A Tutorial to CP4P (Calibration Plot for Proteomics)

Quentin Giai Gianetto, Florence Combes, Claire Ramus,
Yohann Couté, Christophe Bruley, Thomas Burger

October 8, 2015

This supplemental document accompanies the article referred to as *Calibration Plot for Proteomics (CP4P): A graphical tool to visually check the assumptions underlying FDR control in quantitative experiments*, from Q. Giai Gianetto *et al.*. It is a tutorial to the R package CP4P (Calibration Plot for Proteomics).

1 Preliminary notions

1.1 Recap on p-values and on hypothesis testing

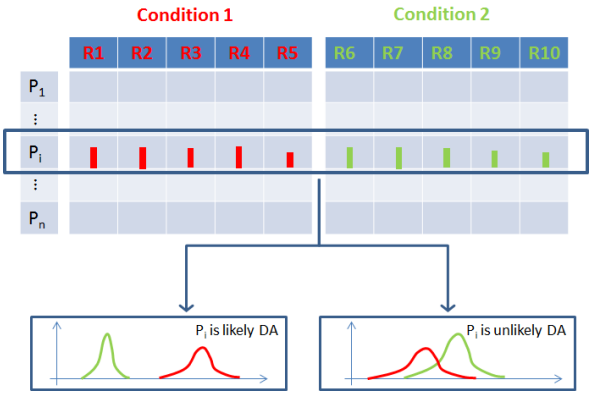

Figure 1: Graphical translation of testing protein $\mathbf{P}_i$, to know whether it is differentially abundant or not.

Having a list of m proteins $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_i, \ldots, \mathbf{P}_m\}$, one is interested in obtaining a list of few proteins that are significantly differentially abundant between two or more conditions. As depicted on Fig. 1, one relies on several abundances measured across several replicates for each condition, and a *null hypothesis significance test* (or simply, a *test*) is considered. To do so, a *score* (or a *statistic*), noted S, is defined to measure the extent to which the abundance values are different between conditions (the more different the conditions, the greater the score). Then, one compares the score S computed from observed abundances of $\mathbf{P}_i$, noted s_i, to the theoretical values it might have if there were no differences between conditions. Finally, the probability $p_i = \mathbb{P}(S \geq s_i | \mathbf{H}_0)$, which is classically known as the *p-value*, is estimated to quantify if s_i is in line with the theoretical distribution of the scores S when there is no differences. In this formula, $S \geq s_i$ refers to the fact that one wants to estimate the probability of observing a greater score than s_i while $\mathbf{H}_0$ refers to the fact that one estimates this probability under the so-called *null hypothesis*, which practically means one assumes $\mathbf{P}_i$ to be *non differentially abundant* (non-DA). In other words, p_i indicates the probability that the distributions of abundances for

$\mathbf{P}_i$ between the conditions look like what they indeed are, under the assumption that $\mathbf{P}_i$ is not DA. Naturally, if p_i is great, there is no reason to reject the idea that $\mathbf{P}_i$ is non-DA. On the other hand, if p_i is really small one naturally thinks there is little chance that $\mathbf{P}_i$ is not DA. As a result, the hypothesis testing does not directly tell us which proteins are *differentially abundant* (DA) or not; instead, it provides a list of m different p-values, that are related to a protein each, and that must be filtered.

1.2 Recap on multiple test correction and on FDR

In a dataset with several thousands of proteins, even if one applies a very stringent filter to the p-values, it is possible to expect that several non-DA proteins are wrongly considered as DA (such proteins are called *false discoveries*), just because they happen to have small p-values by chance. This is why it is necessary to apply a *multiple testing correction* (MTC) afterward.

There are numerous MTC, among which the most popular ones are the methods which control the *false discovery rate* (FDR). Introduced by Benjamini and Hochberg, FDR originally referred to the estimation of the expectation of the proportion of false discoveries in a list of putative discoveries, while now it often refers to a wider class of methods. However, to date, it is possible to summarize a general pattern common to the most used FDR control procedures: (i) compute p_i (the p-value of $\mathbf{P}_i$) ; (ii) reorder the protein list so that $p_{(1)}$ is the smallest p-value and $p_{(m)}$ the greatest; (iii) transform each $p_{(i)}$ into $p^*_{(i)}$ the so-called *adjusted p-value* (sometimes referred to as the *q-value*) which corresponds to the smallest FDR at which the corresponding protein will be concluded DA ; (iv) cut the list to $n \leq m$ so that $p^*_{(n)}$ corresponds to the desired FDR level.

However, most the procedure following this pattern are based on numerous mathematical hypotheses that must be respected to avoid a spurious FDR control. The package `CP4P` proposes simple graphical tools to control these assumptions, globally referred to as p-value calibration.

2 Before using `CP4P`

2.1 Obtaining the p-values

The first issue is to obtain the m p-values for $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_i, \ldots, \mathbf{P}_m\}$. To do so in an automated way, several pieces of software implement various statistical tests, among which the practitioner has to choose. These pieces of software can be split into two groups: Those that are devoted to proteomics, and which propose a biostatistics module pipelined to the output of the quantitation module. For instance, the `Maxquant` suite proposes the module `Perseus` to process quantitative datasets as the one depicted on Fig. 1. Alternatively, it is possible to use generic statistics software, such as `JMP` or `R`. If the m p-values are not directly computed with `R`, it will be necessary to import them in `R`, so as to run `CP4P`, as explained in Section 2.3. To do so, one advises to export the p-values from the software that produced them, in a CSV file. Most of the biostatistics modules propose such CSV exports, either of the entire quantitative dataset, or of the column of interest (which here contains the p-values).

2.2 Installing `CP4P`

The practitioner must have a recent version of `R` (3.2.0 or more recent[1]) installed on his/her workstation. Before installing `CP4P` itself, a number of packages must be installed from the website of the BioConductor project or from the CRAN. To do so, the simplest way is to copy/paste the following commands in the `R` console:

[1]The latest version of `R` is available at https://cran.r-project.org/bin/windows/base

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("limma")
> biocLite("qvalue")
> biocLite("multtest")
> install.packages("MESS")
```

Finally, CP4P can be installed with the following instruction:

```
> install.packages("cp4p")
```

Once CP4P has been installed, the package must be load in any new R session with the `library()` function; Similarly, its help documentation can be accessed with the `help()` function:

```
> library(cp4p)
> help(cp4p)
```

2.3 Importing the p-values

Now the package CP4P is operational. The CSV file containing either the p-values only or the entire dataset can be imported with the `read.table()` function, the numerous arguments of which are detailed in the R help. Basically, the following instruction is sufficient to import the data:

```
> data = read.table("C:/.../repository/data.csv",sep=";",header=TRUE)
```

where the first argument, `"C:/.../repository/data.csv"` refers to the full name of the CSV file (including its path; note that, contrarily to windows, "/" are used instead of "\"), the second argument refers to the symbol separating the columns (`sep="\t"` can be used if tabulations separate the columns); finally, the last argument indicates whether the first line of the file contains a header or not. Alternatively, it is possible to load one of the datasets accompanying the package:

```
> data(LFQRatio2)
```

or

```
> data(LFQRatio25)
```

It is possible to check the import did not go wrong, by displaying the beginning of the dataset:

```
> head(data)
```

where `data` has to be replaced by `LFQRatio2` or `LFQRatio25` if you use the datasets accompanying the package. If the dataframe does not only contain the p-values, it is more convenient to extract the column containing the p-values before applying any of functions from CP4P:

```
> p=data[,i]
```

where `i` is the index of the column containing the p-values. For instance, if you use the datasets accompanying the package, you can write `p=LFQRatio2[,7]` or `p=LFQRatio25[,7]`. From that point on, it is possible to apply the various functions of CP4P, as follow:

```
> estim.pi0(p,pi0.method="ALL")
> calibration.plot(p,pi0.method="ALL")
```

The adjusted p-values with a given method (let us say "slim") results from the following command:

```
> pv=adjust.p(p,pi0.method="slim")
```

The result of this function contains 2 columns, that diplay respectively, the original p-values:

```
> pv$adjp[,1]
```

and the adjusted ones:

```
> pv$adjp[,2]
```

3 Interpretation of the calibration plot

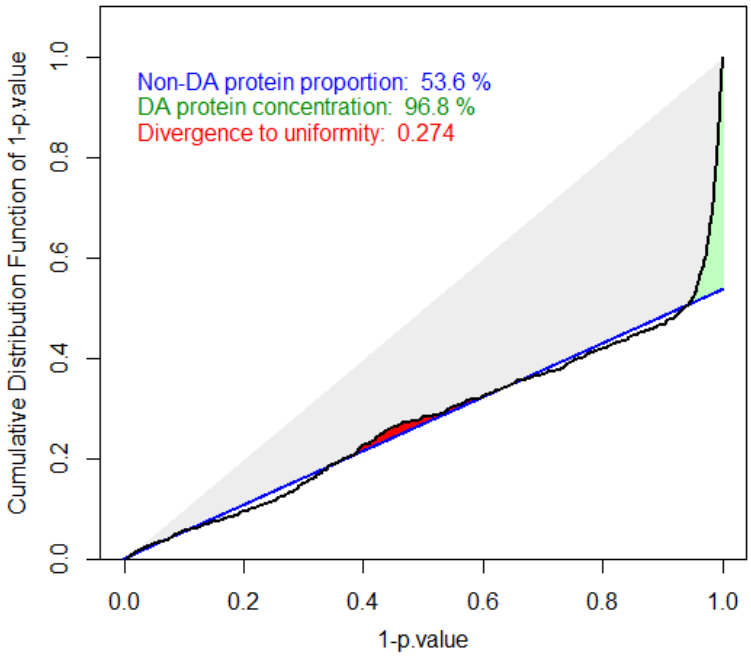

Figure 2: Typical graphical output of the `calibration.plot()` function on a dataset with well-calibrated p-values. The black curve displays the cumulative distribution function of $1-p_i$ ($i \in [1,m]$) as a function of $1-p_i$. The blue line helps visualizing π_0 (the proportion of non-DA proteins) as its equation reads $y = \pi_0 x$. The A area between the right hand side peak of the black curve and the blue line is colored in green: it depicts the extent to which the set of DA proteins have different p-values than other proteins, and consequently, the extent to which they can be discriminated on the basis of a good FDR threshold. The *DA protein concentration* measure reads $1 - A/T$, where T is the gray triangle area (by construction, $T = (1-\pi_0)/2$). The *uniformity underestimation* (in red) corresponds to the area where the black curve is above the blue line apart from the peak at the left hand side (DA protein peak).

Basically, the function `calibration.plot()` of the CP4P package takes as input a vector of p-values that have been previously computed at the hypothesis testing step. As output, it provides a graph similar to Fig. 2, which displays (black curve) the cumulative distribution function of $1-p_i$ ($i \in [1,m]$) as a function of $1-p_i$. As it clearly appears, the curve starts from point [0,0], and is then roughly linear indicating that the non-DA proteins have p-values that are

roughly uniformly distributed. On the other hand, the curve becomes very peaky nearby the $[0.9, 1]$ interval, indicating that there is an important concentration of small p-values, most likely corresponding to DA proteins.

In addition, a blue line is displayed on the graphic. It is expected to have the same trend as the linear part of the black curve, as illustrated on Fig. 2. The slope of this blue line corresponds to an estimation of the proportion of non-DA proteins (classically noted π_0), which is indicated as *non-DA protein proportion* (in blue too). Concretely, its equation reads $y = \pi_0 x$. Note that this gets theoretical justifications. Indeed, the cumulative distribution function of the 1-p-values (denoted $F_{1-p}(x) = P(1-p \leq x)$) can be decomposed in two distributions, one associated to the non-DA proteins (denoted $F_{1-p|non-DA}(x)$) and another associated to the DA proteins (denoted $F_{1-p|DA}(x)$), i.e.:

$$F_{1-p}(x) = \pi_0 \times F_{1-p|non-DA}(x) + (1-\pi_0) \times F_{1-p|DA}(x)$$

When p-values are quite high (1-p-values quite low), it is expected that they are only associated to non-DA proteins and so that $F_{1-p}(x) = \pi_0 \times F_{1-p|non-DA}(x)$. If p-values associated to non-DA protein are uniformly distributed between 0 and 1, we so get $F_{1-p}(x) = \pi_0 \times (1-p)$ which gives the equation of the blue line.

The area A between the right hand side peak of the black curve and the blue line is colored in green. This area is important: it depicts the extent to which the set of DA proteins have different p-values than other proteins, and consequently, the extent to which they can be discriminated on the basis of a good FDR threshold. The thinner this area, the better, as it amounts to having DA proteins with p-values distinctly smaller than the others. In order to propose a quantitative estimation of the quality of the distribution of the p-values in relationship with this green area, we derived the *DA protein concentration* measure that reads $1-A/T$, where T is the gray triangle area (by construction, $T = (1-\pi_0)/2$). This concentration is written in green in the top left corner of the calibration plot, and intuitively, the closer to 100% it is, the better. Note that this quantity gets also theoretical justifications. Indeed, if we keep the same notations as before, the area A can be expressed as $A \approx \int_0^1 (F_{1-p}(x) - \pi_0 F_{1-p|non-DA}(x))dx = (1-\pi_0) \times \int_0^1 F_{1-p|DA}(x)dx$. As a result, the *DA protein concentration* is an approximation of $1 - 2\int_0^1 F_{1-p|DA}(x)dx = 2\int_0^1 F_{p|DA}(x)dx - 1$. Because $\int_0^1 F_{p|DA}(x)dx$ is close to 1 when the p-values associated to DA proteins are distributed nearby 0, the *DA protein concentration* is expected close to 1.

Finally, the *uniformity underestimation* (in red) corresponds to the area where the black curve is above the blue line apart from the peak at the left hand side (DA protein peak). In order to get a conservative adjustment of the p-values (so that it does not under-estimate the FDR), the black curve has to remain below the blue line. Indeed, if $p_{(1)} \leq ... \leq p_{(m)}$ is the ordered sequence of m available p-values, the traditional Benjamini-Hochberg procedure searches for the largest k such that $p_{(k)} \leq (k/m)\alpha$ (α being the desired FDR level) and will lead to keep all proteins for which this inequality is verified. In such a framework, several authors have shown that

$$FDR = \pi_0 \alpha \leq \alpha$$

An adaptive FDR procedure searching for the largest k such that $p_{(k)} \leq (k/m)\alpha/\hat{\pi}_0$ where $\hat{\pi}_0$ is an estimate of π_0 will imply

$$FDR = \pi_0 \alpha/\hat{\pi}_0 \approx \alpha$$

Note that the greater π_0 will be, the more stringent the FDR procedure will be ($(k/m)\alpha/\hat{\pi}_0$ being lower). Thus, the true FDR is expected closer to the desired FDR level if π_0 is adequately estimated. An overestimation of π_0 will lead to conservative p-value adjustments (in such a case the true FDR will be inferior to the desired one). However, a major problem can occur if π_0 is underestimated since the true FDR can next be superior to the desired one (such as it is no more controlled). The *uniformity underestimation* quantity allows to underline this

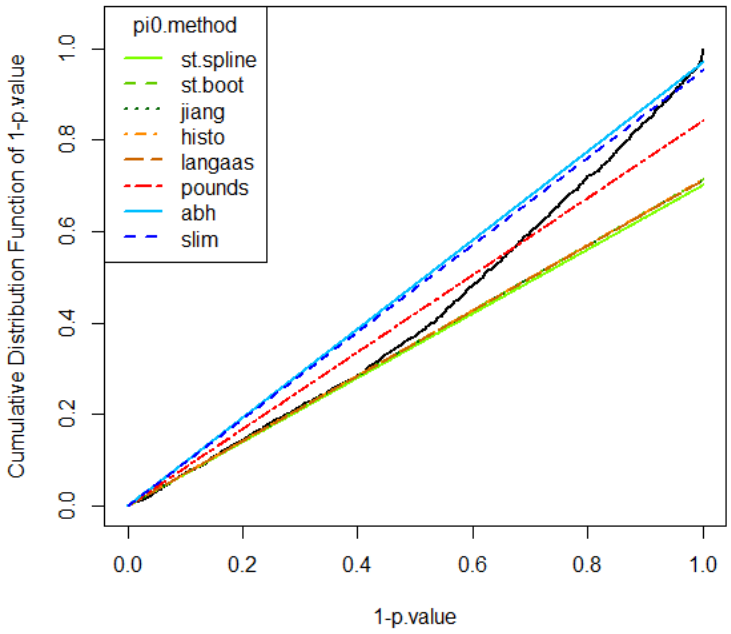

Figure 3: Illustration of the "ALL" optional argument on a real dataset where the default π_0 estimator is inaccurate.

problem. In the ideal case, the left hand side of the black curve always remains below the blue one and the uniformity underestimation is null. However, in practice, as long as the uniformity underestimation remains small (below a guesstimate of 0.5), a procedure to adjust the p-values can be safely used.

As explained, the blue line is of prime importance for the overall visual assessment of the p-value distribution. However, its slope, reflecting π_0 needs to be estimated since it is unknown. However, as any estimator, it is possible to exhibit situations where the default one is inaccurate. For this reason, one may want to use other state-of-the-art estimation methods instead. Concretely, this is implemented in `calibration.plot()` with a second optional argument, which can take several values:

- A value x between 0 and 1, which corresponds to the freely tuned proportion of non-DA proteins, for cases where the practitioner knows the precise content of the sample.
- The name of an estimation method among: "pounds" (default tuning), "st.boot", "st.spline", "langaas", "jiang", "histo", "abh", "slim".
- "ALL": A different line for the eight aforementioned methods is displayed so that the practitioner chooses on his/her own, the most adapted one.

4 Subsequent processing

4.1 Exporting the adjusted p-values

Once the p-values are adjusted, the practitioner can either finish his/her analysis within `R` (see next section), or export them in a CSV file, so as to go with another software, that he/she is confortable with. The `R` command to export the p-values is the following:

```
> write.table(pv$adjp,"C:/.../repository/pv.csv",sep=";")
```

where `pv.csv` is the name of the output file. In addition, it is possible to export any figure produced by `CP4P` by simply saving it into the desired format.

4.2 Finalizing the analysis

The post-processing of the adjusted p-values is extremely simple, and is made of the following steps: (1) order the adjusted p-value from the smallest one to the greatest one, (2) cut this list at some given threshold T, so that the proteins with an adjusted p-value above T are said to be DA with FDR=T. These are simple operations that can be done in `R`, or for those who are not comfortable with programming, with software dedicated to proteomics (such as `Perseus`), or even in `MS Excel`.

5 Supplemental illustrations

5.1 Extreme cases (simulations)

In the extreme case where all the proteins are DA, the blue line as well as the left hand side of the black curve are supposed to follow the abscissa axis, up to the starting of the right hand peak which depicts the small p-values of the DA proteins, such as illustrated on Fig. 4 (left). In the other extreme case where all the proteins are non-DA, both the blue line and the black curve follow the diagonal line, depicted by the upper edge of the grey triangle, such as illustrated on Fig. 4 (right).

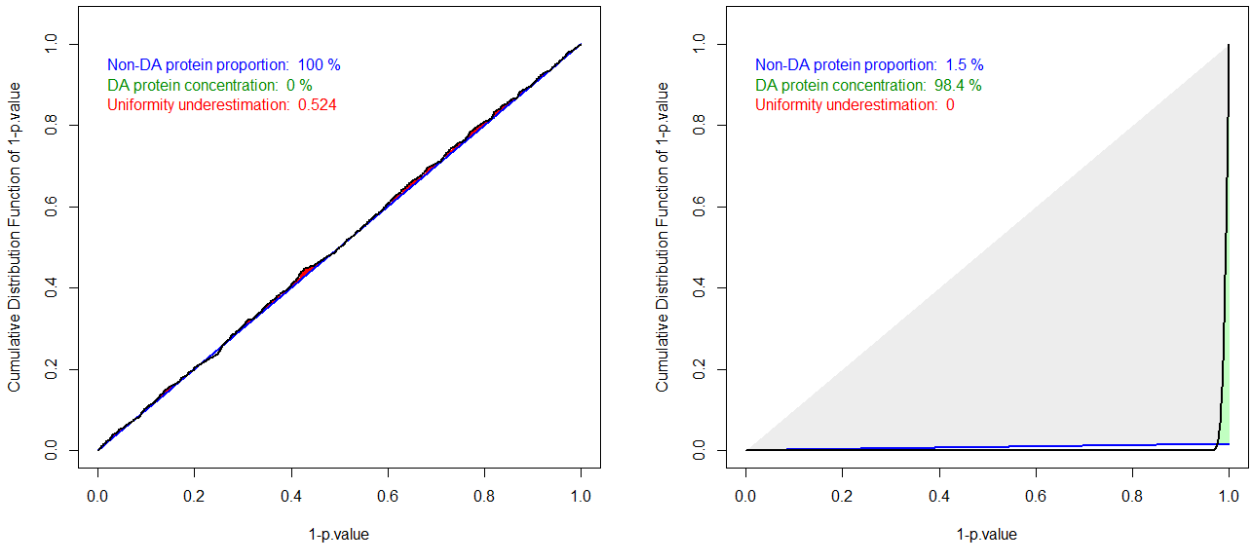

Figure 4: First extreme case where all the proteins are DA (left). Second extreme case where all the proteins are non-DA (rigth).

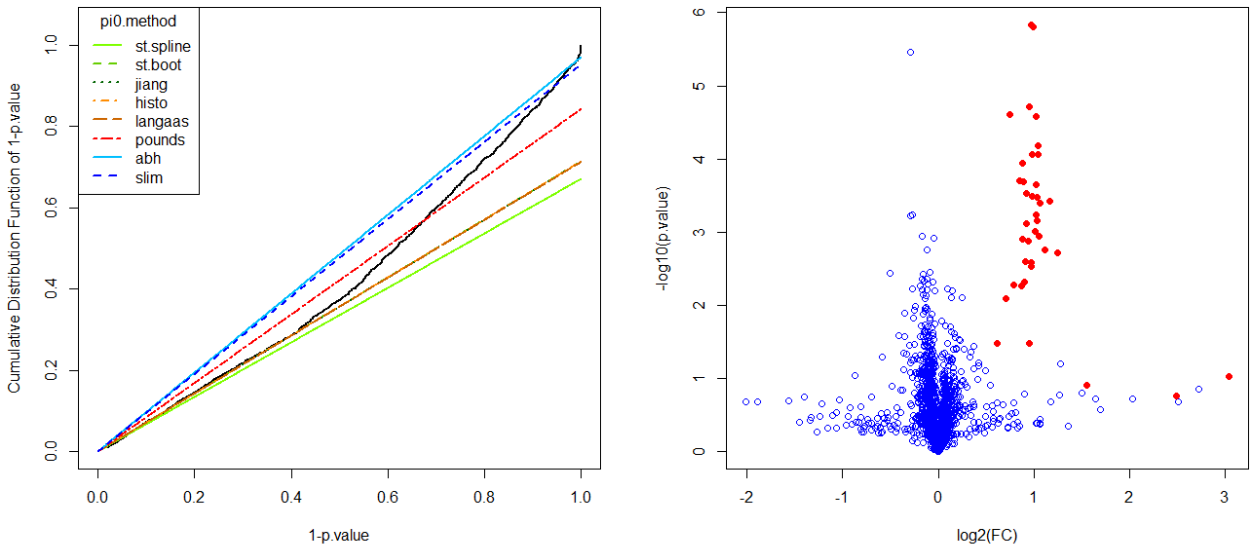

Figure 5: Comparison of all the π_0 estimators (left). The corresponding volcano plot where UPS1 proteins are represented in red (right).

5.2 Detailed illustrations of the LFQRatio2 dataset

This dataset contains UPS1 human proteins that are the only differentially abundant proteins within the conditions (condition 1: 10fmol of injected UPS1 human proteins, and condition 2 5fmol of injected UPS1 human proteins, leading to a ratio that equates to 2) within a yeast background, so that it is possible to trace back which are the DA and non-DA proteins. As the number of DA proteins is known to be small, the π_0 estimate must be as close to one as possible. On Fig. 5 (left), the DA proteins corresponds to the top-post-and right-most vertical part of the black curve.

For better understanding, the behavior of the 8 different estimators are illustrated on Fig. 6, as well as in Tab. 1. Note that on this dataset, "histo" and "st.boot" provide equal estimators.

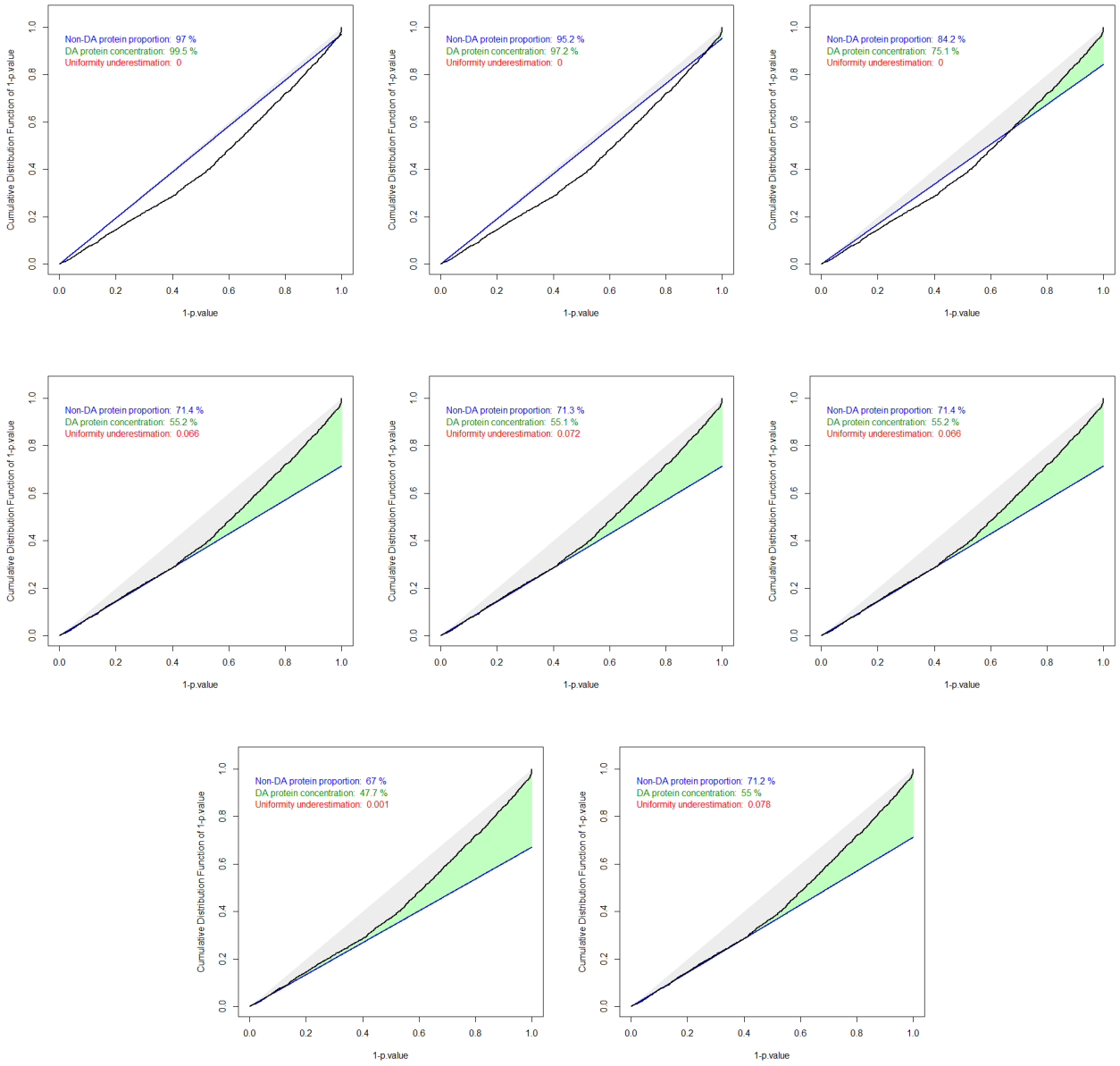

Figure 6: Other calibration plots, with different π_0 estimators, from left to right, from top to bottom: "abh", "slim", "pounds", "histo", "jiang", "st.boot", "st.spline" and "langass".

5.3 Detailed illustrations of the LFQRatio25 dataset

This dataset is similar to the previous one, except for the differences in PS1 concentrations: 25fmol of injected UPS1 human proteins for condition 1, and 10fmol for condition 2, leading to a ratio of 2.5. Fig. 7 and 8 are the counterparts to Fig. 5 and 6 for this dataset.

					FDR control at 5%	
		Estimated π_0	*Uniformity Underestimation*	*DA Protein Concentration*	*Real False Discovery Proportion*	*Proportion of UPS1 proteins*
π_0 estimation method	st.spline	0.6706	0.0017	47.75%	17.24%	24/29
	langaas	0.7122	0.0787	55.06%	17.24%	24/29
	jiang	0.7131	0.0724	55.18%	17.24%	24/29
	st.boot	0.7140	0.0662	55.29%	17.24%	24/29
	histo	0.7140	0.0662	55.29%	17.24%	24/29
	pounds	0.8425	0.0002	75.17%	13.04%	20/23
	slim	0.9524	0.0001	97.25%	13.04%	20/23
	abh	0.9703	0.0001	99.56%	13.04%	20/23
	bh	1	0	-	13.04%	20/23

Table 1: Summary of the various π_0 estimator in relation with various FDR thresholds. As a reference value, the standard Benjamini-Hochberg (bh) procedure, where π_0 is assumed to equate 1, was also considered.

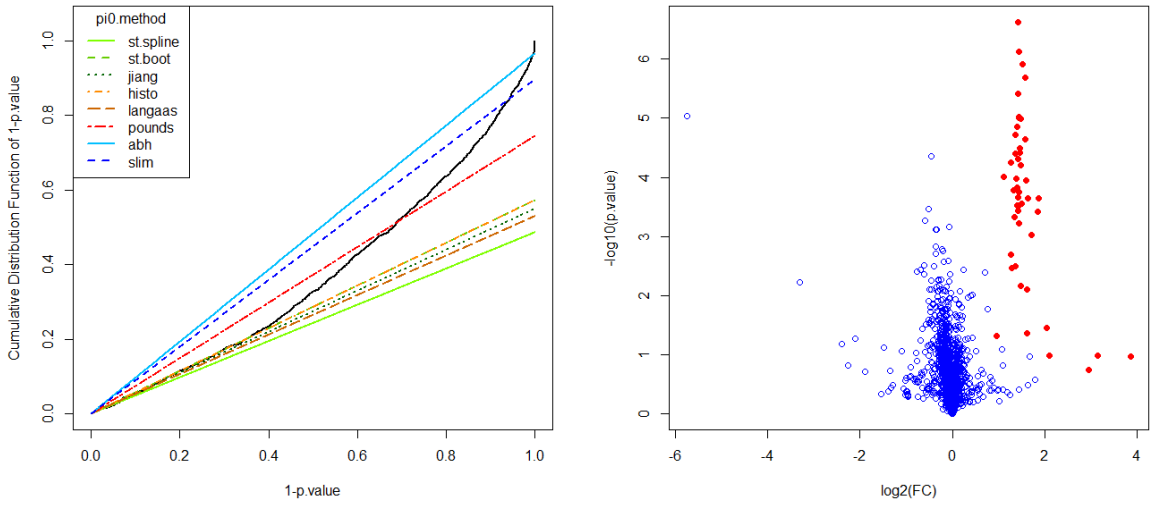

Figure 7: Comparison of all the π_0 estimators (left). The corresponding volcano plot where UPS1 proteins are represented in red (right).

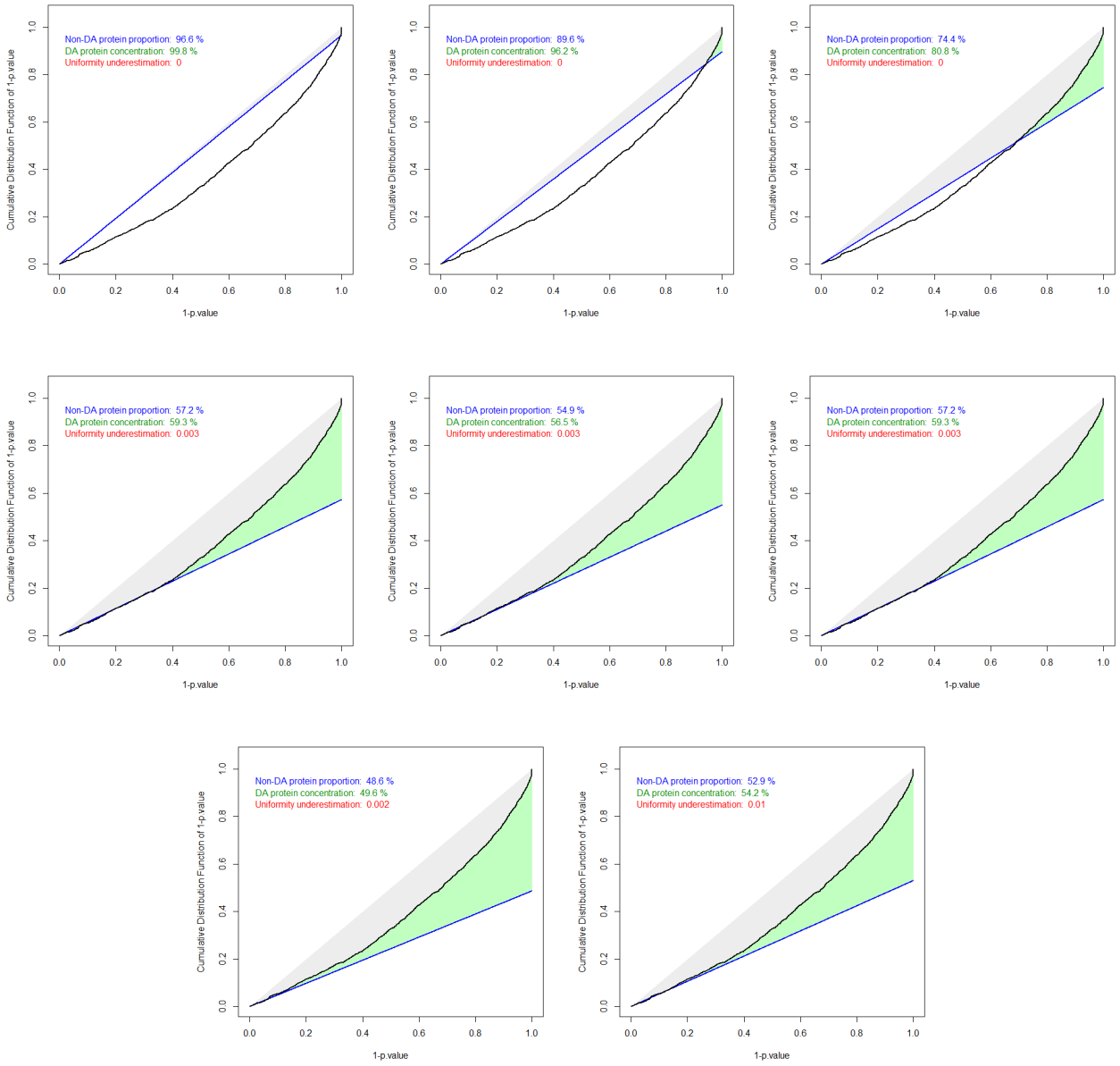

Figure 8: Other calibration plots, with different π_0 estimators, from left to right, from top to bottom: "abh", "slim", "pounds", "histo", "jiang", "st.boot", "st.spline" and "langass".

Finally, Tab. 2 summarizes the associated important values. Note that on this dataset too, "histo" and "st.boot" provide equal estimators. Moreover, in this table, no thresholding on the fold-change was considered prior to the FDR computation, yet such thresholding is of course possible to reduce the number of selected proteins.

5.4 iSa dataset

This is the companion dataset to the following article: Bounab Y, Hesse A-M-, Iannascoli B, et al. *Proteomic Analysis of the SH2Domain-containing Leukocyte Protein of 76 kDa (SLP76) Interactome.* Molecular & Cellular Proteomics. 2013;12(10):2874-2889. doi:10.1074/mcp.M112.025908. The various calibration plots can be found on Fig. 9 and 10: Fig. 9 (left) presents the various π_0 estimators. As a matter of fact, this dataset does not present any underestimation of the uniform distribution for the non-DA. However, as with the UPS1 datasets, the DA concentration is too low if one uses the "langaas", "histo", jiang" or "st.-" estimators. On the other hand "abh" appears as more conservative than necessary, so that "pounds" or "slim" should be promoted.

					FDR control at 5%	
		Estimated π_0	*Uniformity Underestimation*	*DA Protein Concentration*	*Real False Discovery Proportion*	*Proportion of UPS1 proteins*
π_0 estimation method	st.spline	0.4863	0.0022	49.69%	31.5%	37/54
	langaas	0.5298	0.0101	54.28%	29.41%	36/51
	jiang	0.5498	0.0039	56.59%	27.08%	35/48
	st.boot	0.5726	0.0039	59.38%	27.08%	35/48
	histo	0.5726	0.0039	59.38%	27.08%	35/48
	pounds	0.7447	0	80.83%	23.91%	35/46
	slim	0.8964	0	96.25%	20.93%	34/43
	abh	0.9660	0	99.80%	17.07%	34/41
	bh	1	0	-	17.07%	34/41

Table 2: Summary of the various π_0 estimator in relation with various FDR thresholds. As a reference value, the standard Benjamini-Hochberg (bh) procedure, where π_0 is assumed to equate 1, was also considered.

The latter being more conservative, one selects it, as displayed on Fig. 9 (right). The calibration plots with all the other estimators are displayed on Fig. 10.

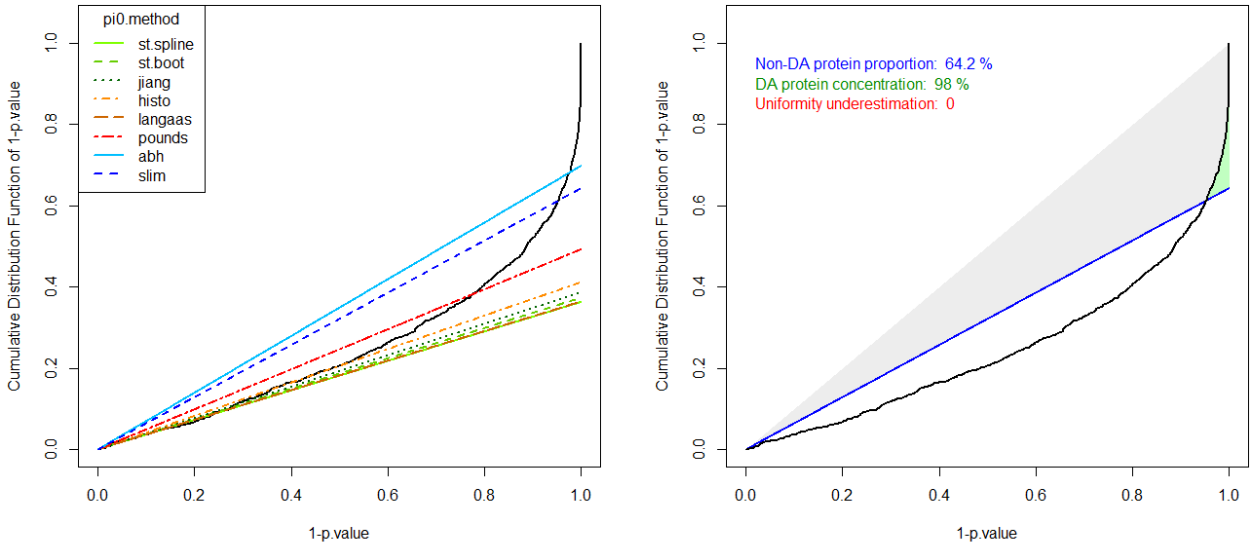

Figure 9: Comparison of all the π_0 estimators (left). The "slim" estimator appears the most adapted one (right).

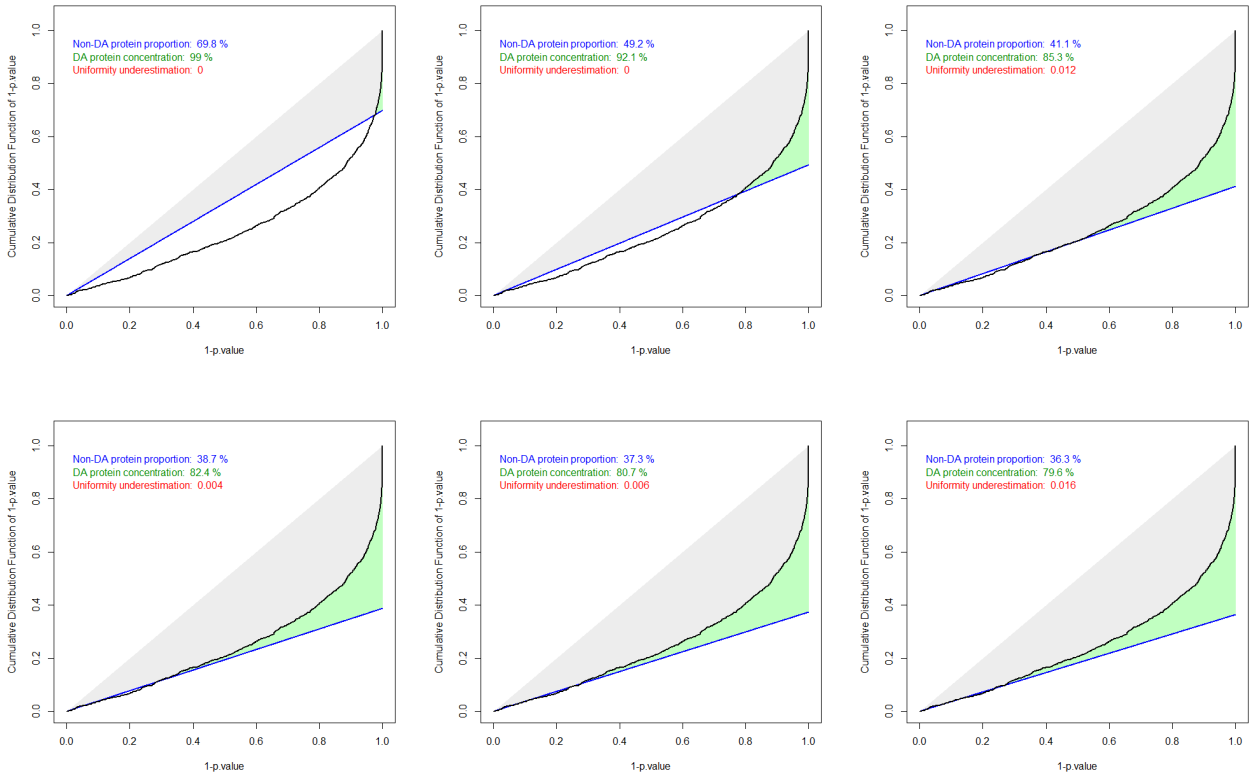

Figure 10: Other calibration plots, with different π_0 estimators, from left to right, from top to bottom: "abh", "pounds", "histo", "jiang", "st.boot", and on the last graphics "langass" or "st.spline", as these last two estimators provides similar plots.

Finally, whatever the FDR threshold, the number of selected proteins is more or less important, depending on the chosen π_0 estimator, as illustrated on Tab. 3. Let us note, that in this table, no thresholding on the fold-change was considered prior to the FDR computation, yet such thresholding is of course possible to reduce the number of selected proteins.

π_0 estimation method		*Estimated π_0*	*Uniformity Underestimation*	*DA Protein Concentration*	*Number of selected proteins (FDR 1%)*	*Number of selected proteins (FDR 5%)*	*Number of selected proteins (FDR 10%)*
	st.spline	0.3625	0.0178	79.55%	304	552	739
	langaas	0.3636	0.0163	79.67%	304	552	738
	st.boot	0.3731	0.0069	80.77%	300	544	733
	jiang	0.3871	0.0047	82.44%	290	538	723
	histo	0.4112	0.0128	85.37%	277	521	711
	pounds	0.4924	0	92.15%	262	479	645
	slim	0.6426	0	98.08%	229	432	579
	abh	0.6980	0	99.02%	225	428	564
	bh	1	0	-	194	371	476

Table 3: Summary of the various π_0 estimator in relation with various FDR thresholds. As a reference value, the standard Benjamini-Hochberg (bh) procedure, where π_0 is assumed to equate 1, was also considered.