Supplementary material for: Investigating Protein Patterns in Human Leukemia Cell Line Experiments: A Bayesian Approach for Extremely Small Sample Sizes

Thierry Chekouo, Francesco Stingo, Caleb Class, Yuanqing Yan, Zachary Bohannan, Yue Wei, Guillermo Garcia-Manero, Samir Hanash, and Kim-Anh Do.

A Additional results on the proteomic data analysis

In this Section, we present an additional set of results from the analysis of the three subproteomic data. Section A.1 summarizes the distribution of the number of protein isoforms across the three subproteomes: TCE, surface and nucleus. Second, in Section A.2, we check the goodness of fit of the proposed approach. Finally, in Section A.3, we illustrate an analysis of the complete dataset that also includes proteins with very low abundance.

A.1 Distribution of protein isoforms

Table A.1 in this Web Appendix summarizes the number of proteins identified and expressed across all the samples from a specific cell line. The TCE subproteome, which includes proteins found in both the nucleus and at the cellular surface (as well as in the cytoplasm), contains a larger number of proteins than the other subproteomes (TCE had 5,842 proteins, of which 4,081 were identified in cell line TF1, 4,549 in cell line u937, and 1,102 in cell line HL60). Fewer proteins (2,341 proteins) were identified in the three subproteomes in cell line HL60.

Table A.1: Summary of the number of proteins identified in each cell line type and subproteomic dataset.

	TF1	u937	HL60	Total
TCE	4081	4549	1102	5824
Nuclear	1836	1633	1286	2470
Surface	1477	1114	1278	2147
Total	5123	5305	2341	6922

A.2 Model checking on the proteomic data analysis

In this section, we perform model checking on the observed data. For that, we aim to compare the posterior predictive protein expressions with the observed protein expressions. We applied this procedure to the 9 observed data sets as described in the paper. Here we focus on *Objective 1*, i.e., detection of groups of protein isoforms based on the change in expression between sensitive and resistant samples at a specific time point. After integrating out the cluster memberships ρ_{jk}, the posterior predictive density of a new protein expression $\boldsymbol{y}_j^{\text{new}}$ can be calculated as:

1. for every experimental condition $k = 1, ..., K$, we predict the cluster of a pair (j, k) as
$$P(\rho_{jk} = h|data) \propto \hat{\pi}_h \mathcal{N}(\boldsymbol{y}_{jk}^{\text{new}}; \boldsymbol{X}_k^T \hat{\boldsymbol{\beta}}_h + \hat{\mu}_h \mathbb{1}_{n_k}, \hat{\sigma}_h^2 \boldsymbol{\Sigma}_{kh}) \quad (1)$$

2. Given $\rho_{jk} = h$, the distribution of $\boldsymbol{y}_{jk}^{\text{new}}$ should then follow a multivariate normal
$$\mathcal{N}(\boldsymbol{X}_k^T \hat{\boldsymbol{\beta}}_h + \hat{\mu}_h \mathbb{1}_{n_k}, \hat{\sigma}_h^2 \boldsymbol{\Sigma}_{kh}) \quad (2)$$

where

- $\hat{\boldsymbol{\beta}}_h$, $\hat{\mu}_h$, $\hat{\sigma}_h^2$ are the posterior means of $\boldsymbol{\beta}_h$, μ_h, σ_h^2 respectively.
- $\hat{\pi}_h = \dfrac{1}{pK + \sum \alpha_h}(\alpha_h + \dfrac{1}{N}\sum_{l=1}^{N} p_h^{(l)})$ is the posterior mean of π_h, and $p_h^{(l)}$ is the number of elements in cluster h for MCMC sample l, N is the total number of MCMC iterations after burn-in.

We then performed a one-sample Kolmogorov-Smirnov test for testing whether the observed gene expressions for each dataset follows the distribution (2). Box plots of p-values (non-adjusted) of protein expressions are shown in Figure A.1. They show large pvalues for the test for every dataset with median values ranging from 0.6 to 0.92. In addition, Table A.2 shows that minimum pvalues for each of the 9 dataset: only dataset TCE with cell line TF1 has a minimum pvalue less than 0.05.

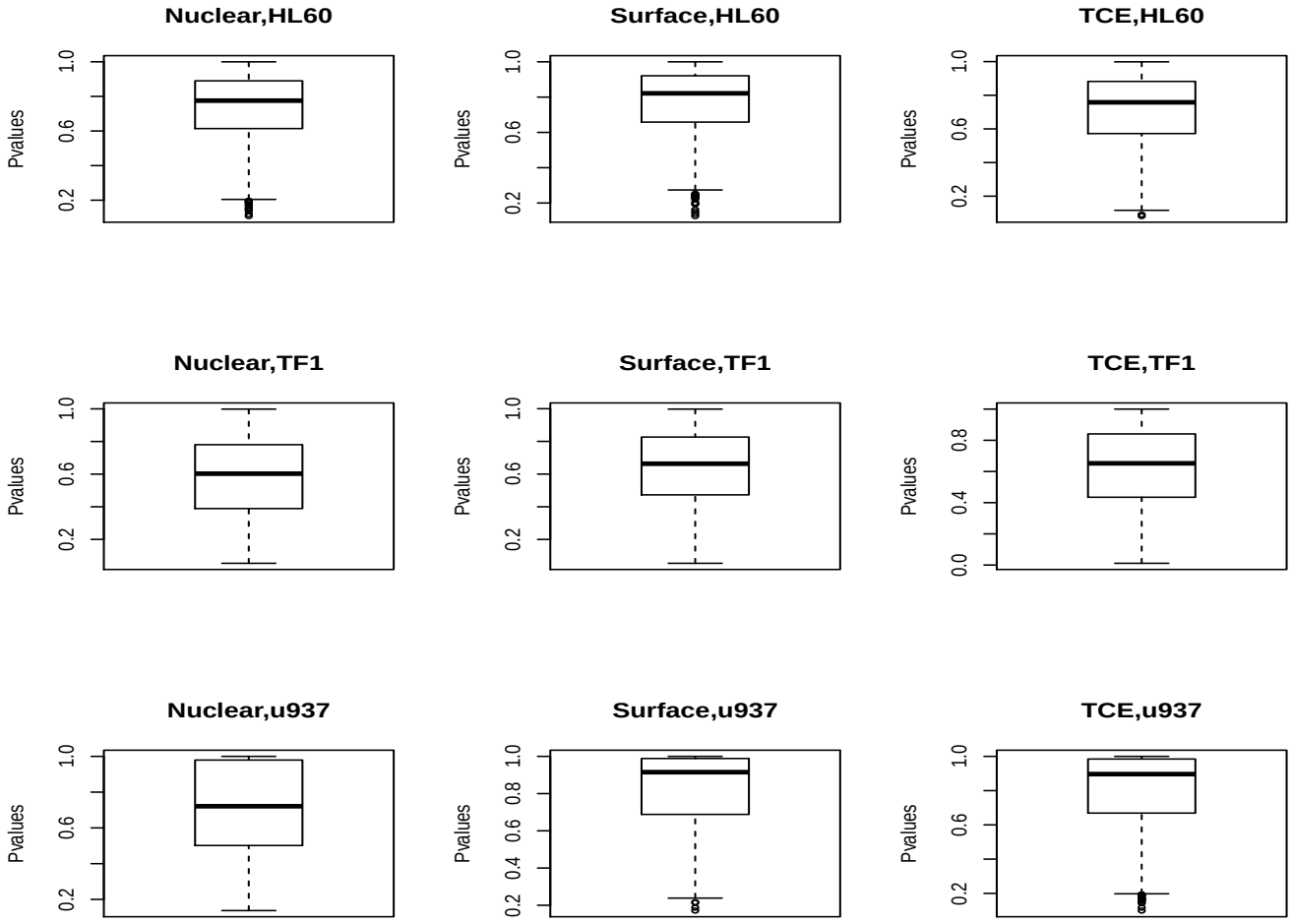
Table A.2: Minimum p-values for each dataset

	Nuclear	Surface	TCE
HL60	0.108	0.126	0.082
TF1	0.053	0.054	0.011
u937	0.137	0.1714	0.099

A.3 Analysis of the complete dataset that includes low abundance protein isoforms

In the main paper, we presented results on proteomic data after excluding protein isoforms with abundance less than 5 in at least one sample, as these low-abundance proteins often do not provide reproducible results. In this Section, we present some results after including now isoforms with abundance less than 5 in only one condition (if we have two conditions) or two conditions (if we have three conditions). Hence, some

Figure A.1: Boxplot of pvalues obtained from the Kolmogorov Smirnov test



of these isoforms would have low-abundance in some conditions and high-abundance in other conditions. Results show that two proteins selected in this new analysis were also selected in the analysis presented in the main manuscript (see Table A.3 and Table 5 in the main manuscript). In Table A.3, boldface protein id names are common proteins selected in both analysis, italic protein names are proteins that are only selected with this new analysis.

B Combined likelihood and Full conditionals

Let $\boldsymbol{y}_{jk} = (y_{jk1}, y_{jk2}, ..., y_{jkn_k})$ be a vector in which each element y_{jki} represents the ($\log_2$-) expression of the feature (e.g., protein) j in sample $i = 1, ..., n_k$ of experimental type $k = 1, ..., K$. In one of our applications (see Likelihood for Objective 2 in Section 3.1 in the paper), k represents whether a sample is from a sensitive or resistant cell line ($k = s \in \{1, 2\}$). The full dataset can be represented by the vector $\boldsymbol{Y} = (\boldsymbol{y}_j) = (\boldsymbol{y}_{jk}, j = 1, ..., p; k = 1, ..., K)$, where p denotes the number of features (proteins). In addition, we introduce a binary q-vector $\boldsymbol{x}_{ki}$ that captures an additional characteristic

Table A.3: List of *Down-Down-Down* proteins identified from u937 in both nuclear and cell-surface data with respect to their posterior probabilities and estimated q-values. Prob(Nucl) and Q(Nucll) are respectively marginal posterior probabilities (jMPP) and estimated q-values for nuclear data, Prob(Surf) and Q(Surf) for cell-surface data.

UniProt id	Gene	Prob(Nucl)	Q(Nucl)	Prob(Surf)	Q(Surf)
P27816-5	–	1.00	0.00	1.00	0.00
C9JL19	HSPD1	0.97	0.01	1.00	0.00
Q8NC51	SERBP1	0.91	0.02	1.00	0.00
P0CG39	POTEJ	0.81	0.05	0.94	0.01
Q07065	CKAP4	0.59	0.09	1.00	0.00
B0YJC5	VIM	0.52	0.10	0.93	0.01

$\boldsymbol{X}$ of our data (e.g., the collection time of the sample in Analysis 2). The generic element of this vector $x_{ki}^{(l)}$ is set to 1 if sample i of type k assumes level $l = 1, ..., q$ of this characteristic, and 0 otherwise. The general likelihood of the model can be written as

$$P(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{a}, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}) = \prod_{j,k} \prod_{i=1}^{n_k} \mathcal{N}(y_{jki}; a_{j\rho_{jk}} + \boldsymbol{x}_{ki}^T \boldsymbol{\beta}_{\rho_{jk}}, \sigma^2_{\rho_{jk}}), \tag{3}$$

where $\boldsymbol{\rho} = (\rho_{jk})$ is the cluster membership variable defined by $\rho_{jk} = h$ if protein j expressed on a sample of type k (i.e., pair (j, k)) belongs to cluster h, and 0 otherwise. When $q \leq 2$, we define priors on $\boldsymbol{\beta}_{\rho_{jk}}$ as defined in the main paper. However, when $q > 2$, priors are instead defined on $\boldsymbol{\beta'}_{\rho_{jk}} = \boldsymbol{C}\boldsymbol{\beta}_{\rho_{jk}}$ (or the q-binary matrix $\boldsymbol{x}_{ki}$ is replaced by $\boldsymbol{x}_{ki}^T \boldsymbol{C}^{-1}$) in order to clearly define our pre-defined patterns, where $\boldsymbol{C}^{-1} = (c_{ll'})$ is a lower triangular contrast matrix of 1's (i.e., $c_{ll'} = 1$ if $l \geq l'$ and 0 otherwise).

Let $p_h = \sum_{j,k} I(\rho_{jk} = h)$ be the number of elements in cluster h, and $p_{kh} = \sum_j I(\rho_{jk} = h)$ be the number of features (proteins) in cluster h expressed in samples of type k. $I(\rho_{jk} = h) = 1$ if $(j, k) \in h$ and 0 otherwise. By integrating out $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ from the likelihood (4) (in the paper) and their prior distributions, we obtain

$$p(\boldsymbol{Y}, \boldsymbol{\rho}|\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \;=\; \prod_{j,k} \mathcal{N}(\boldsymbol{y}_{jk}; \boldsymbol{X}_k^T \boldsymbol{\beta}_{\rho_{jk}} + \mu_{\rho_{jk}} \mathbb{1}_{n_k}, \sigma^2_{\rho_{jk}} \boldsymbol{\Sigma}_{k\rho_{jk}}) \int \prod_{h=1}^{H} \pi_h^{p_h} p(\boldsymbol{\pi}) \mathrm{d}(\boldsymbol{\pi})$$

where $\boldsymbol{X}_k = (\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{kn_k})^T$ is an $n_k \times q$ binary matrix, and for $\rho_{jk} = h$, $\boldsymbol{\Sigma}_{kh} = \mathbb{1}_{n_k} \mathbb{1}_{n_k}^T c_h + \boldsymbol{I}_{n_k}$, its inverse $\boldsymbol{\Sigma}_{kh}^{-1} = \boldsymbol{I}_{n_k} - \frac{c_h}{n_k c_h + 1} \mathbb{1}_{n_k} \mathbb{1}_{n_k}^T$, $\mathbb{1}_{n_k}^T$ the n_k-vector of 1's, and $\boldsymbol{I}_{n_k}$ the identity matrix of size n_k.

We can easily show that the full conditionals of cluster memberships can be written as

$$p(\rho_{jk} = h|\rho_{(-jk)}, \boldsymbol{\beta}_h, \mu_h, \sigma_h^2) \propto \frac{\sum_{(j',k') \neq (j,k)} I(\rho_{j'k'} = h) + \alpha_h}{pK - 1 + \sum_{h=1}^{H} \alpha_h} \left[\mathcal{N}(\boldsymbol{y}_{jk}; \boldsymbol{X}_k^T \boldsymbol{\beta}_h + \mu_h \mathbb{1}_{n_k}, \sigma_h^2 \boldsymbol{\Sigma}_{kh})\right] \tag{4}$$

For any component $|\beta_{hl}| > 0$, we showed that, given τ_{hl}, $\boldsymbol{\rho}$, μ_h, σ_h and $\beta_{h(-l)}$,

$sign(\beta_{hl})\beta_{hl} \sim \mathcal{TN}(\tau_{hl}, \mu_{\beta_{hl}}, \sigma^2_{\beta_{hl}})$ where

$$\begin{aligned}
\sigma^{-2}_{\beta_{hl}} &= \sigma_h^{-2}/b_\beta + \sigma_h^{-2}\sum_{k=1}^{K} p_{kh}\boldsymbol{x}_k^{(l)T}\boldsymbol{\Sigma}_{kh}^{-1}\boldsymbol{x}_k^{(l)} \\
&= \sigma_h^{-2}\left(\frac{1}{b_\beta} + \sum_{k=1}^{K} p_{kh}(\boldsymbol{x}_k^{(l)T}\boldsymbol{x}_k^{(l)} - \frac{c_h}{n_k c_h + 1}\boldsymbol{x}_k^{(l)T}\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T\boldsymbol{x}_k^{(l)})\right) \qquad (5) \\
\mu_{\beta_{hl}} &= \frac{\sigma^2_{\beta_{hl}}}{\sigma_h^2}\left(\frac{\tau_{hl}}{b_\beta} + \sum_{(j,k)\in h} \boldsymbol{x}_k^{(l)T}\boldsymbol{\Sigma}_{kh}^{-1}(\boldsymbol{y}_{jk} - \boldsymbol{x}_{ki}^{(-l)T}\boldsymbol{\beta}_{h(-l)} - \mu_h\mathbb{1}_{n_k})\right) \qquad (6)
\end{aligned}$$

We also showed that given $\boldsymbol{\rho}$, $\boldsymbol{\beta}_h$ and σ_h, $\mu_h \sim \mathcal{N}(\mu_{\mu_h}, \sigma^2_{\mu_h})$ where

$$\begin{aligned}
\sigma^{-2}_{\mu_h} &= 1/c^2 + \sigma_h^{-2}\sum_{k=1}^{K} p_{kh}\mathbb{1}_{n_k}^T\boldsymbol{\Sigma}_{kh}^{-1}\mathbb{1}_{n_k} \qquad (7) \\
&= 1/c^2 + \sigma_h^{-2}\sum_{k=1}^{K} p_{kh}\frac{n_k}{n_k c_h + 1} \qquad (8) \\
\mu_{\mu_h} &= \frac{\sigma^2_{\mu_h}}{\sigma_h^2}\sum_{(j,k)\in h} p_{kh}\mathbb{1}_{n_k}^T\boldsymbol{\Sigma}_{kh}^{-1}(\boldsymbol{y}_{jk} - \boldsymbol{X}_k^T\boldsymbol{\beta}_h) \qquad (9)
\end{aligned}$$

Finally, we updated the variance for each cluster as $\sigma_h^2 \sim \mathcal{IG}(a_{\sigma_h^2}, b_{\sigma_h^2})$ where

$$\begin{aligned}
a_{\sigma_h^2} &= a + \sum_{k=1}^{K}\frac{n_k p_{kh}}{2} \qquad (10) \\
b_{\sigma_h^2} &= b + \frac{1}{2}\sum_{(j,k)\in h}(\boldsymbol{y_{jk}} - \boldsymbol{X}_k^T\boldsymbol{\beta}_h - \mu_h\mathbb{1}_{n_k})^T\boldsymbol{\Sigma}_{kh}^{-1}(\boldsymbol{y_{jk}} - \boldsymbol{X}_k^T\boldsymbol{\beta}_h - \mu_h\mathbb{1}_{n_k}) \qquad (11)
\end{aligned}$$

C Simulation results

Figures C.1 and C.2 in this Web Appendix show simulation results when the absolute effects $|\beta_{hl}|$ are respectively 0.5 and 1. In these figures, we plot different thresholds for classification against the average proportions of non-classified (NC), misclassified (MC) and well-classified (WC) protein expression profiles. We computed the area over the WC curve A_{wc}, which can be considered as a measure of classification performance.

Figure C.3 in the Web Appendix depicts kernel density estimates of the posterior means of all the β_h in absolute values for one simulated dataset ($\beta_{hl} = 0.8$ and $d_s = 2$). It appears that all the densities are centered at their true values, 0.8, confirming a satisfactory performance of our approach.

In some contexts, classifying "Up" as "Down" (or vice versa) might be more problematic than classifying "Up" or "Down" as "Flat" (or vice versa). In order to understand which type of errors influence the misclassification error (MCE) rate, we have computed another type of misclassification rate for which we define misclassification as classifying "Up" as "Down" or vice versa. Table C.1 shows that those MCEs are zero except for the case of only one replicate and a small effect size ($d_s = 1$ and $\beta = 0.5$).

Figure C.1: Simulation results for $|\beta_{hl}| = 0.5$: percentage of non-classified (NC), misclassified (MC) and well-classified (WC) protein expression profiles with respect to different thresholds on marginal posterior probabilities, and the number of samples, d_s, on 50 replicate data.

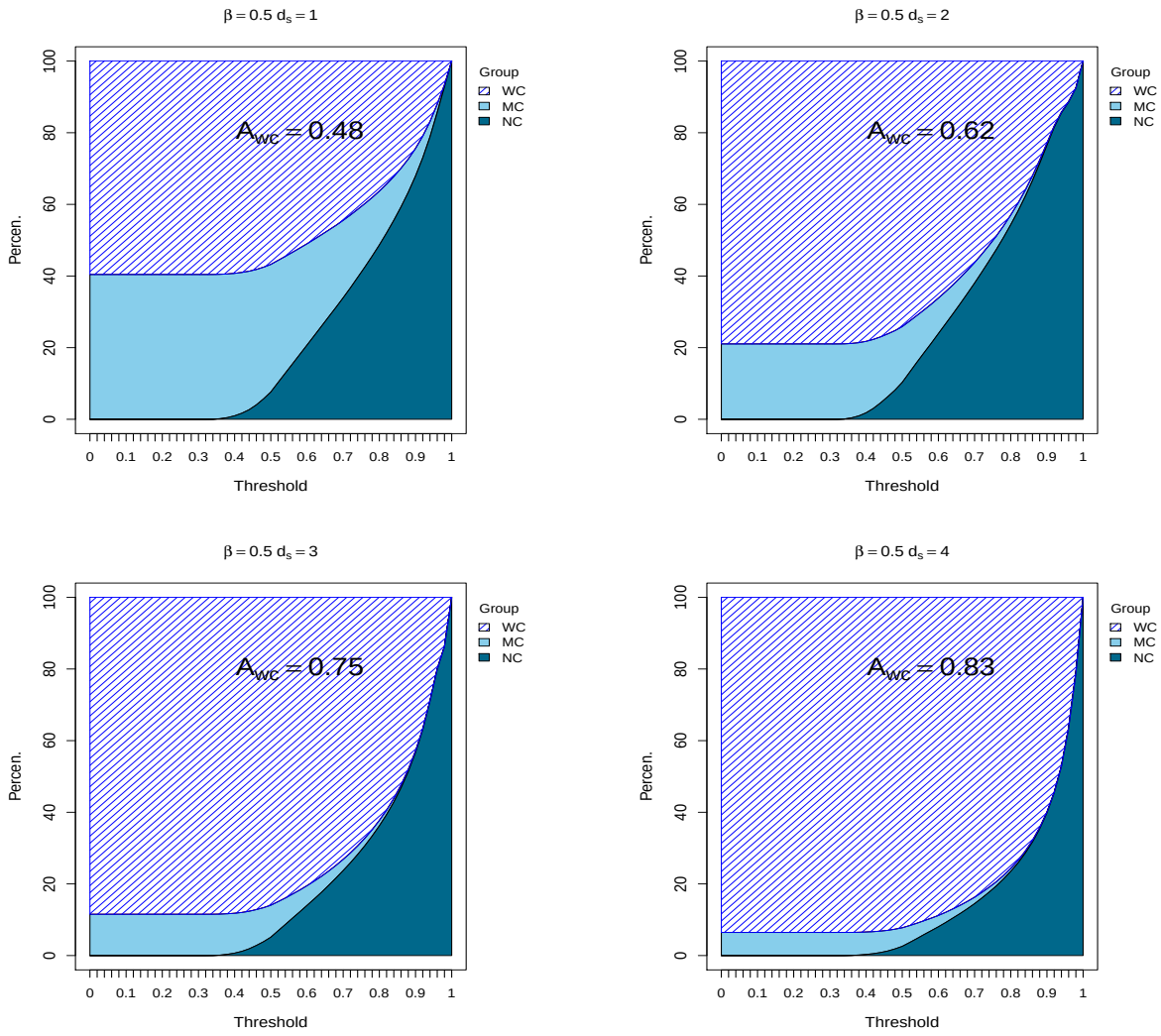


Figure C.2: Simulation results for $|\beta_{hl}| = 1$: percentage of non-classified (NC), misclassified (MC) and well-classified (WC) protein expression profiles with respect to different thresholds on marginal posterior probabilities, and the number of samples, d_s, on 50 replicate data.

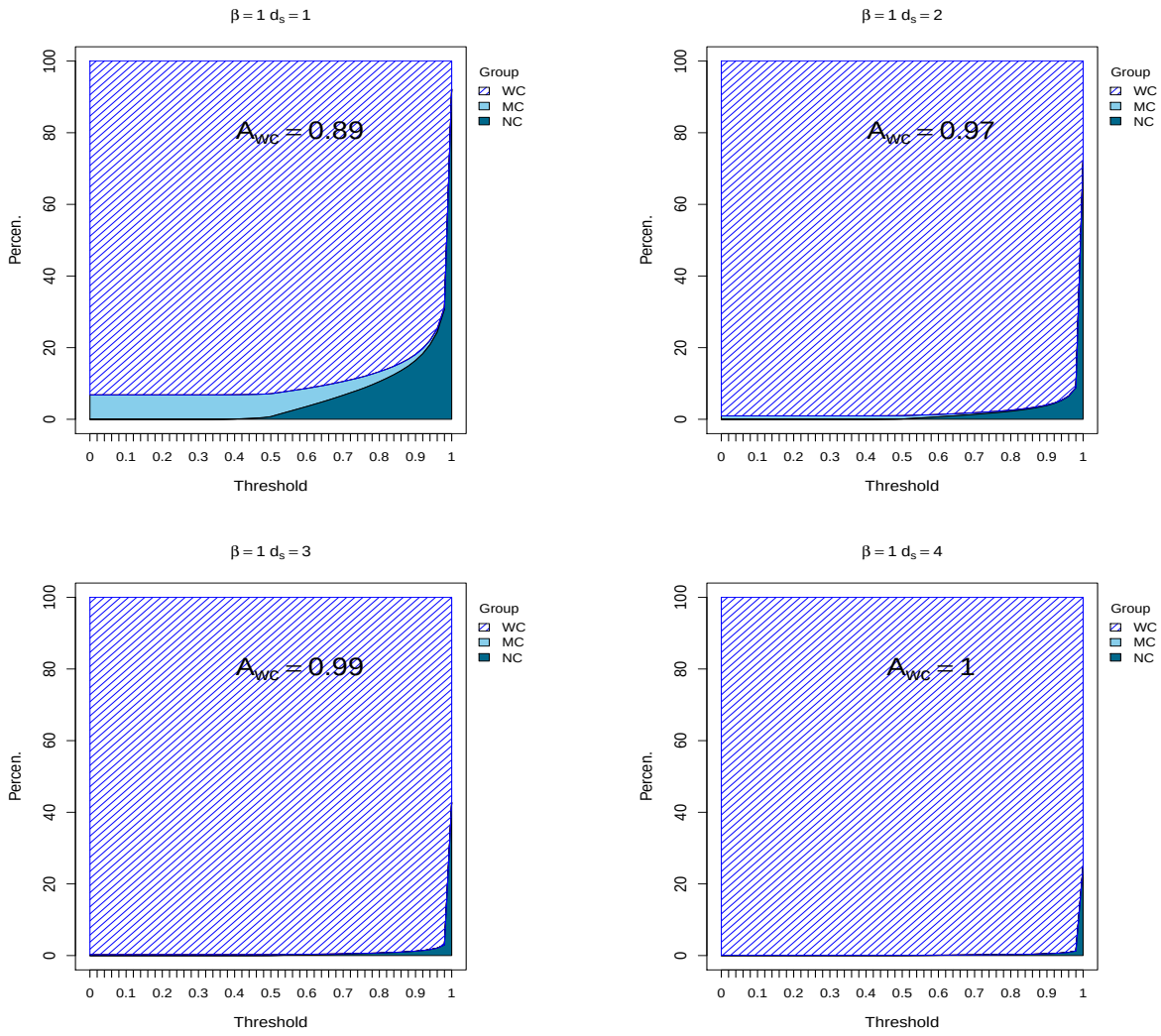


Figure C.3: Posterior density estimates of the slopes β_{hl}'s (or $\log_2$ fold-changes).

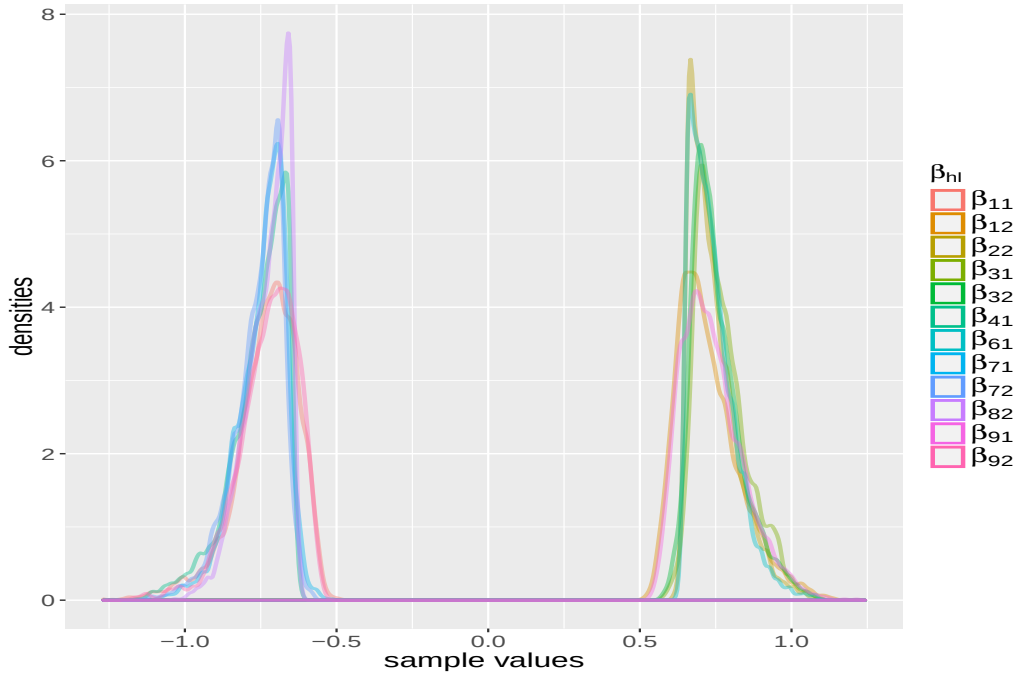


Table C.1: Simulation results. Data are generated as in Section 6 of the manuscript. MCE is the average misclassification error over ten replicate simulated datasets. MCE_{UD} is the average MCE of classifying "Up" as "Down" and vice versa. Standard deviations are within parentheses.

β	d_s	MCE	MCE_{UD}
0.50	1.00	0.404159 (0.000932)	0.003043 (0.000143)
0.80	1.00	0.150247 (0.000700)	0.000031 (0.000009)
1.00	1.00	0.068166 (0.000458)	0.000000 (0.000000)
0.50	2.00	0.210316 (0.000752)	0.000112 (0.000021)
0.80	2.00	0.038422 (0.000345)	0.000000 (0.000000)
1.00	2.00	0.009691 (0.000176)	0.000000 (0.000000)
0.50	3.00	0.115294 (0.000425)	0.000010 (0.000006)
0.80	3.00	0.011844 (0.000171)	0.000000 (0.000000)
1.00	3.00	0.002478 (0.000080)	0.000000 (0.000000)
0.50	4.00	0.064519 (0.000452)	0.000000 (0.000000)
0.80	4.00	0.004175 (0.000151)	0.000000 (0.000000)
1.00	4.00	0.000866 (0.000053)	0.000000 (0.000000)

D Sensitivity analysis

We present an analysis of the sensitivity of our inference to the value of the hyperparameters. In particular, we focus on the prior distribution of three parameters: (a) σ_h^2, the variance parameter for each cluster, (b) τ_{hl}, the threshold parameter on the effect β_{hl}, and (c) π_h, the unknown proportion of the included features (proteins) in component $h = 1, \cdots, H$. For each case, we computed the area over the well-classified (WC) curve A_{wc}, the area under the proportion of misclassified (MC) protein expres-

sion profiles, A_{mc}, and the area under the proportion of non-classified (NC) protein expression profiles, A_{nc} (see Figure C.1 for more details). We analyzed a simulated dataset generated by setting $|\beta_{hl}| = 0.8$, $d_s = 2$ (the number of samples), and all the other parameters to their "default" values as described in the main paper.

D.1 Prior for σ_h^2

We conducted simulations to evaluate the robustness of our clustering results to the choices of the hyperparameters of the prior distribution of σ_h^2. Remember that $\sigma_h^2 \sim \mathcal{IG}(a, b)$, an inverse gamma distribution with parameters a and b. We fixed $b = H^{-1/6} = 0.7$ and investigated five different values of the shape parameter, $a = 0.3, 0.5, 0.8, 1, 2$. We also fixed $a = 2$, and investigated five different values of $b = 0.1, 0.5, 0.7, 2, 4$.

Tables D.1 and D.2 in this Web Appendix show that the results are insensitive to these choices of a and b, even with relatively larger values of σ_h^2 (smaller a or larger b).

Table D.1: Sensitivity results on the hyperparameter a for the number of samples $d_s = 2$ on 10 replications. A_{wc} is the area over the proportion of well-classified (WC) protein expression profiles, A_{mc} is the area under the proportion of misclassified (MC) protein expression profiles, and A_{nc} is the area under the proportion of non-classified (NC) protein expression profiles. Standard deviations are within parentheses

a	A_{nc}	A_{mc}	A_{wc}
0.3	0.072 (0.003)	0.023 (0.001)	0.905 (0.003)
0.5	0.073 (0.003)	0.023 (0.001)	0.904 (0.003)
0.8	0.072 (0.002)	0.023 (0.001)	0.905 (0.002)
1	0.072 (0.003)	0.023 (0.001)	0.904 (0.003)
2	0.073 (0.002)	0.023 (0.001)	0.904 (0.002)

Table D.2: Sensitivity results on the hyperparameter b for the number of samples $d_s = 2$ on 10 replications. A_{wc} is the area over the proportion of well-classified (WC) protein expression profiles, A_{mc} is the area under the proportion of misclassified (MC) protein expression profiles, and A_{nc} is the area under the proportion of non-classified (NC) protein expression profiles. Standard deviations are within parentheses

b	A_{nc}	A_{mc}	A_{wc}
0.1	0.072(0.002)	0.023(0.001)	0.906(0.002)
0.5	0.073(0.002)	0.023(0.001)	0.904(0.002)
0.7	0.071(0.001)	0.023(0.001)	0.906(0.002)
2	0.073(0.003)	0.023(0.001)	0.904(0.003)
4	0.073(0.003)	0.023(0.001)	0.904(0.003)

D.2 Prior for τ_{hl}

The parameter $\boldsymbol{\tau}$ is one of the key parameters in our approach since it truncates the effects, β_{hl}'s, to achieve clear discrimination between clusters. The choice of $\boldsymbol{\tau}$, which is interpreted as the minimum $\log_2$ fold-change, can be determined by the investigator.

In the paper, we imposed a gamma distribution with parameters a_τ and b_τ on this parameter. To study its sensitivity, we set $b_\tau = 10$ and varied a_τ to obtain different thresholds (on average). More specifically, we ran our approach on simulated data for $a_\tau = 2, 4, 6, 8, 10$, which resulted in respective prior means of 0.2, 0.4, 0.6, 0.8 and 1. Table D.3 in the Web Appendix provides the mean areas over ten replicates. This table shows that thresholds closer on average to the true β_{hl} obtain better results (larger A_{wc}, smaller A_{nc} and A_{mc}). Particularly, small values of a_τ (i.e., small values of the prior for τ_{hl}) lead to clusters that are less accurate. For instance, when $a_\tau = 2$ (i.e., on average, log2 fold-changes are larger than 0.2), we have larger numbers of both misclassified and non-classified features (proteins).

Table D.3: Sensitivity results on the hyperparameter a_τ for the number of samples $d_s = 2$ on 10 replications. A_{wc} is the area over the proportion of well-classified (WC) protein expression profiles, A_{mc} is the area under the proportion of misclassified (MC) protein expression profiles and A_{nc} is the area under the proportion of non-classified (NC) protein expression profiles. Standard deviations are within parentheses

a_τ	A_{nc}	A_{mc}	A_{wc}
2	0.188 (0.004)	0.164 (0.007)	0.648 (0.010)
4	0.138 (0.004)	0.064 (0.002)	0.798 (0.006)
6	0.088 (0.004)	0.026 (0.001)	0.885 (0.004)
8	0.071 (0.001)	0.023 (0.001)	0.906 (0.002)
10	0.104 (0.006)	0.046 (0.003)	0.850 (0.008)

D.3 Prior for π

We evaluated the sensitivity of our analysis to the choice of priors, $\pi_h's$, which control the inclusion proportions of the features (proteins) in each cluster. We kept the same expected proportion and varied α_h in order to change the variability of π_h. More specifically, we set the concentration parameters α_h of the Dirichlet distribution prior to five different values: $\alpha_h = 5, 10, 20, 25, 30$. The results presented in Table 3 in the Web Appendix show that our approach seems to be insensitive to the choice of variance of $\boldsymbol{\pi}$.

Table D.4: Sensitivity results on the hyperparameter α_h on 10 replications. A_{wc} is the area over the proportion of well-classified (WC) protein expression profiles, A_{mc} is the area under the proportion of misclassified (MC) protein expression profiles and A_{nc} is the area under the proportion of non-classified (NC) protein expression profiles. Standard deviations are within parentheses

α_h	A_{nc}	A_{mc}	A_{wc}
5	0.071(0.002)	0.024(0.001)	0.906(0.003)
10	0.074(0.002)	0.023(0.001)	0.903(0.002)
20	0.072(0.001)	0.024(0.002)	0.905(0.002)
25	0.072(0.003)	0.023(0.001)	0.905(0.003)
30	0.072(0.002)	0.023(0.001)	0.905(0.002)

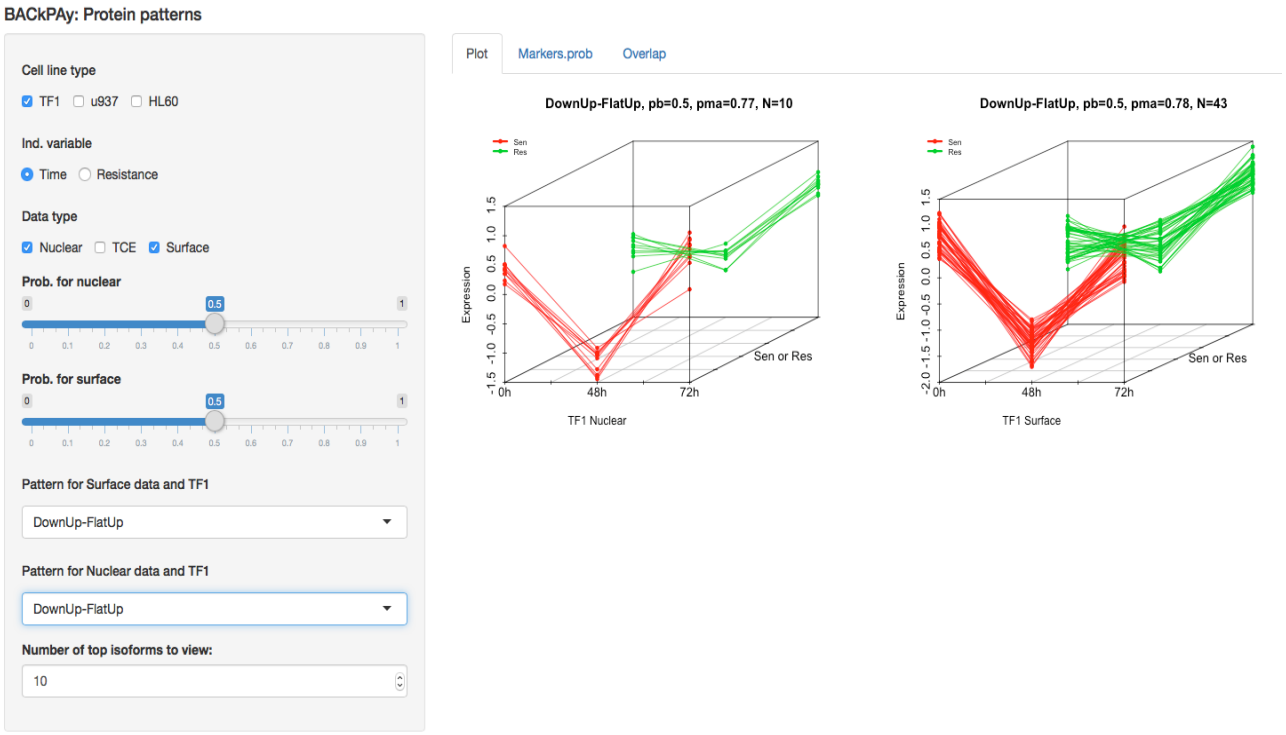
D.4 Prior for the random effect a_{jh}.

We also evaluated the robustness of our clustering results to the choices of c_h, the hyperparameter of the random effect a_{jh}. The results are presented in Table **??**, which shows that the results are insensitive to these choices of c_h, in particular when $c_h \geq 0.5$.

Table D.5: Sensitivity results on the hyperparameter c_h for the number of samples $d_s = 2$ on 5 replicates. A_{wc} is the area over the proportion of well-classified (WC) protein expression profiles, A_{mc} is the area under the proportion of misclassified (MC) protein expression profiles, and A_{nc} is the area under the proportion of non-classified (NC) protein expression profiles. Standard errors are within parentheses

c_h	A_{nc}	A_{mc}	A_{wc}
0.05	0.142 (0.043)	0.117 (0.106)	0.741 (0.133)
0.1	0.105 (0.026)	0.087 (0.063)	0.809 (0.079)
0.2	0.093 (0.017)	0.075 (0.033)	0.832 (0.04)
0.5	0.072 (0.005)	0.023 (0.004)	0.905 (0.006)
0.7	0.071 (0.005)	0.024 (0.004)	0.905 (0.005)
1	0.068 (0.004)	0.023 (0.003)	0.909 (0.004)

Figure E.1: Graphical user interface in the web browser. "pb=0.5" means the selection of proteins in the corresponding group was obtained with a threshold of 0.5. "pmax=0.77" means the (joint) maximum marginal posterior probabilities of inclusion (jMPP) for this group is 0.77. "N=10" means the number of isoforms in this group is 10. "Sen" stands for sensitive, "Res" for resistant.



E Shiny app

In this section, we provide a detailed description of the visualization tools introduced in Section 5.3 of the main text. A screen shot of the Shiny application is presented in Figure E.1 in the Web Appendix.

Our sidebar panel for inputs consists of the following elements: (i) *Cell line type*, which consists of the three cell line types: TF1, u937 and HL60; (ii) *Ind. variable* is either *Time* or *Resistance*. It is *Time* when we compare the protein expression across all the time points in sensitive and resistant cell samples, and *Resistance* when the comparison is between sensitive and resistant samples at different time points; (iii) *Data type* is *Nuclear*, *TCE* or *Surface*; (iv) *Prob. for Data type and cell line type* represents possible values of jMPP; (v) *Pattern for Data type and Cell line type* represents the list of possible pattern groups obtained with a threshold value of the jMPP; (vi) *Number of isoforms to view* shows the maximum number of protein isoforms to view in the output panels *Markers.prob* and *Overlap*.

Our main panel for outputs can be categorized in three groups: (i) the *Plot* tab shows 3-D plots of the observed data in a specific cluster of proteins; (ii) tab *Markers.prob* shows a list of proteins with their corresponding encoded genes and jMPP; and (iii) the *Overlap* tab shows the list of common proteins identified between different types of data, or cell lines, with their corresponding encoded genes and jMPP.

F Comparison of clustering performance

Clustering methods such as K-means (Hartigan and Wong, 1979), hierarchical agglomerative clustering (Hartigan, 1975), model-based clustering (Fraley and Raftery, 2002), and self-organizing map (SOM) (Tamayo et al., 1999) algorithms can also be used to determine expression patterns in time-course data. However, these "standard" clustering algorithms have the following limitations in the context of our goals.

1. Standard clustering methods are unconstrained. We constrain our Bayesian clustering method to give direct and explicit biological interpretation of each cluster with respect to the sign of the effect, which corresponds to a pattern of interest. That is, each cluster can be identified/defined by the sign of the regression coefficients and has a specific interpretation depending on the goal of the analysis (Objective 1 or 2).

2. Our approach is able to find proteins that are differentially expressed between two experimental conditions over time, a feature which is not well defined in standard clustering algorithms.

3. Clustering algorithms could not be directly applied to our cell line data; since we cluster pairs (j, k) (i.e., pairs of protein j's – condition k's), the sample size, n_k, may not be the same between experiments k's. For instance, for cell line HL60, we have $n_1 = 2$ sensitive samples (one at time 0 and another at time 72h) while we have $n_2 = 4$ resistant cell line samples (2 at both time 0 and 72h). This shows another aspect of our clustering approach, which is that it can be applied to data with an unbalanced number of replicates between experiments, k's.

4. Standard clustering methods do not account for duplicate samples as they treat duplicates as "new" samples. Of course, duplicates can be averaged before applying clustering, but this will reduce the power to detect differential proteins.

Despite the limitations of the "standard" clustering algorithms, we have applied them to the simulated data presented in Section 6 in the paper. Note that in this dataset, the number of resistant samples $n_1 = 3 * d_1$ equals n_2, the number of sensitive samples. We used the clustering algorithms itemized hereafter.

- The standard k-means algorithm (Hartigan and Wong, 1979). We fixed the number of clusters at $k = H = 9$, the true number of clusters. We implemented the algorithm using the function *kmeans* of R.

- The hierarchical agglomerative clustering algorithm (Hartigan, 1975) with the complete linkage method. We implemented the algorithm using the function *hclust* in R. We cut the tree at the true number of clusters, $H = 9$.

- A model based-mixture clustering (Fraley and Raftery, 2002) implemented with the R package *mclust*. The clusters are spherical but have different volumes ("VII"). We used the Bayesian information criterion to choose the number of mixture components $G = 1, ..., 12$.

- The self-organizing map (with application in gene clustering) algorithm (Tamayo et al., 1999). We implemented this algorithm in the R package *som*, with one as the x-dimension of the map and $H = 9$ as the y-dimension.

We computed the adjusted Rand index (ARI) (Hubert and Arabie, 1985) to compare the clustering results. The results are presented in Table F.1. Overall, the proposed approach performed comparatively well.

Table F.1: Simulation results. Comparison of BACkPAy with standard clustering algorithms. ARI is the adjusted Rand index over ten replicates. Standard deviations are within parentheses.

Method	β_{hl}	d_s	ARI
BACkPAy	0.5	1	0.353 (0.005)
Hclust	0.5	1	0.248 (0.013)
K-means	0.5	1	0.395 (0.026)
Mclust	0.5	1	0.396 (0.009)
SOM	0.5	1	0.381 (0.02)
BACkPAy	0.8	1	0.702 (0.007)
Hclust	0.8	1	0.53 (0.045)
K-means	0.8	1	0.731 (0.072)
Mclust	0.8	1	0.748 (0.02)
SOM	0.8	1	0.648 (0.08)
BACkPAy	1	1	0.857 (0.005)
Hclust	1	1	0.745 (0.058)
K-means	1	1	0.78 (0.053)
Mclust	1	1	0.865 (0.011)
SOM	1	1	0.685 (0.073)
BACkPAy	0.5	2	0.588 (0.008)
Hclust	0.5	2	0.376 (0.035)
K-means	0.5	2	0.59 (0.048)
Mclust	0.5	2	0.563 (0.007)
SOM	0.5	2	0.541 (0.006)
BACkPAy	0.8	2	0.918 (0.003)
Hclust	0.8	2	0.784 (0.083)
K-means	0.8	2	0.756 (0.046)
Mclust	0.8	2	0.855 (0.006)
SOM	0.8	2	0.734 (0.088)
BACkPAy	1	2	0.978 (0.002)
Hclust	1	2	0.978 (0.018)
K-means	1	2	0.823 (0.003)
Mclust	1	2	0.893 (0.005)
SOM	1	2	0.78 (0.047)
BACkPAy	0.5	3	0.75 (0.008)
Hclust	0.5	3	0.487 (0.04)
K-means	0.5	3	0.655 (0.063)
Mclust	0.5	3	0.67 (0.008)
SOM	0.5	3	0.59 (0.067)
BACkPAy	0.8	3	0.974 (0.001)
Hclust	0.8	3	0.856 (0.061)
K-means	0.8	3	0.713 (0.077)
Mclust	0.8	3	0.878 (0.004)
SOM	0.8	3	0.732 (0.053)
BACkPAy	1	3	0.994 (0.001)
Hclust	1	3	0.975 (0.029)
K-means	1	3	0.838 (0.002)
Mclust	1	3	0.90 (0.01)
SOM	1	3	0.828 (0.058)
BACkPAy	0.5	4	0.854 (0.006)
Hclust	0.5	4	0.51 (0.066)
K-means	0.5	4	0.694 (0.06)
Mclust	0.5	4	0.734 (0.006)
SOM	0.5	4	0.602 (0.08)
BACkPAy	0.8	4	0.991 (0.001)
Hclust	0.8	4	0.842 (0.104)
K-means	0.8	4	0.70 (0.065)
Mclust	0.8	4	0.893 (0.002)
SOM	0.8	4	0.745 (0.008)
BACkPAy	1	4	0.998 (0.00)
Hclust	1	4	0.99 (0.017)
K-means	1	4	0.821 (0.048)
Mclust	1	4	0.902 (0.009)
SOM	1	4	0.818 (0.003)

G Performance of BACkPAy for detecting differential features in larger samples

We ran two additional scenarios with $d_t = 10$ and $d_t = 15$. The results, presented in Table G.1, show that LIMMA and our BACkPAy method outperform the two other methods mostly when the effect size is small. The table also shows that our approach performs very well with a large sample size. We note that it is very unlikely to have that many replicates in cell line experiments.

Table G.1: Simulation results with large numbers of samples per group, d_t. AUC is the average over five replicates of the area under the ROC curve. Standard deviations are within parentheses.

Method	d_t	$\lvert\beta_{hl}\rvert$	AUC
EDGE	10	0.5	0.951 (0.004)
MB-Statistics	10	0.5	0.753 (0.02)
LIMMA	10	0.5	1 (0)
BACkPAy	10	0.5	1 (0)
EDGE	15	0.5	0.978 (0.002)
MB-Statistics	15	0.5	0.762 (0.01)
LIMMA	15	0.5	1 (0)
BACkPAy	15	0.5	1 (0)
EDGE	10	0.8	0.972 (0.003)
MB-Statistics	10	0.8	0.786 (0.018)
LIMMA	10	0.8	1 (0)
BACkPAy	10	0.8	1 (0)
EDGE	15	0.8	0.989 (0.002)
MB-Statistics	15	0.8	0.781 (0.008)
LIMMA	15	0.8	1 (0)
BACkPAy	15	0.8	1 (0)

H R codes for implementation of the competing methods

We provide a detailed description of our implementation of the competing methods and include the *R* code we used to fit these methods. Most of these methods can only be applied if we have at least 2 replicates for each time point.

1. MB-Statistics

```
#times gives the number of distinct time points (it is 3 in our case)
#timepoint is coded as 0, 1 and 2 indicating time 0h, 48h and 72h respectively
#cond is a binary variable indicating if a sample is  sensitive or resistant
#reps is matrix that gives the number of replicates for each
#experimental group
MB.2D <- mb.long(data, method="2", times=times,
time.grp = timepoint, reps=reps, condition.grp=cond)
rhoest=MB.2D$HotellingT2
```

The method is "2" since proteins of interest are those with different expected temporal profiles across two biological conditions (resistant and sensitive samples). Important proteins are obtained by ranking the Hotelling $\tilde{T}^2$, and AUC's are computed using that ranking.

2. LIMMA method

```
#cond is a binary variable indicating if a sample is  sensitive or resistant
#timepoint is coded 0, 1 and 2 indicating time 0h, 48h and 72h
f<-as.factor(paste(cond,timepoint,sep="."))
design <- model.matrix(~0+f)
fit <- lmFit(data, design)
cont.dif <- makeContrasts(Diff1=(f0.0-f1.0), Diff2=(f0.1-f1.1),
Diff3=(f0.2-f1.2),levels=design)
fit2 <- contrasts.fit(fit, cont.dif)
fit3 <- eBayes(fit2)
rhoest=1-fit3$F.p.value
```

Important proteins can be obtained by ranking 1-pvalue, and AUC's are computed using that ranking.

3. EDGE method

```
#ind is the individual factor for repeated observations of the same individuals
#cond is a binary variable indicating if a sample is sensitive or resistant
#timepoint is coded 0, 1 and 2 indicating time 0h, 48h and 72h
full_model<-~grp + ns(tme, df = 1, intercept = FALSE) +
(grp):ns(tme, df = 1, intercept = FALSE)
null_model<-~ns(tme, df = 1, intercept = FALSE)
cov=data.frame(cbind(individual = ind, tme = time, grp = class))
rownames(cov)=colnames(data)
de_obj <- build_models(data, cov = cov, full.model =full_model,
null.model = null_model)
de_lrt <- lrt(de_obj,bs.its = 50,lambda=seq(0,0.9,0.01))
sig_results <- qvalueObj(de_lrt)
qvalues <- sig_results$qvalues
rhoest=1-qvalues
```

Important proteins can be obtained by ranking 1-qvalue, and AUC's are computed using that ranking.

Note that for all the competing methods, the alternative hypothesis is H_1: *protein responds differentially over time in the sensitive relative to the resistant sample.* That is, a protein is differentially expressed (DE) if there is at least one time point when it is differentially expressed between sensitive and resistant samples. Hence, a protein is non-DE when its expression is the same between resistant and sensitive samples at each time point.

I Comparison between BACkPAy and the simple cut-off method

Since the sample size is extremely small for all scenarios considered in this manuscript, a simpler alternative to our approach may consist in directly setting experience-driven cutoff values, e.g., for the log fold change, to determine differential expressed proteins. We showed here that using a simple cutoff provides worst results than our Bayesian methods.. For that,

- We have simulated 1000 protein expressions expressed between 2 modalities of both the experimental and the independent variables. We chose 2 as the number of samples in each group (obtained from the combination of the experimental and independent variables), which gives a total of 2×2×2=8 samples (see the R code below for more information). We varied the slope in each cluster such as every protein has its own slope β_{jh} generated as uniform (0.5,1) in absolute values.

- We have first selected differential expressed proteins based only on setting cut-off values between modalities 0 and 1 of the independent variable for each modality of the experimental variable. Cut-offs were set to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8. 0.9, 1. We call this approach *Simple cutt-off*.

- We have then selected differential expressed proteins using our BACkPAy method by using a threshold of 0.5 on marginal posterior probabilities of selecting DEs. We set the hyperparameters $b_\tau = 20$ and $a_\tau \in \{0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2\}$, which correspond to an average ($\log_2$-) fold change of at least $a_\tau/20 \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8.0.9, 1\}$ (i.e $E(\beta_{hl}) \geq E(\tau_{hl}) = a_\tau/b_\tau$) on the prior distribution. We note that this is not a "hard" constraint on the prior of the effect β_{hl} as its truncated value τ_{hl} is also considered as a parameter (i.e its estimated value is not only guided by the investigator belief but also the data).

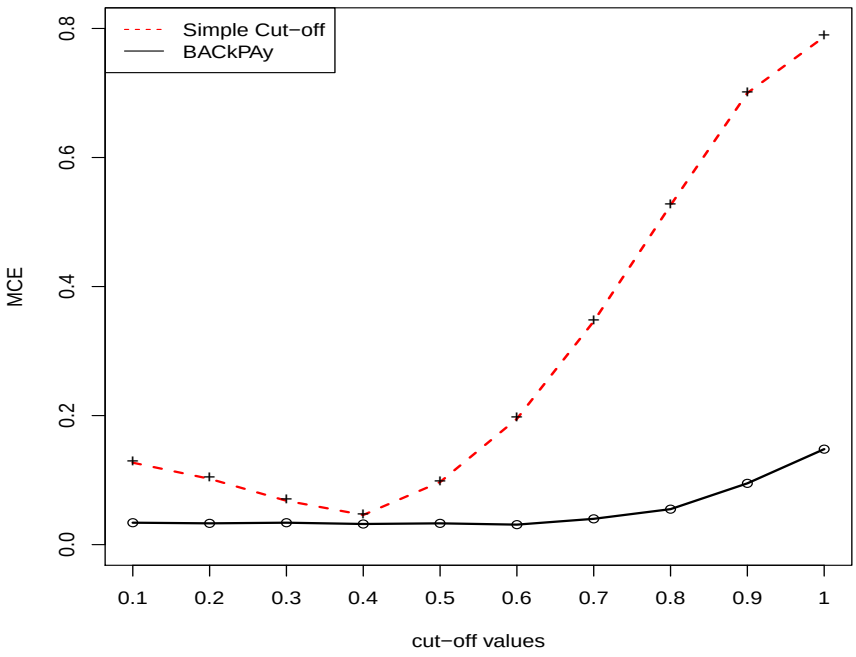
We implemented both the *Simple cutt-off* and BACkPAy methods as described above. We computed the misclassification error (MCE) of detecting DE proteins for both methods (see Figure I.1). The figure shows that BACkPAy is much less sensitive for the choice of a_τ. Moreover, the AUC for selecting DE was all always >0.98. It can be risky to just apply a cut-off to detect DE proteins as it does not account for the variability of data. With the cutt off of 0.5, the minimum effect value used to generate the data, we obtained an MCE of 0.115 with the simple cut-off method. The results are worst for larger cut-offs.

J Unbalanced cluster sizes: simulation results

In this Section, we studied the ability of our method to identify clusters with unbalanced sizes. We have performed a sensitivity analysis on the simulated data by using different proportions between the clusters. Specifically, we generated data as explained in Section 5.4 but with four different scenarios by varying the proportions of elements ((j, k)'s) in clusters $h = 1, 2, 3$:

- S1=(1/3,1/3,1/3);

Figure I.1: A comparison with a simple cut-off method



- S2=(0.2,0.5,0.3);
- S3=(0.15,0.6,0.25);
- S4=(0.08,0.8,0.12);

Cluster 2 ("zero" cluster) elements have no effect on the explanatory variable, and we would expect it to be larger than the others in practice. To run BACkPAy, we have used a common $\alpha_h = 20$ as specified in Section 5.1 of the manuscript. Results in Table J.1 show that our method is insensitive to the proportions between clusters in terms of detecting DE proteins. Moreover, Table J.2, which shows the clustering performance of our method, confirms this result. BACkPAy seems to have better clustering performance with larger number of elements in the "zero" cluster (cluster with zero effect, $\beta_2 = 0$), which can be explained by the ability of our approach to better discriminate clusters through truncated distributions imposed on the slopes β's.

K A simulation study to investigate the effect of the signal-to-noise ratio

First, we have generated a new set of data sets (20 replicates) by increasing the standard deviation in each cluster to $\sigma_h = 0.6$ (from 0.2). We have now more variability in the results, which are overall worse than the results presented in Table 2 of the paper. However, we still observe the same trend, with LIMMA and our method performing similarly in terms of detection DE when the number of samples in each group (d_t) is larger than 1. However, BACkPAy is the only method that performs reasonably in

Table J.1: Simulation results for detecting DE proteins with respect to 4 scenarios of different cluster proportions, the number of duplicates d_t, and the cluster standard deviation σ_h. AUC represents the AUC for detecting DE proteins.

Proportions	d_t	AUC
S1	1	0.957 (0.007)
S2	1	0.930 (0.0055)
S3	1	0.922 (0.002)
S4	1	0.905 (0.0065)
S1	2	0.990 (0.001)
S2	2	0.984 (0.0015)
S3	2	0.982 (0.001)
S4	2	0.975 (0.002)

Table J.2: Simulation results for clustering performance with respect to 4 scenarios of different cluster proportions, the number of duplicate samples d_t. A_{wc} is the area over the proportion of well-classified (WC) protein expression profiles, A_{mc} is the area under the proportion of misclassified (MC) protein expression profiles, and A_{nc} is the area under the proportion of non-classified (NC) protein expression profiles. Standard errors are within parentheses

Proportions	d_t	A_{nc}	A_{mc}	A_{wc}
S1	1	0.082 (0.004)	0.125 (0.004)	0.793 (0.007)
S2	1	0.077 (0.003)	0.116 (0.005)	0.808 (0.007)
S3	1	0.075 (0.005)	0.11 (0.004)	0.815 (0.007)
S4	1	0.081 (0.004)	0.102 (0.005)	0.817 (0.007)
S1	2	0.062 (0.004)	0.062 (0.003)	0.876 (0.005)
S2	2	0.054 (0.003)	0.059 (0.005)	0.887 (0.006)
S3	2	0.046 (0.002)	0.05 (0.004)	0.904 (0.005)
S4	2	0.032 (0.002)	0.033 (0.002)	0.936 (0.003)

the case of one duplicate ($d_t = 1$). LIMMA cannot run when $d_t = 1$ as it does not have enough degrees of freedom to estimate the variance.

In addition to this scenario, we have also generated additional simulated data from a model with a fixed slope (i.e $|\beta_h| = 1$) and a range of cluster standard deviations $\sigma_h \in \{0.4, 0.6, 0.8, 1, 1.2\}$, which correspond to the following signal-to-noise ratios: 2.5, 1.666, 1.25,1, 0.833. The rest of simulation settings is as in Section 5.4. The results are presented in Table K.2, which again shows similar performances of the two approaches (BACkPAy and LIMMA). However, our method results in smaller standard errors.

Table K.1: Simulation results for detecting DE proteins. Data are generated with $\sigma_h = 0.6$. AUC stands for the average over 20 replicates of the area under the ROC curve. Standard deviations are within parentheses.

Method	Replicates	Beta	AUC	Replicates	Beta	AUC
EDGE	1	0.5	0.5 (0.016)	3	0.5	0.608 (0.015)
MB-Statistics	1	0.5	NA (NA)	3	0.5	0.553 (0.019)
LIMMA	1	0.5	NA (NA)	3	0.5	0.676 (0.015)
BACkPAy	1	0.5	0.566 (0.018)	3	0.5	0.672 (0.014)
EDGE	1	0.8	0.507 (0.019)	3	0.8	0.704 (0.013)
MB-Statistics	1	0.8	NA (NA)	3	0.8	0.589 (0.018)
LIMMA	1	0.8	NA (NA)	3	0.8	0.832 (0.011)
BACkPAy	1	0.8	0.646 (0.014)	3	0.8	0.828 (0.010)
EDGE	1	1	0.512 (0.019)	3	1	0.752 (0.011)
MB-Statistics	1	1	NA (NA)	3	1	0.606 (0.017)
LIMMA	1	1	NA (NA)	3	1	0.904 (0.007)
BACkPAy	1	1	0.706 (0.012)	3	1	0.900 (0.008)
EDGE	2	0.5	0.569 (0.015)	4	0.5	0.646 (0.015)
MB-Statistics	2	0.5	0.565 (0.014)	4	0.5	0.584 (0.012)
LIMMA	2	0.5	0.63 (0.018)	4	0.5	0.727 (0.014)
BACkPAy	2	0.5	0.628 (0.019)	4	0.5	0.720 (0.014)
EDGE	2	0.8	0.637 (0.014)	4	0.8	0.755 (0.012)
MB-Statistics	2	0.8	0.641 (0.013)	4	0.8	0.657 (0.011)
LIMMA	2	0.8	0.765 (0.014)	4	0.8	0.885 (0.009)
BACkPAy	2	0.8	0.762 (0.016)	4	0.8	0.880 (0.009)
EDGE	2	1	0.676 (0.013)	4	1	0.802 (0.011)
MB-Statistics	2	1	0.69 (0.012)	4	1	0.697 (0.011)
LIMMA	2	1	0.842 (0.011)	4	1	0.944 (0.005)
BACkPAy	2	1	0.840 (0.013)	4	1	0.939 (0.005)

References

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association 97*(458), 611–631.

Hartigan, J. A. (1975). *Clustering Algorithms* (99th ed.). New York, NY, USA: John Wiley & Sons, Inc.

Hartigan, J. A. and M. A. Wong (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 28*(1), 100–108.

Hubert, L. and P. Arabie (1985, Dec). Comparing partitions. *Journal of Classification 2*(1), 193–218.

Storey, J. D., J. T. Leek, and A. J. Bass (2015). *edge: Extraction of Differential Gene Expression*. R package version 2.6.0.

Table K.2: Simulation results for detecting DE proteins with respect to different values of σ_h. Data are generated with slopes $|\beta_h| = 1$ and according to Section 6.1. AUC stands for the average over 20 replicates of the area under the ROC curve. Standard errors are within parentheses.

Method	Replicates (d_t)	σ_h	AUC
LIMMA	3	0.4	0.981 (0.002)
BACkPAy	3	0.4	0.981 (0.001)
LIMMA	3	0.6	0.905 (0.006)
BACkPAy	3	0.6	0.905 (0.002)
LIMMA	3	0.8	0.811 (0.01)
BACkPAy	3	0.8	0.811 (0.003)
LIMMA	3	1	0.733 (0.012)
BACkPAy	3	1	0.734 (0.004)
LIMMA	3	1.2	0.677 (0.013)
BACkPAy	3	1.2	0.675 (0.003)
LIMMA	4	0.4	0.992 (0.001)
BACkPAy	4	0.4	0.993 (0.001)
LIMMA	4	0.6	0.943 (0.007)
BACkPAy	4	0.6	0.942 (0.002)
LIMMA	4	0.8	0.864 (0.012)
BACkPAy	4	0.8	0.862 (0.004)
LIMMA	4	1	0.787 (0.015)
BACkPAy	4	1	0.784 (0.005)
LIMMA	4	1.2	0.726 (0.016)
BACkPAy	4	1.2	0.722 (0.005)

Storey, J. D., W. Xiao, J. T. Leek, R. G. Tompkins, and R. W. Davis (2005). Significance analysis of time course microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America 102*(36), 12837–12842.

Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences 96*(6), 2907–2912.