuning a model to predict / explain data, we always face the predictability vs interpretability trade-off, naturally impossible to avoid. Sometimes a simpler model easy to explain is good enough, in other cases we don't want to forgo the prediction performance of a more opaque model. In this project we'll focus on the later case and will aim to close the predictability vs interpretability trade-off as much as we can. To try to achieve it, we'll resort to a second model to explain the predictions, that is, we'll handle two models: a prediction model which consumes data to make predictions in the domain of interest, and an interpretable model which receives the prediction model as input to make interpretations of its decisions.

More specifically, the motivation of this project is to explore how to obtain meaningful interpretations in the following real-world setup:

- A classification problem with tabular data. Tabular data is very common in many different domains.
- A dataset with a large number of independent variables. Having to deal with high-dimensional data is specially challenging in terms of interpretability, the higher the number of independent variables, the higher the variance in the universe of explanations and therefore the more valuable would be a robust interpretation method.

- An optimal prediction model. In general opaque models predict better for complex data than simple models, but it comes at a price, their outcomes are harder to explain. We'll use neural network models with tens of thousands of parameters to classify the data, making difficult the identification of features that are influential in the response of the model.

**Interpretable models**

The interpretable model in the scope of this project must have the following properties:

- It must be model-agnostic, a model that only uses the input and the output of the prediction model, not caring about the prediction model itself. This independence could come in handy if at some point the prediction model has to be replaced with some other model performing better, something not unusual in the constantly changing field that is ML. The model-agnostic interpretable model will always remain valid at no development cost.
- It must be capable of dealing with the large number of features in the data. Namely it should be able to provide certain degree of interpretability by simplifying the high-dimensional data space, and it should be reassuring in its interpretations: they should make sense from a human perspective, several interpretations for the same observation should be consistent, etc
- In addition, we want to be able to analyze the interpretations both at a global scale (how the prediction model behaves in general) and at a local scale (explaining the classifications for particular observations of interest in particular local regions of the data). Understanding the prediction model from both perspectives and their relationship is key to get a complete vision of the interpretations and gain trust in the interpretable model.

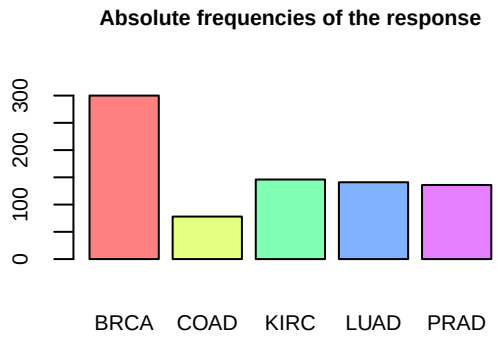
The LASSO (least absolute shrinkage and selection operator) is well suited for these requirements. It can operate in parallel with the prediction model by only using its inputs and outputs. It performs regularization and feature selection to deal with high-dimensionality and it can be fit locally, even in sparse local regions of the data, with the help of LIME (Local Interpretable Model-Agnostic Explanations) as we'll see later.

## Datasets

The implementation of the models will be tested with two different datasets to contrast the results. Both datasets contain a large number of features, being one of them very sparse.

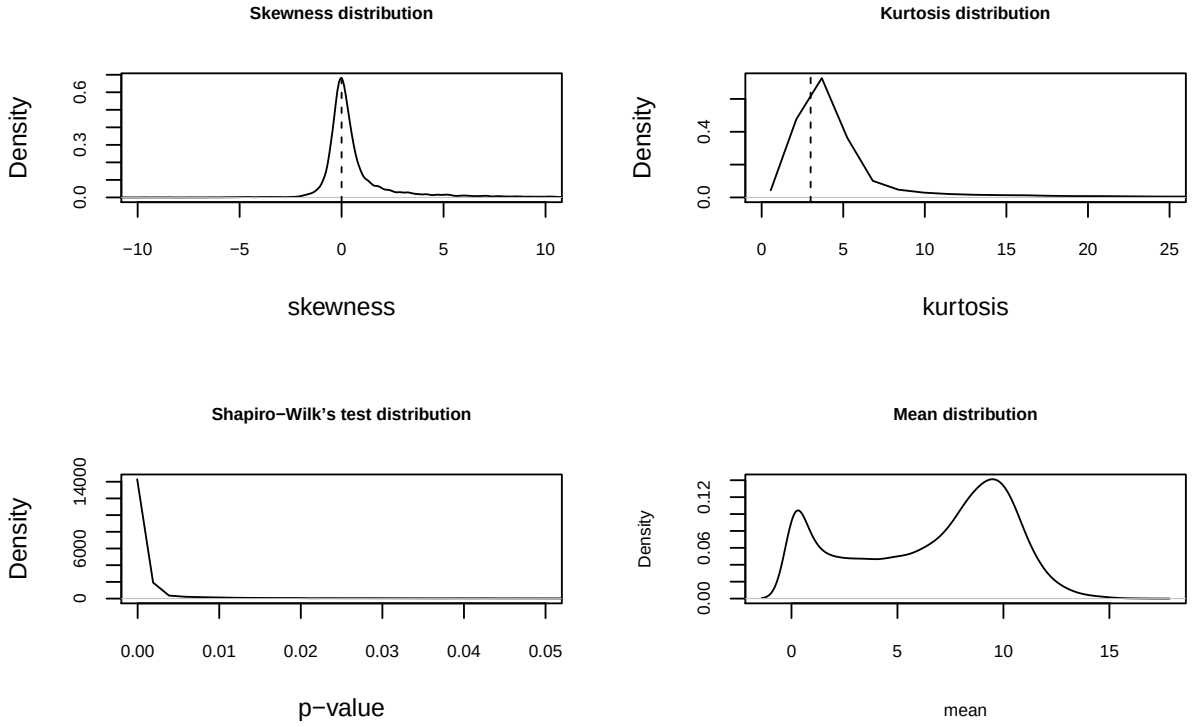
### Genes dataset

The data consists of 801 patients with tumors occurring in different parts of the body. The tumor types covered include: lung adenocarcinoma (LUAD), breast carcinoma (BRCA), kidney renal clear-cell carcinoma (KIRC), colon adenocarcinoma (COAD) and prostate adenocarcinoma (PRAD).

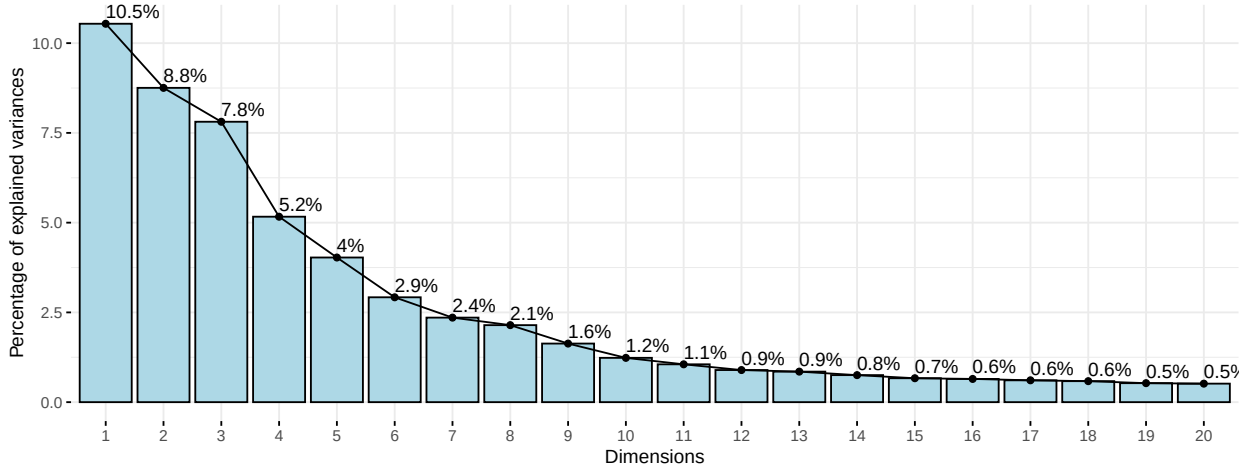



Among more than 20000 RNA sequencing gene expression levels, the goal is to identify which genes could have been altered through mutation causing the condition.

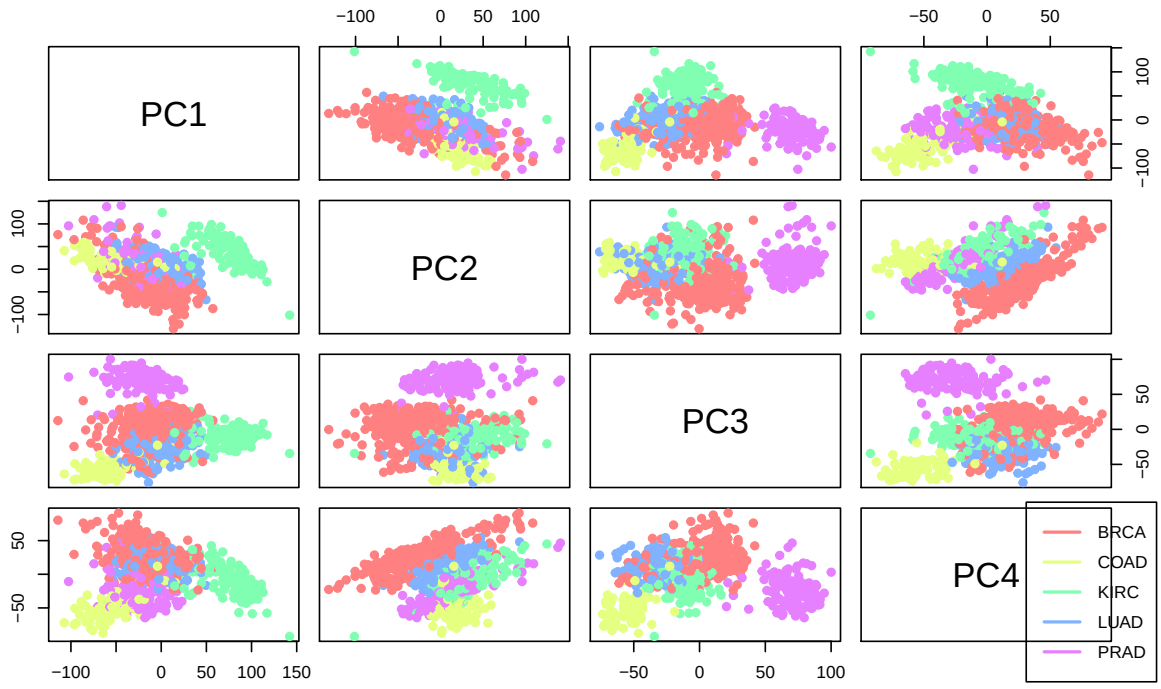
The distribution of values for each gene expression in the dataset is symmetric in general but not normal. In addition, the distribution of means suggests there are two categories of gene expressions.




The data is scaled to help with the training of the prediction model and the interpretations. At least 45% of the variance of the data is explained by the first 10 PCA components. The explained variance concentration in the PCA components, plus having more than 20000 gene expressions and only 5 categories suggests high multicollinearity.




First 4 PCs look enough to classify the types of cancer despite only accounting for a third of the variability of the data:




Classifications are done with a fully connected network with one hidden layer.
The accuracy of the network is close to 1 and the categorical cross-entropy loss is close to 0:

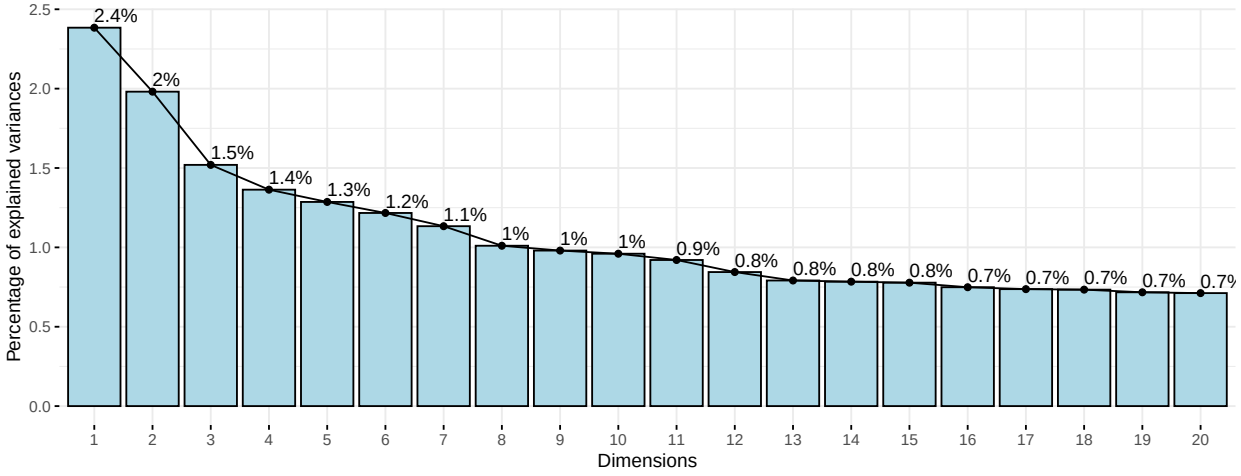
```
## Trained on 640 samples (batch_size=32, epochs=4)
## Final epoch (plot to see history):
##         loss: 0.000003682
##     accuracy: 1
##     val_loss: 0.0006472
## val_accuracy: 1
```

**Proteins dataset**

TODO: Description y analisis

The 9972 observations are proteins that are candidates for having anti-freezing properties. The goal is to identify which observations are antifreeze proteins (AFPs), important for the survival of animals, plants, fungi and bacteria in extreme cold environment conditions.

The 841 features represent attributes of the molecule structure (amino acid and di-peptide compositions).




Classifications are done with a fully connected network with one hidden layer. The training data is downsampled to deal with the imbalanced data - only 0.018% of proteins are AFPs.
The accuracy of the network is above 0.8, with balanced sensitivity / specificity.
TODO: analizar si es mejor evitar mas los falsos positivos

```
## Confusion Matrix and Statistics
##
##    y_test_p
##        1    0
##   1  151 1407
##   0   30 7784
##
##                Accuracy : 0.8467
##                  95% CI : (0.8392, 0.8539)
##     No Information Rate : 0.9807
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.144
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.83425
##             Specificity : 0.84692
##          Pos Pred Value : 0.09692
##          Neg Pred Value : 0.99616
##              Prevalence : 0.01931
##          Detection Rate : 0.01611
##    Detection Prevalence : 0.16624
##       Balanced Accuracy : 0.84058
##
##        'Positive' Class : 1
##
```

## Global surrogate model

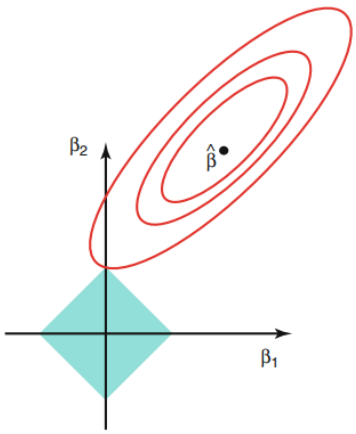
A surrogate model is an interpretable model that tries to mimic a non-interpretable model. The surrogate model is fit with the features of the training data and the outputs of the non-interpretable model, its role is not to make predictions but to provide a way to understand how the non-interpretable model makes predictions. The result is an approximation of the non-interpretable model that helps to identify the features in the data that drive the predictions of the non-interpretable model (which we'll call *black box* from now on) at a global level (i.e. for no particular prediction).

When the surrogate model is fit, overfitting is not something to avoid but rather welcomed since we want to get a good approximation of the black box (which already should have took care of properly generalizing the data). However there is a trade-off between how well the interpretable model fits the black box and the degree of interpretability of the interpretale model. This is known as the fidelity vs interpretability trade-off [2].

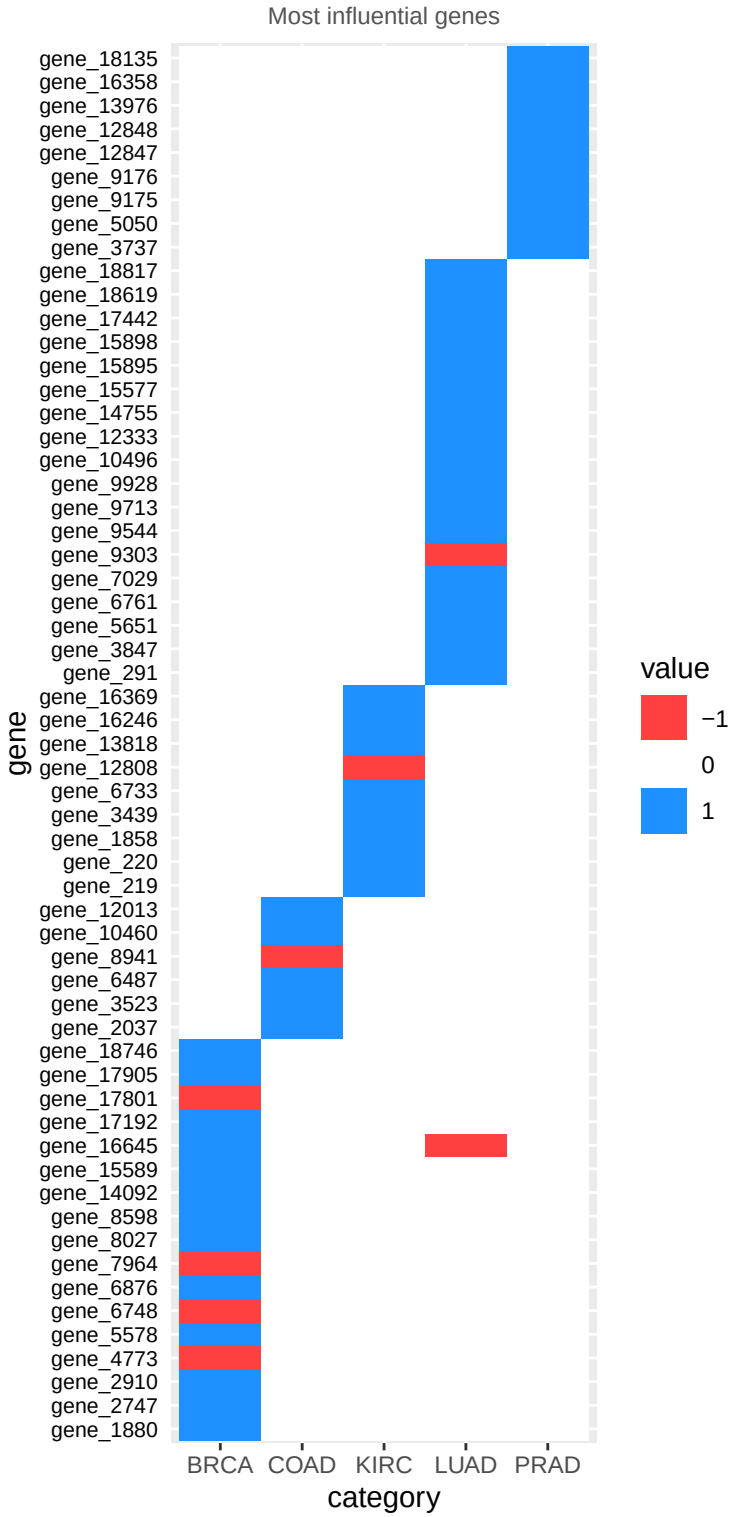
### The lasso

We humans have a hard time dealing with large number of variables at the same time, we have to use abstraction to be able to cope with all the information in an effective way, although introducing all sort of biases in the process. The philosophy behind lasso is somewhat similar, lasso performs feature selection and improves accuracy in high-dimensional data by greatly decreasing variance but also introducing some bias. This makes lasso a good candidate for a global surrogate model that helps us to focus on the bunch of features that are really important to explain the black box outcome.

Lasso is a well known model, it performs regularization by introducing a coefficient penalization component. An intuitive way to understand how lasso works is seeing it as a linear optimization problem where the vertices of the constraint shape lie on axis intersections where some variables are set to zero due to the absolute value function in the penalization.



With more than 20k variables, the genes dataset is quite an extreme case. The aim is to reduce the number of features to a bunch we can handle. It turns out that by fitting the surrogate lasso model with an optimal lambda, the number of features selected by lasso is indeed small.



Lasso managed to get rid of a vast amount of redundant information in form of multicollinearity, while introducing some bias (the selection of features).

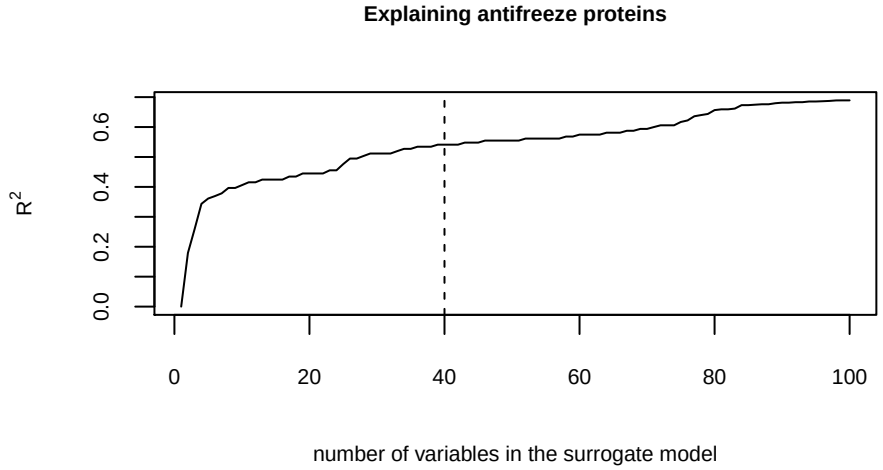
To measure how well lasso explains the black box behavior, we can rely on the proportion of explained variance ($R^2$). It turns out that lasso explains the black box with very high fidelity, and just with a small manageable bunch of features for each category.

```
## $`0`
## [1] 0.991953
##
## $`1`
## [1] 0.991953
##
## $`2`
## [1] 0.991953
##
## $`3`
## [1] 0.991953
##
## $`4`
## [1] 0.991953
```

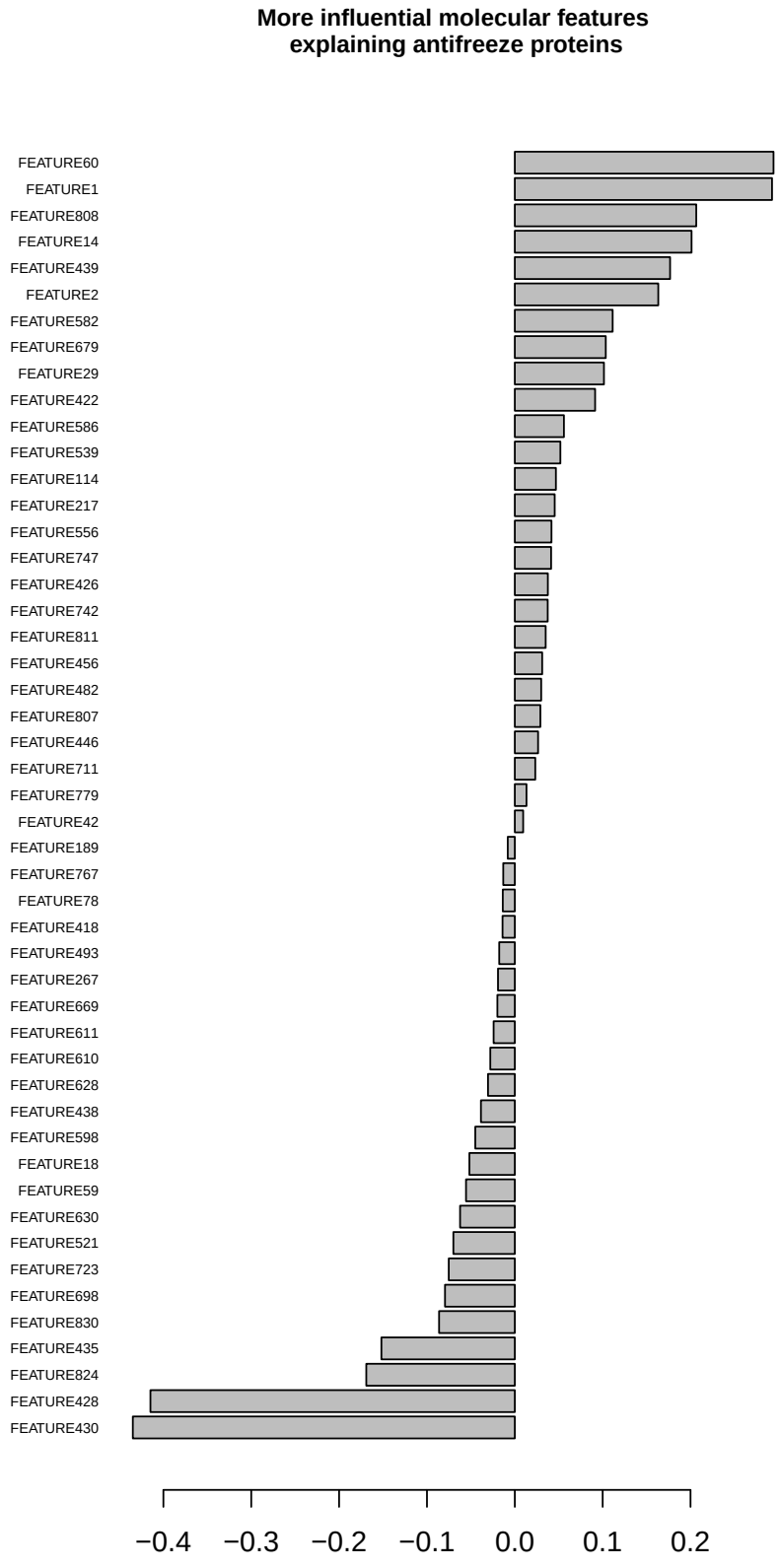
Since lasso almost completely explains the black box behavior, we could probably use it as prediction model in the first place and still get the same accuracy.

Something worth to mention too is how computationally efficient is lasso, crucial when dealing with high-dimensional data.

In contrast, the proteins dataset has far fewer variables to explain. However in this case, lasso struggles to explain the black box with high fidelity with a small set of features. The following plot illustrates the trade-off.




A compromise has to be reached. Should we want to be more confident with the lasso model, we'd have to bear with the burden of more selected features. The following plot shows the 50 more influential features, with $R^2$ around 0.46


More influential molecular features
explaining antifreeze proteins
FEATURE60
FEATURE1
FEATURE808
FEATURE14
FEATURE439
FEATURE2
FEATURE582
FEATURE679
FEATURE29
FEATURE422
FEATURE586
FEATURE539
FEATURE114
FEATURE217
FEATURE556
FEATURE747
FEATURE426
FEATURE742
FEATURE811
FEATURE456
FEATURE482
FEATURE807
FEATURE446
FEATURE711
FEATURE779
FEATURE42
FEATURE189
FEATURE767
FEATURE78
FEATURE418
FEATURE493
FEATURE267
FEATURE669
FEATURE611
FEATURE610
FEATURE628
FEATURE438
FEATURE598
FEATURE18
FEATURE59
FEATURE630
FEATURE521
FEATURE723
FEATURE698
FEATURE830
FEATURE435
FEATURE824
FEATURE428
FEATURE430
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2

## LIME

Global surrogate models provide a global sense of the influence of the features in the response. However if the data structure is complex, it might not explain well the interpretation of some particular predictions if the surrogate global model doesn't fit the black box well enough. Some specific regions of the data space could be dominated by specific features that got overlooked by the global surrogate model, in other words, the fidelity of the global surrogate model is not constant across the data space.

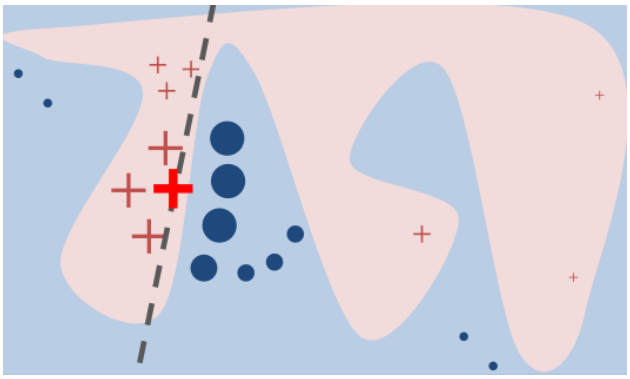
LIME (Local interpretable model-agnostic explanations) is a model-agnostic interpretability model that aims to reduce the fidelity vs interpretability gap at a local level. The method was proposed in the *Why Should I Trust You?: Explaining the Predictions of Any Classifier* paper (Ribeiro, Singh, and Guestrin 2016) [2]. The intuition behind this technique is to approximate the black box locally by an interpretable model by assuming that the data structure is linear around particular inputs, hence allowing for a model "tailored" to the region of interest.

The goal is to find the local model that maximizes, in a balanced way, both the local fidelity and the interpretability. The idea is formalized as:

$$g_x = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Where, in our classification context:
$x$ is the observation of interest for which we want to explain the classification.
$g$ belongs to the universe $G$ of all possible interpretable models (lasso, decision trees, ...).
$g_x$ is the model explaining the classification of $x$ that is optimal in terms of both interpretability and fidelity.
$\Omega(g)$ is a function that measures the complexity of an interpretable model (for instance the number of selected features by lasso).
$f$ represents the black box we are trying to explain. The response of the model is the probability than an observation belongs to a certain class.
$\pi_x$ is a function that measures the distance between any observation in the data space an $x$.
$\mathcal{L}(f, g, \pi_x)$ is a function that measures how unfaithful $g$ is in approximating $f$ in the locality defined by $\pi_x$

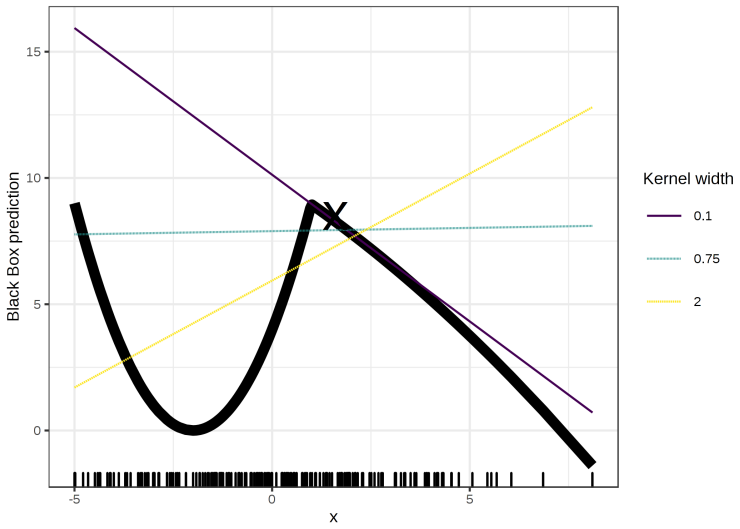
In the illustration example below, each axis represents an explanatory variable of the data, whereas the background color represents a binary classification. The observation of interest is the big red cross. If we zoom in enough, the frontier between different categories is linear and therefore the assumption of LIME is correct and we can fit a linear model with high fidelity. More data is simulated around the observation to compensate for the sparsity of data in the "zommed in" region.



**Implementation**

Different implementations of LIME exist. In this project we'll use the one coded in the R package: *lime*, which implements the following steps:

- Before a local linear model is fit around an observation which classification we want to explain, new data points are simulated in the surroundings of the observation of interest. To achieve that, kernel densities estimations are computed for each variable and sampled to simulate the new data, reducing the sparsity of the data and therefore the variance in the local fit for the explanation. The parameter to tune for this step is the number of simulated data points (*n_permutations*).
- Of data points simulated by sampling from the features kdes, we are specially interested in those in the surroundings to the observation of interest, as we are fitting a local model. So we need to give more weight to the data close to the instance of interest and this is done with a smoothing exponential kernel. The width of the kernel is a parameter of the LIME model (*kernel_width*) and probably the more tricky one to tune as shown in the following example of a 1-dimensional dataset:




In this example changes in the kernel width lead to drastic changes in the local linear fit for the instance of interest (the cross in the plot). Adding more dimensions would increase the sensitivity of the width parameter even more.

The default value in the *lime* package is $0.75\sqrt{p}$, the more number of dimensions (p) the more the data sparse and therefore the kernel width should be increased accordingly. The appropriate value seems to depend on the surroundings of the data point being explained so there isn't a clear rule of thumb to follow for all the cases.

Another parameter is the distance function that measures the proximity between the instance of interest and the simulated data points (*dist_fun*). The default option is Gower's distance but others like Euclidean or Manhattan can also be used.

TODO: Justificar la eleccion de la funcion de distancia para datasets sparse y con variables no normales.

- The next step is to feed the black box with the simulated data to get the classifications required to fit the local model.
- With the simulated data and the corresponding predictions, a local model is fit (*feature_select*). Several models are available, we'll choose lasso for two reasons: the high-dimensionality of our datasets and to

get a better comparison with the global surrogate lasso model. Lasso will use the the smoothing kernel to give more weight to the data points in the neighborhood of the original observation to explain.

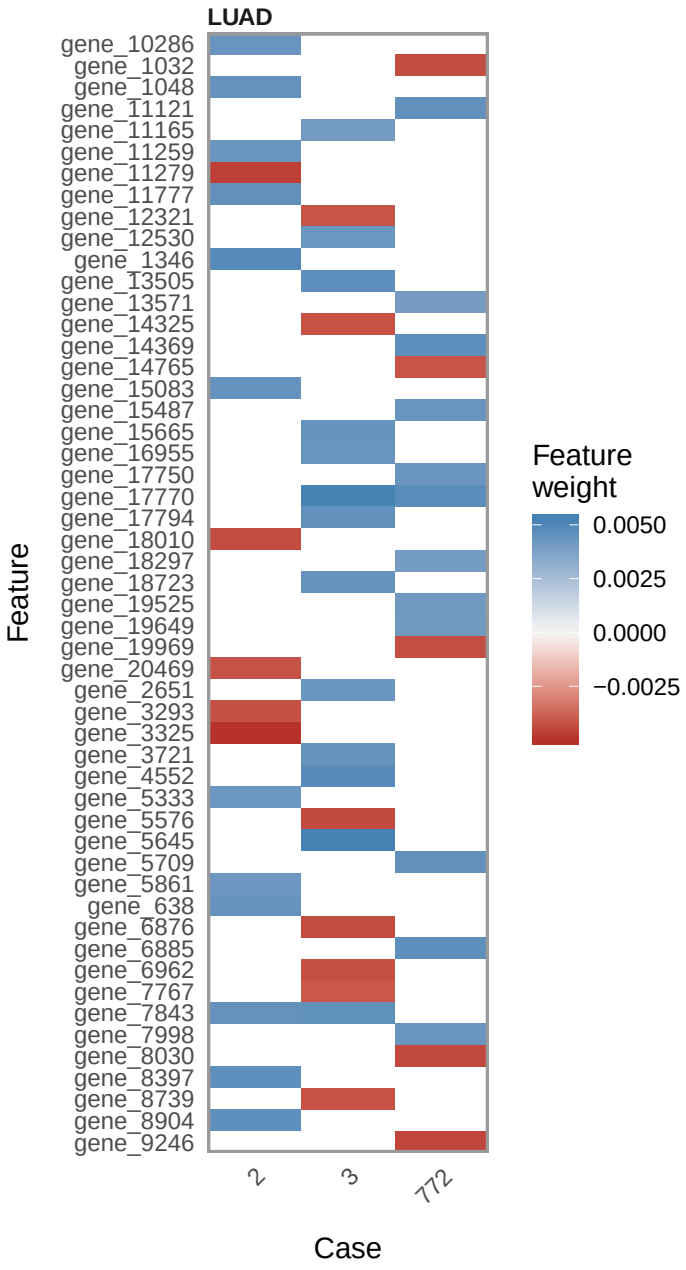
- Finally the coefficients with higher absolute values are selected (*n_features*) to explain the output (*n_labels*, *1* if we are just interested in the selected category).

**Applying LIME to the genes dataset**

The following LIME test is performed on the genes dataset:

- An observation belonging to LUAD category is picked up at random to explain its classification. Observations in LUAD category overlap the most with other categories, they are the hardest to predict and hence more interesting to explain.
- The number of features to explain the observation is set to 20.
- The number of permutations in the simulated data is set to 20000, the maximum possible before start getting memory allocation errors (with 32GB RAM). The local model will be fit in a weighted 20000 permutations x 20257 genes matrix.

The result is very poor, the selected features are completely different between interpretations. Too many genes are competing with each other to explain the predicted category and depending on the simulated data built around the observations (ultimately the seed), the "wining" features will be different.
TODO: Anadir referencia a "On the Robustness of Interpretability Methods" (Alvarez-Melis and Jaakkola 2018). "Sometimes even slight changes in the neighborhood affects strongly obtained explanations."



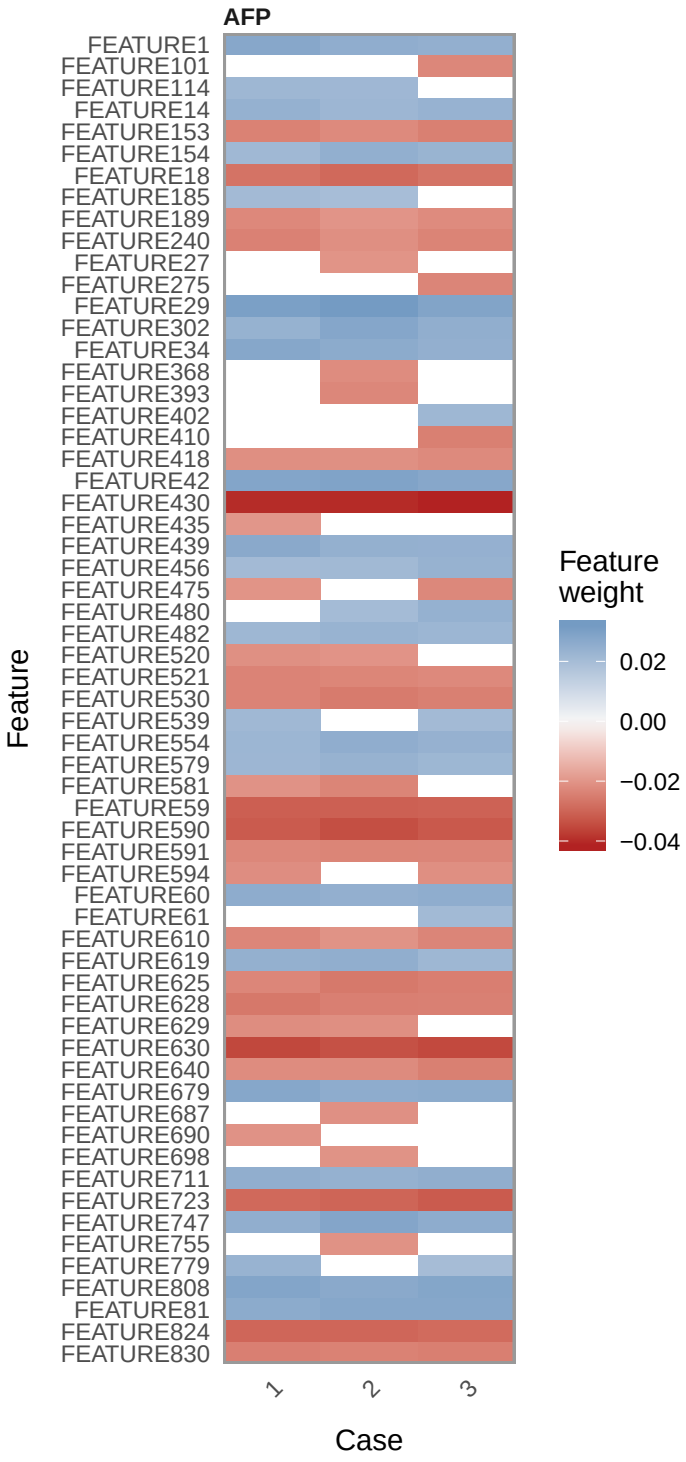

A way to decrease the variance in the selection of features would be to increase either the number of selected features or the number of permutations (if possible) in order to make the simulated data more consistent between interpretations.

Different kernel widths have also been tried but without improvement.

### Applying LIME to the proteins dataset

The same test is performed on the proteins dataset to explain the classification of one of the antifreeze proteins in the training data. The number of selected features remains 50 as with the global surrogate model.

The size of the simulated data is a 50000 permutations * 840 features matrix. The result for observation is quite robust.



## Interpretable data representations

A big advantage with LIME is that the data representation for the interpretations can be decoupled from the original data fed to the black box. We could use whatever is more convenient for the interpretations while still using the original data to get the best possible predictions, as long as we keep a mapping between both data representations. This indirection property, in addition to the model-agnostic property, makes LIME very flexible.

A usual example of interpretable data representation are superpixels in the field of image classification. A superpixel represents a segment of an image that groups pixels that are interconnected and share similar colors. As opposed to individual pixels, this representation is natural for human understanding and simplifies the identification of specific regions that could have high influence in the classification of an image. For instance, the below picture could be labeled as both dog and guitar. LIME, by fitting a surrogate model in a dataset that consists on copies of the original picture where only some superpixels are left enabled and their corresponding classifications done by the black box model can identify which superpixels have an impact in the classification of the image for a particular label. The black box model still uses the pixels to label the images, but the interpretation is done at a higher level, easier to understand for us than individual pixels.

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

(image taken from the paper [2])

Similarly, to get the variations of the data for text, the solution is to turn off single words. A text classifier can rely on abstract word embeddings as features, but the explanation can be based on the presence or absence of words in sentences.

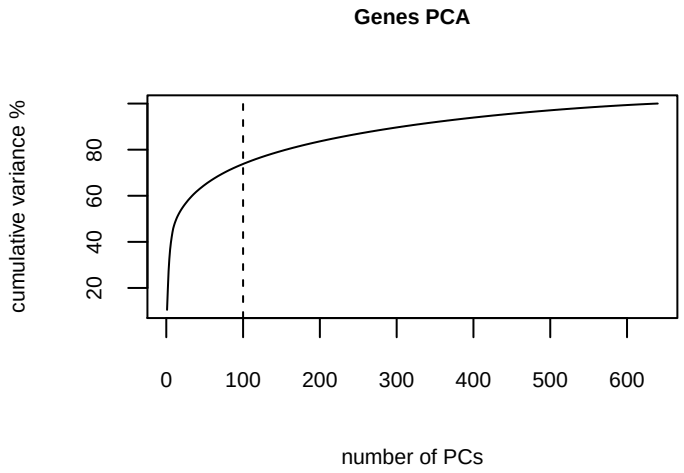
Next we'll try to find relevant data representations for the interpretation of high dimensional tabular data to try to overcome the problems we encountered explaining the original data.

**Interpretation with PCA components**

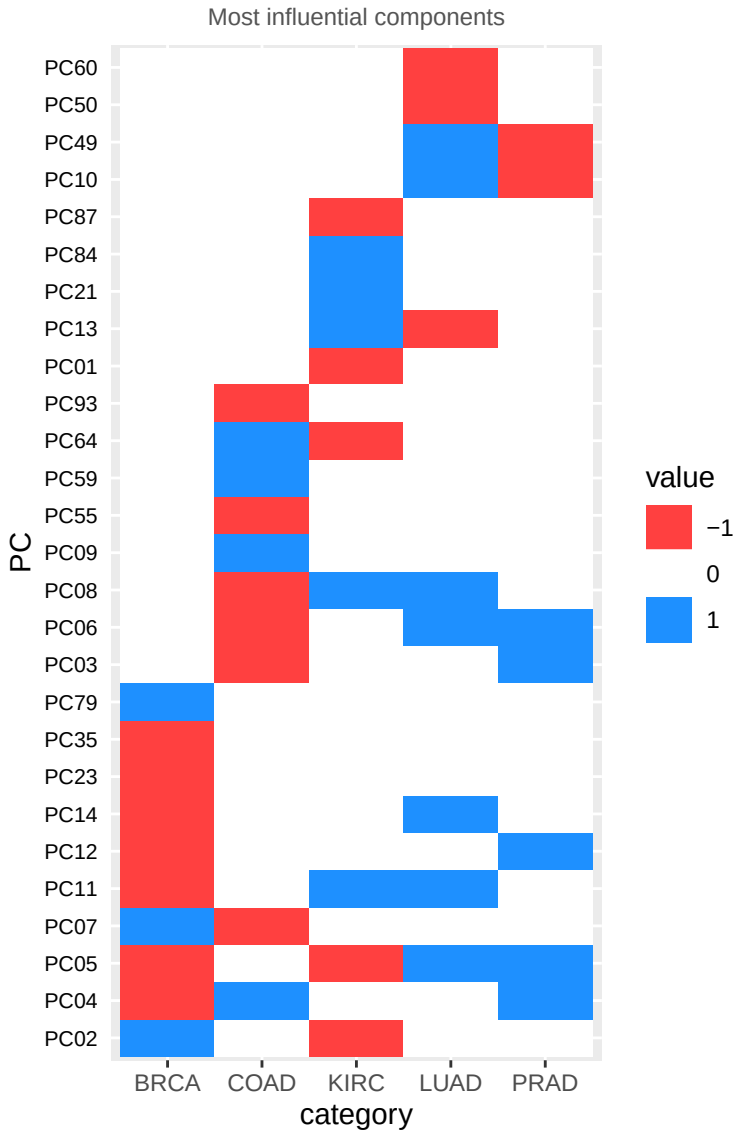
A first method to get a more interpretable data representation with tablar data would be dimensionality reduction. We'll use classic PCA which could help in two ways:

- To reduce the number of features to deal with while increasing fidelity.
- To potentially get hidden meaningful features from linear combinations of the original variables.

For the genes dataset, the number of components included in the interpretable data representation space could be reduced to a percentage of the explained variance, since probably only a few dozens of them will be really influential. We'll use the first 100 PCs, accounting for more than 70% of the explained variance of the data.



The surrogate global lasso model is fit again, this time with the first 100 PCA components.



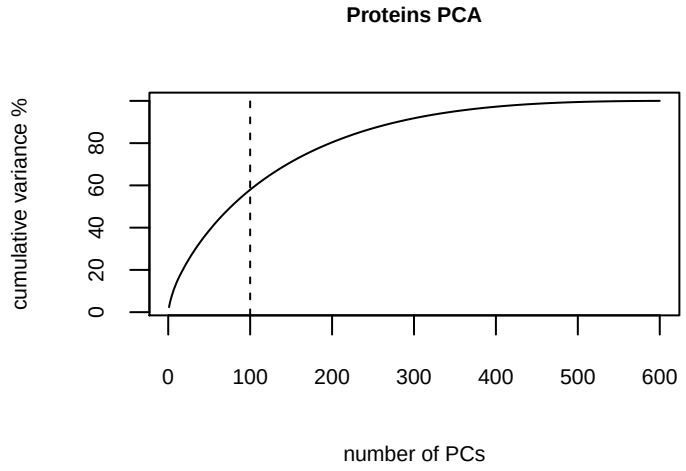

Global fidelity remains high as expected:

```
## $`0`
## [1] 0.9663434
##
## $`1`
## [1] 0.9947884
##
## $`2`
## [1] 0.9990028
##
## $`3`
## [1] 0.9828026
##
```

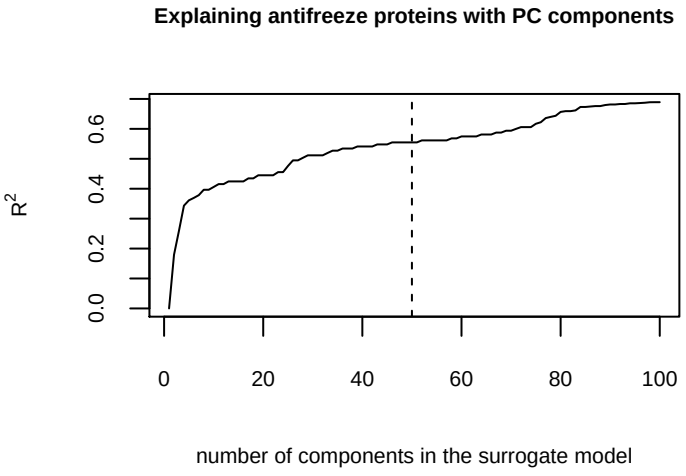
```
## $`4`
## [1] 0.9990028
```

The usefulness of this method will depend on the practitioner, expert in the domain, being able to make sense of the main components involved in the explanations.

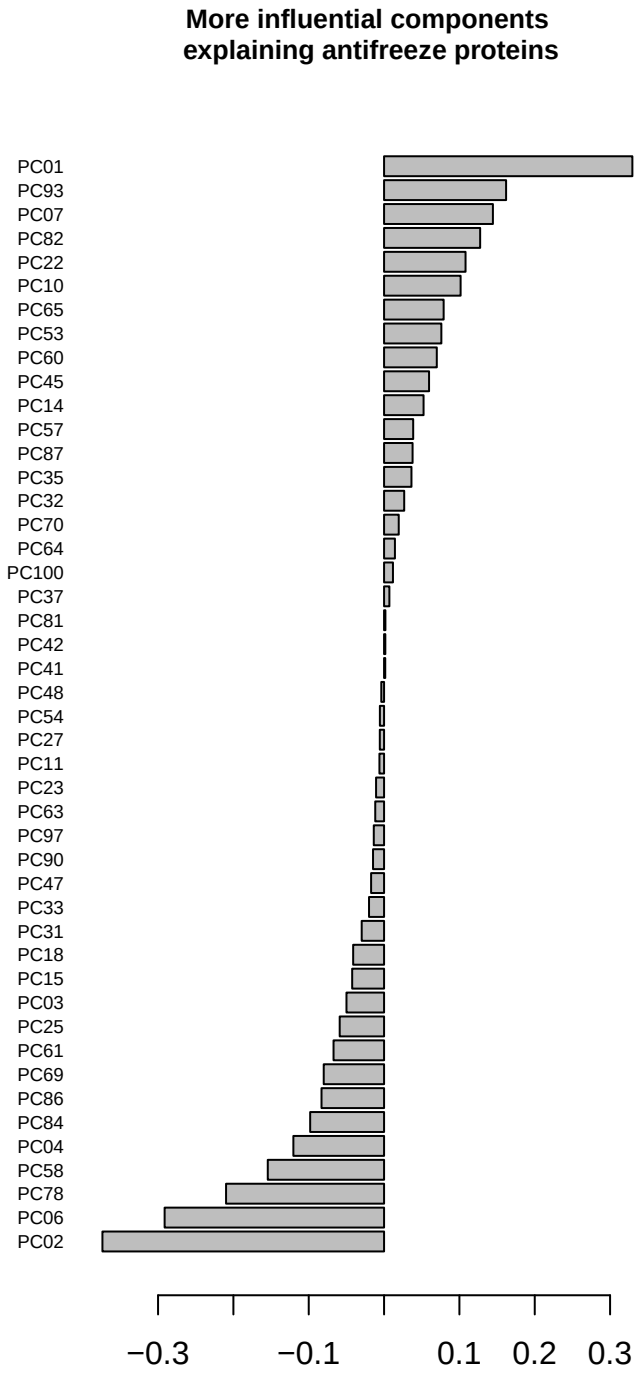
For the proteins dataset, the surrogate global lasso model is fit also with the first PCA components.




Maintaining the same number of interpretable features as with the original data (50), there is an improvement in $R^2$, from 0.46 with the original data to 0.56 with PCA components.

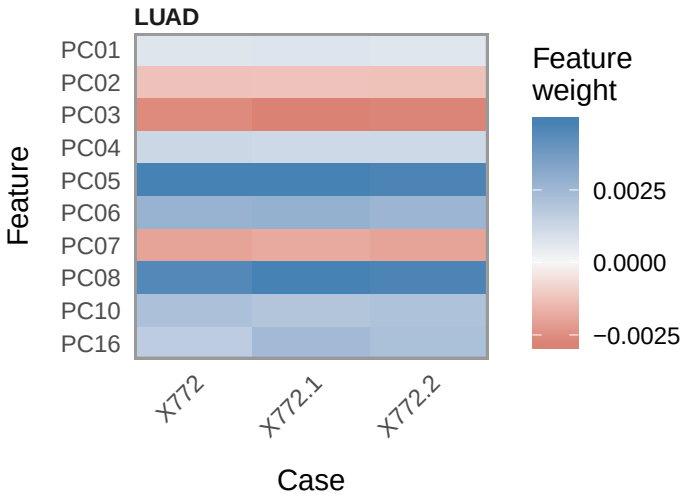


Interestingly enough, some components with low cumulated explained variance like PC93 or PC78 are very influential.



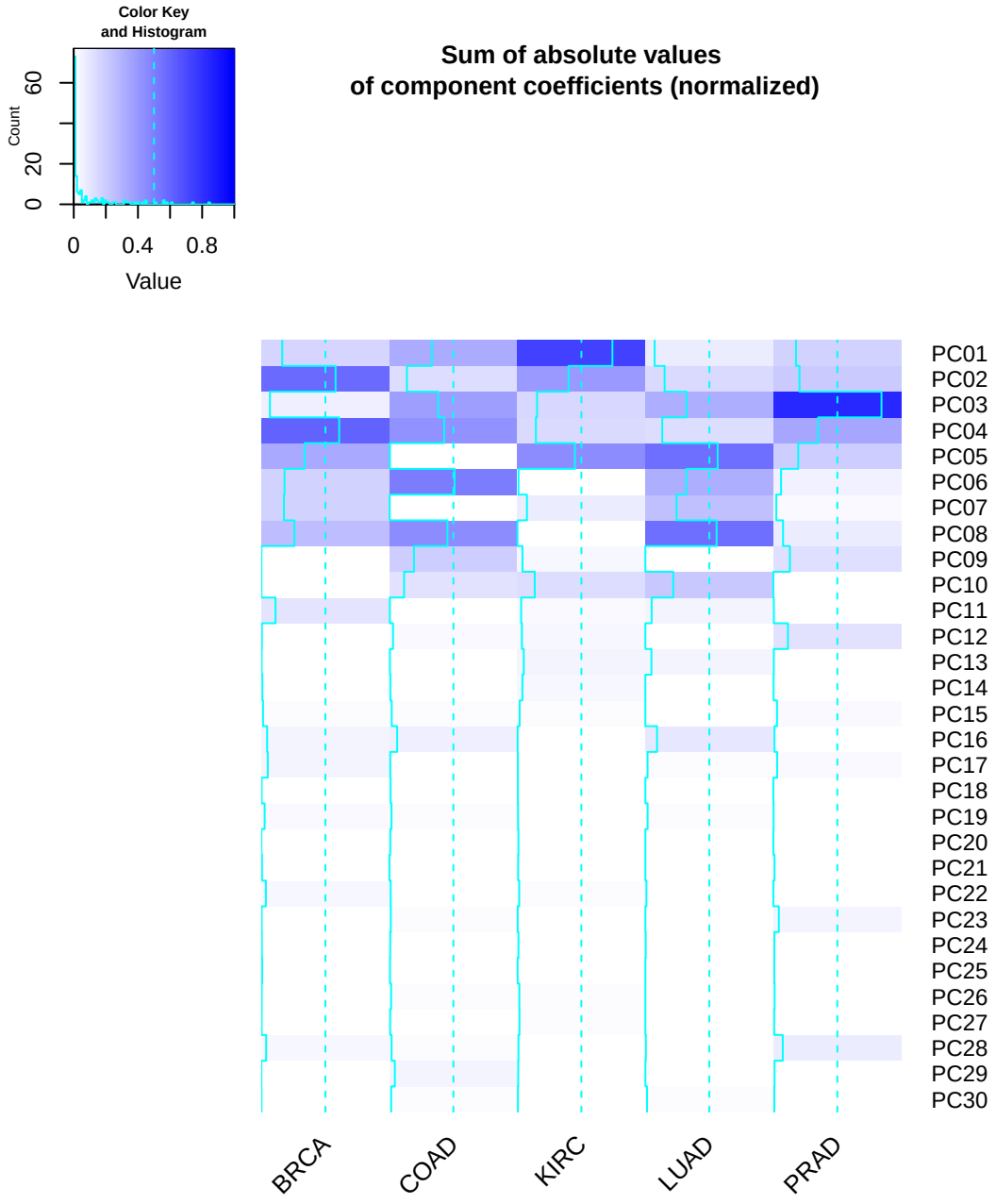

Again, the expert in the domain has to make sense of the components in order for this technique to be useful.

At a local scale, the LIME model applied to the genes dataset is now robust. Several interpretations for a same observation always show very close results.

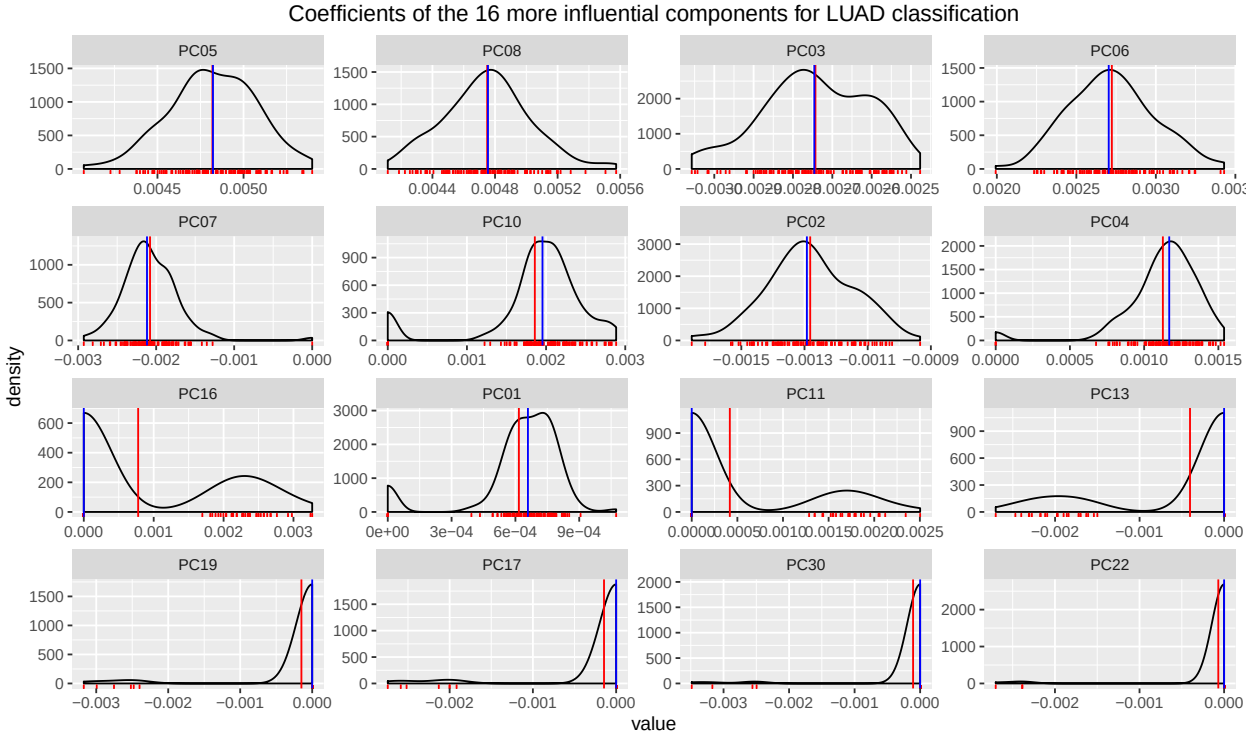



We now check the correlation between the global surrogate model and the local LIME models of all the observations for each category, using the training data (640 observations) and 2000 permutations in LIME for each observation. Some level of general correlation is expected, in both cases the classifications are explained through a lasso model, the main differene being how local / global is the data used for the fitting.

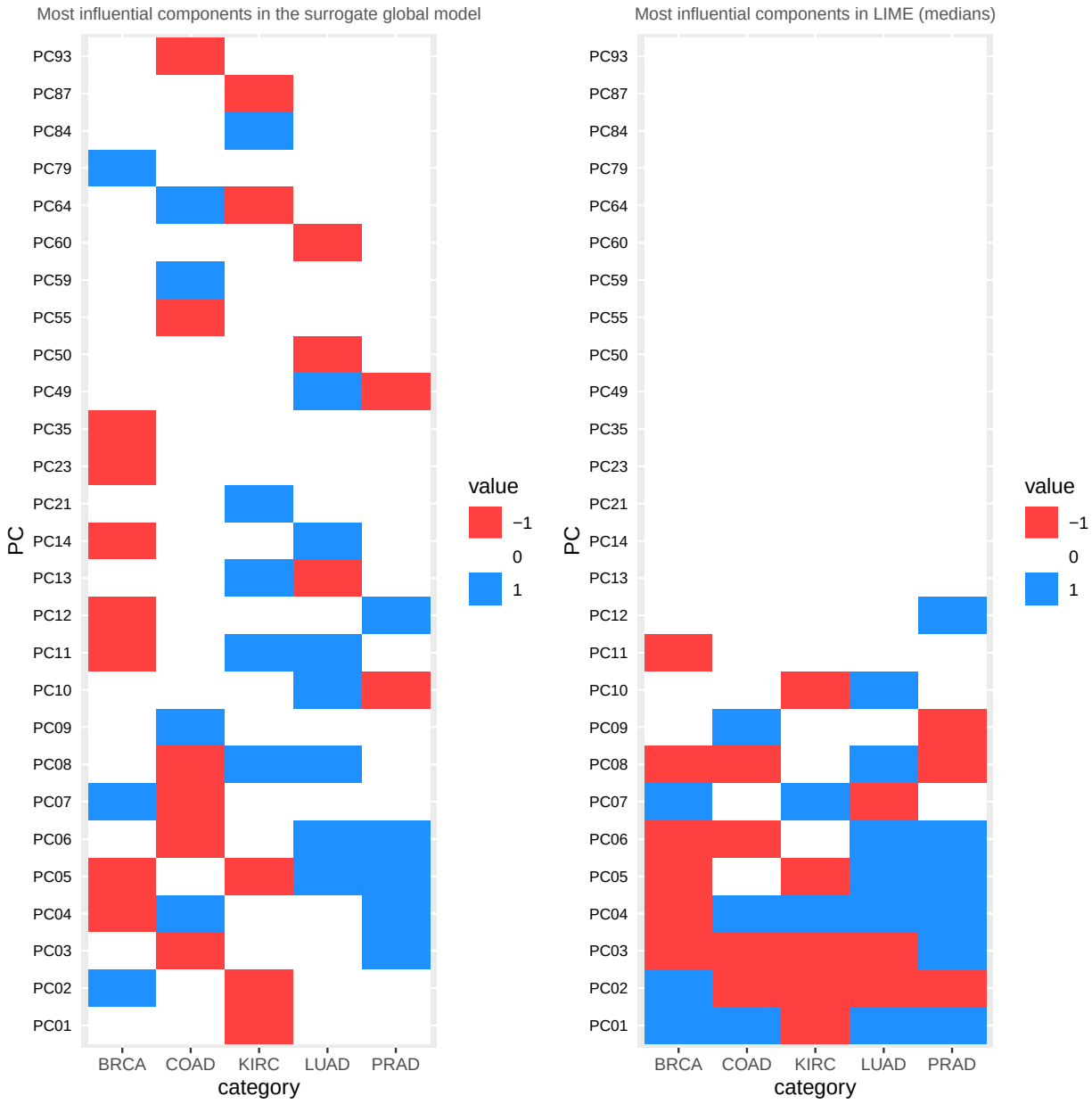
Since LIME interpretations from different observations for the same category will potentially include different components, we have to make a single selection of components summarizing all the LIME interpretations for the category, so we can compare it with the selection done by the global surrogate model. For each category, the selection will consist on the 16 components with highest sum of the absolute values of the coefficients from all the observations, those are the more influential components in aggregate.


Color Key
and Histogram
Count
0
20
60
0
0.4
0.8
Value
Sum of absolute values
of component coefficients (normalized)
PC01
PC02
PC03
PC04
PC05
PC06
PC07
PC08
PC09
PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
PC19
PC20
PC21
PC22
PC23
PC24
PC25
PC26
PC27
PC28
PC29
PC30
BRCA
COAD
KIRC
LUAD
PRAD

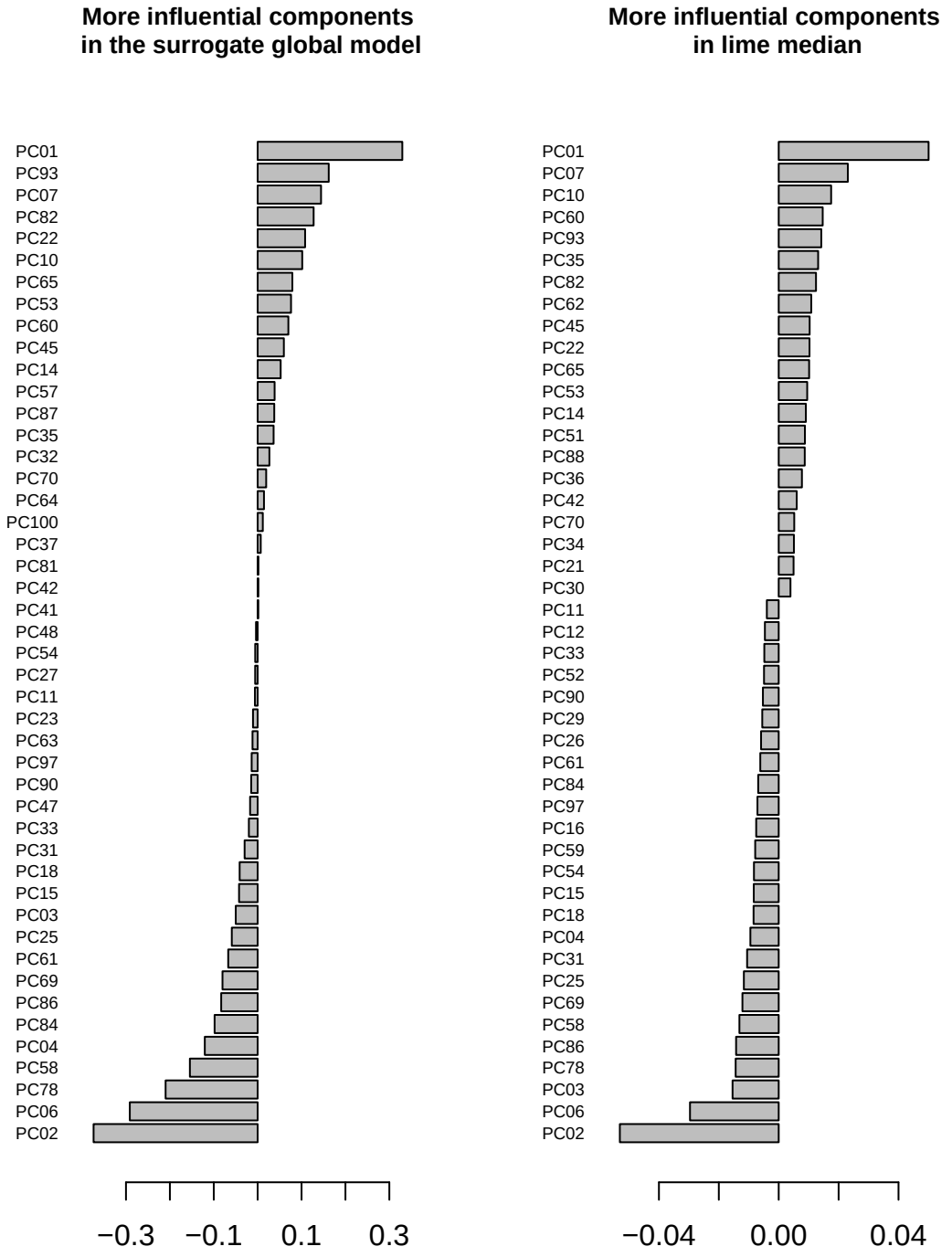
Below are displayed the distributions of the coefficients of the 16 more influential components for category LUAD. The vertical red line represents the mean, the blue one represents the median.




We can see that the more influential components are more or less symmetric, whereas the less influential ones are bimodal, one of the modes lying on value 0 which represents the absence of influence for a subset of observations. To avoid the components that only appear as influential for a few observations (and when they are included in the list of 10 more important components their value is very small) we use the median to compute the coefficients of the components for the whole category. The median will ignore components that come up rarely and have small values by setting them to 0 in the summarize component coefficient.



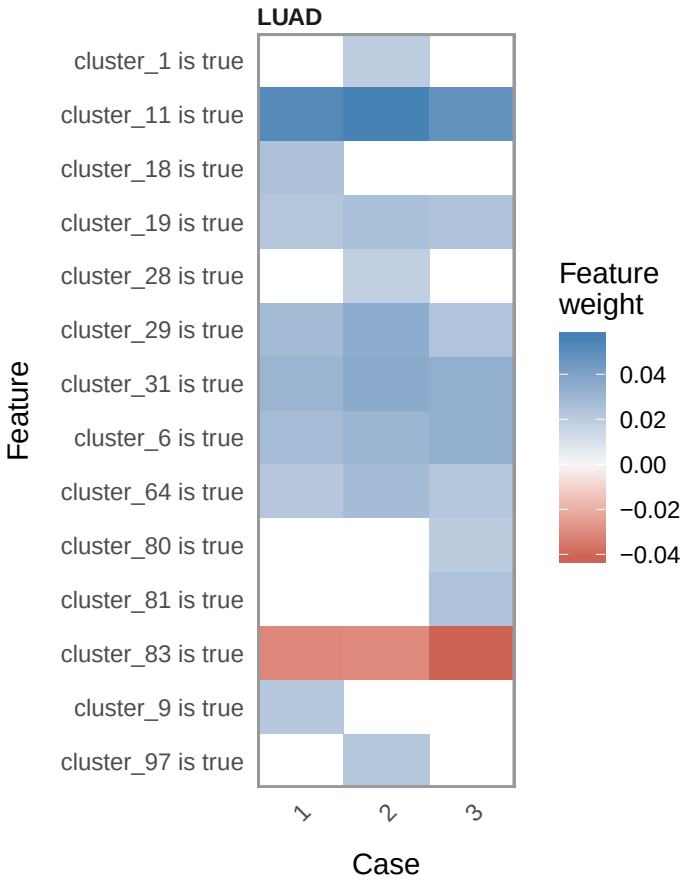

TODO: analizar y explicar el plot




TODO: analizar y explicar el plot

**Interpretation with feature clusters**

TODO: Idea: Usar clusters de variables que esten muy correlacionadas



**Interpretation with Sparce PCA**

TODO

## Conclusions

TODO

## References

TODO

[1] Recital 71 EU GDPR https://www.privacy-regulation.eu/en/recital-71-GDPR.htm

[2] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?: Explaining the Predictions of Any Classifier." In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 1135–44. ACM.

[3] On the Robustness of Interpretability Methods (Alvarez-Melis and Jaakkola 2018).

[4] Przemyslaw Biecek and Tomasz Burzykowski. 2020. "Explanatory Model Analysis: Explore, Explain and Examine Predictive Models." E-Book At< https://pbiecek.github.io/ema/>.

[5] Molnar, Christoph, and others. 2018. "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable." E-Book At< Https://Christophm.github.io/Interpretable-Ml-Book/>.

[6] Sparse Principal Component Analysis https://web.stanford.edu/~hastie/Papers/spc_jcgs.pdf

## Source code and data

TODO

Soruce code: https://github.com/codefluence/TFM

LIME Python code:
https://github.com/marcotcr/lime

Port in R:
https://github.com/thomasp85/lime

Genes Data Source: Samuele Fiorini, samuele.fiorini@dibris.unige.it, University of Genoa, redistributed under Creative Commons license.
measured by a sequencing platform (Illumina HiSeq).

Proteins Data Source: RAFP-Pred: Robust Prediction of Antifreeze Proteins using Localized Analysis of n-Peptide Compositions https://www.groundai.com/project/rafp-pred-robust-prediction-of-antifreeze-proteins-using-localized-analysis-of-n-peptide-compositions/1