Supplemental material to:

Five simple yet essential steps to correctly estimate the rate of false differentially abundant proteins in mass spectrometry analyses

Samuel Wieczorek[1], Quentin Giai Gianetto[2,3], Thomas Burger[1,4] *

[1]Univ. Grenoble Alpes, CEA, INSERM, BIG-BGE, 38000 Grenoble, France
[2]Bioinformatics and Biostatistics Hub, C3BI, Institut Pasteur, USR 3756 IP CNRS, 75015 Paris, France
[3]Proteomics platform, Mass Spectrometry for Biology Unit, Institut Pasteur, USR 2000 IP CNRS, 75015 Paris, France
[4]CNRS, BIG-BGE, F-38000 Grenoble, France
*thomas.burger@cea.fr

1. Log fold change cutoff tuning and consequences

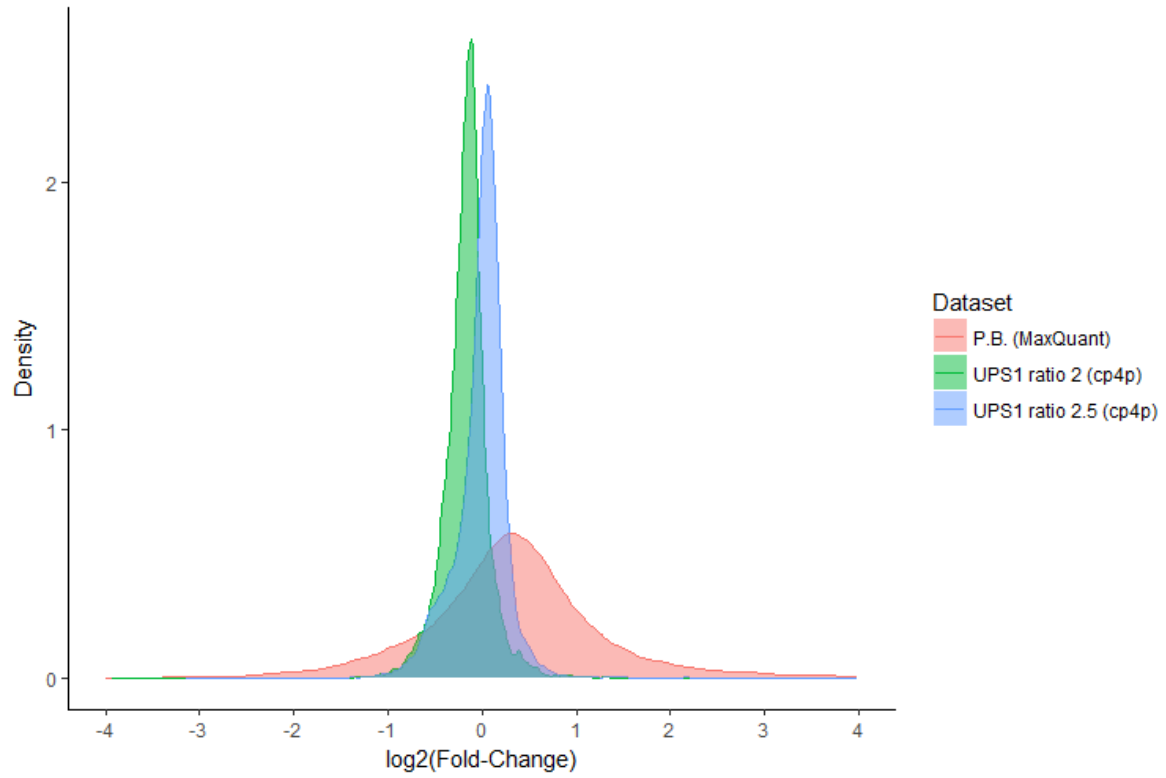

Figure 1: Distributions of logFCs of peptides belonging to proteins having the same concentration between two conditions from three benchmark datasets.

Fig. 1 illustrates why filtering on the logFC is often tempting and sometimes necessary. One considers the logFC distribution of the raw peptide intensities (with no missing values, and only from proteins that are known to be equally concentrated in the compared conditions) of several benchmark datasets from [1] and [2]. While all the logFC should be equal to zero, one practically observes that a varying but significant proportion of peptides have different logFCs. These distributions are generally centered on a value that is close to zero, yet, their variance are rather different from one dataset to another,

due to difference of biological complexity of the sample, of sample preparation and MS analysis reproducibility. With this regards, filtering out particularly low logFC may be helpful. However, doing so does not require displaying a volcano plot, as the logFC distribution should be sufficient to find an adapted threshold. This is why, in latest Prostar releases, the logFC thresholds of all the pairwise comparisons are tuned to a same value, according to what the practitioner reads on the superimposed distributions (see Fig. 2).

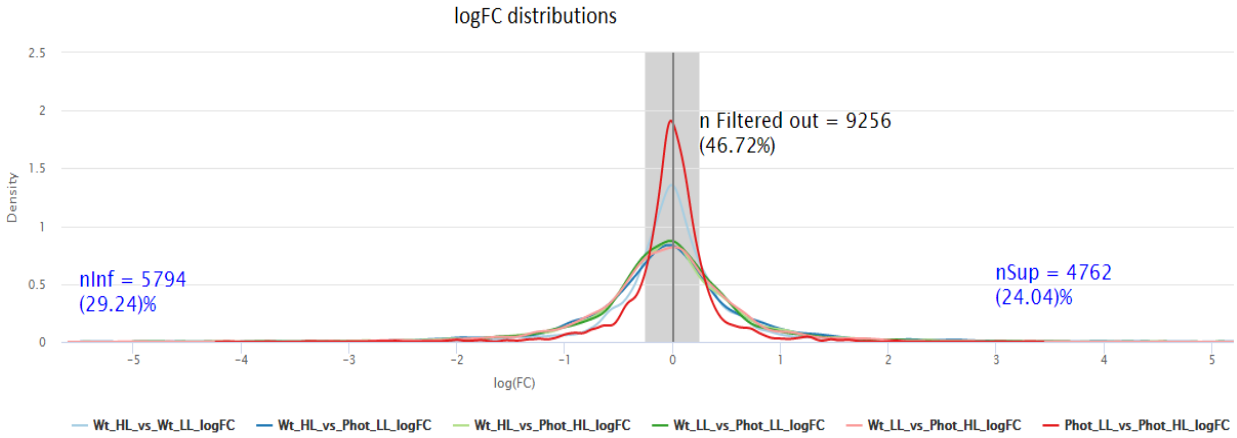

Figure 2: Superimposition of six logFC distributions resulting from a lab dataset (unpublished so far) containing 4 different biological conditions. All the logFC distributions are centered on zero, making it possible to tune the overall logFC threshold to a fairly small value.

Let us note that if some proteins have been filtered out with a high logFC cutoff, we can end up with selected proteins only associated with low p-values. This is not inherently a problem for most of the subsequent processing steps, apart from the estimation of the proportion of non-DA proteins (termed π_0 in the penultimate section of the main article). In fact, most of the π_0 estimation methods are based on the assumption that all the p-values are distributed between 0 and 1. Thus working on a dataset where too many proteins where filtered out due to their logFC may compromise the validity of this assumption and thus lead to inaccurate estimate (and consequently spurious FDR).

Fortunately, there is a simple mathematical trick to recover unbiased π_0 estimates if the p-values are distributed between 0 and a value $p_{max} < 1$: it consists in dividing all the p-values by p_{max} and then applying the methods of the literature. Afterwards, all the downstream FDR analysis can be conducted on the original p-values and with the π_0 estimates obtained on "rescaled" p-values. From a practical viewpoint, it is either necessary to explicitly apply this tricks by coding the corresponding computations, or to rely on a software tools or package which incorporates it, such as for instance CP4P [2], which is directly called from Prostar interface [3].

2. Well-calibrated distribution examples

Fig 3. of the main article depicts a nearly optimal calibration plot, where roughly half of the dataset proteins are non-DA. This figure comes from [2] and was built with a simulated dataset, so as to illustrate a "perfect" scenario. The interest of simulated datasets is too precisely describe how small modifications in the distributions derives into calibration changes. The following figures represents various calibration plots from simulated datasets where only the proportions of non-DA and of DA proteins are changed with respect to that of [2]. To do so, one uses the following R code:

```
> library(cp4p)
> n <- 1000
> pi0 <- 0.5
> pval1 <- rbeta(n*(1-pi0),shape1=0.5, shape2=20)
```

```
> pval2 <- runif(n*pi0)
> pval <- c(pval1,pval2)
> calibration.plot(pval)
> calibration.plot(pval, "ALL")
```

which provides Fig. 3. On the left panel, one has a plot akin to Fig 3 of the main article (up to stochastic variations resulting from different simulation runs), while on the right panel, one observes that the various π_0 estimates concur, which is a sign of correct calibration.

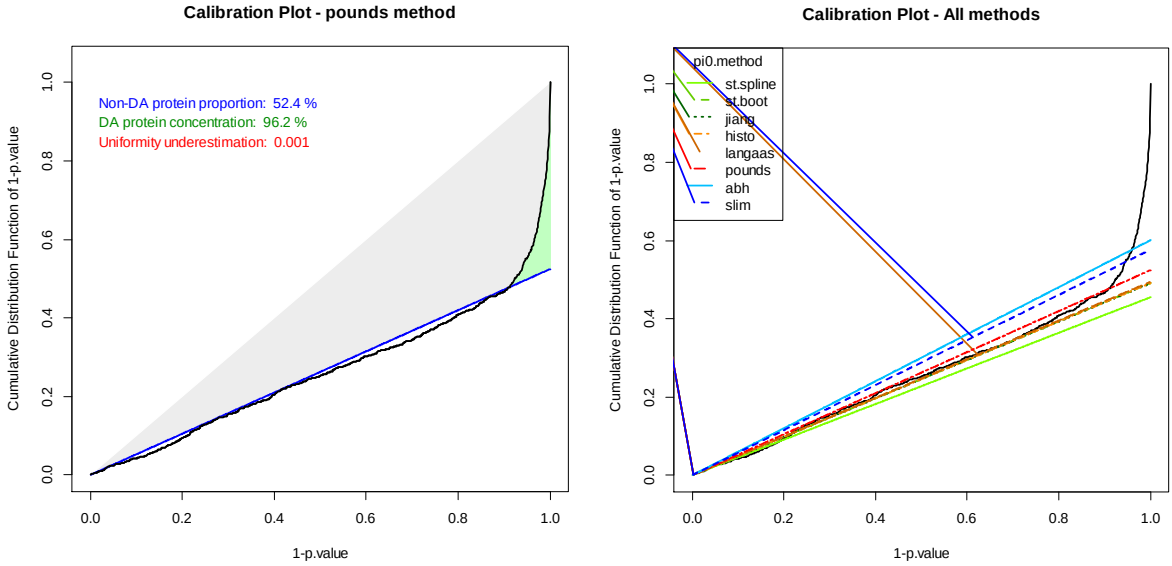

Figure 3: Calibration plots resulting from the code above: The left panel depicts the (default) Pound estimator, while the right panel shows the convergence of the various estimating methods.

In the code above, if one changes π_0 from 0.5 to 0.25 or to 0.75, one obtains the calibration plots of Fig. 4.

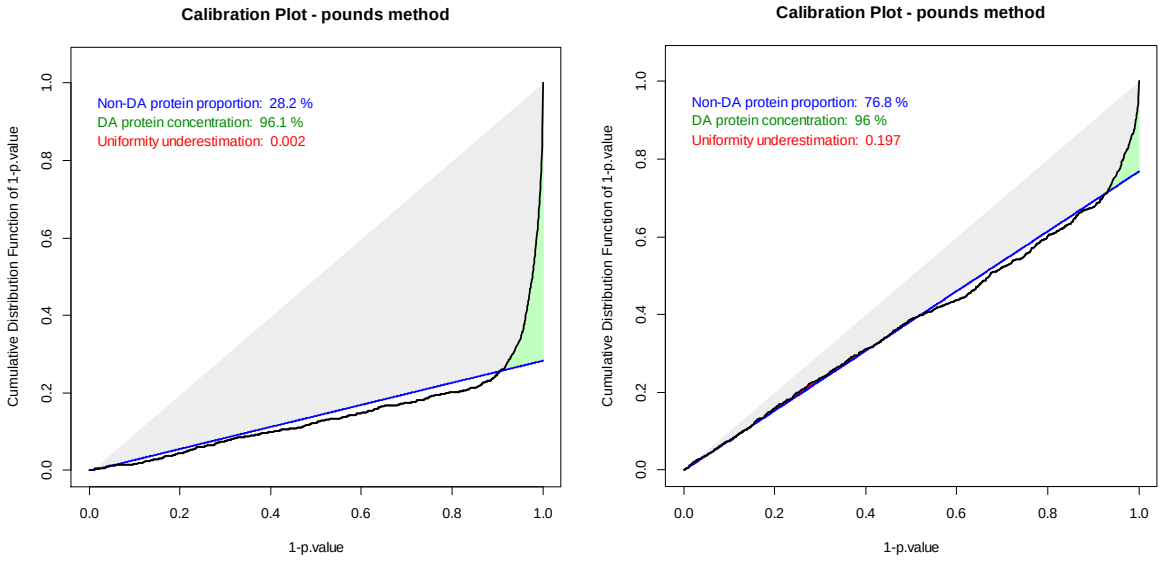

Figure 4: Calibration plots with $\pi_0 = 0.25$ (left panel) and with $\pi_0 = 0.75$ (right panel).

However, if one changes π_0 to some very large proportion, as on Fig. 5, one obtains a calibration plot that may look similar to an ill-calibrated one (as for instance Fig. 11, below), due to the absence of sharp angle at the basis of the green region, which results from the small proportion of DA proteins. In this simulated case, the uniform distribution of the non-DA protein is perfect, so that the various

estimate concurs and it is easy to assess the good calibration. However, in case of a small proportion of non-DA proteins, if the uniformity of non-DA protein is not perfect, the discrimination between a good calibration (as on fig. 5) and a not as good one (Fig. 11) may become difficult.

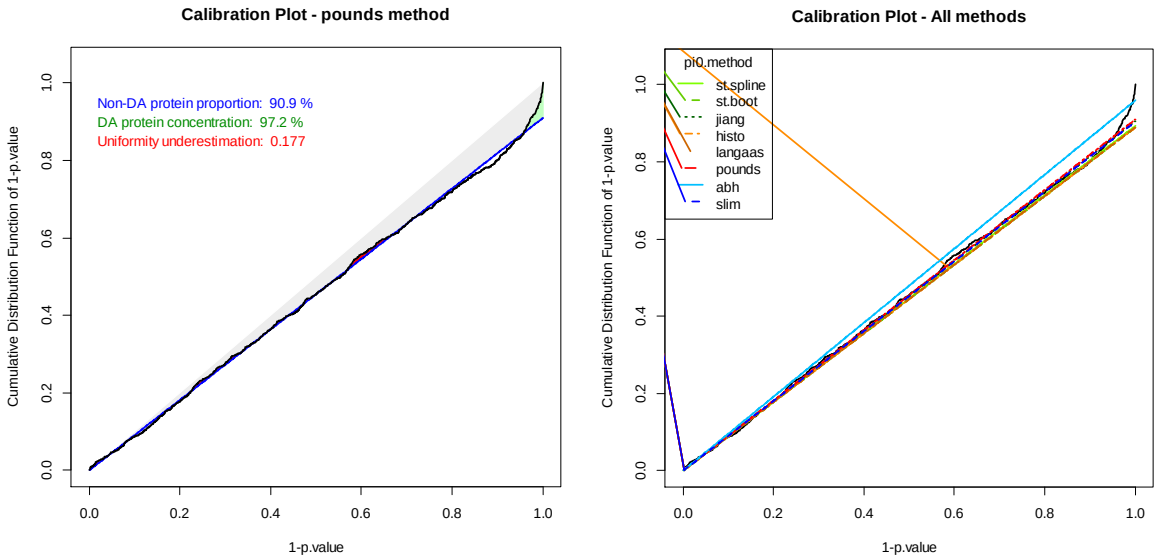

Figure 5: Calibration plots with $\pi_0 = 0.9$

3. Ill-calibrated distribution examples

Now, let us reduce the total amount of tested proteins (from n=1000 to n=100). Although the distributions are in line with the theory, the corresponding calibration plot on Fig. 6 is not satisfying, for the curve displays steps. If $\pi_0 = 0.5$, the steps are thin and evenly distributed on the curve (Fig. 6, left panel), while if If $\pi_0 = 0.1$ (Fig. 6, right panel), the steps are wider but concentrated on the lower left side of the curve. Let us also note that for such stepped curve, the color display may be erratic, for the tool is pushed to the limit it was made for. All this should be a sign that despite a theoretically correct distribution, and thus a theoretically good calibration, the FDR will be unstable. This is why, we advise to consider this type of graph as ill-calibrated rather than well-calibrated. Concretely, the FDR can be stabilized (to the price of a small over-estimation) by selecting an estimation method which provides a larger value for π_0 (see Fig. 7).

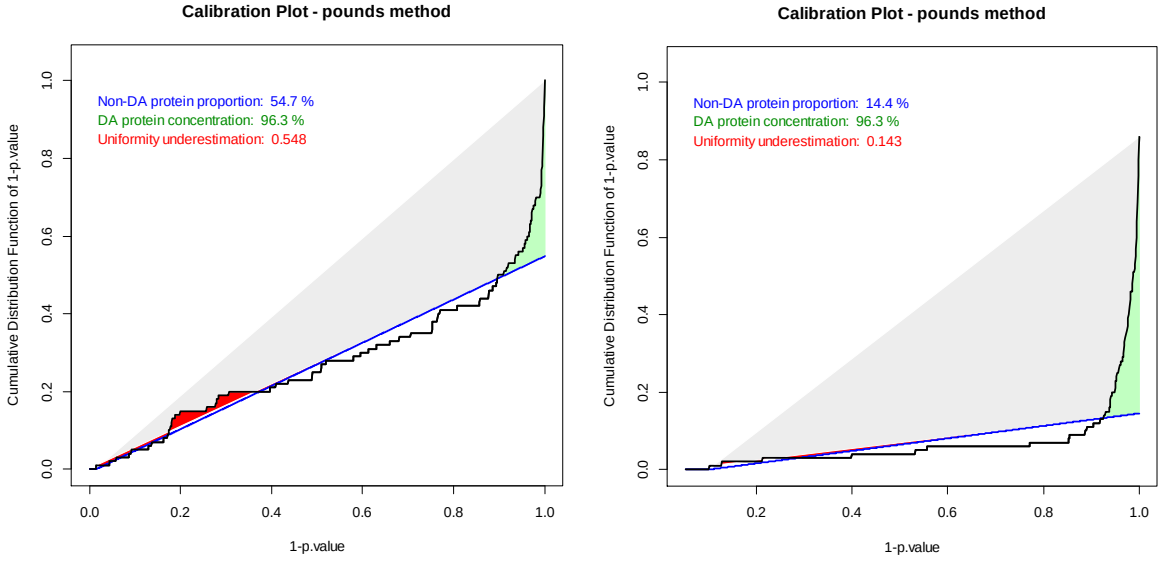

Figure 6: Calibration plots with n=100 and $\pi_0 = 0.5$ (left panel) or $\pi_0 = 0.1$ (right panel).

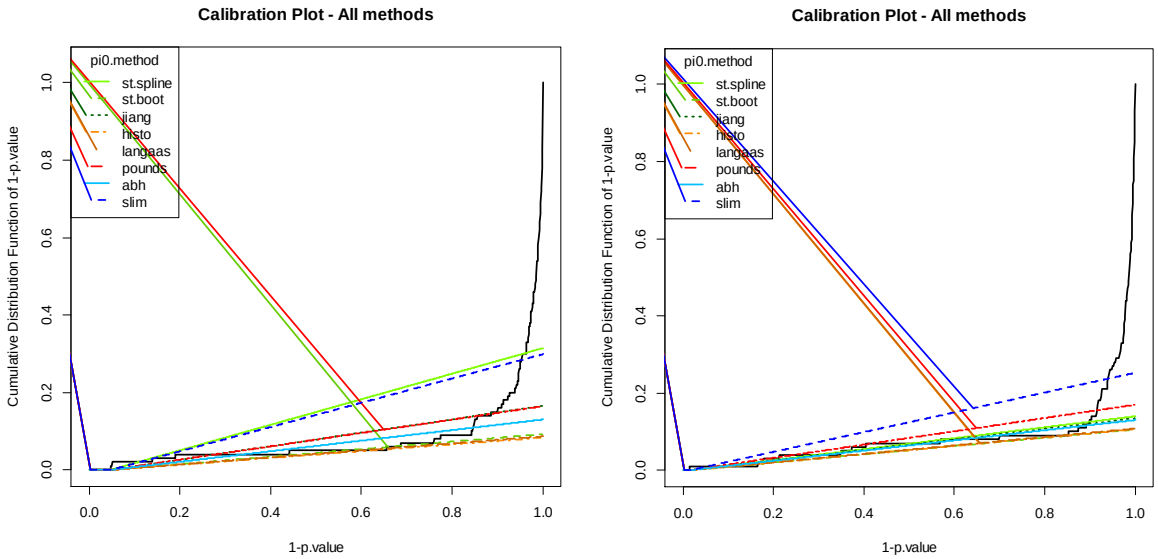

Figure 7: Different calibration plots corresponding to the scenario of Figure 6 (right panel) : Depending on the dataset, the estimate which is the more conservative is not the same, so that tuning it on purpose is necessary.

Now, let us return to the original simulation which provided good calibration and let us add 50 non-DA proteins (thus with a rather high p-value) that have strongly correlated behavior across the samples. This can be achieved by the following code:

```
> n <- 1000
> pi0 <- 0.5
> pval1 <- rbeta(n*(1-pi0),shape1=0.5, shape2=20)
> pval2 <- runif(n*pi0)
> pcol <- jitter(rep(0.75,50))
> pval <- c(pval1,pval2,pcol)
```

As a result of the correlation, their p-values will be roughly similar, which breaks the uniformity assumption. As a result, one observes a calibration of lower quality (Fig. 8). Moreover, depending on the π_0 estimator, for a same calibration curves, the calibration issue can appears as having different origin (either uniformity underestimation or too low concentration). Even though such limit cases are difficult to assess, they must be considered as ill-calibrated. To compensate for this, it is possible to increase π_0 by selecting an appropriate estimator, yet, it is sometime not sufficient, so that one has to fix π_0=1 (as with BH original estimator) to have the most conservative FDR estimate, as illustrated on Fig. 9.

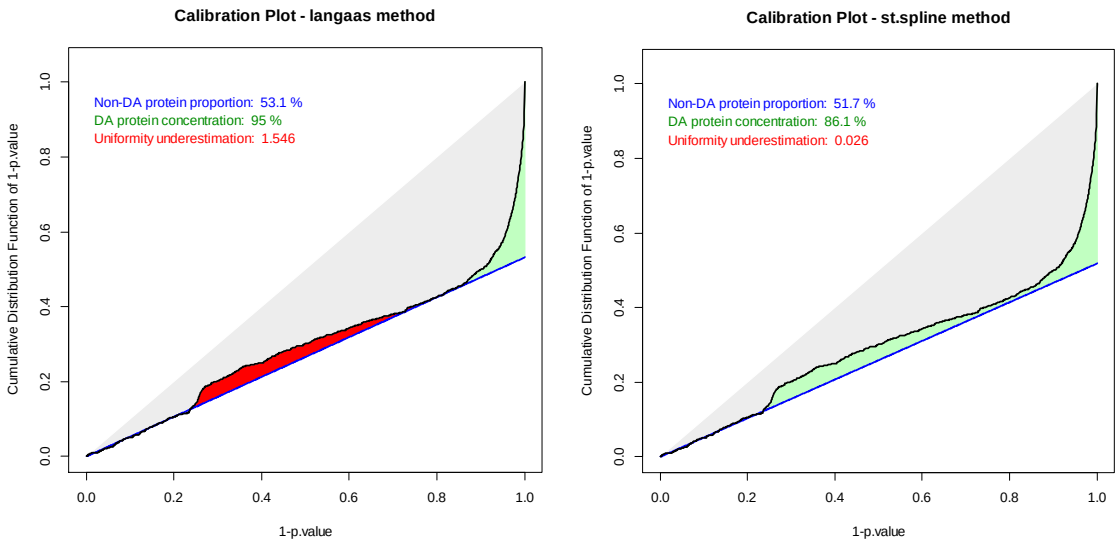

Figure 8: Calibration plots in case of correlated analytes which lead to non-DA uniformity. Depending on the accuracy of the estimator for each case, such calibration plot may appear as having a non uniform non-DA distribution (left panel with Langaas estimate) or an insufficient concentration of DA proteins (right panel, Storey's estimator based on spline).

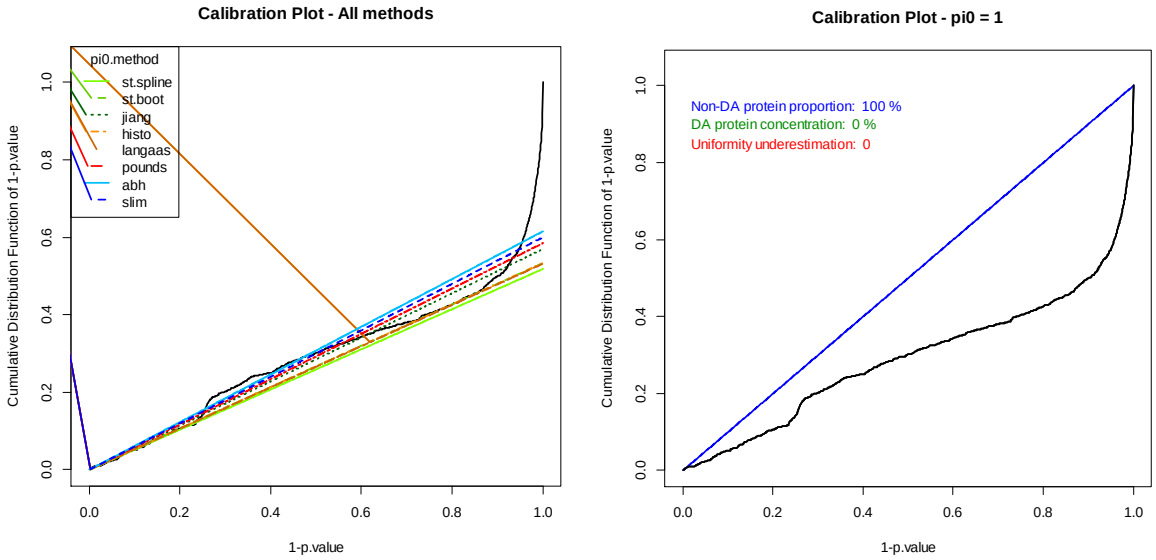

Figure 9:To recover correct calibration, one needs to increase π_0 beyond what the most over-conservative estimate proposes (left panel), so that one has to rely on original BH procedure, by tuning π_0 to 1 (right panel).

Now that the influence of a single yet large group of correlated proteins (50 of them) on the calibration curve is wel--pictured, one easily understands why in real life datasets, one observes more progressive curves. In these datasets, there is no such thing as a single large group of highly correlated proteins but instead, a continuum of more or less correlated pairs or small groups of proteins. Thus, instead of having a single highly visible bump (in red on Fig. 8 left panel) one has a progressive deviation from the straight line depicting the uniformity, as illustrated on Fig 10 (on the basis of the iSa dataset [4]). This dataset typically reproduces the kind of calibration plot one observes on numerous proteomics dataset with a large enough number of proteins to avoid stepped curves, yet with partially correlated proteins and few replicates per conditions, leading to an ill-calibrated plot (progressive curve without sharp angle on the right hand side and diverging π_0 estimates as illustrated on the left panel figure). In this context, it is advised to promote a conservative estimate (here, abh or slim – right panel figure) or even to stick to BH procedure.

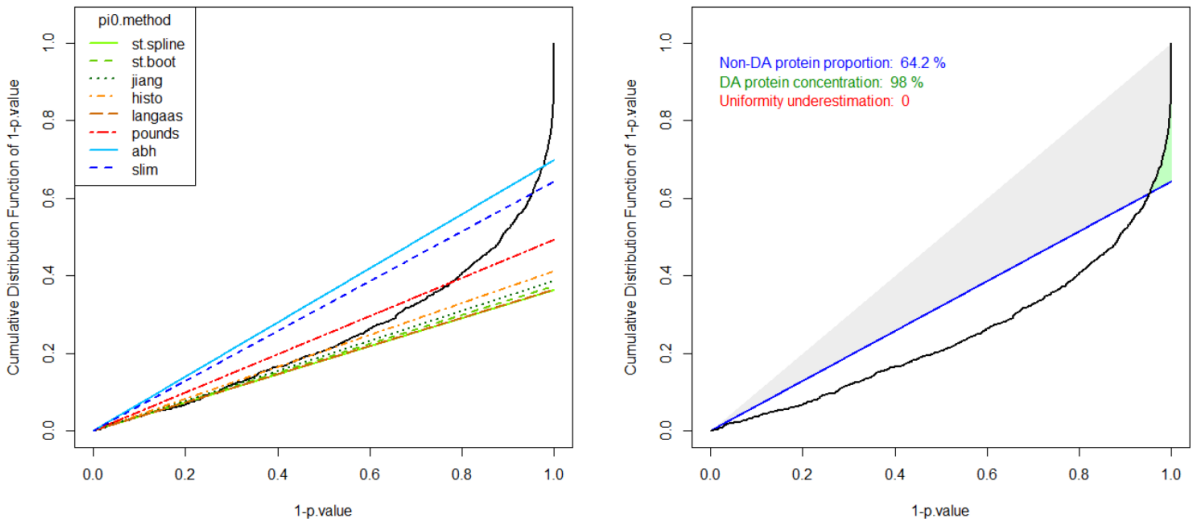

Figure 10: On the left panel, one observes that the various π_0 estimates are not converging. Moreover, due to the very progressive curve and the absence of sharp angle on the right hand side, it is difficult to determine if one estimate is better than the other is, so that an overconservative estimate should be preferred (right panel).

Several datasets can leads to calibration plots related to that of Fig. 10, yet with a more or less flat curve on the left side of the plot: This simply depends on the proportion of non-DA protein, as in Fig 4. However, what is important here is to assess that there is no sharp angle on the right side, making the distinction between DA and non-DA proteins blur, and leading to imprecise FDR estimation. However, in the case were very few proteins are DA, one can end up with a plot which is similar to that of Fig 5, yet, with an important divergence from uniformity which is "hidden" due to the curve being compacted around the diagonal (see Fig. 11). This figure displays the calibration plot of one of the cp4p datasets, where only 41 UPS proteins are DA in a complex yeast background. Due to the various correlation among the yeast proteins, the uniformity of non-DA protein is far from perfect, which explains why all the π_0 estimates are not as in line as on Fig. 5. On such datasets, it is important to fix π_0 to high values, because in addition to be more conservative, it corresponds to the reality underlying the experiment.

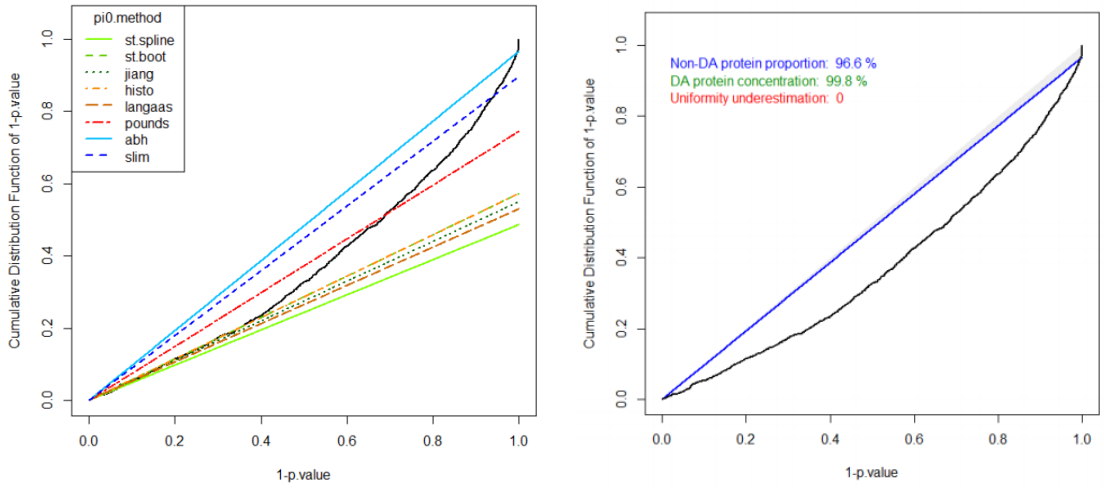

Figure 11: A real dataset (from [3]), with a large proportion of non-DA proteins, which differ from Fig. 5 due to the non-perfect uniformity distribution under the null hypothesis. As often, relying on a more conservative π_0 estimate is efficient to recover uniformity.

Now let us illustrate the influence of the logFC cutoff on the p-value calibration. Let us consider another cp4p dataset accessible through the demo mode of Prostar (One considers the Exp2_R100_prot which exhibits greater fold changes on DA proteins). In absence of filtering (see Fig 12, left panel), one observes an ill but manageable calibration plot. However, if the logFC cut of is tuned to 0.25, so that 65% of the proteins are removed, one ends up with a calibration plot which display a sharper curve (see Fig 12, left panel). However, the various estimate do not converge and the curve is less regular, indicating the FDR will most likely be instable. Thus, to compensate for this, it will be necessary to choose a more conservative π_0 leading to a higher FDR, so that the final benefit of the logFC filter is disputable.

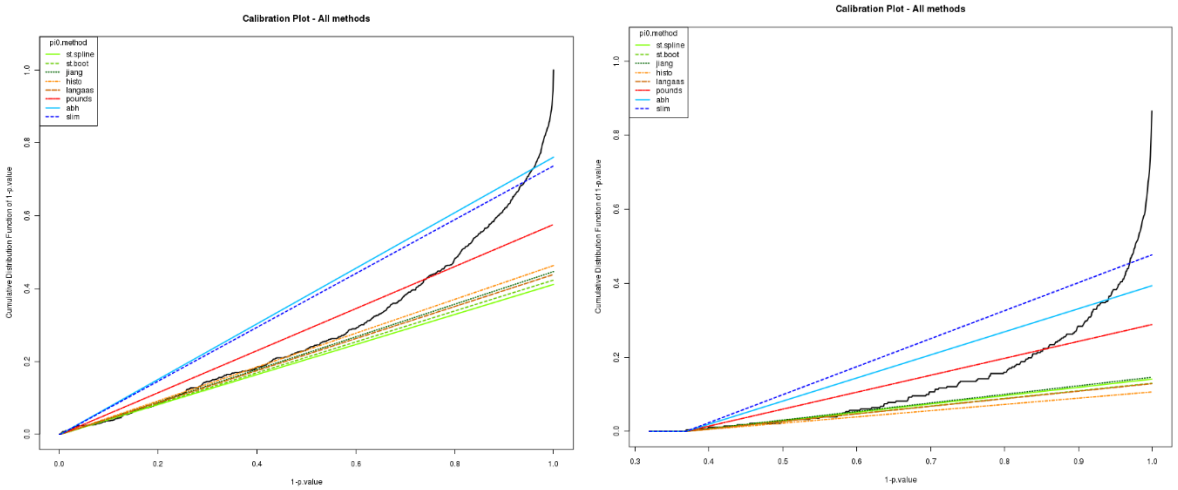

Figure 12: Two calibration plots of a same dataset without logFC filtering (left panel), and with a cutoff equal to to 0.25, so that 65% of the proteins are discarded.

4. Miscalibrated distribution examples

Now, if one increases the logFC cutoff to very high value, with the idea it will help getting rid of all the non-DA proteins, a rather deteriorated calibration plot can be obtained, as on Fig. 13, where a cutoff equal to 1 discards 97% of the proteins. On such cases, the FDR will be spurious not matter the tentative correction with increased π_0 values: the p-values are miscalibrated, and this cannot be corrected on the calibration plot. One has to review previous processing steps to avoid spurious FDR.

Another common situation is to have a large dataset with very few DA proteins. As the dataset is much larger than that of Fig. 13, even with very stringent logFC cutoffs, steps do not appears: the number of remaining proteins after filtering is high enough to have a smooth curve. As the dataset as very few DA proteins, it is difficult to obtain a low enough FDR, so that it can be tempting to increase the logFC cutoff: thanks to the dataset size, no steps appears so that the FDR can be assumed to be rather stable. Consequently, the logFC cutoff parameter can be devoid from its original use, and turn into a way to isolate on the volcao plot the very few proteins which look the most interesting. Although any reader of this article has now understand that it does not correspond to any good practice, the consequence on the calibration plot are immediate, as one observes a curve such as the one of Fig. 14. Concretely, any FDR computed after such calibration will have no value. The only solution in such a case is to decrease the logFC cutoff and to accept that it is not possible to have a low FDR with a large number of DA proteins if the dataset does not contain enough DA protein with large enough fold changes.

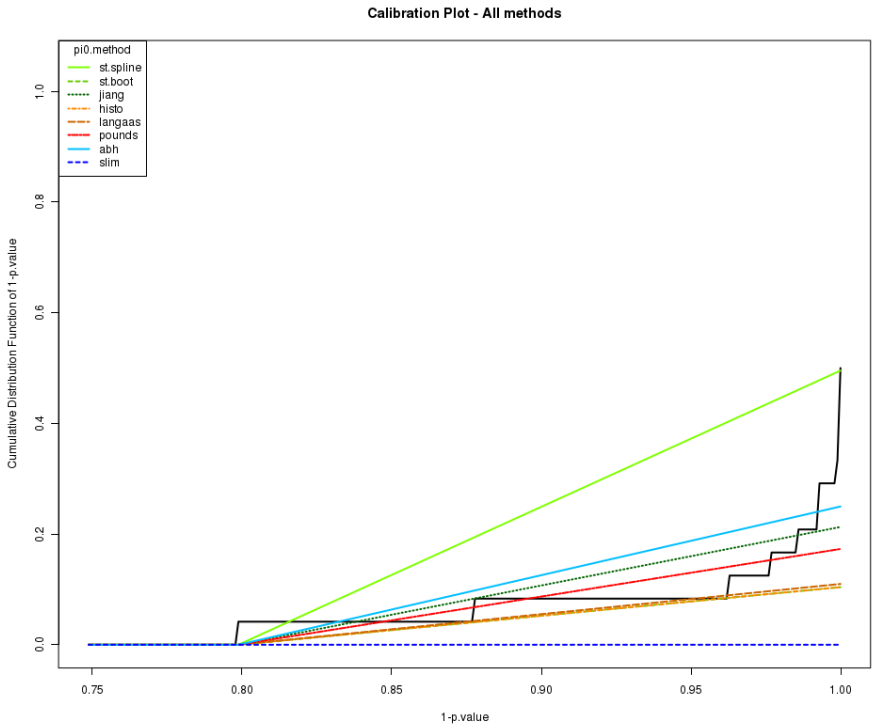

Figure 13: Once 97% of the proteins were discarded due to a too stringent logFC cutoff, the calibration curve is very irregular.

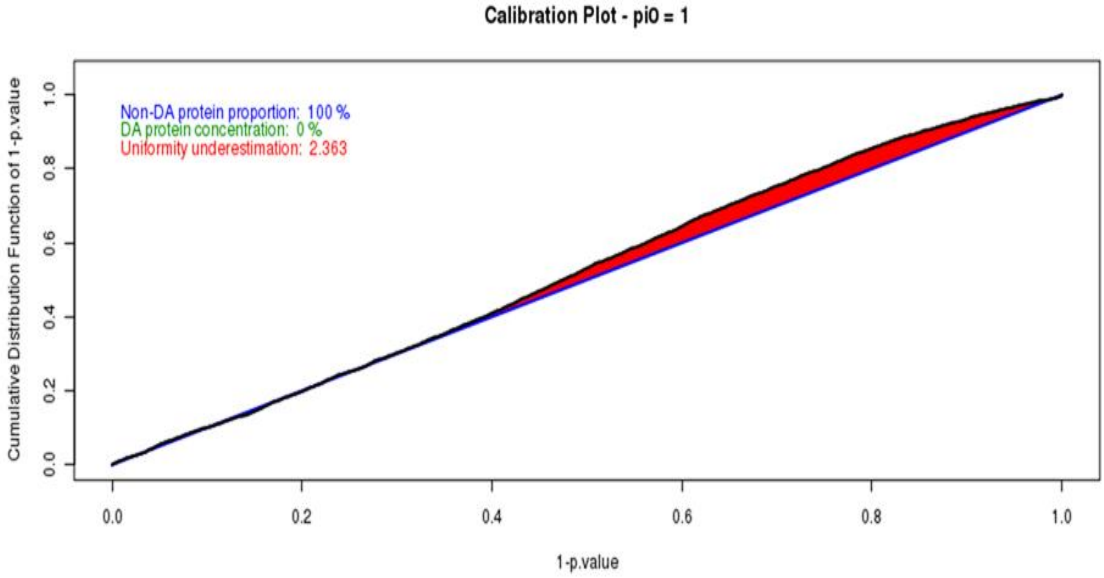

Figure 14: The effect of a too stringent logFC cutoff on a dataset with almost no clearly DA proteins (the corresponding data remains unpublished in this form, for the calibration is obviously not acceptable)

Finally, although the authors of the article have never witnessed such a dataset in real life, it is theoretically possible to have miscalibrated plot due to a bump akin to that of Fig. 8 (left panel) yet which is so big that it makes it impossible to compensate. Such calibration could be witnessed in the case of a sample preparation which enriched a subset of highly correlated proteins which appears to be completely non-DA. A simulated illustration of such type of miscalibration is proposed on Fig. 3 (left panel) in [2].

5. About the approximation of the logarithmized fold change

The approximation of the logarithmized fold-change is not often discussed, while in practice, it has consequences. Concretely, by using logFC (the approximation) instead of the real logarithmized fold-change, one replaces an arithmetic mean by a geometric one. These two types of means are known to provide different results. Depending on the numerical values, the logFC can be rather close to the real logarithmized fold-change, or not.

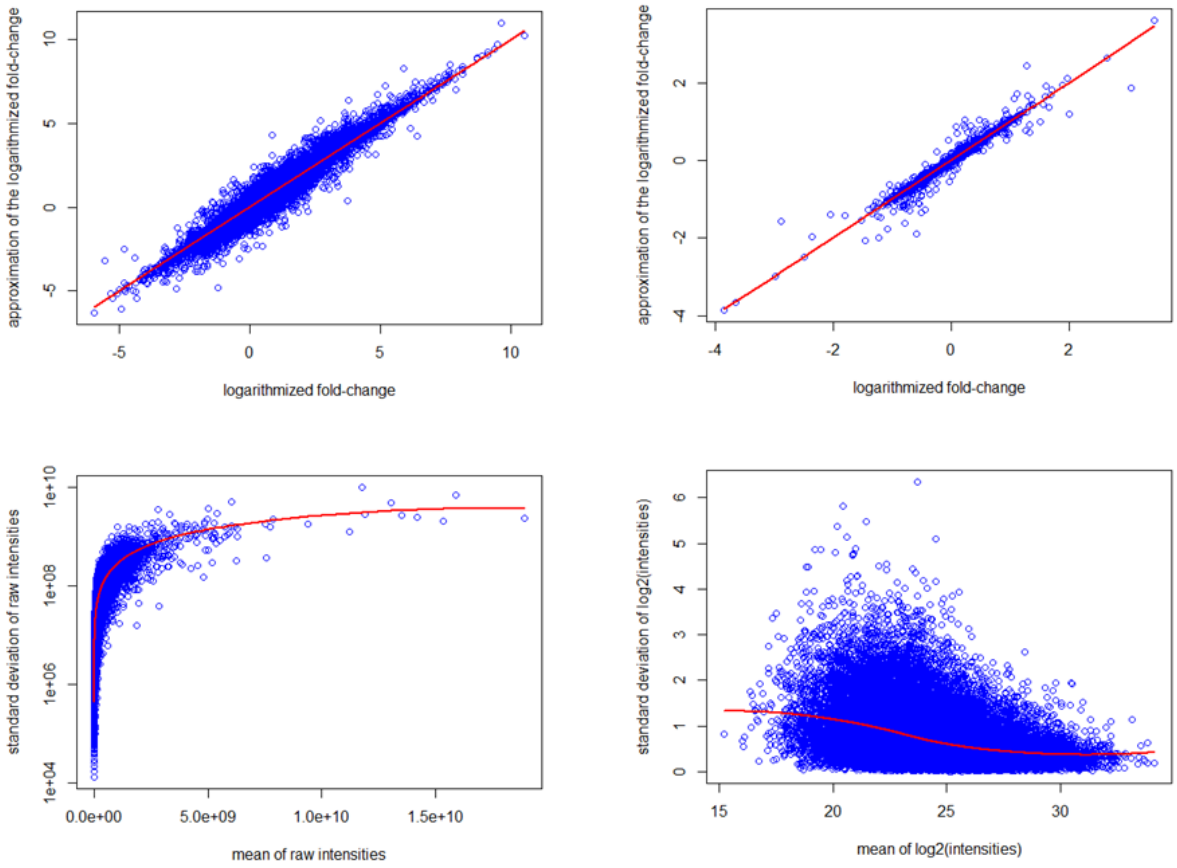

*Figure 15: **A** : logarithmized fold-change (x-axis) versus its approximation, temred logFC (y-axis) for 34114 peptides from the proteome benchmark dataset used in the MaxLFQ publication [1]. The red line has for equation y=x. **B** : logarithmized fold-change (x-axis) versus logFC (y-axis) for 8843 peptides from the UPS 2-fold dataset used in the cp4p publication. The red line has for equation y=x. **C** : Average observed intensities (x-axis) versus standard deviations of these raw intensities (y-axis) for 34114 peptides from the proteome benchmark dataset used in the MaxLFQ publication. The trend line in red has been estimated using the loess function in R. **D** : Average log2 intensities (x-axis) versus standard deviation of these log2 intensities (y-axis) for 34114 peptides from the proteome benchmark dataset used in the MaxLFQ publication. The trend line in red has been estimated using the loess function in R.*

However, on real datasets, they are often close to one other. For instance, in Fig. 15.A and Fig. 15.B, we estimate a red trend line y = x for both datasets, so that, in average, the approximations and the real values are equal. However, the noise around this trend has a standard deviation of 0.303 in Fig. 15.A. It means the approximation stands in a 95% confidence interval [+/-0.59]. In Fig. 15.B, the standard deviation is estimated at 0.058 around the trend (95% confidence interval is +/- 0.11). This illustrates well that the uncertainty of the approximation can vary from a dataset to another.

Besides, the approximation is more representative of what we compare after performing a log-transformation than the real logarithmized fold-change. Concretely, the log-transformation is important to delete a bias related to the intensity levels in the differential analysis. Indeed, there is generally a positive correlation between the observed means and standard deviations of raw intensities of peptides/proteins (Fig. 15.C). This can be a problem because in absence of log-transformation, low-intensity proteins are more likely to be selected in the differential analysis (see explanation below). The log-transformation allows diminishing the correlation between the means and standard deviations, and therefore can be used to mitigate this bias related to the intensity levels (Fig. 15.D). This is why log-transformation is suitable before differential analysis. In such case, the test statistics is a ratio which numerator is logFC (this why it is termed “t-test difference” in some software tools). On the other hand, this makes the two dimensions of the volcano plot largely dependent, which does not make is the most suitable representation.

To illustrate the bias induced by the correlation between the observed means and standard deviations of raw intensities, let us imagine a protein with high intensity levels in two conditions A and B, but having a mean difference between A and B in a same order of magnitude that a protein displaying lower intensity levels in both conditions. If the means and standard deviations of intensities are positively correlated in the dataset, such a high-intensity protein will tend to have a higher standard deviation than the low-intensity protein. As a result, the test statistic of this high-intensity protein will be lower than the one of the low-intensity protein, even if they have the same mean differences of intensities. Therefore, for the same mean differences of intensities between both conditions, there will be a tendency to have higher p-values for high-intensity proteins than for the low-intensity ones, what will lead to favor the selection of low-intensity proteins in the differential analysis.

6. Supplemental references

[1] Cox, J., Hein, M. Y., Luber, C. A., Paron, I., Nagaraj, N., & Mann, M. (2014). Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Molecular & cellular proteomics*, *13*(9), 2513-2526.

[2] Giai Gianetto, Q., Combes, F., Ramus, C., Bruley, C., Couté, Y., & Burger, T. (2016). Calibration plot for proteomics: A graphical tool to visually check the assumptions underlying FDR control in quantitative experiments. *Proteomics*, *16*(1), 29-32.

[3] Wieczorek, S., Combes, F., Lazar, C., Giai Gianetto, Q., Gatto, L., Dorffer, A., ... & Burger, T. (2016). DAPAR & ProStaR: software to perform statistical analyses in quantitative discovery proteomics. *Bioinformatics*, *33*(1), 135-136.

[4] Bounab, Y., Iannascoli, B., Grieco, L., Couté, Y., Niarakis, A., Roncagalli, R., ... & Garin, J. (2013). Proteomic analysis of the SH2domain-containing leukocyte protein of 76 kDa (SLP76) interactome. *Molecular & Cellular Proteomics*, *12*(10), 2874-2889.