UNIVERSITÉ DE STRASBOURG

ECOLE DOCTORALE MATHEMATIQUES SCIENCES DE L'INFORMATION ET DE L'INGENIEUR (ED 269)

Institut de Recherche Mathématique Avancée, UMR 7501

Institut Pluridisciplinaire Hubert Curien, UMR 7178

THÈSE présentée par :

Marie CHION

soutenue le : **16 décembre 2021**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Mathématiques appliquées

Développements de nouvelles méthodologies statistiques pour l'analyse de données de protéomique quantitative

THÈSE dirigée par :

M. BERTRAND Frédéric	Professeur des Universités, Université de Technologie de Troyes
Mme CARAPITO Christine	Chargée de Recherches, CNRS – Université de Strasbourg

RAPPORTEURS :

M. BURGER Thomas	Directeur de recherches, CNRS – Université Grenoble Alpes
Mme VIALANEIX Nathalie	Directrice de recherches, INRAE Toulouse

---

AUTRES MEMBRES DU JURY :

M. BIRMELE Etienne	Professeur des Universités, Université de Strasbourg
M. THEVENOT Etienne	Ingénieur Chercheur, CEA Saclay

Institut de Recherche Mathématique Avancée, UMR 7501
7, rue René Descartes
67084 Strasbourg Cedex, France

Institut Pluridisciplinaire Hubert Curien, UMR 7178
25, rue Becquerel
67087 Strasbourg Cedex, France

To everyone who has ever felt like they are not good enough,
nor worth it.

"Science is a social process: it is not about lone geniuses. Most science is incremental and communal. It is not about huge leaps or flashes of insight. It is usually about putting together existing pieces in a new way, and small insights and results build up to larger understandings. You are a member of the scientific community. You earned your place here, and you have already arrived." (Chris Moore)

Résumé de la thèse

L'analyse protéomique consiste à étudier le protéome, à savoir l'ensemble des protéines exprimées par un système biologique donné, à un moment donné et dans des conditions données. La spectrométrie de masse (MS) et la chromatographie liquide (LC) ont connu une véritable révolution instrumentale ces vingt dernières années, qui permet d'analyser aujourd'hui des protéomes complexes et d'identifier et de quantifier plusieurs milliers de protéines en quelques heures d'analyses LC-MS/MS. La complexité croissante des données MS massives ainsi générées, a naturellement suscité la nécessité de développer des outils et des méthodologies statistiques adaptés et dédiés à l'interprétation de ces données. Ces développements sont capitaux pour permettre d'envisager des études protéomiques à plus large échelle et à haut débit. L'objectif de cette thèse est de développer de nouvelles méthodologies pour l'analyse statistique des données de protéomique quantitative.

Développement d'une méthodologie d'estimation de quantités absolues de peptides à partir de données acquises en mode data-independent acquisition

En mode *data-independent acquisition* (DIA) (Gillet et al., 2012; Ludwig et al., 2018), l'ensemble de la gamme de masse est couverte pour acquérir une carte de fragmentation complète des protéomes étudiés. Le spectromètre de masse acquiert des spectres de fragmentation à partir de larges fenêtres de masse consécutivement isolées pour générer des spectres MS/MS multiplexes. La quantification des peptides se fait ensuite en MS/MS, ce qui permet une quantification plus précise et plus spécifique qu'en MS, tel que c'est le cas en mode DDA.

Contexte

Cette première partie de mon travail de thèse a été réalisée dans le cadre d'une collaboration avec le Dr. Muriel Bonnet (UMR Herbivores, INRA, Clermont-Ferrand) (Bonnet et al., 2020), au cours de laquelle 64 échantillons de muscles bovins pour lesquels 20 peptides correspondant aux 10 protéines potentiels biomarqueurs pour la tendreté et le persillage de la

viande de boeuf ont été analysés par une méthode DIA. Une première étape de quantification ciblée couplée à la dilution isotopique utilisant des peptides synthétiques marqués, a permis de déterminer la quantité absolue des 20 peptides d'intérêt au sein de chacun des 64 échantillons considérés. Pour ce faire, la relation suivante a été utilisée :

$$\text{Quantité du peptide} = \frac{\text{Quantité du peptide synthétique}}{\text{Intensité du peptide synthétique}} \times \text{Intensité du peptide.}$$

Une seconde étape de quantification globale a permis de mesurer l'intensité de près de 5500 peptides dans les 64 échantillons considérés. En protéomique quantitative, une hypothèse forte est faite, selon laquelle la quantité d'un peptide est proportionnelle à son intensité au travers d'un facteur de réponse. Celui-ci est spécifique au peptide et à l'échantillon considérés. Ainsi :

$$\text{Quantité du peptide} = \text{Facteur de réponse} \times \text{Intensité du peptide.}$$

L'objectif ici a été de tirer partie des données des deux méthodes de quantification. A partir des données d'intensité et de quantité obtenues en quantification ciblée grâce à des peptides standards internes marqués sur un sous-ensemble de peptides, il s'agit d'ajuster un modèle de lissage par spline monotone, expliquant la quantité d'un peptide par son intensité dans l'échantillon considéré. Ce modèle a permis ensuite d'estimer les quantités pour l'ensemble des peptides dont les intensités ont été mesurées durant l'analyse en mode DIA.

Lissage par spline monotone

La méthode de lissage par spline monotone combine la régression par I-spline (Ramsay, 1988) avec l'estimation des paramètres par la méthode des moindres carrés non négatifs, à l'aide de l'algorithme de Lawson-Hanson par exemple (Lawson and Hanson, 1995). Dans ce travail, les modèles utilisés sont des combinaisons linéaires de I-splines, tels que :

$$f(x) = \sum_i a_i I_i(x|k,t),$$

où les a_i sont les paramètres à estimer et les $\{I_i\}_i$ constituent une base de fonctions I-splines. Une fonction I-spline s'écrit comme l'intégrale d'une fonction M-spline (fonction non-négative polynomiale par morceaux) :

$$I_i(x|k,t) = \int_L^x M_i(u|k,t)\,du,$$

où k est le degré de la I-spline et L est la borne inférieure du domaine.

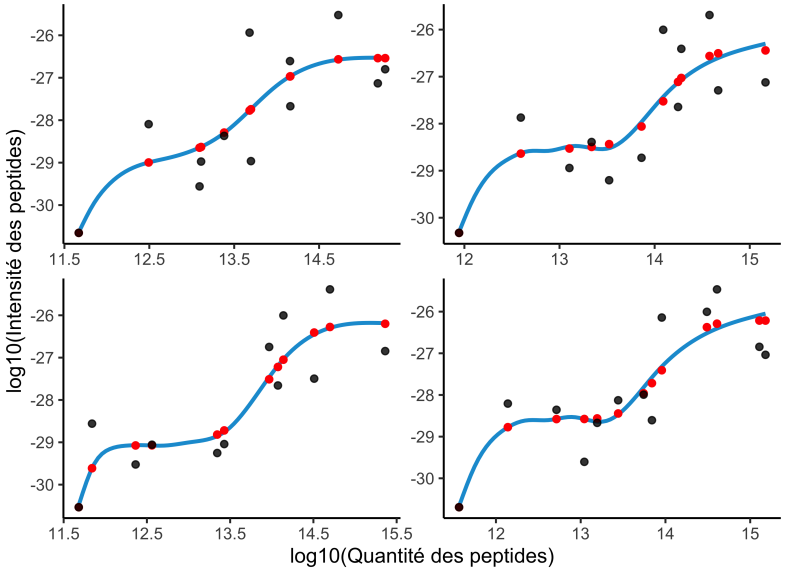


Figure 1: Lissage par spline monotone sur 4 des 64 échantillons. Les points en noir représentent les valeurs utilisées pour l'ajustement du modèle, les points en rouges sont les valeurs prédites par le modèle.

Résultats

Un modèle de lissage par spline monotone a ainsi été ajusté pour chacun des 64 échantillons. Un extrait de représentation graphique est présenté à la Figure 1. Les performances des modèles de lissage par spline monotone ont été comparées à celles du modèle linéaire, au travers de la racine de l'erreur quadratique moyenne (RMSE). La Figure 2 illustre une meilleure qualité d'ajustement aux données de la spline monotone en comparaison au modèle linéaire. Les performances en termes de prédiction ont été évaluées à partir des quantités absolues des protéines d'intérêt. Près de 53% des estimations des quantités varient dans un rapport de 2 par rapport aux quantités issues de la quantification ciblée et près de 80% des échantillons présentent une grande cohérence entre les deux méthodes. Les estimations des 5500 peptides ont ensuite été interprétées biologiquement et se sont avérées être fidèles à la littérature scientifique et conforment aux attendus sur le protéome du muscle bovin.

Une perspective utilisant le cadre probabiliste et non parametriques des processus Gaussiens a également été proposé. Cette approche aboutit à une amélioration des performances prédictives associées à une quantification de l'incertitude en tout point.

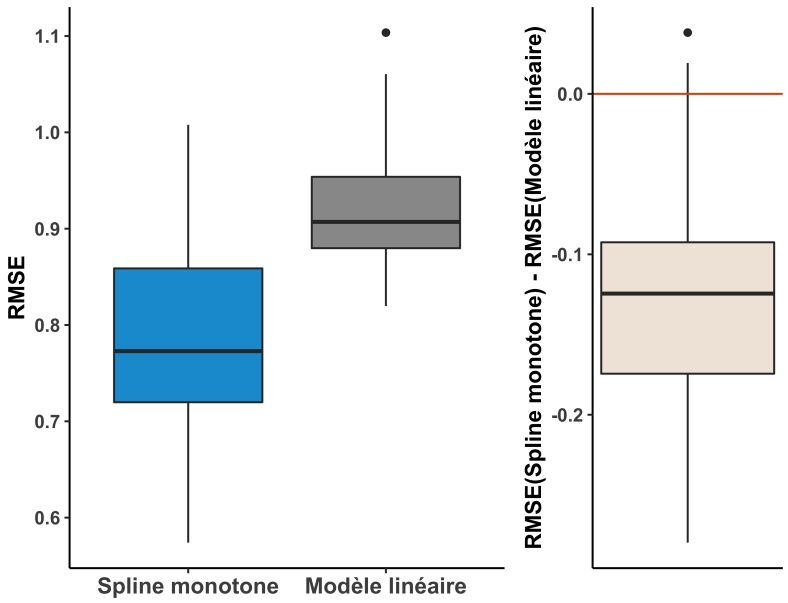


Figure 2: Comparaison des RMSE du modèle de spline monotone et du modèle linéaire. Le panneau de gauche décrit les distributions des RMSE pour la spline monotone en bleu et celle des RMSE pour le modèle linéaire en gris foncé. Le panneau de droite décrit la différence entre les RMSE de la spline monotone et ceux du modèle linéaire.

Développement d'une méthodologie de prise en compte de la variabilité pour l'imputation multiple de données de protéomique quantitative

Contexte

En mode *data-dependent acquisition* (DDA), le spectromètre de masse génère dans une première étape les spectres MS pour tous les peptides. Les peptides les plus intenses sont ensuite sélectionnés pour générer leurs spectres MS/MS. La quantification des peptides se fait en extrayant l'aire sous la courbe du pic chromatographique obtenu en MS. En protéomique quantitative, les valeurs manquantes peuvent être d'origines biochimiques, analytiques, ou bioinformatiques. Dans les principaux logiciels d'analyse statistique pour les données de protéomique quantitative, il est notamment proposé d'imputer ces valeurs manquantes. Ainsi, les logiciels Perseus (Tyanova et al., 2016), MSstats (Choi et al., 2014) et ProStaR (Wieczorek et al., 2017) proposent des méthodes d'imputation simple. Or, cette approche consiste à remplacer les valeurs manquantes une seule fois et considérer par la suite le jeu de données comme ayant toujours été complet. Il n'est donc pas tenu compte de la variabilité liée au processus aléatoire d'imputation. Des méthodes d'imputation simple améliorée sont également disponibles dans ProStaR (Giai Gianetto et al., 2020) et PANDA-view (Chang et al.,

2018). Or, il s'avère qu'en pratique, dans les logiciels mentionnés, les jeux de données imputés sont combinés pour n'obtenir qu'un jeu de données final, considéré par la suite comme ayant toujours été complet. Bien que le biais de l'estimateur des paramètres obtenu après cette imputation simple améliorée soit plus faible en valeur absolue qu'après une imputation simple usuelle, la variabilité liée au processus d'imputation n'est pas explicitement prise en compte.

Prise en compte de l'incertitude liée à l'imputation multiple

Dans cette deuxième partie de mon travail de thèse, j'ai d'abord implémenté une méthode rigoureuse d'imputation multiple, en suivant les règles de Rubin (Little and Rubin, 2019) (Figure 3).

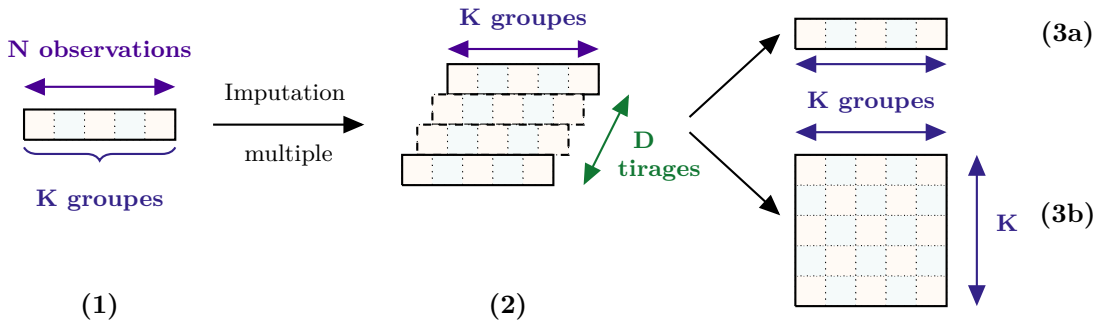


Figure 3: Méthodologie d'imputation multiple. **(1)** Jeu de données initial contenant des valeurs manquantes, avec N observations réparties dans $groupI$ groupes. **(2)** L'imputation multiple renvoie D estimateurs pour le vecteur de paramètres d'intérêt. **(3a)** Les D estimateurs sont combinés grâce à la première règle de Rubin pour obtenir l'estimateur combiné. **(3a)** L'estimateur de la matrice de variance-covariance de l'estimateur combiné est donné par la deuxième règle de Rubin.

Soit $\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}}$ l'estimateur du vecteur de paramètres d'intérêt $\hat{\boldsymbol{\beta}}_{\boldsymbol{p}}$ pour l'analyte p, obtenu par le d-ème jeu de données imputé et W_d la matrice de variance-covariance de $\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}}$. Les D estimateurs, correspondant aux D imputations, du vecteur de paramètres d'intérêt sont combinés pour obtenir l'estimateur combiné selon la première règle de Rubin :

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{p}} = \frac{1}{D}\sum_{d=1}^{D}\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}}.$$

La deuxième règle de Rubin permet d'obtenir l'estimateur combiné de la matrice de variance-covariance de l'estimateur combiné du vecteur de paramètres. Celui-ci prend en compte à la fois la variabilité intra-imputation et la variabilité inter-imputation (illustrant la variabilité due aux valeurs manquantes) selon l'équation suivante :

$$\hat{\boldsymbol{\Sigma}}_p = \frac{1}{D}\sum_{d=1}^{D} W_d + \frac{D+1}{D(D-1)}\sum_{d=1}^{D}(\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})^T(\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}).$$

Cet estimateur de la matrice de variance-covariance est ensuite projeté pour obtenir un paramètre univarié de variabilité. Cette variance est ensuite modérée selon un modèle hiérarchique bayésien (Smyth, 2004) pour construire la statistique du test t-modéré (Phipson et al., 2016) telle que :

$$T_{pj[\text{mod}]} = \frac{\hat{\beta}_{pj}}{\sqrt{\hat{\sigma}^2_{p[\text{mod}]}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}_{j,j}}},$$

où:

- $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}_{j,j}$ est le j-ème élément diagonal de la matrice $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$.
- $\hat{\sigma}^2_{p[\text{mod}]}$ est l'estimateur modéré de σ^2_p.

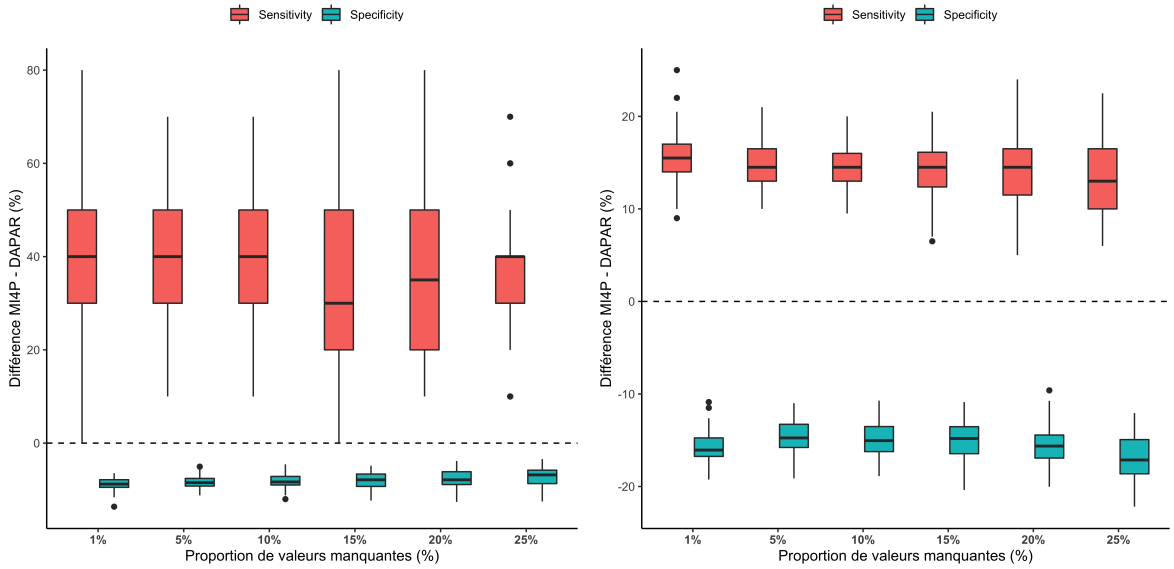
Sous l'hypothèse nulle, $\mathcal{H}_0$, $T_{pj[\text{mod}]}$ suit une loi de Student à $d_p + d_0$ degrés de liberté.

Résultats

La méthodologie développée a été implémentée (de l'étape d'imputation multiple jusqu'à celle du test t-modéré) sous forme d'un package `R` appelé `mi4p` et a été comparée au package `R DAPAR`, couramment utilisé pour l'analyse statistique des données de protéomique quantitative. Les performances de ces deux méthodes ont été comparées grâce aux indicateurs suivants : les taux de vrais/faux positifs/négatifs, la sensibilité, la spécificité, la précision, le F-Score et le coefficient de corrélation de Matthews.

DONNÉES SIMULÉES Cette méthodologie a été testée dans un premier temps sur des données simulées. En particulier, un plan de simulation de 100 jeux de données a été établi d'après le modèle utilisé par Lazar et al. (2016). Les jeux de données ont ensuite été amputés selon un mécanisme de manquants aléatoirement (MAR), en proportions croissantes : 1%, 5%, 10%, 15%, 20% et 25%. Plusieurs méthodes d'imputation multiple ont été comparées : la régression linéaire bayésienne (Schafer, 1997), le maximum de vraisemblance (algorithme EM), les forêts aléatoires et l'analyse en composantes principales (Giai Gianetto, 2021), ainsi que les k plus proches voisins (Troyanskaya et al., 2001). Les résultats obtenus sur les données simulées montrent un compromis entre sensibilité et spécificité, comme illustré sur la Figure 4a.

Deuxièmement, nous avons considéré des plans de simulation avec un mélange de valeurs manquantes non aléatoires (MNAR) et de valeurs manquantes complètement aléatoires (MCAR). En particulier, un plan de simulation de 100 ensembles de données a été établi suivant un plan expérimental adapté de Giai Gianetto et al. (2020) et mis en œuvre dans le package `R imp4p` via la fonction `sim.data` (Giai Gianetto, 2021). Dans ce cas, l'imputation multiple a été réalisée par la méthode du maximum de vraisemblance. Dans ce contexte, un compromis entre la sensibilité et la spécificité peut être à nouveau observé, comme l'indique la Figure 4b.



(a) MAR values.

(b) MNAR + MCAR values.

Figure 4: Comparaison entre `mi4p` et `DAPAR` en termes de distribution des différences de sensibilité, de spécificité et de F-Score sur les 100 ensembles de données simulées. L'imputation multiple a été réalisée à l'aide de la méthode d'estimation du maximum de vraisemblance.

DONNÉES RÉELLES Notre méthodologie a également été évaluée sur des jeux de données réels et contrôlés. Ainsi, nous avons considéré un premier jeu de données réel provenant de Muller et al. (2016). L'expérience a porté sur six mélanges de peptides, composés d'un fond constant de levure (*Saccharomyces cerevisiae*), dans lesquels des quantités croissantes de mélanges de protéines standard UPS1 (Sigma) ont été ajoutées à 0.5, 1, 2.5, 5, 10 et 25 fmol. Dans un deuxième ensemble de données bien calibré, la levure a été remplacée par un lysat total plus complexe de *Arabidopsis thaliana* dans lequel le mélange UPS1 a été ajouté en 7 quantités différentes, à savoir 0.05, 0.25, 0.5, 1.25, 2.5, 5 et 10 fmol. Pour chaque mélange, des triplicats techniques ont été constitués. Cette expérience imite un cas réel d'analyse protéomique quantitative différentielle. En comparaison avec le package `DAPAR`, le compromis sensibilité/spécificité est confirmé, avec une nette diminution du nombre de faux positifs et une amélioration du F-Score, comme l'illustre le Tableau 1 sur l'expérience *Arabidopsis thaliana* + UPS.

Condition vs. 10fmol	Vrais positifs	Faux positifs	Sensibilité	Specificité	F-Score
0.05fmol	-2.3%	-43%	-2.3%	+15%	+62.7%
0.25fmol	-1.5%	-43%	-1.4%	+13.9%	+65.3%
0.5fmol	-1.5%	-50.6%	-1.4%	+10.8%	+81.4%
1.25fmol	-2.3%	-62.6%	-2.3%	+10.9%	+119.8%
2.5fmol	-25.6%	-69.3%	-25.5%	+2.4%	+45.9%
5fmol	-30.3%	-65.2%	-30.4%	+5.5%	+56.1%

Table 1: Comparaison `mi4p` vs `DAPAR` en termes de pourcentages de vrais et faux positifs, de sensibilité, de spécificité ainsi que de F-Score. L'imputation multiple a été réalisée par la méthode du maximum de vraisemblance.

Développement d'un cadre bayésien pour l'analyse protéomique différentielle

Contexte

Dans l'approche de Smyth (2004), ainsi que dans notre méthodologie décrite dans la section précédente, un modèle hiérarchique est utilisé pour déduire la distribution *a posteriori* de l'estimateur de la variance pour chaque analyte. L'espérance de cette distribution est ensuite utilisée comme une estimation modérée de la variance et est injectée directement dans l'expression de la statistique t. Cependant, il pourrait être intéressant d'étendre cette approche à la fois pour la position et la dispersion des vecteurs étudiés. Au lieu de s'appuyer simplement sur les estimations modérées, cette partie de mon travail de thèse tire parti d'une approche entièrement bayésienne. La définition d'un modèle hiérarchique avec des distributions *a priori* à la fois sur les paramètres de moyenne et de variance permet d'introduire à fournir une quantification de l'incertitude pour l'analyse différentielle. L'inférence est donc réalisée en calculant la distribution *a posteriori* de la différence d'intensité moyenne des peptides entre deux conditions expérimentales.

Un cadre bayésien pour l'évaluation des différences de moyennes

Rappelons notre contexte de protéomique différentielle qui consiste à évaluer les différences entre les valeurs d'intensité moyenne des P peptides ou protéines quantifiés dans les N échantillons répartis dans les K conditions. La structure hiérarchique modélisée pour chaque groupe $k = 1, \ldots, K$ peut être représentée par le modèle graphique proposé dans Figure 5.

Le modèle génératif pour le vecteur d'intensité des peptides, $\boldsymbol{y}_k \in \mathbb{R}^P$, peut être écrit comme suit :

$$\boldsymbol{y}_k = \boldsymbol{\mu}_k + \boldsymbol{\varepsilon}_k, \ \forall k = 1, \ldots, K,$$

où :

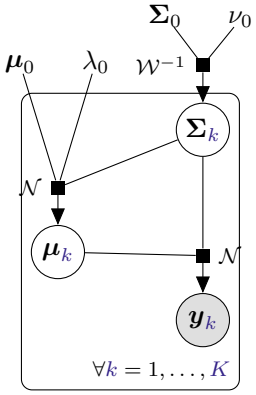


Figure 5: Modèle graphique de la structure hiérarchique du modèle génératif pour le vecteur $\boldsymbol{y}_k$ des intensités dans les K groupes d'échantillons biologiques, *i.e.* les K conditions expérimentales.

- $\boldsymbol{\mu}_k \mid \boldsymbol{\Sigma}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \frac{1}{\lambda_0}\boldsymbol{\Sigma}_k\right)$ est la vecteur des moyennes *a priori* des intensités dans le k-ème groupe,
- $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}_k)$ est le terme d'erreur dans le k-ème groupe,
- $\boldsymbol{\Sigma}_k \sim \mathcal{W}^{-1}(\boldsymbol{\Sigma}_0, \nu_0)$ est la matrice de variance-covariance *a priori* du k-ème groupe,

avec $\{\boldsymbol{\mu}_0, \lambda_0, \boldsymbol{\Sigma}_0, \nu_0\}$ un ensemble d'hyperparamètres qui doivent être choisis comme hypothèses de modélisation. Le présent cadre vise à estimer une distribution *a posteriori* pour chaque vecteur de paramètre moyen $\boldsymbol{\mu}_k$, en partant des mêmes hypothèses préalables dans chaque groupe. La comparaison entre les moyennes de tous les groupes ne repose alors que sur la capacité d'échantillonner directement à partir de ces distributions et de générer un grand nombre de réalisations pour la différence des moyennes.

Cependant, comme nous l'avons souligné précédemment, ces ensembles de données contiennent souvent des données manquantes. Supposons ainsi que $\mathcal{H}$ soit l'ensemble de toutes les données observées, nous définissons :

- $\boldsymbol{y}_k^{(0)} = \{y_{k,n}^p \in \mathcal{H},\ n = 1, \ldots N_k,\ p = 1, \ldots, P\}$, l'ensemble des éléments qui sont observés dans le k-ème groupe,
- $\boldsymbol{y}_k^{(1)} = \{y_{k,n}^p \notin \mathcal{H},\ n = 1, \ldots N_k,\ p = 1, \ldots, P\}$, l'ensemble des éléments qui sont manquants dans le k-ème groupe.

De plus, comme nous restons dans le contexte de l'imputation multiple, $\{\tilde{\boldsymbol{y}}_k^{(1),1}, \ldots, \tilde{\boldsymbol{y}}_k^{(1),D}\}$ peut être défini comme l'ensemble des D tirages d'un processus d'imputation appliqué aux données manquantes dans le k-ième groupe. Dans ce contexte, une approximation pour la distribution *a posteriori* après imputation multiple de $\boldsymbol{\mu}_k$ peut être déduite pour chaque groupe, comme indiqué dans la Proposition 5.1.

Proposition 5.1. *Pour tout $k = 1, \dots, K$, la distribution a posteriori de $\boldsymbol{\mu}_k$ peut être approximée par un mélange de distributions t multivariées et multiplement imputées, telles que :*

$$p(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}) \simeq \frac{1}{D} \sum_{d=1}^{D} T_{\nu_k} \left(\boldsymbol{\mu}; \tilde{\boldsymbol{\mu}}_k^{(d)}, \tilde{\boldsymbol{\Sigma}}_k^{(d)} \right),$$

où :

- $\nu_k = \nu_0 + N_k - P + 1$,
- $\tilde{\boldsymbol{\mu}}_k^{(d)} = \dfrac{\lambda_0 \boldsymbol{\mu}_0 + N_k \bar{\boldsymbol{y}}_k^{(d)}}{\lambda_0 + N_k}$,
- $\tilde{\boldsymbol{\Sigma}}_k^{(d)} = \dfrac{\boldsymbol{\Sigma}_0 + \sum_{n=1}^{N_k} (\tilde{\boldsymbol{y}}_{k,n}^{(d)} - \bar{\boldsymbol{y}}_k^{(d)})(\tilde{\boldsymbol{y}}_{k,n}^{(d)} - \bar{\boldsymbol{y}}_k^{(d)})^\intercal + \dfrac{\lambda_0 N_k}{(\lambda_0 + N_k)} (\bar{\boldsymbol{y}}_k^{(d)} - \boldsymbol{\mu}_0)(\bar{\boldsymbol{y}}_k^{(d)} - \boldsymbol{\mu}_0)^\intercal}{(\nu_0 + N_k - P + 1)(\lambda_0 + N_k)}$,

où nous avons introduit $\tilde{\boldsymbol{y}}_{k,n}^{(d)} = \begin{bmatrix} \boldsymbol{y}_{k,n}^{(0)} \\ \tilde{\boldsymbol{y}}_{k,n}^{(1),d} \end{bmatrix}$ pour représenter le d-ème vecteur imputé et le vecteur moyen correspondant $\bar{\boldsymbol{y}}_k^{(d)} = \dfrac{1}{N_k} \sum_{n=1}^{N_k} \tilde{\boldsymbol{y}}_{k,n}^{(d)}$.

D'autre part, en supposant qu'il n'y a pas de corrélations entre les intensités des peptides (c'est-à-dire que $\boldsymbol{\Sigma}$ est diagonale), le problème se réduit à l'analyse de P problèmes d'inférence indépendants (puisque $\boldsymbol{\mu}$ est supposé gaussien). Dans cette approche univariée, l'imputation (multiple) n'est plus nécessaire. En utilisant la même notation que précédemment et l'hypothèse de non-corrélation, la Proposition 5.1 peut être réécrite comme suit :

$$p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}\right) = \prod_{p=1}^{P} T_{2\alpha_0^p + N_k^p} \left(\mu_k^p;\ \mu_{k,N}^p,\ \hat{\sigma_k^p}^2 \right),$$

avec :

- $\mu_{k,N}^p = \dfrac{N_k^p \bar{y}_k^{p,(0)} + \lambda_0^p \mu_0^p}{\lambda_0^p + N_k^p}$,
- $\hat{\sigma_k^p}^2 = \dfrac{\beta_0^p + \frac{1}{2} \sum_{n=1}^{N_k^p} (y_{k,n}^{p,(0)} - \bar{y}_k^{p,(0)})^2 + \dfrac{\lambda_0 N_k^p}{2(\lambda_0^p + N_k^p)} (\bar{y}_k^{p,(0)} - \mu_0^p)^2}{(\alpha_0^p + \frac{N_k^p}{2})(\lambda_0^p + N_k^p)}$.

Résultats

L'un des principaux avantages de notre méthodologie est de prendre en compte la corrélation entre les peptides. Pour illustrer cette propriété, nous avons utilisé un ensemble de données protéomiques réel présenté dans Section 1.3.2, à savoir l'ensemble *Arabidopsis thaliana* +

UPS. Afin de mettre en évidence les gains que nous pouvons attendre d'une telle modélisation, nous avons comparé sur la Figure 6 le résultat d'une analyse différentielle utilisant notre méthode univariée et multivariée. Dans cet exemple, nous avons considéré un groupe de 9 peptides provenant de la même protéine (`P12081ups|SYHC_HUMAN_UPS`), dont les intensités peuvent être raisonnablement considérées comme corrélées. La distribution a posteriori de la différence des vecteur moyennes $\boldsymbol{\mu}_5 - \boldsymbol{\mu}_7$ entre les deux conditions a été calculée, et le premier peptide (`AALEELVK`) a été extrait pour une visualisation graphique. Parallèlement, l'algorithme univarié a également été appliqué pour calculer la différence *a posteriori* $\mu_5 - \mu_7$, uniquement sur le peptide `AALEELVK`. Le graphe en haut de la Figure 6 présente cette dernière approche, tandis que le cas multivarié est présenté en-dessous. Alors que la position des deux distributions est proche comme prévu, l'approche multivariée tire parti des informations provenant des peptides corrélés pour réduire l'incertitude de l'estimation *a posteriori*. Cette variance plus faible fournit un intervalle restreint de valeurs probables, permettant une estimation plus précise de la taille de l'effet et une confiance accrue dans l'inférence qui en résulte (décider si le peptide est différentiel ou non).

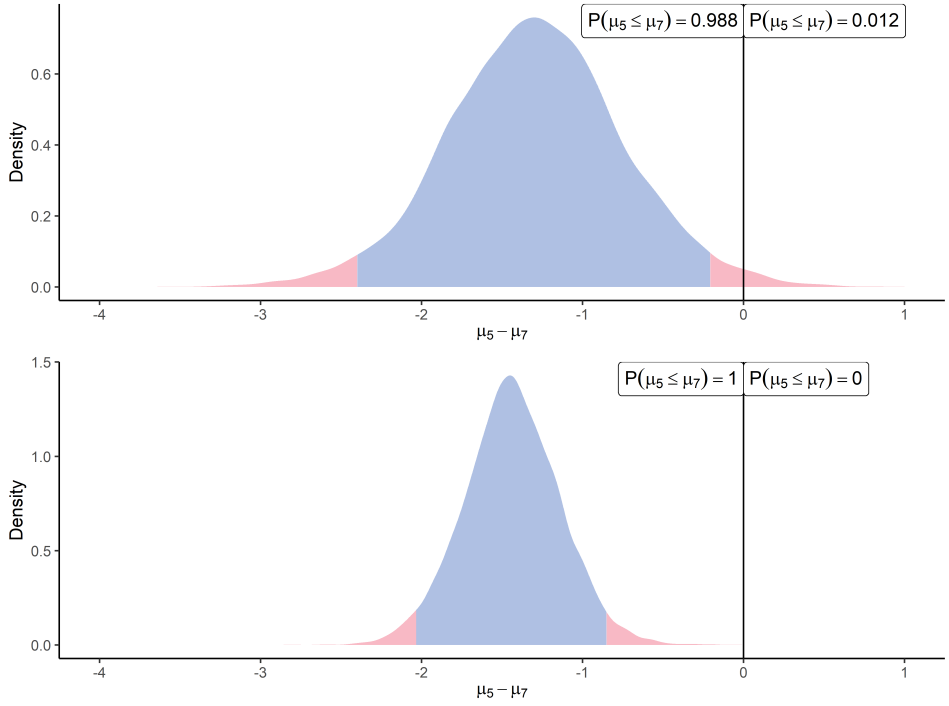


Figure 6: Distributions *a posteriori* de la différence des moyennes $\mu_5 - \mu_7$ pour le peptide `AALEELVK` de la protéine `P12081ups|SYHC_HUMAN_UPS`en utilisant l'approche univariée (haut) et l'approche multivariée (bas). L'intervalle de confiance à 95% est indiqué par la région centrale bleue.

Publications

L. Muller, L. Fornecker, M. Chion, A. Van Dorsselaer, S. Cianférani, T. Rabilloud, and C. Carapito. Extended investigation of tube-gel sample preparation: A versatile and simple choice for high throughput quantitative proteomics. *Scientific Reports*, 8(1):8260, Dec. 2018. ISSN 20452322. doi: 10.1038/s41598-018-26600-4

L. Quibel, P. Helluy, M. Chion, and P. Ricka. Mélanger des gaz raides pour créer de nouvelles lois d'état. Research Report, IRMA, Université de Strasbourg ; EDF R&D, Apr. 2019

J. Bons, G. Husson, M. Chion, M. Bonnet, M. Maumy-Bertrand, F. Delalande, S. Cianférani, F. Bertrand, B. Picard, and C. Carapito. Combining label-free and label-based accurate quantifications with SWATH-MS: Comparison with SRM and PRM for the evaluation of bovine muscle type effects. *PROTEOMICS*, 21(10):2000214, 2021. ISSN 1615-9861. doi: 10.1002/pmic.202000214

M. Chion, C. Carapito, and F. Bertrand. Accounting for multiple imputation-induced variability for differential analysis in mass spectrometry-based label-free quantitative proteomics. *arXiv:2108.07086 [q-bio, stat]*, Aug. 2021a

M. Chion, C. Carapito, and F. Bertrand. Towards a more accurate differential analysis of multiple imputed proteomics data with mi4limma. In *Statistical Analysis of Proteomic Data*. Springer US, 2022. ISBN 978-1-07-161966-7

Communications

Communications orales dans des conférences internationales à comité de lecture

Chion, M., Bons, J., Bonnet, M., Maumy-Betrand, M., Carapito, C. & Bertrand, F.: Modèle de régression par spline monotone pour données de protéomique quantitative. 52èmes Journées de Statistique 2021 – 7 au 11 juin 2021 – En ligne.

Chion, M., Bons, J., Bonnet, M., Maumy-Betrand, M., Carapito, C. & Bertrand, F.: Using monotone spline smoothing to combine label-free and label-based accurate quantifications with DIA-MS: application to bovine muscle samples. e-Chimiométrie 2021 – 2 au 3 février 2021 – En ligne.

Chion, M., Carapito, C. & Bertrand, F.: Dealing with imputation-caused variance using moderated t-test. UseR! 2020 – Annulé à cause de la COVID-19.

Chion, M., Bertrand, F. & Carapito, C. : Imputation multiple et prise en compte de l'incertitude pour les données de protéomique quantitative. 51èmes Journées de Statistique – 3 au 7 juin 2019 – Nancy, France.

Séminaires invités

Chion, M. : Developing statistical methodologies for quantitative proteomics. Journées du LabEx IRMIA – 7 octobre 2020 – Strasbourg, France.

Chion, M., Carapito, C. & Bertrand, F. : Développement de nouvelles méthodologies d'analyse de données protéomiques. Journée de rentrée de l'École Doctorale 269 – 8 octobre 2019 – Strasbourg, France.

Communications par affiche

Chion, M., Bertrand, F. & Carapito, C. : Dealing with imputation-caused variance in label-free quantitative proteomics data. May Institute on Computation and statistics for mass spectrometry and proteomics at Northeastern University – 27 avril au 8 mai 2020 – Boston, MA, États-Unis - En ligne.

Chion, M., Carapito, C. & Bertrand, F. : Imputation multiple et prise en compte de l'incertitude pour les données de protéomique quantitative. Journées Statistique & Santé – 10 au 11 octobre 2019 – Paris, France.

Chion, M., Carapito, C. & Bertrand, F. : Développement de nouvelles méthodologies d'analyse de données protéomiques. Journée de rentrée de l'École Doctorale 269 – 8 octobre 2019 – Strasbourg, France.

Chion, M., Muller, L., Pythoud, N., Carapito, C. & Bertrand, F. : Dealing with imputation-caused variance in label-free quantitative proteomics data. SMAP 2019 – 16 au 19 septembre 2019 – Strasbourg, France.

Chion, M., Carapito, C. & Bertrand, F. : Imputation multiple et prise en compte de l'incertitude pour les données de protéomique quantitative. 1er Symposium du Groupement de Recherche Masses de Données, Informations et Connaissances en Sciences – 26 au 28 juin 2019 – Rennes, France.

Contents

List of abbreviations

AUC: Area under the curve
DDA: Data-dependent acquisition
DDA: Data-independent acquisition
DNA: Desoxyribonucleic acid
EB: Empirical Bayes
EBI: European Bioinformatics Institute
FA: Formic acid
FDR: False discovery rate
GP: Gaussian process
kNN: k-nearest neighbours
LC: Liquid chromatography
LLOQ: Lower limit of quantification
LOD: Limit of detection
LOQ: Limit of quantification
LT: *longissimus thoracis* muscle
MCMC: Markov chain Monte Carlo
mice: Multiple imputation using chained equations
MLE: Maximum likelihood estimation
MS: Mass spectrometry
MS/MS: Tandem mass spectrometry
PCA: Principal component analysis
PDF: Probability density function
PFF: Peptide Fragmentation Fingerprinting
PIR: Protein Information Resource
PSM: Peptide-Spectrum Match
RF: Random forests
RNA: Ribonucleic acid
RMSE: Root-mean-square error
SIB: Swiss Institute of Bioinformatics
SIL: Stable Isotope-Labelled

SM: *Semimembranosus* uscle
ST: *Semitendinosus* muscle
SWATH-MS: Sequential Windowed Acquisition of All Theoretical Fragment Ion Mass Spectra
ULOQ: Upper limit of quantification

List of Figures

List of Tables

1

Introduction

1.1 Biological framework

1.1.1 Proteome and proteomics

The genetic information of a given biological system (cell, tissue, organ, biological fluid or organism) is explained by the central dogma of molecular biology (Cobb, 2017). Briefly, the information from a section of a DNA (desoxyribonucleic acid) molecule is transcripted into RNA (ribonucleic acid), which is then translated into amino acid sequences, called proteins. The proteome is defined as the entire set of proteins expressed by a genome of a given biological system, at a given time and under given conditions (Wilkins et al., 1996; Anderson and Anderson, 1998). Whereas a genome remains constant, proteomes differ depending on the considered cell and time. For example, the human genome represents around 20 thousand

genes, while the human proteome represents more than a million proteoforms, *i.e.* different forms of proteins (Smith and Kelleher, 2013; Aebersold et al., 2018; Smith and Kelleher, 2018). Proteomics aims at identifying, characterising and quantifying the proteome (James, 1997; Blackstock and Weir, 1999), leading to a snapshot of the system considered.

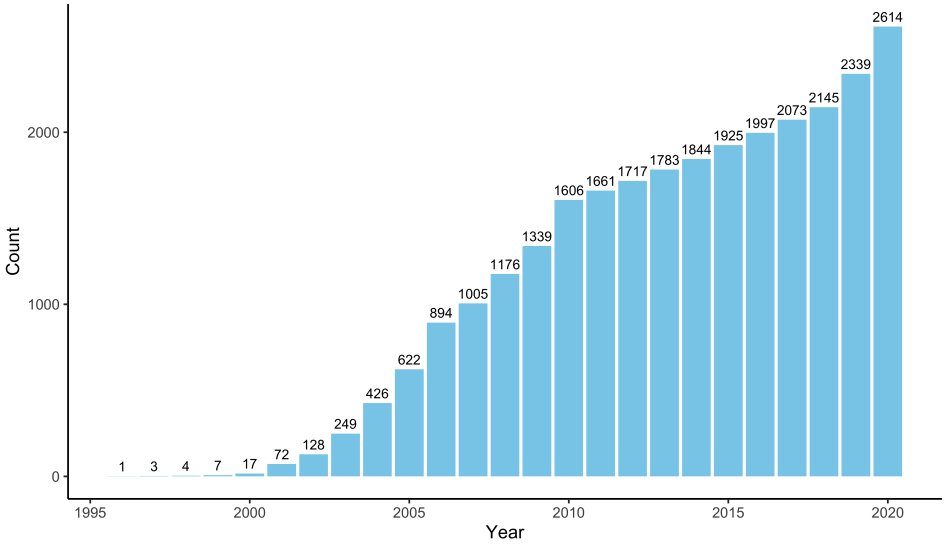


Figure 1.1: Number of PubMed entries per year which title contains both words "proteomics" and "biomarker" from 1995 to 2020.

The proteome depends on and reflects the physiological state of a tissue or a cell. Its analysis, therefore, makes it possible to study the biological processes behind disturbances, such as diseases or environmental conditions. Indeed, one can conduct differential analysis to compare a given proteome over different conditions. For instance, the blood proteome of healthy versus diseased patients can be compared by extracting quantitative measures of all identified proteins. Hypothesis testing can then be conducted to derive the subset of proteins that are very likely to be differentially expressed between the two groups of patients, which can be seen as the population of key proteins that may be involved in the disease mechanisms. Therefore, proteomics analysis plays an increasingly important role in the quest for biomarkers of all kinds of pathologies (Figure 1.1). A biomarker is defined by Strimbu and Tavel (2010) as a biological entity that can be measured and quantified in an accurate and reproducible manner and that reflects clinical signs indicative of health or disease. In addition, thanks to the technological revolution in all areas of omics over the past ten years, multi-omics analyses represent a new research area of interest, aiming at integrating information acquired at various levels, such as the genome, the transcriptome, the epigenome, the proteome, or the metabolome (Olivier et al., 2019; Schleiss et al., 2021).

1.1.2 Mass spectrometry-based proteomics

Depending on the nature of the information sought, different proteomics approaches can be used (Dupree et al., 2020; Brown et al., 2020). In the context of this thesis, only the so-called "bottom-up" approach has been considered and is detailed hereafter (Figure 1.2). It consists of analysing peptides generated after the enzymatic digestion of proteins, generally using trypsin and then inferring the proteins thanks to identifying the proteotypic peptides (*i.e.* found in only a single known protein) that belong to them. The hundreds of thousands of peptides generated are usually separated by liquid chromatography (LC) before being analysed by tandem mass spectrometry (MS/MS). During the latter, the mass of each peptide is determined beforehand and then, the detected peptides are successively isolated, fragmented, and their fragmentation spectra acquired to allow their further identification (Zhang et al., 2013; Gillet et al., 2016). From the identity of the peptides, it is possible to infer the identity of the proteins present in the starting complex mixture. This approach is the most widely used in high-throughput today, thanks to the instrumental revolution that has led to ultra-fast scanning and sensitive mass analysers and the parallel development of adapted bioinformatics tools to interpret the massive data that are generated (Dupree et al., 2020).

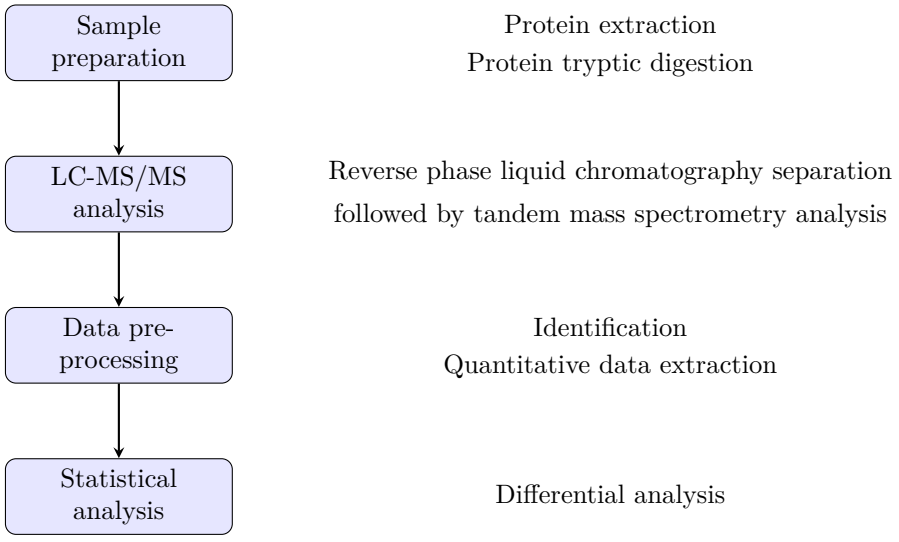


Figure 1.2: Bottom-up proteomics workflow.

1.1.2.a Liquid chromatography coupled to tandem mass-spectrometry

Protein digestion, while generating peptides that are more easily analysable, inevitably also increases sample complexity by generating hundreds of thousands of peptides (Tsiatsiani and Heck, 2015; Gillet et al., 2016). Those peptides need to be separated before entering the

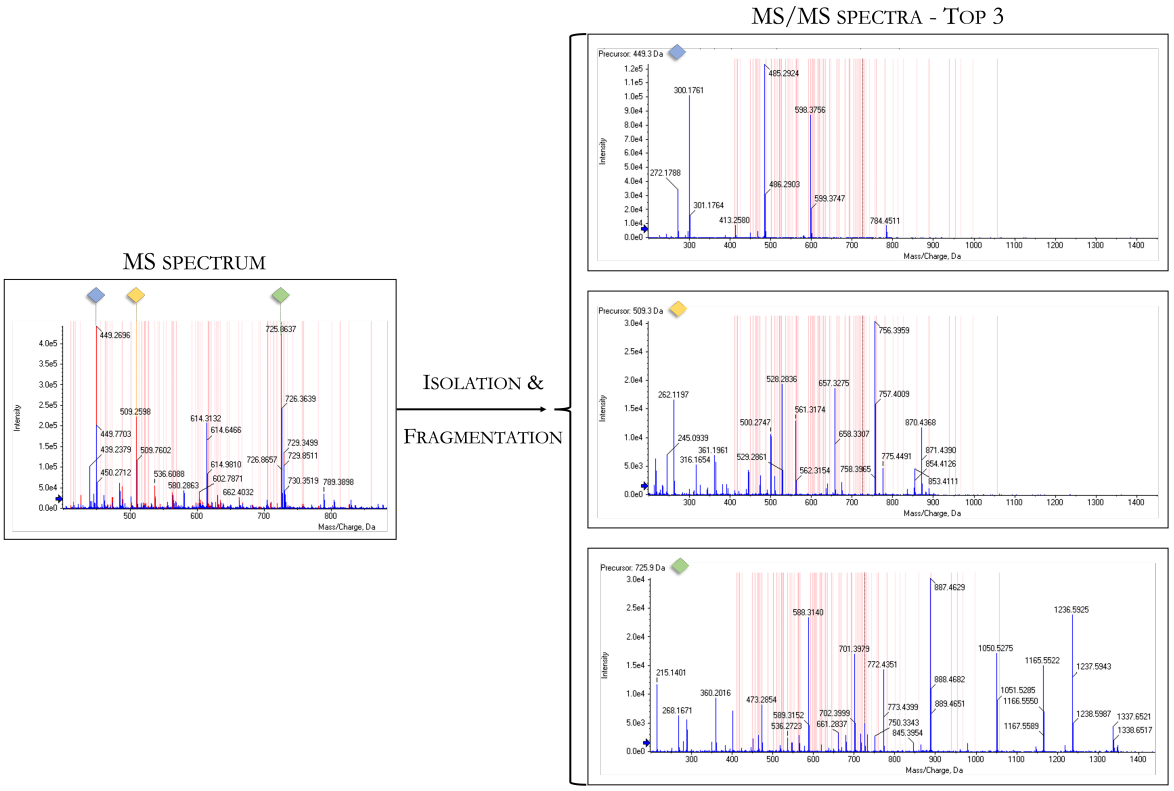


Figure 1.3: Scheme of the data-dependent acquisition mode. In this example, the three most intense precursor ions on the MS spectrum (449.3 m/z in blue, 503.3 m/z in yellow and 725.9 m/z in green) are sequentially isolated and fragmented, resulting in 3 MS/MS spectra of the fragment ions. *Courtesy of Joanna Bons.*

mass spectrometer to improve ionisation efficiency and proteome coverage. Reverse-phase liquid chromatography aims at eluting peptides depending on their hydrophobicity (Niessen, 2006). Once the peptides have eluted from the reverse-phase chromatography column, they are ionised before entering the mass spectrometer, using electrospray ionisation. The mass spectrometer first measures the mass-to-charge ratio (m/z) and the intensity of each peptide ion in order to generate a MS (or MS1) spectrum. Secondly, peptides are individually selected, isolated and fragmented. The instrument generates then MS/MS (or MS2) spectra by measuring the m/z ratio and intensities of all the fragments generated (Steen and Mann, 2004).

1.1.2.b Data-Dependent Acquisition

The Data-Dependent Acquisition (DDA) mode is still the most commonly used in bottom-up proteomic analysis. It consists of the consecutive acquisition of MS and MS/MS spectra cyclically along the chromatographic gradient. During each cycle, a MS spectrum is first acquired, then the N (user-defined) most intense precursor ions of this spectrum are se-

quentially isolated in real-time in a narrow m/z range and fragmented before being analysed to generate N MS/MS spectra (Stahl et al., 1996). On Figure 1.3, a Top 3 strategy is illustrated. The stochastic character of selecting the peptides to be fragmented limits the number of peptides sequenced during analysis, and therefore the number of peptides and proteins identified. Consequently, the DDA mode also lacks reproducibility. Table 1.1 illustrates how the numbers of protein-level and peptide-level identifications overlap across injection triplicates on various samples and mass spectrometers considered.

Samples composition	**Mass spectrometer**	**Protein-level recovery**	**Peptide-level recovery**
UPS1+YEAST	Q-Exactive +	75%	56%
UPS1+ARATH	HF-X	76%	53%
UPS1+ARATH	TimsTOF	73%	54%

Table 1.1: Overlap proportion between triplicates for various samples and mass spectrometers in DDA mode considered during my PhD thesis.

1.1.2.c Identification

Several strategies have been developed for the automatic assignment of peptide sequences from MS/MS spectra (Nesvizhskii, 2010). *De novo* sequencing consists of extracting peptide sequences directly from the MS/MS spectra. It is beneficial for the study of organisms whose genomes are not sequenced. Otherwise, spectra-centric database searching is the most commonly used method and is described hereafter.

SEARCH ENGINES In "bottom-up" proteomic analysis, peptide identifications are performed by converting the raw data generated by the mass spectrometer into a file containing information on the mass of the precursor ions, the mass of their associated fragment ions and their respective intensities. From this file named "peak list", the identification of the peptides is performed using the "Peptide Fragmentation Fingerprinting" (PFF) approach (Martin et al., 2004). It consists of comparing experimental mass lists with theoretical masses, resulting from the digestion and the *in silico* fragmentation of all proteins contained in a specific protein sequence database. The identified peptides are then grouped for protein inference to identify the proteins present in the samples. However, this protein inference can be complex, especially when peptides are shared between several proteins or when a protein is only identified through a single peptide (Nesvizhskii and Aebersold, 2005). All these steps are performed automatically by search engines such as Andromeda (Cox et al., 2011), Mascot (Matrix Science, London, UK) (Perkins et al., 1999), MS-GF+ (Kim and Pevzner, 2014), Comet (Eng et al., 2015), Sequest (Tabb, 2015) and X!Tandem (Craig and Beavis, 2004).

Protein database	Reference	Creators	Number of proteins (July 16, 2021)
RefSeq	O'Leary et al. (2016)	NCBI	209,035,492
UniProtKB	The UniProt Consortium (2021)	EBI / PIR / SIB	Swiss-Prot: 565,264 TrEMBL: 219,174,961
neXtProt (Human proteome)	Zahn-Zabal et al. (2020)	SIB	20,379

Table 1.2: Examples of some protein databases available. NCBI: "National Center of Biological Information", EBI: "European Bioinformatics Institute"; PIR: "Protein Information Resource"; SIB: "Swiss Institute of Bioinformatics"

PROTEIN DATABASES Peptide assignment is limited to the sequences present in the protein sequence database. Thus, it is crucial to work with the most appropriate database for the biological samples considered. Moreover, extracting relevant and quality information requires high-quality databases. Several databases are available, differing in their quality of annotation, their completeness and their degree of redundancy (Nesvizhskii and Aebersold, 2005). Some of them are described in Table 1.2. Protein databases are frequently updated, notably due to the discovery of new coding sequences, the sequencing of new variants (from the annotation of genomic libraries) and expert manual verifications of entries. Thus, the problem of not assigning a large proportion of MS/MS spectra, can be reduced by using regularly updated databases.

VALIDATION Proteomics identifications are most commonly validated using the target-decoy strategy (Elias and Gygi, 2007). It consists of performing searches with a database containing target protein sequences, *i.e.* real proteins, and decoy protein sequences, *i.e.* inverted or scrambled protein sequences. These dummy sequences still retain the same amino acid frequency, proteins and peptides sizes as the corresponding target proteins and peptides. The false discovery proportion (FDP) is then estimated via the false discovery rate (FDR), being the proportion of decoy sequences assigned to MS/MS spectra among the total number of assigned sequences (Navarro and Vázquez, 2009; Burger, 2018):

$$\text{FDR} = 2 \times \frac{\#\text{Assigned decoy sequences}}{\#\text{Assigned decoy sequences} + \#\text{Assigned target sequences}} \times 100 \qquad (1.1)$$

However, target-decoy competition-based FDR has several drawbacks, including lack of stability and accuracy. Hence, Couté et al. (2020) proposed a FDR-control method based on the theoretical framework of Benjamini-Hochberg (Benjamini and Hochberg, 1995).

1.1.3 Quantitative Proteomics

1.1.3.a Label-free global quantification

SPECTRAL COUNTING Quantification by counting MS/MS spectra is based on the correlation between the abundance of a protein and the number of MS/MS spectra (or PSM for "Peptide Spectrum Match") that led to the identification of this protein (Liu et al., 2004). This approach has the advantage of facilitating data processing since the results are directly obtained from the tools used for protein identification and validation. However, spectral counting shows several drawbacks due to DDA mass spectrometry (Lundgren et al., 2010). On the one hand, the undersampling generates missing values and affects the repeatability of spectral counting data provided. Note that the absence of PSM in a condition is not necessarily synonymous with the absence of the protein. On the other hand, the discriminatory nature of spectral count data leads to biased quantification of low abundance proteins (Lee et al., 2019). Moreover, the number of PSM depends on the length of the amino acid sequence, leading to small proteins being less accurately quantified than bigger ones. To cope with this issue, normalisation strategies have been developed (Blein-Nicolas and Zivy, 2016; Ankney et al., 2018). Another issue comes with peptides shared by two or more proteins, as it becomes tedious to determine to which protein those peptides should be assigned. Indeed, quantifying proteins by assigning MS/MS spectra to all proteins a peptide could originate from is inaccurate. Thus, these spectra are usually proportionally distributed to all possible proteins by considering the distribution of the unique peptides or they are excluded from the spectral count (Zhang et al., 2010; Bantscheff et al., 2012).

EXTRACTED ION CHROMATOGRAM QUANTIFICATION Quantification by Extracted Ion Chromatogram (XIC) is based on the correlation between the abundance of a peptide and its chromatographic MS signal (Blein-Nicolas and Zivy, 2016). Thus, this strategy employs the MS signal of peptides, as described in Figure 1.4, by integrating the intensity of each ion over its chromatographic elution profile (Bantscheff et al., 2012; Cappadona et al., 2012). However, peptide identification is enabled by the data collected at the MS/MS level. Consequently, DDA mode needs to be well-parameterised to, on the one hand, collect enough MS spectra to reconstruct chromatographic peaks and then perform accurate quantification and, on the other hand, to collect sufficient high-quality and numbers of MS/MS spectra to reach a satisfying depth and coverage of the proteome. Protein quantification inference is usually computed by aggregating the observed intensities at the peptide level, using the sum or the weighted average. Thus, the XIC strategy provides as many measurements as there are quantified peptide ions, leading to each protein in a given sample being measured as many times as it has peptide ions that have been assigned to it (Belouah et al., 2019). Difficulties arising from shared peptides have been highlighted and tackled (Blein-Nicolas et al., 2012; Gerster et al., 2014; Jacob et al., 2019), as well as those coming from missing values (Karpievitch et al., 2009; Richardson et al., 2012).

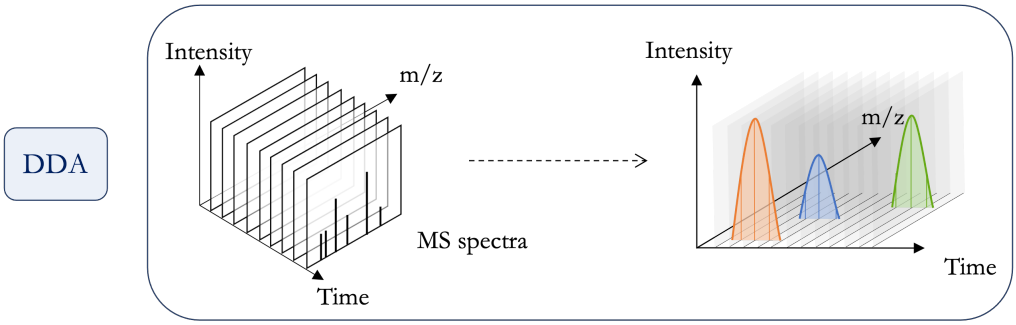


Figure 1.4: XIC-MS quantification in DDA mode. *Courtesy of Joanna Bons.*

1.1.3.b Label-based targeted quantification

Targeted proteomics gathers MS-based proteomics methods aiming at specifically detecting and accurately quantifying a list of beforehand selected proteins and peptides based on fragment ion chromatograms. Contrary to MS1-based quantification strategies, targeted proteomics monitors the targeted features across their entire chromatographic elution peak, leveraging the missing value issue. As a result, they offer a highly reproducible and accurate quantification of the targeted peptides and proteins.

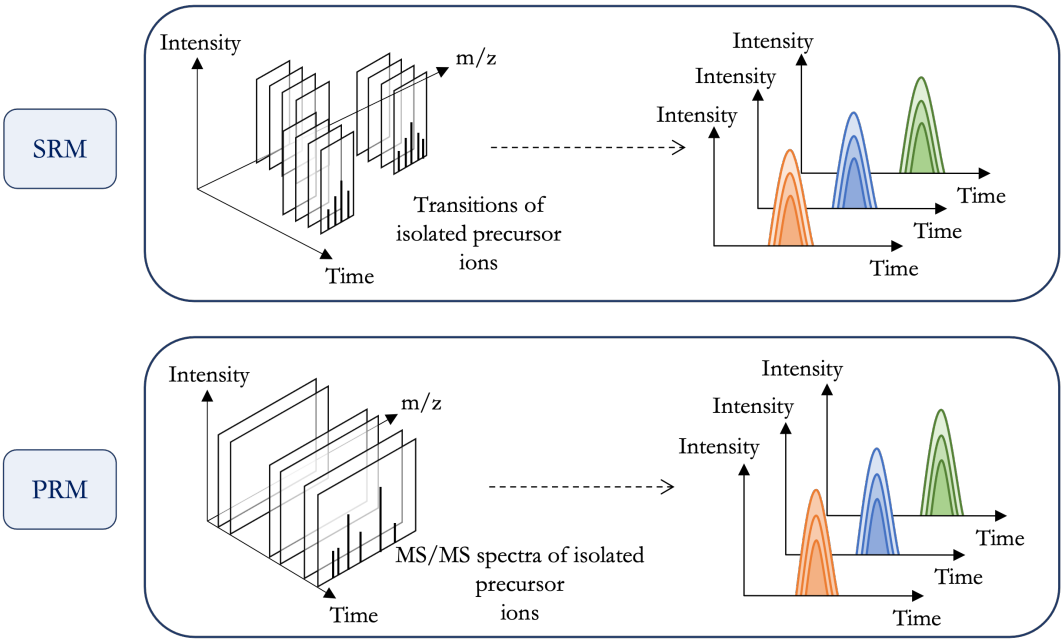


Figure 1.5: Scheme of Selected Reaction Monitoring and Parallel Reaction Monitoring. *Courtesy of Joanna Bons.*

SELECTED AND PARALLEL REACTION MONITORING Selected Reaction Monitoring (SRM) has long been the reference method for targeted proteomics approaches (Ankney et al., 2018). The precursor and fragment ions to sequentially isolate are predefined by the experimenter on a transition list. The precursor ion and fragment ion pair is called transition. Finally, ion chromatograms are extracted for each transition and are grouped when originating from the same precursor ion (Figure 1.5). With the emergence of high resolution/accurate mass instruments, a new targeted strategy, Parallel Reaction Monitoring (PRM), has been developed. Contrary to SRM, no fragment ion selection is required *a priori*. Instead, all ions are co-analysed, and full-scan MS/MS spectra are generated. Finally, chromatographic peaks are extracted for each transition (Figure 1.5).

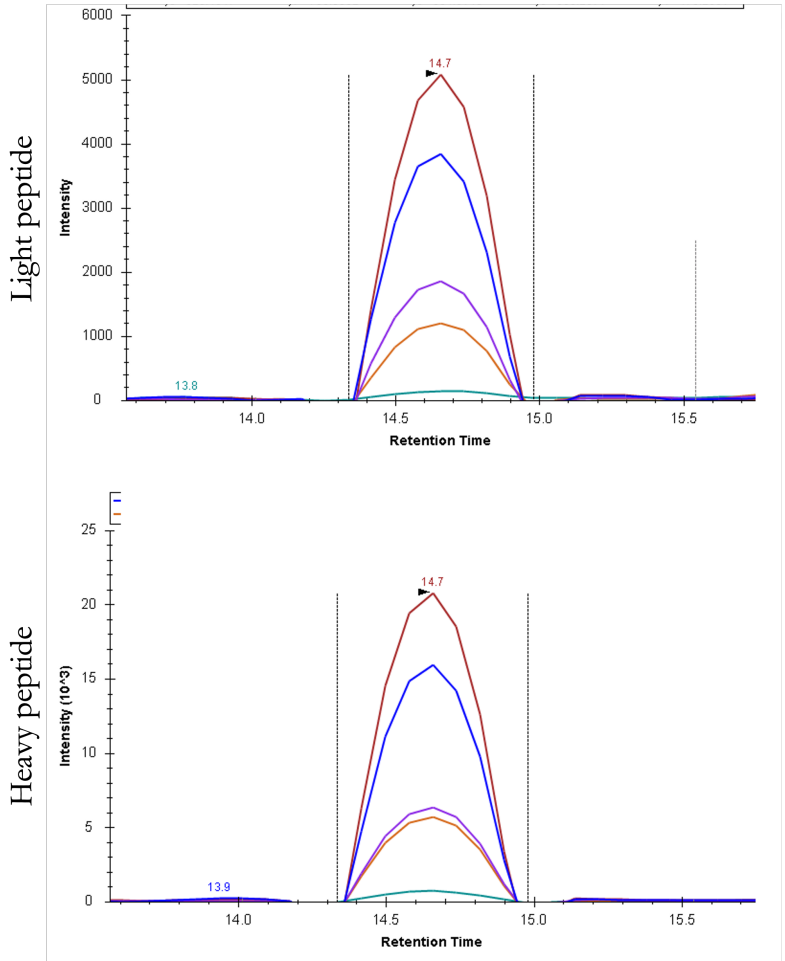


Figure 1.6: Light/heavy similarity in targeted quantification. *Courtesy of Joanna Bons.*

ISOTOPIC DILUTION - ABSOLUTE QUANTIFICATION It is possible to reach the absolute quantification level by adding precisely known amounts of stable isotope-labelled standards into the samples – either as peptides or as full-length proteins. A standard labelled peptide presents the same amino acid sequence and physicochemical properties as its corresponding endogenous peptide to quantify, except that its mass is incremented due to the labelling. Quantification of each peptide is based on the area under the curve (AUC), obtained by summing the corresponding transitions' AUC (Figure 1.6). Thus, it is analogous to XIC quantification performed in DDA mode but conducted at the MS/MS level. Finally, the ratio between light peptide AUC and heavy peptide AUC is determined using the equation 1.2 to obtain the light peptide quantity in the sample.

$$\text{Light quantity} = \frac{\text{Heavy quantity}}{\text{Heavy AUC}} \times \text{Light AUC}. \tag{1.2}$$

However, it is to be noted that peptide intensity is linearly correlated with its quantity if the peptide signal is comprised within its linearity range (Vidova and Spacil, 2017). Thus, generating calibration curves and linearity range determinations is a prerequisite to reach absolute quantification.

1.1.4 Data-Independent Acquisition

1.1.4.a Principle

Proteomics would ideally allow quantifying all proteins in large sample cohorts. On the one hand, DDA-based global quantification shows large-scale capability by relatively and approximatively quantifying thousands of proteins. On the other hand, SRM and PRM, when coupled to isotope dilution, offer a chance to reach robust absolute quantification with higher sensitivity, dynamic range and accuracy and reproducibility on finite lists of peptides/proteins. Data-independent acquisition (DIA) combines the strength of those two approaches by sequencing all detectable peptides in a given m/z range, regardless of any information concerning the precursor (Doerr, 2015; Vidova and Spacil, 2017). Thus, it combines the coverage of DDA approaches and the quantification accuracy provided by using MS/MS signals, as is the case in pure targeted approaches.

Gillet et al. (2012) introduced a DIA strategy called SWATH-MS, for "Sequential Windowed Acquisition of all THeoretical fragment ion Mass Spectra". This methodology aims at acquiring MS/MS spectra by sequentially isolating and fragmenting all precursor ions contained in a few-m/z-wide isolation window. This process results in multiplexed MS/MS spectra, providing digital maps of all peptides contained in a given biological sample (Ludwig et al., 2018).

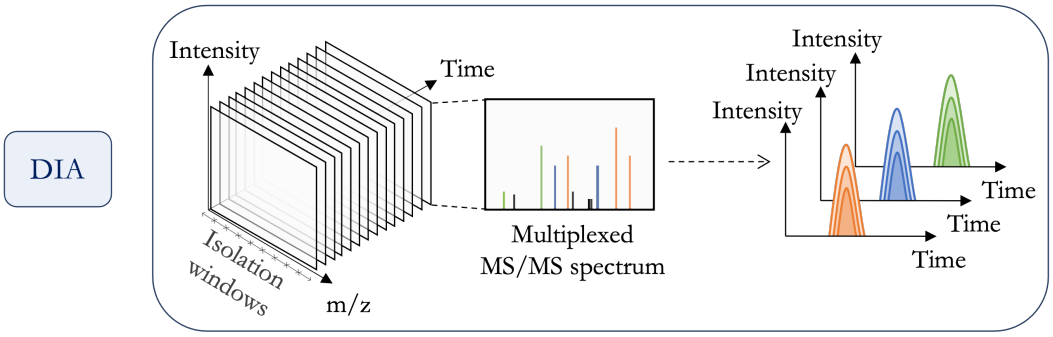


Figure 1.7: Scheme of Data-Independent Acquisition mode. *Courtesy of Joanna Bons.*

1.1.4.b Peptide-centric DIA data extraction

The analysis of DIA data using a peptide-centric approach can be performed using a targeted data extraction or by directly matching spectra against a sequence database.

SPECTRAL LIBRARY-BASED TARGETED DATA EXTRACTION The spectral library consists of MS/MS spectra that have been assigned to a peptide sequence with a high level of confidence (Ludwig et al., 2018; Schubert et al., 2015a). These spectra are generally collected from acquisitions performed in DDA mode on fractionated samples or more recently, from the deconvolution of spectra acquired in DIA mode (hybrid spectral libraries). When using a spectral library for DIA data extraction, only the peptides contained in the library are targeted. Thus, it is essential to ensure that this library is as exhaustive as possible. When the library is generated from DDA analyses, the limitations of this acquisition mode do not allow the use of the single analysis of the same sample in DDA for the "ideal" extraction of DIA data, due to the restriction of the search space.

To avoid sample-specific spectral libraries generation, public spectral libraries can be used. They can be extracted from repositories dedicated to the collection of various types of MS data (as in 1.1.2.c). This information can come from platforms dedicated to the collection of spectral libraries for DIA-SWATH data extraction, among which are PeptideAtlas (Desiere et al., 2006), MassIVE/PRIDE (Wang et al., 2018) or SWATHAtlas (Rosenberger et al., 2014). The latter, created in 2014, however, contains only a limited number of libraries: 18 spectral libraries covering 11 organisms (as of August, 2nd 2021).

Finally, when coupled with isotope dilution, DIA can reach the performances of pure targeted approaches for absolute quantification of the targets for which labelled standards were added (Bons et al., 2021).

SPECTRAL LIBRARY-FREE STRATEGY Instead of using a spectral library to extract DIA data, the multiplexed spectra can be algorithmically deconvoluted in pseudoMS/MS spectra before being searched against a protein sequence database. The rise of artificial intelligence

tools is also reflected in DIA data processing, as described by Xu et al. (2020) and Meyer (2021) in their recent reviews. One such tool is DeepMass (Tiwary et al., 2019), which can predict peptide fragmentation patterns using a machine learning algorithm trained on tens of millions of MS/MS spectra. Prosit (Gessulat et al., 2019) is a flexible deep neural network architecture capable of predicting retention times, fragmentation and MS/MS spectra of peptides. pDeep (Zeng et al., 2019) is also capable of predicting peptide fragmentation from different fragmentation modes. Note that the intensity of the predicted spectra is instrument-dependent (Xu et al., 2020). Data processing software such as Spectronaut (Biognosys) or DIA-NN (Demichev et al., 2020), now use deep neural networks to improve their processing. The recently released MaxDIA also uses deep learning techniques thanks to DeepMass:Prism (Sinitcyn et al., 2021).

1.2 Statistical framework

1.2.1 Empirical Bayes for equality of means testing

Differential proteomics analysis consists of identifying peptides or proteins (analytes) which are differentially expressed between experimental conditions. These experiments produce high-dimensional data (1,000-10,000 analytes) with a small number of independent replicates of each condition (1-10 replicates). As a result, univariate statistical methods applied to each analyte might provide imprecise results (Phipson et al., 2016). Similar problems arise in gene expression datasets. Therefore, Baldi and Long (2001) and Wright and Simon (2003) introduced the empirical Bayes framework for analysing microarray expression data. The empirical Bayes procedure (Efron and Morris, 1971; Casella, 1985) enables to leverage information from the entire dataset when inferring on a single individual. Lönnstedt and Speed (2002) and Smyth (2004) used a parametric empirical Bayes approach using a simple mixture of normal models and a conjugate prior. Furthermore, they derived an expression for the posterior odds of differential expression for each gene. The method was implemented into an `R` package called `limma`, available on Bioconductor (Smyth et al., 2003) and is widely used for analysing gene expression datasets, including quantitative proteomics datasets. Phipson et al. (2016) recalls the work of Smyth (2004) using a genomic experiment in which the expression levels of G genes are measured for N RNA samples and hence assumed the following model for each gene $g \in \{1, \ldots, G\}$:

$$\mathbb{E}(Y_g) = \boldsymbol{X}\boldsymbol{\beta_g}, \tag{1.3}$$

where:

- $\boldsymbol{y_g} = (y_{g1}, \ldots, y_{gN})^{\mathrm{T}}$ is the response vector for the g-th gene across the N RNA samples considered,

- $\boldsymbol{X}$ is a $N \times K$ full rank matrix which corresponds to the design matrix,
- $\boldsymbol{\beta_g} = (\beta_{g1}, \ldots, \beta_{gK})$ is the unknown coefficient vector which parametrises the average expression levels in each experimental condition considered.

Note that the equation 1.3 is in a matrix form using the N realisations of the univariate random variables Y_g. For each gene $g \in \{1, \ldots, G\}$ and sample $n \in \{1, \ldots, N\}$, the random variables Y_g are assumed to be independent with:

$$\mathbb{V}\mathrm{ar}(Y_g) = \sigma_g^2, \tag{1.4}$$

where σ_g^2 is the unknown variance. The vector of parameters $\boldsymbol{\beta_g}$ can be estimated using the least squares method:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{g}} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y_g}. \tag{1.5}$$

Set $\hat{\boldsymbol{\mu}}_{\boldsymbol{g}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{g}}$. The estimators $\hat{\sigma}_g^2$ of the residual sample variances can be written as:

$$\hat{\sigma}_g^2 = \frac{1}{d_g}(\boldsymbol{y_g} - \hat{\boldsymbol{\mu}}_{\boldsymbol{g}})^{\mathrm{T}}(\boldsymbol{y_g} - \hat{\boldsymbol{\mu}}_{\boldsymbol{g}}), \tag{1.6}$$

with d_g being the residual degrees of freedom.

Under this model, $\hat{\sigma}_g^2$ is assumed to follow a scaled chi-square distribution conditional to σ_g^2. Taking advantage of the parallel structure of proteomics data leads to assuming a prior distribution for the unknown variances σ_g^2. This assumption translates how these variances vary across the analytes. Hence, the following Bayesian hierarchical model is considered:

$$\begin{cases} \hat{\sigma}_g^2 \mid \sigma_g^2 \sim \dfrac{\sigma_g^2}{d_g} \times \chi^2_{d_g} \\ \dfrac{1}{\sigma_g^2} \sim \dfrac{1}{d_0 \times s_0^2} \times \chi^2_{d_0} \end{cases} \tag{1.7}$$

This leads to the following posterior distribution of $\dfrac{1}{\sigma_g^2}$ conditional to $\hat{\sigma}_g^2$:

$$\frac{1}{\sigma_g^2} \mid \hat{\sigma}_g^2 \sim \frac{1}{d_g \times \hat{\sigma}_g^2 + d_0 \times s_0^2}\chi^2_{d_g + d_0}. \tag{1.8}$$

A moderated variance estimator is then derived from the posterior mean:

$$\hat{\sigma}^2_{g[\mathrm{mod}]} = \frac{d_g \times \hat{\sigma}_g^2 + d_0 \times s_0^2}{d_g + d_0}. \tag{1.9}$$

Smyth (2004) proposes a moderated t-test based on the usual t-test to test the null hypothesis $\mathcal{H}_0 : \beta_{gk} = 0$. The test statistic associated to this test is built by replacing the variance

estimation $\hat{\sigma}_g^2$ by the moderated one $\hat{\sigma}_{g[\text{mod}]}^2$:

$$T_{j[\text{mod}]} = \frac{\hat{\beta}_{gk}}{\hat{\sigma}_{g[\text{mod}]}^2 \sqrt{(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{\Omega_g} \boldsymbol{X})_{j,j}^{-1}}} \tag{1.10}$$

where $(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{\Omega_g} \boldsymbol{X})_{j,j}^{-1}$ is the j-th diagonal element in the matrix $(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{\Omega_g} \boldsymbol{X})^{-1}$. Under the $\mathcal{H}_0$ hypothesis, $T_{j[\text{mod}]}$ follows a Student distribution with $d_g + d_0$ degrees of freedom.

1.2.2 Missing values description

In quantitative proteomics, missing values arise from a variety of reasons. An analyte can have a missing intensity value because it is simply not present in the biological sample in the first place. However, a missing intensity value can also be due to biochemical, analytical and bioinformatical reasons that can be intrinsically related (O'Brien et al., 2018). For example, a peptide can be missing because its intensity falls below the limit of detection of the mass spectrometer due to a low ionisation efficiency. Furthermore, missing values are also caused by the inherent stochasticity of data-dependent analysis, as explained in Section 1.1.2.b. Moreover, identification issues due to search engine errors, missed cleavages, or shared peptides can also lead to a missing intensity value for a peptide in a sample and an observed value for the same peptide in another sample. Different strategies can be used to tackle this issue depending on missingness patterns and mechanisms. In this section, the notation is adapted from Imbert and Vialaneix (2018). Let $\boldsymbol{Y} = (Y_j)_{1 \leq j \leq P}$ be a vector of P random variables, $\boldsymbol{y_i} = (y_{ij})_{1 \leq j \leq P}$ the vector of their realisations for an individual i and $\mathcal{Y} = (y_{ij})_{1 \leq i \leq N,\, 1 \leq j \leq P}$ denote the data matrix. Similarly, let $\mathcal{M} = (m_{ij})_{1 \leq i \leq N,\, 1 \leq j \leq P}$ denote the corresponding missingness indicator matrix and M its associated random variable, such that:

$$m_{ij} = \begin{cases} 0 & \text{if } y_{ij} \text{ is observed.} \\ 1 & \text{if } y_{ij} \text{ is missing.} \end{cases} \tag{1.11}$$

Consequently, a partition of Y can be defined as follows, with Y_0 and Y_1 respectively being the observed and the missing parts of Y_j:

$$Y = M\boldsymbol{Y_1} + (1 - M)\boldsymbol{Y_0}. \tag{1.12}$$

1.2.2.a Missingness patterns

Missingness patterns describe which values of the dataset are missing or observed (Little and Rubin, 2019). Several types of missing values can be distinguished (van Buuren, 2018):

- A missing data pattern is said to be univariate if there is only one variable with missing data. If there is more than one variable with missing data, it is called multivariate.

- A missing data pattern is said to be monotone if the variables can be ordered such that, when an observation is missing for a given variable, then all subsequent variables for that same individual are also missing. Otherwise, it is said to be non-monotone or general.
- A missing data pattern is said to be connected if any observed data point can be reached from any other observed data point through a sequence of horizontal or vertical moves. Otherwise, it is called unconnected.

Figure 1.8 describes the typical missingness pattern in a quantitative proteomics dataset. The pattern is multivariate, general and unconnected.

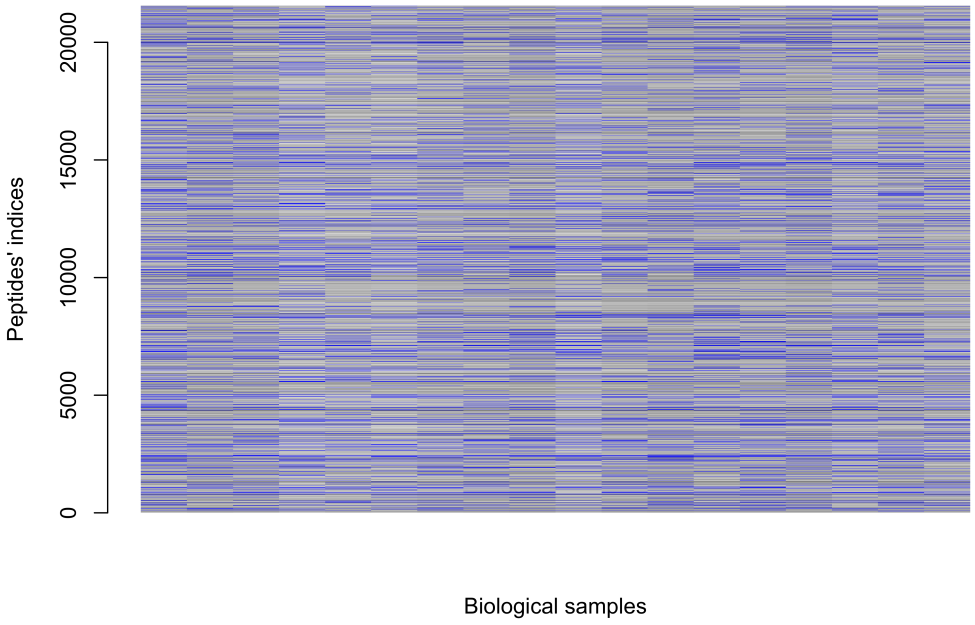


Figure 1.8: Missing values patterns in a quantitative proteomics dataset.

1.2.2.b Missingness mechanisms

Missingness mechanism denotes the process which governs the probabilities of a data point to be missing or observed Rubin (1976). It is characterised by $f_{\boldsymbol{M}|\boldsymbol{Y}}$ the conditional distribution of $\boldsymbol{M}$ given $\boldsymbol{Y}$, depending on unknown parameters $\boldsymbol{\phi}$.

- Data are missing completely at random (MCAR) if the conditional distribution of $\boldsymbol{M}$ given $\boldsymbol{Y}$ does not depend on the values of the data, missing or observed. Hence, for

all i and any distinct values $\boldsymbol{y_i}$, $\boldsymbol{y_i}^*$ in the sample space of $\boldsymbol{Y}$:

$$f_{\boldsymbol{M}|\boldsymbol{Y}}(\boldsymbol{m_i}|\boldsymbol{y_i}, \boldsymbol{\phi}) = f_{\boldsymbol{M}|\boldsymbol{Y}}(\boldsymbol{m_i}|\boldsymbol{y_i}^*, \boldsymbol{\phi}). \tag{1.13}$$

- Data are missing at random (MAR) if the conditional distribution of $\boldsymbol{M}$ given $\boldsymbol{Y}$ depends only on the observed values. Hence, for all i, for all $\boldsymbol{y_{0,i}}$ in the sample space of $\boldsymbol{Y_0}$ and any distinct values $\boldsymbol{y_{1,i}}$, $\boldsymbol{y_{1,i}}^*$ in the sample space of $\boldsymbol{Y_1}$:

$$f_{\boldsymbol{M}|\boldsymbol{Y}}(\boldsymbol{m_i}|\boldsymbol{y_{0,i}}, \boldsymbol{y_{1,i}}, \boldsymbol{\phi}) = f_{\boldsymbol{M}|\boldsymbol{Y}}(\boldsymbol{m_i}|\boldsymbol{y_{0,i}}, \boldsymbol{y_{1,i}}^*, \boldsymbol{\phi}). \tag{1.14}$$

- Data are missing not at random (MNAR) if the conditional distribution of $\boldsymbol{M}$ given $\boldsymbol{Y}$ depends on the values of the data. Hence, for some i and some distinct values $\boldsymbol{y_{1,i}}$, $\boldsymbol{y_{1,i}}^*$ in the sample space of $\boldsymbol{Y_1}$:

$$f_{\boldsymbol{M}|\boldsymbol{Y}}(\boldsymbol{m_i}|\boldsymbol{y_{0,i}}, \boldsymbol{y_{1,i}}, \boldsymbol{\phi}) \neq f_{\boldsymbol{M}|\boldsymbol{Y}}(\boldsymbol{m_i}|\boldsymbol{y_{0,i}}, \boldsymbol{y_{1,i}}^*, \boldsymbol{\phi}). \tag{1.15}$$

Different strategies can be used to deal with missing values in a dataset. Identifying the missingness mechanism in the dataset permits the choice of an appropriate method. The simplest way consists of deleting from the dataset all the observations for which there are missing values, leading to a complete-case dataset (van Buuren, 2018). This procedure, called complete-case analysis or listwise deletion, is convenient yet unwise. Indeed, under MCAR data, it provides unbiased estimates of means, variances and regression weights (Little and Rubin, 2019). Otherwise, listwise deletion can produce biased estimates, as shown in Schafer and Graham (2002). Pairwise deletion (or available-case analysis) consists of calculating estimators using all observed values. It attempts to fix the loss of information (van Buuren, 2018). However, if the data are not MCAR, the estimates can still be biased. Further problems arise from multivariate analysis, especially with covariance and correlation estimation. Indeed, these estimators are calculated using different subsamples and subsamples' sizes, leading to an unclear choice of sample size for standard error calculation. The problem is exacerbated with highly correlated variables (Little, 1992). Other statistical methods for analysing only available data were reviewed by Imbert and Vialaneix (2018).

1.2.2.c Missingness nomenclature in quantitative proteomics

Missing values in a proteomics dataset can be classified following the Rubin's distinction on missingness mechanisms. Hence, Wieczorek et al. (2017) highlight missing values as such:

- MCAR values are caused by the combination of multiple minor errors which cannot be explained by the nature or the intensity of the analyte.
- MNAR values are produced by analytes' intensities below the lower limit of detection of the mass spectrometer.

Condition	1			2			3		
Sample	1	2	3	4	5	6	7	8	9
Intensity	NA	NA	NA	23.0	NA	NA	21.9	22.4	21.9

Table 1.3: Example of missingness nomenclature in a quantitative proteomics dataset.

Furthermore, they define a nomenclature for missingness in proteomics data, dividing missing values as follows:

- Partially observed values (POV) are values which are missing in some replicates of a given experimental condition. These are a mixture of MCAR and MNAR values.
- Missing in an entire condition (MEC) are values which are missing in all replicates of a given experimental condition. In absence of alternative evidence, these missing values are generally considered as MNAR values.

Table 1.3 illustrates the POV-MEC nomenclature on a given analyte. This example depicts a quantitative proteomics experiment on three conditions with triplicates. The values in condition 1 are MEC values whereas those in condition 2 are POV values.

1.2.3 Missing values imputation

1.2.3.a Single imputation

Another way to cope with missing data is to use methods that account for the missing information. For the last decades, researchers advocated the use of a single technique called imputation. Imputing missing values consists of replacing a missing value with a value derived using a user-defined formula (such as the mean, the median or a value provided by an expert, thus considering the user's knowledge). Hence it makes it possible to perform the analysis as if the data were complete. More particularly, the vector of parameters of interest can be then estimated. Single imputation means completing the dataset once and considering the imputed dataset as if it was never incomplete, see Figure 1.9. However, single imputation has the major disadvantage of discarding the variability from the missing data and the imputation process. It may also lead to a biased estimator of the vector of parameters of interest.

1.2.3.b Multiple imputation

Multiple imputation described by Rubin (1987) closes this loophole by generating several imputed datasets. These datasets are then used to build a combined estimator of the vector of parameters of interest, by usually using the mean of the estimates among all the imputed datasets, see Figure 1.10. Let $\boldsymbol{\theta}$ be a vector of parameters of interest estimated using $(\boldsymbol{Y_0}, \boldsymbol{Y_1})$ as defined in 1.2.2. The idea is to relate the observed-data posterior distribution to

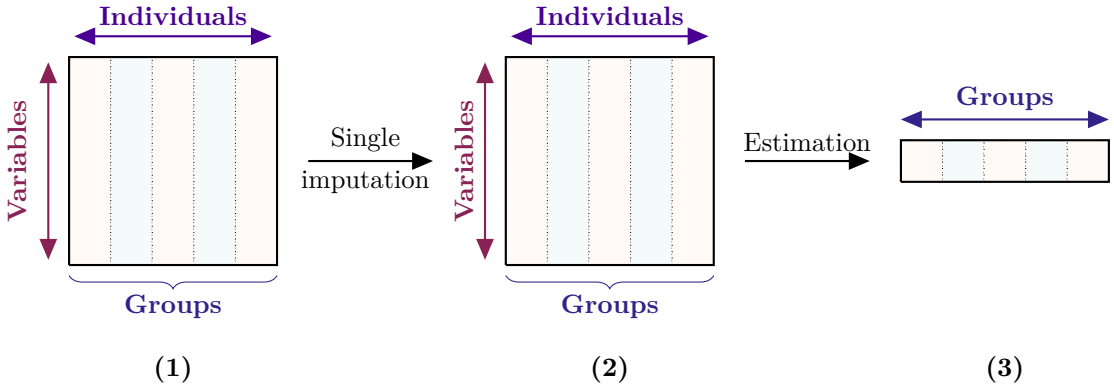


Figure 1.9: Single imputation strategy. **(1)** Initial dataset with missing values. **(2)** Single imputation provides an imputed dataset. **(3)** The vector of parameters of interest is estimated based on the single imputed dataset.

the "complete-data" posterior distribution that would have been obtained if we had observed the missing data $\boldsymbol{Y_1}$, namely:

$$p(\boldsymbol{\theta}|\boldsymbol{Y_0}) = \int p(\boldsymbol{\theta}, \boldsymbol{Y_1}|\boldsymbol{Y_0}) d\boldsymbol{Y_1} = \int p(\boldsymbol{\theta}|\boldsymbol{Y_1}, \boldsymbol{Y_0}) p(\boldsymbol{Y_1}|\boldsymbol{Y_0}) d\boldsymbol{Y_1}. \tag{1.16}$$

Consequently, $p(\boldsymbol{\theta}|\boldsymbol{Y_0})$, the posterior distribution of $\boldsymbol{\theta}$, can be simulated as follows:

1. For $d = 1, \ldots, D$, draw the missing values $\boldsymbol{Y_{1,d}}$ from their joint posterior distribution $p(\boldsymbol{Y_1}|\boldsymbol{Y_0})$.
2. Impute the drawn values to complete the dataset.
3. Draw $\boldsymbol{\theta}$ from its "completed-data" posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{Y_0 Y_{1,d}})$.

In the case of posterior means and variances adequately summarising the posterior distribution, Equation 1.16 can be replaced by:

$$\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{Y_0}) = \mathbb{E}[\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{Y_1}, \boldsymbol{Y_0})|\boldsymbol{Y_0})], \tag{1.17}$$

and:

$$\mathbb{V}\text{ar}(\boldsymbol{\theta}|\boldsymbol{Y_0}) = \mathbb{E}[\mathbb{V}\text{ar}(\boldsymbol{\theta}|\boldsymbol{Y_1}, \boldsymbol{Y_0})|\boldsymbol{Y_0}] + \mathbb{V}\text{ar}[\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{Y_1}, \boldsymbol{Y_0})|\boldsymbol{Y_0}]. \tag{1.18}$$

Multiple imputation provides an effective approximation of Equation 1.16 as:

$$p(\boldsymbol{\theta}|\boldsymbol{Y_0}) \approx \frac{1}{D} \sum_{d=1}^{D} p(\boldsymbol{\theta}|\boldsymbol{Y_0}, \boldsymbol{Y_{1,d}}), \tag{1.19}$$

where $\boldsymbol{Y_{1,d}} \sim p(\boldsymbol{Y_1}|\boldsymbol{Y_0})$ are draws of $\boldsymbol{Y_1}$ from the posterior predictive distribution of the missing values. Similarly, approximations of mean and variance estimator can be formulated

as:

$$\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{Y_0}) \approx \int \boldsymbol{\theta} \frac{1}{D} \sum_{d=1}^{D} p(\boldsymbol{\theta}|\boldsymbol{Y_0}, \boldsymbol{Y_{1,d}}) = \frac{1}{D} \sum_{d=1}^{D} \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{Y_0}, \boldsymbol{Y_{1,d}}), \tag{1.20}$$

and:

$$\mathbb{V}\text{ar}(\boldsymbol{\theta}|\boldsymbol{Y_0}) \approx \frac{1}{D} \sum_{d=1}^{D} V_d + \frac{1}{D-1} \sum_{d=1}^{D} (\hat{\boldsymbol{\theta}}_{1,d} - \bar{\boldsymbol{\theta}}), \tag{1.21}$$

with:

- V_d is the complete-data posterior variance of $\boldsymbol{\theta}$ calculated for the d-th dataset $(\boldsymbol{Y_0}, \boldsymbol{Y_{1,d}})$.
- $\hat{\boldsymbol{\theta}}_{1,d} = \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{Y_0}, \boldsymbol{Y_{1,d}})$
- $\bar{\boldsymbol{\theta}} = \frac{1}{D} \hat{\boldsymbol{\theta}}_{1,d}$

This combined estimator given in Equation 1.20 is known as the first Rubin's rule. The second Rubin's rule (Equation 1.21) states a formula to estimate the variance of the combined estimator, decomposing it as the sum of the within-imputation variance component and the between-imputation component.

The rule of thumb suggested by White et al. (2011) takes the number of imputed datasets D as the percentage of missing values in the original dataset. Recent work focused on better estimating the Fraction of Missing Information (Pan and Wei, 2018) or improving that rule (von Hippel, 2020). Note that Rubin's rules cannot be used in order to get a combined imputed dataset but instead provide an estimator of the vector of parameters of interest and an estimator of its covariance matrix both based on multiple imputation, see Figure 1.10.

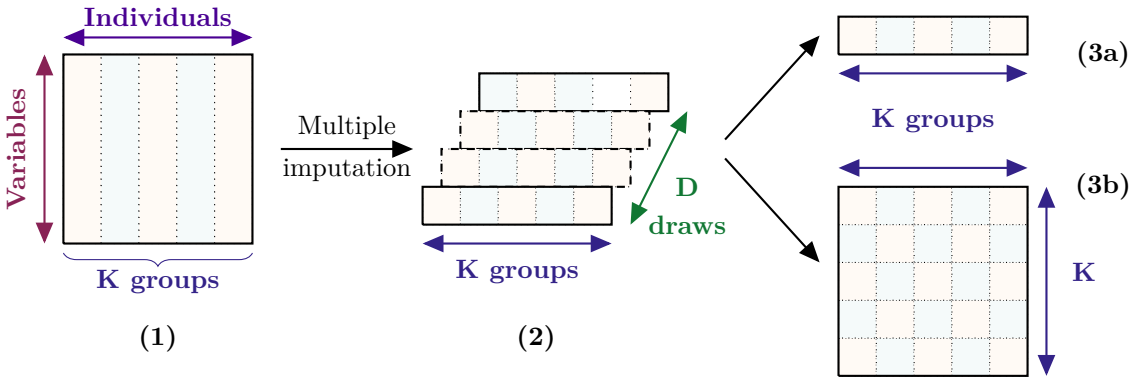


Figure 1.10: Multiple imputation strategy. **(1)** Initial dataset with missing values. It is supposed that variables are split into K groups. **(2)** Multiple imputation provides D estimators for the vector of parameters of interest. **(3a)** The **D** estimators are combined using the first Rubin's rule to get the combined estimator. **(3a)** The estimator of the variance-covariance matrix of the combined estimator is provided by the second Rubin's rule.

1.2.3.c Imputation methods in quantitative proteomics

Several methods for imputing missing values in mass spectrometry-based proteomics datasets were developed in the last decade. However, the recent benchmarks of imputation algorithms do not reach a consensus (as shown in Table 1.4). This is mainly due to the complex nature of the underlying missing values mechanism.

References	Imputation methods	Evaluation datasets
Karpievitch et al. (2012)	**Single imputation:** MLE	**Simulated dataset:** 10 samples, 2 groups, 1400 proteins
Choi et al. (2014)	**Single imputation:** Accelerated Failure Time model	
Webb-Robertson et al. (2015)	**Single imputation:** Single-Value Approaches (LOD1, LOD2, RTI) Local Similarity Approaches (KNN, LLS, LSA, REM, MBI) Global-Structure Approaches (PPCA and BPCA)	**Real datasets:** Mouse plasma + Shewanella oneidensis, 60 samples, 1518 peptides Human Plasma, 71 samples, 48 vs 23 T2D, 6729 peptides Mouse Lung, 32 samples, 6295 peptides
Lazar et al. (2016)	**Single imputation:** kNN, SVD, MLE, MinDet, MinProb	**Simulated dataset:** Karpievitch et al. (2012) 1000 peptides, 20 replicates **Real dataset:** Zhang et al. (2014)
Yin et al. (2016)	**Multiple imputation:** MCMC + FCS	**Real dataset:** Framingham Heart Study Offspring cohort 861 plasma proteins, 135 samples MCAR amputation on the 261 entirely observed proteins Application to 544 partially unobserved proteins (40% missing values)
Li et al. (2020)	**Single imputation:** Two-step lasso method, kNN, TR-kNN, RF, DanteR, Min	**Real datasets:** Bai et al. (2014); Kirwan et al. (2014); Fang et al. (2015)
Goeminne et al. (2020)	Hurdle model	**Real dataset:** Paulovich et al. (2010)
Giai Gianetto et al. (2020)	**Multiple imputation:** MI, PCA, MLE, kNN, IGCDA, RF, SLSA	**Simulated dataset:** Ramus et al. (2016)
Liu and Dongre (2020)	**Single imputation:** BPCA, kNN, MinProb, MLE, QRLIC, SVD, DetMin	**Real datasets:** 1-4 groups, 9-56 samples, 1847-6932 proteins Available on PRIDE repositories **Simulated datasets:** Based on the real datasets 3 groups, 27-60 samples, 2800-3500 proteins
Jin et al. (2021)	**Single imputation:** left-censored methods, kNN, LLS, RF, SVD, BPCA	**Real datasets:** (E.coli + Yeast) + UPS, 7 groups, 56 samples Immune cell dataset, 3 vs 4 samples Amputation of complete cases
Shen et al. (2021)	**Single imputation:** swKNN, pwKNN, Min/2, Mean, PPCA, NIPALS, SVD, SVT, FRMF, CAM	**Real dataset:** Herrington et al. (2018) Amputation of complete cases from real datasets
Song and Yu (2021)	**Single imputation:** XGboost, mean, kNN, BPCA, LLS, RF	**Real datasets:** Kinases expression of human colon and rectal cancer cell line : 65 samples, 235 kinases Proteome about the interstitial lung disease : 11 samples, random draw of 500 completely observed proteins Ovarian cancer proteome dataset : 25 samples, random draw of 400 completely observed proteins

Table 1.4: State of the art on imputation methods used in quantitative proteomics and type of datasets used for evaluation purposes.

Imputation methods are abbreviated in Table 1.4 as follows:
BPCA: Bayesian principal component analysis
CAM: Convex analysis of mixtures
FCS: Fully conditional specification
FRMF: Fused regularisation matrix factorisation
kNN: k-nearest neighbours
LLS: Local least-squares
LOD1: Half of the global minimum
LOD2: Half of the peptide minimum
LSA: Least-squares adaptive
MBI: Model-based imputation
MCMC: Monte-Carlo Markov chains
MI: Multiple imputation
mice: Multiple imputation using chained equations
MinDet: Deterministic minimum
MinProb: Probabilistic minimum
MLE: Maximum likelihood estimation
NIPALS: Non-linear estimation by iterative partial least squares
PCA: Principal component analysis
PPCA: Probabilistic principal component analysis
pwKNN: Protein-wise k-nearest neighbours
QRLIC: Quantile regression imputation of left-censored missing data
SLSA: Structured least squares algorithm
SVD: Singular value decomposition
SVT: Singular value thresholding
swKNN: Sample-wise k-nearest neighbours
REM: Regularised expectation maximisation
RF: Random forests
RTI: Random tail imputation

1.2.3.d Software implementation

In state-of-the-art software for statistical analysis in label-free quantitative proteomics, single imputation is the most commonly used method to deal with missing values. In the `MSstats R` package (available on Bioconductor), Choi et al. (2014) distinguish missing completely at random values and missing values due to low intensities. The user can then choose to impute the censored value using a threshold value or an accelerated failure time model. The `Perseus` software by Tyanova et al. (2016) offers three methods for single imputation: either imputing by "NaN", impute by a user-defined constant or impute according to a

Gaussian distribution in order to simulate intensities, which are lower than the limit of detection. Recently, Goeminne et al. (2020) implemented a single imputation method based on a hurdle model in their `MSqRob R` package (Goeminne et al., 2018). As far as machine learning is concerned, Song and Yu (2021) suggested a method for imputing missing values in label-free mass spectrometry-based proteomics datasets, using the `XGboost` algorithm.

The `ProStaR` software based on the `DAPAR R` package (available on Bioconductor) and developed by Wieczorek et al. (2017) makes the most of the POV/MEC nomenclature for imputation purposes (Wieczorek et al., 2019). The software allows single imputation, using either a small quantile from the distribution of the considered biological sample, the k-nearest neighbours (kNN) algorithm or the structured least squares adaptative algorithm or by choosing a fixed value. The `PANDA-view` software developed by Chang et al. (2018) also enables the use of the kNN algorithm or a fixed value. Moreover, both software programs give the possibility to impute the dataset several times before combining the imputed datasets in order to get a final dataset without any missing values. `PANDA-view` relies on the `mice R` package by van Buuren and Groothuis-Oudshoorn (2011), whereas `ProStaR` accounts for the nature of missing values and imputes them with the proteomics-devoted `imp4p R` package implemented by Giai Gianetto et al. (2020).

Software	References	Imputation methods
`MSqRob`	Goeminne et al. (2020)	**Single imputation** Hurdle model
`MSstats`	Choi et al. (2014)	**Single imputation** Accelerated Failure Time model User-defined constant
`PANDA-view`	Chang et al. (2018)	**Single imputation** k-nearest neighbours User-defined constant **Multiple imputation** `mice R` package
`Perseus`	Tyanova et al. (2016)	**Single imputation** Gaussian distribution User-defined constant NaN imputation
`ProStaR`	Wieczorek et al. (2017)	**Single imputation** Quantile imputation k-nearest neighbours Structured Least Squares Adaptative **Multiple imputation** `imp4p R` package

Table 1.5: Summary of imputation methods available in state-of-the-art quantitative proteomics software packages.

However, note that there are some statistical methods for analysing proteomics data that rely neither on imputing nor filtering missing values. Luo et al. (2009) suggested the use of a Bayesian hierarchical model-based method for iTRAQ data. A similar method is proposed by O'Brien et al. (2018) using Bayesian selection model but for label-free data analysis. As far as label-free data analysis is concerned, Taylor et al. (2013) compared the performances of an accelarated failure time (supposing that missing values result from censoring below a detection model) model to a mixture model (supposing that missing values result from a combination of censoring and absence of an analyte). Ryu et al. (2014) presented a method based on a censored regression for intensity-dependent missing values and a filtering of the quantificiation-dependent missing values. Chen et al. (2014) described a penalised expectation-maximisation algorithm that incorporates missing data mechanism.

1.2.4 Multivariate empirical Bayes

As far as a vector of parameters of interest is concerned and not a scalar, the second Rubin's rule (Equation (1.21)) provides a variance-covariance matrix estimator and not a variance estimator. The univariate moderated t-test described in Section 1.2.1 is not applicable *per se*. An extension to the multivariate case was suggested by Madsen et al. (2019) for detection of differential methylation in gene expression studies.

Let $\hat{\mathbf{\Sigma}}$ be the estimator of the variance-covariance matrix $\mathbf{\Sigma}$ of the vector of parameters of interest $\boldsymbol{\beta}$, obtained using the second Rubin's rule. Assume that $\boldsymbol{\beta}$ has length K. The distribution of $\hat{\mathbf{\Sigma}}$ conditional to $\mathbf{\Sigma}$ is assumed to follow a Wishart distribution. A prior inverse-Wishart distribution is assumed on $\mathbf{\Sigma}$. Hence, both assumptions lead to the following multivariate Bayesian hierarchical model:

$$\begin{cases} \hat{\mathbf{\Sigma}} \mid \mathbf{\Sigma} \sim \mathcal{W}_K(\frac{1}{d}\mathbf{\Sigma}, d) \\ \mathbf{\Sigma^{-1}} \sim \mathcal{W}_K(\frac{1}{\nu_0}\mathbf{\Sigma}_0, \nu_0) \end{cases} \tag{1.22}$$

Madsen et al. (2019) provide then a moderated F-test by phrasing the equality of means testing as a reduction from two nested multiple regression models.

Let $\boldsymbol{\mathcal{X}}$ be a random matrix of size (ν_0, K) which rows are independently drawn from a K-variate normal distribution with zero mean:

$$\mathcal{X} \sim \mathcal{N}_K(0, \frac{1}{\nu_0}\mathbf{\Sigma}_0). \tag{1.23}$$

Suppose $\mathbf{\Sigma^{-1}}$, being a random matrix of size (K, K), can be expressed as:

$$\mathbf{\Sigma^{-1}} = \boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}. \tag{1.24}$$

Then the distribution of $\mathbf{\Sigma^{-1}}$ defines the Wishart distribution with scale parameter $\frac{1}{\nu_0}\mathbf{\Sigma}_0$

and ν_0 degrees of freedom and can be written as (Wishart, 1928):

$$\mathbf{\Sigma^{-1}} \sim \mathcal{W}_K(\frac{1}{\nu_0}\mathbf{\Sigma}_0, \nu_0). \tag{1.25}$$

Note that:

- When $\frac{1}{\nu_0}\mathbf{\Sigma}_0 = \boldsymbol{I}_K$, the Wishart distribution is said to be standard.
- When $K = 1$, the Wishart distribution is a χ^2 distribution with ν_0 degrees of freedom.

If $\nu_0 \geq K$, the density of $\mathbf{\Sigma^{-1}}$ is given by the following expression (Anderson, 2003):

$$f(\mathbf{\Sigma^{-1}}) = \frac{1}{2^{\frac{\nu_0 K}{2}} |\frac{1}{\nu_0}\mathbf{\Sigma}_0|^{\frac{\nu_0}{2}} \Gamma_K(\frac{\nu_0}{2})} |\mathbf{\Sigma^{-1}}|^{\frac{\nu_0 - K - 1}{2}} \exp\left[-\frac{1}{2}\mathrm{tr}(\nu_0 \mathbf{\Sigma}_0^{-1}\mathbf{\Sigma^{-1}})\right], \tag{1.26}$$

where $|\mathbf{\Sigma^{-1}}|$ denotes the determinant of $\mathbf{\Sigma^{-1}}$ and Γ_K is the multivariate gamma function defined as:

$$\Gamma_K(\frac{\nu_0}{2}) = \pi^{\frac{K(K-1)}{4}} \prod_{j=1}^{p} \Gamma\left[\frac{\nu_0 - j + 1}{2}\right]. \tag{1.27}$$

If $\mathbf{\Sigma^{-1}} \sim \mathcal{W}_K(\frac{1}{\nu_0}\mathbf{\Sigma}_0, \nu_0)$, then $\mathbf{\Sigma} = (\mathbf{\Sigma^{-1}})^{-1}$ is said to follow an Inverse-Wishart distribution with scale parameter $\nu_0 \mathbf{\Sigma}_0^{-1}$ and ν_0 degrees of freedom, which can be written as (Mardia et al., 1979):

$$\mathbf{\Sigma} \sim \mathcal{W}_K^{-1}(\nu_0 \mathbf{\Sigma}_0^{-1}, \nu_0). \tag{1.28}$$

Thus, the density of $\mathbf{\Sigma}$ can be expressed as (Gelman, 2015):

$$f(\mathbf{\Sigma}) = \frac{|\nu_0 \mathbf{\Sigma}_0^{-1}|^{\frac{\nu_0}{2}}}{2^{\frac{\nu_0 K}{2}} \Gamma_K(\frac{\nu_0}{2})} |\mathbf{\Sigma}|^{\frac{-(\nu_0 + K + 1)}{2}} \exp\left[-\frac{1}{2}\mathrm{tr}(\nu_0 \mathbf{\Sigma}_0^{-1}\mathbf{\Sigma^{-1}})\right]. \tag{1.29}$$

In Bayesian statistics, the Wishart distribution is the conjugate prior to the precision matrix (inverse variance-covariance matrix) of a multivariate Gaussian distribution. Similarly, the Inverse-Wishart distribution is the conjugate prior to the variance-covariance matrix of a multivariate Gaussian distribution (Bishop, 2006).

1.2.5 Regression under monotonicity constraint

In quantitative proteomics, it is assumed that the quantity of an analyte is proportional to its measured intensity through a response factor specific to the analyte and the biological sample considered. The double quantification enabled by DIA-SWATH-MS (as described in section 1.1.4) produces two datasets. On the one hand, the targeted quantification dataset for which heavy labelled standards were spiked provides the measured intensities of the analytes of interest (of the order of 10 analytes) as well as their quantity derived from the known amounts

of spiked standards (see section 1.1.3.b). On the other hand, the global quantification dataset contains the measured intensities for all analytes in the biological samples considered (of the order of 1000 analytes). The next step consists of taking advantage of the exhaustive nature of SWATH-MS acquisition mode to attempt quantifying accurately all proteins present in the biological samples considered. Schubert et al. (2015b) assumed a linear correlation between summed MS/MS intensities and concentrations of proteins. He et al. (2019) proposed a strategy relying on the Total Protein Approach algorithm from Wiśniewski et al. (2015). Although the linear model is a convenient approximation, it has to be questioned notably due to ionisation efficiency as highlighted by O'Brien et al. (2018).

In statistics, the problem posed beforehand implies the estimation of a function f such as:

$$y = f(x)$$

where x and y respectively denote the intensity and the quantity of an analyte. Here, the estimation of f is constrained by the quantitative proteomics hypothesis. Indeed, f belongs to the space of monotone (non-decreasing) functions.

1.2.5.a Isotonic regression

The imposition of the monotonicity constraint on the shape of the regression function is a widely tackled issue (Kelly and Rice, 1990; El Faouzi and Escoufier, 1991; Rigollet and Weed, 2019; Mehrjoo et al., 2020). Barlow et al. (1972) introduced the isotonic regression (also called monotonic regression) based on the pool-adjacent-violator algorithm of Robertson et al. (1988) for least-squares parameters estimation. Wu et al. (2015) proposed a penalised least squares estimator to resolve the inconsistency of isotonic regression at boundaries. Several other smoothing techniques accounting for the monotonic constraint were suggested. Friedman and Tibshirani (1984) describes a procedure combining local averaging and isotonic regression. Mammen (1991) provides an estimator combining kernel estimation with an isotonisation step through the pool adjacent violator algorithm. Isotonisation of general kernel-type estimators was also discussed by Hall and Huang (2001). Smoothing splines appear to be the method of choice for constrained smoothing (Mammen et al., 2001).

1.2.5.b Monotone splines

A polynomial regression spline on a given interval $[L, U]$ is a piecewise polynomial with specified continuity constraints (de Boor, 1978; Wegman and Wright, 1983). These are incorporated into a knot sequence $t = \{t_1, \ldots, t_{d+k}\}$, where d is the number of free parameters and k is the order of the spline (corresponding to a polynomial of degree $k - 1$). Note that the knot sequence partitions $[L, U]$ into subintervals.

Widespread application of regression spline requires a suitable set of basis splines. A basis of functions is a set $\{\phi_1, \phi_2, \ldots, \phi_B\}$ coming from a functional space $\mathcal{S}$, such as each element

of $\mathcal{S}$ can be defined as a unique linear combination of the elements of $\{\phi_1, \phi_2, \ldots, \phi_B\}$. Curry and Schoenberg (1966) suggest a set of non-negative basis splines called the M-splines. With the previously defined knot sequence t, the M-spline family is defined such as:

$$\forall b \in \{1, \ldots, B\}, \mathrm{M}_b(x; k, t) \begin{cases} \geq 0 & \text{if } t_b \leq x < t_{b+k} \\ = 0 & \text{otherwise} \end{cases} \tag{1.30}$$

and with the normalisation:

$$\int_{-\infty}^{+\infty} \mathrm{M}_b(x; k, t)\, dx = 1. \tag{1.31}$$

The M-splines family can alternatively be recursively defined for $t_b \leq x < t_{b+1}$ as:

$$\mathrm{M}_b(x; 1, t) = \frac{1}{t_{b+1} - t_b}, \quad \begin{matrix} t_b \leq x < t_{b+1} \\ \text{and 0 otherwise,} \end{matrix} \tag{1.32}$$

$$\forall k > 1, \mathrm{M}_b(x; k, t) = \frac{k[(x - t_b)\mathrm{M}_b(x; k-1, t) + (t_{b+k} - x)\mathrm{M}_{b+1}(x; k-1, t)]}{(k-1)(t_{b+1} - t_b)}. \tag{1.33}$$

Using the M-spline family as a set of basis splines allows to write any spline s as the linear combination $s = \sum_b a_b \mathrm{M}_b$. The non-negativity of s can be assured by using Equation (1.30) and by choosing coefficients a_b such as $a_b \geq 0$ and $\sum_b a_b = 1$. I-splines are then defined by Ramsay (1988) as integrated M-splines to build a basis of monotone splines:

$$\mathrm{I}_b(x; k, t) = \int_L^x \mathrm{M}_b(u; k, t) du. \tag{1.34}$$

With j such as $t_j \leq x < t_{j+1}$,

$$\mathrm{I}_b(x; k, t) = \begin{cases} 0 & \text{if } b > j, \\ \displaystyle\sum_{m=b}^{j} \frac{(t_{m+k+1} - t_m)}{k+1} \mathrm{M}_m(x; k+1, t) & \text{if } j - k + 1 \leq b \leq j, \\ 1 & \text{if } b < j - k + 1. \end{cases} \tag{1.35}$$

Using now the I-spline family as a set of basis splines (Figure 1.11) allows to write any spline s as the linear combination $s = \sum_b \alpha_b \mathrm{I}_b$. The non-decrease of s can be assured by choosing coefficients α_b such as $\alpha_b \geq 0$ and $\sum_b \alpha_b = 1$. These can be estimated by non-negative least squares (Chen and Plemmons, 2009), by solving:

$$\underset{x, x \geq 0}{\operatorname{argmin}} ||s(x) - y||_2^2. \tag{1.36}$$

Splines regression notably belong to the wider literature of functional data analysis, which provides information about curves varying on a continuum (Ramsay and Silverman, 2005).

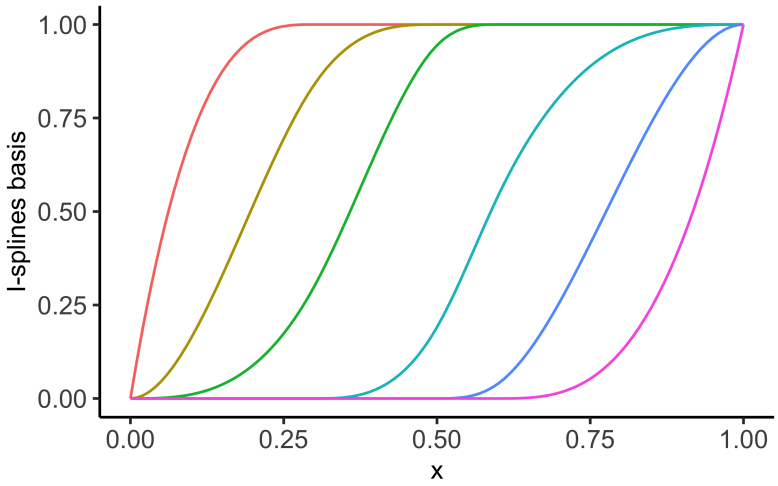


Figure 1.11: I-splines basis of order 3 associated with interior knots 0.3, 0.5 and 0.6. (After Ramsay (1988))

In this framework expanding functional data into function bases can also be achieved by Fourier bases or wavelet bases (Wang et al., 2016).

1.3 Contributions

Proteomics consists in studying the proteome, *i.e.* the set of proteins expressed by a given biological system, at a given time and under given conditions. Mass spectrometry (MS) and liquid chromatography (LC) have undergone a real instrumental revolution in the last twenty years, allowing the analysis of complex proteomes and the identification and quantification of several thousand proteins in a few hours of LC-MS/MS analysis. The increasing complexity of the massive MS data thus generated has naturally led to the need to develop adapted statistical tools and methodologies dedicated to interpreting these data. These developments are crucial to allow for larger scale and high throughput proteomic studies.

1.3.1 Development of a methodology to estimate absolute quantities of peptides from *data-independent acquisition* data

1.3.1.a Context and motivation

In *data-independent acquisition* (DIA) mode (Gillet et al., 2012; Ludwig et al., 2018), the entire mass range is covered to acquire a complete fragmentation map of the proteomes under study. The mass spectrometer acquires fragmentation spectra from consecutively isolated large mass windows to generate multiplexed MS/MS spectra. Peptide quantification is then performed using the MS/MS step, which allows a more precise and specific quantification than the MS step, as it is the case in *data-dependent acquisition* (DDA) mode.

This part of this thesis work was carried out in collaboration with Dr Muriel Bonnet (UMR Herbivores, INRA, Clermont-Ferrand) where 64 bovine muscle samples for which 20 peptides corresponding to the 10 potential biomarker proteins for beef tenderness and marbling were analyzed by a DIA method (Bonnet et al., 2020). A first step of targeted quantification coupled with isotopic dilution using labelled synthetic peptides enabled the determination of the absolute amount of the 20 peptides of interest within each of the 64 samples considered. For this purpose, the following relationship was used:

$$\text{Peptide quantity} = \frac{\text{Synthetic peptide quantity}}{\text{Synthetic peptide intensity}} \times \text{Peptide intensity}.$$

In the same dataset, all peptides were fragmented and a global quantification allowed us to measure the intensity of nearly 5500 peptides in the 64 samples considered. In quantitative proteomics, a strong assumption is made by stating that the quantity of a peptide is proportional to its intensity through a response factor. This one is specific to the considered peptide and the considered sample. Formally, such an assumption can be written as:

$$\text{Peptide quantity} = \text{Response factor} \times \text{Peptide intensity}.$$

The objective of Chapter 2 was to take advantage of data from both quantification methods. From the intensity and quantity data obtained in targeted quantification with internal standard peptides labelled on a subset of peptides, we fit a monotone spline smoothing model, explaining the quantity of a peptide by its intensity in the considered sample. This model was then used to estimate the quantities for all the peptides whose intensities were measured during the DIA analysis.

1.3.1.b Monotone spline smoothing

The monotone spline smoothing method combines I-spline regression with non-negative least squares parameter estimation, using for example the Lawson-Hanson algorithm (Lawson and Hanson, 1995). In this work, the models are defined as linear combinations of I-splines, such as:

$$f(x) = \sum_i a_i I_i(x|k,t),$$

where a_i are the coefficients to be estimated and I_{ii} constitutes a base of I-splines functions. An I-spline function is defined as the integral of an M-spline (non-negative piecewise polynomial function):

$$I_i(x|k,t) = \int_L^x M_i(u|k,t)\,du,$$

where k is the degree of the I-spline and L is the lower bound of the domain.

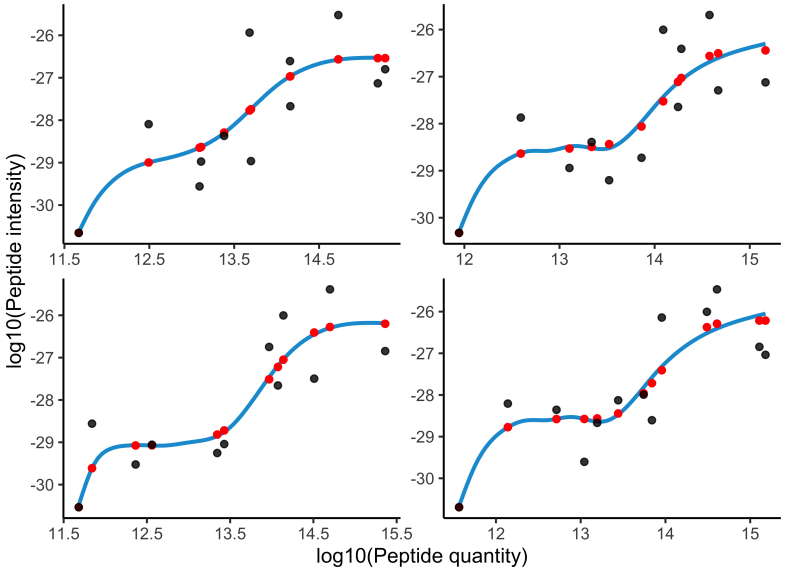


Figure 1.12: Monotone spline smoothing on 4 out of 64 samples. The black points represent the values used for model fitting, the red points are the values predicted by the model at those locations.

1.3.1.c Experiments and results

A monotonic spline smoothing model was thus fitted for each of the 64 samples. An excerpt of the graphical representation is presented in Figure 1.12. The performance of the monotonic spline models was compared to that of the linear model, through the root mean square error (RMSE). As illustrated on Figure 1.13, monotonic spline regression enhances the data fit compared to the linear model. Prediction performance was evaluated using absolute quantities of the proteins of interest. Nearly 53% of the quantity estimates varied within a ratio of 2 from the quantities derived from the targeted quantification and nearly 80% of the samples showed high consistency between the two methods. Quantity estimates of the 5500 peptides were then interpreted biologically and found to be consistent with the scientific literature on the bovine muscle proteome.

As a perspective, we also proposed an alternative probabilistic and non parametric framework, based on Gaussian process regression. This approach enhances fitting performance while additionally providing uncertainty quantification for the predicted values.

1.3.2 Development of a rigorous multiple imputation methodology for label-free quantitative proteomics data acquired in *data-dependent acquisition* mode

1.3.2.a Context and motivation

In *data-dependent acquisition* mode (DDA), the mass spectrometer generates in a first step MS spectra for all peptides. The most intense peptides are then selected to generate their

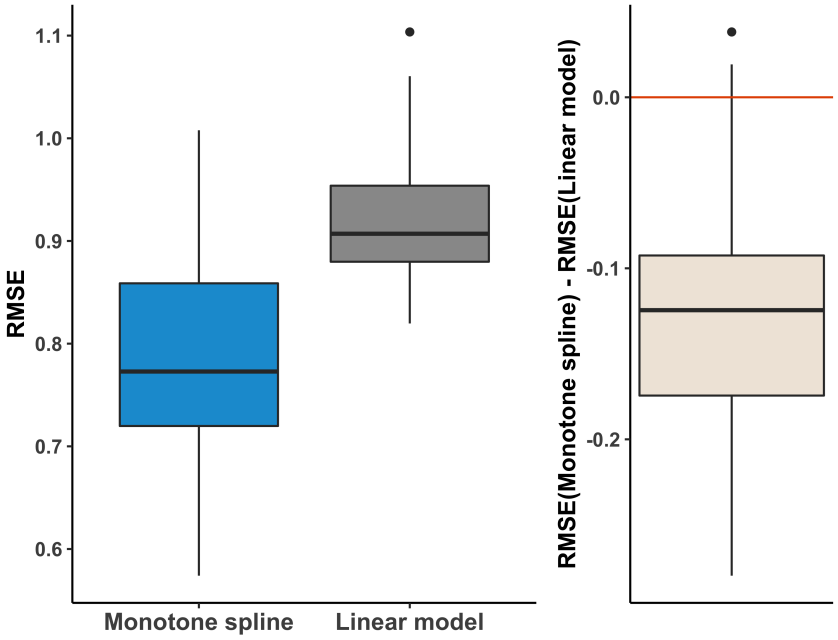


Figure 1.13: Comparison of root mean square errors (RMSE) between the monotone spline model and the linear model. Left panel depicts the comparison of the RMSE distributions for both methods. Right panel represents the difference in terms of RMSE of the monotone spline to the RMSE of the linear model.

MS/MS spectra. The quantification of the peptides is done by extracting the area under the curve of the chromatographic peak obtained in MS. In quantitative proteomics, missing values can be of biochemical, analytical, or bioinformatics origin. In the main statistical analysis software for quantitative proteomics data, it is notably proposed to impute these missing values. Thus, the software Perseus (Tyanova et al., 2016), MSstats (Choi et al., 2014) and ProStaR (Wieczorek et al., 2017) propose simple imputation methods. However, this method consists of replacing missing values only once and then considering the dataset as having always been complete. The variability related to the imputation process is therefore not taken into account. Improved single imputation methods are also available in ProStaR (Giai Gianetto et al., 2020) and PANDA-view (Chang et al., 2018). However, it turns out that in practice in the mentioned software, the imputed datasets are combined to obtain only one final dataset, which is subsequently considered to have always been complete. Although the bias of the parameter estimator obtained after this improved simple imputation is smaller in absolute value than after a usual simple imputation, the variability related to the imputation process is not rigorously taken into account.

1.3.2.b Accounting for multiple imputation-induced variability

This methodological part of my thesis work described in Chapter 3, consisted in first implementing a rigorous multiple imputation method, following Rubin's rules (Little and Rubin, 2019) (Figure 1.14). Let the vector $\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}}$ be the estimator of the parameter vector of interest $\boldsymbol{\beta}_{\boldsymbol{p}}$, obtained by the d-th imputed dataset and W_d the variance-covariance matrix of $\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}}$. The D estimators, corresponding to the D imputations, of the parameter of interest are averaged to obtain the combined estimator according to Rubin's first rule:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{p}} = \frac{1}{D}\sum_{d=1}^{D}\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}}.$$

The second Rubin's rule allows to obtain the combined estimator of the variance-covariance matrix of the combined estimator of $\hat{\boldsymbol{\beta}}_{\boldsymbol{p}}$. This takes into account both the intra-imputation variability and the inter-imputation variability (illustrating the variability due to missing values) as follows:

$$\hat{\boldsymbol{\Sigma}}_p = \frac{1}{D}\sum_{d=1}^{D} W_d + \frac{D+1}{D(D-1)}\sum_{d=1}^{D}(\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})^T(\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}).$$

This estimator of the variance-covariance matrix is then projected to obtain a univariate parameter of variability. This variance is then moderated according to a Bayesian hierarchical model (Smyth, 2004) to construct the moderated t-test statistic (Phipson et al., 2016) as:

$$T_{pj[\text{mod}]} = \frac{\hat{\beta}_{pj}}{\sqrt{\hat{\sigma}^2_{p[\text{mod}]}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}_{j,j}}},$$

with:

- $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}_{j,j}$ the j-th diagonal element in the matrix $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$
- $\hat{\sigma}^2_{p[\text{mod}]}$ is the moderated estimate of σ^2_p.

Under the null hypothesis $\mathcal{H}_0$, $T_{pj[\text{mod}]}$ follows a Student distribution with $d_p + d_0$ degrees of freedom.

1.3.2.c Experiments and results

The developed methodology has been implemented (from the multiple imputation step to the t-moderated test step) as an `R` package called `mi4p` and has been compared to the `DAPAR` `R` package commonly used for statistical analysis of quantitative proteomics data. The performance of these two methods was compared using the following indicators: true/false positive/negative rates, sensitivity, specificity, precision, F-Score and Matthews correlation coefficient.

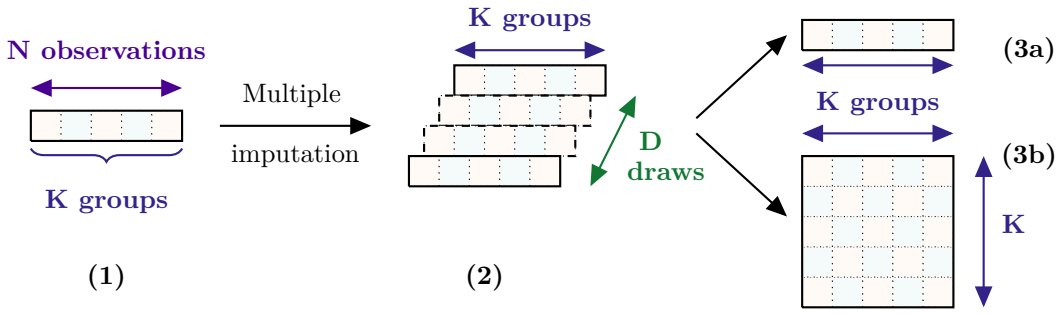
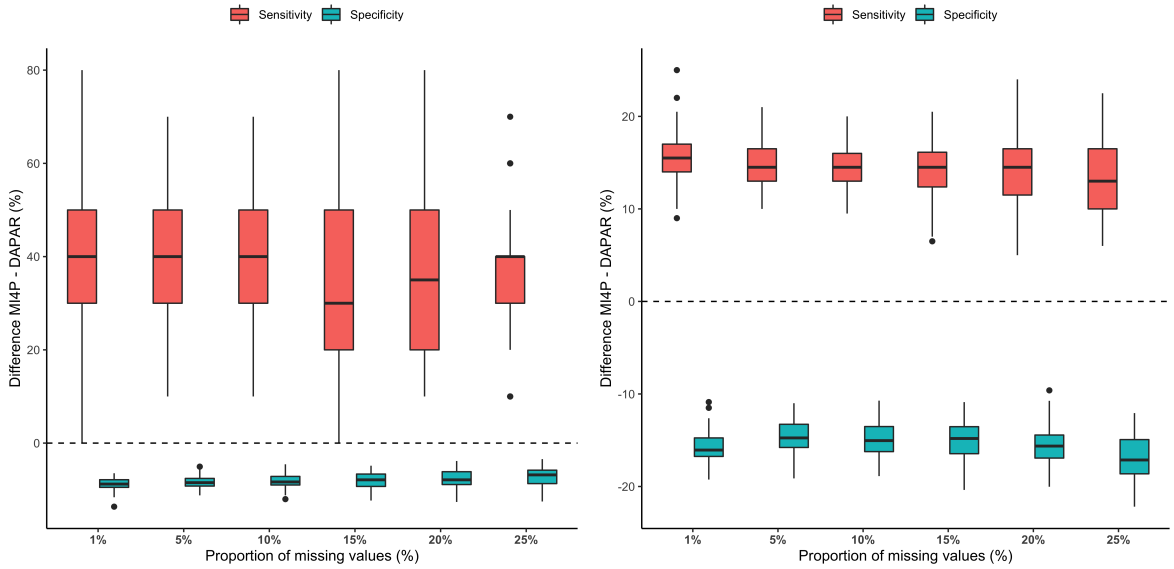


Figure 1.14: Multiple imputation strategy. **(1)** Initial dataset with missing values. It is supposed to have N observations that are split into K groups. **(2)** Multiple imputation provides D estimators for the vector of parameters of interest. **(3a)** The D estimators are combined using the first Rubin's rule to get the combined estimator. **(3b)** The estimator of the variance-covariance matrix of the combined estimator is provided by the second Rubin's rule.

SIMULATED DATASETS We first considered simulation designs with missing at random (MAR) values. In particular, a simulation plan of 100 datasets was established according to the following model (Lazar et al., 2016).The datasets were then amputated according to a missing at random (MAR) mechanism, with increasing proportions: 1%, 5%, 10%, 15%, 20% and 25%. Several multiple imputation methods were compared: Bayesian linear regression (Schafer, 1997), maximum likelihood (EM algorithm), random forests and principal component analysis (Giai Gianetto, 2021), as well as k nearest neighbors (Troyanskaya et al., 2001). The results obtained on the simulated data exhibit a trade-off between sensitivity and specificity, as illustrated in Figure 1.15.

Secondly, we considered simulation designs with a mixture of missing not at random (MNAR) and missing completely at random (MCAR) values. In particular, a simulation plan of 100 datasets was established following an experimental design adapted from Giai Gianetto et al. (2020) and implemented in the `imp4p` `R` package through the `sim.data` function (Giai Gianetto, 2021). A trade-off between sensitivity and specificity can be observed: sensitivity is increased by 15% in average while specificity is decreased by 15% in average for the `mi4p` workflow compared to the `DAPAR` one.

REAL DATASETS Our methodology has also been evaluated on real and controlled datasets. Thus, we considered a first real dataset from Muller et al. (2016). The experiment involved six peptide mixtures, composed of a constant yeast (*Saccharomyces cerevisiae*) background, into which increasing amounts of UPS1 standard proteins mixtures (Sigma) were spiked at 0.5, 1, 2.5, 5, 10 and 25 fmol, respectively. In a second well-calibrated dataset, yeast was replaced by a more complex total lysate of *Arabidopsis thaliana* in which UPS1 was spiked in 7 different amounts, namely 0.05, 0.25, 0.5, 1.25, 2.5, 5 and 10 fmol. For each mixture, technical triplicates were constituted. This experiment mimics a real case of differential quantitative proteomic analysis. In comparison with the package `DAPAR`, the sensitivity/specificity



(a) MAR values.

(b) MNAR + MCAR values.

Figure 1.15: `mi4p` vs `DAPAR` comparison in terms of distribution of differences in sensitivity and specificity on the 100 simulated data sets. Multiple imputation was performed using the maximum likelihood estimation method.

trade-off is confirmed, with a clear decrease in the number of false positives and a remarkable improvment of the F-Score, as illustrated by Table 1.6 on the *Arabidopsis thaliana* + UPS experiment.

Condition vs. 10fmol	True positives	False positives	Sensitivity	Specificity	F-Score
0.05fmol	-2.3%	-43%	-2.3%	+15%	+62.7%
0.25fmol	-1.5%	-43%	-1.4%	+13.9%	+65.3%
0.5fmol	-1.5%	-50.6%	-1.4%	+10.8%	+81.4%
1.25fmol	-2.3%	-62.6%	-2.3%	+10.9%	+119.8%
2.5fmol	-25.6%	-69.3%	-25.5%	+2.4%	+45.9%
5fmol	-30.3%	-65.2%	-30.4%	+5.5%	+56.1%

Table 1.6: Comparison `mi4p` vs `DAPAR` in terms of percentages of true and false positives and F-Score. Multiple imputation was performed using the maximum likelihood method.

1.3.3 Development of a Bayesian framework for differential proteomics analysis

1.3.3.a Context and motivation

In the state-of-the-art approach of Smyth (2004), as well as in our methodology described in the previous section, a hierarchical model is used to deduce the posterior distribution of the variance estimator for each analyte. The expectation of this distribution is then used as a

moderated estimation of variance and is injected directly in the expression of the t-statistic. However, the model could easily be extended to account both for location and dispersion estimations. Instead of relying simply on moderated estimates, this part of my thesis work takes advantage from a fully Bayesian approach. By defining a hierarchical model with prior distributions both on mean and variance parameters, we aim at providing in Chapter 5 an adequate quantification of the uncertainty for differential analysis. Inference is performed by computing the posterior distribution for the difference of mean peptide intensity between two experimental conditions.

1.3.3.b A Bayesian framework for evaluating mean differences

Let us recall that our differential proteomics context consists in assessing the differences in mean intensity values for P peptides or proteins quantified in N samples divided into K conditions. The hierarchical generative structure assumed for each group $k = 1, \dots, K$ can be represented in the graphical model in Figure 1.16.

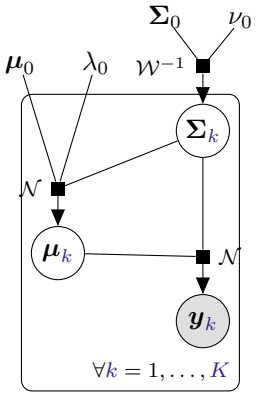


Figure 1.16: Graphical model of the hierarchical structure of the generative model for the vector $\boldsymbol{y}_k$ of peptide intensities in K groups of biological samples, *i.e.* K experimental conditions.

The generative model for a vector of peptide intensities $\boldsymbol{y}_k \in \mathbb{R}^P$, can be written as:

$$\boldsymbol{y}_k = \boldsymbol{\mu}_k + \boldsymbol{\varepsilon}_k, \ \forall k = 1, \dots, K,$$

where:

- $\boldsymbol{\mu}_k \mid \boldsymbol{\Sigma}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \frac{1}{\lambda_0}\boldsymbol{\Sigma}_k\right)$ is the prior mean intensities vector of the k-th group,
- $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}_k)$ is the error term of the k-th group,
- $\boldsymbol{\Sigma}_k \sim \mathcal{W}^{-1}(\boldsymbol{\Sigma}_0, \nu_0)$ is the prior variance-covariance matrix of the k-th group,

with $\{\boldsymbol{\mu}_0, \lambda_0, \boldsymbol{\Sigma}_0, \nu_0\}$ a set of hyper-parameters that needs to be chosen as modelling hypotheses. The present framework aspires at estimating a posterior distribution for each mean

parameter vector $\boldsymbol{\mu}_k$, starting from same the prior assumptions in each group. The comparison between means of all groups would then only rely on the ability to sample directly from these distributions and compute posterior realisations of the means' difference. However, as previously pointed out, such datasets often contain missing data and we shall introduce here consistent notation. Assume $\mathcal{H}$ to be the set of all observed data, we additionally define:

- $\boldsymbol{y}_k^{(0)} = \{y_{k,n}^p \in \mathcal{H},\ n = 1, \dots N_k,\ p = 1, \dots, P\}$, the set of elements that are observed in the k-th group,
- $\boldsymbol{y}_k^{(1)} = \{y_{k,n}^p \notin \mathcal{H},\ n = 1, \dots N_k,\ p = 1, \dots, P\}$, the set of elements that are missing the k-th group.

Moreover, as we remain in the context of multiple imputation, $\{\tilde{\boldsymbol{y}}_k^{(1),1}, \dots, \tilde{\boldsymbol{y}}_k^{(1),D}\}$ can be defined as the set of D draws of an imputation process applied on missing data in the k-th group. In such context, a closed-form approximation for the multiple-imputed posterior distribution of $\boldsymbol{\mu}_k$ can be derived for each group as stated in Proposition 1.1.

Proposition 1.1. *For all $k = 1, \dots, K$, the posterior distribution of $\boldsymbol{\mu}_k$ can be approximated by a mixture of multiple-imputed multivariate t-distributions, such as:*

$$p(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}) \simeq \frac{1}{D} \sum_{d=1}^{D} T_{\nu_k}\left(\boldsymbol{\mu}; \tilde{\boldsymbol{\mu}}_k^{(d)}, \tilde{\boldsymbol{\Sigma}}_k^{(d)}\right)$$

with:

- $\nu_k = \nu_0 + N_k - P + 1$,
- $\tilde{\boldsymbol{\mu}}_k^{(d)} = \dfrac{\lambda_0 \boldsymbol{\mu}_0 + N_k \bar{\boldsymbol{y}}_k^{(d)}}{\lambda_0 + N_k}$,
- $\tilde{\boldsymbol{\Sigma}}_k^{(d)} = \dfrac{\boldsymbol{\Sigma}_0 + \sum\limits_{n=1}^{N_k} (\tilde{\boldsymbol{y}}_{k,n}^{(d)} - \bar{\boldsymbol{y}}_k^{(d)})(\tilde{\boldsymbol{y}}_{k,n}^{(d)} - \bar{\boldsymbol{y}}_k^{(d)})^\intercal + \dfrac{\lambda_0 N_k}{(\lambda_0 + N_k)} (\bar{\boldsymbol{y}}_k^{(d)} - \boldsymbol{\mu}_0)(\bar{\boldsymbol{y}}_k^{(d)} - \boldsymbol{\mu}_0)^\intercal}{(\nu_0 + N_k - P + 1)(\lambda_0 + N_k)}$,

where we introduced the shorthand $\tilde{\boldsymbol{y}}_{k,n}^{(d)} = \begin{bmatrix} \boldsymbol{y}_{k,n}^{(0)} \\ \tilde{\boldsymbol{y}}_{k,n}^{(1),d} \end{bmatrix}$ to represent the d-th imputed vector of observed data, and the corresponding average vector $\bar{\boldsymbol{y}}_k^{(d)} = \dfrac{1}{N_k} \sum\limits_{n=1}^{N_k} \tilde{\boldsymbol{y}}_{k,n}^{(d)}$.

Besides, under the assumption that there is no correlations between peptides' intensities (*i.e.* $\boldsymbol{\Sigma}$ being diagonal), the problem reduces to the analysis of P independent inference problems (as $\boldsymbol{\mu}$ is supposed Gaussian). In this univariate context, (multiple-)imputation is no longer needed. Using the same notation as before and the uncorrelated assumption, Proposition 1.1 can be rewritten as:

$$p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}\right) = \prod_{p=1}^{P} T_{2\alpha_0^p + N_k^p}\left(\mu_k^p;\ \mu_{k,N}^p,\ \hat{\sigma_k^p}^2\right),$$

with:

- $\mu^p_{k,N} = \dfrac{N^p_k \bar{y}^{p,(0)}_k + \lambda^p_0 \mu^p_0}{\lambda^p_0 + N^p_k}$,
- $\hat{\sigma^p_k}^2 = \dfrac{\beta^p_0 + \frac{1}{2}\sum_{n=1}^{N^p_k}(y^{p,(0)}_{k,n} - \bar{y}^{p,(0)}_k)^2 + \frac{\lambda_0 N^p_k}{2(\lambda^p_0 + N^p_k)}(\bar{y}^{p,(0)}_k - \mu^p_0)^2}{(\alpha^p_0 + \frac{N^p_k}{2})(\lambda^p_0 + N^p_k)}$.

1.3.3.c Experiments and results

One of the main benefits of our methodology is to account for between-peptides correlation. As an illustration of such property, we used a real proteomics dataset introduced in Section 1.3.2, namely the *Arabidopsis thaliana* + UPS dataset. As a benefit from the Bayesian framework, the inference can be performed by visualising the posterior distribution of the mean's difference, which informs both on the effect size and its uncertainty. Such probabilistic statements can be especially valuable for practitionners when it comes to tricky decision-making problems. In order to highlight the gains that we may expect from modelling peptides' correlations, we displayed on Figure 1.17 the comparison between a differential analysis using our univariate method or using the multivariate approach. In this

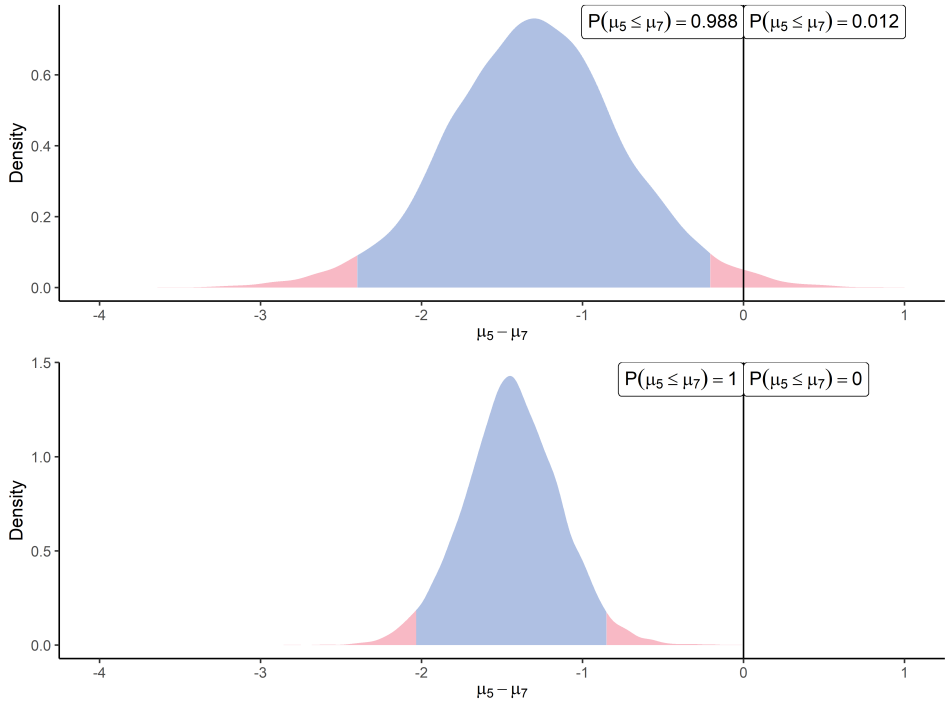


Figure 1.17: Posterior distributions of the mean difference $\mu_5 - \mu_7$ for the `AALEELVK` peptide from the `P12081ups|SYHC_HUMAN_UPS` protein using the univariate approach (top) and the multivariate approach (bottom). The 95% credible interval is indicated by the blue central region.

example, we purposefully considered a group of 9 peptides coming from the same protein (`P12081ups|SYHC_HUMAN_UPS`), which intensities may undoubtedly be correlated to some degree. The posterior difference of the mean vector $\boldsymbol{\mu}_5 - \boldsymbol{\mu}_7$ between two conditions has been computed, and the first peptide (`AALEELVK`) has been extracted for graphical visualisation. Meanwhile, the univariate algorithm has also been applied to compute the posterior difference $\mu_5 - \mu_7$, solely on the peptide `AALEELVK`. The top panel of Figure 1.17 displays the latter approach, while the multivariate case is exhibited on the bottom panel. While the location parameter of the two distributions is close as expected, the multivariate approach takes advantage of the information coming from the correlated peptides to reduce the uncertainty in the posterior estimation. This lower variance provides a tighter range of probable values, enabling a more precise estimation of the effect size and increased confidence in the resulting inference (deciding whether the peptide is differential or not).

1.3.4 Implementation

The work described in Chapter 3 was implemented in the following `R` package, available on the CRAN. The development version can also be found on `https://github.com/mariechion/mi4p`. A tutorial using this package is presented in Chapter 4.

M. Chion, C. Carapito, F. Bertrand, G. Smyth, D. McCarthy, H. Borges, T. Burger, Q. Giai-Gianetto, and S. Wieczorek. Mi4p: Multiple Imputation for Proteomics, Aug. 2021b

1.3.5 Published articles and preprints

The work presented in this manuscript led to several published articles and preprints:

J. Bons, G. Husson, M. Chion, M. Bonnet, M. Maumy-Bertrand, F. Delalande, S. Cianférani, F. Bertrand, B. Picard, and C. Carapito. Combining label-free and label-based accurate quantifications with SWATH-MS: Comparison with SRM and PRM for the evaluation of bovine muscle type effects. *PROTEOMICS*, 21(10):2000214, 2021. ISSN 1615-9861. doi: 10.1002/pmic.202000214

M. Chion, C. Carapito, and F. Bertrand. Accounting for multiple imputation-induced variability for differential analysis in mass spectrometry-based label-free quantitative proteomics. *arXiv:2108.07086 [q-bio, stat]*, Aug. 2021a

M. Chion, C. Carapito, and F. Bertrand. Towards a more accurate differential analysis of multiple imputed proteomics data with mi4limma. In *Statistical Analysis of Proteomic Data*. Springer US, 2022. ISBN 978-1-07-161966-7

2

Monotone spline smoothing for peptides' absolute quantification

The work described in this chapter is part of the work described in the article Bons et al. (2021) in collaboration with Dr Joanna Bons (LSMBO, IPHC, Strasbourg, France) and Drs. Muriel Bonnet and Brigitte Picard (UMR Herbivores, INRA, Saint-Gènes-Champanelle, France).

2.1 Introduction

Tenderness and marbling, associated with intramuscular fat content, constitute the main quality traits for beef meat conditioning, consumer satisfaction and economic performances of beef production. These traits highly vary depending on muscle type, animal (breed, gender, age) and rearing management (Couvreur et al., 2019). These qualities can currently be measured only after the slaughter of the animal by chemical quantification of intramuscular lipids (marbling), mechanical measurements of tenderness, or sensory evaluation of meat perception (Picard et al., 2018). The goal is to ultimately develop a prognosis tool for evaluating and predicting tenderness and marbling of carcasses or living animals, which the professionals of the beef sector could use. Several studies have highlighted candidate protein biomarkers for both traits (Bazile et al., 2019; Gagaoua et al., 2020), or even qualified some (Bonnet et al., 2020; Gagaoua et al., 2021). Although they represent a promising path towards quality meat assessment on alive animals or early post-mortem carcasses (Picard et al., 2015), no candidate has been validated as a biomarker to date. Validating the relationships between some candidate proteins and the two quality traits on a large scale requires quantifying the abundance of the proteins.

The work presented in this chapter is based on a DIA-MS experiment (acquired in SWATH-MS mode on a TripleTOF instrument from Sciex) combined with isotope dilution using heavy labelled AQUA peptides for targeted quantification of ten candidate biomarkers of beef meat tenderness or marbling in a cohort of 64 bovine muscle tissues. This cohort is expected to cover a wide biological range of these traits. Besides the absolute quantification of the ten targeted proteins, we have introduced a new method to estimate amounts of all detectable proteins, thus taking full advantage of the global proteome map recorded in DIA-MS data. From the intensities and quantities obtained in targeted quantification, we propose to fit monotone spline models explaining the quantity of a peptide by its intensity in

the considered sample. These models then allow us to estimate the amounts of all detected peptides thanks to the use of internal labelled standards for a subset of targeted peptides. Combining this quantitative information enabled gaining insights into muscle-type effects on the candidate biomarkers of beef meat qualities and muscle metabolism.

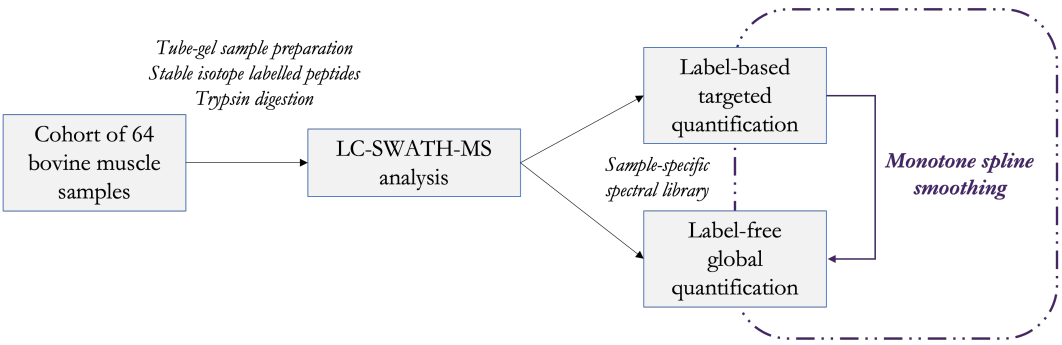


Figure 2.1: Summary of the experimental workflow considered in Chapter 2.

2.2 Materials

2.2.1 Sample preparation

Sixty-four muscle samples from previous experiments were used (Picard et al., 2014, 2018, 2019). They consist of 23 samples of *semimembranosus* (SM, fast oxido-glycolytic with intermediate intramuscular fat content), 33 samples of *longissimus thoracis* (LT, mid oxidative muscle with high intramuscular fat content), and eight samples of *semitendinosus* (ST, fast glycolytic with low intramuscular fat content) muscles. These muscle samples were collected on cows and young bulls from several breeds (Rouge des Prés, Limousine, Blonde d'Aquitaine) to be representative of cattle used in the French beef production. In addition, they have been chosen to represent a wide variety of tenderness and marbling. Proteins were extracted and samples were prepared as described in Bonnet et al. (2020). Briefly, 30 μg proteins were prepared in triplicate using a tube-gel protocol slightly adapted from Muller et al. (2016). Eleven samples were pooled as a representative matrix for method development, external quality control and generating a spectral library necessary for SWATH-MS data interpretation. The matrix pool was prepared in tube-gel for method development and quality control on the one hand and fractionated by SDS-PAGE for generating the spectral library on the other hand as described hereafter. A concentration-balanced mixture of 20 accurately quantified stable isotope labelled peptides (Spike Tides TL, JPT Peptide Technologies, Berlin, Germany) was spiked in each sample for absolute quantification. Retention time standards (iRT; Biognosys, Schlieren, Switzerland) were additionally spiked in all samples analysed in data-dependent acquisition (DDA) and SWATH-MS modes.

2.2.2 Representative matrix preparation

Eleven samples were pooled to constitute a representative matrix for method development, external quality control and generating a spectral library necessary for SWATH-MS data interpretation. The matrix pool was prepared in tube-gels on the one hand. It was also fractionated by SDS-PAGE on the other hand to generate a spectral library in DDA mode. After denaturation at 95°C for 5 min, 90 μg proteins of the representative matrix in loading buffer (4% SDS, 0.1 M DTT, 20% glycerol, 12.5 mM Tris-HCl, pH 6.8, 0.05% bromophenol blue) were loaded onto a 12%-acrylamide SDS-PAGE gel and separated in 8 bands. The gel was fixed with 50% ethanol, 3% phosphoric acid before staining with colloidal Silver Blue. Each band was excised, cut into small pieces, washed, reduced and alkylated, and proteins were digested using trypsin enzyme.

2.2.3 Liquid chromatography and mass spectrometry

SWATH-MS analyses were performed on an ekspert™ nanoLC 400 system coupled with a TripleTOF 6600 mass spectrometer (both from Sciex, Concord, Canada). Six μg peptides were separated on a Zorbax 300SB-C18 column (150 mm × 0.3 mm, 3.5 µm diameter particles; Agilent). The solvent system consisted of 0.1% FA in H2O (solvent A) and 0.1% FA in ACN (solvent B). Peptides were loaded onto the column and eluted at 5 μL/min with the following gradient of solvent B: linear from 5% to 25% in 47 min, linear from 25% to 35% in 10 min, and up to 70% in 2 min. A SWATH-MS method consisting of 100 variable windows covering the 200-1,600 m/z range with an overlap of 1 m/z was developed. MS spectra were collected for 150 ms, and MS/MS spectra for 45 ms in high-sensitivity mode, resulting in a duty cycle of 4.7 s. The collision energy for each window was the one applied to a 2+ ion centred upon the window with a spread of 10 eV.

DDA analyses were performed on the same system as SWATH-MS analyses, using the same chromatographic conditions. MS spectra were collected at 400-1,250 m/z for 150 ms. The most intense precursor ions with an intensity exceeding 10 counts/s and charge states 2-4 were selected for fragmentation, and MS/MS spectra were collected in high sensitivity mode at 200-1,600 m/z using dynamic accumulation. A dynamic exclusion time was set to 18 s. Dynamic collision energy was used.

2.2.4 Data preprocessing

2.2.4.a Spectral library generation

Wiff files corresponding to the DDA analyses of the eight SDS-PAGE gel bands, as well as the concentration-balanced mixture of heavy peptides alone and spiked in the reference matrix were converted into mgf files using the Protein Pilot 5.0 software (Sciex). Data were searched using Mascot search engine (version 2.5.1; Matrix Science, London, UK)

against an in-house concatenated target-decoy *Bos taurus* UniProtKB-TrEMBL database (31,928 entries, release 06/2017), supplemented with the retention time standards, trypsin and common contaminants, and generated with the database toolbox from MSDA (Carapito et al., 2014). The following parameters were applied: trypsin as digestion enzyme, one permitted missed cleavage, a mass tolerance of 15 ppm on the precursor ions and 0.05 Da on the fragment ions, carbamidomethylation of cysteine residues as fixed modification, and oxidation of methionine residues, label:13C(6)15N(2) and label:13C(6)15N(4) as variable modifications. Mascot result files were loaded into the ProlineStudio 2.0 software (Bouyssié et al., 2020) and identifications were validated on pretty rank equal to 1, 1% false discovery rate (FDR) on peptide spectrum matches on e-value, and 1% FDR on protein sets based on Mascot Modified Mudpit scoring.

A spectral library was generated with the Skyline software (Pino et al., 2020) (version 3.7.1.11099), by importing Mascot result files and fixing a cut-off score of 0.95. Finally, it contained 1,111 validated proteins and 8,349 validated proteotypic peptides.

2.2.4.b Selection of 10 candidate biomarkers and proteotypic peptides

Ten candidate biomarkers of tenderness or marbling were selected according to the criteria defined in Bonnet et al. (2020). The list of the targeted proteins and peptides is reported in Table 2.1.

Protein Name	Protein ID	Peptide Sequence
Four and a half LIM domains 1 (FHL1)	tr|F1MR86|F1MR86_BOVIN	CLQPLASETFVAK NPITGFGK
Malate dehydrogenase (MDH1)	sp|Q3T145|MDHC_BOVIN	LGVTSDDVK VIVVGNPANTNCLTASK
Troponin T, slow skeletal muscle (TNNT1)	sp|Q8MKH6|TNNT1_BOVIN	AQELSDWIHQLESEK YEINVLYNR
Peroxiredoxin-6 (PRDX6)	sp|O77834|PRDX6_BOVIN	LAPEFAK VIISLQLTAEK
$\alpha\beta$-crystallin (CRYAB)	sp|P02510|CRYAB_BOVIN	FSVNLDVK HFSPEELK
Retinal dehydrogenase 1 (ALDH1A1)	sp|P48644|AL1A1_BOVIN	LECGGGPWGNK QAFQIGSPWR
Triosephosphate isomerase (TPI1)	sp|Q5E956|TPIS_BOVIN	NNLGELINTLNAAK VVLAYEPVWAIGTGK
Heat shock protein beta-1 (HSPB1)	sp|Q3T149|HSPB1_BOVIN	ALPAAAIEGPAYNR SATQSAEITIPVTFQAR
Myosin-1 (heavy chain-IIx, MYH1)	sp|Q9BE40|MYH1_BOVIN	GQTVEQVYNAVGALAK TLALLFSGPASGEAEGGPK
β-enolase 3 (ENO3)	sp|Q3ZC09|ENOB_BOVIN	TAIQAAGYPDK VNQIGSVTESIQACK

Table 2.1: List of the 10 candidate biomarkers and their selected proteotypic peptides of beef meat tenderness or marbling selected for the absolute quantification.

2.2.4.c Targeted absolute quantification data processing

For SWATH-MS analysis, extraction was performed using the following parameters: the 3 to 6 most intense product ions were extracted. Resolving power was set to 50,000, and only scans within 3 min of the predicted retention time, determined using iRT standards, were used. Finally, chromatographic peaks were investigated to manually adjust peak integration boundaries and remove interfered transitions. At least three transitions were kept per precursor ion. Signal at the peptide level was obtained by summing the corresponding transition peak areas.

Peptides' limit of detection (LOD) and limits of quantification (LOQ), namely the lower limit of quantification (LLOQ) and upper limit of quantification (ULOQ), were determined using calibration curves. Eight different amounts of the concentration-balanced mixture of heavy-labelled peptides were spiked into the representative matrix: 1000-, 100-, 10-, 2-, 1-fold diluted, and 5-, 10-, 50-fold concentrated. LOD was calculated as the lower point for which the peak apex intensity was higher than 3-fold noise value. To determine the linear quantification range of each peptide, the following criteria were applied: coefficient of variation (CV) $\leq 20\%$ between analytical triplicates, coefficient of determination (R^2) ≥ 0.99 between the peptide signal and the injected quantity, $R^2 \geq 0.99$ between the back-calculated injected quantity and the real injected quantity, and 80-120% accuracy by back-calculating the expected injected quantity using the linear regression equation. LLOQ corresponds to the lower point and ULOQ to the higher point satisfying all the criteria.

After ensuring that peptides are within their linear range, the ratios between the endogen and the accurately quantified stable isotope-labelled peptides were used to determine the quantity of endogenous peptides. Results are reported in Table A.1 in the Appendix chapter.

2.2.4.d Global quantification data processing

SWATH-MS data was processed with Skyline using appropriate settings and the above described spectral library. Validated proteotypic peptides were extracted using the same parameters as for targeted absolute quantification. Peaks were reintegrated using the target decoy approach of the mProphet peak-scoring model (Reiter et al., 2011), and a q-value was assigned to each peak. Only precursors with a q-value below 0.01 were kept, and peptide intensity was obtained by summing all precursor intensities.

2.3 Methods

2.3.1 Monotone spline smoothing

A monotone spline regression model was fitted for each of the 64 bovine samples considered (as described in Section 1.2.5.b), using the data obtained at the peptide-level from the label-based quantification step. Monotone spline smoothing combines I-spline regression analysis and non negative least squares estimation to ensure monotonicity.

Let y represent the $\log_{10}$-intensity of a peptide and z represent the $\log_{10}$-quantity of a peptide. Then, for $n = 1, \ldots, N$, the fitted models are in the form of:

$$z = f_n(y) = \sum_b a_b^n I_b(y|k,t) \tag{2.1}$$

where:

- $b = 1, \ldots, B$ indexes the number of basis functions. Here, as we set the number of knots to 5, $B = 6$,
- a_b^n is the b-th coefficient in the I-splines basis expansion for sample n,
- $\mathrm{I}_b(y;k,t) = \int_L^y \mathrm{M}_b(u;k,t)du$.

In this chapter, the targeted proteomics experiment considered was conducted on $N = 64$ biological samples in which P_n peptides of interest were accurately quantified, $9 \leq P_n \leq 13$. For each sample $n = 1, \ldots, N$:

- $(y_1, y_2, \ldots, y_{P_n})$ denotes a P_n-dimensional sample of y such as $L < y_1 \leq y_2 \leq \cdots \leq y_{P_n} < U$. Note that L and U refer to lower and upper limits of quantification respectively,
- $(z_1, z_2, \ldots, z_{P_n})$ denotes a P_n-dimensional sample of z.

Parameters of the regression models (a_b^n) were estimated using the Lawson-Hanson algorithm for non-negative least square estimation by solving:

$$a_b^n = \underset{y, y \geq 0}{\operatorname{argmin}} ||f_n(y) - z||_2^2 \tag{2.2}$$

These models were then used to estimate the quantity of peptides, which intensities were determined in the label-free quantification step. No predictions were computed nor derived for intensity values lying outside of the observed intensity range. Quantity estimations of oxidised peptides and their counterparts were summed, and the amount estimations of the two most abundant peptides were averaged to obtain individual protein quantity estimations.

2.3.2 Analysis of variance model for differential analysis

An additional analysis of muscle type (SM, LT, ST) effect on the abundance of the proteins was performed on a subset of 51 samples including only Rouge des Prés cows to overcome the effects of animal type and rearing practices. A one-way analysis of variance (ANOVA) was performed for each protein abundance assayed by SWATH-MS, when the protein was identified in at least 80% of the samples, to evaluate their dependence on the muscle. The one-way ANOVA model can be written as such:

$$z_{ij_i} = \mu + \alpha_i + \epsilon_{ij_i} \tag{2.3}$$

where:

- $i = 1, \ldots, 3$ indexes the muscle type: $i = 1$ corresponds to muscle type SM, $i = 2$ corresponds to muscle type LT and $i = 3$ corresponds to muscle type ST,
- $j_i = 1, \ldots, J_i$ where $J_1 = 23$ (as there are 23 muscle type SM samples), $J_2 = 20$ (as there are 20 muscle type LT samples) and $J_3 = 8$ (as there are 8 muscle type ST samples),
- z_{ij_i} denotes the abundance of the protein j_i in the muscle type i,
- μ denotes the mean of all proteins,
- α_i denotes the muscle type effect, so that $\alpha_1 + \alpha_2 + \alpha_3 = 0$,
- $\epsilon_{ij_i} \sim \mathcal{N}(0, \sigma^2)$ denotes the error term.

Hypothesis testing was performed at a 5% significance level. Hence, a *post-hoc* Tukey's test for multiple comparisons was performed when the result of the ANOVA Fisher's test was significant, *i.e.* when the resulting p-value was lower or equal than 5%.

2.3.3 Software implementation

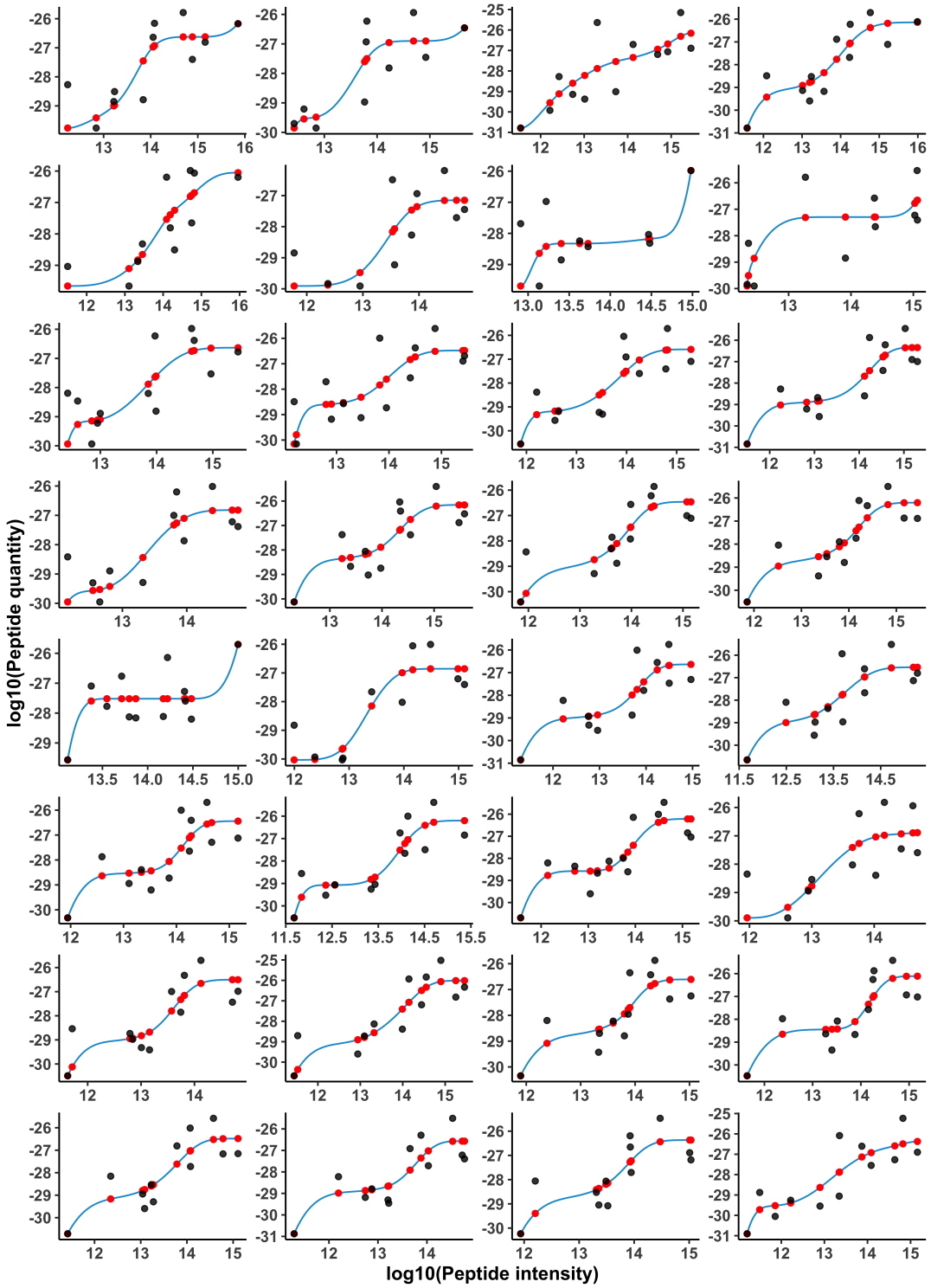
I-spline analysis was conducted using the `splines2` `R` package (Wang and Yan, 2021) and non negative least squares models were fitted using the `nnls` `R` package (Mullen and van Stokkum, 2012). The ANOVA model and the Tukey test for multiple comparisons were performed using the `agricolae` package (de Mendiburu, 2021).

2.4 Results

2.4.1 Added value of SWATH-MS for accurate protein quantification

Monotone spline smoothing was performed to estimate accurate quantification from SWATH-MS data, as described in Section 2.3.1. The fitted curves for each biological sample considered are represented in blue on Figure 2.2. The black dots represent the values used for model fitting, and the red dots are the values predicted by the model. As expected, non-linear relation between both variables was observed, and all fitted models were increasing. The location of the values estimated by the model varies largely between all biological samples, highlighting the intrinsic irregular nature of the measurements. Despite this property, defining a common basis of I-splines for all 64 samples preserves a unified framework on which the curves are fitted. However, although the same basis of functions is used, the associated coefficients are estimated for each biological sample separately, thanks to the corresponding data only. Moreover, some curves show a plateau that arises from the monotonicity constraint. Indeed, if the first data point appears to have a relatively high quantity value, the curve cannot reach data points that have lower quantity values. Notice that such locally decreasing behaviour is generally due to the presence of noise in the measurements that alters the underlying monotonic signal. The quality of fit of the monotone spline regression model was compared to the usual linear model using the root-mean-square error (RMSE). Figure 2.3 shows that monotone spline regression outperforms the linear model for almost all biological samples considered in terms of RMSE. Furthermore, an exact binomial test was performed: with a 95% confidence level, there is a probability of at least 93% that monotone spline provides a lower RMSE than linear regression.

Silva et al. (2006) proposed a protein amount estimation method based on the three most intense tryptic peptides of a given protein. As only two peptides per protein of interest were available in our work (see Table 2.1), a "top 2" strategy was applied on predicted peptide amounts to derive protein amounts. This consists of summing the 2 most abundant peptides from a given protein to infer its quantity. The accuracy between the label-based and label-free accurate quantifications was assessed on the candidate biomarkers: 53% of the amount estimations are consistent within a factor 2 with the absolute label-based quantification (Figure 2.4). The TNNT1 protein shows an atypical behaviour which can be explained by peptides' detection problems. Furthermore, high consistency ($R^2 \geq 0.70$) between both approaches was obtained for 33% of the samples, and even 83% of them when excluding the previously highlighted TNNT1 protein (Figure 2.5). Amounts were estimated for 585 additional proteins (296 proteins per sample in average), and ranged between 6.36 and 2,074 fmol/μg. Hence, our established protein amount estimation strategy offers a global profiling of the bovine muscle proteomes and thus allows gaining insights into muscle metabolism.


log10(Peptide quantity)
log10(Peptide intensity)

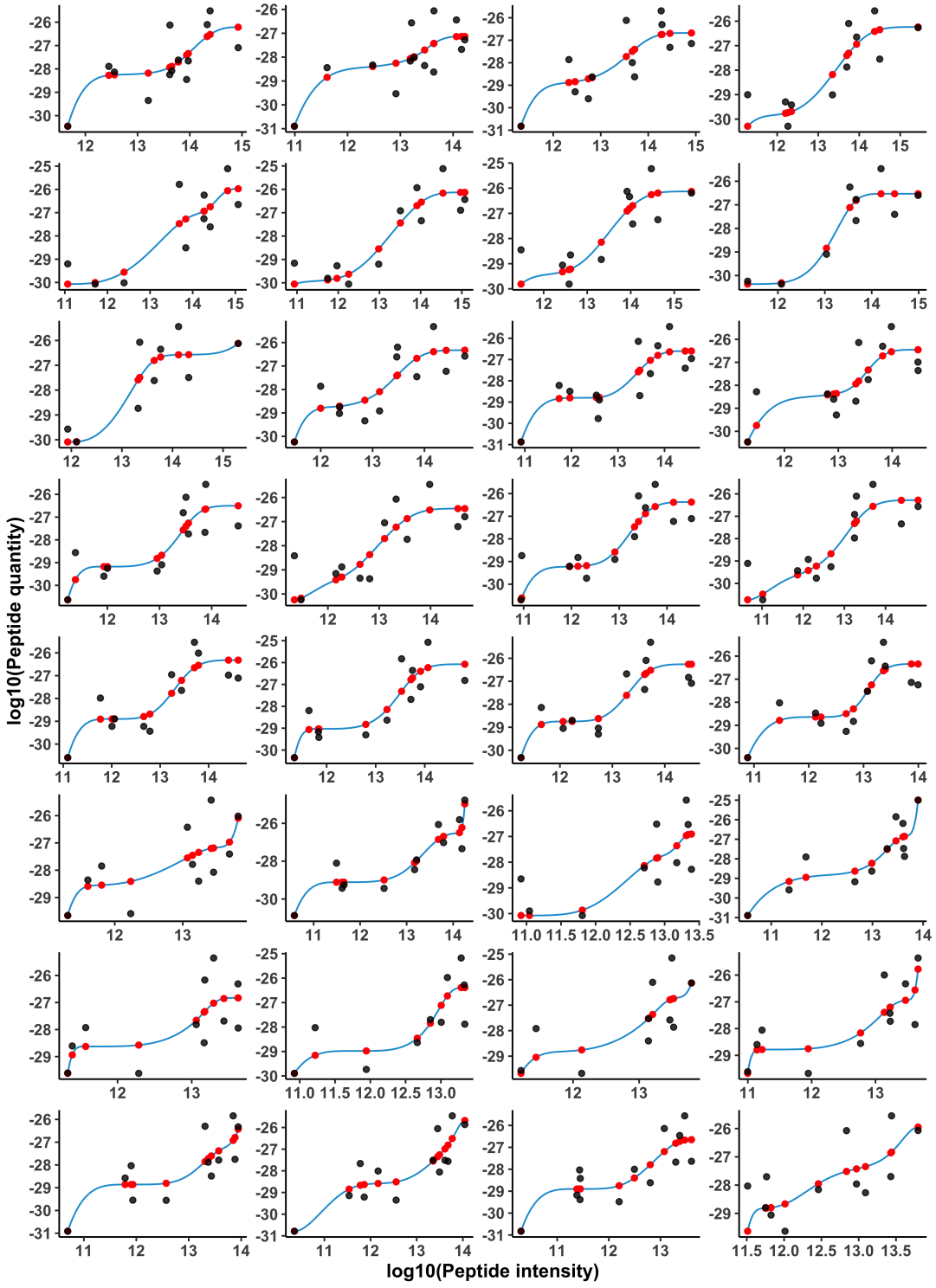


Figure 2.2: Monotone spline regression curves for the 64 biological samples considered. The black dots represent the values used for model fitting, the red dots are the values predicted by the model.

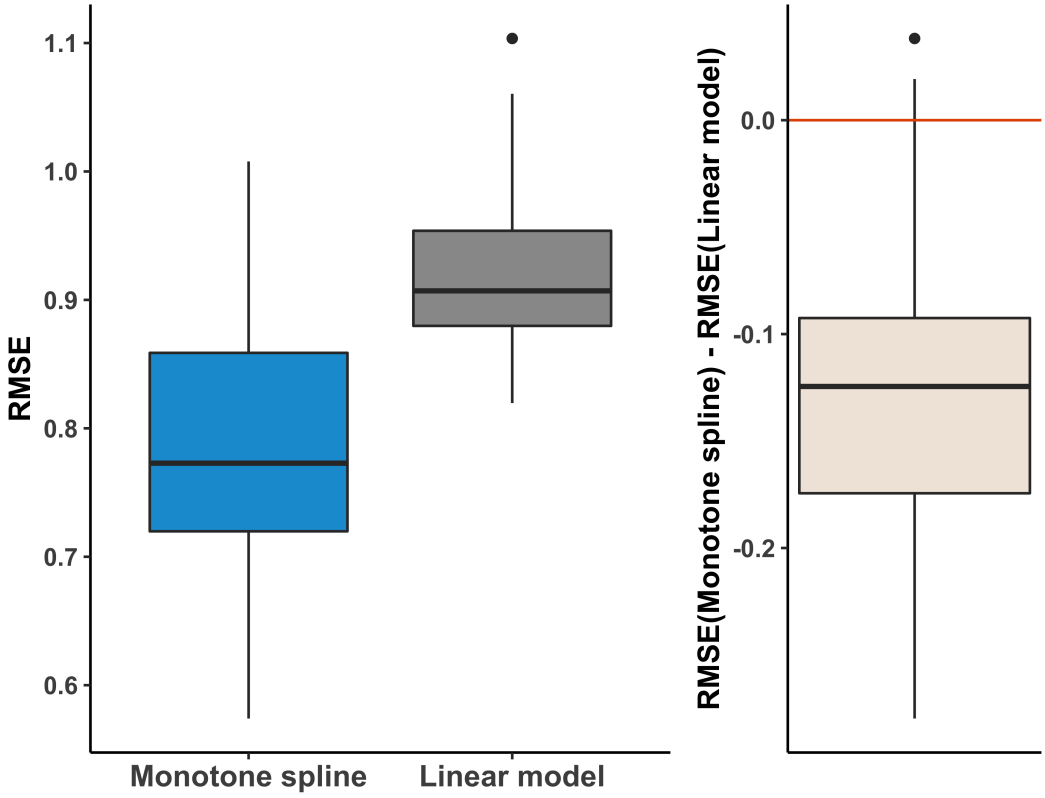


Figure 2.3: Comparison of root mean square errors (RMSE) between the monotone spline model and the linear model. Left panel depicts the comparison of the RMSE distributions for both methods. Right panel represents the difference in terms of RMSE of the monotone spline to the RMSE of the linear model.

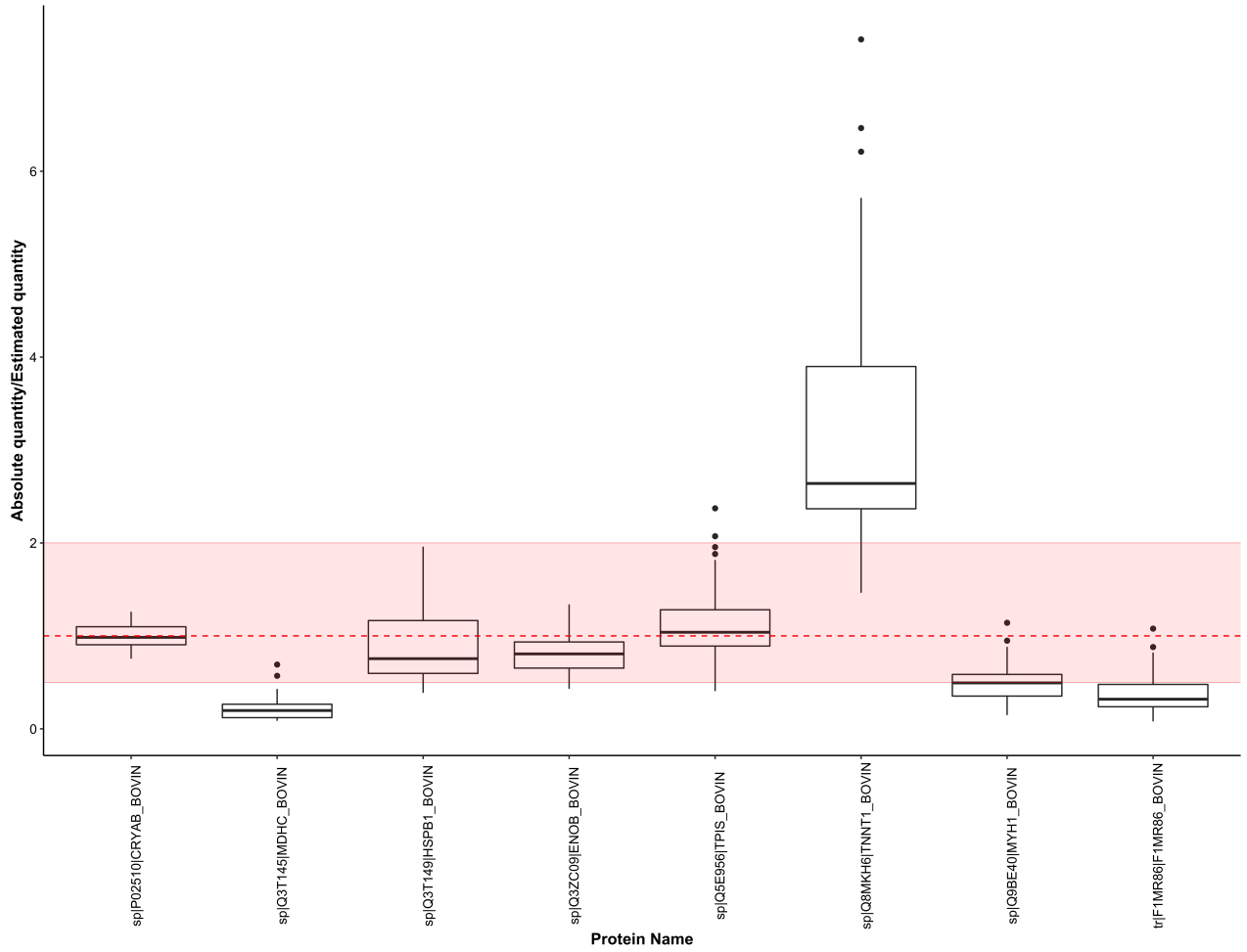


Figure 2.4: Evaluation of the accuracy of the label-free protein quantification based on SWATH-MS. Boxplots representing the distribution of the ratios of the label-based absolute quantity over the label-free estimated quantity for the eight accurately quantified proteins. The red dashed line corresponds to the expected value of 1, and the red rectangle to a factor of 2.

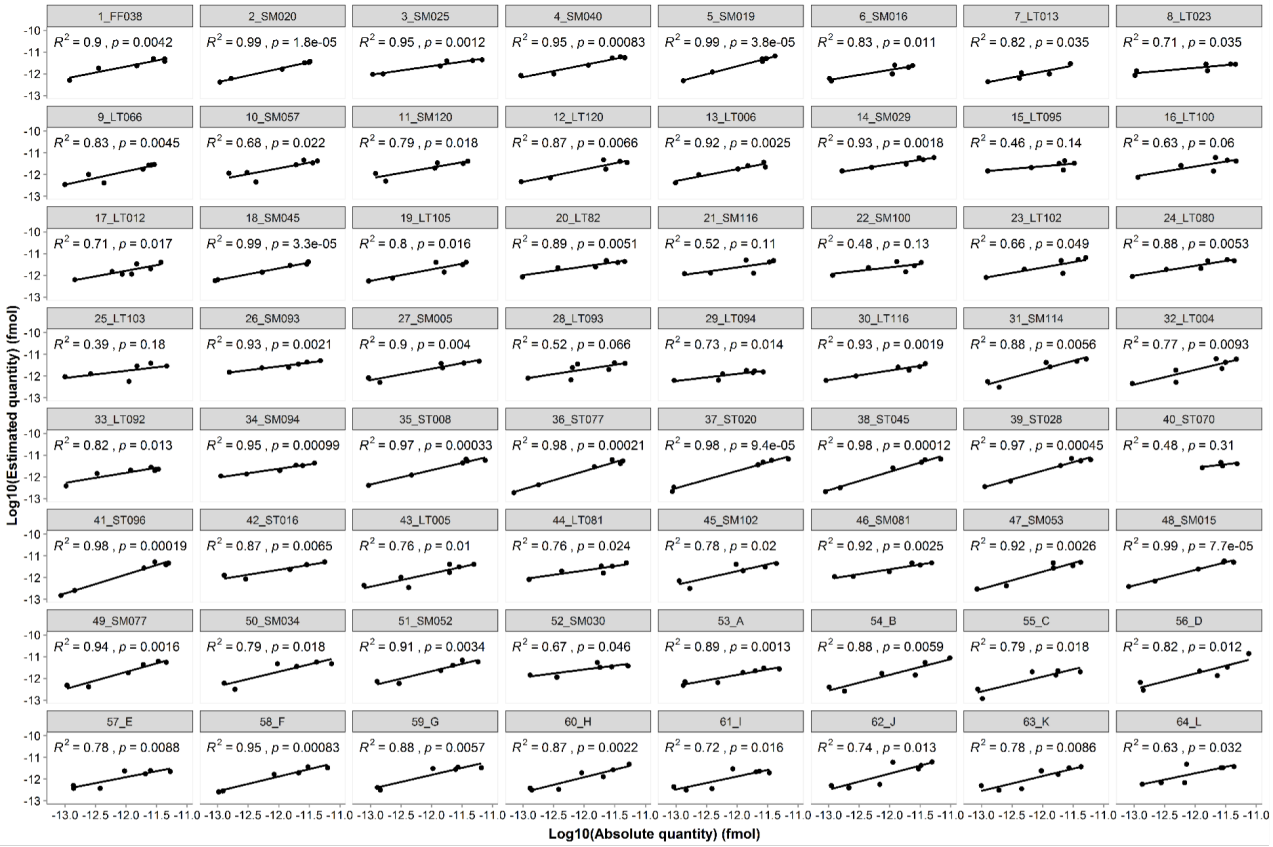


Figure 2.5: Linear quantification correlation analysis between the label-free and the label-based quantifications based on SWATH-MS when the protein TNNT1 is excluded from the comparison assay.

2.4.2 Muscle type effect of candidate biomarkers of beef tenderness or marbling

A differential analysis using the ANOVA model described in Section 2.3.2 was conducted using the quantity estimates extracted for all proteins detected in the SWATH-MS assay. A further biological analysis of these results revealed that the abundance of proteins related to glycolytic and oxidative pathways were consistent with the metabolic and contractile properties of the LT, SM and ST muscles, as depicted in Table 2.2. Indeed, among the 585 proteins quantified in the three muscles, six (GAPDHS, GAPDH, ENO1, PKM, GPI, PGK1) were annotated by the related Gene Ontology term, GO:0006096 glycolytic process (energy release from carbohydrates). Of these, phosphoglycerate kinase 1 (PGK1), pyruvate kinase (PKM), and glucose-6-phosphate isomerase (GPI) were quantified in more than 80% of the 51 muscles, and as expected were less abundant in less glycolytic muscle LT and higher abundant in the glycolytic ST and SM muscles. The low abundance of PKM, GPI and PGK1 in the oxidative highly marbled LT muscle was also consistent with the negative correlation between the abundance of these proteins and the intramuscular fat values reported by Bazile et al. (2019). Among proteins annotated by the Gene Ontology term (GO:0006099) involved in tricarboxylic acid cycle (energy release from carbohydrates, fats and proteins), FH, DLST and MDH2 were less abundant in the ST muscle, LT and SM being equal in accordance with the contractile and metabolic properties of these muscles described in the literature (Listrat et al., 2020). In summary, the differences observed between the three muscles for the 11 proteins mentioned in Table 2.2 are all consistent with the contractile and metabolic properties of the muscles: the ST contained the fewest proteins associated with the slow oxidative type and the most proteins linked to the rapid glycolytic type, the opposite is observed in the LT, the SM being intermediate.

Gene Ontology number and term	Gene Name	SM ($J_1 = 23$)				LT ($J_2 = 20$)				ST ($J_3 = 8$)				
		Mean	SD	Min	Max	Mean	SD	Min	Max	Mean	SD	Min	Max	P
GO:0006099 (tricarboxylic acid cycle)	**FH**	42.06^{ab}	15.84	14.25	78.50	44.16^{a}	15.05	17.70	73.52	28.88^{b}	12.48	11.93	53.90	0.05
	DLST	36.81	16.03	16.91	81.81	36.63	14.59	17.53	62.95	22.80	9.23	14.43	40.91	0.08
	MDH2	70.59	32.16	23.84	127.91	69.19	31.79	23.54	134.42	38.20	25.15	10.86	86.62	0.06
GO:000609 (glycolytic process)	**PKM**	401.21^{ab}	76.62	235.89	525.51	360.53^{b}	112.74	84.41	543.05	490.87 a	127.12	325.01	692.26	0.01
	GPI	575.80^{a}	103.38	389.76	811.27	467.79^{b}	126.87	125.35	653.44	574.66^{ab}	103.57	430.94	788.27	0.007
	PGK1	751.60^{ab}	164.79	381.34	1057.51	637.15^{b}	207.03	139.82	958.69	914.26^{a}	142.31	734.42	1097.47	0.002
Not annotated	**TNNT3**	60.19^{b}	42.82	16.24	207.74	86.85^{b}	46.31	27.22	213.30	137.29^{a}	67.20	48.35	253.68	0.001
GO:0006635, (fatty acid beta-oxidation)	**ECHS1**	47.27^{ab}	19.01	16.53	84.17	59.92^{a}	35.97	22.93	179.49	32.26^{b}	15.21	11.47	63.29	0.05
GO:0006122 (mitochondrial electron transport ubiquinol to cytochrome c)	**UQCRC2**	32.20	11.48	16.61	59.92	37.42	14.36	16.96	64.07	23.72	9.52	10.89	35.34	0.08
GO:0006123 (mitochondrial electron transport cytochrome c to oxygen)	**COX5A**	36.68^{a}	13.25	15.12	60.79	36.01^{ab}	14.02	16.05	63.61	22.20^{b}	14.67	10.87	53.08	0.05
	MT_CO2	39.22	19.69	7.40	81.95	43.65	17.16	17.34	75.40	25.64	17.52	14.54	60.39	0.14

Table 2.2: Protein abundances assayed by SWATH-MS in up to 51 samples composed of *longissimus thoracis* (LT) *semimembranosus* (SM) and *semitendinosus* (ST) muscles. SD: Standard deviation. Values followed by different letters (a, b) are significantly different from each other at a 5% significance level. Gene ontology annotations within the biological process category were identified using the PROTEINside werb service (https://www.proteinside.org/; Kaspric et al. (2015))

2.5 Conclusion

We performed monotone spline smoothing using spiked-in heavy labelled synthetic peptides label-based quantification data to explain absolute amounts of targeted proteins by their intensities. Our approach led to a better fit than the simple linear regression in terms of root-mean-square error. Then, we estimated absolute quantities from their intensities, for all proteins quantified using a fragment-extracted chromatogram approach, thus extending the quantified proteome coverage. Further biological analysis of the predicted absolute protein quantities showed that our results were consistent with the literature on bovine muscles. However, monotone spline smoothing suffers from severe drawbacks from a statistical perspective. First, splines are known for pathological behaviour close to the domain boundaries. Moreover, splines do not provide a proper quantification for uncertainty in a probabilistic context (although some approximations in this sense were proposed as in Ramsay (1988)). More importantly, we face a problem of over-parametrisation leading to over-fitting issues, as small samples (from 9 to 13 observations for each model) are used to estimate six splines' coefficients (as we previously fixed the number of knots to 5). Finally, the fitting curves in Figure 2.2 show similar patterns of variation, suggesting a similar underlying phenomenon, which could benefit from being modelled as such.

2.6 Perspectives: Gaussian processes with shared covariance

2.6.1 Modelling

Suppose that we observe batches of functional data coming from multiple sources, corresponding to multiple observations of the same phenomenon. In terms of variations, we could fairly assume that the underlying functions to reconstruct from the data present many common properties and a behaviour somehow characteristic of the studied phenomenon. When modelling functional data using Gaussian processes, it has been recalled in Duvenaud (2014) that all properties of the sampled functions are characterised by GP's covariance structure (*i.e.* the kernel). Thus, by sharing their covariance structure, for instance by defining a common set of kernel's hyper-parameters for all GPs, one can enforce the resulting functions to present analogous properties. More importantly, when learning optimal values of the hyper-parameters in such a framework, all batches of data are used to optimise a unique set of hyper-parameters, thus avoiding the pitfalls of over-fitting and resulting in much more robust estimates. When moving towards the prediction step, the functions sampled from the posterior distributions would share common properties learned from all batches of data while preserving a specific trend and values (fitting only one batch of functional data). Such a procedure recalls the philosophy of a classic idea called *multi-task learning* in the machine learning literature (Caruana, 1997). As previously discussed above, our proteomic context

typically falls in such context. Thus, the biological samples considered in our experiment can be seen as batches of data describing the same phenomenon, namely the relation between the intensity and the quantity of a given peptide in the sample. In order to clarify the relationships between the different quantities, let us illustrate them with the associated graphical model in Figure 2.6.

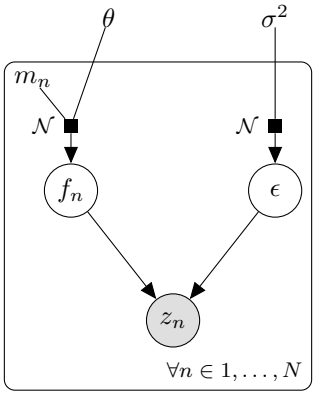


Figure 2.6: Graphical model of dependencies between variables in the Gaussian Process with shared covariance structure model.

Maintaining the same notation as previously, the generative model can be written as:

$$z = f_n(y) + \varepsilon, \ \forall n \in 1, \ldots, N, \tag{2.4}$$

where:

- $f_n(\cdot) \sim \mathcal{GP}(m_n(\cdot), \Sigma_\theta(\cdot, \cdot)), \ \forall n \in 1, \ldots, N,$
- $\varepsilon(\cdot) \sim \mathcal{GP}(0, \sigma^2 I)$ is the error term,

with:

- $m_n(\cdot)$, arbitrary prior mean functions,
- $\Sigma_\theta(\cdot, \cdot)$, a covariance kernel of hyper-parameters θ,
- $\sigma^2 \in \mathbb{R}^+$, the noise variance.

Notice that slightly different models can be achieved by modifying some assumptions. For example, when considering a prior mean parameter common to all functions (*i.e.* $m(\cdot) = m_n(\cdot), \forall n = 1, \ldots, N$), each function f_n becomes a realisation of the exact same prior GP, and the predictive distribution only differs when conditioning over the observed data. Besides, practitioners might want to consider the error terms to be sample-specific (*i.e.* $\epsilon_n(\cdot) \sim \mathcal{GP}(0, \sigma_n^2 I)$) and this assumptions would still lead to a tractable inference, although it would highly increase the number of hyper-parameters to learn.

Assuming independence between all f_n and ε and considering a set of N observed samples $\{(\boldsymbol{y_1}, \boldsymbol{z_1}), \ldots, (\boldsymbol{y_N}, \boldsymbol{z_N})\}$, we can express the likelihood of the model as:

$$p(\boldsymbol{z_n} \mid \boldsymbol{y_n}) \sim \mathcal{N}(\boldsymbol{z_n};\ m_n(\boldsymbol{y_n}), \Sigma_\theta(\boldsymbol{y_n}, \boldsymbol{y_n}) + \sigma^2 I),\ \forall n \in 1, \ldots, N.$$

Recall that each input-output couple $(\boldsymbol{y_n}, \boldsymbol{z_n})$ is a set of two vectors, thus the above distribution is a P_n-dimensional Gaussian vector as well. As in most of GP frameworks, the present inference is based on an *empirical-Bayes* approach, by computing maximum-likelihood estimates of the hyper-parameters.

2.6.2 Inference

For the sake of concision, let us note $\boldsymbol{\Psi}^n_{\theta,\sigma^2} = \Sigma_\theta(\boldsymbol{y_n}, \boldsymbol{y_n}) + \sigma^2 I$ as well as $\boldsymbol{z} = \{\boldsymbol{z_1}, \ldots, \boldsymbol{z_N}\}$ and $\boldsymbol{y} = \{\boldsymbol{y_1}, \ldots, \boldsymbol{y_N}\}$. Therefore, the inference step comes down to the following Proposition 2.1.

Proposition 2.1. *The maximisation problem for computing maximum likelihood estimates of the hyper-parameters is defined as:*

$$\begin{aligned}\left(\hat{\theta}, \hat{\sigma}^2\right) &= \underset{(\theta,\sigma^2)}{\operatorname{argmax}}\, p(\boldsymbol{z} \mid \boldsymbol{y}, \theta, \sigma^2) \\ &= \underset{(\theta,\sigma^2)}{\operatorname{argmax}} -\frac{1}{2}\sum_{n=1}^{N} \log|\boldsymbol{\Psi}^n_{\theta,\sigma^2}| + (\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))^\intercal {\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1} (\boldsymbol{y_n} - m_n(\boldsymbol{y_n})).\end{aligned}$$

Proof. Considering the independence between all samples, which share their hyper-parameters, the complete log-likelihood $\mathcal{L}$ is straightforward to decompose as:

$$\begin{aligned}\log p(\boldsymbol{z} \mid \boldsymbol{y}, \theta, \sigma^2) &= \log \prod_{n=1}^{N} p(\boldsymbol{z_n} \mid \boldsymbol{y_n}, \theta, \sigma^2) \\ &= \sum_{n=1}^{N} \log \mathcal{N}\left(\boldsymbol{z_n};\ m_n(\boldsymbol{y_n}), \boldsymbol{\Psi}^n_{\theta,\sigma^2}\right) \\ &= -\frac{1}{2}\sum_{n=1}^{N} \log|\boldsymbol{\Psi}^n_{\theta,\sigma^2}| + (\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))^\intercal {\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1} (\boldsymbol{y_n} - m_n(\boldsymbol{y_n})) + C\end{aligned}$$

The logarithm being an increasing function maximising the above expression would lead to the desired result. Moreover, we can also derive analytical gradients that may be leveraged through gradient-based optimisation algorithms. Let us consider the case of an arbitrary hyper-parameter $\gamma \in \{\theta, \sigma^2\}$ for illustration purpose:

$$\frac{\partial \mathcal{L}(\gamma)}{\partial \gamma} = \frac{\partial \log p(\boldsymbol{z} \mid \boldsymbol{y}, \theta, \sigma^2)}{\partial \gamma}$$

$$
\begin{aligned}
&= \frac{\partial}{\partial \gamma}\left[-\frac{1}{2}\sum_{n=1}^{N}\log|\boldsymbol{\Psi}^n_{\theta,\sigma^2}| + (\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))^\intercal {\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1}(\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))\right] \\
&= -\frac{1}{2}\sum_{n=1}^{N}\mathrm{tr}\left(\frac{\partial}{\partial \gamma}\left[\log \boldsymbol{\Psi}^n_{\theta,\sigma^2}\right]\right) + (\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))^\intercal \frac{\partial {\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1}}{\partial \gamma}(\boldsymbol{y_n} - m_n(\boldsymbol{y_n})) \\
&= -\frac{1}{2}\sum_{n=1}^{N}\mathrm{tr}\left({\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1}\frac{\partial \boldsymbol{\Psi}^n_{\theta,\sigma^2}}{\partial \gamma}\right) - (\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))^\intercal {\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1}\frac{\partial \boldsymbol{\Psi}^n_{\theta,\sigma^2}}{\partial \gamma}{\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1}(\boldsymbol{y_n} - m_n(\boldsymbol{y_n})) \\
&= -\frac{1}{2}\sum_{n=1}^{N}\mathrm{tr}\left(\left[\underbrace{{\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1} - {\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1}(\boldsymbol{y_n} - m_n(\boldsymbol{y_n})(\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))^\intercal {\boldsymbol{\Psi}^n_{\theta,\sigma^2}}^{-1}}_{\mathcal{A}}\right]\frac{\partial \boldsymbol{\Psi}^n_{\theta,\sigma^2}}{\partial \gamma}\right)
\end{aligned}
$$

Note that the term $\mathcal{A}$ does not depend upon the derivation with respect to γ and thus remains constant for all elements of the gradient. The derivative $\frac{\partial \boldsymbol{\Psi}^n_{\theta,\sigma^2}}{\partial \gamma}$ is specific to the kernel definition that is used in the model, and is defined as a covariance matrix where each element is the derivative with respect to γ of the corresponding element in the original matrix. □

Let us point out that, since classical kernels used for GP regression generally rely on a handful of hyper-parameters, the over fitting and over parametrisation issues pointed out with I-splines regression disappear in this context. Where 64×6 parameters were estimated in the previous models, this number decreases to 5 with GPs (using a linear and squared exponential kernel) in the following application on the same data.

2.6.3 Prediction

Contrarily to many regression methods, a final step is required for making predictions once the learning is completed. Hence, although optimal values of the hyper-parameters have been estimated in the prior GP distributions, the prediction step consists in computing the posteriors by conditioning over observed data. To this end, let us introduce $\boldsymbol{z_n^*}$ the target output vector (of arbitrary dimension) associated with inputs $\boldsymbol{y_n^*}$, for which we seek a prediction. As a GP is an infinite-dimensional object, the finite-dimensional evaluation of the joint vector $(\boldsymbol{z_n}, \boldsymbol{z_n^*})^\intercal$ is Gaussian such as:

$$
p(\begin{bmatrix}\boldsymbol{z_n}\\ \boldsymbol{z_n^*}\end{bmatrix} \mid \begin{bmatrix}\boldsymbol{y_n}\\ \boldsymbol{y_n^*}\end{bmatrix}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y_n}\\ \boldsymbol{y_n^*}\end{bmatrix}; \begin{bmatrix}m_n(\boldsymbol{y_n})\\ m_n(\boldsymbol{y_n^*})\end{bmatrix}, \begin{pmatrix}\boldsymbol{\Psi}^{nn}_{\hat{\theta},\hat{\sigma}^2} & \boldsymbol{\Psi}^{n*}_{\hat{\theta},\hat{\sigma}^2}\\ \boldsymbol{\Psi}^{*n}_{\hat{\theta},\hat{\sigma}^2} & \boldsymbol{\Psi}^{**}_{\hat{\theta},\hat{\sigma}^2}\end{pmatrix}\right).
$$

For such Gaussian vector, it is well-known that the conditional remains Gaussian, and the posterior distribution can thus be computed as (Rasmussen et al., 2006):

$$
p(\boldsymbol{z_n^*} \mid \boldsymbol{z_n}, \boldsymbol{y_n^*}, \boldsymbol{y_n}) = \mathcal{N}\left(\boldsymbol{z_n^*}; \hat{m}_n^*, \hat{\boldsymbol{\Psi}}^*_n\right),
$$

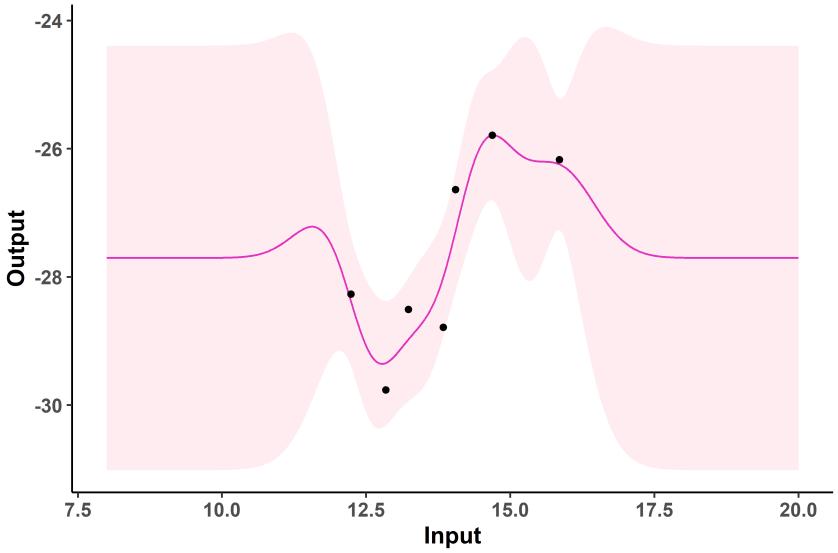


Figure 2.7: Illustration of the Gaussian Process regression using the targeted quantification data. The data points (in black) coming from one biological sample $(\boldsymbol{y_n}, \boldsymbol{z_n})$ were used for computing the posterior predictive curve and its associated 95% credible interval.

where:

- $\hat{m}_n^* = m_n(\boldsymbol{y_n^*}) + \boldsymbol{\Psi}^{*n}_{\hat{\theta},\hat{\sigma}^2} {\boldsymbol{\Psi}^{nn}_{\hat{\theta},\hat{\sigma}^2}}^{-1} (\boldsymbol{y_n} - m_n(\boldsymbol{y_n}))$,
- $\hat{\boldsymbol{\Psi}}_n^* = \boldsymbol{\Psi}^{**}_{\hat{\theta},\hat{\sigma}^2} - \boldsymbol{\Psi}^{*n}_{\hat{\theta},\hat{\sigma}^2} {\boldsymbol{\Psi}^{nn}_{\hat{\theta},\hat{\sigma}^2}}^{-1} \boldsymbol{\Psi}^{n*}_{\hat{\theta},\hat{\sigma}^2}$.

Thanks to this analytical distribution, the posterior mean constitutes a prediction for any target input, and credible intervals can be derived from the associated posterior variance values as illustrated in Figure 2.7. One can notice that the prediction closely fits data points, with an uncertainty that is naturally adjusting according to the distance between targets and observations.

Moreover, outside the range of observed values, this prediction slowly dives towards the prior mean function (which is set constant here) while the credible interval rapidly widens as it moves away from data. This behaviour is typical from GP regression and highlights in an elegant way how Bayesian inference, even in such non-parametric frameworks, takes advantage of the information contained within data to update a prior belief on a phenomenon simply by computing the associated posterior distribution.

2.6.4 Experiments

For conducting the experiments on the targeted quantification dataset according to the previously introduced model, assumptions on the covariance structure were chosen close to

the previous linear or monotonic approaches. To this end, a compound kernel as the sum of the linear and squared exponential (SE) kernels has been defined as such:

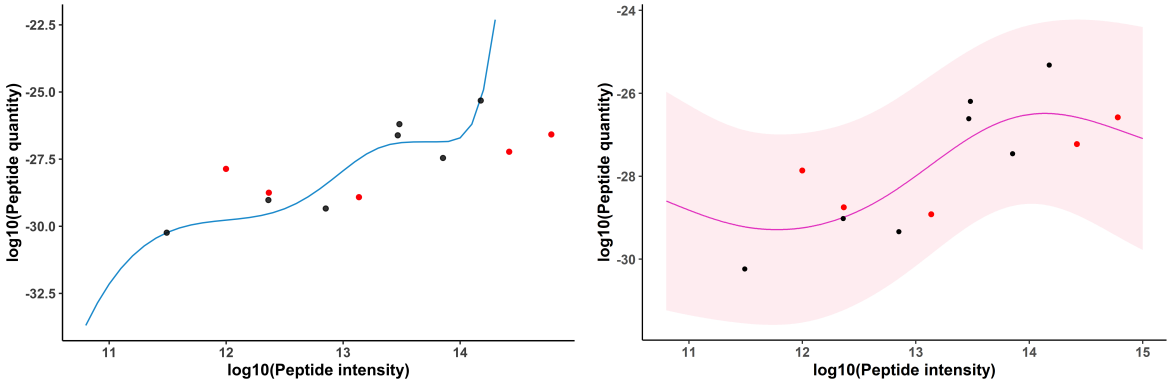
$$\Sigma_{\text{LIN}-\text{SE}}(x, x') = v^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right) + \alpha^2 x x' + \beta^2.$$

Therefore, the model presents a total of 5 hyper-parameters, with $\{v^2, \ell^2\}$ controlling the SE-term that induces flexible and smooth functions, while $\{\alpha^2, \beta^2\}$ govern the linear trend and σ^2 still accounts for the noise. The initial values for all these hyper-parameters used for leaning were chosen randomly and the optimisation procedure were run with multiple initialisations to increase chances of finding a global maximum of the likelihood. In practice, almost all initialisations led to identical optima, increasing the confidence in the robustness we could expect of the learning.

Note that in the previous sections, the parameters of the monotone spline smoothing models were estimated using all observations for each biological sample. Similarly, the evaluation of the fitting performance was conducted by calculating the RMSE on all those observations. This approach is arguable from a statistical point of view. In this section, the targeted quantification dataset was thus split into a training set and a testing set in this section. First, the training dataset was constituted by randomly drawing seven observations from each batch of data (corresponding to each biological sample n considered). Then, the remaining observations were passed onto the testing dataset. Hence, the training dataset was composed of 448 observations across the 64 biological samples considered, and the testing dataset was composed of 291 observations across the 64 batches. The parameters of the monotone spline smoothing models were estimated using the training dataset for each batch separately, whereas the hyper-parameters of the GPs regression model were estimated using all observations at once. The fitting performance of both methods was then evaluated by calculating the RMSE on the testing subset for each batch.

Figure 2.8 provides an example on a particular biological sample of the fitted curves for both methods considered. As previously highlighted, monotone spline smoothing performs poorly beyond boundaries knots (Figure 2.8a). These pathological behaviours are generally avoided in GPs frameworks (Figure 2.8b), as the prediction tends towards the prior mean function with increasing uncertainty in the absence of data (see Section 2.6.3). Furthermore, the confidence intervals that arise from the GPs framework provide the practitioners a valuable tool to assess the confidence they may grant to the quantity estimations obtained.

Table 2.3 compares the performance between the three prediction methods considered in this chapter, in terms of RMSE distribution, 95%-confidence interval coverage, number of parameters to estimate as well as training and testing durations. As usual with probabilistic methods, uncertainty quantification comes with a price in terms of computational resources, as it is well-known that GPs have $\mathcal{O}(P_n^3)$ complexity (Bishop, 2006). Although the training step using GPs is computationally time-consuming (note that it only needs to



(a) Monotone spline smoothing.

(b) Gaussian processes.

Figure 2.8: Comparison of the fitted curves with respect to the training data (represented with black points) as well as testing data (represented with red points), on a given biological sample.

	Mean	SD	D1	Q1	Median	Q3	D9	$CI_{95\%}$ coverage	Number of parameters	Training time	Prediction time
Linear	1.15	0.26	0.83	0.98	1.14	1.33	1.51	/	128	0.3	0.1
Splines	28.4	89.26	0.82	1.08	1.31	4.24	67.50	/	384	0.5	0.1
GPs	1.22	0.35	0.81	0.99	1.23	1.44	1.63	0.955	5	42.4	2.9

Table 2.3: Performance comparison of linear model, monotone spline smoothing and GPs in terms of RMSE distribution, 95%-confidence interval coverage, number of parameters to estimate and training and testing time.

be performed once), the further prediction step remains relatively fast. In terms of RMSE distribution, while monotone spline smoothing suffers from severe pathological cases that drags the mean to extremely high values, linear models and GPs both have fairly good performances. Besides, we proposed to compute a $CI_{95\%}$-coverage as being the ratio of testing points effectively lying with the 95% credible interval. This measure empirically confirms that the uncertainty is adequately quantified with a value of 0.955, really close to the theoretical threshold that we shall expect. Such a result seems particularly reassuring when it comes to asses the degree of confidence that one should have in the future predictions computed for non-observed peptide intensities.

We already mentioned that a serious issue of the linear and spline approaches, which fit functions on batches of data independently, lie in their massive over-parametrisation. This problem appears particularly pregnant here with hundreds of parameters that sometimes lead to severe over-fittings, and more generally raises the question of interpretability. Since GPs achieve similar or better predictive performances with a handful of shared hyper-parameters, the proposed model surely provides a more parsimonious and meaningful representation of the underlying structure of the relationships between peptide intensities and quantities in biological samples.

2.6.5 Limits and possible extension

As previously noted, we used constant prior mean functions, which is classical in GP frameworks, that the prediction curves tend towards in the absence of data. Although allowing the introduction of expert knowledge to help the learning, this approach often remains insufficient when it comes to forecasting values outside the range of observed data. However, a new algorithm has recently been developed (Leroy et al., 2020) to improve long-term predictions in GP models by assuming a common mean process across multiple batches of data, as suggested by the results of our study. Considering the similar trends between all biological samples, our predictions would surely benefit from an additional information sharing and we shall consider enhancing our applicative results in a near future from this method.

3

Accounting for multiple imputation-induced variability in label-free quantitative proteomics

This chapter presents a new methodology which aims at accounting for multiple imputation-induced variability downstream the statistical analysis in differential proteomics experiments (Chion et al., 2021a).

3.1 Introduction

3.1.1 Context

Missing values in label-free quantitative proteomics arise from a plethora of reasons, as described in Section 1.2.2.c. Restraining the quantitative dataset to a complete-case one by removing all analytes with missing values might create a biased analysis or remove analytes that could be of interest in the context of differential analysis. Therefore, imputing missing values is common practice in label-free quantitative proteomics. Imputation aims at replacing a missing value with a user-defined one. However, the imputation itself may not be optimally considered downstream of the imputation process, as imputed datasets are often considered as if they had always been complete. Hence, the uncertainty due to the imputation is not adequately taken into account. We provide a rigorous multiple imputation strategy, leading to a less biased estimation of the parameters' variability thanks to Rubin's rules. The imputation-based peptide's intensities' variance estimator is then moderated using Bayesian hierarchical models. This estimator is finally included in moderated t-test statistics to provide differential analyses results. This workflow can be used both at peptide and protein-level in quantification datasets. For protein-level results based on peptide-level quantification data, an aggregation step is also included.

Results: Our methodology, named `mi4p`, was compared to the state-of-the-art `limma` workflow implemented in the `DAPAR` `R` package, both on simulated and real datasets. We observed a trade-off between sensitivity and specificity, while the overall performance of `mi4p` outperforms `DAPAR` in terms of F-Score.

Availability: The methodology here described is implemented under the `R` environment

and can be found on GitHub: `https://github.com/mariechion/mi4p`. The `R` scripts which led to the results presented here can also be found on this repository. The real datasets are available on ProteomeXchange under the dataset identifiers PXD003841 and PXD027800.

3.1.2 Model

Consider a quantitative proteomics experiment in which the intensities of P analytes in N samples are measured. The N samples are split into K experimental conditions to be compared. This experiment can be represented in terms of a linear model for each analyte p. Let Y_{pn} be the random variable representing the $\log_2$-intensity of the analyte p in the sample n and y_{pn} its realisation. The following linear model is considered, according to Phipson et al. (2016):

$$\mathbb{E}(Y_p) = \boldsymbol{X}\boldsymbol{\beta_p}, \tag{3.1}$$

where:

$Y_p = (Y_{p1}, \ldots, Y_{pN})^{\mathrm{T}}$

$\boldsymbol{X}$ is a $N \times K$ full rank matrix which corresponds to the design matrix.

$\boldsymbol{\beta_p} = (\beta_{p1}, \ldots, \beta_{pK})$ is the unknown vector of the model parameters which describes the average expression levels in each experimental condition.

For each analyte p, Y_{pn} are assumed to be independent with:

$$\mathbb{V}\mathrm{ar}(Y_p) = \sigma_p^2, \tag{3.2}$$

where σ_p^2 is the unknown variance.

3.2 Methodology description

3.2.1 Multiple imputation

This work focuses on some of the most commonly used methods, which are described in Table 3.1. The k-nearest neighbours (`kNN`) method imputes missing values by averaging the k-nearest observations of the given missing value in terms of Euclidean distance. This method was described by Hastie et al. (2001) and Troyanskaya et al. (2001) and implemented in the `impute R` package (Hastie et al., 2021). The Maximum Likelihood Estimation (`MLE`) method imputed missing values using the EM algorithm proposed by Schafer (1997) and implemented in the `imp4p R` package (Giai Gianetto, 2021). The Bayesian linear regression (`norm`) method imputes missing values using the normal model and following the method described by Rubin (1987) and implemented by van Buuren and Groothuis-Oudshoorn (2011) in the `mice R` package. The Principal Component Analysis (`PCA`) method imputes missing values using

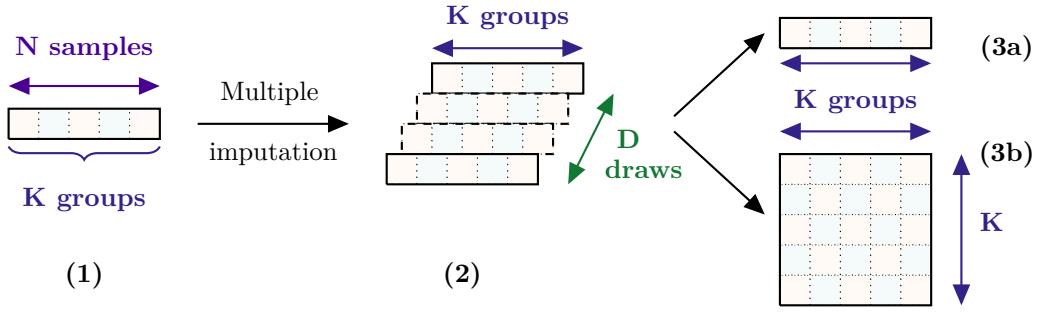


Figure 3.1: Multiple imputation strategy. **(1)** Initial dataset with missing values. It is supposed to have N observations that are split into K groups. **(2)** Multiple imputation provides D estimators for the vector of parameters of interest. **(3a)** The D estimators are combined using the first Rubin's rule to get the combined estimator. **(3a)** The estimator of the variance-covariance matrix of the combined estimator is provided by the second Rubin's rule.

Method	Implementation	References
k-nearest neighbours	`impute.knn` (`impute` R package)	Hastie et al. (2021) Hastie et al. (2001) Troyanskaya et al. (2001)
Maximum likelihood estimation	`impute.mle` (`imp4p` R package)	Giai Gianetto (2021) Schafer (1997) van Buuren and Groothuis-Oudshoorn (2011)
Bayesian linear regression	`mice` (`mice` R package)	Rubin (1987) Schafer (1997)
Principal component analysis	`impute.pca` (`imp4p` R package)	Giai Gianetto (2021) Husson and Josse (2012)
Random forests	`impute.RF` (`imp4p` R package)	Giai Gianetto (2021) Stekhoven and Bühlmann (2012)

Table 3.1: Overview of the imputation methods considered in this work.

the algorithm proposed by Husson and Josse (2012) and implemented in the `imp4p` R package (Giai Gianetto, 2021). The Random Forests (`RF`) method imputes missing values using the algorithm proposed by Stekhoven and Bühlmann (2012) and implemented in the `imp4p` R package (Giai Gianetto, 2021).

We repeated the imputation process D times to obtain D imputed datasets. We set the number of draws D equal to the proportion of missing values in the dataset, as recommended by White et al. (2011).

3.2.2 Estimation

The objective of multiple imputation is to estimate from D drawn datasets the vector of parameters of interest $\boldsymbol{\beta_p} = (\beta_{p1}, \ldots, \beta_{pK})$ and its variance-covariance matrix $\boldsymbol{\Sigma_p}$. Notably, accounting for multiple-imputation-based variability is possible thanks to Rubin's rules, which provide an accurate estimation of these parameters. Hence, the first Rubin's rule

provides the combined estimator of $\boldsymbol{\beta_p}$:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{p}} = \frac{1}{D}\sum_{d=1}^{D}\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}}, \tag{3.3}$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}}$ is the estimator of $\boldsymbol{\beta_p}$ in the d-imputed dataset. The second Rubin's rule gives the combined estimator of the variance-covariance matrix for each estimated vector of parameters of interest for peptide p through the D imputed datasets such as:

$$\hat{\boldsymbol{\Sigma}}_p = \frac{1}{D}\sum_{d=1}^{D} W_d + \frac{D+1}{D(D-1)}\sum_{d=1}^{D}(\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})^T(\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}), \tag{3.4}$$

where W_d denotes the variance-covariance matrix of $\hat{\boldsymbol{\beta}}_{\boldsymbol{p,d}}$, *i.e.* the variability of the vector of parameters of interest as estimated in the d-th imputed dataset.

3.2.3 Projection

State-of-the-art tests, including Student's t-test, Welch's t-test and moderated t-test, rely on the variance estimation. Here, the variability induced by multiple imputation is described by a variance-covariance matrix, given by Equation (3.4). Therefore, a projection step is required to get a univariate variance parameter.

Rubin's second rule decomposes the variability of the combined dataset as the sum of the within-imputation variability and the between-imputation variability. Thus, analytes whose values have been imputed should have a greater variance estimation than if the multiple imputation-induced variability was not accounted for. This amounts to "penalising" analytes for which intensity values were not observed and subsequently imputed. Hence, the projection method needs to be wisely chosen. Therefore, we benchmarked several projection methods:

- **max:** projects the matrix using the maximum of the elements of the matrix;

$$\hat{\sigma}_p^2 = \max_{i,j}\left(\hat{\Sigma}_{p,(i,j)}\right)$$

- **max.dg:** projects the matrix using the maximum of the diagonal elements of the matrix;

$$\hat{\sigma}_p^2 = \max_{k}\left(\hat{\Sigma}_{p,(k,k)}\right)$$

- **max.dg.pond:** projects the matrix using the maximum of the diagonal elements of the matrix, ponderated by the design matrix;

$$\hat{\sigma}_p^2 = \max_{k}\left(\hat{\Sigma}_{p,(k,k)}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)$$

- **norm.1:** projects the matrix using the one-norm;

$$\hat{\sigma}_p^2 = ||\hat{\mathbf{\Sigma}}_p||_1 = \max_j \sum_i |\hat{\Sigma}_{p,(i,j)}|$$

- **norm.F:** projects the matrix using the Frobenius norm;

$$\hat{\sigma}_p^2 = ||\hat{\mathbf{\Sigma}}_p||_F = \mathrm{tr}\left(\hat{\mathbf{\Sigma}}_p^{\mathrm{T}} \hat{\mathbf{\Sigma}}_p\right)$$

- **norm.I:** projects the matrix using the infinity norm;

$$\hat{\sigma}_p^2 = ||\hat{\mathbf{\Sigma}}_p||_\infty = \max_i \sum_j |\hat{\Sigma}_{p,(i,j)}|$$

We compared all these methods to the residual variance estimation of the linear model in the `limma` framework, described in Equation (3.1). We used the data from the *Arabidopsis thaliana* + UPS experiment, described in Section 3.4.1.a. As the aim is to take into account the variability added by the random imputation process, we compared the distributions of the variance estimations of analytes which values were imputed.

Method	Min	D1	Q1	Median	Mean	Q3	D9	Max
limma	**0.08**	**0.17**	**0.21**	**0.27**	**0.33**	**0.34**	**0.54**	**3.21**
max	0.06	0.16	0.21	0.29	0.33	0.41	0.56	1.89
max.dg	0.06	0.16	0.21	0.29	0.33	0.41	0.56	1.89
max.dg.pond	**0.1**	**0.27**	**0.36**	**0.51**	**0.58**	**0.72**	**0.97**	**3.27**
norm.1	0.1	0.25	0.33	0.45	0.52	0.64	0.87	3.27
norm.F	0.09	0.22	0.28	0.35	0.42	0.47	0.65	3.24
norm.I	0.1	0.25	0.33	0.45	0.52	0.64	0.87	3.27

Table 3.2: Comparison of the main statistics on the distributions of variance estimations obtained after projection of the covariance matrices, using different projection methods.

Figure 3.2 shows the distributions of variance estimations obtained after the projection of the covariance matrices for all previously mentioned projection methods. We can observe that the **max** and **max.dg** methods have similar distributions, as well as the **norm.1** and **norm.I** methods. Note that the **max** and **max.dg** methods sometimes lead to lower values than using the residual variance estimation. This observation can be explained by the expressions of those two projection methods, where each group's size is not taken into account. We added the **max.dg.pond** method to the benchmark in order to address this issue by multiplying the diagonal element of the covariance matrix by $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$. The **max.dg.pond** method produces the greatest values for variance estimation compared to the standard **limma** one. All these observations are confirmed by Table 3.2, where usual location parameters are provided.

Giving those observations, we chose in our work to perform projection using the following

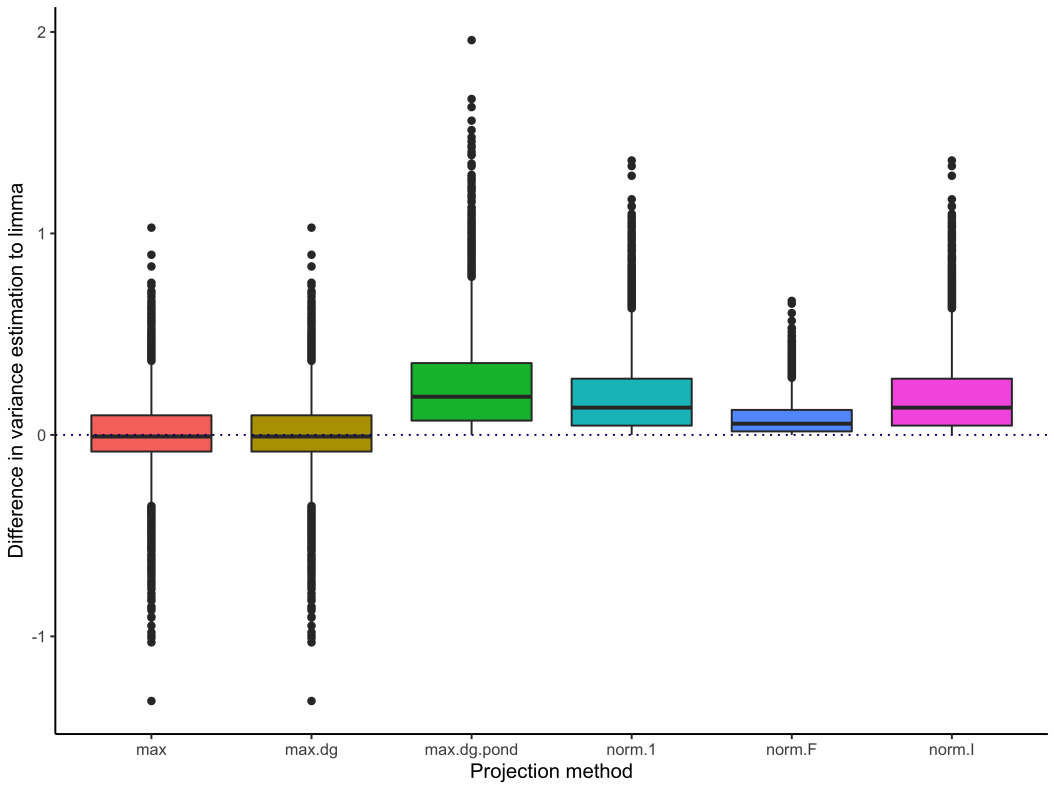


Figure 3.2: Distribution of the difference of variance estimations projected using different projection methods compared to the standard limma method.

formula (corresponding to the aforementioned **max.dg.pond** method):

$$\hat{\sigma}_p^2 = \max_k \left(\hat{\Sigma}_{p,(k,k)} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{X} \right), \tag{3.5}$$

where $\hat{\Sigma}_{p,(k,k)}$ is the k-th diagonal element of the matrix $\hat{\boldsymbol{\Sigma}}_p$ and $\boldsymbol{X}$ is the design matrix.

Nevertheless, it is to be noted that this choice for the projection method is not without consequences. Indeed, this method is the one that penalises imputed analytes the most, among all methods considered in the previous benchmark. However, analytes that show high variance estimations might be wrongly considered non differentially expressed, as their distributions in each condition to be compared can overlap.

3.2.4 Hypotheses testing

In our work, we focus our methodology on the moderated t-test introduced by Smyth (2004). This testing technique relies on the empirical Bayes procedure, commonly used in microarray data analysis, and to a more recent extent for differential analysis in quantitative proteomics (Wieczorek et al., 2017). The moderated t-test procedure relies on the following Bayesian

hierarchical model:

$$\begin{cases} \hat{\sigma}_p^2 \mid \sigma_p^2 \sim \dfrac{\sigma_p^2}{d_p} \times \chi^2_{d_p} \\ \dfrac{1}{\sigma_p^2} \sim \dfrac{1}{d_0 \times s_0^2} \times \chi^2_{d_0} \end{cases} \tag{3.6}$$

where σ_p^2 is the peptide-wise variance, d_0 and s_0 are hyperparameters to be estimated (Phipson et al., 2016). This leads to the following posterior distribution of $\frac{1}{\sigma_p^2}$ conditional to $\hat{\sigma}_p^2$:

$$\frac{1}{\sigma_p^2} \mid \hat{\sigma}_p^2 \sim \frac{1}{d_p \times \hat{\sigma}_p^2 + d_0 \times s_0^2} \chi^2_{d_p + d_0} \tag{3.7}$$

From there, a so-called moderated variance estimator $\hat{\sigma}^2_{p[\text{mod}]}$ of the variance σ_p^2 is derived from the posterior mean:

$$\hat{\sigma}^2_{p[\text{mod}]} = \frac{d_p \times \hat{\sigma}_p^2 + d_0 \times s_0^2}{d_p + d_0} \tag{3.8}$$

This estimator $\hat{\sigma}^2_{p[\text{mod}]}$ is then computed in the test statistic associated to the null hypothesis $\mathcal{H}_0 : \beta_{pj} = 0$, by replacing the usual sample variance by $\hat{\sigma}^2_{p[\text{mod}]}$ into to the classical t-statistic, just as Smyth (2004) (see Equation 3.9). Therefore, the results of this testing procedure account both for the specific structure of the data and the uncertainty caused by the multiple imputation step.

$$T_{pj[\text{mod}]} = \frac{\hat{\beta}_{pj}}{\sqrt{\hat{\sigma}^2_{p[\text{mod}]} (\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X})^{-1}_{j,j}}} \tag{3.9}$$

with $(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X})^{-1}_{j,j}$ the j-th diagonal element in the matrix $(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X})^{-1}$. Under the null hypothesis $\mathcal{H}_0$, $T_{pj[\text{mod}]}$ follows a Student distribution with $d_p + d_0$ degrees of freedom.

As there are as many tests performed as the number of peptides considered, the proportion of falsely rejected hypotheses has to be controlled. Here, the False Discovery Rate control procedure from Benjamini and Hochberg (1995) was performed using the `cp4p` `R` package, by Giai Gianetto et al. (2016).

3.2.5 Aggregation

The methodology implemented in the `mi4p` `R` package can be applied to peptide-level quantification data as well as protein-level quantification data. However, we were interested in evaluating our method at a peptide-level dataset and inferring results on a protein level, as it is common practice in proteomics. Therefore, for intensity aggregation, we chose to sum all unique peptides' intensities for each protein. We then adjusted our pipeline as follows:

1. Out-filtration of non-unique peptides from the peptide-level quantification dataset.
2. Normalisation of the $\log_2$-transformed peptide intensities.
3. Multiple imputation of $\log_2$-transformed peptide intensities.

4. Aggregation by summing all peptides intensities (non-$\log_2$-transformed) from a given protein in each imputed dataset.
5. $\log_2$-transformation of protein intensities.
6. Estimation of variance-covariance matrix.
7. Projection of the estimated variance-covariance matrix.
8. Moderated t-testing on the combined protein-level dataset

3.3 Experiments on simulated datasets

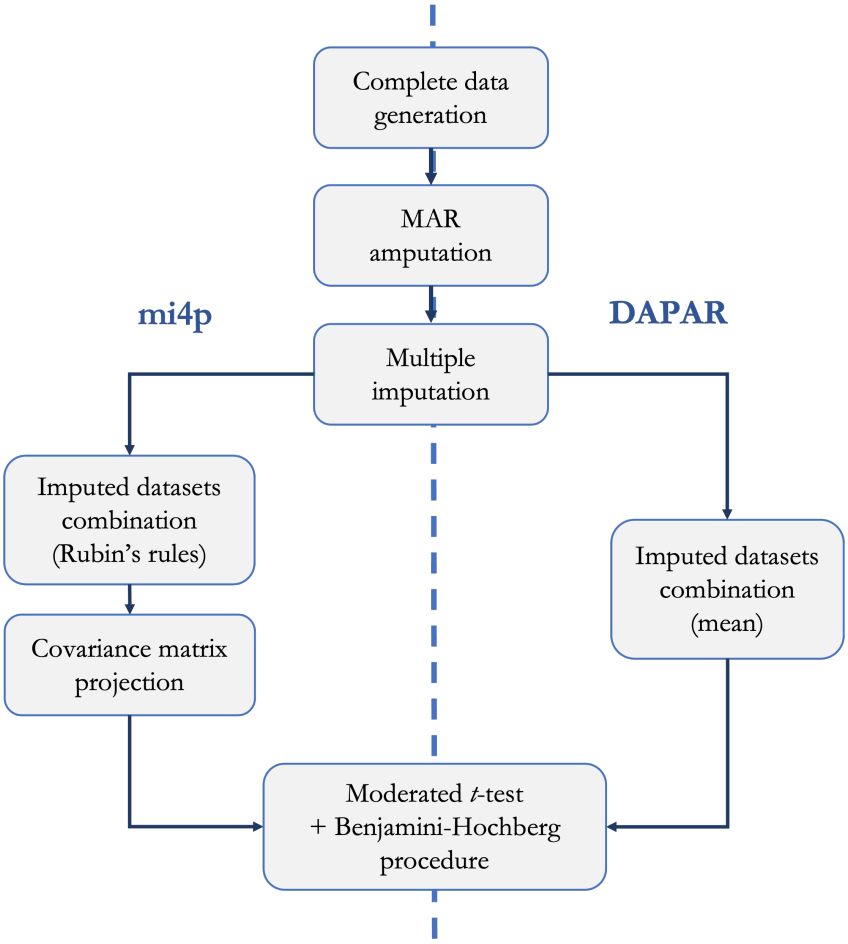


Figure 3.3: Workflow of the simulation study conducted for performance evaluation of the `mi4p` methodology and comparison to the one implemented in the `DAPAR` `R` package.

3.3.1 Under Missing At Random assumption

3.3.1.a Simulation designs

We evaluated our methodology on three types of simulated datasets. First, we considered an experimental design where the distributions of the two groups to be compared scarcely overlap. This design led to a fixed effect one-way analysis of variance model (ANOVA), which can be written as:

$$y_{pnk} = \mu + \delta_{pk} + \epsilon_{pnk} \tag{3.10}$$

with $\mu = 100$, $\delta_{pk} = 100$ if $1 \leq p \leq 10$ and $k = 2$ and $\delta_{pk} = 0$ otherwise and $\epsilon_{pnk} \sim \mathcal{N}(0, 1)$. Here, y_{pnk} represents the log-transformed abundance of peptide p in the n-th sample. Thus, we generated 100 datasets by considering 200 individuals and 10 variables, divided into 2 groups of 5 variables, using the following steps:

1. For the first 10 rows of the data frame, set as differentially expressed, draw the first 5 observations (first group) from a Gaussian distribution with a mean of 100 and a standard deviation of 1. Then draw the remaining 5 observations (second group) from a Gaussian distribution with a mean of 200 and a standard deviation of 1.

2. For the remaining 190 rows, set as non-differentially expressed, draw the first 5 observations as well as the last 5 observations from a Gaussian distribution with a mean of 100 and a standard deviation of 1.

Secondly, we considered an experimental design, where the distributions of the two groups to be compared might highly overlap. Hence, we based it on the random hierarchical ANOVA model by Lazar et al. (2016), derived from Karpievitch et al. (2012). The simulation design follows the following model:

$$y_{pnk} = P_p + G_{pk} + \epsilon_{pnk} \tag{3.11}$$

where y_{pnk} is the log-transformed abundance of peptide p in the n-th sample, P_p is the mean value of peptide p, G_{pk} is the mean differences between the condition groups, and ϵ_{pnk} is the random error terms, which stands for the peptide-wise variance. We generated 100 datasets by considering 1000 individuals and 20 variables, divided into 2 groups of 10 variables, using the following steps:

1. Generate the peptide-wise effect P_p by drawing 1000 observations from a Gaussian distribution with a mean of 1.5 and a standard deviation of 0.5.

2. Generate the group effect G_{pk} by drawing 200 observations (for the 200 individuals set as differentially expressed) from a Gaussian distribution with a mean of 1.5 and a standard deviation of 0.5 and 800 observations fixed to 0.

3. Build the first group dataset by replicating 10 times the sum of P_p and the random error term, drawn from a Gaussian distribution of mean 0 and standard deviation 0.5.

4. Build the second group dataset by replicating 10 times the sum of P_p, G_{pk} and the random error term drawn from a Gaussian distribution of mean 0 and standard deviation 0.5.

5. Bind both datasets to get the complete dataset.

Finally, we considered an experimental design similar to the second one, but with random effects P_p and G_{pk}. The 100 datasets were generated as follows.

1. For the first group, replicate 10 times (for the 10 variables in this group) a draw from a mixture of 2 Gaussian distributions. The first one has the following parameters: a mean of 1.5 and a standard deviation of 0.5 (corresponds to P_p). The second one has the following parameters: a mean of 0 and a standard deviation of 0.5 (corresponds to ϵ_{pnk}).

2. For the second group replicate 10 times (for the 10 variables in this group) a draw from a mixture of the following 3 distributions.

   (a) The first one is a Gaussian distribution with the following parameters: a mean of 1.5 and a standard deviation of 0.5 (corresponds to P_p).

   (b) The second one is the mixture of a Gaussian distribution with a mean of 1.5 and a standard deviation of 0.5 for the 200 first rows (set as differentially expressed) and a zero vector for the remaining 800 rows (set as not differentially expressed). This mixture illustrates the G_{pk} term in the previous model.

   (c) The third distribution has the following parameters: a mean of 0 and a standard deviation of 0.5 (corresponds to ϵ_{pnk}).

All simulated datasets were then amputed to produce MAR missing values in the following proportions: 1%, 5%, 10%, 15%, 20% and 25%.

3.3.1.b Comparison of imputation methodologies

To compare the imputation methods considered in this chapter and described in Table 3.1, we used the synthetic data from the aforementioned second set of MAR simulations. Let us highlight that reviews on imputation methods evaluation often base their study on real datasets by subsetting them to complete data and amputating them afterwards (Table 1.4). However, such approaches remain limited, as the parameters of the data cannot be controlled. Recall that we simulated 100 datasets, which were amputated afterwards. Hence both imputed and real values can be accessed. In this section, we aim at evaluating the potential bias that can arise from the imputation process. We based our comparison on the amputated datasets with a proportion of missing values of 10%, so we impute each dataset $D_Q = 10$ times. Consider then the set of all missing values coming from the $Q = 100$ datasets. Let

n_q denote the number of missing values in the q-th dataset, with $1 \leq q \leq Q$. The set of all missing data is then constituted of $N_Q = \sum_{q=1}^{Q} n_q$ elements. In our work, we take the number of draws for multiple imputation as the percentage of missing values. Therefore, multiple imputation produces ten vectors of size N_Q corresponding to the ten draws of the considered vector.

IMPUTATION ERROR FOR EACH DRAW To evaluate the performance of the imputation methodologies considered in this chapter, we first consider the error on each draw. Let y_i denote the i-th value in the previously defined set and $y_i^{(d)}$ the d-th draw for y_i. Hence, we define the error $\varepsilon_i^{(d)}$ for each imputed value $y_i^{(d)}$ as:

$$\varepsilon_i^{(d)} = y_i^{(d)} - y_i, \forall i \in 1, \ldots, N_Q, \forall d \in 1, \ldots, D_Q.$$

The $D_Q \times N_Q$ errors are calculated for all imputation method considered, namely `kNN`, `MLE`, `norm`, `PCA` and `RF` (detailed in Table 3.1). To compare the performances of these methods,

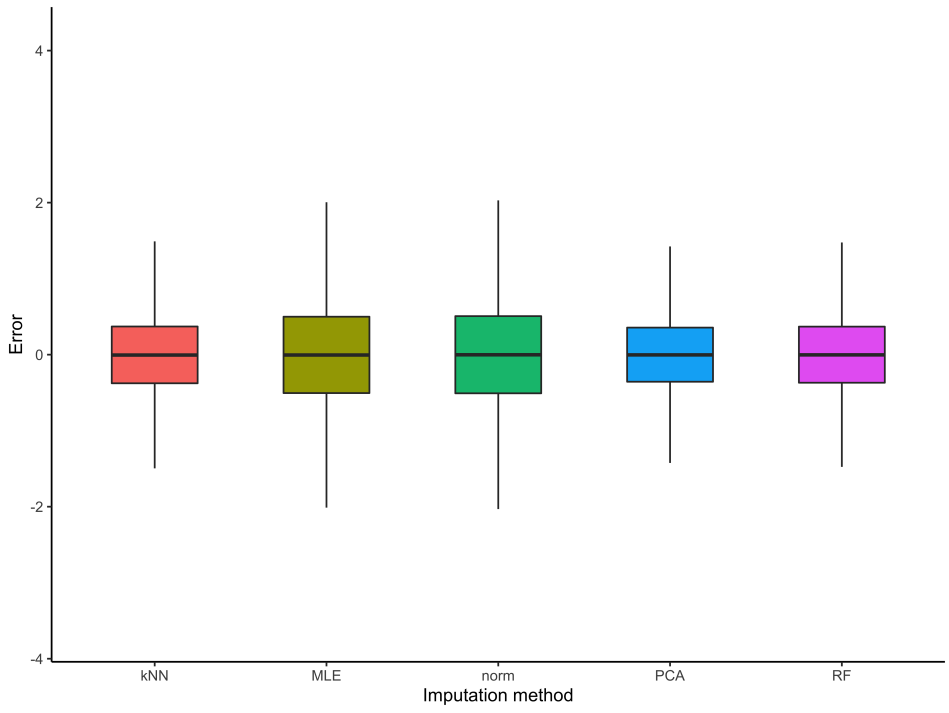


Figure 3.4: Distribution of empirical errors for the five imputation methods considered on the second set of MAR simulations.

Figure 3.4 summarises the distributions of $(\varepsilon_i^{(d)})_{i=1,\ldots,N_q, d=1,\ldots,D_Q}$ for the five imputation methods considered. First, it is comforting to observe that the errors are all centred on zero. Moreover, let us also point out that the `MLE` and `norm` methods provide a slightly increased

variability than other methods. The `kNN`, `PCA` and `RF` methods show equivalent performance as far as single imputation is concerned.

IMPUTATION ERROR FOR THE MEAN OF DRAWS Following the first Rubin's rule (Equation (3.3)), the D_Q drawn datasets are combined using the mean. In order to provide additional insights about the empirical errors of the different multiple-imputation procedures, let us compute the differences between the averaged imputed values used in practice and the actual values. For each imputation method, the errors are averaged over the D_Q draws (corresponding to the D_Q different imputations), which we expect to stabilise the error values. In contrast to the previous approach, the associated formula becomes:

$$\varepsilon_i = \frac{1}{D_Q} \sum_{d=1}^{D_Q} y_i^{(d)} - y_i, \ \forall i \in 1, \ldots, N_Q.$$

Figure 3.5 suggests equivalent performance for all five methods as far as the mean of all imputed datasets is concerned. In terms of variability, we can still observe a slightly increased interquartile range for the `MLE` imputation method.

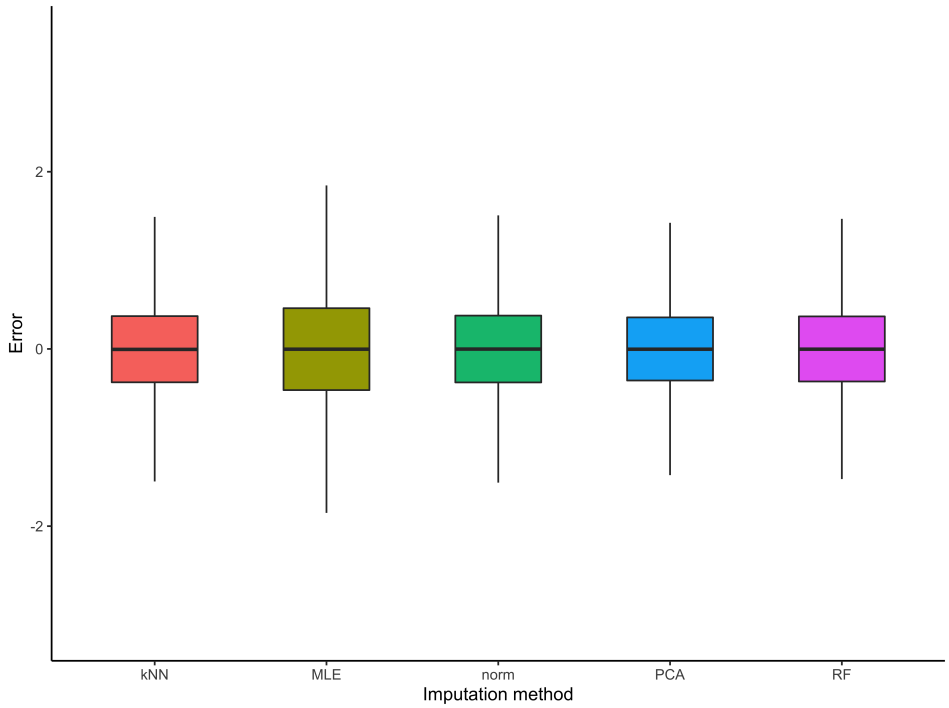


Figure 3.5: Distribution of errors of the averaged imputed values for the five imputation methods considered on the second set of MAR simulations.

COMPUTATION TIME As a complement to determine the advantages of each approach, we compared the running time of all imputation processes are compared as well. Therefore, we considered the total time needed for imputing D_Q times each simulated dataset. According to the boxplots on Figure 3.6 highlight the `MLE` and `kNN` method to be the fastest.Compared to `MLE` imputation method, the `PCA` method is on average 3.5 times slower and the `norm` and `RF` methods are respectively on average 7.4 times and 8.1 times slower. At this stage of the comparison, as all imputation methods exhibit comparable performances in terms of imputation bias, a preference can be drawn for the `kNN` and `MLE` methods.

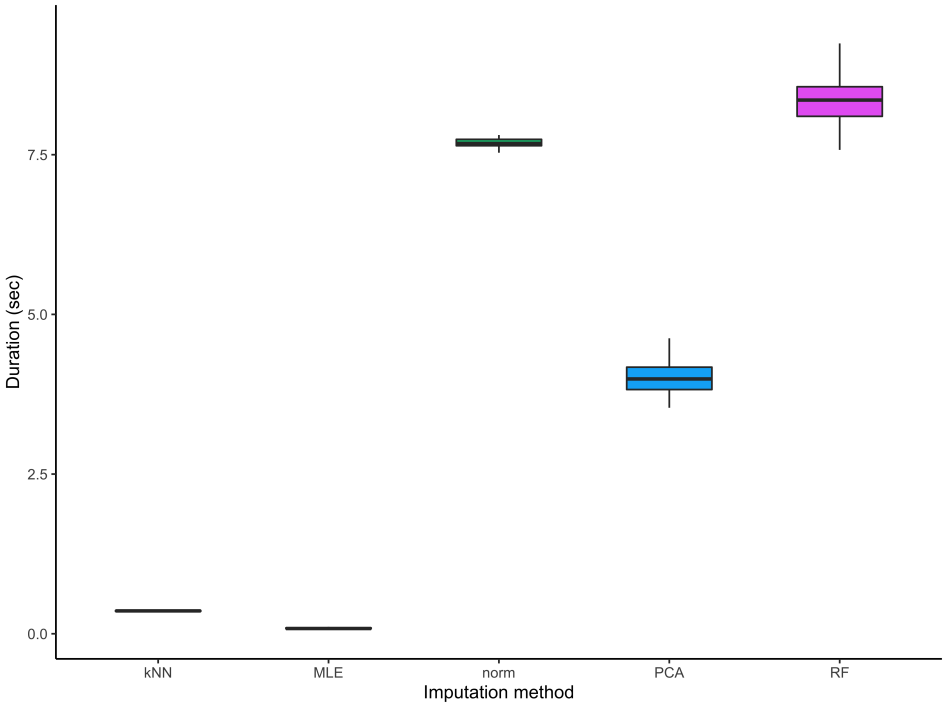


Figure 3.6: Distributions of duration of the imputation process for the five imputation methods considered on the second set of MAR simulations.

INFLUENCE ON TESTING RESULTS The evaluation of performance for our `mi4p` methodology relies on the results produced by the testing procedure. For the MAR simulation designs, testing results were provided for all imputation methods considered. However, we could observe that no positives were produced for some datasets. As a summary, Table 3.3 describes under which conditions such pathological cases arise in the second set of MAR simulations. The `mi4p` workflow dramatically underperforms at detecting positives when using the `norm` imputation method. The high number of pathological cases can be explained by this method being a global one (*i.e.* to the full dataset), whereas other methods considered are local in that they are applied experimental condition-wise. Therefore, the `norm` method might

lead to an increased between-imputation variability. Otherwise, no pathological cases occur while using the `mi4p` method on this particular set of simulated datasets. However, a few pathological cases can be consistently observed when using the `DAPAR` workflow, regardless of the chosen imputation method. Overall, the `MLE` imputation offers a slight advantage over other methods.

Imputation method	Testing workflow	1%	5%	10%	15%	20%	25%
kNN	**DAPAR**	0	0	2	2	2	1
	MI4P	0	0	0	0	0	0
MLE	**DAPAR**	0	0	2	1	1	0
	MI4P	0	0	0	0	0	0
norm	**DAPAR**	0	0	2	2	1	0
	MI4P	0	0	0	7	26	57
PCA	**DAPAR**	0	0	2	2	3	0
	MI4P	0	0	0	0	0	0
RF	**DAPAR**	0	0	3	2	3	0
	MI4P	0	0	0	0	0	0

Table 3.3: Number of pathological cases for each simulation condition on the second set of MAR simulations.

A GLIMPSE OF REAL DATASETS IMPUTATION As a conclusion of this thorough analysis of synthetic data, let us draw some perspectives for the subsequent real datasets study (see Section 3.4). At this stage, `kNN` and `MLE` imputation methods might equivalently be considered. However, in quantitative proteomics datasets, rows sometimes present more than 50% missing values. When this threshold is exceeded, current kNN method implementations only use mean imputation for these rows. However, mean imputation results in identical imputed values and no between-imputation variability arises, preventing from taking advantage of our `mi4p` methodology.

In contrast, the `MLE` imputation method still provides reliable imputations for a reduced computational cost in all situations. Moreover, the `MLE` method offers a more principled and interpretable approach compared to alternatives, which also motivated our choice to retain this method for further analysis of both MNAR + MCAR simulated datasets and real datasets.

3.3.1.c Indicators of performance

We compared our methodology to the `limma` testing pipeline implemented in the state-of-the-art `ProStaR` software, through the `DAPAR R` package (Wieczorek et al., 2017). Both aim at classifying peptides or proteins as differentially or not differentially expressed. In our work, we define a positive as a peptide/protein which is considered as differentially expressed. Similarly, we define a negative as a peptide/protein which is considered as not

differentially expressed. Hence, the results of both methods lead to the following confusion matrix for the peptides or proteins considered:

		Tested as	
		Differentially expressed	Not differentially expressed
Actually	Differentially expressed	True positives (TP)	False negatives (FN)
	Not differentially expressed	False positives (TP)	True negatives (TN)

Table 3.4: Confusion matrix of a differential proteomics experiment.

To assess the performances of both methods, we used measures based on the confusion matrix. We considered sensitivity (also known as true positive rate or recall), specificity (also known as true negative rate), precision (also known as positive predictive value), F-score and Matthews correlation coefficient.

Let TP, TN, FP and FN respectively denote the numbers of true positives, true negatives, false positives, and false negatives. Sensitivity indicates the proportion of true positives among all the positives and is expressed as (Yerushalmy, 1947):

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{3.12}$$

Specificity indicates the proportion of true negatives among all actual negatives and is expressed as (Yerushalmy, 1947):

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.13}$$

Precision indicates the proportion of true positives among all tested positives and is expressed as (Kent et al., 1955):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.14}$$

The F-Score indicates the accuracy of the testing procedure by combining sensitivity and precision (Sasaki, 2007). Note that we considered the F_1-Score, giving equal importance to sensitivity and precision. It can be expressed as:

$$F\text{−Score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} = \frac{TP}{TP + \frac{1}{2} \times (FP + FN)} \tag{3.15}$$

Chicco and Jurman (2020) claims that Matthews correlation coefficient (MCC) by Matthews (1975) should be prefered to the F_1-Score in binary classification. The MCC is expressed

as:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FP)}} \tag{3.16}$$

3.3.1.d Results and discussion

We first compared our methodology to the state-of-the-art `DAPAR` workflow (Figure 3.3) using the indicators detailed above. Following the comparison of imputation methods provided in Section 3.3.1.b, only results obtained on `MLE`-imputed datasets are detailed hereafter. Results obtained with other imputation methods can be found in the Appendix chapter.

Let us first assess the performance of the first set of MAR simulations. Note that the distributions of intensity values within each experimental condition for differentially expressed analytes are separate for this set of simulations. Indeed, let us recall that the intensity values for those analytes were drawn from a $\mathcal{N}(100, 1)$ distribution for the first condition and from a $\mathcal{N}(200, 1)$ distribution for the second one. The distributions of the differences in sensitivity, specificity, precision, F-score and Matthews correlation coefficient between `mi4p` and `DAPAR` for all missing values proportion were summarised on the boxplots on Figure 3.7. Detailed results can be additionally found in Table A.2 in the Appendix chapter. Both

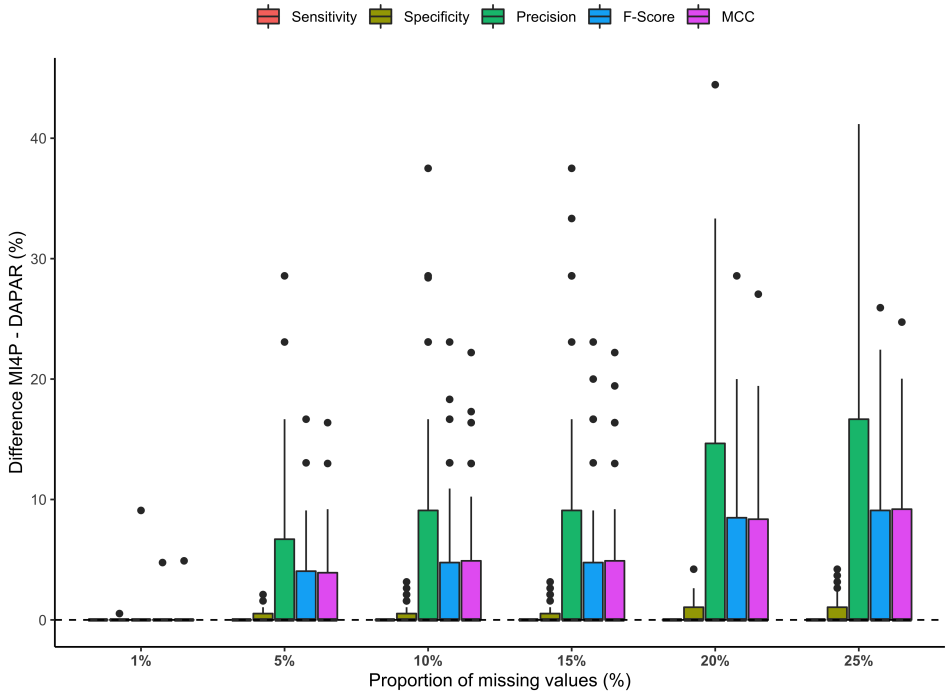


Figure 3.7: Distributions of differences in sensitivity, specificity, precision, F-score and Matthews correlation coefficient for the first MAR set of simulations.

methods show equivalent performance for a small proportion of missing values (1%), where

the imputation process induces little variability. Above 5% missing values, we observe that precision, F-Score and Matthews correlation coefficient are increasingly improved with the `mi4p` workflow. Moreover, let us highlight that sensitivity remains at 100% and specificity is slightly improved, regardless of the missing value proportion.

Compared to the first one, the second and the third sets of MAR simulations illustrate a case where the distributions of intensity values within each experimental condition for differentially expressed analytes are closer. Indeed, recall that the intensity values for these analytes were approximately drawn from a $\mathcal{N}(1.5, 0.5)$ distribution for the first condition and a $\mathcal{N}(3, 0.5)$ distribution for the second one. Figure 3.8 summarises the evolution of

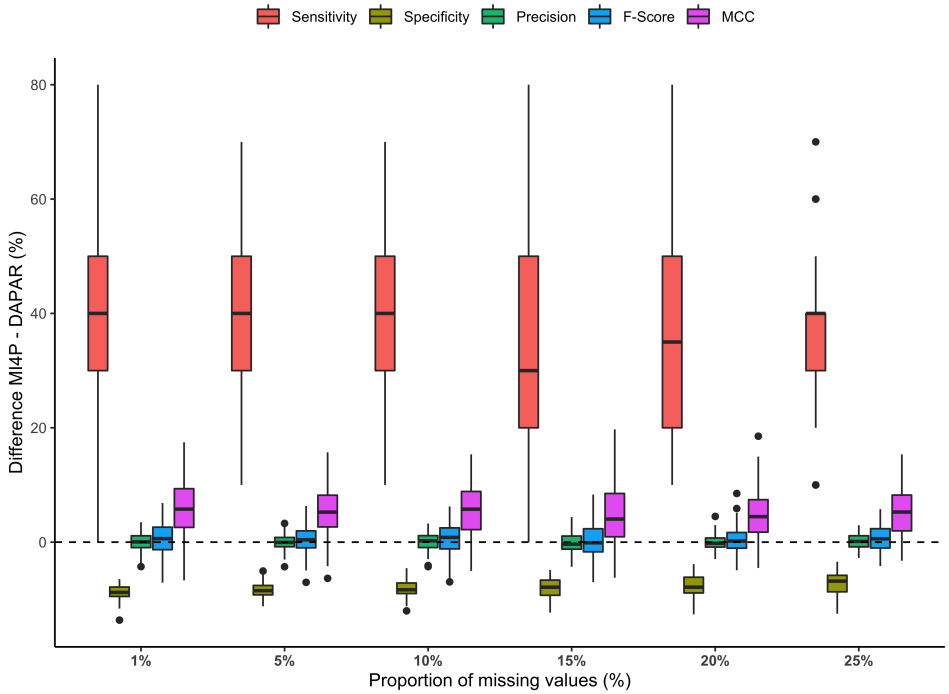


Figure 3.8: Distributions of differences in sensitivity, specificity, precision, F-score and Matthews correlation coefficient for the second MAR set of simulations.

the distribution of differences in sensitivity, specificity, precision, F-score and Matthews correlation coefficient between `mi4p` and `DAPAR` depending on the proportion of missing values in the second set of MAR simulations. Detailed results can be found in Table A.7 in the Appendix chapter. For all proportions of missing values, we observe a trade-off between sensitivity and specificity. Indeed, a slight loss in specificity (yet remaining above 99%) provides a greater gain in terms of sensitivity. Precision performance remains equivalent in both methods. The mean of F-scores and Matthews correlation coefficients across the 100 datasets are also increased with the `mi4p` workflow compared to the `DAPAR` one, suggesting a global improvement of the testing procedure's accuracy.

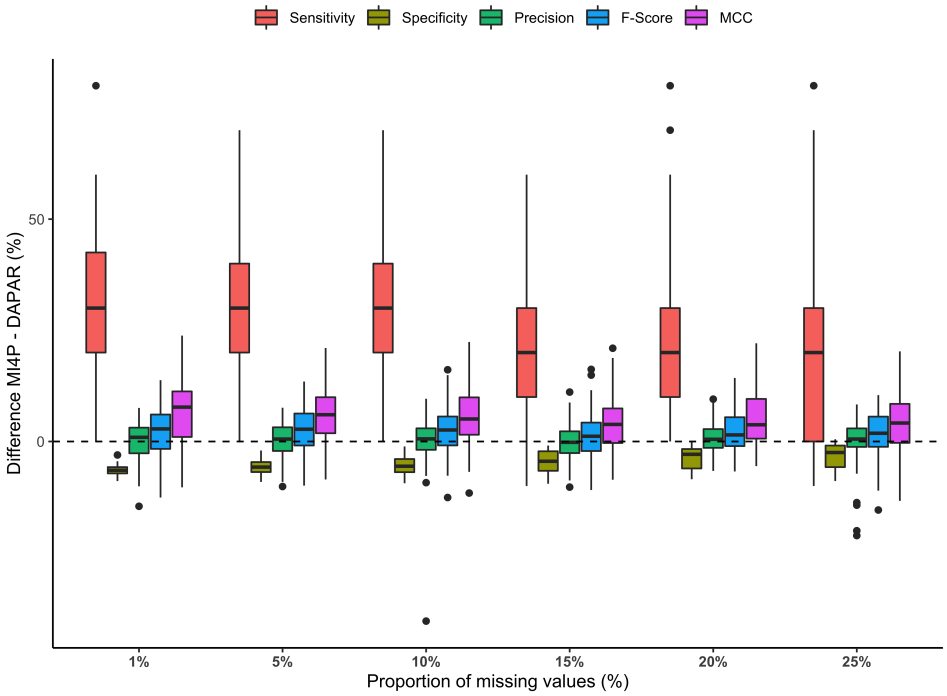


Figure 3.9: Distributions of differences in sensitivity, specificity, precision, F-score and Matthews correlation coefficient for the third MAR set of simulations.

The third set of MAR simulations extend the second one from fixed to random effects. The difference in performance indicators represented on Figure 3.9 remains equivalent to the one observed in the previous set of simulations. However, the detailed results described in Table A.12 suggest that both `mi4p` and `DAPAR` methods underperform on data simulated based on random effects simulated data compared to the fixed effect simulation design. Furthermore, let us point out that the linear model on which both methods rely (see Equation (3.1)) is not designed to account for random effects and thus struggles to capture such a source of variability. Therefore, we notice an overall underperformance of both `mi4p` and `DAPAR` methods in the third set of MAR simulations (Table A.12) compared to the second one (Table A.7).

3.3.2 Under Missing Completely At Random and Not At Random assumption

3.3.2.a Simulation designs

The previous results were provided using only missing at random data. This section extends the simulation study to a mixture of missing completely at random and missing not at random data. As highlighted in Section 1.2.2, this is the most widely encountered missingness mechanism in quantitative proteomics data. The data used in this section were simulated

following an experimental design adapted from Giai Gianetto et al. (2020) and implemented in the `imp4p` `R` package through the `sim.data` function (Giai Gianetto, 2021).

The first set of simulations was based on the following experimental design. Two experimental conditions with ten biological samples each were considered, for which the log-intensities of 1000 analytes were simulated. Among them, 200 were set to be differentially expressed. Hence, the 200 differentially expressed analytes have log-intensities drawn from a Gaussian distribution with a mean of 12.5 in the first condition and 25 in the second one. The remaining simulated log-intensities of non differentially expressed analytes are drawn for both conditions from a Gaussian distribution with a mean of 12.5. The standard deviation in each condition for all analytes is set to 2. Other parameters to be passed as arguments in the `sim.data` function were set to default values.

The second set of simulations considered extends the first one by increasing the number of simulated analytes to 10,000, among which 500 are differentially expressed. Note that in this design, the proportion of differentially expressed analytes is decreased from 20% to 5%. For both simulation studies, six datasets were built with 1%, 5%, 10%, 20% and 25% missing values.

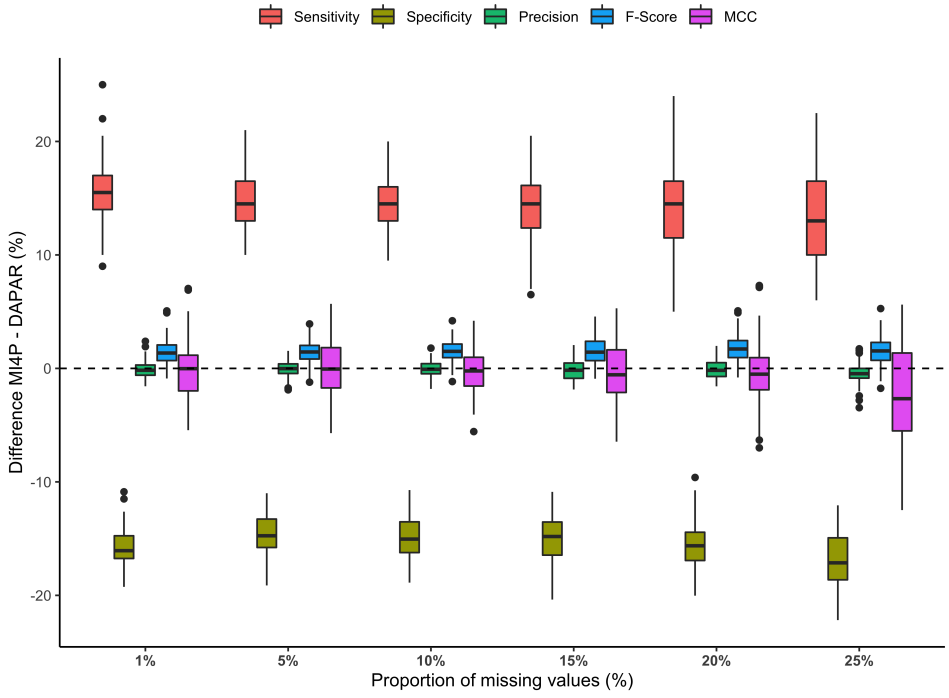


Figure 3.10: Distributions of differences in sensitivity, specificity, precision, F-score and Matthews correlation coefficient for the first MCAR + MNAR set of simulations.

3.3.2.b Results and discussion

The distributions of the difference of the previously described indicators of performance between the `mi4p` and the `DAPAR` workflows for the first set of simulations are showed on Figure 3.10. A trade-off between sensitivity and specificity can be observed: sensitivity is

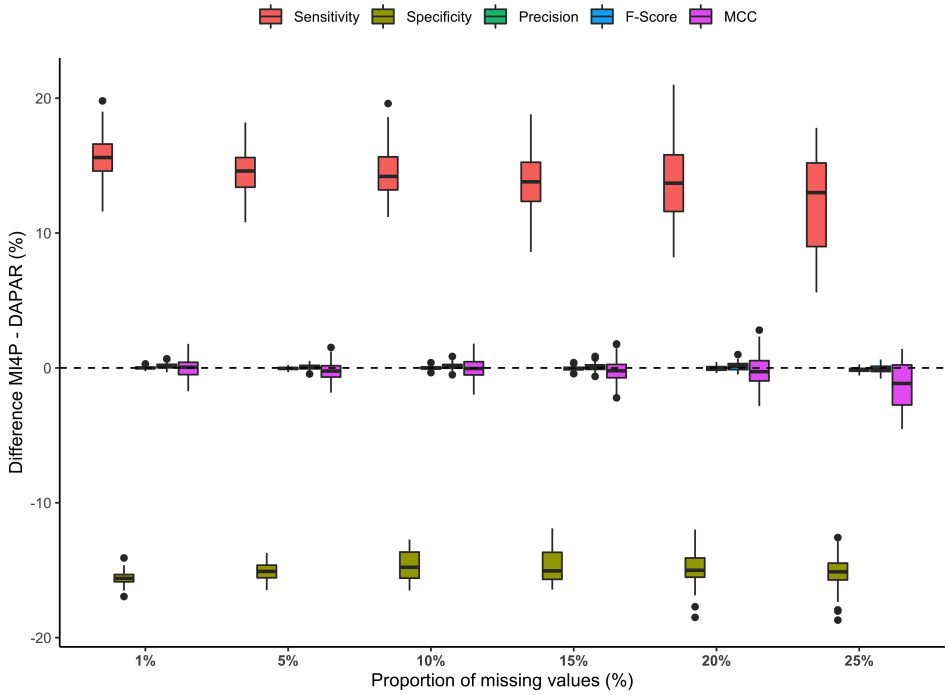


Figure 3.11: Distributions of differences in sensitivity, specificity, precision, F-score and Matthews correlation coefficient for the second MCAR + MNAR set of simulations.

increased by 15% in average while specificity is decreased by 15% in average for the `mi4p` workflow compared to the `DAPAR` one. Furthermore, performance in terms of precision are equivalent for both methods. As far as global performances are concerned, the F-Score is slightly increased by an average of 2% and the MCC is quite stable, with a slight decrease observed for the data with the highest missing values proportion. Figure 3.11 depicts the distributions of the difference of the previously described indicators of performance between the `mi4p` and the `DAPAR` workflows for the second set of simulation. The dispersions of the distributions are globally reduced, but the same trends as in the first set of simulations can be observed. Detailed results for both sets of simulations can be found in Table A.17 and Table A.18. Overall performance in terms of sensitivity, specificity, and precision are quite low for both `mi4p` and `DAPAR` methods, mainly due to the large number of false positives. In particular, precision performance drops when the number of analytes considered is increased. Moreover, the poor performance in terms of MCC suggests that both methods behave almost

as random guess classifier. Hence, the relevance of the chosen imputation method should be questioned in this framework.

3.4 Experiments on real datasets

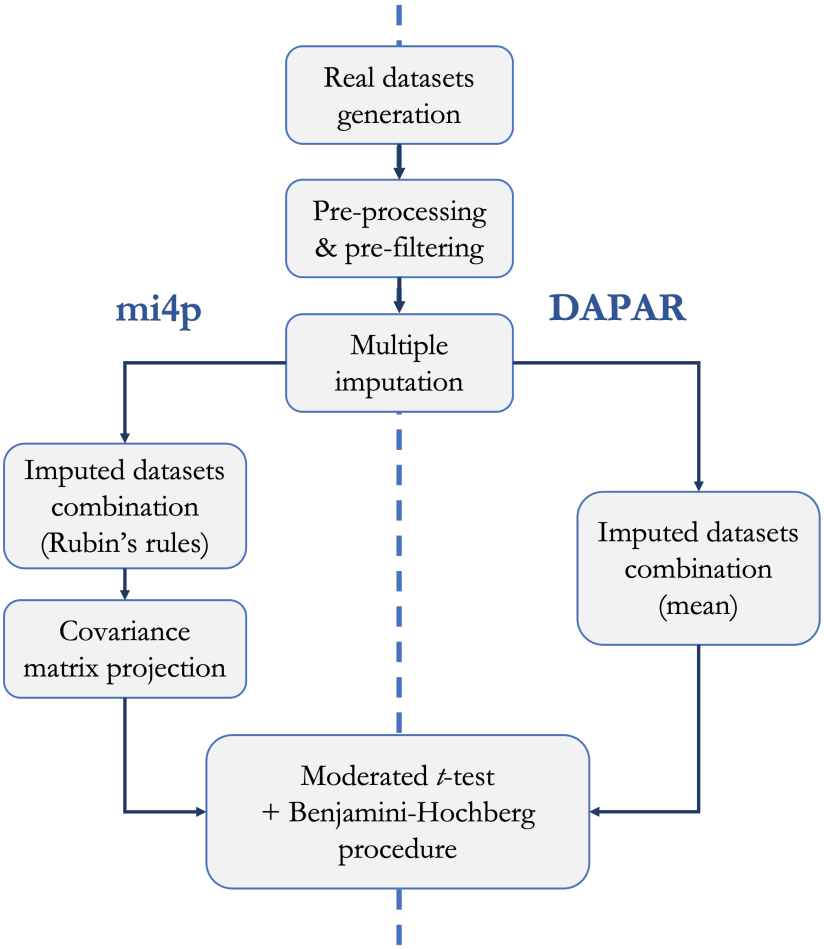


Figure 3.12: Workflow of the study on real datasets conducted for performance evaluation of the `mi4p` methodology and comparison to the one implemented in the `DAPAR` `R` package.

3.4.1 Real datasets generation

3.4.1.a Complex total cell lysates spiked UPS1 standard protein mixtures

We consider a first real dataset from Muller et al. (2016). The experiment involved six peptide mixtures, composed of a constant yeast (*Saccharomyces cerevisiae*) background, into which increasing amounts of UPS1 standard proteins (48 recombinant human proteins,

Merck) were spiked at 0.5, 1, 2.5, 5, 10 and 25 fmol, respectively. In a second well-calibrated dataset, yeast was replaced by a more complex total lysate of *Arabidopsis thaliana* in which UPS1 was spiked in 7 different amounts, namely 0.05, 0.25, 0.5, 1.25, 2.5, 5 and 10 fmol. For each mixture, technical triplicates were constituted. The *Saccharomyces cerevisiae* dataset was acquired on a nanoLC-MS/MS coupling composed of a nanoAcquity UPLC device (Waters) coupled to a Q-Exactive Plus mass spectrometer (Thermo Scientific, Bremen, Germany) as extensively described in Muller et al. (2016). The *Arabidopsis thaliana* dataset was acquired on a nanoLC-MS/MS coupling composed of nanoAcquity UPLC device (Waters) coupled to a Q-Exactive HF-X mass spectrometer (Thermo Scientific, Bremen, Germany) as described hereafter.

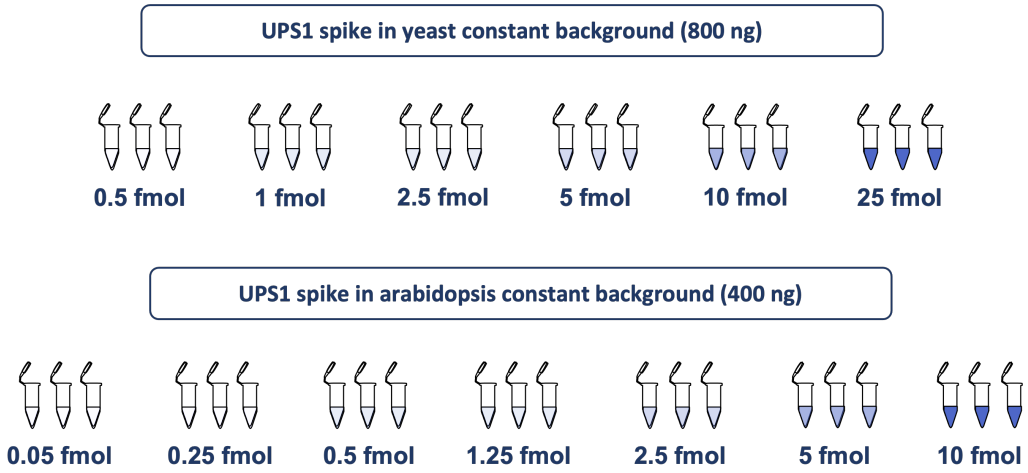


Figure 3.13: Illustration of the spike experiments considered for generating real datasets.

3.4.1.b Data preprocessing

For the *Saccharomyces cerevisiae* and *Arabidopsis thaliana* datasets, Maxquant software was used to identify peptides and derive extracted ion chromatograms. Peaks were assigned with the Andromeda search engine with full trypsin specificity. The database used for the searches was concatenated in house with the *Saccharomyces cerevisiae* entries extracted from the UniProtKB-SwissProt database (16 April 2015, 7806 entries) or the *Arabidopsis thaliana* entries (09 April 2019, 15 818 entries) and those of the UPS1 proteins (48 entries). The minimum peptide length required was seven amino acids and a maximum of one missed cleavage was allowed. Default mass tolerances parameters were used. The maximum false discovery rate was 1% at peptide and protein levels with the use of a decoy strategy. For the *Arabidopsis thaliana* + UPS1 experiment, data were extracted both with and without Match Between Runs and 2 pre-filtering criteria were applied prior to statistical analysis: only peptides with at least 1 out of 3 quantified values in each condition on one hand and

2 out of 3 on the other hand were kept. Thus, 4 datasets derived from the *Arabidopsis thaliana* + UPS1 were considered. For the *Saccharomyces cerevisiae* + UPS1 experiment, the same filtering criteria were applied, but only on data extracted with Match Between Runs, leading to 2 datasets considered.

An additional normalisation step was performed on each dataset considered. Normalising peptides' or proteins' intensities aims at reducing batch effects, sample-level variations and therefore better comparing intensities across studied biological samples Wang et al. (2021). In this work, we chose to perform quantile normalisation (as described by Bolstad et al. (2003)), using the `normalize.quantiles` function from the `preprocessCore R` package (Bolstad, 2021).

3.4.1.c Supplemental methods for Arabidopsis thaliana dataset

Peptide separation was performed on an ACQUITY UPLC BEH130 C18 column (250 mm × 75 µm with 1.7 µm diameter particles) and a Symmetry C18 precolumn (20 mm ×180 µm with 5 µm diameter particles; Waters). The solvent system consisted of 0.1% FA in water (solvent A) and 0.1% FA in ACN (solvent B). The samples were loaded into the enrichment column over 3 min at 5 µL/min with 99% of solvent A and 1% of solvent B. The peptides were eluted at 400 nL/min with the following gradient of solvent B: from 3 to 20% over 63 min, 20 to 40% over 19 min, and 40 to 90% over 1 min. The MS capillary voltage was set to 2kV at 250 °C. The system was operated in a data-dependent acquisition mode with automatic switching between MS (mass range 375–1500 m/z with R = 120 000, automatic gain control fixed at 3 × 106 ions, and a maximum injection time set at 60 ms) and MS/MS (mass range 200–2000 m/z with R = 15 000, automatic gain control fixed at 1 × 105, and the maximal injection time set to 60 ms) modes. The twenty most abundant peptides were selected on each MS spectrum for further isolation and higher energy collision dissociation fragmentation, excluding unassigned and monocharged ions. The dynamic exclusion time was set to 40s.

3.4.2 Evaluation of the methodology

3.4.2.a Indicators of performance

We compared our methodology to the one implemented in the `DAPAR` `R` package (Wieczorek et al., 2017) on the real datasets previously described. To assess the performances of both methods, we used the same indicators as the ones used with simulated datasets and described in Section 3.3.1.c. Here, we consider as actual positives UPS1 peptides or proteins and actual negatives *Saccharomyces cerevisiae* and *Arabidopsis thaliana* peptides or proteins.

3.4.2.b Results on real datasets

The trade-off suggested by the simulation study is confirmed by the results obtained on the real datasets. In the *Saccharomyces cerevisiae* + UPS1 experiment, a decrease of 70% in the number of false positives is observed, improving the specificity and precision (Table A.25 in the Appendix chapter). However, this costs in the number of true positives (see Table 3.5), thus decreasing the sensitivity.

Condition vs. 25fmol	True positives	False positives	Sensitivity	Specificity	F-Score
0.5fmol	-2.7%	-67.2%	-2.7%	+1.6%	+53.6%
1fmol	-1.6%	-71.1%	-0.5%	+0.9%	+37.8%
2.5fmol	-3.2%	-75.8%	-3.3%	+0.7%	+26.9%
5fmol	-14.3%	-78.7%	-14.3%	+0.5%	+11.4%
10fmol	-41.9%	-75.2%	-41.9%	+0.5%	-14.4%

Table 3.5: Performance of the `mi4p` methodology expressed in percentage with respect to `DAPAR` workflow, on *Saccharomyces cerevisiae* + UPS1 experiment, with Match Between Runs and at least 1 out of 3 quantified values in each condition. Missing values (6%) were imputed using the maximum likelihood estimation method.

Condition vs. 10fmol	True positives	False positives	Sensitivity	Specificity	F-Score
0.05fmol	-2.3%	-43%	-2.3%	+15%	+62.7%
0.25fmol	-1.5%	-43%	-1.4%	+13.9%	+65.3%
0.5fmol	-1.5%	-50.6%	-1.4%	+10.8%	+81.4%
1.25fmol	-2.3%	-62.6%	-2.3%	+10.9%	+119.8%
2.5fmol	-25.6%	-69.3%	-25.5%	+2.4%	+45.9%
5fmol	-30.3%	-65.2%	-30.4%	+5.5%	+56.1%

Table 3.6: Performance of the `mi4p` methodology expressed in percentage with respect to `DAPAR` workflow, on *Arabidopsis thaliana* + UPS1 experiment, with at least 1 out of 3 quantified values in each condition. Missing values (6%) were imputed using the maximum likelihood estimation method.

The same trend is observed in the *Arabidopsis thaliana* + UPS1 experiment; the number of false positives is decreased by 50% (see Table 3.6 and Table A.19 in the Appendix chapter), thus improving specificity and precision at the cost of sensitivity. The loss in sensitivity

is larger in the highest points of the range in both experiments. The structure of the calibrated datasets used here can explain these observations. Indeed, the quantitative dataset considered takes into account all samples from all conditions, while the testing procedure focuses on one-vs-one comparisons. Two issues can be raised:

- The data preprocessing step can lead to more data filtering than necessary. For instance, we chose to use the filtering criterion such that rows with at least one quantified value in each condition were kept. The more conditions are considered, the more stringent the rule is, possibly leading to a poorer dataset (with fewer observations) for the conditions of interest.

- The imputation process is done on the whole dataset, as well as the estimation step. Then, while projecting the variance-covariance matrix, the estimated variance (later used in the test statistic) is the same for all comparisons. Thus, if one is interested in comparing conditions with fewer missing values, the variance estimator will be penalised by the presence of conditions with more missing values in the initial dataset.

This phenomenon is illustrated in Table A.20, where solely the two highest points of the range have been compared, only using the quantitative data from those two conditions. More peptides have been taken into account for the statistical analysis. This strategy leads to overall better scores for precision, F-score and Matthews correlation coefficient compared to the previous framework.

As far as data extracted without the Match Between Runs algorithm are concerned, the results were equivalent in both methods considered in the *Arabidopsis thaliana* + UPS1 experiment (as illustrated in Tables A.22 and A.23 in the Appendix chapter). Furthermore, the same observations can be drawn from datasets filtered with the criterion of a minimum of 2 out of 3 observed values in each group for the *Arabidopsis thaliana* + UPS1 experiment (Tables A.21 and A.23 in the Appendix chapter) as well as for the *Saccharomyces cerevisiae* + UPS1 experiment (Table A.26 in the Appendix chapter). These observations translate a loss of global information in the dataset, as filtering criteria lead to fewer peptides considered with fewer missing values per peptide.

The `mi4p` methodology also provides better results at the protein-level (after aggregation) in terms of specificity, precision, F-score and Matthews correlation coefficient, with a minor loss in sensitivity (Table A.27 in the Appendix chapter). In particular, a decrease of 63.2% to 80% in the number of false positives is observed with a lower loss on the number of true positives and on sensitivity (up to 2.6%) for the *Saccharomyces cerevisiae* + UPS1 experiment, as illustrated in Table 3.7. As far as the *Arabidopsis thaliana* + UPS1 experiment is concerned, the same trend is observed (Table S8.22). Indeed, the number of false positives is decreased by 31% to 66.8%, with a maximum loss in the number of true positives of 9.8%, as illustrated in Table 3.8.

Condition vs. 25fmol	True positives	False positives	Sensitivity	Specificity	F-Score
0.5fmol	0%	-73.3%	0%	+2.9%	+61.1%
1fmol	-2.4%	-80%	-2.4%	+2.3%	+51.4%
2.5fmol	0%	-70.4%	0%	+0.8%	+20.9%
5fmol	-2.4%	-63.2%	-2.4%	+0.5%	+11.6%
10fmol	-2.6%	-69.6%	-2.6%	+0.7%	+16.5%

Table 3.7: Performance of the `mi4p` methodology (with the aggregation step) expressed in percentage with respect to `DAPAR` workflow, on *Saccharomyces cerevisiae* + UPS1 experiment, with at least 1 out of 3 quantified values in each condition. Missing values were imputed using the Maximum Likelihood Estimation method.

Condition vs. 10fmol	True positives	False positives	Sensitivity	Specificity	F-Score
0.05fmol	0%	-27.6%	0%	+18.3%	+34.2%
0.25fmol	0%	-25.7%	0%	+18.1%	+31%
0.5fmol	0%	-31%	0%	+15.2%	+39.5%
1.25fmol	0%	-65.3%	0%	+12.1	+119.2%
2.5fmol	-2.4%	-66.8%	-2.4%	+5.8%	+88.3%
5fmol	-9.8%	-57.3%	-9.8%	+12.9%	+78.9%

Table 3.8: Performance of the `mi4p` methodology (with the aggregation step) expressed in percentage with respect to `DAPAR` workflow, on *Arabidopsis thaliana* + UPS1 experiment, with at least 1 out of 3 quantified values in each condition. Missing values were imputed using the Maximum Likelihood Estimation method.

3.5 Conclusion and perspectives

In this chapter, we presented as a key step of a workflow a rigorous multiple imputation method by combining the imputed datasets using Rubin's rules. We thus obtained for each analyte on the one hand a combined estimator of the vector of interest parameters, and on the other hand, an estimator of its corresponding variance-covariance matrix. Hence, both within- and between-imputation variabilities are accounted for. The variance-covariance matrix was projected in order to get a univariate parameter of variance for each analyte. We then considered this variability downstream of the statistical analysis by including it in the well-known moderated t-test statistic. In addition, we provided insights on the comparison of imputation methods as well as on the benchmark of several projections methods.

Our methodology was implemented in a publicly available `R` package named `mi4p` (presented in Chapter 4). Its performance was compared on both simulated and real datasets to the `DAPAR` state-of-the-art methodology, using confusion matrix-based indicators. The results showed a trade-off between those indicators. In real datasets, the methodology reduces the number of false positives in exchange for a minor reduction of the number of true positives. The results are similar among all imputation methods considered, especially when the proportion of missing values is small. Our methodology with an additional aggregation step provides better results with a minor loss in sensitivity and can be of interest for proteomicists who will benefit from results at the protein level while using peptide-level

quantification data.

Simulation studies pointed out more false positives in datasets with MNAR and MCAR values than with MAR values. While the considered datasets were simulated differently, this observation requires further investigation, in particular on the imputation method used. Giai Gianetto et al. (2020) proposed an imputation strategy that combines MCAR-devoted and MNAR-devoted imputation algorithms. Likewise, Gardner and Freitas (2021) recently showed that combinatorial MAR/MNAR approaches perform most accurately and reproducibly on bottom-up proteomics data regardless of the missing value type (except for high MNAR proportions). Moreover, several methodological questions are being considered as perspective works. First, the consequences of the multiple imputation process on the number of degrees of freedom in the hierarchical Bayesian model and the moderated t-statistic should be addressed. Furthermore, one can be interested in evaluating the methodology's performance when the moderation and the projections steps are switched. This approach would first lead to a Bayesian moderated variance-covariance matrix estimator, which would be projected afterwards. Then, a further development would remove the projection step and preserve the information provided by the moderated variance-covariance matrix estimator to conduct a multivariate test for equality of means, using a Hotelling's T^2-distribution. Finally, instead of deriving the posterior mean of the variance or the variance-covariance matrix from the Bayesian hierarchical model, it could be interesting to keep all the information provided by the distribution itself and derive information both on location and dispersion. This problem is addressed in Chapter 5.

4

Multiple imputation for proteomics: a tutorial with mi4p R package

The `R` package `mi4p` contains method for analysing multiple imputed quantitative proteomics data (Chion et al., 2021b). This chapter displays the functionalities of the `mi4p` package, namely a rigorous multiple imputation method and a multiple testing framework which accounts for the imputation-induced variability. While the functions implemented in the `mi4p` package correspond to the methods described in Chapter 3, this chapter can be read and used independently. Furthermore, an example use case is provided on the *Arabidopsis thaliana* + UPS1 experiment recalled from Chapter 3, which data are publicly available on ProteomeXchange under the dataset identifier PXD027800.

This chapter is an extension of a book chapter, to be published in *Methods in Molecular Biology* (Springer) (Chion et al., 2022).

4.1 Materials

4.1.1 Requirements

The workflow presented in this protocol is implemented under the `R` package `mi4p`. To use it, the `R` environment is required (Team, 2021b). For a better user experience, the `R Studio` integrated development environment is recommended (Team, 2021a).

4.1.2 Data format - Quantitative data

The quantitative data should be provided as a data frame or a matrix. Rows should describe the peptides and columns the biological samples. Thus, each cell of the matrix contains the measured (or missing) abundance of the peptide in the considered sample. Although statistical analysis at the peptide-level is recommended, the methodology described in this chapter can be used at protein-level. A schematic view of the quantitative dataset is pictured in Figure 4.1.

4.1.3 Data format - Experimental data

The experimental data should be provided as a two-columns data frame or matrix. The first column should contain the names of the biological samples and should be named `Sample.Name`, the second column should contain the names of the corresponding considered condition and should be named `Condition`.

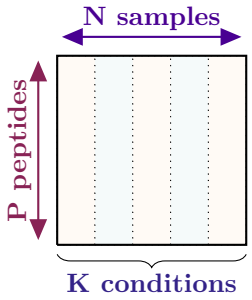


Figure 4.1: Schematic representation of a quantitative dataset to be provided in the `mi4p` package. There should be P rows corresponding to P peptides and N columns corresponding to N samples, which are spread between K conditions.

4.1.4 Data format - Imputed data

The multiple imputed data should be provided as an array of as many matrices as the $drawD$ draws used for multiple imputation. Each imputed matrix should be of the same size as the quantitative data. A schematic view of imputed data is pictured in Figure 4.2.

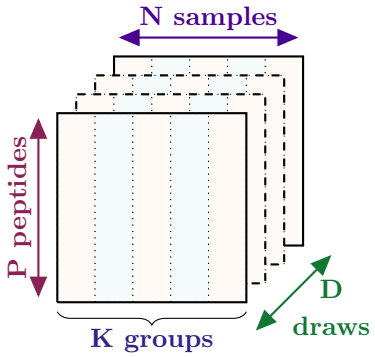


Figure 4.2: Schematic representation of the imputed datasets to be provided in the `mi4p` package. An array of D matrices corresponding to the D draws in the multiple imputation algorithm should be yielded. Each matrix should have P rows corresponding to P peptides and N columns corresponding to N samples, which are spread between K conditions.

4.1.5 Package install and loading

The `mi4p` package can be installed and loaded directly from the CRAN as follows:

```
install.packages("mi4p")
library(mi4p)
```

The development version of the `mi4p` package can be installed from GitHub as follows:

```
install.packages("devtools")
devtools::install_github("mariechion/mi4p")
library(mi4p)
```

4.2 Methods

4.2.1 Multiple imputation

Multiple imputation consists of imputing D times the missing values in the original quantitative dataset. This results in D imputed datasets. Multiple imputation is provided in `mi4p`, using the `multi.impute` function:

```
multi.impute(data, metadata, imp.meth, nb.imp)
```

The `data` argument refers to the original quantitative dataset that contains missing values. The `metadata` argument refers to the experimental dataset. The `imp.meth` argument denotes the chosen multiple imputation algorithm. While the only suggested algorithms for multiple imputation are taken from the `imp4p` (Giai Gianetto, 2021), `mice` (van Buuren et al., 2021) and `impute` (Hastie et al., 2021) packages, the user can choose any other algorithm and recall the imputed matrices in the next steps, under the aforementioned constraints in Section 4.1.4. The `nb.imp` argument describes the number of draws to be done. By default, it is equal to the percentage of missing values in the original quantitative dataset. The `multi.impute` function returns an array of as many imputed matrices as `nb.imp`.

4.2.2 Estimation

The objective of multiple imputation is to estimate from D drawn datasets the vector of parameters of interest and its variance-covariance matrix. Notably, accounting for multiple-imputation-based variability is possible thanks to Rubin's rules, which provide an accurate estimation of these parameters. In `mi4p`, the vectors of parameters of interest are the vectors of the peptides' intensity mean in each condition considered. There are as many vectors to be estimated (and as many corresponding variance-covariances matrices) as the number of peptides in the quantitative dataset.

1. The first Rubin's rule leads to a combined estimator of the vector of intensity means. Let $\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}}$ be the estimated vector of parameters for peptide p in the d-th imputed dataset. The first Rubin's rule gives the combined estimator for peptide p through the D imputed datasets such as:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{p}} = \frac{1}{D}\sum_{d=1}^{D}\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}} \tag{4.1}$$

. To compute the estimators for all peptides in the quantification dataset, the `rubin1.all` function should be used:

```
rubin1.all(imp.data, metadata, funcmean)
```

The `imp.data` argument refers to the array of imputed matrices and the `metadata` argument to the experimental dataset. The `funcmean` argument specifies the method for mean estimation. Here the default `funcmean` function is `meanImp_emmeans` and relies on the estimated marginal means algorithm. The `meanImp_emmeans` function computes the estimated marginal means for specified factors or factor combinations in a linear model for a given imputed dataset. Estimated marginal means are also known as least-squares means or predicted marginal means and are predictions from a linear model over a reference grid. The `rubin1.all` function returns a list of estimated vector of intensity means in each condition for all peptides in the quantitative dataset (*i.e.* the length of the returned list equals the number of rows of `imp.data`). To return only the combined estimator for a specific peptide, the `rubin1.one` function should be used:

```
rubin1.one(peptide, imp.data, metadata, funcmean)
```

The `peptide` argument denotes the row index of the considered peptide in the quantitative dataset.

2. The second Rubin's rule gives the combined estimator of the variance-covariance matrix for each estimated vector of parameters of interest for peptide p through the D imputed datasets such as:

$$\hat{\boldsymbol{\Sigma}}_p = \frac{1}{D}\sum_{d=1}^{D} W_d + \frac{D+1}{D(D-1)}\sum_{d=1}^{D}(\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}})^T(\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{p}}) \tag{4.2}$$

where W_d denotes the variance-covariance matrix of $\hat{\boldsymbol{\beta}}_{\boldsymbol{p},\boldsymbol{d}}$, *i.e.* the variability of the vector of parameters of interest as estimated in the d-th imputed dataset. The idea behind this rule is to decompose the variability into two components: the within-imputation variability and the between-imputation variability. To compute the estimators for all peptides in the quantification dataset, the `rubin2.all` function should be used:

```
rubin2.all(imp.data, metadata, funcmean, funcvar)
```

The `imp.data` argument refers to the array of imputed matrices and the `metadata` argument to the experimental dataset. The `funcmean` and `funcvar` arguments specifies the method for mean and variance-covariance estimation respectively. Here the default function for the `funcmean` argument is `meanImp_emmeans` and the default function for

the `funcvar` function is `within_variance_comp_emmeans`. Both functions rely on the estimated marginal means algorithm (Lenth et al., 2021). To return the within-imputation component only (respectively the between-imputation component) for all peptides, the `rubin2wt.all` function (respectively the `rubin2bt.all` function) should be used:

```
rubin2wt.all(imp.data, metadata, funcvar)
rubin2bt.all(imp.data, metadata, funcmean)
```

The `rubin2.all`,`rubin2wt.all` and `rubin2bt.all` functions return lists of square matrices. The length of the list equals to the number of peptides considered, *i.e.* to the number of rows in `imp.data`. The size of the matrices is equal to the number of conditions considered, *i.e* to the number of levels of the `Condition` factor in the `metadata` dataset. To return only the combined estimator for a specific peptide, the `rubin2.one` function should be used:

```
rubin2.one(peptide, imp.data, metadata, funcmean, funcvar)
```

The `peptide` argument denotes the row index of the considered peptide in the quantitative dataset. Likewise, to return the within-imputation component and/or the between-imputation component for a specific peptide, the `rubin2wt.all` and `rubin2bt.all` functions should be used:

```
rubin2wt.one(peptide, imp.data, metadata, funcvar)
rubin2bt.one(peptide, imp.data, metadata, funcmean)
```

The `rubin2.one`, `rubin2wt.one` and `rubin2bt.one` functions return a square matrix. The size of the matrix is equal to the number of conditions considered, *i.e* to the number of levels of the "Condition" factor in the `metadata` dataset.

4.2.3 Projection

State-of-the-art tests, including Student's t-test, Welch's t-test and moderated t-test, rely on the variance estimation. Here, the variability induced by multiple imputation is described by a variance-covariance matrix. Therefore, a projection step is required to get a univariate variance parameter. As described in Section 3.2.3, projection is performed using the following formula:

$$\hat{\sigma}_p^2 = \max_k \left(\hat{\Sigma}_{p,(k,k)} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{X} \right) \tag{4.3}$$

where $\hat{\Sigma}_{p,(k,k)}$ is the k-th diagonal element of the matrix $\hat{\boldsymbol{\Sigma}}_p$ and $\boldsymbol{X}$ is the design matrix. This step is performed under the `mi4p` package using the `proj_matrix` function:

```
proj_matrix(VarRubin.mat, metadata)
```

The `VarRubin.mat` denotes a variance-covariance matrix, as computed with `rubin2.one`, or a list of variance-covariance matrices, as computed with `rubin2.all` (Section 4.2.2). The `metadata` argument refers to the experimental dataset. The `proj_matrix` function returns either a variance estimator for a given peptide, or a list of variance estimators for all the peptides considered.

Note that to keep all the pieces of information contained in the variance-covariance matrix, an extended version of the workflow presented in this chapter to the multivariate case is currently being implemented in `mi4p`. This multivariate extension will make it possible to fully take into account the effect of the imputation process, and thus the presence of missing values, on the precision of the estimate.

4.2.4 Moderated t-test

Several testing methods can be used. For gene expression data, the recommended method is moderated t-testing (Smyth, 2004). As described in Section 3.2.4, in `mi4p`, the projected variance from multiple imputation is computed in the test statistic associated to the null hypothesis $\mathcal{H}_0 : \beta_{pj} = 0$, by replacing the usual moderated variance estimator by $\hat{\sigma}^2_{p[\text{mod}]}$(see Equation 4.4). Therefore, the results of this testing procedure account both for the specific structure of the data and the uncertainty caused by the multiple imputation step.

$$T_{pj[\text{mod}]} = \frac{\hat{\beta}_{pj}}{\hat{\sigma}^2_{p[\text{mod}]}\sqrt{(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}_{j,j}}} \tag{4.4}$$

with $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}_{j,j}$ the j-th diagonal element in the matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}^{-1}$. Under the null hypothesis $\mathcal{H}_0$, $T_{pj[\text{mod}]}$ follows a Student distribution with $d_p + d_0$ degrees of freedom.

This step is performed under the `mi4p` package using the `mi4limma` function:

```
mi4limma(imp.data, metadata, VarRubin.S2)
```

The `imp.data` argument refers to the array of imputed datasets. The `metadata` refers to the experimental dataset. The `VarRubin.S2` corresponds to the list of projected variance estimator for each peptide, as computed with the `proj_matrix` function (Section 4.2.3). The `mi4limma` function returns a list of p-values and a list of log-transformed fold change for all peptides.

4.2.5 Complete workflow

As an alternative to the step-by-step workflow described above, the complete `mi4p` workflow can be run with a single command:

```
mi4p.uni(data, metadata, imp.meth)
```

The `data` argument refers to the quantitative dataset, the `metadata` argument refers to the experimental dataset and the `imp.meth` argument specifies the imputation method to be used (Section 4.2.1). The `mi4p.uni` function returns a list of p-values and a list of log-transformed fold change for all peptides.
The `mi4p.uni` function includes the four steps described above: multiple imputation (Section 4.2.1), estimation (Section 4.2.2), projection (Section 4.2.3) and moderated t-testing (Section 4.2.4). A synoptic view of the functions which can be used in each step is provided in Table 4.1.

<table>
<tr><th></th><th>Imputation</th><th>Estimation</th><th>Projection</th><th>Test</th></tr>
<tr><td rowspan="2">For one specific peptide</td><td></td><td>rubin1.one
rubin2.one
rubin2wt.one
rubin2bt.one</td><td rowspan="2">proj_matrix</td><td></td></tr>
<tr><td rowspan="2">For all peptides</td><td>multi.impute</td><td>rubin1.all
rubin2.all
rubin2wt.all
rubin2bt.all</td><td>proj_matrix</td><td>mi4limma</td></tr>
<tr><td colspan="4">mi4p.uni</td></tr>
</table>

Table 4.1: Overview of the functions included in `mi4p` package

4.3 Example use case: the Arabidopsis thaliana + UPS1 experiment

Set the random number generator to 17 to strictly get the results provided in Section 3.4.2.b. Otherwise, this step is optional.

```
set.seed(17)
```

4.3.1 Data loading and preprocessing

First we import the MaxQuant peptide-level quantification dataset as well as the corresponding experimental dataset.

```
# Import the experimental dataset
MetadataMQ <- read.delim("metadataMQ.txt")
# Prints the first lines of the MetadataMQ dataset
head(MetadataMQ)

##          Sample.name Condition Bio.Rep
## 1 Intensity.Point1_1    Point1       1
## 2 Intensity.Point1_2    Point1       2
## 3 Intensity.Point1_3    Point1       3
## 4 Intensity.Point2_1    Point2       4
## 5 Intensity.Point2_2    Point2       5
## 6 Intensity.Point2_3    Point2       6

# Import the MaxQuant peptide-level quantification dataset
peptidesMQ <- read.delim("peptides.txt")
```

The quantification dataset requires several preprocessing steps such as replacing 0 intensity values by NA, applying filtering criteria and log-transforming intensity values.

```
# Return indices of the "Intensity." columns, which corresponds
# the intensities measured in each biological sample.
col.ind <- grep(pattern = "Intensity.", x = colnames(peptidesMQ))
# Replace 0 intensity values by NA.
peptidesMQ[,col.ind][peptidesMQ[,col.ind]==0] = NA
# Keep peptides with at least 1 quantified value out of 3 in
# each condition, i.e. maximum 2 NA values out of 3.
peptidesMQ<-peptidesMQ[which(
  apply(is.na(peptidesMQ[,grep(pattern = "Intensity.Point1",
                               x = colnames(peptidesMQ))]),1,sum)<3 &
```

```
  apply(is.na(peptidesMQ[,grep(pattern = "Intensity.Point2",
                               x = colnames(peptidesMQ))]),1,sum)<3 &
  apply(is.na(peptidesMQ[,grep(pattern = "Intensity.Point3",
                               x = colnames(peptidesMQ))]),1,sum)<3 &
  apply(is.na(peptidesMQ[,grep(pattern = "Intensity.Point4",
                               x = colnames(peptidesMQ))]),1,sum)<3 &
  apply(is.na(peptidesMQ[,grep(pattern = "Intensity.Point5",
                               x = colnames(peptidesMQ))]),1,sum)<3 &
  apply(is.na(peptidesMQ[,grep(pattern = "Intensity.Point6",
                               x = colnames(peptidesMQ))]),1,sum)<3 &
  apply(is.na(peptidesMQ[,grep(pattern = "Intensity.Point7",
                               x = colnames(peptidesMQ))]),1,sum)<3),]
```

Reverse amino acid sequences and contaminants can be accessed in a MaxQuant quantification dataset respectively through the columns named `Reverse` and the `Potential.contaminant`. Where appropriate, these variables have `"+"` values.

```
# Remove reverse amino acid sequences and contaminants.
peptidesMQ <- subset(x = peptidesMQ,
                     subset = peptidesMQ$Reverse!="+" &
                              peptidesMQ$Potential.contaminant!="+")
# log2-transformation of intensity values
peptidesMQ[,col.ind] <- log2(peptidesMQ[,col.ind])
```

Note that the quantitative dataset, as described in Section 4.1.2, is accessed by subsetting the MaxQuant quantification dataset to the columns which names begin by `Intensity.`: Hence, using the `col.ind` indices obtained above, we get:

```
data.pept <- peptidesMQ[,col.ind]
# Prints the first 5 rows and the 3 first columns
# of the quantitative dataset
data.pept[1:5,1:3]

##    Intensity.Point1_1 Intensity.Point1_2 Intensity.Point1_3
## 2            21.12698           18.80395           21.94449
## 3            26.91696           27.20387           27.34054
## 4            19.49009           20.12073           19.62763
## 5            23.63145           23.74458           23.95651
## 11           26.40344           26.14703           26.21642

# Prints the first 5 rows and the 4th to 6th columns
```

```
# of the quantitative dataset
data.pept[1:5,4:6]

##    Intensity.Point2_1 Intensity.Point2_2 Intensity.Point2_3
## 2            21.66453           20.15091           20.79329
## 3            27.12504           27.21225           27.36144
## 4            19.49082                 NA           19.71084
## 5            23.74571           23.57115           23.77364
## 11           26.20903           26.19075           26.40440
```

The proportion of missing values can then be evaluated as follows:

```
sum(is.na(data.pept))/prod(dim(data.pept))*100

## [1] 6.342999
```

4.3.2 Intensity normalisation

Normalisation of intensity values is performed here using the quantile normalisation method implemented in the `normalize.quantiles` function of the `preprocessCore` package by Bolstad (2021).

```
# Performs normalisation of quantitative data
data.norm <- normalize.quantiles(x = as.matrix(data.pept))
# Prints the first 5 rows and the 3 first columns
# of the normalised quantitative dataset
data.norm[1:5,1:3]

##          [,1]     [,2]     [,3]
## [1,] 21.53369 19.45501 22.18413
## [2,] 27.17581 27.39488 27.49992
## [3,] 20.15151 20.61713 19.85821
## [4,] 23.91997 23.97905 24.17425
## [5,] 26.66667 26.36947 26.39388

# Prints the first 5 rows and the 4th to 6th columns
# of the normalised quantitative dataset
data.norm[1:5,4:6]

##          [,1]     [,2]     [,3]
## [1,] 21.90344 20.56434 20.91377
```

```
## [2,] 27.24751 27.42404 27.43740
## [3,] 19.82967       NA 19.86756
## [4,] 23.93839 23.85008 23.91765
## [5,] 26.36962 26.42745 26.49293
```

4.3.3 Multiple imputation

Multiple imputation is performed using the `multi.impute` function of the `mi4p` package. Note that by setting the `nb.imp` argument to `NULL`, the number of draws for multiple imputation will be set to the ceiling value of the proportion of missing values in the dataset to be imputed.

```
# Performs multiple imputation with maximum
# likelihood estimation method
data.imp <- multi.impute(data = data.norm,
                         conditions = MetadataMQ$Condition,
                         nb.imp = NULL,
                         method = "MLE")
# Prints the structure of the imputed data
str(data.imp)

##  num [1:14321, 1:21, 1:7] 21.5 27.2 20.2 23.9 26.7 ...
```

The `multi.impute` function returns an array of imputed datasets, as described in Section 4.1.4.

4.3.4 Variance-covariance matrices estimation

The estimation of the variance-covariance matrices as described by the second Rubin's rule (Section 4.2.2) is performed using the `rubin2.all` function of the `mi4p` package.

```
# Computes the variance-covariance matrices estimation.
VarRubin.mat <- rubin2.all(data = data.imp,
                           metacond = MetadataMQ$Condition)
```

The `rubin2.all` function returns a list of as many covariance matrices as peptides considered in the quantitative dataset. The dimension of the covariance matrices is the number of conditions to be compared, *i.e.* the number of different conditions given in the experimental dataset.

```
# Returns the type of VarRubin.mat.
typeof(VarRubin.mat)

## [1] "list"

# Returns the length of VarRubin.mat.
length(VarRubin.mat)

## [1] 14321

# Returns the structure of the first element in VarRubin.mat.
str(VarRubin.mat[[1]])

##  num [1:7, 1:7] 1.67e-01 1.11e-16 -1.79e-16 7.78e-16 -2.78e-17 ...
```

4.3.5 Variance-covariance matrices projection

The projection of each estimated variance-covariance matrix as described in Section 4.2.3 is performed using the `proj_matrix` function of the `mi4p` package.

```
# Computes the variance-covariance matrices projection.
VarRubin.S2 <- proj_matrix(VarRubin.matrix = VarRubin.mat,
                           metadata = MetadataMQ)
```

The `proj_matrix` function produces a numeric vector of as many variance estimation as peptides considered in the quantitative dataset.

```
# Returns the structure of VarRubin.S2.
str(VarRubin.S2)

##  num [1:14321] 0.6657 0.0345 0.462 0.0588 0.0255 ...
```

4.3.6 Moderated t-testing

The moderated t-testing procedure as described in Section 4.2.4 is performed using the `mi4limma` function of the `mi4p` package.

```
# Computes the moderated $t$-testing procedure.
res.mi4limma <- mi4limma(qData = apply(data.imp,1:2,mean),
                         sTab = MetadataMQ,
                         VarRubin = sqrt(VarRubin.S2))
```

The `mi4limma` function returns a list of 2 elements. The first one contains a list of all peptides' logFC for all considered comparisons. The second one contains a list of all peptides p-values for all considered comparisons.

```
# Returns the structure of res.mi4limma.
str(res.mi4limma)

## List of 2
##  $ logFC  :'data.frame': 14321 obs. of  21 variables:
##   ..$ Point1_vs_Point2_logFC: num [1:14321] -0.0696 -0.0128  ...
##   ..$ Point1_vs_Point3_logFC: num [1:14321] -1.12363 -0.28671  ...
##   ..$ Point1_vs_Point4_logFC: num [1:14321] -0.6875 -0.6834  ...
##   ..$ Point1_vs_Point5_logFC: num [1:14321] 0.1934 -0.4671 ...
##   ..$ Point1_vs_Point6_logFC: num [1:14321] -0.986 -1.228  ...
##   ..$ Point1_vs_Point7_logFC: num [1:14321] 1.011 -0.924 ...
##   ..$ Point2_vs_Point3_logFC: num [1:14321] -1.0541 -0.2739  ...
##  [...]
##  $ P_Value:'data.frame': 14321 obs. of  21 variables:
##   ..$ Point1_vs_Point2_pval: num [1:14321] 0.912 0.938 ...
##   ..$ Point1_vs_Point3_pval: num [1:14321] 0.09 0.096 ...
##   ..$ Point1_vs_Point4_pval: num [1:14321] 0.28608 0.00064  ...
##   ..$ Point1_vs_Point5_pval: num [1:14321] 0.7603 0.0108  ...
##   ..$ Point1_vs_Point6_pval: num [1:14321] 1.33e-01 1.02e-06 ...
##   ..$ Point1_vs_Point7_pval: num [1:14321] 1.24e-01 3.13e-05  ...
##   ..$ Point2_vs_Point3_pval: num [1:14321] 0.11 0.11  ...
##   [...]
```

Note that these p-values can be seen as "raw" p-values. Indeed, in this particular multiple testing framework, they need to be ajusted, as described in Section 3.2.4. This adjusting procedure is here performed using the `adjust.p` of the `cp4p` package by Giai Gianetto et al. (2016). For example, adjusted p-values for the comparison between the `"Point1"` condition and the `"Point7"` condition are obtained as follows:

```
mi4limma.1vs7.raw <- res.mi4limma$P_Value$Point1_vs_Point7_pval
mi4limma.1vs7.adj <- adjust.p(p = mi4limma.1vs7.raw,
                              alpha = 0.01)$adjp$adjusted.p

## Procedure of Benjamini-Hochberg is used. pi0 is fixed to 1.

# Returns the structure of mi4limma.1vs7.adj.
str(mi4limma.1vs7.adj)

##  num [1:14321] 0.31517 0.00118 0.48271 0.12873 0.60317 ...
```

5

A Bayesian framework for differential proteomics analysis

In the state-of-the-art approach of Smyth (2004), as well as in our methodology described in Chapter 3, a hierarchical model is used to deduce the posterior distribution of the variance estimator for each analyte. The expectation of this distribution is then used as a moderated estimation of variance and is injected directly in the expression of the t-statistic. However, instead of relying simply on the moderated estimates, it could make sense to take advantage from a fully Bayesian approach.

The topic of missing data has been under investigation for a long time in the Bayesian community, in particular in simple cases involving conjugate priors (Dominici et al., 2000). Despite such theoretical advances, practitionners in proteomics often still rely on old fashioned tools, like t-tests, for conducting most of the differential analyses. Recently, some authors provided convenient approaches and associated implementations (Kruschke, 2013) for handling differential analysis problems with Bayesian inference. For instance, the `R` package `BEST` (standing for Bayesian Estimation Supersedes T-test) has widely contributed to the diffusion of those practices. The present chapter follows a similar idea, by taking advantage of standard results from Bayesian inference with conjugate priors in hierarchical models, to derive a methodology that is tailored to handle our multiple imputation context. Furthermore, we also aim at tackling the more general problem of multivariate differential analysis, to account for possible correlations between analytes.

By defining a hierarchical model with prior distributions both on mean and variance parameters, we aim at providing an adequate quantification of the uncertainty for differential analysis. Inference is thus performed by computing the posterior distribution for the difference of mean peptide intensity between two experimental conditions. In contrast to more flexible models that can be achieved with hierarchical structures, our choice of conjugate priors maintains analytical expressions for direcly sampling from posterior distributions without needing MCMC methods, resulting in a fast inference procedure in practice.

Section 5.1 presents well-known results about Bayesian inference for Gaussian-inverse-gamma conjugated priors. Following analogous results for the multivariate case, Section 5.2 introduces a general Bayesian framework for evaluating mean differences in our differential proteomics context. Section 5.3 provides insights on the particular case where the considered analytes are uncorrelated. Finally, Section 5.4 illustrates hands-on examples on a real proteomics dataset and highlights the benefits of such a multivariate Bayesian framework for practitioners.

5.1 Background: Bayesian inference for Gaussian-inverse-gamma conjugated priors

Before deriving our complete workflow, let us first recall some classical results from Bayesian inference that will further serve our aim. The purpose of this section is twofold. By first fully detailing proofs of results in the univariate case that are often admitted, we pave the way to the development of our subsequent contribution in a multivariate framework.

Let us assume a generative model such as:

$$y = \mu + \varepsilon,$$

where:

- $\mu \mid \sigma^2 \sim \mathcal{N}\left(\mu_0, \frac{1}{\lambda_0}\sigma^2\right)$ is the prior distribution over the mean,
- $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is the error term,
- $\sigma^2 \sim \Gamma^{-1}(\alpha_0, \beta_0)$ is the prior distribution over the variance,

with $\{\mu_0, \lambda_0, \alpha_0, \beta_0\}$ an arbitrary set of prior hyper-parameters. We provide in Figure 5.1 an illustration of the hypotheses taken over such hierarchical generative model. From the

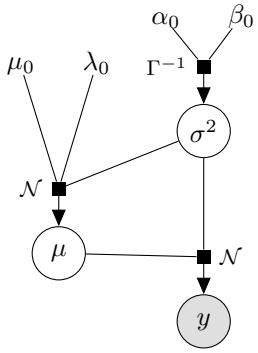


Figure 5.1: Graphical model of the hierarchical structure when assuming a Gaussian-inverse-gamma prior, conjugated with a Gaussian likelihood with unknown mean and variance.

previous hypotheses, we can deduce the likelihood of the model for a sample of observations $\boldsymbol{y} = \{y_1, \ldots, y_N\}$:

$$\begin{aligned} p(\boldsymbol{y} \mid \mu, \sigma^2) &= \prod_{n=1}^{N} p(y_n \mid \mu, \sigma^2) \\ &= \prod_{n=1}^{N} \mathcal{N}\left(y_n;\ \mu, \sigma^2\right), \end{aligned}$$

Let us recall that such assumptions consists in defining a prior Gaussian-inverse-gamma distribution, which is conjugated with the Gaussian distribution with unknown mean μ and variance σ^2. The probability density function (PDF) of such a prior distribution can be written as:

$$p(\mu, \sigma^2 \mid \mu_0, \lambda_0, \alpha_0, \beta_0) = \frac{\sqrt{\lambda_0}}{\sqrt{2\pi}} \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \left(\frac{1}{\sigma^2}\right)^{\alpha_0 + \frac{3}{2}} \exp\left(-\frac{2\beta_0 + \lambda_0(\mu - \mu_0)^2}{2\sigma^2}\right).$$

In this particular case, it is a well-known result that the inference is tractable and the posterior distribution remains a Gaussian-inverse-gamma (Murphy, 2007). Let us recall below the complete development of this derivation by identification of the analytical form

(we ignore conditioning over the hyper-parameters for convenience):

$$
\begin{aligned}
p(\mu, \sigma^2 \mid \boldsymbol{y}) &\propto p(\boldsymbol{y} \mid \mu, \sigma^2) \times p(\mu, \sigma^2) \\
&= \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mu)^2\right) \\
&\quad \times \frac{\sqrt{\lambda_0}}{\sqrt{2\pi}}\frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\left(\frac{1}{\sigma^2}\right)^{\alpha_0+\frac{3}{2}} \exp\left(-\frac{2\beta_0 + \lambda_0(\mu-\mu_0)^2}{2\sigma^2}\right) \\
&\propto \left(\frac{1}{\sigma^2}\right)^{\alpha_0+\frac{N+3}{2}} \exp\left(\underbrace{-\frac{2\beta_0 + \lambda_0(\mu-\mu_0)^2 + \sum_{n=1}^{N}(y_n-\mu)^2}{2\sigma^2}}_{\mathcal{A}}\right).
\end{aligned}
$$

Let us introduce Lemma 5.1 below to decompose the term $\mathcal{A}$ as desired:

Lemma 5.1. *Assume a set* $\boldsymbol{x_1}, \ldots, \boldsymbol{x_N} \in \mathbb{R}^q$*, and note* $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x_n}$ *the associated average vector. For any* $\boldsymbol{\mu} \in \mathbb{R}^q$*:*

$$
\sum_{n=1}^{N}(\boldsymbol{x_n} - \boldsymbol{\mu})(\boldsymbol{x_n} - \boldsymbol{\mu})^\intercal = N(\bar{\boldsymbol{x}} - \boldsymbol{\mu})(\bar{\boldsymbol{x}} - \boldsymbol{\mu})^\intercal + \sum_{n=1}^{N}(\boldsymbol{x_n} - \bar{\boldsymbol{x}})(\boldsymbol{x_n} - \bar{\boldsymbol{x}})^\intercal.
$$

Proof.

$$
\begin{aligned}
\sum_{n=1}^{N}(\boldsymbol{x_n} - \boldsymbol{\mu})(\boldsymbol{x_n} - \boldsymbol{\mu})^\intercal &= \sum_{n=1}^{N}\boldsymbol{x_n}\boldsymbol{x_n}^\intercal + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal - 2\boldsymbol{x_n}\boldsymbol{\mu}^\intercal \\
&= N\boldsymbol{\mu}\boldsymbol{\mu}^\intercal - 2N\bar{\boldsymbol{x}}\boldsymbol{\mu}^\intercal + \sum_{n=1}^{N}\boldsymbol{x_n}\boldsymbol{x_n}^\intercal \\
&= N\boldsymbol{\mu}\boldsymbol{\mu}^\intercal + N\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal + N\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal - 2N\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal - 2N\bar{\boldsymbol{x}}\boldsymbol{\mu}^\intercal + \sum_{n=1}^{N}\boldsymbol{x_n}\boldsymbol{x_n}^\intercal \\
&= N\left(\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal - \boldsymbol{\mu}\boldsymbol{\mu}^\intercal - 2\bar{\boldsymbol{x}}\boldsymbol{\mu}^\intercal\right) + \sum_{n=1}^{N}\boldsymbol{x_n}\boldsymbol{x_n}^\intercal + \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal - 2\boldsymbol{x_n}\bar{\boldsymbol{x}}^\intercal \\
&= N\left(\bar{\boldsymbol{x}} - \boldsymbol{\mu}\right)\left(\bar{\boldsymbol{x}} - \boldsymbol{\mu}\right)^\intercal + \sum_{n=1}^{N}(\boldsymbol{x_n} - \bar{\boldsymbol{x}})(\boldsymbol{x_n} - \bar{\boldsymbol{x}})^\intercal.
\end{aligned}
$$

□

Applying this result in our context for $q = 1$, we obtain:

$$
\mathcal{A} = -\frac{1}{2\sigma^2}\left(2\beta_0 + \lambda_0(\mu - \mu_0)^2 + N(\bar{y} - \mu)^2 + \sum_{n=1}^{N}(y_n - \bar{y})^2\right)
$$

$$
\begin{aligned}
&= -\frac{1}{2\sigma^2}\left(2\beta_0 + \sum_{n=1}^{N}(y_n-\bar{y})^2 + (\lambda_0+N)\mu^2 - 2\mu(N\bar{y}+\lambda_0\mu_0) + N\bar{y}^2 + \lambda_0\mu_0^2\right)\\
&= -\frac{1}{2\sigma^2}\left(2\beta_0 + \sum_{n=1}^{N}(y_n-\bar{y})^2 + N\bar{y}^2 + \lambda_0\mu_0^2\right.\\
&\quad \left. + (\lambda_0+N)\left[\mu^2 - 2\mu\frac{N\bar{y}+\lambda_0\mu_0}{\lambda_0+N} + \left(\frac{N\bar{y}+\lambda_0\mu_0}{\lambda_0+N}\right)^2 - \left(\frac{N\bar{y}+\lambda_0\mu_0}{\lambda_0+N}\right)^2\right]\right)\\
&= -\frac{1}{2\sigma^2}\left(2\beta_0 + \sum_{n=1}^{N}(y_n-\bar{y})^2 + N\bar{y}^2 + \lambda_0\mu_0^2 - \frac{(N\bar{y}+\lambda_0\mu_0)^2}{\lambda_0+N}\right.\\
&\quad \left. + (\lambda_0+N)\left(\mu - \frac{N\bar{y}+\lambda_0\mu_0}{\lambda_0+N}\right)^2\right)\\
&= -\frac{1}{2\sigma^2}\left(2\beta_0 + \sum_{n=1}^{N}(y_n-\bar{y})^2 + \frac{(\lambda_0+N)(N\bar{y}^2+\lambda_0\mu_0^2) - N^2\bar{y}^2 - \lambda_0^2\mu_0^2 + 2N\bar{y}\lambda_0\mu_0}{\lambda_0+N}\right.\\
&\quad \left. + (\lambda_0+N)\left(\mu - \frac{N\bar{y}+\lambda_0\mu_0}{\lambda_0+N}\right)^2\right)\\
&= -\frac{1}{2\sigma^2}\left(2\beta_0 + \sum_{n=1}^{N}(y_n-\bar{y})^2 + \frac{\lambda_0 N}{\lambda_0+N}(\bar{y}-\mu_0)^2 + (\lambda_0+N)\left(\mu - \frac{N\bar{y}+\lambda_0\mu_0}{\lambda_0+N}\right)^2\right).
\end{aligned}
$$

Therefore, the above expression can be identified as a Gaussian-inverse-gamma PDF by writing:

$$
p(\mu,\sigma^2 \mid \boldsymbol{y}) \propto \left(\frac{1}{\sigma^2}\right)^{\alpha_N+\frac{3}{2}} \exp\left(-\frac{2\beta_N + \lambda_N(\mu-\mu_N)^2}{2\sigma^2}\right), \tag{5.1}
$$

with:

- $\mu_N = \dfrac{N\bar{y}+\lambda_0\mu_0}{\lambda_0+N}$,
- $\lambda_N = \lambda_0 + N$,
- $\alpha_N = \alpha_0 + \dfrac{N}{2}$,
- $\beta_N = \beta_0 + \dfrac{1}{2}\sum\limits_{n=1}^{N}(y_n-\bar{y})^2 + \dfrac{\lambda_0 N}{2(\lambda_0+N)}(\bar{y}-\mu_0)^2$.

The normalising constant is induced by this characteristic formulation and the joint posterior distribution can be expressed as:

$$
\mu,\sigma^2 \mid \boldsymbol{y} \sim \mathcal{N}\Gamma^{-1}\left(\mu_N,\lambda_N,\alpha_N,\beta_N\right) \tag{5.2}
$$

Although these update formulas provide a valuable result in itself, we shall see in the sequel that we are more interested in the marginal distribution over the mean parameter μ, for comparison purposes. Computing this marginal from the joint posterior in Equation (5.2)

remains tractable as well by integrating over σ^2:

$$
\begin{aligned}
p(\mu \mid \boldsymbol{y}) &= \int p(\mu, \sigma^2 \mid \boldsymbol{y}) \, \mathrm{d}\sigma^2 \\
&= \frac{\sqrt{\lambda_N}}{\sqrt{2\pi}} \frac{\beta_N^{\alpha_N}}{\Gamma(\alpha_N)} \int \left(\frac{1}{\sigma^2}\right)^{\alpha_N + \frac{3}{2}} \exp\left(-\frac{2\beta_N + \lambda_N(\mu - \mu_N)^2}{2\sigma^2}\right) \mathrm{d}\sigma^2 \\
&= \frac{\sqrt{\lambda_N}}{\sqrt{2\pi}} \frac{\beta_N^{\alpha_N}}{\Gamma(\alpha_N)} \frac{\Gamma(\alpha_N + \frac{1}{2})}{(\beta_N + \frac{\lambda_N}{2}(\mu - \mu_N)^2)^{\alpha_N + \frac{1}{2}}} \\
&\quad \times \int \underbrace{\frac{(\beta_N + \frac{\lambda_N}{2}(\mu - \mu_N)^2)^{\alpha_N + \frac{1}{2}}}{\Gamma(\alpha_N + \frac{1}{2})} \left(\frac{1}{\sigma^2}\right)^{\alpha_N + \frac{1}{2} + 1} \exp\left(-\frac{\beta_N + \frac{\lambda_N}{2}(\mu - \mu_N)^2}{\sigma^2}\right)}_{\Gamma^{-1}(\alpha_N + \frac{1}{2},\ \beta_N + \frac{\lambda_N}{2}(\mu - \mu_N)^2)} \mathrm{d}\sigma^2 \\
&= \frac{\sqrt{\lambda_N}}{\sqrt{2\pi}} \frac{\beta_N^{\alpha_N}}{\Gamma(\alpha_N)} \frac{\Gamma(\alpha_N + \frac{1}{2})}{(\beta_N + \frac{\lambda_N}{2}(\mu - \mu_N)^2)^{\alpha_N + \frac{1}{2}}} \times 1 \\
&= \frac{\Gamma(\alpha_N + \frac{1}{2})}{\Gamma(\alpha_N)} \frac{\sqrt{\lambda_N}}{\sqrt{2\pi}} \frac{\beta_N^{\alpha_N + \frac{1}{2}}}{\sqrt{\beta_N}} (\beta_N + \frac{\lambda_N}{2}(\mu - \mu_N)^2)^{-\alpha_N - \frac{1}{2}} \\
&= \frac{\Gamma(\alpha_N + \frac{1}{2})}{\Gamma(\alpha_N)} \frac{\sqrt{\lambda_N}}{\sqrt{2\pi}} \frac{\beta_N^{\alpha_N + \frac{1}{2}}}{\sqrt{\beta_N}} \beta_N^{-\alpha_N - \frac{1}{2}} (1 + \frac{\alpha_N \lambda_N}{2\alpha_N \beta_N}(\mu - \mu_N)^2)^{-\alpha_N - \frac{1}{2}} \\
&= \frac{\Gamma(\alpha_N + \frac{1}{2})}{\Gamma(\alpha_N)} \frac{\sqrt{\alpha_N \lambda_N}}{\sqrt{2\alpha_N \pi \beta_N}} (1 + \frac{1}{2\alpha_N} \frac{\alpha_N \lambda_N (\mu - \mu_N)^2}{\beta_N})^{-\alpha_N - \frac{1}{2}} \\
&= \underbrace{\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \frac{1}{\sqrt{\pi \nu \hat{\sigma}^2}} (1 + \frac{1}{\nu} \frac{(\mu - \mu_N)^2}{\hat{\sigma}^2})^{-\frac{\nu+1}{2}}}_{T_\nu(\mu;\ \mu_N, \hat{\sigma}^2)},
\end{aligned}
$$

with:

- $\nu = 2\alpha_N$,
- $\hat{\sigma}^2 = \dfrac{\beta_N}{\alpha_N \lambda_N}$.

The marginal posterior distribution over μ can thus be expressed as a non-standardised Student's t-distribution that we express below in terms of the initial hyper-parameters:

$$
\mu \mid \boldsymbol{y} \sim T_{2\alpha_0 + N}\left(\frac{N\bar{y} + \lambda_0 \mu_0}{\lambda_0 + N}, \frac{\beta_0 + \frac{1}{2}\sum_{n=1}^{N}(y_n - \bar{y})^2 + \frac{\lambda_0 N}{2(\lambda_0 + N)}(\bar{y} - \mu_0)^2}{(\alpha_0 + \frac{N}{2})(\lambda_0 + N)}\right). \tag{5.3}
$$

The derivation of this analytical formula provides a valuable tool for computing straightforward posterior distribution for the mean parameter in such context. We shall see in the next section how to leverage this approach to introduce a novel means' comparison methodology for a more general framework, to handle both multidimensional and missing data.

5.2 General Bayesian framework for evaluating mean differences ...

Recalling our differential proteomics context that consists in assessing the differences in mean intensity values for P peptides or proteins quantified in N samples divided into K conditions. As before, Figure 5.2 illustrates the hierarchical generative structure assumed for each group $k = 1, \dots, K$.

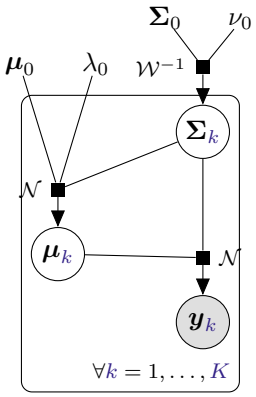


Figure 5.2: Graphical model of the hierarchical structure of the generative model for the vector $\boldsymbol{y}_k$ of peptide intensities in K groups of biological samples, *i.e.* K experimental conditions.

Maintaining the notation analogous to previous ones, the generative model for $\boldsymbol{y}_k \in \mathbb{R}^P$, can be written as:

$$\boldsymbol{y}_k = \boldsymbol{\mu}_k + \boldsymbol{\varepsilon}_k, \ \forall k = 1, \dots, K,$$

where:

- $\boldsymbol{\mu}_k \mid \boldsymbol{\Sigma}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \frac{1}{\lambda_0}\boldsymbol{\Sigma}_k\right)$ is the prior mean intensities vector of the k-th group,
- $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}_k)$ is the error term of the k-th group,
- $\boldsymbol{\Sigma}_k \sim \mathcal{W}^{-1}(\boldsymbol{\Sigma}_0, \nu_0)$ is the prior variance-covariance matrix of the k-th group,

with $\{\boldsymbol{\mu}_0, \lambda_0, \boldsymbol{\Sigma}_0, \nu_0\}$ a set of hyper-parameters that needs to be chosen as modelling hypotheses and $\mathcal{W}^{-1}$ represents the Inverse-Wishart distribution, previously introduced in **??**, and used as the conjugate prior for an unknown covariance matrix of a multivariate Gaussian distribution.

Traditionally, in Bayesian inference, those quantities need to be carefully chosen for the estimation to be as accurate as possible, in particular with low sample sizes. The incorporation of expert or prior knowledge on the model would also come from the adequate setting of these hyper-parameters. However, our final purpose in this chapter is not much about estimating but instead focused on comparing groups' mean (i.e. differential analysis). Interestingly, providing a perfect estimation of the posterior distributions over

$\{\boldsymbol{\mu}_k\}_{k=1,\dots,K}$ does not appear as the main concern here, as the posterior difference of means (i.e. $p(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'} \mid \boldsymbol{y}_k, \boldsymbol{y}_{k'})$) represents the actual quantity of interest. Although providing meaningful prior hyper-parameters leads to adequate uncertainty quantification, we shall, above all, take those quantities equal for all groups. This choice would ensure an unbiased comparison, which would constitute a valuable alternative to the traditional and somehow limited t-tests. Indeed, inference based on hypothesis testing and p-values has been widely called into question over the past decade (Wasserstein et al., 2019). Additionally, t-tests do not provide any insight on effect sizes or uncertainty quantification (in contrast to Bayesian inference as emphasized by Kruschke and Liddell (2018)).

The present framework aspires at estimating a posterior distribution for each mean parameter vector $\boldsymbol{\mu}_k$, starting from the same prior assumptions in each group. The comparison between means of all groups would then only rely on the ability to sample directly from these distributions and compute empirical posteriors for the means' difference. As a bonus, this framework remains compatible with multiple imputations strategies previously introduced to handle missing data that frequently arise in applicative contexts (see Chapter 3).

From the previous hypotheses, we can deduce the likelihood of the model for an i.i.d. sample $\{\boldsymbol{y}_{k,1}, \dots, \boldsymbol{y}_{k,N_k}\}$:

$$\begin{aligned} p(\boldsymbol{y}_{k,1}, \dots, \boldsymbol{y}_{k,N_k} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) &= \prod_{n=1}^{N_k} p(\boldsymbol{y}_{k,n} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \prod_{n=1}^{N_k} \mathcal{N}\left(\boldsymbol{y}_{k,n};\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \end{aligned}$$

However, as previously pointed out, such datasets often contain missing data and we shall introduce here consistent notation. Assume $\mathcal{H}$ to be the set of all observed data, we additionally define:

- $\boldsymbol{y}_k^{(0)} = \{y_{k,n}^p \in \mathcal{H},\ n = 1, \dots N_k,\ p = 1, \dots, P\}$, the set of elements that are observed in the k-th group,
- $\boldsymbol{y}_k^{(1)} = \{y_{k,n}^p \notin \mathcal{H},\ n = 1, \dots N_k,\ p = 1, \dots, P\}$, the set of elements that are missing the k-th group.

Moreover, as we remain in the context of multiple imputation, $\{\tilde{\boldsymbol{y}}_k^{(1),1}, \dots, \tilde{\boldsymbol{y}}_k^{(1),D}\}$ can be defined as the set of D draws of an imputation process applied on missing data in the k-th group. In such context, a closed-form approximation for the multiple-imputed posterior distribution of $\boldsymbol{\mu}_k$ can be derived for each group as stated in Proposition 5.1.

Proposition 5.1. *For all $k = 1, \dots, K$, the posterior distribution of $\boldsymbol{\mu}_k$ can be approximated*

by a mixture of multiple-imputed multivariate t-distributions, such as:

$$p(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}) \simeq \frac{1}{D}\sum_{d=1}^{D} T_{\nu_k}\left(\boldsymbol{\mu}; \tilde{\boldsymbol{\mu}}_k^{(d)}, \tilde{\boldsymbol{\Sigma}}_k^{(d)}\right)$$

with:

- $\nu_k = \nu_0 + N_k - P + 1$,
- $\tilde{\boldsymbol{\mu}}_k^{(d)} = \dfrac{\lambda_0 \boldsymbol{\mu}_0 + N_k \bar{\boldsymbol{y}}_k^{(d)}}{\lambda_0 + N_k}$,
- $\tilde{\boldsymbol{\Sigma}}_k^{(d)} = \dfrac{\boldsymbol{\Sigma}_0 + \sum_{n=1}^{N_k} (\tilde{\boldsymbol{y}}_{k,n}^{(d)} - \bar{\boldsymbol{y}}_k^{(d)})(\tilde{\boldsymbol{y}}_{k,n}^{(d)} - \bar{\boldsymbol{y}}_k^{(d)})^\intercal + \dfrac{\lambda_0 N_k}{(\lambda_0 + N_k)}(\bar{\boldsymbol{y}}_k^{(d)} - \boldsymbol{\mu}_0)(\bar{\boldsymbol{y}}_k^{(d)} - \boldsymbol{\mu}_0)^\intercal}{(\nu_0 + N_k - P + 1)(\lambda_0 + N_k)}$,

where we introduced the shorthand $\tilde{\boldsymbol{y}}_{k,n}^{(d)} = \begin{bmatrix} \boldsymbol{y}_{k,n}^{(0)} \\ \tilde{\boldsymbol{y}}_{k,n}^{(1),d} \end{bmatrix}$ *to represent the* d*-th imputed vector of observed data, and the corresponding average vector* $\bar{\boldsymbol{y}}_k^{(d)} = \dfrac{1}{N_k}\sum_{n=1}^{N_k} \tilde{\boldsymbol{y}}_{k,n}^{(d)}$.

This analytical formulation is particularly convenient for our purpose and, as we shall see in the proof below, merely comes from imputation.

Proof. For the sake of clarity, let us omit the k groups here and first consider a general case with $\boldsymbol{y}_k = \boldsymbol{y} \in \mathbb{R}^P$. Moreover, let us focus on only one imputed dataset, and maintain the notation $\tilde{\boldsymbol{y}}_1^{(d)}, \dots, \tilde{\boldsymbol{y}}_N^{(d)} = \boldsymbol{y}_1, \dots, \boldsymbol{y}_N$ for convenience. From the hypotheses of the model, we can derive $\mathcal{L}$, the posterior log-PDF over $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, following the same idea as for the univariate case presented Section 5.1:

$$\begin{aligned}
\mathcal{L} &= \log p(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_N) \\
&= \log \underbrace{p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_N \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})}_{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} + \log \underbrace{p(\boldsymbol{\mu}, \boldsymbol{\Sigma})}_{\mathcal{NW}^{-1}(\boldsymbol{\mu}_0, \lambda_0, \boldsymbol{\Sigma}_0, \nu_0)} + C_1 \\
&= -\frac{N}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\left(\sum_{n=1}^{N}(\boldsymbol{y}_n - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_n - \boldsymbol{\mu})\right) \\
&\quad - \frac{\nu_0 + P + 2}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\left(\operatorname{tr}\left(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}^{-1}\right) - \frac{\lambda_0}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)\right) + C_2 \\
&= -\frac{1}{2}\Bigg[(\nu_0 + P + 2 + N)\log|\boldsymbol{\Sigma}| + \operatorname{tr}\left(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}^{-1}\right) \\
&\quad + \sum_{n=1}^{N}\operatorname{tr}\left((\boldsymbol{y}_n - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_n - \boldsymbol{\mu})\right) + \operatorname{tr}\left(\lambda_0(\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0)\right)\Bigg] + C_2
\end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{2}\Bigg[(\nu_0 + P + 2 + N)\log|\boldsymbol{\Sigma}| + \operatorname{tr}\Bigg(\boldsymbol{\Sigma}^{-1}\Big\{\boldsymbol{\Sigma}_0 + \lambda_0(\boldsymbol{\mu}-\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\intercal \\
&\quad \underbrace{+ N(\bar{\boldsymbol{y}}-\boldsymbol{\mu})(\bar{\boldsymbol{y}}-\boldsymbol{\mu})^\intercal + \sum_{n=1}^{N}(\boldsymbol{y}_n-\bar{\boldsymbol{y}})(\boldsymbol{y}_n-\bar{\boldsymbol{y}})^\intercal}_{\text{Lemma 1}}\Big\}\Bigg)\Bigg] + C_2 \\
&= -\frac{1}{2}\Bigg[(\nu_0 + P + 2 + N)\log|\boldsymbol{\Sigma}| + \operatorname{tr}\Bigg(\boldsymbol{\Sigma}^{-1}\Big\{\boldsymbol{\Sigma}_0 + \sum_{n=1}^{N}(\boldsymbol{y}_n-\bar{\boldsymbol{y}})(\boldsymbol{y}_n-\bar{\boldsymbol{y}})^\intercal \\
&\quad + (N+\lambda_0)\boldsymbol{\mu}\boldsymbol{\mu}^\intercal - \boldsymbol{\mu}\left(N\bar{\boldsymbol{y}}^\intercal + \lambda_0\boldsymbol{\mu}_0^\intercal\right) - \left(\lambda_0\boldsymbol{\mu}_0 + N\bar{\boldsymbol{y}}\right)\boldsymbol{\mu}^\intercal + \lambda_0\boldsymbol{\mu}_0\boldsymbol{\mu}_0^\intercal + N\bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^\intercal\Big\}\Bigg)\Bigg] + C_2 \\
&= -\frac{1}{2}\Bigg[(\nu_0 + P + 2 + N)\log|\boldsymbol{\Sigma}| \\
&\quad + \operatorname{tr}\Bigg(\boldsymbol{\Sigma}^{-1}\Big\{\boldsymbol{\Sigma}_0 + \sum_{n=1}^{N}(\boldsymbol{y}_n-\bar{\boldsymbol{y}})(\boldsymbol{y}_n-\bar{\boldsymbol{y}})^\intercal + \frac{N\lambda_0}{N+\lambda_0}(\bar{\boldsymbol{y}}-\boldsymbol{\mu}_0)(\bar{\boldsymbol{y}}-\boldsymbol{\mu}_0)^\intercal \\
&\quad + (N+\lambda_0)\left(\boldsymbol{\mu} - \frac{N\bar{\boldsymbol{y}}+\lambda_0\boldsymbol{\mu}_0}{N+\lambda_0}\right)\left(\boldsymbol{\mu} - \frac{N\bar{\boldsymbol{y}}+\lambda_0\boldsymbol{\mu}_0}{N+\lambda_0}\right)^\intercal\Big\}\Bigg)\Bigg] + C_2 \\
&= -\frac{1}{2}\Bigg[(\nu_N + P + 2)\log|\boldsymbol{\Sigma}| + \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_N\right) + \lambda_N\left(\boldsymbol{\mu}-\boldsymbol{\mu}_N\right)^\intercal\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_N\right)\Bigg] + C_2.
\end{aligned}
$$

By identification, we recognise the log-PDF that characterises the Gaussian-inverse-Wishart distribution $\mathcal{NIW}^{-1}(\boldsymbol{\mu}_N, \lambda_N, \boldsymbol{\Sigma}_N, \nu_N)$ with:

- $\boldsymbol{\mu}_N = \dfrac{N\bar{\boldsymbol{y}} + \lambda_0\boldsymbol{\mu}_0}{N+\lambda_0}$,
- $\lambda_N = \lambda_0 + N$,
- $\boldsymbol{\Sigma}_N = \boldsymbol{\Sigma}_0 + \sum_{n=1}^{N}(\boldsymbol{y}_N-\bar{\boldsymbol{y}})(\boldsymbol{y}_N-\bar{\boldsymbol{y}})^\intercal + \dfrac{\lambda_0 N}{(\lambda_0+N)}(\bar{\boldsymbol{y}}-\boldsymbol{\mu}_0)(\bar{\boldsymbol{y}}-\boldsymbol{\mu}_0)^\intercal$,
- $\nu_N = \nu_0 + N$.

Once more, we can integrate over $\boldsymbol{\Sigma}$ to compute the mean's marginal posterior distribution by identifying the PDF of the inverse-Wishart distribution $\mathcal{W}^{-1}\Big(\boldsymbol{\Sigma}_N + \lambda_N\left(\boldsymbol{\mu}-\boldsymbol{\mu}_N\right)\left(\boldsymbol{\mu}-\boldsymbol{\mu}_N\right)^\intercal, \nu_N + 1\Big)$ and by reorganising the terms:

$$
\begin{aligned}
p(\boldsymbol{\mu} \mid \boldsymbol{y}) &= \int p(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \boldsymbol{y})\,\mathrm{d}\boldsymbol{\Sigma} \\
&= \frac{\lambda_N^{\frac{P}{2}}|\boldsymbol{\Sigma}_N|^{\frac{\nu_N}{2}}}{(2\pi)^{\frac{P}{2}}2^{\frac{P\nu_N}{2}}\Gamma_P\left(\frac{\nu_N}{2}\right)} \\
&\quad \times \int |\boldsymbol{\Sigma}|^{-\frac{\nu_N+P+2}{2}}\exp\left(-\frac{1}{2}\left(\operatorname{tr}\left(\boldsymbol{\Sigma}_N\boldsymbol{\Sigma}^{-1}\right) - \frac{\lambda_N}{2}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_N\right)^\intercal\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_N\right)\right)\right)\mathrm{d}\boldsymbol{\Sigma}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\lambda_N^{\frac{P}{2}} |\boldsymbol{\Sigma}_{\boldsymbol{N}}|^{\frac{\nu_N}{2}}}{(2\pi)^{\frac{P}{2}} 2^{\frac{P\nu_N}{2}} \Gamma_P\left(\frac{\nu_N}{2}\right)} \times \frac{2^{\frac{P(\nu_N+1)}{2}} \Gamma_P\left(\frac{\nu_N+1}{2}\right)}{|\boldsymbol{\Sigma}_N + \lambda_N (\boldsymbol{\mu} - \boldsymbol{\mu}_N)(\boldsymbol{\mu} - \boldsymbol{\mu}_N)^\intercal|^{\frac{\nu_N+1}{2}}} \times 1 \\
&= \frac{\pi^{p(p-1)/4} \prod_{p=0}^{P-1} \Gamma\left(\frac{\nu_N+1-p}{2}\right)}{\pi^{P(P-1)/4} \prod_{p=1}^{P} \Gamma\left(\frac{\nu_N+1-p}{2}\right)} \times \frac{\lambda_N^{\frac{P}{2}}}{\pi^{\frac{P}{2}}} \\
&\quad \times \underbrace{\frac{|\boldsymbol{\Sigma}_N|^{\frac{\nu_N}{2}}}{|\boldsymbol{\Sigma}_N|^{\frac{\nu_N+1}{2}}} \times \left(1 + \lambda_N (\boldsymbol{\mu} - \boldsymbol{\mu}_N)^\intercal \boldsymbol{\Sigma}_N^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_N)\right)^{-\frac{\nu_N+1}{2}}}_{\text{Matrix determinant lemma}} \\
&= \frac{\Gamma\left(\frac{\nu_N+1}{2}\right)}{\Gamma\left(\frac{\nu_N+1-P}{2}\right)} \times \frac{[\lambda_N(\nu_N - P + 1)]^{\frac{P}{2}}}{[\pi(\nu_N - P + 1)]^{\frac{P}{2}} |\boldsymbol{\Sigma}_N|^{\frac{1}{2}}} \\
&\quad \times \left(1 + \frac{\lambda_N(\nu_N - P + 1)}{(\nu_N - P + 1)} (\boldsymbol{\mu} - \boldsymbol{\mu}_N)^\intercal \boldsymbol{\Sigma}_N^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_N)\right)^{-\frac{\nu_N+1}{2}} \\
&= \frac{\Gamma\left(\frac{(\nu_N-P+1)+P}{2}\right)}{\Gamma\left(\frac{\nu_N-P+1}{2}\right) [\pi(\nu_N - P + 1)]^{\frac{P}{2}} |\frac{\boldsymbol{\Sigma}_N}{\lambda_N(\nu_N - P + 1)}|^{\frac{1}{2}}} \\
&\quad \times \left(1 + \frac{1}{\nu_N - P + 1} (\boldsymbol{\mu} - \boldsymbol{\mu}_N)^\intercal \left(\frac{\boldsymbol{\Sigma}_N}{\lambda_N(\nu_N - P + 1)}\right)^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_N)\right)^{-\frac{(\nu_N-P+1)+P}{2}}.
\end{aligned}
$$

The above expression corresponds to the PDF of a multivariate t-distribution $\mathcal{T}_\nu\left(\boldsymbol{\mu}_N, \hat{\boldsymbol{\Sigma}}\right)$, with:

- $\nu = \nu_N - P + 1$,
- $\hat{\boldsymbol{\Sigma}} = \dfrac{\boldsymbol{\Sigma}_N}{\lambda_N(\nu_N - P + 1)}$.

Therefore, we demonstrated that for each group and imputed dataset, the complete-data posterior over $\boldsymbol{\mu}_k$ happens to be a multivariate t-distribution. Thus, following Rubin's rules for multiple imputation (see Equation (1.19) in Section 1.2.3.b), we can propose an approximation to the true posterior distribution (that is only conditioned over observed values):

$$
\begin{aligned}
p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}\right) &= \int p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}, \boldsymbol{y}_k^{(1)}\right) p\left(\boldsymbol{y}_k^{(1)} \mid \boldsymbol{y}_k^{(0)}\right) \mathrm{d}\boldsymbol{y}_k^{(1)} \\
&\simeq \frac{1}{P} \sum_{p=1}^{P} p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}, \tilde{\boldsymbol{y}}_k^{(1),d}\right)
\end{aligned}
$$

Leading to the desired results when evaluating the previously derived posterior distribution on each multiple-imputed dataset. □

Thanks to Proposition 5.1, we have an explicit formula for approximating, using multiple

imputed datasets, the posterior distribution of the mean vector for each group. Although such linear combination of multivariate t-distributions is not a known specific distribution in itself, it is now straightforward to generate realisations of samples of the posterior by simply drawing from the D multivariate t-distributions, each being specific to an imputed dataset, and then compute the mean of the D vectors. Therefore, the empirical distribution resulting from a high number of samples generated by this procedure would be easy to visualise and manage for comparison purpose. Generating the empirical distribution of the mean's difference between two groups k and k' then comes directly, by computing the difference between each couple of samples drawn from both posterior distributions $p(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)})$ and $p(\boldsymbol{\mu}'_k \mid \boldsymbol{y}_{k'}^{(0)})$. In Bayesian statistics, relying on empirical distributions drawn from the posterior is common practice in the context of Markov chain Monte Carlo (MCMC) algorithms, but often comes at a high computational cost. In our framework, we managed to maintain the best of both worlds since deriving analytical distributions from model hypotheses offers the benefits of probabilistic inference with adequate uncertainty quantification, while remaining tractable and not relying on MCMC procedures. The computational cost of the method thus roughly remains as low as frequentist counterparts since merely a few updating calculus and drawing from t-distributions are needed.

As usual when it comes to compare the mean between two groups, we still need to assess if the posterior distribution of the difference appear, in a sense, to be sufficiently away from zero. This practical inference choice is not specific to our context and remains highly dependent on the context of the study. Moreover, as the present model is multi-dimensional, we may also raise the question of the metric used to compute the difference between vectors. In a sense, our posterior distribution of the mean's differences offers an elegant solution to the traditional problem of multiple testing often encountered in applied science and allows tailored definitions of what could be called a *meaningful* result (*significant* does not appear anymore as an appropriate term in this more general context). For example, displaying the distribution of the squared difference would penalise large differences in elements of the mean vector whereas absolute difference would give a more balanced conception of the average divergence from one group to the other. Clearly, as any marginal of a multivariate t-distribution remains a (multivariate) t-distribution, it is also straightforward to compare specific elements of the mean vectors merely by restraining to the appropriate dimension. Recalling our proteomics context, this means that we could still compare mean intensity of peptides between groups one peptide at a time, or choosing to compare all peptides at once and thus accounting for possible correlations between peptides in each group. However, an appropriate manner to account for those correlations could be to subset peptides using their protein groups.

Let us provide in Algorithm 1 a summary of the whole procedure for comparing mean vectors of two different experimental conditions in terms of posterior distribution.

Algorithm 1 Posterior distribution of the vector of mean's difference

Initialise the hyper-posteriors $\boldsymbol{\mu}_0^k = \boldsymbol{\mu}_0^{k'}$, $\lambda_0^k = \lambda_0^{k'}$, $\boldsymbol{\Sigma}_0^k = \boldsymbol{\Sigma}_0^{k'}$, $\nu_0^k = \nu_0^{k'}$

for $d = 1, \ldots, D$ **do**

Compute $\{\boldsymbol{\mu}_N^{k,(d)}, \lambda_N^k, \boldsymbol{\Sigma}_N^{k,(d)}, \nu_N^k\}$ and $\{\boldsymbol{\mu}_N^{k',(d)}, \lambda_N^{k'}, \boldsymbol{\Sigma}_N^{k',(d)}, \nu_N^{k'}\}$ from hyper-posteriors and data

Draw R realisations $\hat{\boldsymbol{\mu}}_k^{(d)[r]} \sim T_{\nu_N^k}\left(\boldsymbol{\mu}_N^{k,(d)}, \dfrac{\boldsymbol{\Sigma}_N^{k,(d)}}{\lambda_N^k \nu_N^k}\right)$; $\hat{\boldsymbol{\mu}}_{k'}^{(d)[r]} \sim T_{\nu_N^{k'}}\left(\boldsymbol{\mu}_N^{k',(d)}, \dfrac{\boldsymbol{\Sigma}_N^{k',(d)}}{\lambda_N^{k'} \nu_N^{k'}}\right)$

end for

for $r = 1, \ldots, R$ **do**

Compute $\hat{\boldsymbol{\mu}}_k^{[r]} = \dfrac{1}{D}\sum_{d=1}^{D} \hat{\boldsymbol{\mu}}_k^{(d)[s]}$ and $\hat{\boldsymbol{\mu}}_{k'}^{[r]} = \dfrac{1}{D}\sum_{d=1}^{D} \hat{\boldsymbol{\mu}}_{k'}^{(d)[r]}$ to combine samples

Generate a realisation $\hat{\boldsymbol{\mu}}_\Delta^{[r]} = \hat{\boldsymbol{\mu}}_k^{[r]} - \hat{\boldsymbol{\mu}}_{k'}^{[r]}$ from the difference's distribution

end for

return $\{\hat{\boldsymbol{\mu}}_\Delta^{[1]}, \ldots, \hat{\boldsymbol{\mu}}_\Delta^{[R]}\}$, an R-sample drawn from the posterior distribution of the mean's difference

5.3 The uncorrelated case: no more multiple testing nor imputation

Let us notice that modelling covariances between all variables as in Proposition 5.1 often constitutes a challenge, which is computationally expensive in high dimensions and not always adapted. However, we detailed in Section 5.1 results that are classical in Bayesian inference, but somehow not widespread enough in applied science, especially when it comes to comparing means. In particular, we can leverage these results to adapt Algorithm 1 to the univariate case, for handling the same problem as in Chapter 3 with a more probabilistic flavour. Indeed, when the absence of correlations between peptides is assumed (*i.e.* $\boldsymbol{\Sigma}$ being diagonal), the problem reduces to the analysis of P independent inference problems (as $\boldsymbol{\mu}$ is supposed Gaussian) and the posterior distributions can be derived in closed-form, as we recalled in Equation (5.1). Moreover, let us highlight a nice property coming with this relaxing assumption is that (multiple-)imputation is no longer needed in this context. Using the same notation as before and the uncorrelated assumption (and thus the induced independence between analytes for $p \neq p'$), we can write:

$$p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}\right) = \int p\left(\boldsymbol{\mu}_k, \boldsymbol{y}_k^{(1)} \mid \boldsymbol{y}_k^{(0)}\right) \mathrm{d}\boldsymbol{y}_k^{(1)} \tag{5.4}$$

$$= \int p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}, \boldsymbol{y}_k^{(1)}\right) p\left(\boldsymbol{y}_k^{(1)} \mid \boldsymbol{y}_k^{(0)}\right) \mathrm{d}\boldsymbol{y}_k^{(1)} \tag{5.5}$$

$$= \int \prod_{p=1}^{P} \left\{ p\left(\mu_k^p \mid y_k^{p,(0)}, y_k^{p,(1)}\right) p\left(y_k^{p,(1)} \mid y_k^{p,(0)}\right) \right\} \mathrm{d}\boldsymbol{y}_k^{(1)} \tag{5.6}$$

$$= \prod_{p=1}^{P} \int \left\{ p\left(\mu_k^p \mid y_k^{p,(0)}, y_k^{p,(1)}\right) p\left(y_k^{p,(1)} \mid y_k^{p,(0)}\right) \mathrm{d}y_k^{p,(1)} \right\} \tag{5.7}$$

$$= \prod_{p=1}^{P} p\left(\mu_k^p \mid y_k^{p,(0)}\right) \tag{5.8}$$

$$= \prod_{p=1}^{P} T_{2\alpha_0^p + N_k^p}\left(\mu_k^p;\ \mu_{k,N}^p,\ \hat{\sigma_k^p}^2\right), \tag{5.9}$$

with:

- $\mu_{k,N}^p = \dfrac{N_k^p \bar{y}_k^{p,(0)} + \lambda_0^p \mu_0^p}{\lambda_0^p + N_k^p}$,
- $\hat{\sigma_k^p}^2 = \dfrac{\beta_0^p + \frac{1}{2}\sum_{n=1}^{N_k^p} (y_{k,n}^{p,(0)} - \bar{y}_k^{p,(0)})^2 + \frac{\lambda_0 N_k^p}{2(\lambda_0^p + N_k^p)} (\bar{y}_k^{p,(0)} - \mu_0^p)^2}{(\alpha_0^p + \frac{N_k^p}{2})(\lambda_0^p + N_k^p)}$.

In this context, it can be noticed that $p\left(\boldsymbol{\mu}_k \mid \boldsymbol{y}_k^{(0)}\right)$ factorises naturally over $p = 1, \ldots, P$, and thus only depends upon the data that have actually been observed for each peptide. Indeed, we observe that the integration over the missing data $\boldsymbol{y}_k^{(1)}$ is straightforward in this framework and neither the Rubin's approximation or even imputation (whether multiple or not) appear necessary. The observed data $\boldsymbol{y}_k^{(0)}$ already bear all the useful information as if each unobserved values could simply be ignored without effect on the posterior distribution.

Let us emphasise on the fact that this property of factorisation and tractable integration over missing data comes directly from the covariance structure as a diagonal matrix, and thus only constitutes a particular case, though convenient, of the previous model. However, in the context of differential analysis in proteomics, analysing each peptide as an independent problem is a common practice, as seen in Chapter 3, and we shall notice that the Bayesian framework tackles this issue in an elegant and somehow simpler way. In particular, the classical inference approach based on hypothesis testing performs numerous successive tests for all peptides. Such an approach often leads to the pitfall of multiple testing which needs to be carefully dealt with. Interestingly, we can notice that the above model also avoid multiple testing (as it does not rely on hypothesis testing and the definition of some threshold) while maintaining the convenient interpretations of Bayesian probabilistic inference. To conclude, whereas the analytical derivation of posterior distributions with Gaussian-inverse-gamma constitutes a well-known results, our proposition to define such probabilistic mean's comparison procedure provides, under the standard uncorrelated-peptides assumption, an

elegant and handy alternative to classical techniques that naturally tackles both the imputation and multiple testing issues. Let us provide in Algorithm 2 the pseudo-code of the inference procedure in order to highlight differences with the fully-correlated case:

Algorithm 2 Posterior distribution of the mean's difference

for $p = 1, \ldots, P$ **do**

Initialise the hyper-posteriors $\mu_0^{k,p} = \mu_0^{k',p}$, $\lambda_0^{k,p} = \lambda_0^{k',p}$, $\alpha_0^{k,p} = \alpha_0^{k',p}$, $\beta_0^{k,p} = \beta_0^{k',p}$

Compute $\{\mu_N^{k,p}, \lambda_N^{k,p}, \alpha_N^{k,p}, \beta_N^{k,p}\}$ and $\{\mu_N^{k',p}, \lambda_N^{k',p}, \alpha_N^{k',p}, \beta_N^{k',p}\}$ from hyper-posteriors and data

Draw R realisations $\hat{\mu}_k^{p,[r]} \sim T_{\alpha_N^{k,p}}\left(\mu_N^{k,p}, \frac{\beta_N^{k,p}}{\lambda_N^{k,p}\alpha_N^{k,p}}\right)$, $\hat{\mu}_{k'}^{p,[r]} \sim T_{\alpha_N^{k',p}}\left(\mu_N^{k',p}, \frac{\beta_N^{k',p}}{\lambda_N^{k',p}\alpha_N^{k',p}}\right)$

for $r = 1, \ldots, R$ **do**

Generate a realisation $\hat{\mu}_\Delta^{p,[r]} = \hat{\mu}_k^{p,[r]} - \hat{\mu}_{k'}^{p,[r]}$ from the difference's distribution

end for

end for

return $\{\hat{\boldsymbol{\mu}}_\Delta^{[1]}, \ldots, \hat{\boldsymbol{\mu}}_\Delta^{[R]}\}$, an R-sample drawn from the posterior distribution of the mean's difference

5.4 Experiments

To illustrate our methodology, we used a real proteomics dataset already introduced in Chapter 3, namely the *Arabidopsis thaliana* + UPS dataset, with the Match between Runs algorithm and at least one quantified value in each experimental condition. Briefly, let us recall that UPS proteins were spiked in increasing amounts into a constant background of *Arabidopsis thaliana* (ARATH) protein lysate. Hence, UPS proteins are differentially expressed, and ARATH proteins are not. For illustration purposes, we arbitrarily chose to focus the examples on the `P12081ups|SYHC_HUMAN_UPS` and the `sp|F4I893|ILA_ARATH` proteins. Note that both proteins have nine quantified peptides. Unless otherwise stated, we took the examples of the `AALEELVK` UPS peptide and the `VLPLIIPILSK` ARATH peptide and the following values have been set for the prior hyper-parameters:

- $\mu_0 = 20, \ \forall p = 1, \ldots, P,$
- $\lambda = 1,$
- $\alpha_0 = 1,$
- $\beta_0 = 1,$
- $\boldsymbol{\Sigma}_0 = I_P,$

- $\nu_0 = 10$.

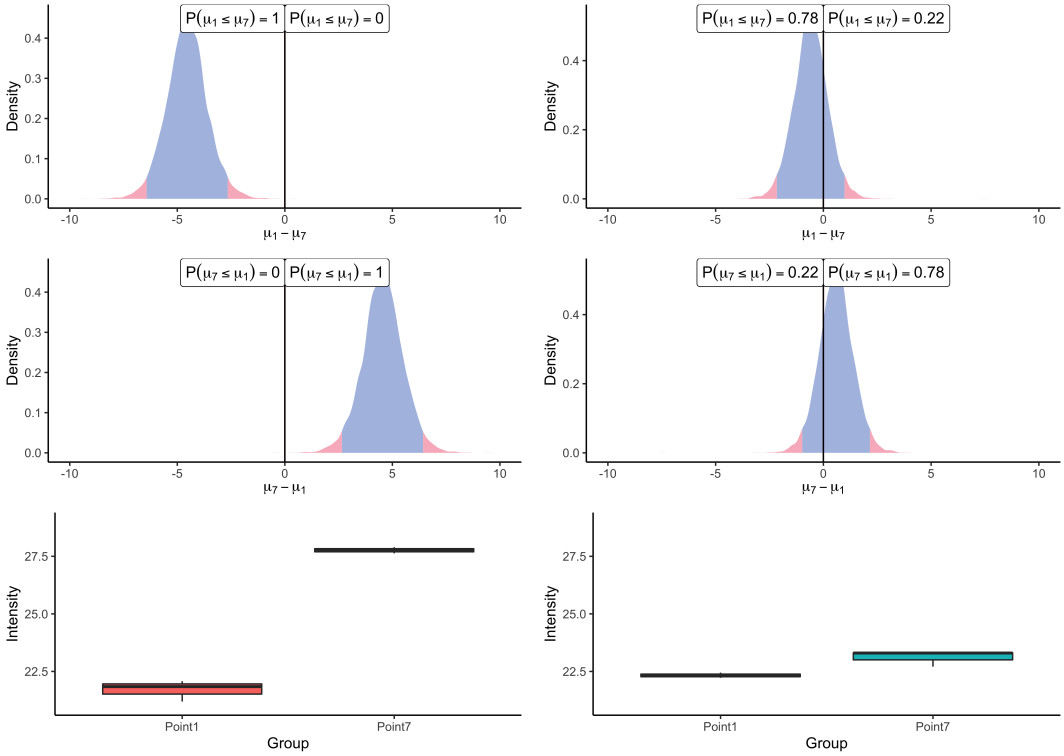
These values correspond to the practical insights acquired from our previous studies, while remaining relatively vague in terms of prior variance. As previously stated, it is essential for these values to be identical in all groups for ensuring a fair and unbiased comparison. In the case where more expert information would be accessible, its incorporation would be possible, for instance, through the definition of a more precise prior mean (μ_0) associated with a more confident prior variance (encoded through α_0 and β_0). Additionally let us recall that in our real datasets, the constants of the values take the values:

- $\forall k = 1, \ldots, K,\ N_k = 3$ data points, in the absence of missing data,
- $P = 9$ peptides, when using the multivariate model,
- $D = 7$ draws of imputation,
- $R = 10^4$ sample points from the posterior distributions.

Let us emphasise that, in this context where the number N_k of observed biological samples is extremely low, in particular when data are missing, we should expect a perceptible influence of the prior hyper-parameters, as well as an inherent uncertainty in the posteriors. However, this influence has been reduced to the minimum in all the subsequent graphs for the sake of clarity and for assuring a good understanding of the underlying properties of the methodology. The high number R of sample points drawn from the posteriors assures the empirical distribution to be smoothly displayed on the graph, but one should note that sampling is really quick in practice, and this number can be easily increased if necessary.

5.4.1 Univariate Bayesian inference for differential analysis

First, let us illustrate the univariate framework described in Section 5.3. In this experience, we compared the intensity means in the lowest (0.05 fmol UPS) and the highest points (10 fmol UPS) of the UPS spike range. Let us recall that our univariate algorithm does not rely on imputation and should be applied directly on raw data. For the sake of illustration, the chosen peptides were observed entirely in all three biological samples of both experimental conditions. Resulting from the application of our univariate algorithm, posterior distributions of the mean difference for both peptides are represented on Figure 5.3. As the analysis consists in a comparison between conditions, the 0 value has been highlighted on the x-axis for assessing both the direction and the magnitude of the difference. The distance to zero of the distributions indicates whether the peptide is differentially expressed or not. In particular, Figure 5.3a shows the posterior distribution of the means difference for the UPS peptide. Its location, far from zero, indicates a high probability (almost surely in this case) that the mean intensity of this peptide differs between the two considered groups. Conversely, the posterior distribution of the difference of means for the ARATH peptide



(a) `AALEELVK` peptide from the `P12081ups|SYHC_HUMAN_UPS` protein.

(b) `VLPLIIPILSK` peptide from the `sp|F4I893|ILA_ARATH` protein.

Figure 5.3: Posterior distributions of the difference of means between the 0.05 fmol UPS spike condition (μ_1) and the 10 fmol UPS spike condition (μ_7) and the corresponding boxplots summarising the observed data. The 95% credible interval is indicated by the blue central region.

(Figure 5.3b) suggests that the probability that means differ is low. Those conclusions support the summaries of raw data depicted on the bottom panel of Figure 5.3. Moreover, the posterior distribution provides additional insights on whether a peptide is under-expressed or over-expressed in a condition compared to another. For example, looking back to the UPS peptide, Figure 5.3a suggests an over-expression of the `AALEELVK` peptide in the seventh group (being the condition with the highest amount of UPS spike) compared to the first group (being the condition with the lowest amount of UPS spike), which is consistent with the experimental design. Furthermore, the middle panel merely highlights the fact that the posterior distribution of the difference $\mu_1 - \mu_7$ is the symmetric of $\mu_7 - \mu_1$, thus the sense of the comparison only remains an aesthetic choice.

5.4.2 The benefit of intra-protein correlation

One of the main benefits of our methodology is to account for between-peptides correlation, as described in Section 5.2. As the first illustration of such property, we modelled

correlations between all quantified peptides derived from the same protein. In order to highlight the gains that we may expect from such modelling, we displayed on Figure 5.4 the comparison between a differential analysis using our univariate method or using the multivariate approach. Recall the quantification data from the previous subsection. In this

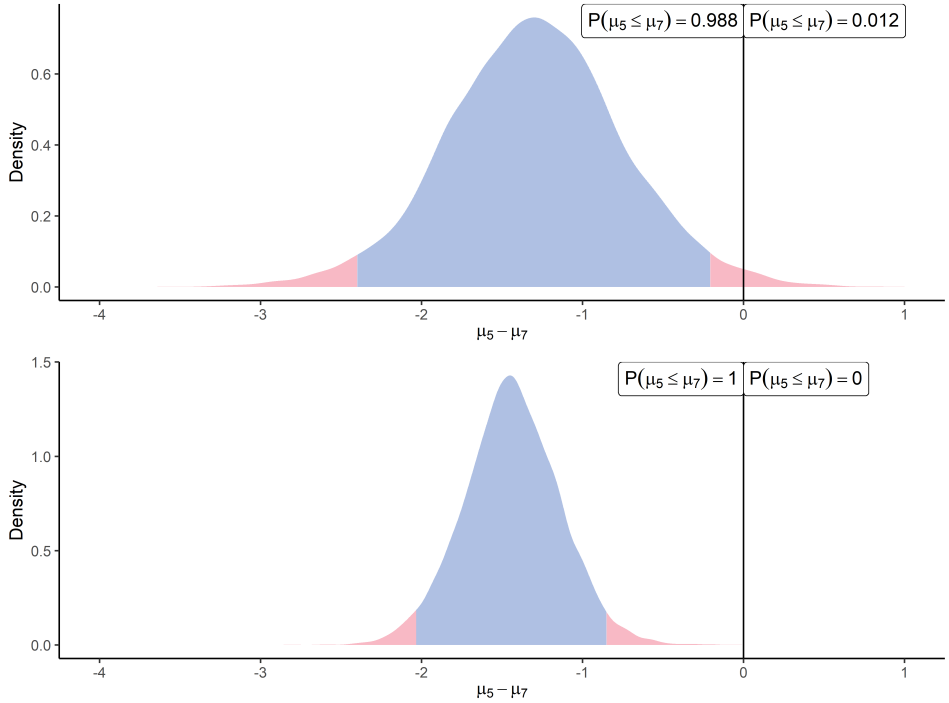


Figure 5.4: Posterior distributions of the mean difference $\mu_5 - \mu_7$ for the `AALEELVK` peptide from the `P12081ups|SYHC_HUMAN_UPS` protein using the univariate approach (top) and the multivariate approach (bottom). The 95% credible interval is indicated by the blue central region.

example, we purposefully considered a group of 9 peptides coming from the same protein (`P12081ups|SYHC_HUMAN_UPS`), which intensities may undoubtedly be correlated to some degree. We consider in this section the comparison of intensity means between the fifth point (2.5 fmol UPS - $\boldsymbol{\mu}_5$) and the seventh point (10 fmol UPS - $\boldsymbol{\mu}_7$) of the UPS spike range. The posterior difference of the mean vector $\boldsymbol{\mu}_5 - \boldsymbol{\mu}_7$ between two conditions has been computed, and the first peptide (`AALEELVK`) has been extracted for graphical visualisation. Meanwhile, the univariate algorithm has also been applied to compute the posterior difference $\mu_5 - \mu_7$, solely on the peptide `AALEELVK`. The top panel of Figure 5.4 displays the latter approach, while the multivariate case is exhibited on the bottom panel. One should observe clearly that, while the location parameter of the two distributions is close as expected, the multivariate approach takes advantage of the information coming from the correlated peptides to reduce the uncertainty in the posterior estimation. This lower variance provides a tighter range of probable values, enabling a more precise estimation of the effect size and increased

confidence in the resulting inference (deciding whether the peptide is differential or not).

5.4.3 The mirage of imputed data

After discussing the advantages and the valuable interpretative properties of our methods, let us mention a pitfall that one should avoid for the inferences to remain valid. In the case of univariate analysis, we pointed out thanks to Equation (5.4) that all the useful information is contained on observed data, and no imputation is needed since we already integrated out all missing data. Imputation does actually not even make sense in one dimension since, by definition, a missing data point is simply equivalent to an unobserved one, and we shall gain more information only by collecting more data. Therefore, one should be really careful when dealing with imputed datasets and keep in mind that imputation somehow *creates* new data points that do not bear any additional information. Thus, there is a risk of artificially decreasing the uncertainty of our estimated posterior distributions simply by considering more data points in the computations than what was genuinely observed. For

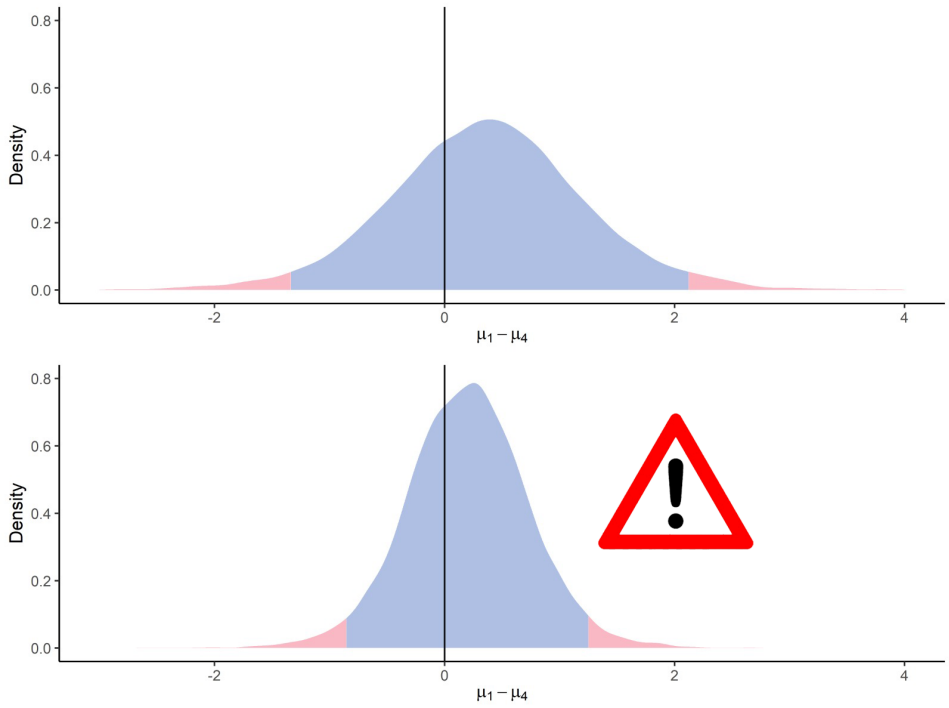


Figure 5.5: Posterior distributions of the mean difference $\mu_1 - \mu_4$ for the `EVQELAQEAAER` peptide from the `sp|F4I893|ILA_ARATH` protein using the observed dataset (top) and the imputed dataset (bottom). The 95% credible interval is indicated by the blue central region.

instance, imagine a dummy example where 10 points are effectively observed, and 1000 remain missing. It would be a massive error and underestimation of the true variance to impute the 1000 missing points (say with the average of the ten observed ones) and use

the resulting 1010-dimensional vector for computing the posterior distributions of the mean. Let us mention that such a problem is not specific to our framework and more generally also applies to Rubin's rules. One should keep in mind that those approximations only holds for a reasonable ratio of missing data. Otherwise, one may consider adapting the method, for example, by penalising the degree of freedom in the relevant t-distributions. To illustrate this issue, we displayed on Figure 5.5 an example of our univariate algorithm applied both on the observed dataset (top panel) and the imputed dataset (bottom panel). In this context, we observe a reduced variance for the imputed data. However, this behaviour is just an artefact of the phenomenon mentioned above: the bottom graph is merely not valid, and only raw data should be used in our univariate algorithm to avoid spurious inference results. More generally, while imputation is sometimes needed for the methods to work, one should always keep in mind that it always constitutes a bias (although controlled) that should be accounted for with tailored solutions, as this manuscript intends to provide.

5.4.4 Acknowledging the effect size

After discussing methodological aspects, let us dive into more biological-related properties displayed on Figure 5.6. The three panels describe the increasing differences that can be

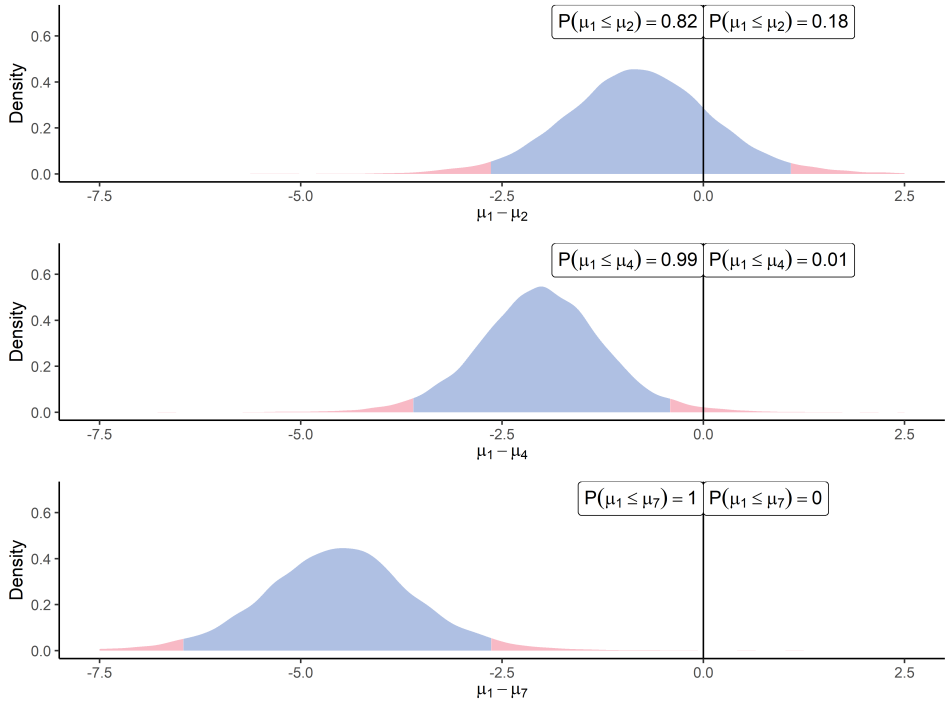


Figure 5.6: Posterior distributions of the mean differences $\mu_1 - \mu_2$, $\mu_1 - \mu_4$ and $\mu_1 - \mu_7$ for the `AALEELVK` peptide from the `P12081ups|SYHC_HUMAN_UPS` protein. The 95% credible interval is indicated by the blue central region.

observed when we compare sequentially the first point (0.05 fmol UPS) of the UPS spike range (μ_1) to the second one (0.25 fmol UPS - μ_2), the fourth one (1.25 fmol UPS - μ_4) and the highest one (25 fmol UPS - μ_7). The experimental design suggests that the difference in means for a UPS peptide should increase with respect to the amount of UPS proteins that was spiked in the biological sample (see Chapter 2). This illustration offers a perspective on how this difference becomes more and more noticeable, though mitigated by the inherent variability. Such an explicit and adequately quantified variance, and the induced uncertainty in the estimation, should help practitioners to make more educated decisions with the appropriate degree of caution. In particular, Figure 5.6 highlights the importance to consider the effect size (increasing here), which is crucial when studying the underlying biological phenomenon. Such a graph may recall us that statistical inference should be more about offering helpful insights to experts of a particular domain, rather than defining automatic and blind decision-making procedures (Betensky, 2019). Moreover, let us point out that current statistical tests used for differential analysis express their results solely as p-values. One should keep in mind that, no matter their value, they do not provide any information about the effect size of the phenomenon (Sullivan and Feinn, 2012).

5.4.5 About protein inference

To conclude on the practical usage of the proposed multivariate algorithm, let us develop ideas for comparing simultaneously multiple peptides or proteins. As highlighted before, accounting for the covariances between peptides tends to reduce the uncertainty on the posterior distribution of a unique peptide. However, we only exhibited examples comparing one peptide at a time between two conditions, although in applications, practitioners often need to compare thousands of them simultaneously. From a practical point of view, while possible in theory, we probably want to avoid modelling the correlations between every combination of peptides into a full rank matrix for at least two reasons.

First, it probably does not bear much sense to assume that all peptides in a biological sample interact with no particular structure. Secondly, it appears unreasonable to do so from a statistical and practical point of view. Computing and storing a matrix with roughly 10^4 rows and columns induces a computational and memory burden that would complicate the procedure while potentially leading to unreliable objects if matrices are estimated merely on a few data points, as for our example. However, a more promising approach would consist in deriving a sparse approach by levering the underlying structure of data from a biological perspective. If we reasonably assume, as before, that only peptides from common proteins present non-negligible correlations, it is then straightforward to define a block-diagonal matrix for the complete vector of peptides, which would be far more reasonable to estimate. Such an approach would take advantage of both of our algorithms by using the factorisation (as in Equation (5.4)) over thousands of proteins to sequentially estimate a high number of low dimensional mean vectors. Assuming an example with a thousand proteins

containing ten peptides each, the approximate computing and storage requirements would be reduced from a $(10^4)^2 = 10^8$ order of magnitude (due to one high-dimensional matrix) to $10^3 \times 10^2 = 10^5$ (a thousand of small matrices). In our applicative context, the strategy of dividing a big problem into independent smaller ones appear beneficial from both the applicative and statistical perspective.

This being said, the question of the *global* inference, in contrast with a peptide-by-peptide approach, remains pregnant. To illustrate this topic, let us provide on Figure 5.7 an example of simultaneous differential analysis for nine peptides from the same protein. According to our previous recommendations, we accounted for the correlations through the multivariate algorithm and displayed the results in posterior mean's differences for each peptide from the `P12081ups|SYHC_HUMAN_UPS` protein at once (*i.e.* $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_7$). In this example, eight peptides over nine contained in the protein are clearly differential in the same direction with comparable effect sizes, corroborating our intuition of correlated quantities. However, the situation may become far trickier when distributions lie closer to 0 on the x-axis or if only one peptide presents a clear differential pattern. As multiple and heterogeneous situations could be encountered, we do not provide here recommendations for directly dealing with protein-scale inference. Once again, the criterium for deciding what should be considered as *different enough* is highly dependent on the context and reasonable hypotheses, and no arbitrary threshold may bear any kind of general relevancy. However, we should still point out that our Bayesian framework provides convenient and natural interpretations in terms of a probability for each peptide individually. It is then straightforward to construct probabilistic decision rules and combine them to reach a multivariate inference tool, for instance, by computing an average probability for the means' difference to be below 0 across all peptides. However, one should note that probability rules prevent directly deriving global probabilistic statements without closely looking at dependencies between the single events (for instance, the factorisation in Equation (5.4) holds thanks to the induced independence between peptides). Although such an automatic procedure cannot replace the expert analysis, it may still provide a handy tool for extracting the most noteworthy results from a massive number of comparisons, which the practitioner should look at more closely afterwards. Therefore, once a maximal risk of the adverse event or a minimum probability of the desired outcome has been defined, one may derive the adequate procedure to reach those properties.

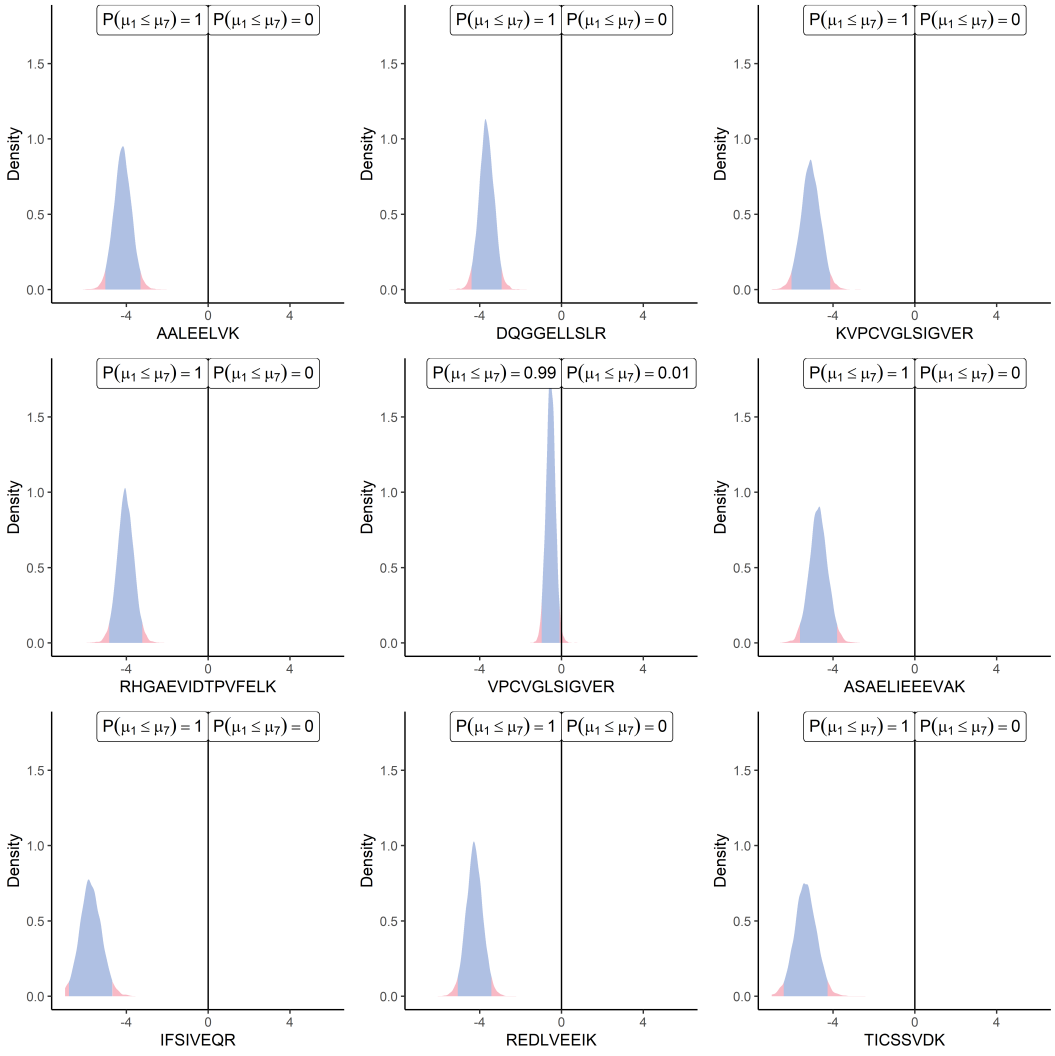


Figure 5.7: Posterior distributions of mean difference $\mu_1 - \mu_7$ for the nine peptides from the `P12081ups|SYHC_HUMAN_UPS` protein using the multivariate approach. The 95% credible interval is indicated by the blue central region.

5.5 Conclusion and perspectives

This chapter presents a Bayesian inference framework to tackle the problem of differential analysis in both univariate and multivariate context, while accounting for possible missing data. We proposed two algorithms, levering classical results from conjugate priors to compute posterior distributions and easily sample the difference of means when comparing groups of interest. For handling the recurrent problem of missing data, our multivariate approach takes advantage of the multiple imputations' approximation, while the univariate framework allows us to simply ignore this issue. In addition, this methodology aims at providing information not only on the probability of the means' difference to be null, but also on the uncertainty quantification as well as the effect sizes, which are crucial in a biological framework.

We believe that such probabilistic statements offer valuable inference tools to the practitioners. In the particular context of differential proteomics, this methodology allows us to account for between-peptides correlations. With an adequate decision rule and an appropriate correlation structure, Bayesian inference could be used in large-scale proteomics experiments, such as label-free global quantification strategies. Nevertheless, targeted proteomics experiments could already benefit from this approach, as the set of considered peptides is restricted. Furthermore, such experiments used in biomarker research could greatly benefit from the quantification of the uncertainty and the assessment of the effect sizes.

Although promising and illustrated on real applicative problems, this work still remains under development and would necessitate a further extensive simulation study for assessing more precisely the properties of the method. Readers could also benefit from more insights about practical usage, by providing intuitions for calibration of the hyper-parameters or precise estimations of the expected running times. Finally, while we considered the influences at a protein-scale, introducing correlations according to different biological features would represent an interesting path to explore.

A

Appendix

A.1 Appendix for Chapter 2

Protein ID	Peptide Sequence	SWATH-MS			
		LOD	LLOQ	ULOQ	DR
sp|P02510|CRYAB_BOVIN	HFSPEELK	12,50	62,50	6250	100
	FSVNLDVK	1250,00	-	-	-
sp|Q3T149|HSPB1_BOVIN	ALPAAAIEGPAYNR	12,50	12,50	6250	500
	SATQSAEITIPVTFQAR	62,50	125,00	6250	50
sp|P48644|AL1A1_BOVIN	QAFQIGSPWR	12,50	62,50	6250	100
	LECGGGPWGNK	12,50	62,50	6250	100
sp|O77834|PRDX6_BOVIN	VIISLQLTAEK	62,50	625,00	6250	10
	LAPEFAK	-	-	-	-
sp|Q8MKH6|TNNT1_BOVIN	YEINVLYNR	31,25	31,25	3125	100
	AQELSDWIHQLESEK	6250,00			
sp|Q3T145|MDHC_BOVIN	VIVVGNPANTNCLTASK	6,25	6,25	3125	500
	LGVTSDDVK	6,25	6,25	3125	500
tr|F1MR86|F1MR86_BOVIN	NPITGFGK	6,25	6,25	3125	500
	CLQPLASETFVAK	31,25	31,25	3125	100
sp|Q9BE40|MYH1_BOVIN	TLALLFSGPASGEAEGGPK	62,50	62,50	6250	100
	GQTVEQVYNAVGALAK	625,00	-	-	-
sp|Q5E956|TPIS_BOVIN	VVLAYEPVWAIGTGK	62,50	62,50	6250	100
	NNLGELINTLNAAK	625,00	625,00	6250	10
sp|Q3ZC09|ENOB_BOVIN	TAIQAAGYPDK	62,50	62,50	6250	100
	VNQIGSVTESIQACK	12,50	12,50	6250	500

Table A.1: Limit of detection, limits of quantification and dynamic ranges of the SWATH-MS assay established with the accurately quantified stable isotope-labelled peptides. The limits of detection and quantification are expressed in fmol/µg of muscular protein to limit bias towards the chromatographic system. DR: Dynamic range. LLOQ: Lower limit of quantification. LOD: Limit of detection. ULOQ: Upper limit of quantification.

A.2 Appendix for Chapter 3

In Chapter 3, we aim at describing a new workflow for differential analysis of proteomics data that accounts for the variability induced by the multiple imputation process. Our methodology was compared on both simulated and real datasets to the `DAPAR` state-of-the-art methodology, using confusion matrix-based indicators described in Section 3.3.1.c. In this section, we provide detailed results for those indicators for all the considered experiments.

A.2.1 Evaluation on simulated datasets

A.2.1.a Under Missing At Random assumption

EVALUATION ON THE FIRST SET OF MAR SIMULATIONS In the following, we provide the detailed results of the evaluation of the performance of the `mi4p` workflow compared to the `DAPAR` workflow on the first set of MAR simulations. Results are expressed as the mean of the given indicator over the 100 simulated datasets $\pm$ the mean of the standard deviations of the given indicator over the 100 simulated datasets. Results are based on adjusted p-values using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) with a false discovery rate of 1% and a significance level of 5%.

Table A.2, page 135: Performance evaluation on the first set of MAR simulations imputed using maximum likelihood estimation.

Table A.3, page 136: Performance evaluation on the first set of MAR simulations imputed using k-nearest neighbours.

Table A.4, page 137: Performance evaluation on the first set of MAR simulations imputed using Bayesian linear regression.

Table A.5, page 138: Performance evaluation on the first set of MAR simulations imputed using principal component analysis.

Table A.6, page 139: Performance evaluation on the first set of MAR simulations imputed using random forests.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	10 ± 0	0.5 ± 0.7	189.5 ± 0.7	0 ± 0	100 ± 0	99.8 ± 0.4	95.9 ± 5.7	97.8 ± 3.1	97.8 ± 3.1
	MI4P	10 ± 0	0.5 ± 0.7	189.5 ± 0.7	0 ± 0	100 ± 0	99.8 ± 0.4	96 ± 5.7	97.9 ± 3.1	97.8 ± 3.1
5%	**DAPAR**	10 ± 0	0.8 ± 1	189.2 ± 1	0 ± 0	100 ± 0	99.6 ± 0.5	92.9 ± 7.6	96.2 ± 4.2	96.1 ± 4.2
	MI4P	10 ± 0	0.5 ± 0.7	189.5 ± 0.7	0 ± 0	100 ± 0	99.8 ± 0.4	95.9 ± 6.1	97.8 ± 3.3	97.8 ± 3.4
10%	**DAPAR**	10 ± 0	1.2 ± 1.3	188.8 ± 1.3	0 ± 0	100 ± 0	99.4 ± 0.7	90.3 ± 9.3	94.6 ± 5.4	94.6 ± 5.3
	MI4P	10 ± 0	0.6 ± 0.8	189.4 ± 0.8	0 ± 0	100 ± 0	99.7 ± 0.4	95.3 ± 6.8	97.5 ± 3.7	97.4 ± 3.8
15%	**DAPAR**	10 ± 0	1.3 ± 1.3	188.7 ± 1.3	0 ± 0	100 ± 0	99.3 ± 0.7	89.6 ± 9.4	94.2 ± 5.4	94.2 ± 5.4
	MI4P	10 ± 0	0.6 ± 1	189.4 ± 1	0 ± 0	100 ± 0	99.7 ± 0.5	95.3 ± 7.4	97.4 ± 4.2	97.4 ± 4.2
20%	**DAPAR**	10 ± 0	2.2 ± 1.7	187.7 ± 1.7	0 ± 0	100 ± 0	98.8 ± 0.9	83.1 ± 10.9	90.4 ± 6.6	90.5 ± 6.4
	MI4P	10 ± 0	1.3 ± 1.7	188.6 ± 1.8	0 ± 0	100 ± 0	99.3 ± 0.9	89.8 ± 11.4	94.2 ± 6.7	94.3 ± 6.6
25%	**DAPAR**	10 ± 0.2	2.9 ± 2.1	186.8 ± 2.2	0 ± 0	100 ± 0	98.5 ± 1.1	79.7 ± 12.5	88.2 ± 7.9	88.3 ± 7.5
	MI4P	10 ± 0.2	1.6 ± 1.8	188 ± 2.1	0 ± 0	100 ± 0	99.2 ± 1	88.3 ± 12	93.3 ± 7.2	93.4 ± 7

Table A.2: Performance evaluation on the first set of MAR simulations imputed using maximum likelihood estimation.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	10 ± 0	0.4 ± 0.6	189.6 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	96.3 ± 5.4	98 ± 2.9	98 ± 2.9
	MI4P	10 ± 0	0.4 ± 0.6	189.6 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	96.3 ± 5.4	98 ± 2.9	98 ± 2.9
5%	**DAPAR**	10 ± 0	0.3 ± 0.5	189.7 ± 0.5	0 ± 0	100 ± 0	99.9 ± 0.3	97.7 ± 4.5	98.8 ± 2.4	98.7 ± 2.5
	MI4P	10 ± 0	0.3 ± 0.5	189.7 ± 0.5	0 ± 0	100 ± 0	99.9 ± 0.3	97.7 ± 4.5	98.8 ± 2.4	98.7 ± 2.5
10%	**DAPAR**	10 ± 0	0.3 ± 0.6	189.7 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	97.2 ± 4.9	98.5 ± 2.6	98.5 ± 2.7
	MI4P	10 ± 0	0.3 ± 0.6	189.7 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	97.2 ± 4.9	98.5 ± 2.6	98.5 ± 2.7
15%	**DAPAR**	10 ± 0.1	0.2 ± 0.6	189.8 ± 0.6	0 ± 0.1	99.9 ± 1	99.9 ± 0.3	97.9 ± 4.7	98.8 ± 2.6	98.8 ± 2.6
	MI4P	10 ± 0.1	0.2 ± 0.6	189.8 ± 0.6	0 ± 0.1	99.9 ± 1	99.9 ± 0.3	97.9 ± 4.7	98.8 ± 2.6	98.8 ± 2.6
20%	**DAPAR**	9.9 ± 0.2	0.4 ± 0.7	189.6 ± 0.7	0.1 ± 0.2	99.4 ± 2.4	99.8 ± 0.4	96.2 ± 5.8	97.6 ± 3.3	97.6 ± 3.3
	MI4P	9.9 ± 0.2	0.4 ± 0.7	189.6 ± 0.7	0.1 ± 0.2	99.4 ± 2.4	99.8 ± 0.4	96.2 ± 5.8	97.6 ± 3.3	97.6 ± 3.3
25%	**DAPAR**	9.8 ± 0.5	0.9 ± 1	189.1 ± 1	0.2 ± 0.5	97.7 ± 4.7	99.5 ± 0.5	92.7 ± 7.8	94.9 ± 4.7	94.8 ± 4.8
	MI4P	9.8 ± 0.5	0.9 ± 1	189.1 ± 1	0.2 ± 0.5	97.7 ± 4.7	99.5 ± 0.5	92.7 ± 7.8	94.9 ± 4.7	94.8 ± 4.8

Table A.3: Performance evaluation on the first set of MAR simulations imputed using k-nearest neighbours.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	10 ± 0.2	0.2 ± 0.4	189.8 ± 0.4	0 ± 0.2	99.8 ± 2	99.9 ± 0.2	98.4 ± 3.7	99 ± 2.2	99 ± 2.2
	MI4P	9.9 ± 0.3	0.2 ± 0.4	189.8 ± 0.4	0.1 ± 0.3	99.3 ± 2.9	99.9 ± 0.2	98.3 ± 4	98.7 ± 2.8	98.7 ± 2.8
5%	**DAPAR**	10 ± 0.2	0.2 ± 0.4	189.8 ± 0.4	0 ± 0.2	99.6 ± 2	99.9 ± 0.2	98.6 ± 3.7	99 ± 2.1	99 ± 2.2
	MI4P	9.7 ± 0.5	0.2 ± 0.4	189.8 ± 0.4	0.3 ± 0.5	96.9 ± 5.4	99.9 ± 0.2	97.9 ± 4.1	97.3 ± 3.4	97.2 ± 3.5
10%	**DAPAR**	10 ± 0	0.2 ± 0.5	189.8 ± 0.5	0 ± 0	100 ± 0	99.9 ± 0.2	97.8 ± 4.1	98.9 ± 2.1	98.8 ± 2.2
	MI4P	9.6 ± 0.7	0.1 ± 0.3	189.9 ± 0.3	0.4 ± 0.7	95.5 ± 6.9	100 ± 0.1	99.2 ± 2.6	97.2 ± 4	97.1 ± 4
15%	**DAPAR**	10 ± 0	0.3 ± 0.6	189.7 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	97.2 ± 4.9	98.5 ± 2.6	98.5 ± 2.7
	MI4P	9.2 ± 0.9	0 ± 0.2	190 ± 0.2	0.8 ± 0.9	91.7 ± 8.8	100 ± 0.1	99.6 ± 1.8	95.3 ± 4.9	95.3 ± 4.8
20%	**DAPAR**	10 ± 0	0.6 ± 0.8	189.4 ± 0.8	0 ± 0	100 ± 0	99.7 ± 0.4	94.6 ± 6.4	97.1 ± 3.5	97.1 ± 3.6
	MI4P	8.9 ± 1	0 ± 0.1	190 ± 0.1	1.1 ± 1	89.1 ± 10.3	100 ± 0.1	99.9 ± 1	93.9 ± 6.1	93.9 ± 5.9
25%	**DAPAR**	10 ± 0.1	1.2 ± 1.1	188.8 ± 1.1	0 ± 0.1	99.9 ± 1	99.4 ± 0.6	90.3 ± 8	94.7 ± 4.6	94.6 ± 4.6
	MI4P	8.9 ± 1.1	0 ± 0	190 ± 0	1.1 ± 1.1	89.3 ± 11.1	100 ± 0	100 ± 0	94 ± 6.7	94.1 ± 6.4

Table A.4: Performance evaluation on the first set of MAR simulations imputed using Bayesian linear regression.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	10 ± 0	0.5 ± 0.7	189.5 ± 0.7	0 ± 0	100 ± 0	99.7 ± 0.4	95.8 ± 6	97.8 ± 3.3	97.7 ± 3.3
	MI4P	10 ± 0	0.5 ± 0.7	189.5 ± 0.7	0 ± 0	100 ± 0	99.7 ± 0.4	95.8 ± 6	97.8 ± 3.3	97.7 ± 3.3
5%	**DAPAR**	10 ± 0	0.6 ± 0.8	189.4 ± 0.8	0 ± 0	100 ± 0	99.7 ± 0.4	94.6 ± 6.8	97.1 ± 3.7	97 ± 3.7
	MI4P	10 ± 0	0.6 ± 0.8	189.4 ± 0.8	0 ± 0	100 ± 0	99.7 ± 0.4	94.6 ± 6.8	97.1 ± 3.7	97 ± 3.7
10%	**DAPAR**	10 ± 0	1 ± 1.1	189 ± 1.1	0 ± 0	100 ± 0	99.5 ± 0.6	91.8 ± 8.3	95.5 ± 4.7	95.5 ± 4.7
	MI4P	10 ± 0	1 ± 1.1	189 ± 1.1	0 ± 0	100 ± 0	99.5 ± 0.6	91.8 ± 8.3	95.5 ± 4.7	95.5 ± 4.7
15%	**DAPAR**	10 ± 0	1.2 ± 1.2	188.8 ± 1.2	0 ± 0	100 ± 0	99.4 ± 0.6	90.1 ± 8.9	94.5 ± 5.1	94.5 ± 5.1
	MI4P	10 ± 0	1.2 ± 1.2	188.8 ± 1.2	0 ± 0	100 ± 0	99.4 ± 0.6	90.1 ± 8.9	94.5 ± 5.1	94.5 ± 5.1
20%	**DAPAR**	10 ± 0	1.9 ± 1.5	188 ± 1.5	0 ± 0	100 ± 0	99 ± 0.8	85.1 ± 9.8	91.6 ± 5.9	91.6 ± 5.7
	MI4P	10 ± 0	1.9 ± 1.5	188 ± 1.5	0 ± 0	100 ± 0	99 ± 0.8	85.4 ± 9.8	91.8 ± 5.9	91.8 ± 5.7
25%	**DAPAR**	10 ± 0.2	2.5 ± 1.6	187.2 ± 1.7	0 ± 0	100 ± 0	98.7 ± 0.9	81 ± 10.5	89.1 ± 6.4	89.2 ± 6.1
	MI4P	10 ± 0.2	2.6 ± 1.6	186.8 ± 2	0 ± 0	100 ± 0	98.6 ± 0.9	80.5 ± 10.5	88.8 ± 6.4	88.9 ± 6.2

Table A.5: Performance evaluation on the first set of MAR simulations imputed using principal component analysis.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	10 ± 0	0.5 ± 0.7	189.5 ± 0.7	0 ± 0	100 ± 0	99.8 ± 0.4	96 ± 6	97.9 ± 3.3	97.8 ± 3.3
	MI4P	10 ± 0	0.5 ± 0.7	189.5 ± 0.7	0 ± 0	100 ± 0	99.8 ± 0.4	96 ± 6	97.9 ± 3.3	97.8 ± 3.3
5%	**DAPAR**	10 ± 0	0.4 ± 0.6	189.6 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	96 ± 5.3	97.9 ± 2.8	97.8 ± 2.9
	MI4P	10 ± 0	0.4 ± 0.6	189.6 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	96 ± 5.3	97.9 ± 2.8	97.8 ± 2.9
10%	**DAPAR**	10 ± 0	0.5 ± 0.8	189.5 ± 0.8	0 ± 0	100 ± 0	99.7 ± 0.4	95.8 ± 6.7	97.7 ± 3.7	97.7 ± 3.7
	MI4P	10 ± 0.1	0.5 ± 0.8	189.5 ± 0.8	0 ± 0.1	99.8 ± 1.4	99.7 ± 0.4	95.9 ± 6.4	97.7 ± 3.6	97.6 ± 3.6
15%	**DAPAR**	10 ± 0	0.3 ± 0.6	189.7 ± 0.6	0 ± 0	100 ± 0	99.8 ± 0.3	97.2 ± 5.3	98.5 ± 2.9	98.5 ± 2.9
	MI4P	10 ± 0.1	0.4 ± 0.7	189.6 ± 0.7	0 ± 0.1	99.8 ± 1.4	99.8 ± 0.3	96.8 ± 5.5	98.2 ± 3	98.1 ± 3
20%	**DAPAR**	10 ± 0.1	0.4 ± 0.6	189.5 ± 0.7	0 ± 0.1	99.8 ± 1.4	99.8 ± 0.3	96.3 ± 5.4	97.9 ± 3	97.9 ± 3.1
	MI4P	10 ± 0.1	0.4 ± 0.6	189.4 ± 0.8	0 ± 0.1	99.8 ± 1.4	99.8 ± 0.3	96 ± 5.4	97.8 ± 3	97.7 ± 3.1
25%	**DAPAR**	10 ± 0.2	0.3 ± 0.6	189.4 ± 0.9	0 ± 0.1	99.9 ± 1	99.8 ± 0.3	97.5 ± 5	98.6 ± 2.7	98.6 ± 2.8
	MI4P	9.9 ± 0.2	0.3 ± 0.6	189.1 ± 1.3	0 ± 0.2	99.7 ± 1.7	99.9 ± 0.3	97.5 ± 4.9	98.5 ± 2.7	98.5 ± 2.8

Table A.6: Performance evaluation on the first set of MAR simulations imputed using random forests.

EVALUATION ON THE SECOND SET OF MAR SIMULATIONS In the following, we provide the detailed results of the evaluation of the performance of the `mi4p` workflow compared to the `DAPAR` workflow on the second set of MAR simulations. Results are expressed as the mean of the given indicator over the 100 simulated datasets ± the mean of the standard deviations of the given indicator over the 100 simulated datasets. Results are based on adjusted p-values using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) with a false discovery rate of 1% and a significance level of 5%.

Table A.7, page 141: Performance evaluation on the second set of MAR simulations imputed using maximum likelihood estimation.

Table A.8, page 142: Performance evaluation on the second set of MAR simulations imputed using k-nearest neighbours.

Table A.9, page 143: Performance evaluation on the second set of MAR simulations imputed using Bayesian linear regression.

Table A.10, page 144: Performance evaluation on the second set of MAR simulations imputed using principal component analysis.

Table A.11, page 145: Performance evaluation on the second set of MAR simulations imputed using random forests.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	80.8 ± 11.4	1.9 ± 1.5	798.1 ± 1.5	119.2 ± 11.4	40.4 ± 5.7	99.8 ± 0.2	97.8 ± 1.6	56.9 ± 5.9	58.2 ± 4.5
	MI4P	166.9 ± 5	6.3 ± 2.7	793.7 ± 2.7	33.1 ± 5	83.4 ± 2.5	99.2 ± 0.3	96.4 ± 1.4	89.4 ± 1.5	87.4 ± 1.6
5%	**DAPAR**	80.8 ± 12.1	2.4 ± 1.8	797.6 ± 1.8	119.2 ± 12.1	40.4 ± 6.1	99.7 ± 0.2	97.3 ± 1.9	56.8 ± 6.1	58 ± 4.6
	MI4P	164.2 ± 6.1	6.1 ± 3.5	793.9 ± 3.5	35.8 ± 6.1	82.1 ± 3	99.2 ± 0.4	96.5 ± 1.9	88.7 ± 1.5	86.6 ± 1.6
10%	**DAPAR**	78.8 ± 11.9	2.4 ± 1.6	797.6 ± 1.6	121.2 ± 11.9	39.4 ± 5.9	99.7 ± 0.2	97.1 ± 1.8	55.8 ± 6.1	57.1 ± 4.7
	MI4P	160.7 ± 7.8	5.6 ± 3.8	794.4 ± 3.8	39.3 ± 7.8	80.4 ± 3.9	99.3 ± 0.5	96.7 ± 2.1	87.7 ± 1.9	85.6 ± 2
15%	**DAPAR**	80.3 ± 11.4	3.3 ± 1.9	796.7 ± 1.9	119.7 ± 11.4	40.1 ± 5.7	99.6 ± 0.2	96.1 ± 2.1	56.4 ± 5.8	57.3 ± 4.6
	MI4P	159 ± 8.8	6.7 ± 5.1	793.3 ± 5.1	41 ± 8.8	79.5 ± 4.4	99.2 ± 0.6	96.2 ± 2.7	86.9 ± 2.1	84.7 ± 2.2
20%	**DAPAR**	81.3 ± 11.6	4 ± 2.1	796 ± 2.1	118.7 ± 11.6	40.7 ± 5.8	99.5 ± 0.3	95.4 ± 2.4	56.8 ± 5.9	57.4 ± 4.7
	MI4P	158 ± 9.8	7.2 ± 5.4	792.8 ± 5.4	42 ± 9.8	79 ± 4.9	99.1 ± 0.7	95.8 ± 2.9	86.5 ± 2.3	84.2 ± 2.3
25%	**DAPAR**	82.5 ± 12.3	4.7 ± 2.7	795.3 ± 2.7	117.5 ± 12.3	41.2 ± 6.2	99.4 ± 0.3	94.7 ± 2.8	57.2 ± 6	57.5 ± 4.8
	MI4P	154.5 ± 10.4	6.9 ± 6.2	793.1 ± 6.2	45.5 ± 10.4	77.3 ± 5.2	99.1 ± 0.8	96 ± 3.3	85.4 ± 2.5	83.1 ± 2.4

Table A.7: Performance evaluation on the second set of MAR simulations imputed using maximum likelihood estimation.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	80.5 ± 12.1	1.8 ± 1.4	798.2 ± 1.4	119.5 ± 12.1	40.2 ± 6	99.8 ± 0.2	97.9 ± 1.6	56.8 ± 6.3	58.1 ± 4.9
	MI4P	167.9 ± 4.8	6.6 ± 2.5	793.4 ± 2.5	32 ± 4.8	84 ± 2.4	99.2 ± 0.3	96.2 ± 1.4	89.7 ± 1.4	87.6 ± 1.7
5%	**DAPAR**	79.6 ± 12.4	1.9 ± 1.7	798.1 ± 1.7	120.4 ± 12.4	39.8 ± 6.2	99.8 ± 0.2	97.8 ± 1.9	56.2 ± 6.5	57.7 ± 5
	MI4P	169.6 ± 4.3	6.7 ± 2.8	793.3 ± 2.8	30.4 ± 4.3	84.8 ± 2.2	99.2 ± 0.4	96.2 ± 1.5	90.1 ± 1.4	88.1 ± 1.6
10%	**DAPAR**	78.2 ± 13.5	2 ± 1.7	798 ± 1.7	121.8 ± 13.5	39.1 ± 6.8	99.8 ± 0.2	97.7 ± 1.8	55.5 ± 7.1	57.1 ± 5.4
	MI4P	170.8 ± 4.3	6.3 ± 2.8	793.7 ± 2.8	29.2 ± 4.3	85.4 ± 2.2	99.2 ± 0.4	96.5 ± 1.5	90.6 ± 1.4	88.7 ± 1.6
15%	**DAPAR**	79 ± 14.1	2 ± 1.7	798 ± 1.7	121 ± 14.1	39.5 ± 7	99.8 ± 0.2	97.6 ± 1.8	55.9 ± 7.3	57.4 ± 5.6
	MI4P	171.6 ± 4.5	6.2 ± 3.1	793.8 ± 3.1	28.4 ± 4.5	85.8 ± 2.2	99.2 ± 0.4	96.5 ± 1.7	90.8 ± 1.4	89 ± 1.7
20%	**DAPAR**	77.2 ± 16.8	1.9 ± 1.6	798.1 ± 1.6	122.8 ± 16.8	38.6 ± 8.4	99.8 ± 0.2	97.7 ± 1.9	54.7 ± 9.8	56.4 ± 7.9
	MI4P	171.1 ± 4.7	5.7 ± 2.7	794.3 ± 2.7	28.9 ± 4.7	85.5 ± 2.3	99.3 ± 0.3	96.8 ± 1.5	90.8 ± 1.4	89 ± 1.7
25%	**DAPAR**	74.4 ± 16.8	1.8 ± 1.7	798.2 ± 1.7	125.6 ± 16.8	37.2 ± 8.4	99.8 ± 0.2	97.7 ± 1.9	53.3 ± 9.8	55.3 ± 7.8
	MI4P	170.3 ± 4.9	5.9 ± 2.9	794.1 ± 2.9	29.7 ± 4.9	85.1 ± 2.5	99.3 ± 0.4	96.7 ± 1.6	90.5 ± 1.5	88.6 ± 1.8

Table A.8: Performance evaluation on the second set of MAR simulations imputed using k-nearest neighbours method.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	80.7 ± 11.9	1.9 ± 1.6	798.1 ± 1.6	119.3 ± 11.9	40.4 ± 6	99.8 ± 0.2	97.8 ± 1.8	56.8 ± 6.1	58.2 ± 4.7
	MI4P	165.7 ± 5	5.4 ± 2.4	794.6 ± 2.4	34.3 ± 5	82.8 ± 2.5	99.3 ± 0.3	96.9 ± 1.3	89.3 ± 1.5	87.3 ± 1.7
5%	**DAPAR**	80.5 ± 12.5	2.3 ± 1.7	797.7 ± 1.7	119.5 ± 12.5	40.3 ± 6.2	99.7 ± 0.2	97.3 ± 1.8	56.6 ± 6.4	57.9 ± 4.9
	MI4P	157.3 ± 5.5	2.5 ± 1.7	797.5 ± 1.7	42.6 ± 5.5	78.7 ± 2.8	99.7 ± 0.2	98.5 ± 1	87.4 ± 1.7	85.5 ± 1.7
10%	**DAPAR**	79.6 ± 12.8	2.7 ± 2	797.3 ± 2	120.4 ± 12.8	39.8 ± 6.4	99.7 ± 0.2	96.9 ± 2.1	56.1 ± 6.5	57.3 ± 5
	MI4P	156.2 ± 5.7	2.4 ± 1.6	797.6 ± 1.6	43.8 ± 5.7	78.1 ± 2.8	99.7 ± 0.2	98.5 ± 1	87.1 ± 1.8	85.2 ± 1.9
15%	**DAPAR**	80.6 ± 15	3.2 ± 2.4	796.8 ± 2.4	119.4 ± 15	40.3 ± 7.5	99.6 ± 0.3	96.3 ± 2.5	56.3 ± 8.3	57.3 ± 6.6
	MI4P	150.7 ± 6.7	1.6 ± 1.2	798.4 ± 1.2	49.3 ± 6.7	75.3 ± 3.4	99.8 ± 0.1	98.9 ± 0.8	85.5 ± 2.2	83.6 ± 2.2
20%	**DAPAR**	80.5 ± 15.3	3.9 ± 2.6	796.1 ± 2.6	119.5 ± 15.3	40.3 ± 7.6	99.5 ± 0.3	95.5 ± 2.7	56.2 ± 8.1	57 ± 6.3
	MI4P	144 ± 6.9	0.9 ± 1	799.1 ± 1	56 ± 6.9	72 ± 3.4	99.9 ± 0.1	99.4 ± 0.7	83.4 ± 2.3	81.7 ± 2.3
25%	**DAPAR**	79.7 ± 17.6	4.6 ± 3.2	795.4 ± 3.2	120.3 ± 17.6	39.9 ± 8.8	99.4 ± 0.4	94.8 ± 2.8	55.5 ± 9.5	56.3 ± 7.3
	MI4P	137.2 ± 6.7	0.6 ± 0.8	799.4 ± 0.8	62.8 ± 6.7	68.6 ± 3.3	99.9 ± 0.1	99.6 ± 0.6	81.2 ± 2.4	79.5 ± 2.3

Table A.9: Performance evaluation on the second set of MAR simulations imputed using Bayesian linear regression.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	80.6 ± 11.8	1.9 ± 1.5	798.1 ± 1.5	119.4 ± 11.8	40.3 ± 5.9	99.8 ± 0.2	97.8 ± 1.7	56.8 ± 6.1	58.1 ± 4.8
	MI4P	168.1 ± 4.8	6.8 ± 2.7	793.2 ± 2.7	31.9 ± 4.8	84 ± 2.4	99.2 ± 0.3	96.1 ± 1.5	89.7 ± 1.5	87.6 ± 1.7
5%	**DAPAR**	80.9 ± 12.6	2.4 ± 1.8	797.6 ± 1.8	119.1 ± 12.6	40.4 ± 6.3	99.7 ± 0.2	97.2 ± 2	56.8 ± 6.5	58 ± 5
	MI4P	170 ± 4.6	7.6 ± 2.9	792.5 ± 2.9	30 ± 4.6	85 ± 2.3	99.1 ± 0.4	95.8 ± 1.6	90 ± 1.4	88 ± 1.6
10%	**DAPAR**	79.9 ± 13	2.8 ± 1.9	797.2 ± 1.9	120.1 ± 13	40 ± 6.5	99.7 ± 0.2	96.8 ± 2	56.2 ± 6.6	57.4 ± 5.1
	MI4P	172.1 ± 4.6	8.2 ± 3	791.8 ± 3	27.9 ± 4.6	86.1 ± 2.3	99 ± 0.4	95.5 ± 1.5	90.5 ± 1.4	88.5 ± 1.6
15%	**DAPAR**	81.8 ± 12.9	3.6 ± 2.5	796.4 ± 2.5	118.2 ± 12.9	40.9 ± 6.4	99.6 ± 0.3	95.9 ± 2.5	57 ± 6.5	57.8 ± 5.1
	MI4P	174.2 ± 4	9.4 ± 3.6	790.6 ± 3.6	25.8 ± 4	87.1 ± 2	98.8 ± 0.5	94.9 ± 1.9	90.8 ± 1.3	88.8 ± 1.6
20%	**DAPAR**	82.1 ± 15.4	4.4 ± 2.6	795.6 ± 2.6	117.9 ± 15.4	41 ± 7.7	99.5 ± 0.3	95.1 ± 2.7	56.8 ± 8	57.4 ± 6.2
	MI4P	175.6 ± 4.1	11.3 ± 4.1	788.7 ± 4.1	24.4 ± 4.1	87.8 ± 2.1	98.6 ± 0.5	94 ± 2	90.8 ± 1.5	88.7 ± 1.8
25%	**DAPAR**	83.3 ± 14.6	5.3 ± 2.9	794.7 ± 2.9	116.7 ± 14.6	41.6 ± 7.3	99.3 ± 0.4	94.1 ± 2.8	57.3 ± 7.3	57.5 ± 5.8
	MI4P	176.3 ± 4.5	13 ± 3.8	787 ± 3.8	23.7 ± 4.5	88.1 ± 2.3	98.4 ± 0.5	93.2 ± 1.9	90.6 ± 1.5	88.4 ± 1.8

Table A.10: Performance evaluation on the second set of MAR simulations imputed using principal component analysis.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	80.8 ± 11.7	1.9 ± 1.5	798.1 ± 1.5	119.2 ± 11.7	40.4 ± 5.8	99.8 ± 0.2	97.8 ± 1.7	56.9 ± 6	58.2 ± 4.7
	MI4P	168 ± 4.7	6.8 ± 2.7	793.2 ± 2.7	32 ± 4.7	84 ± 2.4	99.2 ± 0.3	96.1 ± 1.4	89.6 ± 1.4	87.6 ± 1.7
5%	**DAPAR**	80.7 ± 12.7	2.4 ± 1.9	797.6 ± 1.9	119.3 ± 12.7	40.3 ± 6.3	99.7 ± 0.2	97.2 ± 2	56.7 ± 6.5	57.9 ± 5
	MI4P	169.9 ± 4.4	7.5 ± 3	792.5 ± 3	30.1 ± 4.4	85 ± 2.2	99.1 ± 0.4	95.8 ± 1.6	90 ± 1.4	88 ± 1.6
10%	**DAPAR**	79.9 ± 12.5	2.7 ± 1.8	797.3 ± 1.8	120.1 ± 12.5	40 ± 6.3	99.7 ± 0.2	96.8 ± 2	56.3 ± 6.4	57.5 ± 5
	MI4P	171.6 ± 4.6	8.1 ± 3.1	792 ± 3.1	28.4 ± 4.6	85.8 ± 2.3	99 ± 0.4	95.5 ± 1.6	90.4 ± 1.5	88.4 ± 1.7
15%	**DAPAR**	81.4 ± 13.8	3.5 ± 2.4	796.5 ± 2.4	118.6 ± 13.8	40.7 ± 6.9	99.6 ± 0.3	96 ± 2.4	56.8 ± 7.1	57.6 ± 5.5
	MI4P	173.5 ± 4	9.3 ± 3.8	790.7 ± 3.8	26.5 ± 4	86.8 ± 2	98.8 ± 0.5	94.9 ± 1.9	90.6 ± 1.4	88.6 ± 1.7
20%	**DAPAR**	82.1 ± 13.5	4.4 ± 2.6	795.6 ± 2.6	117.9 ± 13.5	41.1 ± 6.8	99.4 ± 0.3	95 ± 2.6	57 ± 6.9	57.5 ± 5.4
	MI4P	174.4 ± 4.1	10.9 ± 3.9	789.1 ± 3.9	25.6 ± 4.1	87.2 ± 2	98.6 ± 0.5	94.1 ± 2	90.5 ± 1.4	88.4 ± 1.7
25%	**DAPAR**	82.2 ± 16	5 ± 2.9	795 ± 2.9	117.8 ± 16	41.1 ± 8	99.4 ± 0.4	94.4 ± 2.8	56.8 ± 8.5	57.2 ± 6.7
	MI4P	174.7 ± 4.5	12.4 ± 4	787.6 ± 4	25.3 ± 4.5	87.3 ± 2.2	98.5 ± 0.5	93.4 ± 1.9	90.3 ± 1.5	88 ± 1.8

Table A.11: Performance evaluation on the second set of MAR simulations imputed using random forests.

EVALUATION ON THE THIRD SET OF MAR SIMULATIONS In the following, we provide the detailed results of the evaluation of the performance of the `mi4p` workflow compared to the `DAPAR` workflow on the third set of MAR simulations. Results are expressed as the mean of the given indicator over the 100 simulated datasets ± the mean of the standard deviations of the given indicator over the 100 simulated datasets. Results are based on adjusted p-values using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) with a false discovery rate of 1% and a significance level of 5%.

Table A.12, page 147: Performance evaluation on the third set of MAR simulations imputed using maximum likelihood estimation.

Table A.13, page 148: Performance evaluation on the third set of MAR simulations imputed using k-nearest neighbours.

Table A.14, page 149: Performance evaluation on the third set of MAR simulations imputed using Bayesian linear regression.

Table A.15, page 150: Performance evaluation on the third set of MAR simulations imputed using principal component analysis.

Table A.16, page 151: Performance evaluation on the third set of MAR simulations imputed using random forests.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	25.6 ± 10.7	0.5 ± 0.8	799.5 ± 0.8	174.4 ± 10.7	12.8 ± 5.4	99.9 ± 0.1	98.3 ± 2.4	22.2 ± 8.4	31.2 ± 7.4
	MI4P	91 ± 10.6	2.7 ± 1.8	797.3 ± 1.8	109 ± 10.6	45.5 ± 5.3	99.7 ± 0.2	97.2 ± 1.8	61.8 ± 4.9	61.9 ± 4
5%	**DAPAR**	25.6 ± 10.2	0.4 ± 0.7	799.6 ± 0.7	174.4 ± 10.2	12.8 ± 5.1	99.9 ± 0.1	98.5 ± 2.4	22.3 ± 7.9	31.4 ± 6.8
	MI4P	83 ± 13.6	2.1 ± 1.8	797.9 ± 1.8	117 ± 13.6	41.5 ± 6.8	99.7 ± 0.2	97.6 ± 1.9	57.9 ± 6.7	59 ± 5.1
10%	**DAPAR**	25.9 ± 10.8	0.6 ± 0.7	799.4 ± 0.7	174.1 ± 10.8	13 ± 5.4	99.9 ± 0.1	96.1 ± 14	22.5 ± 8.6	31.1 ± 8.3
	MI4P	80.2 ± 18.2	2.3 ± 2.1	797.7 ± 2.1	119.8 ± 18.2	40.1 ± 9.1	99.7 ± 0.3	97.5 ± 2	56.2 ± 9.2	57.6 ± 6.9
15%	**DAPAR**	26.6 ± 11.5	0.8 ± 1	799.2 ± 1	173.4 ± 11.5	13.3 ± 5.7	99.9 ± 0.1	96.5 ± 10.3	23 ± 9	31.5 ± 8.2
	MI4P	71.9 ± 22.7	2.1 ± 2.3	797.9 ± 2.3	128.1 ± 22.7	35.9 ± 11.3	99.7 ± 0.3	97.7 ± 2.3	51.4 ± 12.3	54 ± 9.1
20%	**DAPAR**	28.5 ± 12.1	1.1 ± 1.3	798.9 ± 1.3	171.5 ± 12.1	14.2 ± 6.1	99.9 ± 0.2	95.4 ± 10.4	24.3 ± 9.3	32.3 ± 8.5
	MI4P	67.1 ± 22.4	1.9 ± 2.3	798.1 ± 2.3	132.9 ± 22.4	33.6 ± 11.2	99.8 ± 0.3	97.8 ± 2.3	48.8 ± 12.4	52 ± 9.2
25%	**DAPAR**	26.9 ± 12.4	1.3 ± 1.4	798.7 ± 1.4	173.1 ± 12.4	13.4 ± 6.2	99.8 ± 0.2	96.2 ± 4	23 ± 9.7	31.1 ± 8.6
	MI4P	61.2 ± 24	2 ± 2.8	798 ± 2.8	138.8 ± 24	30.6 ± 12	99.7 ± 0.4	97.7 ± 2.8	45.2 ± 13.6	49.2 ± 10

Table A.12: Performance evaluation on the third set of MAR simulation imputed using maximum likelihood estimation

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	26 ± 10.4	0.5 ± 0.8	799.5 ± 0.8	174 ± 10.4	13 ± 5.2	99.9 ± 0.1	98.5 ± 2.3	22.5 ± 8.1	31.5 ± 7
	MI4P	95.8 ± 9.8	3.1 ± 1.9	796.9 ± 1.9	104.2 ± 9.8	47.9 ± 4.9	99.6 ± 0.2	96.9 ± 1.8	64 ± 4.4	63.6 ± 3.7
5%	**DAPAR**	25.4 ± 11.1	0.4 ± 0.7	799.6 ± 0.7	174.6 ± 11.1	12.7 ± 5.5	99.9 ± 0.1	98.5 ± 2.5	22.1 ± 8.7	31.1 ± 7.5
	MI4P	98 ± 9.9	2.9 ± 1.8	797.1 ± 1.8	102 ± 9.9	49 ± 4.9	99.6 ± 0.2	97.1 ± 1.7	65 ± 4.4	64.6 ± 3.7
10%	**DAPAR**	24.5 ± 10.6	0.6 ± 0.9	799.4 ± 0.9	175.5 ± 10.6	12.3 ± 5.3	99.9 ± 0.1	95.8 ± 14.1	21.4 ± 8.4	30.2 ± 7.9
	MI4P	101.1 ± 9.5	3.2 ± 1.8	796.8 ± 1.8	98.9 ± 9.5	50.6 ± 4.8	99.6 ± 0.2	97 ± 1.6	66.3 ± 4.1	65.6 ± 3.5
15%	**DAPAR**	25.1 ± 12.2	0.4 ± 0.7	799.6 ± 0.7	174.9 ± 12.2	12.5 ± 6.1	99.9 ± 0.1	96.4 ± 14.1	21.7 ± 9.7	30.4 ± 9.2
	MI4P	103.8 ± 10.9	2.6 ± 1.4	797.4 ± 1.4	96.2 ± 10.9	51.9 ± 5.4	99.7 ± 0.2	97.6 ± 1.3	67.6 ± 4.7	66.8 ± 4
20%	**DAPAR**	24.7 ± 13.2	0.4 ± 0.7	799.6 ± 0.7	175.3 ± 13.2	12.3 ± 6.6	99.9 ± 0.1	95.6 ± 17.1	21.3 ± 10.4	29.9 ± 10.1
	MI4P	106.2 ± 11.9	2.7 ± 1.7	797.3 ± 1.7	93.8 ± 11.9	53.1 ± 5.9	99.7 ± 0.2	97.6 ± 1.4	68.6 ± 5	67.7 ± 4.3
25%	**DAPAR**	24.7 ± 12.3	0.6 ± 0.9	799.4 ± 0.9	175.3 ± 12.3	12.3 ± 6.2	99.9 ± 0.1	96.8 ± 10.3	21.4 ± 9.7	30.1 ± 8.9
	MI4P	105.4 ± 11.1	2.9 ± 1.9	797.1 ± 1.9	94.6 ± 11.1	52.7 ± 5.5	99.6 ± 0.2	97.4 ± 1.6	68.2 ± 4.7	67.3 ± 4

Table A.13: Performance evaluation on the third set of MAR simulations imputed using k-nearest neighbours method.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	25.8 ± 10.6	0.5 ± 0.8	799.5 ± 0.8	174.2 ± 10.6	12.9 ± 5.3	99.9 ± 0.1	98.3 ± 2.5	22.4 ± 8.4	31.3 ± 7.4
	MI4P	87.9 ± 9.5	2.2 ± 1.6	797.8 ± 1.6	112.1 ± 9.5	43.9 ± 4.8	99.7 ± 0.2	97.6 ± 1.7	60.4 ± 4.5	60.9 ± 3.7
5%	**DAPAR**	25.6 ± 10.7	0.5 ± 0.7	799.5 ± 0.7	174.4 ± 10.7	12.8 ± 5.4	99.9 ± 0.1	98.4 ± 2.4	22.3 ± 8.4	31.3 ± 7.3
	MI4P	63.1 ± 10.4	0.5 ± 0.7	799.5 ± 0.7	136.9 ± 10.4	31.5 ± 5.2	99.9 ± 0.1	99.2 ± 1.1	47.6 ± 6.1	51.4 ± 4.6
10%	**DAPAR**	24.4 ± 11.5	0.6 ± 0.8	799.4 ± 0.8	175.6 ± 11.5	12.2 ± 5.7	99.9 ± 0.1	96 ± 14.1	21.2 ± 9.2	29.9 ± 8.8
	MI4P	37.2 ± 11.3	0.1 ± 0.3	799.9 ± 0.3	162.8 ± 11.3	18.6 ± 5.6	100 ± 0	99.7 ± 0.9	31 ± 8.1	38.8 ± 6.4
15%	**DAPAR**	24.9 ± 12.4	0.7 ± 0.9	799.3 ± 0.9	175.1 ± 12.4	12.5 ± 6.2	99.9 ± 0.1	95.7 ± 14	21.6 ± 9.7	30.1 ± 9.2
	MI4P	17.6 ± 11.7	0 ± 0.2	800 ± 0.2	182.4 ± 11.7	8.8 ± 5.8	100 ± 0	92.9 ± 25.6	15.6 ± 9.8	24.5 ± 11.1
20%	**DAPAR**	23.3 ± 12.4	0.7 ± 1	799.3 ± 1	176.7 ± 12.4	11.6 ± 6.2	99.9 ± 0.1	96.3 ± 10.5	20.2 ± 9.8	28.9 ± 9.2
	MI4P	6.4 ± 6.9	0 ± 0	800 ± 0	193.6 ± 6.9	3.2 ± 3.5	100 ± 0	74 ± 44.1	6 ± 6.3	12.8 ± 9.8
25%	**DAPAR**	24.1 ± 11.8	0.8 ± 1.2	799.2 ± 1.2	175.8 ± 11.8	12.1 ± 5.9	99.9 ± 0.1	97.4 ± 3.5	21 ± 9.3	29.7 ± 8.2
	MI4P	1.7 ± 3.2	0 ± 0	800 ± 0	198.3 ± 3.2	0.9 ± 1.6	100 ± 0	43 ± 49.8	1.7 ± 3	5 ± 6.8

Table A.14: Performance evaluation on the third set of MAR simulation imputed using Bayesian linear regression.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	25.8 ± 10.2	0.5 ± 0.8	799.5 ± 0.8	174.2 ± 10.2	12.9 ± 5.1	99.9 ± 0.1	98.3 ± 2.4	22.4 ± 8	31.4 ± 7
	MI4P	95.7 ± 9.9	3.2 ± 1.8	796.8 ± 1.8	104.3 ± 9.9	47.9 ± 4.9	99.6 ± 0.2	96.8 ± 1.7	63.9 ± 4.4	63.5 ± 3.7
5%	**DAPAR**	24.9 ± 10.4	0.5 ± 0.7	799.5 ± 0.7	175.2 ± 10.4	12.4 ± 5.2	99.9 ± 0.1	98.2 ± 2.5	21.7 ± 8.3	30.6 ± 7.5
	MI4P	97.7 ± 9.5	3 ± 1.8	797 ± 1.8	102.3 ± 9.5	48.8 ± 4.7	99.6 ± 0.2	97 ± 1.7	64.8 ± 4.2	64.4 ± 3.6
10%	**DAPAR**	24.5 ± 10.6	0.6 ± 0.9	799.4 ± 0.9	175.5 ± 10.6	12.3 ± 5.3	99.9 ± 0.1	95.8 ± 14.1	21.4 ± 8.4	30.2 ± 7.9
	MI4P	101.1 ± 9.5	3.2 ± 1.8	796.8 ± 1.8	98.9 ± 9.5	50.6 ± 4.8	99.6 ± 0.2	97 ± 1.6	66.3 ± 4.1	65.6 ± 3.5
15%	**DAPAR**	24.2 ± 12.4	0.7 ± 0.9	799.3 ± 0.9	175.8 ± 12.4	12.1 ± 6.2	99.9 ± 0.1	95.7 ± 14	21 ± 9.7	29.6 ± 9.1
	MI4P	104.6 ± 10.1	3.4 ± 2.1	796.6 ± 2.1	95.4 ± 10.1	52.3 ± 5.1	99.6 ± 0.3	96.9 ± 1.8	67.8 ± 4.3	66.8 ± 3.7
20%	**DAPAR**	23.6 ± 12.2	0.7 ± 0.9	799.3 ± 0.9	176.4 ± 12.2	11.8 ± 6.1	99.9 ± 0.1	94.7 ± 17.1	20.5 ± 9.7	29 ± 9.7
	MI4P	110 ± 10.1	3.7 ± 2.1	796.3 ± 2.1	90 ± 10.1	55 ± 5.1	99.5 ± 0.3	96.8 ± 1.7	70 ± 4.2	68.7 ± 3.6
25%	**DAPAR**	24.7 ± 11.3	0.8 ± 1.2	799.2 ± 1.2	175.3 ± 11.3	12.3 ± 5.7	99.9 ± 0.1	97.2 ± 3.6	21.4 ± 8.9	30.2 ± 7.7
	MI4P	113.6 ± 9.3	4.4 ± 2.3	795.6 ± 2.3	86.4 ± 9.3	56.8 ± 4.6	99.4 ± 0.3	96.3 ± 1.7	71.3 ± 3.6	69.7 ± 3.2

Table A.15: Performance evaluation on the third set of MAR simulation imputed using principal component analysis.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	25.7 ± 10.2	0.5 ± 0.7	799.5 ± 0.7	174.3 ± 10.2	12.8 ± 5.1	99.9 ± 0.1	98.5 ± 2.3	22.3 ± 8	31.3 ± 7
	MI4P	95.8 ± 9.8	3.1 ± 1.9	796.9 ± 1.9	104.2 ± 9.8	47.9 ± 4.9	99.6 ± 0.2	96.9 ± 1.8	63.9 ± 4.4	63.6 ± 3.7
5%	**DAPAR**	25.2 ± 10.5	0.5 ± 0.7	799.5 ± 0.7	174.8 ± 10.5	12.6 ± 5.2	99.9 ± 0.1	98.4 ± 2.5	21.9 ± 8.2	31 ± 7.1
	MI4P	97.7 ± 9.8	3 ± 1.8	797 ± 1.8	102.3 ± 9.8	48.8 ± 4.9	99.6 ± 0.2	97.1 ± 1.7	64.8 ± 4.3	64.4 ± 3.6
10%	**DAPAR**	24.4 ± 11.4	0.5 ± 0.8	799.5 ± 0.8	175.6 ± 11.4	12.2 ± 5.7	99.9 ± 0.1	95.2 ± 17.1	21.2 ± 9.1	29.9 ± 9.1
	MI4P	102.2 ± 9.9	2.9 ± 1.7	797.1 ± 1.7	97.8 ± 9.9	51.1 ± 4.9	99.6 ± 0.2	97.3 ± 1.6	66.9 ± 4.3	66.1 ± 3.7
15%	**DAPAR**	25.4 ± 12.7	0.5 ± 0.8	799.5 ± 0.8	174.6 ± 12.7	12.7 ± 6.3	99.9 ± 0.1	96.4 ± 14.1	21.9 ± 10	30.5 ± 9.5
	MI4P	105.7 ± 10.1	2.7 ± 1.6	797.3 ± 1.6	94.3 ± 10.1	52.8 ± 5.1	99.7 ± 0.2	97.5 ± 1.4	68.4 ± 4.3	67.5 ± 3.7
20%	**DAPAR**	25.1 ± 12.5	0.4 ± 0.7	799.5 ± 0.7	174.9 ± 12.5	12.5 ± 6.3	99.9 ± 0.1	95.6 ± 17.1	21.7 ± 9.8	30.4 ± 9.5
	MI4P	110.8 ± 10.2	3 ± 1.9	797 ± 1.9	89.2 ± 10.2	55.4 ± 5.1	99.6 ± 0.2	97.4 ± 1.5	70.5 ± 4.1	69.3 ± 3.5
25%	**DAPAR**	26.7 ± 12.1	0.7 ± 1	799.3 ± 1	173.3 ± 12.1	13.3 ± 6	99.9 ± 0.1	97.8 ± 3.2	23 ± 9.5	31.6 ± 8.4
	MI4P	113.9 ± 9.8	3.4 ± 2	796.6 ± 2	86.1 ± 9.8	57 ± 4.9	99.6 ± 0.3	97.1 ± 1.6	71.7 ± 3.9	70.2 ± 3.4

Table A.16: Performance evaluation on the third set of MAR simulation imputed using random forests.

A.2.1.b Under Missing Completely At Random and Not At Random assumption

In the following, we provide the detailed results of the evaluation of the performance of the `mi4p` workflow compared to the `DAPAR` workflow on both sets of MCAR + MNAR simulations, using the maximum likelihood estimation method for imputation. Results are expressed as the mean of the given indicator over the 100 simulated datasets ± the mean of the standard deviations of the given indicator over the 100 simulated datasets. Results are based on adjusted p-values using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) with a false discovery rate of 1% and a significance level of 5%.

Table A.17, page 153: Performance evaluation on the first set of MCAR + MNAR simulations imputed using maximum likelihood estimation.

Table A.18, page 154: Performance evaluation on the first set of MCAR + MNAR simulations imputed using maximum likelihood estimation.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	123.7 ± 7.7	490.5 ± 17.4	309.5 ± 17.4	76.3 ± 7.7	61.8 ± 3.8	38.7 ± 2.2	20.1 ± 1.1	30.4 ± 1.6	0.4 ± 3.4
	MI4P	154.8 ± 6.5	617.2 ± 12	182.8 ± 12	45.2 ± 6.5	77.4 ± 3.3	22.8 ± 1.5	20 ± 0.7	31.8 ± 1.2	0.2 ± 3.4
5%	**DAPAR**	125.1 ± 6.6	495.7 ± 17.1	303.9 ± 17	74.9 ± 6.6	62.5 ± 3.3	38 ± 2.1	20.1 ± 0.9	30.5 ± 1.3	0.5 ± 2.8
	MI4P	154.4 ± 5.3	613.1 ± 13.5	186.1 ± 12.8	45.5 ± 5.3	77.2 ± 2.7	23.3 ± 1.6	20.1 ± 0.7	31.9 ± 1	0.5 ± 2.9
10%	**DAPAR**	123.9 ± 6.9	477.8 ± 18.3	312.6 ± 17.6	76.1 ± 6.9	61.9 ± 3.5	39.6 ± 2.2	20.6 ± 1.1	30.9 ± 1.6	1.2 ± 3.3
	MI4P	152.9 ± 6.1	590.3 ± 17.6	193.2 ± 15.6	47.1 ± 6.1	76.5 ± 3.1	24.7 ± 2	20.6 ± 0.8	32.4 ± 1.2	1 ± 3.3
15%	**DAPAR**	124.9 ± 7.2	436 ± 18.6	317.4 ± 18.3	75.1 ± 7.2	62.5 ± 3.6	42.1 ± 2.3	22.3 ± 1.1	32.8 ± 1.7	3.8 ± 3.2
	MI4P	153.3 ± 6	540.4 ± 18.8	201.1 ± 14.8	46.7 ± 6	76.7 ± 3	27.1 ± 1.9	22.1 ± 0.8	34.3 ± 1.2	3.5 ± 3.2
20%	**DAPAR**	124.4 ± 7.6	396.1 ± 19.1	326.7 ± 23.9	75.6 ± 7.6	62.2 ± 3.8	45.2 ± 2.8	23.9 ± 1.1	34.5 ± 1.6	6.1 ± 3.1
	MI4P	153.3 ± 5.9	492.2 ± 21.7	206.6 ± 18.9	46.7 ± 5.9	76.6 ± 2.9	29.5 ± 2.2	23.8 ± 0.9	36.3 ± 1.2	5.7 ± 2.9
25%	**DAPAR**	119.7 ± 7.6	349.3 ± 19.3	324.4 ± 25.3	80.3 ± 7.6	59.8 ± 3.8	48.1 ± 2.9	25.5 ± 1.3	35.8 ± 1.8	6.7 ± 3.6
	MI4P	146.9 ± 6.9	439.9 ± 23.9	200 ± 28.1	53.1 ± 6.9	73.4 ± 3.5	31.1 ± 2.8	25.1 ± 1.1	37.3 ± 1.5	4.2 ± 4.4

Table A.17: Performance evaluation on the first set of MCAR + MNAR simulation imputed using maximum likelihood estimation.

%MV	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
1%	**DAPAR**	306.5 ± 10	6152.2 ± 65.7	3847.8 ± 65.7	193.5 ± 10	61.3 ± 2	38.5 ± 0.7	4.7 ± 0.1	8.8 ± 0.3	-0.1 ± 0.9
	MI4P	384.5 ± 9.5	7711 ± 46.5	2289 ± 46.5	115.5 ± 9.5	76.9 ± 1.9	22.9 ± 0.5	4.7 ± 0.1	8.9 ± 0.2	-0.1 ± 1
5%	**DAPAR**	311.7 ± 11	6133.4 ± 64.2	3847 ± 66.3	188.3 ± 11	62.3 ± 2.2	38.5 ± 0.6	4.8 ± 0.2	9 ± 0.3	0.4 ± 1
	MI4P	384.6 ± 9.2	7631.6 ± 67.8	2336.6 ± 55.5	115.4 ± 9.2	76.9 ± 1.8	23.4 ± 0.6	4.8 ± 0.1	9 ± 0.2	0.2 ± 0.9
10%	**DAPAR**	311.9 ± 10.8	6007.3 ± 91.1	3862.8 ± 91	188.1 ± 10.8	62.4 ± 2.2	39.1 ± 0.9	4.9 ± 0.2	9.1 ± 0.3	0.7 ± 1
	MI4P	384 ± 9.7	7400.3 ± 127.1	2397.3 ± 89.2	116 ± 9.7	76.8 ± 1.9	24.5 ± 0.9	4.9 ± 0.1	9.3 ± 0.2	0.6 ± 0.9
15%	**DAPAR**	315.6 ± 11.5	5566.6 ± 125.6	3903.5 ± 153.6	184.4 ± 11.5	63.1 ± 2.3	41.2 ± 1.3	5.4 ± 0.2	9.9 ± 0.3	1.9 ± 1
	MI4P	384.2 ± 11	6842.1 ± 172.1	2470.2 ± 117.6	115.8 ± 11	76.8 ± 2.2	26.5 ± 1	5.3 ± 0.2	9.9 ± 0.3	1.7 ± 1.1
20%	**DAPAR**	315.6 ± 13.2	5157.3 ± 123.1	3916.9 ± 200.8	184.4 ± 13.2	63.1 ± 2.6	43.1 ± 1.6	5.8 ± 0.2	10.6 ± 0.4	2.8 ± 1.1
	MI4P	384.4 ± 10.7	6312.7 ± 210.3	2493.2 ± 164.8	115.6 ± 10.7	76.9 ± 2.1	28.3 ± 1.2	5.7 ± 0.2	10.7 ± 0.3	2.6 ± 1.1
25%	**DAPAR**	310.1 ± 14.3	4696.8 ± 117.4	3831.4 ± 260.5	189.9 ± 14.3	62 ± 2.9	44.9 ± 1.9	6.2 ± 0.2	11.3 ± 0.5	3.2 ± 1.3
	MI4P	370.2 ± 12.6	5752.6 ± 224.1	2449.7 ± 248.5	129.8 ± 12.6	74 ± 2.5	29.8 ± 1.7	6.1 ± 0.2	11.2 ± 0.3	1.9 ± 1.6

Table A.18: Performance evaluation on the second set of MCAR + MNAR simulation imputed using maximum likelihood estimation.

A.2.2 Evaluation on real datasets

A.2.2.a Evaluation using the Arabidopsis thaliana + UPS1 experiment

This section provides the evaluation of the `mi4p` workflow compared to the `DAPAR` workflow on the the *Arabidopsis thaliana* + UPS1 experiment. Results are based on adjusted p-values using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) and a false discovery rate of 1% and a significance level of 5%. Missing values were imputed using maximum likelihood estimation.

Table A.19, page 156: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, filtered with at least 1 quantified value in each condition.

Table A.20, page 156: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, filtered with at least 1 quantified value in each condition and focusing only on the comparison 5fmol vs. 10fmol.

Table A.21, page 157: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, filtered with at least 2 quantified values in each condition.

Table A.22, page 158: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, extracted without Match Between Runs and filtered with at least 1 quantified value in each condition.

Table A.23, page 159: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, extracted without Match Between Runs and filtered with at least 2 quantified value in each condition.

Table A.24, page 160: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset at the protein-level, filtered with at least 1 quantified values in each condition.

Condition (vs 10fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.05fmol	**DAPAR**	132	3677	10507	5	96.4	74.1	3.5	6.7	15.5
	MI4P	129	2095	12089	8	94.2	85.2	5.8	10.9	21.3
0.25fmol	**DAPAR**	135	3466	10718	2	98.5	75.6	3.7	7.2	16.6
	MI4P	133	1974	12210	4	97.1	86.1	6.3	11.9	22.9
0.5fmol	**DAPAR**	134	2495	11689	3	97.8	82.4	5.1	9.7	20.2
	MI4P	132	1233	12951	5	96.4	91.3	9.7	17.6	29.1
1.25fmol	**DAPAR**	132	2118	12066	5	96.4	85.1	5.9	11.1	21.8
	MI4P	129	792	13392	8	94.2	94.4	14	24.4	35.1
2.5fmol	**DAPAR**	125	473	13711	12	91.2	96.7	20.9	34	42.8
	MI4P	93	145	14039	44	67.9	99	39.1	49.6	50.9
5fmol	**DAPAR**	122	1100	13084	15	89.1	92.2	10	18	28.3
	MI4P	85	383	13801	52	62	97.3	18.2	28.1	32.5

Table A.19: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, filtered with at least 1 quantified value in each condition.

Condition (vs 10fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
5fmol	**DAPAR**	372	226	15522	196	65.5	98.6	62.2	63.8	62.5
	MI4P	348	179	15569	220	61.3	98.9	66	63.6	62.3

Table A.20: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, filtered with at least 1 quantified value in each condition and focusing only on the comparison 5fmol vs. 10fmol.

Condition (vs 10fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.05fmol	**DAPAR**	74	2989	8880	3	96.1	74.8	2.4	4.7	13
	MI4P	74	2989	8880	3	96.1	74.8	2.4	4.7	13
0.25fmol	**DAPAR**	76	2837	9032	1	98.7	76.1	2.6	5.1	13.9
	MI4P	76	2837	9032	1	98.7	76.1	2.6	5.1	13.9
0.5fmol	**DAPAR**	76	1905	9964	1	98.7	83.9	3.8	7.4	17.8
	MI4P	76	1905	9964	1	98.7	83.9	3.8	7.4	17.8
1.25fmol	**DAPAR**	75	1411	10458	2	97.4	88.1	5	9.6	20.7
	MI4P	75	1411	10458	2	97.4	88.1	5	9.6	20.7
2.5fmol	**DAPAR**	70	232	11637	7	90.9	98	23.2	36.9	45.3
	MI4P	70	232	11637	7	90.9	98	23.2	36.9	45.3
5fmol	**DAPAR**	67	686	11183	10	87	94.2	8.9	16.1	26.7
	MI4P	67	686	11183	10	87	94.2	8.9	16.1	26.7

Table A.21: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, filtered with at least 2 quantified values in each condition.

Condition (vs 10fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.05fmol	**DAPAR**	16	1567	6173	1	94.1	79.8	1	2	8.6
	MI4P	16	1567	6173	1	94.1	79.8	1	2	8.6
0.25fmol	**DAPAR**	16	1461	6279	1	94.1	81.1	1.1	2.1	9
	MI4P	16	1461	6279	1	94.1	81.1	1.1	2.1	9
0.5fmol	**DAPAR**	15	895	6845	2	88.2	88.4	1.6	3.2	11.1
	MI4P	15	895	6845	2	88.2	88.4	1.6	3.2	11.1
1.25fmol	**DAPAR**	16	880	6860	1	94.1	88.6	1.8	3.5	12.1
	MI4P	16	880	6860	1	94.1	88.6	1.8	3.5	12.1
2.5fmol	**DAPAR**	13	139	7601	4	76.5	98.2	8.6	15.4	25.2
	MI4P	13	139	7601	4	76.5	98.2	8.6	15.4	25.2
5fmol	**DAPAR**	11	419	7321	6	64.7	94.6	2.6	4.9	12.1
	MI4P	11	419	7321	6	64.7	94.6	2.6	4.9	12.1

Table A.22: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, extracted without Match Between Runs and filtered with at least 1 quantified value in each condition.

Condition (vs 10fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.05fmol	**DAPAR**	8	1234	4119	1	88.9	76.9	0.6	1.3	6.4
	MI4P	8	1234	4119	1	88.9	76.9	0.6	1.3	6.4
0.25fmol	**DAPAR**	8	1150	4203	1	88.9	78.5	0.7	1.4	6.7
	MI4P	8	1150	4203	1	88.9	78.5	0.7	1.4	6.7
0.5fmol	**DAPAR**	8	742	4611	1	88.9	86.1	1.1	2.1	8.9
	MI4P	8	742	4611	1	88.9	86.1	1.1	2.1	8.9
1.25fmol	**DAPAR**	8	536	4817	1	88.9	90	1.5	2.9	10.7
	MI4P	8	536	4817	1	88.9	90	1.5	2.9	10.7
2.5fmol	**DAPAR**	6	83	5270	3	66.7	98.4	6.7	12.2	20.9
	MI4P	6	83	5270	3	66.7	98.4	6.7	12.2	20.9
5fmol	**DAPAR**	6	274	5079	3	66.7	94.9	2.1	4.2	11.3
	MI4P	6	274	5079	3	66.7	94.9	2.1	4.2	11.3

Table A.23: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset, extracted without Match Between Runs and filtered with at least 2 quantified values in each condition.

Condition (vs 10fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.05fmol	**DAPAR**	41	1040	1557	0	100	60	3.8	7.3	15.1
	MI4P	41	753	1844	0	100	71	5.2	9.8	19.1
0.25fmol	**DAPAR**	41	1072	1525	0	100	58.7	3.7	7.1	14.7
	MI4P	41	797	1800	0	100	69.3	4.9	9.3	18.4
0.5fmol	**DAPAR**	40	848	1749	1	97.6	67.3	4.5	8.6	17
	MI4P	40	585	2012	1	97.6	77.5	6.4	12	21.8
1.25fmol	**DAPAR**	41	409	2188	0	100	84.3	9.1	16.7	27.7
	MI4P	41	142	2455	0	100	94.5	22.4	36.6	46
2.5fmol	**DAPAR**	41	208	2389	0	100	92	16.5	28.3	38.9
	MI4P	40	69	2528	1	97.6	97.3	36.7	53.3	59
5fmol	**DAPAR**	41	475	2122	0	100	81.7	7.9	14.7	25.5
	MI4P	37	203	2394	4	90.2	92.2	15.4	26.3	35.5

Table A.24: Performance evaluation on the *Arabidopsis thaliana* + UPS1 dataset at the protein-level, filtered with at least 1 quantified values in each condition.

A.2.2.b Evaluation using the Saccharomyces cerevisiae + UPS1 experiment

This section provides the evaluation of the `mi4p` workflow compared to the `DAPAR` workflow on the the *Saccharomyces cerevisiae* + UPS1 experiment. Results are based on adjusted p-values using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) and a false discovery rate of 1% and a significance level of 5%. Missing values were imputed using maximum likelihood estimation.

Table A.25, page 162: Performance evaluation on the *Saccharomyces cerevisiae* + UPS1 dataset, filtered with at least 1 quantified value in each condition.

Table A.26, page 163: Performance evaluation on the *Saccharomyces cerevisiae* + UPS1 dataset, filtered with at least 2 quantified values in each condition.

Table A.27, page 164: Performance evaluation on the *Saccharomyces cerevisiae* + UPS1 dataset, at the protein-level and filtered with at least 1 quantified values in each condition.

Condition (vs 25fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.5fmol	**DAPAR**	188	439	18067	4	97.9	97.6	30	45.9	53.5
	MI4P	183	144	18362	9	95.3	99.2	56	70.5	72.7
1fmol	**DAPAR**	186	246	18260	6	96.9	98.7	43.1	59.6	64.1
	MI4P	183	71	18435	9	95.3	99.6	72	82.1	82.7
2.5fmol	**DAPAR**	185	161	18345	7	96.4	99.1	53.5	68.8	71.4
	MI4P	179	39	18467	13	93.2	99.8	82.1	87.3	87.4
5fmol	**DAPAR**	182	108	18398	10	94.8	99.4	62.8	75.5	76.9
	MI4P	156	23	18483	36	81.2	99.9	87.2	84.1	84
10fmol	**DAPAR**	148	109	18397	44	77.1	99.4	57.6	65.9	66.2
	MI4P	86	27	18479	106	44.8	99.9	76.1	56.4	58.1

Table A.25: Performance evaluation on the *Saccharomyces cerevisiae* + UPS1 dataset, filtered with at least 1 quantified value in each condition.

Condition (vs 25fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.5fmol	**DAPAR**	131	146	16316	4	97	99.1	47.3	63.6	67.4
	MI4P	131	146	16316	4	97	99.1	47.3	63.6	67.4
1fmol	**DAPAR**	130	59	16403	5	96.3	99.6	68.8	80.2	81.2
	MI4P	130	59	16403	5	96.3	99.6	68.8	80.2	81.2
2.5fmol	**DAPAR**	130	30	16432	5	96.3	99.8	81.2	88.1	88.4
	MI4P	130	30	16432	5	96.3	99.8	81.2	88.1	88.4
5fmol	**DAPAR**	127	19	16443	8	94.1	99.9	87	90.4	90.4
	MI4P	127	19	16443	8	94.1	99.9	87	90.4	90.4
10fmol	**DAPAR**	96	18	16444	39	71.1	99.9	84.2	77.1	77.2
	MI4P	96	18	16444	39	71.1	99.9	84.2	77.1	77.2

Table A.26: Performance evaluation on the *Saccharomyces cerevisiae* + UPS1 dataset, filtered with at least 2 quantified values in each condition.

Condition (vs 25fmol)	Method	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Specificity (%)	Precision (%)	F-score (%)	MCC (%)
0.5fmol	**DAPAR**	42	90	2285	0	100	96.2	31.8	48.3	55.3
	MI4P	42	24	2351	0	100	99	63.6	77.8	79.4
1fmol	**DAPAR**	42	65	2310	0	100	97.3	39.3	56.4	61.8
	MI4P	41	13	2362	1	97.6	99.5	75.9	85.4	85.8
2.5fmol	**DAPAR**	41	27	2348	1	97.6	98.9	60.3	74.5	76.2
	MI4P	41	8	2367	1	97.6	99.7	83.7	90.1	90.2
5fmol	**DAPAR**	42	19	2356	0	100	99.2	68.9	81.6	82.6
	MI4P	41	7	2368	1	97.6	99.7	85.4	91.1	91.2
10fmol	**DAPAR**	39	23	2352	3	92.9	99	62.9	75	75.9
	MI4P	38	7	2368	4	90.5	99.7	84.4	87.4	87.2

Table A.27: Performance evaluation on the *Saccharomyces cerevisiae* + UPS1 dataset, at the protein-level and filtered with at least 1 quantified values in each condition.

Bibliography

R. Aebersold, J. N. Agar, I. J. Amster, M. S. Baker, C. R. Bertozzi, E. S. Boja, C. E. Costello, B. F. Cravatt, C. Fenselau, B. A. Garcia, Y. Ge, J. Gunawardena, R. C. Hendrickson, P. J. Hergenrother, C. G. Huber, A. R. Ivanov, O. N. Jensen, M. C. Jewett, N. L. Kelleher, L. L. Kiessling, N. J. Krogan, M. R. Larsen, J. A. Loo, R. R. Ogorzalek Loo, E. Lundberg, M. J. MacCoss, P. Mallick, V. K. Mootha, M. Mrksich, T. W. Muir, S. M. Patrie, J. J. Pesavento, S. J. Pitteri, H. Rodriguez, A. Saghatelian, W. Sandoval, H. Schlüter, S. Sechi, S. A. Slavoff, L. M. Smith, M. P. Snyder, P. M. Thomas, M. Uhlén, J. E. Van Eyk, M. Vidal, D. R. Walt, F. M. White, E. R. Williams, T. Wohlschlager, V. H. Wysocki, N. A. Yates, N. L. Young, and B. Zhang. How many human proteoforms are there? *Nature Chemical Biology*, 14(3):206–214, Mar. 2018. ISSN 1552-4469. doi: 10.1038/nchembio.2576. [Cited on page 4.]

N. L. Anderson and N. G. Anderson. Proteome and proteomics: New technologies, new concepts, and new words. *ELECTROPHORESIS*, 19(11):1853–1861, 1998. ISSN 1522-2683. doi: 10.1002/elps.1150191103. [Cited on page 3.]

T. Anderson. *An Introduction to Multivariate Statistical Analysis, 3rd Edition*. 2003. [Cited on page 27.]

J. A. Ankney, A. Muneer, and X. Chen. Relative and Absolute Quantitation in Mass Spectrometry-Based Proteomics. *Annual Review of Analytical Chemistry (Palo Alto, Calif.)*, 11(1):49–77, June 2018. ISSN 1936-1335. doi: 10.1146/annurev-anchem-061516-045357. [Cited on pages 9 and 11.]

Y. Bai, J.-Y. Kim, J. M. Watters, B. Fang, F. Kinose, L. Song, J. M. Koomen, J. K. Teer, K. Fisher, Y. A. Chen, U. Rix, and E. B. Haura. Adaptive responses to dasatinib-treated lung squamous cell cancer cells harboring DDR2 mutations. *Cancer Research*, 74(24):7217–7228, Dec. 2014. ISSN 1538-7445. doi: 10.1158/0008-5472.CAN-14-0505. [Cited on page 23.]

P. Baldi and A. D. Long. A Bayesian framework for the analysis of microarray expression data: Regularized t -test and statistical inferences of gene changes. *Bioinformatics*, 17

(6):509–519, June 2001. ISSN 1367-4803. doi: 10.1093/bioinformatics/17.6.509. [Cited on page 14.]

M. Bantscheff, S. Lemeer, M. M. Savitski, and B. Kuster. Quantitative mass spectrometry in proteomics: Critical review update from 2007 to the present. *Analytical and Bioanalytical Chemistry*, 404(4):939–965, Sept. 2012. ISSN 1618-2650. doi: 10.1007/s00216-012-6203-4. [Cited on page 9.]

R. E. Barlow, D. J. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression*. J. Wiley, 1972. ISBN 978-0-471-04970-8. [Cited on page 28.]

J. Bazile, B. Picard, C. Chambon, A. Valais, and M. Bonnet. Pathways and biomarkers of marbling and carcass fat deposition in bovine revealed by a combination of gel-based and gel-free proteomic analyses. *Meat Science*, 156:146–155, Oct. 2019. ISSN 1873-4138. doi: 10.1016/j.meatsci.2019.05.018. [Cited on pages 43 and 56.]

I. Belouah, M. Blein-Nicolas, T. Balliau, Y. Gibon, M. Zivy, and S. Colombié. Peptide filtering differently affects the performances of XIC-based quantification methods. *Journal of Proteomics*, 193:131–141, Feb. 2019. ISSN 1874-3919. doi: 10.1016/j.jprot.2018.10.003. [Cited on page 9.]

Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. ISSN 0035-9246. [Cited on pages 8, 73, 134, 140, 146, 152, 155, and 161.]

R. A. Betensky. The p-Value Requires Context, Not a Threshold. *The American Statistician*, 73(sup1):115–117, Mar. 2019. ISSN 0003-1305. doi: 10.1080/00031305.2018.1529624. [Cited on page 128.]

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Aug. 2006. ISBN 978-0-387-31073-2. [Cited on pages 27 and 63.]

W. P. Blackstock and M. P. Weir. Proteomics: Quantitative and physical mapping of cellular proteins. *Trends in Biotechnology*, 17(3):121–127, Mar. 1999. ISSN 0167-7799. doi: 10.1016/S0167-7799(98)01245-1. [Cited on page 4.]

M. Blein-Nicolas and M. Zivy. Thousand and one ways to quantify and compare protein abundances in label-free bottom-up proteomics. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, 1864(8):883–895, Aug. 2016. ISSN 1570-9639. doi: 10.1016/j.bbapap.2016.02.019. [Cited on page 9.]

M. Blein-Nicolas, H. Xu, D. de Vienne, C. Giraud, S. Huet, and M. Zivy. Including shared peptides for estimating protein abundances: A significant improvement for quantitative proteomics. *PROTEOMICS*, 12(18):2797–2801, 2012. ISSN 1615-9861. doi: 10.1002/pmic.201100660. [Cited on page 9.]

B. Bolstad. preprocessCore: A collection of pre-processing functions. Bioconductor version: Release (3.13), 2021. [Cited on pages 89 and 104.]

B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics (Oxford, England)*, 19(2):185–193, Jan. 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/19.2.185. [Cited on page 89.]

M. Bonnet, J. Soulat, J. Bons, S. Léger, L. De Koning, C. Carapito, and B. Picard. Quantification of biomarkers for beef meat qualities using a combination of Parallel Reaction Monitoring- and antibody-based proteomics. *Food Chemistry*, 317:126376, July 2020. ISSN 0308-8146. doi: 10.1016/j.foodchem.2020.126376. [Cited on pages iv, 31, 43, 44, and 46.]

J. Bons, G. Husson, M. Chion, M. Bonnet, M. Maumy-Bertrand, F. Delalande, S. Cianférani, F. Bertrand, B. Picard, and C. Carapito. Combining label-free and label-based accurate quantifications with SWATH-MS: Comparison with SRM and PRM for the evaluation of bovine muscle type effects. *PROTEOMICS*, 21(10):2000214, 2021. ISSN 1615-9861. doi: 10.1002/pmic.202000214. [Cited on pages 13 and 43.]

D. Bouyssié, A.-M. Hesse, E. Mouton-Barbosa, M. Rompais, C. Macron, C. Carapito, A. Gonzalez de Peredo, Y. Couté, V. Dupierris, A. Burel, J.-P. Menetrey, A. Kalaitzakis, J. Poisat, A. Romdhani, O. Burlet-Schiltz, S. Cianférani, J. Garin, and C. Bruley. Proline: An efficient and user-friendly software suite for large-scale proteomics. *Bioinformatics*, 36 (10):3148–3155, May 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa118. [Cited on page 46.]

K. A. Brown, J. A. Melby, D. S. Roberts, and Y. Ge. Top-down proteomics: Challenges, innovations, and applications in basic and clinical research. *Expert Review of Proteomics*, 17(10):719–733, Oct. 2020. ISSN 1478-9450. doi: 10.1080/14789450.2020.1855982. [Cited on page 5.]

T. Burger. Gentle Introduction to the Statistical Foundations of False Discovery Rate in Quantitative Proteomics. *Journal of Proteome Research*, 17(1):12–22, Jan. 2018. ISSN 1535-3893. doi: 10.1021/acs.jproteome.7b00170. [Cited on page 8.]

S. Cappadona, P. R. Baker, P. R. Cutillas, A. J. R. Heck, and B. van Breukelen. Current challenges in software solutions for mass spectrometry-based quantitative proteomics. *Amino*

Acids, 43(3):1087–1108, Sept. 2012. ISSN 1438-2199. doi: 10.1007/s00726-012-1289-8. [Cited on page 9.]

C. Carapito, A. Burel, P. Guterl, A. Walter, F. Varrier, F. Bertile, and A. Van Dorsselaer. MSDA, a proteomics software suite for in-depth Mass Spectrometry Data Analysis using grid computing. *Proteomics*, 14(9):1014–1019, May 2014. ISSN 1615-9861. doi: 10.1002/pmic.201300415. [Cited on page 46.]

R. Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, July 1997. ISSN 1573-0565. doi: 10.1023/A:1007379606734. [Cited on page 58.]

G. Casella. An Introduction to Empirical Bayes Data Analysis. *The American Statistician*, 39(2):83–87, 1985. ISSN 0003-1305. doi: 10.2307/2682801. [Cited on page 14.]

C. Chang, K. Xu, C. Guo, J. Wang, Q. Yan, J. Zhang, F. He, and Y. Zhu. PANDA-view: An easy-to-use tool for statistical analysis and visualization of quantitative proteomics data. *Bioinformatics*, 34(20):3594–3596, Oct. 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty408. [Cited on pages vii, 25, and 33.]

D. Chen and R. J. Plemmons. Nonnegativity constraints in numerical analysis. In *In A. Bultheel and R. Cools (Eds.), Symposium on the Birth of Numerical Analysis, World Scientific*. Press, 2009. [Cited on page 29.]

L. S. Chen, R. L. Prentice, and P. Wang. A penalized EM algorithm incorporating missing data mechanism for Gaussian parameter estimation. *Biometrics*, 70(2):312–322, 2014. ISSN 1541-0420. doi: 10.1111/biom.12149. [Cited on page 26.]

D. Chicco and G. Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, Jan. 2020. ISSN 1471-2164. doi: 10.1186/s12864-019-6413-7. [Cited on page 81.]

M. Chion, C. Carapito, and F. Bertrand. Accounting for multiple imputation-induced variability for differential analysis in mass spectrometry-based label-free quantitative proteomics. *arXiv:2108.07086 [q-bio, stat]*, Aug. 2021a. [Cited on page 67.]

M. Chion, C. Carapito, F. Bertrand, G. Smyth, D. McCarthy, H. Borges, T. Burger, Q. Giai-Gianetto, and S. Wieczorek. Mi4p: Multiple Imputation for Proteomics, Aug. 2021b. [Cited on page 95.]

M. Chion, C. Carapito, and F. Bertrand. Towards a more accurate differential analysis of multiple imputed proteomics data with mi4limma. In *Statistical Analysis of Proteomic Data*. Springer US, 2022. ISBN 978-1-07-161966-7. [Cited on page 95.]

M. Choi, C.-Y. Chang, T. Clough, D. Broudy, T. Killeen, B. MacLean, and O. Vitek. MSstats: An R package for statistical analysis of quantitative mass spectrometry-based

proteomic experiments. *Bioinformatics*, 30(17):2524–2526, Sept. 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu305. [Cited on pages vii, 23, 24, 25, and 33.]

M. Cobb. 60 years ago, Francis Crick changed the logic of biology. *PLOS Biology*, 15(9): e2003243, Sept. 2017. ISSN 1545-7885. doi: 10.1371/journal.pbio.2003243. [Cited on page 3.]

Y. Couté, C. Bruley, and T. Burger. Beyond Target–Decoy Competition: Stable Validation of Peptide and Protein Identifications in Mass Spectrometry-Based Discovery Proteomics. *Analytical Chemistry*, 92(22):14898–14906, Nov. 2020. ISSN 0003-2700. doi: 10.1021/acs.analchem.0c00328. [Cited on page 8.]

S. Couvreur, G. Le Bec, D. Micol, and B. Picard. Relationships Between Cull Beef Cow Characteristics, Finishing Practices and Meat Quality Traits of Longissimus thoracis and Rectus abdominis. *Foods*, 8(4):141, Apr. 2019. doi: 10.3390/foods8040141. [Cited on page 43.]

J. Cox, N. Neuhauser, A. Michalski, R. A. Scheltema, J. V. Olsen, and M. Mann. Andromeda: A Peptide Search Engine Integrated into the MaxQuant Environment. *Journal of Proteome Research*, 10(4):1794–1805, Apr. 2011. ISSN 1535-3893. doi: 10.1021/pr101065j. [Cited on page 7.]

R. Craig and R. C. Beavis. TANDEM: Matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, June 2004. ISSN 1367-4803. doi: 10.1093/bioinformatics/bth092. [Cited on page 7.]

H. B. Curry and I. J. Schoenberg. On Pólya frequency functions IV: The fundamental spline functions and their limits. *Journal d'Analyse Mathématique*, 17(1):71–107, Dec. 1966. ISSN 1565-8538. doi: 10.1007/BF02788653. [Cited on page 29.]

C. de Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer-Verlag, New York, 1978. ISBN 978-0-387-95366-3. [Cited on page 28.]

F. de Mendiburu. Agricolae: Statistical Procedures for Agricultural Research, June 2021. [Cited on page 49.]

V. Demichev, C. B. Messner, S. I. Vernardis, K. S. Lilley, and M. Ralser. DIA-NN: Neural networks and interference correction enable deep proteome coverage in high throughput. *Nature Methods*, 17(1):41–44, Jan. 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0638-x. [Cited on page 14.]

F. Desiere, E. W. Deutsch, N. L. King, A. I. Nesvizhskii, P. Mallick, J. Eng, S. Chen, J. Eddes, S. N. Loevenich, and R. Aebersold. The PeptideAtlas project. *Nucleic Acids Research*, 34(suppl_1):D655–D658, Jan. 2006. ISSN 0305-1048. doi: 10.1093/nar/gkj040. [Cited on page 13.]

A. Doerr. DIA mass spectrometry. *Nature Methods*, 12(1):35–35, Jan. 2015. ISSN 1548-7105. doi: 10.1038/nmeth.3234. [Cited on page 12.]

F. Dominici, G. Parmigiani, and M. Clyde. Conjugate Analysis of Multivariate Normal Data with Incomplete Observations. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 28(3):533–550, 2000. ISSN 0319-5724. doi: 10.2307/3315963. [Cited on page 109.]

E. J. Dupree, M. Jayathirtha, H. Yorkey, M. Mihasan, B. A. Petre, and C. C. Darie. A Critical Review of Bottom-Up Proteomics: The Good, the Bad, and the Future of This Field. *Proteomes*, 8(3):14, Sept. 2020. doi: 10.3390/proteomes8030014. [Cited on page 5.]

D. Duvenaud. *Automatic Model Construction with Gaussian Processes*. Thesis, University of Cambridge, Nov. 2014. [Cited on page 58.]

B. Efron and C. Morris. Limiting the Risk of Bayes and Empirical Bayes Estimators–Part I: The Bayes Case. *Journal of the American Statistical Association*, 66(336):807–815, 1971. ISSN 0162-1459. doi: 10.2307/2284231. [Cited on page 14.]

N. El Faouzi and Y. Escoufier. Modélisation I-spline et comparaison de courbes de croissance. *Revue de Statistique Appliquée*, 39(1):51–64, 1991. [Cited on page 28.]

J. E. Elias and S. P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4(3):207–214, Mar. 2007. ISSN 1548-7105. doi: 10.1038/nmeth1019. [Cited on page 8.]

J. K. Eng, M. R. Hoopmann, T. A. Jahan, J. D. Egertson, W. S. Noble, and M. J. MacCoss. A Deeper Look into Comet—Implementation and Features. *Journal of The American Society for Mass Spectrometry*, 26(11):1865–1874, Nov. 2015. ISSN 1879-1123. doi: 10.1007/s13361-015-1179-x. [Cited on page 7.]

B. Fang, M. A. Hoffman, A.-S. Mirza, K. M. Mishall, J. Li, S. M. Peterman, K. S. M. Smalley, K. H. Shain, P. M. Weinberger, J. Wu, U. Rix, E. B. Haura, and J. M. Koomen. Evaluating kinase ATP uptake and tyrosine phosphorylation using multiplexed quantification of chemically labeled and post-translationally modified peptides. *Methods (San Diego, Calif.)*, 81:41–49, June 2015. ISSN 1095-9130. doi: 10.1016/j.ymeth.2015.03.006. [Cited on page 23.]

J. Friedman and R. Tibshirani. The Monotone Smoothing of Scatterplots. *Technometrics*, 26(3):243–250, Aug. 1984. ISSN 0040-1706. doi: 10.1080/00401706.1984.10487961. [Cited on page 28.]

M. Gagaoua, M. Bonnet, and B. Picard. Protein Array-Based Approach to Evaluate Biomarkers of Beef Tenderness and Marbling in Cows: Understanding of the Underly-

ing Mechanisms and Prediction. *Foods*, 9(9):1180, Sept. 2020. doi: 10.3390/foods9091180. [Cited on page 43.]

M. Gagaoua, E. M. C. Terlouw, A. M. Mullen, D. Franco, R. D. Warner, J. M. Lorenzo, P. P. Purslow, D. Gerrard, D. L. Hopkins, D. Troy, and B. Picard. Molecular signatures of beef tenderness: Underlying mechanisms based on integromics of protein biomarkers from multi-platform proteomics studies. *Meat Science*, 172:108311, Feb. 2021. ISSN 0309-1740. doi: 10.1016/j.meatsci.2020.108311. [Cited on page 43.]

M. L. Gardner and M. A. Freitas. Multiple Imputation Approaches Applied to the Missing Value Problem in Bottom-Up Proteomics. *International Journal of Molecular Sciences*, 22(17):9650, Jan. 2021. doi: 10.3390/ijms22179650. [Cited on page 93.]

A. Gelman. *Bayesian Data Analysis 3rd Ed*, volume 1542. 2015. ISBN 978-85-7811-079-6. doi: 10.1017/CBO9781107415324.004. [Cited on page 27.]

S. Gerster, T. Kwon, C. Ludwig, M. Matondo, C. Vogel, E. M. Marcotte, R. Aebersold, and P. Bühlmann. Statistical Approach to Protein Quantification *. *Molecular & Cellular Proteomics*, 13(2):666–677, Feb. 2014. ISSN 1535-9476, 1535-9484. doi: 10.1074/mcp.M112.025445. [Cited on page 9.]

S. Gessulat, T. Schmidt, D. P. Zolg, P. Samaras, K. Schnatbaum, J. Zerweck, T. Knaute, J. Rechenberger, B. Delanghe, A. Huhmer, U. Reimer, H.-C. Ehrlich, S. Aiche, B. Kuster, and M. Wilhelm. Prosit: Proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, 16(6):509–518, June 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0426-7. [Cited on page 14.]

Q. Giai Gianetto. Imp4p: Imputation for Proteomics, Mar. 2021. [Cited on pages ix, 35, 68, 69, 85, and 97.]

Q. Giai Gianetto, F. Combes, C. Ramus, C. Bruley, Y. Couté, and T. Burger. Calibration plot for proteomics: A graphical tool to visually check the assumptions underlying FDR control in quantitative experiments. *PROTEOMICS*, 16(1):29–32, 2016. ISSN 1615-9861. doi: 10.1002/pmic.201500189. [Cited on pages 73 and 107.]

Q. Giai Gianetto, S. Wieczorek, Y. Couté, and T. Burger. A peptide-level multiple imputation strategy accounting for the different natures of missing values in proteomics data. *bioRxiv*, page 2020.05.29.122770, May 2020. doi: 10.1101/2020.05.29.122770. [Cited on pages vii, ix, 23, 25, 33, 35, 85, and 93.]

L. C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner, and R. Aebersold. Targeted Data Extraction of the MS/MS Spectra Generated by Data-independent Acquisition: A New Concept for Consistent and Accurate Proteome Analysis

*. *Molecular & Cellular Proteomics*, 11(6), June 2012. ISSN 1535-9476, 1535-9484. doi: 10.1074/mcp.O111.016717. [Cited on pages iv, 12, and 30.]

L. C. Gillet, A. Leitner, and R. Aebersold. Mass Spectrometry Applied to Bottom-Up Proteomics: Entering the High-Throughput Era for Hypothesis Testing. *Annual Review of Analytical Chemistry*, 9(1):449–472, 2016. doi: 10.1146/annurev-anchem-071015-041535. [Cited on page 5.]

L. J. Goeminne, K. Gevaert, and L. Clement. Experimental design and data-analysis in label-free quantitative LC/MS proteomics: A tutorial with MSqRob. *Journal of Proteomics*, 2018. ISSN 18767737. doi: 10.1016/j.jprot.2017.04.004. [Cited on page 25.]

L. J. E. Goeminne, A. Sticker, L. Martens, K. Gevaert, and L. Clement. MSqRob Takes the Missing Hurdle: Uniting Intensity- and Count-Based Proteomics. *Analytical Chemistry*, 92(9):6278–6287, May 2020. ISSN 0003-2700, 1520-6882. doi: 10.1021/acs.analchem.9b04375. [Cited on pages 23 and 25.]

P. Hall and L.-S. Huang. Nonparametric kernel regression subject to monotonicity constraints. *The Annals of Statistics*, 29(3):624–647, June 2001. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1009210683. [Cited on page 28.]

T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein. Imputing Missing Data for Gene Expression Arrays. *Technical report, Stanford Statistics Department*, 1, Dec. 2001. [Cited on pages 68 and 69.]

T. Hastie, R. Tibshirani, B. Narasimhan, and G. Chu. Impute: Imputation for microarray data. Bioconductor version: Release (3.13), 2021. [Cited on pages 68, 69, and 97.]

B. He, J. Shi, X. Wang, H. Jiang, and H.-J. Zhu. Label-free absolute protein quantification with data-independent acquisition. *Journal of Proteomics*, 200:51–59, May 2019. ISSN 1874-3919. doi: 10.1016/j.jprot.2019.03.005. [Cited on page 28.]

D. M. Herrington, C. Mao, S. J. Parker, Z. Fu, G. Yu, L. Chen, V. Venkatraman, Y. Fu, Y. Wang, T. D. Howard, G. Jun, C. F. Zhao, Y. Liu, G. Saylor, W. R. Spivia, G. B. Athas, D. Troxclair, J. E. Hixson, R. S. Vander Heide, Y. Wang, and J. E. Van Eyk. Proteomic Architecture of Human Coronary and Aortic Atherosclerosis. *Circulation*, 137 (25):2741–2756, June 2018. ISSN 1524-4539. doi: 10.1161/CIRCULATIONAHA.118.034365. [Cited on page 23.]

F. Husson and J. Josse. Handling missing values in exploratory multivariate data analysis methods. *Journal de la SFdS*, 153:79–99, Jan. 2012. [Cited on page 69.]

A. Imbert and N. Vialaneix. Décrire, prendre en compte, imputer et évaluer les valeurs manquantes dans les études statistiques : une revue des approches existantes. *Journal de la Société Française de Statistique*, 159(2):1–55, Sept. 2018. [Cited on pages 16 and 18.]

L. Jacob, F. Combes, and T. Burger. PEPA test: Fast and powerful differential analysis from relative quantitative proteomics data using shared peptides. *Biostatistics*, 20(4):632–647, Oct. 2019. ISSN 1465-4644. doi: 10.1093/biostatistics/kxy021. [Cited on page 9.]

P. James. Protein identification in the post-genome era: the rapid rise of proteomics. *Quarterly Reviews of Biophysics*, 30(4):279–331, Nov. 1997. ISSN 1469-8994, 0033-5835. doi: 10.1017/S0033583597003399. [Cited on page 4.]

L. Jin, Y. Bi, C. Hu, J. Qu, S. Shen, X. Wang, and Y. Tian. A comparative study of evaluating missing value imputation methods in label-free proteomics. *Scientific Reports*, 11(1):1760, Dec. 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-81279-4. [Cited on page 23.]

Y. Karpievitch, J. Stanley, T. Taverner, J. Huang, J. N. Adkins, C. Ansong, F. Heffron, T. O. Metz, W. J. Qian, H. Yoon, R. D. Smith, and A. R. Dabney. A statistical framework for protein quantitation in bottom-up MS-based proteomics. *Bioinformatics*, 25(16): 2028–2034, Aug. 2009. ISSN 13674803. doi: 10.1093/bioinformatics/btp362. [Cited on page 9.]

Y. V. Karpievitch, A. R. Dabney, and R. D. Smith. Normalization and missing value imputation for label-free LC-MS analysis. *BMC Bioinformatics*, 13(16):S5, Nov. 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-S16-S5. [Cited on pages 23 and 75.]

N. Kaspric, B. Picard, M. Reichstadt, J. Tournayre, and M. Bonnet. ProteINSIDE to Easily Investigate Proteomics Data from Ruminants: Application to Mine Proteome of Adipose and Muscle Tissues in Bovine Foetuses. *PLOS ONE*, 10(5):e0128086, May 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0128086. [Cited on page 57.]

C. Kelly and J. Rice. Monotone Smoothing with Application to Dose-Response Curves and the Assessment of Synergism. *Biometrics*, 46(4):1071–1085, 1990. ISSN 0006-341X. doi: 10.2307/2532449. [Cited on page 28.]

A. Kent, M. M. Berry, F. U. Luehrs, and J. W. Perry. Machine literature searching VIII. Operational criteria for designing information retrieval systems. *American Documentation*, 6(2):93–101, 1955. ISSN 1936-6108. doi: 10.1002/asi.5090060209. [Cited on page 81.]

S. Kim and P. A. Pevzner. MS-GF+ makes progress towards a universal database search tool for proteomics. *Nature Communications*, 5(1):5277, Oct. 2014. ISSN 2041-1723. doi: 10.1038/ncomms6277. [Cited on page 7.]

J. A. Kirwan, R. J. M. Weber, D. I. Broadhurst, and M. R. Viant. Direct infusion mass spectrometry metabolomics dataset: A benchmark for data processing and quality control. *Scientific Data*, 1(1):140012, June 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.12. [Cited on page 23.]

J. K. Kruschke. Bayesian estimation supersedes the t test. *Journal of Experimental Psychology. General*, 142(2):573–603, May 2013. ISSN 1939-2222. doi: 10.1037/a0029146. [Cited on page 109.]

J. K. Kruschke and T. M. Liddell. The Bayesian New Statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a Bayesian perspective. *Psychonomic Bulletin & Review*, 25(1):178–206, Feb. 2018. ISSN 1531-5320. doi: 10.3758/s13423-016-1221-4. [Cited on page 115.]

C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 1995. ISBN 978-0-89871-356-5. doi: 10.1137/1.9781611971217. [Cited on pages v and 31.]

C. Lazar, L. Gatto, M. Ferro, C. Bruley, and T. Burger. Accounting for the Multiple Natures of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation Strategies. *Journal of Proteome Research*, 15(4):1116–1125, Apr. 2016. ISSN 1535-3893. doi: 10.1021/acs.jproteome.5b00981. [Cited on pages ix, 23, 35, and 75.]

H. Y. Lee, E. G. Kim, H. R. Jung, J. W. Jung, H. B. Kim, J. W. Cho, K. M. Kim, and E. C. Yi. Refinements of LC-MS/MS Spectral Counting Statistics Improve Quantification of Low Abundance Proteins. *Scientific Reports*, 9(1):13653, Sept. 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-49665-1. [Cited on page 9.]

R. V. Lenth, P. Buerkner, M. Herve, J. Love, H. Riebl, and H. Singmann. Emmeans: Estimated Marginal Means, aka Least-Squares Means, Aug. 2021. [Cited on page 99.]

A. Leroy, P. Latouche, B. Guedj, and S. Gey. MAGMA: Inference and Prediction with Multi-Task Gaussian Processes. *arXiv:2007.10731 [cs, stat]*, July 2020. [Cited on page 65.]

Q. Li, K. Fisher, W. Meng, B. Fang, E. Welsh, E. B. Haura, J. M. Koomen, S. A. Eschrich, B. L. Fridley, and Y. A. Chen. GMSimpute: A generalized two-step Lasso approach to impute missing values in label-free mass spectrum analysis. *Bioinformatics*, 36(1):257–263, Jan. 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz488. [Cited on page 23.]

A. Listrat, M. Gagaoua, J. Normand, D. J. Andueza, D. Gruffat, G. Mairesse, G. Chesneau, B. P. Mourot, C. Gobert, and B. Picard. Are there consistent relationships between major connective tissue components, intramuscular fat content and muscle fibre types in cattle muscle? *Animal*, 14(6):1204–1212, Jan. 2020. ISSN 1751-7311. doi: 10.1017/S1751731119003422. [Cited on page 56.]

R. Little and D. Rubin. *Statistical Analysis with Missing Data, Third Edition*, volume 26 of *Wiley Series in Probability and Statistics*. Wiley edition, 2019. ISBN 978-0-470-52679-8. [Cited on pages viii, 16, 18, and 34.]

R. J. A. Little. Regression With Missing X's: A Review. *Journal of the American Statistical Association*, 87(420):1227–1237, 1992. ISSN 0162-1459. doi: 10.2307/2290664. [Cited on page 18.]

H. Liu, R. G. Sadygov, and J. R. Yates. A Model for Random Sampling and Estimation of Relative Protein Abundance in Shotgun Proteomics. *Analytical Chemistry*, 76(14): 4193–4201, July 2004. ISSN 0003-2700. doi: 10.1021/ac0498563. [Cited on page 9.]

M. Liu and A. Dongre. Proper imputation of missing values in proteomics datasets for differential expression analysis. *Briefings in Bioinformatics*, June 2020. ISSN 1477-4054. doi: 10.1093/bib/bbaa112. [Cited on page 23.]

I. Lönnstedt and T. Speed. Replicated Microarray Data. *Statistica Sinica*, 12(1):31–46, 2002. ISSN 1017-0405. [Cited on page 14.]

C. Ludwig, L. Gillet, G. Rosenberger, S. Amon, B. C. Collins, and R. Aebersold. Data-independent acquisition-based SWATH-MS for quantitative proteomics: A tutorial. *Molecular Systems Biology*, 14(8):e8126, Aug. 2018. ISSN 1744-4292. doi: 10.15252/msb.20178126. [Cited on pages iv, 12, 13, and 30.]

D. H. Lundgren, S.-I. Hwang, L. Wu, and D. K. Han. Role of spectral counting in quantitative proteomics. *Expert Review of Proteomics*, 7(1):39–53, Feb. 2010. ISSN 1478-9450. doi: 10.1586/epr.09.69. [Cited on page 9.]

R. Luo, C. M. Colangelo, W. C. Sessa, and H. Zhao. Bayesian Analysis of iTRAQ Data with Nonrandom Missingness: Identification of Differentially Expressed Proteins. *Statistics in Biosciences*, 1(2):228, Oct. 2009. ISSN 1867-1772. doi: 10.1007/s12561-009-9013-2. [Cited on page 26.]

T. Madsen, M. Świtnicki, M. Juul, and J. S. Pedersen. EBADIMEX: An empirical Bayes approach to detect joint differential expression and methylation and to classify samples. *Statistical Applications in Genetics and Molecular Biology*, 18(6), Nov. 2019. ISSN 15446115. doi: 10.1515/sagmb-2018-0050. [Cited on page 26.]

E. Mammen. Estimating a Smooth Monotone Regression Function. *The Annals of Statistics*, 19(2):724–740, June 1991. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1176348117. [Cited on page 28.]

E. Mammen, J. S. Marron, B. A. Turlach, and M. P. Wand. A General Projection Framework for Constrained Smoothing. *Statistical Science*, 16(3):232–248, Aug. 2001. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1009213727. [Cited on page 28.]

K. V. Mardia, J. T. Kent, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979. ISBN 978-0-12-471250-8. [Cited on page 27.]

B. Martin, C. Daniel, and E. M. Helmut. Bioinformatics in Proteomics. *Current Pharmaceutical Biotechnology*, 5(1):79–88, Jan. 2004. [Cited on page 7.]

B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, Oct. 1975. ISSN 0005-2795. doi: 10.1016/0005-2795(75)90109-9. [Cited on page 81.]

M. Mehrjoo, M. Jafari Jozani, and M. Pawlak. Wind turbine power curve modeling for reliable power prediction using monotonic regression. *Renewable Energy*, 147:214–222, Mar. 2020. ISSN 0960-1481. doi: 10.1016/j.renene.2019.08.060. [Cited on page 28.]

J. G. Meyer. Deep learning neural network tools for proteomics. *Cell Reports Methods*, 1 (2):100003, June 2021. ISSN 2667-2375. doi: 10.1016/j.crmeth.2021.100003. [Cited on page 14.]

K. M. Mullen and I. H. M. van Stokkum. Nnls: The Lawson-Hanson algorithm for non-negative least squares (NNLS), Mar. 2012. [Cited on page 49.]

L. Muller, L. Fornecker, A. V. Dorsselaer, S. Cianférani, and C. Carapito. Benchmarking sample preparation/digestion protocols reveals tube-gel being a fast and repeatable method for quantitative proteomics. *PROTEOMICS*, 16(23):2953–2961, 2016. ISSN 1615-9861. doi: 10.1002/pmic.201600288. [Cited on pages x, 35, 44, 87, and 88.]

L. Muller, L. Fornecker, M. Chion, A. Van Dorsselaer, S. Cianférani, T. Rabilloud, and C. Carapito. Extended investigation of tube-gel sample preparation: A versatile and simple choice for high throughput quantitative proteomics. *Scientific Reports*, 8(1):8260, Dec. 2018. ISSN 20452322. doi: 10.1038/s41598-018-26600-4. [Not cited.]

K. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. Nov. 2007. [Cited on page 110.]

P. Navarro and J. Vázquez. A Refined Method To Calculate False Discovery Rates for Peptide Identification Using Decoy Databases. *Journal of Proteome Research*, 8(4):1792–1796, Apr. 2009. ISSN 1535-3893. doi: 10.1021/pr800362h. [Cited on page 8.]

A. I. Nesvizhskii. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics*, 73 (11):2092–2123, Oct. 2010. ISSN 1874-3919. doi: 10.1016/j.jprot.2010.08.009. [Cited on page 7.]

A. I. Nesvizhskii and R. Aebersold. Interpretation of Shotgun Proteomic Data. *Molecular & Cellular Proteomics*, 4(10):1419–1440, Oct. 2005. ISSN 1535-9476, 1535-9484. doi: 10.1074/mcp.R500012-MCP200. [Cited on pages 7 and 8.]

W. M. A. Niessen. *Liquid Chromatography-Mass Spectrometry*. CRC Press, Boca Raton, third edition, Aug. 2006. ISBN 978-0-429-11680-3. doi: 10.1201/9781420014549. [Cited on page 6.]

J. J. O'Brien, H. P. Gunawardena, J. A. Paulo, X. Chen, J. G. Ibrahim, S. P. Gygi, and B. F. Qaqish. The effects of nonignorable missing data on label-free mass spectrometry proteomics experiments. *Annals of Applied Statistics*, 12(4):2075–2095, Dec. 2018. ISSN 1932-6157, 1941-7330. doi: 10.1214/18-AOAS1144. [Cited on pages 16, 26, and 28.]

N. A. O'Leary, M. W. Wright, J. R. Brister, S. Ciufo, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei, A. Astashyn, A. Badretdin, Y. Bao, O. Blinkova, V. Brover, V. Chetvernin, J. Choi, E. Cox, O. Ermolaeva, C. M. Farrell, T. Goldfarb, T. Gupta, D. Haft, E. Hatcher, W. Hlavina, V. S. Joardar, V. K. Kodali, W. Li, D. Maglott, P. Masterson, K. M. McGarvey, M. R. Murphy, K. O'Neill, S. Pujar, S. H. Rangwala, D. Rausch, L. D. Riddick, C. Schoch, A. Shkeda, S. S. Storz, H. Sun, F. Thibaud-Nissen, I. Tolstoy, R. E. Tully, A. R. Vatsan, C. Wallin, D. Webb, W. Wu, M. J. Landrum, A. Kimchi, T. Tatusova, M. DiCuccio, P. Kitts, T. D. Murphy, and K. D. Pruitt. Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1):D733–D745, Jan. 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv1189. [Cited on page 8.]

M. Olivier, R. Asmis, G. A. Hawkins, T. D. Howard, and L. A. Cox. The Need for Multi-Omics Biomarker Signatures in Precision Medicine. *International Journal of Molecular Sciences*, 20(19):4781, Sept. 2019. ISSN 1422-0067. doi: 10.3390/ijms20194781. [Cited on page 4.]

Q. Pan and R. Wei. Improved methods for estimating fraction of missing information in multiple imputation. *Cogent mathematics & statistics*, 5:1551504, 2018. ISSN 2574-2558. doi: 10.1080/25742558.2018.1551504. [Cited on page 21.]

A. G. Paulovich, D. Billheimer, A.-J. L. Ham, L. Vega-Montoto, P. A. Rudnick, D. L. Tabb, P. Wang, R. K. Blackman, D. M. Bunk, H. L. Cardasis, K. R. Clauser, C. R. Kinsinger, B. Schilling, T. J. Tegeler, A. M. Variyath, M. Wang, J. R. Whiteaker, L. J. Zimmerman, D. Fenyo, S. A. Carr, S. J. Fisher, B. W. Gibson, M. Mesri, T. A. Neubert, F. E. Regnier, H. Rodriguez, C. Spiegelman, S. E. Stein, P. Tempst, and D. C. Liebler. Interlaboratory Study Characterizing a Yeast Performance Standard for Benchmarking LC-MS Platform Performance *. *Molecular & Cellular Proteomics*, 9(2):242–254, Feb. 2010. ISSN 1535-9476, 1535-9484. doi: 10.1074/mcp.M900222-MCP200. [Cited on page 23.]

D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *ELECTROPHORESIS*, 20(18):3551–3567, 1999. ISSN 1522-2683. doi: 10.1002/(SICI)1522-2683(19991201)20:18<3551::AID-ELPS3551>3.0.CO;2-2. [Cited on page 7.]

B. Phipson, S. Lee, I. J. Majewski, W. S. Alexander, and G. K. Smyth. Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression. *The Annals of Applied Statistics*, 10(2):946–963, June 2016. ISSN 1932-6157, 1941-7330. doi: 10.1214/16-AOAS920. [Cited on pages ix, 14, 34, 68, and 73.]

B. Picard, M. Gagaoua, D. Micol, I. Cassar-Malek, J.-F. Hocquette, and C. E. M. Terlouw. Inverse Relationships between Biomarkers and Beef Tenderness According to Contractile and Metabolic Properties of the Muscle. *Journal of Agricultural and Food Chemistry*, 62 (40):9808–9818, Oct. 2014. ISSN 0021-8561. doi: 10.1021/jf501528s. [Cited on page 44.]

B. Picard, B. Lebret, I. Cassar-Malek, L. Liaubet, C. Berri, E. Le Bihan-Duval, J. F. Hocquette, and G. Renand. Recent advances in omic technologies for meat quality management. *Meat Science*, 109:18–26, Nov. 2015. ISSN 0309-1740. doi: 10.1016/j.meatsci.2015.05.003. [Cited on page 43.]

B. Picard, M. Gagaoua, M. Al-Jammas, L. D. Koning, A. Valais, and M. Bonnet. Beef tenderness and intramuscular fat proteomic biomarkers: Muscle type effect. *PeerJ*, 6: e4891, June 2018. ISSN 2167-8359. doi: 10.7717/peerj.4891. [Cited on pages 43 and 44.]

B. Picard, M. Gagaoua, M. Al Jammas, and M. Bonnet. Beef tenderness and intramuscular fat proteomic biomarkers: Effect of gender and rearing practices. *Journal of Proteomics*, 200:1–10, May 2019. ISSN 1874-3919. doi: 10.1016/j.jprot.2019.03.010. [Cited on page 44.]

L. K. Pino, B. C. Searle, J. G. Bollinger, B. Nunn, B. MacLean, and M. J. MacCoss. The Skyline ecosystem: Informatics for quantitative mass spectrometry proteomics. *Mass Spectrometry Reviews*, 39(3):229–244, May 2020. ISSN 1098-2787. doi: 10.1002/mas.21540. [Cited on page 46.]

L. Quibel, P. Helluy, M. Chion, and P. Ricka. Mélanger des gaz raides pour créer de nouvelles lois d'état. Research Report, IRMA, Université de Strasbourg ; EDF R&D, Apr. 2019. [Not cited.]

J. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer-Verlag, New York, second edition, 2005. ISBN 978-0-387-40080-8. doi: 10.1007/b98888. [Cited on page 29.]

J. O. Ramsay. Monotone Regression Splines in Action. *Statistical Science*, 3(4):425–441, Nov. 1988. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1177012761. [Cited on pages v, 29, 30, and 58.]

C. Ramus, A. Hovasse, M. Marcellin, A. M. Hesse, E. Mouton-Barbosa, D. Bouyssié, S. Vaca, C. Carapito, K. Chaoui, C. Bruley, J. Garin, S. Cianférani, M. Ferro, A. Van Dorssaeler, O. Burlet-Schiltz, C. Schaeffer, Y. Couté, and A. Gonzalez de Peredo. Benchmarking quantitative label-free LC-MS data processing workflows using a complex spiked proteomic

standard dataset. *Journal of Proteomics*, 132:51–62, Jan. 2016. ISSN 18767737. doi: 10.1016/j.jprot.2015.11.011. [Cited on page 23.]

C. E. Rasmussen, C. K. I. Williams, and F. Bach. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 978-0-262-18253-9. [Cited on page 61.]

L. Reiter, O. Rinner, P. Picotti, R. Hüttenhain, M. Beck, M.-Y. Brusniak, M. O. Hengartner, and R. Aebersold. mProphet: Automated data processing and statistical validation for large-scale SRM experiments. *Nature Methods*, 8(5):430–435, May 2011. ISSN 1548-7105. doi: 10.1038/nmeth.1584. [Cited on page 47.]

K. Richardson, R. Denny, C. Hughes, J. Skilling, J. Sikora, M. Dadlez, A. Manteca, H. R. Jung, O. N. Jensen, V. Redeker, R. Melki, J. I. Langridge, and J. P. Vissers. A Probabilistic Framework for Peptide and Protein Quantification from Data-Dependent and Data-Independent LC-MS Proteomics Experiments. *OMICS: A Journal of Integrative Biology*, 16(9):468–482, Sept. 2012. doi: 10.1089/omi.2012.0019. [Cited on page 9.]

P. Rigollet and J. Weed. Uncoupled isotonic regression via minimum Wasserstein deconvolution. *Information and Inference: A Journal of the IMA*, 8(4):691–717, Dec. 2019. ISSN 2049-8772. doi: 10.1093/imaiai/iaz006. [Cited on page 28.]

T. Robertson, F. T. Wright, and R. Dykstra. *Order Restricted Statistical Inference*. John Wiley & Sons, Chichester, 1988. ISBN 978-0-471-91787-8. [Cited on page 28.]

G. Rosenberger, C. C. Koh, T. Guo, H. L. Röst, P. Kouvonen, B. C. Collins, M. Heusel, Y. Liu, E. Caron, A. Vichalkovski, M. Faini, O. T. Schubert, P. Faridi, H. A. Ebhardt, M. Matondo, H. Lam, S. L. Bader, D. S. Campbell, E. W. Deutsch, R. L. Moritz, S. Tate, and R. Aebersold. A repository of assays to quantify 10,000 human proteins by SWATH-MS. *Scientific Data*, 1(1):140031, Sept. 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.31. [Cited on page 13.]

D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, Dec. 1976. ISSN 0006-3444. doi: 10.1093/biomet/63.3.581. [Cited on page 17.]

D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, 1987. ISBN 978-0-471-65574-9. [Cited on pages 19, 68, and 69.]

S. Y. Ryu, W.-J. Qian, D. G. Camp, R. D. Smith, R. G. Tompkins, R. W. Davis, and W. Xiao. Detecting differential protein expression in large-scale population proteomics. *Bioinformatics*, 30(19):2741–2746, Oct. 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu341. [Cited on page 26.]

Y. Sasaki. The truth of the F-measure. Jan. 2007. [Cited on page 81.]

J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman and Hall/CRC, New York, Aug. 1997. ISBN 978-0-367-80302-5. doi: 10.1201/9780367803025. [Cited on pages ix, 35, 68, and 69.]

J. L. Schafer and J. W. Graham. Missing data: Our view of the state of the art. *Psychological Methods*, 7(2):147–177, June 2002. ISSN 1082-989X. [Cited on page 18.]

C. Schleiss, R. Carapito, L.-M. Fornecker, L. Muller, N. Paul, O. Tahar, A. Pichot, M. Tavian, A. Nicolae, L. Miguet, L. Mauvieux, R. Herbrecht, S. Cianferani, J.-N. Freund, C. Carapito, M. Maumy-Bertrand, S. Bahram, F. Bertrand, and L. Vallat. Temporal multiomic modeling reveals a B-cell receptor proliferative program in chronic lymphocytic leukemia. *Leukemia*, 35(5):1463–1474, May 2021. ISSN 1476-5551. doi: 10.1038/s41375-021-01221-5. [Cited on page 4.]

O. T. Schubert, L. C. Gillet, B. C. Collins, P. Navarro, G. Rosenberger, W. E. Wolski, H. Lam, D. Amodei, P. Mallick, B. MacLean, and R. Aebersold. Building high-quality assay libraries for targeted analysis of SWATH MS data. *Nature Protocols*, 10(3):426–441, Mar. 2015a. ISSN 1750-2799. doi: 10.1038/nprot.2015.015. [Cited on page 13.]

O. T. Schubert, C. Ludwig, M. Kogadeeva, M. Zimmermann, G. Rosenberger, M. Gengenbacher, L. C. Gillet, B. C. Collins, H. L. Röst, S. H. E. Kaufmann, U. Sauer, and R. Aebersold. Absolute Proteome Composition and Dynamics during Dormancy and Resuscitation of Mycobacterium tuberculosis. *Cell Host & Microbe*, 18(1):96–108, July 2015b. ISSN 1931-3128. doi: 10.1016/j.chom.2015.06.001. [Cited on page 28.]

M. Shen, Y.-T. Chang, C.-T. Wu, S. J. Parker, G. Saylor, Y. Wang, G. Yu, J. E. V. Eyk, R. Clarke, D. M. Herrington, and Y. Wang. Comparative Assessment and Outlook on Methods for Imputing Proteomics Data. Preprint, In Review, Mar. 2021. [Cited on page 23.]

J. C. Silva, M. V. Gorenstein, G.-Z. Li, J. P. C. Vissers, and S. J. Geromanos. Absolute quantification of proteins by LCMSE: A virtue of parallel MS acquisition. *Molecular & cellular proteomics: MCP*, 5(1):144–156, Jan. 2006. ISSN 1535-9476. doi: 10.1074/mcp.M500230-MCP200. [Cited on page 50.]

P. Sinitcyn, H. Hamzeiy, F. Salinas Soto, D. Itzhak, F. McCarthy, C. Wichmann, M. Steger, U. Ohmayer, U. Distler, S. Kaspar-Schoenefeld, N. Prianichnikov, Ş. Yılmaz, J. D. Rudolph, S. Tenzer, Y. Perez-Riverol, N. Nagaraj, S. J. Humphrey, and J. Cox. MaxDIA enables library-based and library-free data-independent acquisition proteomics. *Nature Biotechnology*, pages 1–11, July 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-00968-7. [Cited on page 14.]

L. M. Smith and N. L. Kelleher. Proteoform: A single term describing protein complexity. *Nature Methods*, 10(3):186–187, Mar. 2013. ISSN 1548-7105. doi: 10.1038/nmeth.2369. [Cited on page 4.]

L. M. Smith and N. L. Kelleher. Proteoforms as the next proteomics currency. *Science*, 359 (6380):1106–1107, Mar. 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aat1884. [Cited on page 4.]

G. K. Smyth. Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1–25, Jan. 2004. ISSN 1544-6115. doi: 10.2202/1544-6115.1027. [Cited on pages ix, xi, 14, 15, 34, 36, 72, 73, 100, and 108.]

G. K. Smyth, Y. H. Yang, and T. Speed. Statistical Issues in cDNA Microarray Data Analysis. In M. J. Brownstein and A. B. Khodursky, editors, *Functional Genomics: Methods and Protocols*, Methods in Molecular Biology, pages 111–136. Humana Press, Totowa, NJ, 2003. ISBN 978-1-59259-364-4. doi: 10.1385/1-59259-364-X:111. [Cited on page 14.]

J. Song and C. Yu. Missing Value Imputation using XGboost for Label-Free Mass Spectrometry-Based Proteomics Data. Preprint, Bioinformatics, Apr. 2021. [Cited on pages 23 and 25.]

D. C. Stahl, K. M. Swiderek, M. T. Davis, and T. D. Lee. Data-controlled automation of liquid chromatography/tandem mass spectrometry analysis of peptide mixtures. *Journal of the American Society for Mass Spectrometry*, 7(6):532–540, June 1996. doi: 10.1016/1044-0305(96)00057-8. [Cited on page 7.]

H. Steen and M. Mann. The abc's (and xyz's) of peptide sequencing. *Nature Reviews Molecular Cell Biology*, 5(9):699–711, Sept. 2004. ISSN 1471-0080. doi: 10.1038/nrm1468. [Cited on page 6.]

D. J. Stekhoven and P. Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, Jan. 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr597. [Cited on page 69.]

K. Strimbu and J. A. Tavel. What are Biomarkers? *Current opinion in HIV and AIDS*, 5 (6):463–466, Nov. 2010. ISSN 1746-630X. doi: 10.1097/COH.0b013e32833ed177. [Cited on page 4.]

G. M. Sullivan and R. Feinn. Using Effect Size—or Why the P Value Is Not Enough. *Journal of Graduate Medical Education*, 4(3):279–282, Sept. 2012. ISSN 1949-8349. doi: 10.4300/JGME-D-12-00156.1. [Cited on page 128.]

D. L. Tabb. The SEQUEST Family Tree. *Journal of the American Society for Mass Spectrometry*, 26(11):1814–1819, Nov. 2015. ISSN 1044-0305. doi: 10.1007/s13361-015-1201-3. [Cited on page 7.]

S. L. Taylor, G. S. Leiserowitz, and K. Kim. Accounting for undetected compounds in statistical analyses of mass spectrometry 'omic studies. *Statistical Applications in Genetics and Molecular Biology*, 12(6):703–722, Dec. 2013. ISSN 1544-6115. doi: 10.1515/sagmb-2013-0021. [Cited on page 26.]

R. Team. RStudio: Integrated Development Environment for R. RStudio, PBC., 2021a. [Cited on page 95.]

R. C. Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, 2021b. [Cited on page 95.]

The UniProt Consortium. UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, Jan. 2021. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100. [Cited on page 8.]

S. Tiwary, R. Levy, P. Gutenbrunner, F. Salinas Soto, K. K. Palaniappan, L. Deming, M. Berndl, A. Brant, P. Cimermancic, and J. Cox. High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis. *Nature Methods*, 16(6):519–525, June 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0427-6. [Cited on page 14.]

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, June 2001. ISSN 1367-4803. doi: 10.1093/bioinformatics/17.6.520. [Cited on pages ix, 35, 68, and 69.]

L. Tsiatsiani and A. J. R. Heck. Proteomics beyond trypsin. *The FEBS Journal*, 282(14): 2612–2626, 2015. ISSN 1742-4658. doi: 10.1111/febs.13287. [Cited on page 5.]

S. Tyanova, T. Temu, P. Sinitcyn, A. Carlson, M. Y. Hein, T. Geiger, M. Mann, and J. Cox. The Perseus computational platform for comprehensive analysis of (prote)omics data. *Nature Methods*, 13(9):731–740, Sept. 2016. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.3901. [Cited on pages vii, 24, 25, and 33.]

S. van Buuren. *Flexible Imputation of Missing Data, Second Edition*. CRC Press, July 2018. ISBN 978-0-429-96035-2. [Cited on pages 16 and 18.]

S. van Buuren and K. Groothuis-Oudshoorn. Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(1):1–67, Dec. 2011. ISSN 1548-7660. doi: 10.18637/jss.v045.i03. [Cited on pages 25, 68, and 69.]

S. van Buuren, K. Groothuis-Oudshoorn, G. Vink, R. Schouten, A. Robitzsch, P. Rockenschaub, L. Doove, S. Jolani, M. Moreno-Betancur, I. White, P. Gaffert, F. Meinfelder, B. Gray, and V. Arel-Bundock. Mice: Multivariate Imputation by Chained Equations, Jan. 2021. [Cited on page 97.]

V. Vidova and Z. Spacil. A review on mass spectrometry-based quantitative proteomics: Targeted and data independent acquisition. *Analytica Chimica Acta*, 964:7–23, Apr. 2017. ISSN 0003-2670. doi: 10.1016/j.aca.2017.01.059. [Cited on page 12.]

P. T. von Hippel. How Many Imputations Do You Need? A Two-stage Calculation Using a Quadratic Rule. *Sociological Methods & Research*, 49(3):699–718, Aug. 2020. ISSN 0049-1241, 1552-8294. doi: 10.1177/0049124117747303. [Cited on page 21.]

J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional Data Analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295, June 2016. ISSN 2326-8298. doi: 10.1146/annurev-statistics-041715-033624. [Cited on page 30.]

M. Wang, J. Wang, J. Carver, B. S. Pullman, S. W. Cha, and N. Bandeira. Assembling the Community-Scale Discoverable Human Proteome. *Cell Systems*, 7(4):412–421.e5, Oct. 2018. ISSN 2405-4712. doi: 10.1016/j.cels.2018.08.004. [Cited on page 13.]

M. Wang, L. Jiang, R. Jian, J. Y. Chan, Q. Liu, M. P. Snyder, and H. Tang. RobNorm: Model-based robust normalization method for labeled quantitative mass spectrometry proteomics data. *Bioinformatics*, 37(6):815–821, Mar. 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa904. [Cited on page 89.]

W. Wang and J. Yan. Splines2: Regression Spline Functions and Classes, Aug. 2021. [Cited on page 49.]

R. L. Wasserstein, A. L. Schirm, and N. A. Lazar. Moving to a World Beyond "p $<$ 0.05". *The American Statistician*, 73(sup1):1–19, Mar. 2019. ISSN 0003-1305. doi: 10.1080/00031305.2019.1583913. [Cited on page 115.]

B.-J. M. Webb-Robertson, H. K. Wiberg, M. M. Matzke, J. N. Brown, J. Wang, J. E. McDermott, R. D. Smith, K. D. Rodland, T. O. Metz, J. G. Pounds, and K. M. Waters. Review, Evaluation, and Discussion of the Challenges of Missing Value Imputation for Mass Spectrometry-Based Label-Free Global Proteomics. *Journal of Proteome Research*, 14(5):1993–2001, May 2015. ISSN 1535-3893, 1535-3907. doi: 10.1021/pr501138h. [Cited on page 23.]

E. J. Wegman and I. W. Wright. Splines in Statistics. *Journal of the American Statistical Association*, 78(382):351–365, June 1983. ISSN 0162-1459. doi: 10.1080/01621459.1983.10477977. [Cited on page 28.]

I. R. White, P. Royston, and A. M. Wood. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine*, 30(4):377–399, Feb. 2011. ISSN 02776715. doi: 10.1002/sim.4067. [Cited on pages 21 and 69.]

S. Wieczorek, F. Combes, C. Lazar, Q. Giai Gianetto, L. Gatto, A. Dorffer, A.-M. Hesse, Y. Couté, M. Ferro, C. Bruley, and T. Burger. DAPAR & ProStaR: Software to perform statistical analyses in quantitative discovery proteomics. *Bioinformatics (Oxford, England)*, 33(1):135–136, Jan. 2017. ISSN 1367-4811. doi: 10.1093/bioinformatics/btw580. [Cited on pages vii, 18, 25, 33, 72, 80, and 90.]

S. Wieczorek, F. Combes, H. Borges, and T. Burger. Protein-Level Statistical Analysis of Quantitative Label-Free Proteomics Data with ProStaR. In V. Brun and Y. Couté, editors, *Proteomics for Biomarker Discovery: Methods and Protocols*, Methods in Molecular Biology, pages 225–246. Springer, New York, NY, 2019. ISBN 978-1-4939-9164-8. doi: 10.1007/978-1-4939-9164-8_15. [Cited on page 25.]

M. R. Wilkins, C. Pasquali, R. D. Appel, K. Ou, O. Golaz, J.-C. Sanchez, J. X. Yan, A. A. Gooley, G. Hughes, I. Humphery-Smith, K. L. Williams, and D. F. Hochstrasser. From Proteins to Proteomes: Large Scale Protein Identification by Two-Dimensional Electrophoresis and Arnino Acid Analysis. *Bio/Technology*, 14(1):61–65, Jan. 1996. ISSN 1546-1696. doi: 10.1038/nbt0196-61. [Cited on page 3.]

J. Wishart. The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population. *Biometrika*, 20A(1-2):32–52, Dec. 1928. ISSN 0006-3444. doi: 10.1093/biomet/20A.1-2.32. [Cited on page 27.]

J. R. Wiśniewski, H. Koepsell, A. Gizak, and D. Rakus. Absolute protein quantification allows differentiation of cell-specific metabolic routes and functions. *PROTEOMICS*, 15 (7):1316–1325, 2015. ISSN 1615-9861. doi: 10.1002/pmic.201400456. [Cited on page 28.]

G. W. Wright and R. M. Simon. A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics*, 19(18):2448–2455, Dec. 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/btg345. [Cited on page 14.]

J. Wu, M. C. Meyer, and J. D. Opsomer. Penalized isotonic regression. *Journal of Statistical Planning and Inference*, 161:12–24, June 2015. ISSN 0378-3758. doi: 10.1016/j.jspi.2014.12.008. [Cited on page 28.]

L. L. Xu, A. Young, A. Zhou, and H. L. Röst. Machine Learning in Mass Spectrometric Analysis of DIA Data. *Proteomics*, 20(21-22):e1900352, Nov. 2020. ISSN 1615-9861. doi: 10.1002/pmic.201900352. [Cited on page 14.]

J. Yerushalmy. Statistical problems in assessing methods of medical diagnosis, with special reference to X-ray techniques. *Public Health Reports (Washington, D.C.: 1896)*, 62(40): 1432–1449, Oct. 1947. ISSN 0094-6214. [Cited on page 81.]

X. Yin, D. Levy, C. Willinger, A. Adourian, and M. G. Larson. Multiple imputation and analysis for high-dimensional incomplete proteomics data. *Statistics in Medicine*, 35(8): 1315–1326, 2016. ISSN 1097-0258. doi: 10.1002/sim.6800. [Cited on page 23.]

M. Zahn-Zabal, P.-A. Michel, A. Gateau, F. Nikitin, M. Schaeffer, E. Audot, P. Gaudet, P. D. Duek, D. Teixeira, V. Rech de Laval, K. Samarasinghe, A. Bairoch, and L. Lane. The neXtProt knowledgebase in 2020: Data, tools and usability improvements. *Nucleic Acids Research*, 48(D1):D328–D334, Jan. 2020. ISSN 0305-1048. doi: 10.1093/nar/gkz995. [Cited on page 8.]

W.-F. Zeng, X.-X. Zhou, W.-J. Zhou, H. Chi, J. Zhan, and S.-M. He. MS/MS Spectrum Prediction for Modified Peptides Using pDeep2 Trained by Transfer Learning. *Analytical Chemistry*, 91(15):9724–9731, Aug. 2019. ISSN 0003-2700. doi: 10.1021/acs.analchem.9b01262. [Cited on page 14.]

W. Zhang, Y. Wei, V. Ignatchenko, L. Li, S. Sakashita, N.-A. Pham, P. Taylor, M. S. Tsao, T. Kislinger, and M. F. Moran. Proteomic profiles of human lung adeno and squamous cell carcinoma using super-SILAC and label-free quantification approaches. *PROTEOMICS*, 14(6):795–803, 2014. ISSN 1615-9861. doi: 10.1002/pmic.201300382. [Cited on page 23.]

Y. Zhang, Z. Wen, M. P. Washburn, and L. Florens. Refinements to Label Free Proteome Quantitation: How to Deal with Peptides Shared by Multiple Proteins. *Analytical Chemistry*, 82(6):2272–2281, Mar. 2010. ISSN 0003-2700. doi: 10.1021/ac9023999. [Cited on page 9.]

Y. Zhang, B. R. Fonslow, B. Shan, M.-C. Baek, and J. R. Yates. Protein Analysis by Shotgun/Bottom-up Proteomics. *Chemical Reviews*, 113(4):2343–2394, Apr. 2013. ISSN 0009-2665. doi: 10.1021/cr3003533. [Cited on page 5.]

Université de Strasbourg

Marie CHION

Développement de nouvelles méthodologies statistiques pour l'analyse de données de protéomique quantitative

Résumé

L'analyse protéomique consiste à étudier l'ensemble des protéines exprimées par un système biologique donné, à un moment donné et dans des conditions données. Les récents progrès technologiques en spectrométrie de masse et en chromatographie liquide permettent d'envisager aujourd'hui des études protéomiques à large échelle et à haut débit.

Ce travail de thèse porte sur le développement de méthodologies statistiques pour l'analyse des données de protéomique quantitative et présente ainsi trois principales contributions. La première partie propose d'utiliser des modèles de régression par spline monotone pour estimer les quantités de tous les peptides détectés dans un échantillon grâce à l'utilisation de standards internes marqués pour un sous-ensemble de peptides ciblés. La deuxième partie présente une stratégie de prise en compte de l'incertitude induite par le processus d'imputation multiple dans l'analyse différentielle, également implémentée dans le package R mi4p. Enfin, la troisième partie propose un cadre bayésien pour l'analyse différentielle, permettant notamment de tenir compte des corrélations entre les intensités des peptides.

Mots-clés : Données de grande dimension, modèles de régression, valeurs manquantes, imputation multiple, analyse différentielle, données de protéomique quantitative.

Résumé en anglais

Proteomic analysis consists of studying all the proteins expressed by a given biological system, at a given time and under given conditions. Recent technological advances in mass spectrometry and liquid chromatography make it possible to envisage large-scale and high-throughput proteomic studies.

This thesis work focuses on developing statistical methodologies for the analysis of quantitative proteomics data and thus presents three main contributions. The first part proposes to use monotone spline regression models to estimate the amounts of all peptides detected in a sample using internal standards labelled for a subset of targeted peptides. The second part presents a strategy to account for the uncertainty induced by the multiple imputation process in the differential analysis, also implemented in the mi4p R package. Finally, the third part proposes a Bayesian framework for differential analysis, making it notably possible to consider the correlations between the intensities of peptides.

Keywords: High-dimensional data, regression models, missing values, multiple imputation, differential analysis, quantitative proteomics data.