Supplemental Materials: A Statistical Framework for Protein Quantitation in Bottom-Up MS-based Proteomics

(May 30, 2009)

1 Censored Likelihood Model

Let Y_{ijs} and $\boldsymbol{x_{ijs}}$ be the intensity and a vector of covariates, respectively, for protein i, peptide j, and sample s, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, m_i$, $s = 1, 2, \ldots, n$. We assume the linear model $Y_{ijs} = \boldsymbol{x_{ijs}}'\boldsymbol{\beta_i} + \epsilon_{ijs}$, where $\epsilon_{ijs} \sim N(0, \sigma_{ij})$, and the ϵ variables are mutually independent. Note that, with an appropriate choice of the model matrix composed of the $\boldsymbol{x}_{ijs}$, this model parameterization is equivalent to the ANOVA model formulation used in equation (1) of the main paper:

$$Y_{ijks} = \mathrm{PROT}_i + \mathrm{PEP}_{ij} + \mathrm{GRP}_{ik} + \epsilon_{ijks}, \tag{1}$$

for protein i, peptide j, comparison group k, and replicate s.

Let U_{ijs} be an indicator of whether a random ("completely at random" in statistical parlance) mechanism causes observation $\{ijs\}$ to be missing (1 if missing, 0 otherwise). Let $P(U_{ijs} = 1) = \pi_s$, and assume the U and Y variables are mutually independent. Let W_{ijs} be the overall missingness indicator, equal to 0 if and only if $U_{ijs} = 0$ and $Y_{ijs} > c_{ij}$, with c_{ij} an unkown censoring point for peptide j of protein i. As described in the main paper,

$$P(W_{ijs} = 1) = \pi_s + (1 - \pi_s)\Phi\big(\zeta_{ijs}\big), \tag{2}$$

where $\delta_{ij} = 1/\sigma_{ij}$, $\zeta_{ijs} = \delta_{ij}\big(c_{ij} - \boldsymbol{x_{ijs}}'\boldsymbol{\beta_i}\big)$, $\Phi(\cdot)$ is the CDF of the standard normal distribution: $\Phi(x) = \int_{-\infty}^{x} \phi(t)dt$, and $\phi(\cdot)$ is the standard normal *pdf.* We would like to estimate the unknown parameters $\boldsymbol{\theta} = (\pi, \boldsymbol{c}, \boldsymbol{\beta}, \boldsymbol{\delta})$ by maximizing the resulting log-likelihood for each protein:

$$\sum_{j=1}^{m_i}\sum_{s=1}^{n}\left\{(1 - W_{ijs})\left[\log\delta_{ij} - \frac{1}{2}\delta_{ij}^2\big(y_{ijs} - \boldsymbol{x_{ijs}}'\boldsymbol{\beta_i}\big)^2\right] + W_{ijs}\log\left[\pi_s + (1 - \pi_s)\Phi\big(\zeta_{ijs}\big)\right]\right\}.$$

1.1 Estimation

1.1.1 Censoring cutoffs $\boldsymbol{c}$

The likelihood is increasing in the c_{ij}. However, there is a natural upper bound for each c_{ij} in that any observed y_{ijs} had to satisfy $y_{ijs} > c_{ij}$. Thus, the MLE's are $\hat{c}_{ij} = \min\{y_{ijs} : i = s, \ldots, n\}$.

1.1.2 Random missingness parameters π

Preliminary results suggest that the MLEs for the π_s are poor estimates. We will consider a simple alternative estimator, then treat it as known when estimating the other parameters. Figure 2 in the main document shows missingness proportions versus observed means for peptides for one sample of the diabetes dataset. Assuming that some of the peptides have intensities that are high enough to prevent censoring, the right hand side of the figure tells us about π_s. In particular, we can estimate π_s as the verticle offset from zero at the right side of the figure. To estimate the systematic trend of the figure, we fit a natural cubic spline with 5 degrees of freedom. Splines allow for flexible parameterization of arbitrary, nonlinear, functions, and natural cubic splines have some nice theoretical properties. We estimate π_s as the fitted value of the curve at the maximum observed sample mean.

1.1.3 Remaining parameters $\boldsymbol{\theta_0} = (\boldsymbol{\beta}, \boldsymbol{\delta})$

MLE's for the remaining parameters must be found by numerically maximizing the log-likelihood function. Let

$$\Psi_s(x) = \frac{(1-\pi_s)\phi(x)}{\pi_s + (1-\pi_s)\Phi(x)},$$

and note that $\Psi_s'(x) = -\Psi_s(x)[x + \Psi_s(x)]$. Generic numerical optimization algorithms require just the first and second derivatives of the log-likelihood, which we can now write as

$$\begin{aligned}
\frac{\partial l_i}{\partial \delta_{ij}} &= \sum_{s=1}^{n} \left\{ \frac{1}{\delta_{ij}} \left[(1-W_{ijs}) \left(1 - \delta_{ij}^2 \big(y_{ijs} - \boldsymbol{x_{ijs}}' \boldsymbol{\beta_i} \big)^2 \right) + W_{ijs} \zeta_{ijs} \Psi_s\big(\zeta_{ijs}\big) \right] \right\} \\
\frac{\partial l_i}{\partial \boldsymbol{\beta_i}} &= \sum_{j=1}^{m_i} \sum_{s=1}^{n} \left\{ \delta_{ij} \left[(1-W_{ijs}) \delta_{ij} \big(y_{ijs} - \boldsymbol{x_{ijs}}' \boldsymbol{\beta_i} \big) - W_{ijs} \Psi_s\big(\zeta_{ijs}\big) \right] \boldsymbol{x_{ijs}} \right\} \\
\frac{\partial^2 l_i}{\partial \delta_{ij}^2} &= \sum_{s=1}^{n} \left\{ \frac{1}{\delta_{ij}^2} \left[-(1-W_{ijs}) \left(1 + \delta_{ij}^2 \big(y_{ijs} - \boldsymbol{x_{ijs}}' \boldsymbol{\beta_i} \big)^2 \right) + W_{ijs} \zeta_{ijs}^2 \Psi_s'\big(\zeta_{ijs}\big) \right] \right\} \\
\frac{\partial^2 l_i}{\partial \delta_{ij} \partial \delta_{ij'}} &= 0, \; j \neq j' \\
\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \boldsymbol{\beta_i}'} &= \sum_{j=1}^{m_i} \sum_{s=1}^{n} \left\{ \delta_{ij}^2 \left[-(1-W_{ijs}) + W_{ijs} \Psi_s'\big(\zeta_{ijs}\big) \right] \big(\boldsymbol{x_{ijs}} \boldsymbol{x_{ijs}}' \big) \right\} \\
\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \delta_{ij}} &= \sum_{s=1}^{n} \left\{ \left[2(1-W_{ijs}) \delta_{ij} \big(y_{ijs} - \boldsymbol{x_{ijs}}' \boldsymbol{\beta_i} \big) - W_{ijs} \left(\Psi_s\big(\zeta_{ijs}\big) + \zeta_{ijs} \Psi_s'\big(\zeta_{ijs}\big) \right) \right] \boldsymbol{x_{ijs}} \right\}.
\end{aligned}$$

1.2 Information Content

Maximum likelihood theory defines information content using second derivative matrices of the log-likelihood. The *observed* information is just the matrix of negative second derivatives derived above:

$$\boldsymbol{I}_0(\boldsymbol{\theta}_i) = \begin{pmatrix} \boldsymbol{I}_0(\boldsymbol{\delta}_i) & \boldsymbol{I}_0(\boldsymbol{\beta}_i, \boldsymbol{\delta}_i) \\ \boldsymbol{I}_0'(\boldsymbol{\beta}_i, \boldsymbol{\delta}_i) & \boldsymbol{I}_0(\boldsymbol{\beta}_i) \end{pmatrix}$$

where

$$
\begin{aligned}
\boldsymbol{I}_0(\boldsymbol{\delta}_i) &= -\frac{\partial^2 l_i}{\partial \delta_{ij}^2} \\
\boldsymbol{I}_0(\boldsymbol{\beta}_i) &= -\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \boldsymbol{\beta_i}'} \\
\boldsymbol{I}_0(\boldsymbol{\beta}_i, \boldsymbol{\delta}_i) &= -\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \delta_{ij}}.
\end{aligned}
$$

The *expected* information matrix takes the expectation of the above values with respect to the random vectors (Y_{ijs}, W_{ijs}) and can be derived as

$$
\boldsymbol{I}(\boldsymbol{\theta}_i) = \begin{pmatrix} \boldsymbol{I}(\boldsymbol{\delta}_i) & \boldsymbol{I}(\boldsymbol{\beta}_i, \boldsymbol{\delta}_i) \\ \boldsymbol{I}'(\boldsymbol{\beta}_i, \boldsymbol{\delta}_i) & \boldsymbol{I}(\boldsymbol{\beta}_i) \end{pmatrix}
$$

where

$$
\begin{aligned}
\boldsymbol{I}(\delta_{ij}) &= -\mathrm{E}\left(\frac{\partial^2 l_i}{\partial \delta_{ij}^2}\right) = \sum_{s=1}^{n} \frac{1}{\delta_{ij}^2}\Big[2(1-\kappa_{ijs}) - \zeta_{ijs}^2 \Psi_s(\zeta_{ijs})\kappa_{ijs}\Big] \\
\boldsymbol{I}(\boldsymbol{\beta}_i) &= -\mathrm{E}\left(\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \boldsymbol{\beta_i}'}\right) = \sum_{j=1}^{m_i}\sum_{s=1}^{n} \delta_{ij}^2 \Big[1 - \kappa_{ijs}\big(1 + \Psi_s'(\zeta_{ijs})\big)\Big]\big(\boldsymbol{x}_{ijs}\boldsymbol{x}_{ijs}'\big) \\
\boldsymbol{I}(\boldsymbol{\beta}_i, \delta_{ij}) &= -\mathrm{E}\left(\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \delta_{ij}}\right) = \sum_{s=1}^{n} \Big[\Psi_s(\zeta_{ijs}) + \zeta_{ijs}\Psi_s'(\zeta_{ijs})\Big]\kappa_{ijs}\boldsymbol{x}_{ijs},
\end{aligned}
$$

and where $\kappa_{ijs} = \pi_s + (1-\pi_s)\phi(\zeta_{ijs})$.

The diagonal entries of the inverse expected information matrix are approximately equal to the square of relevant model parameter standard errors, with large sample sizes. However, in the presence of missing values, the observed information is arguably more appropriate as a measure of information content. This is because the observed information takes into account the number of missing values in the observed dataset, whereas the expected information does not. For more details, see Little and Rubin (2002).

1.3 A Specific Example

Consider a protein with m_i peptides in the diabetes data, where there are 10 samples each for diabetics and healthy controls. Let Y_{ijks} be the log peak intensity for the jth peptide of protein i in sample s of group k, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, m_i$, $k = 1, 2$, $s = 1, 2, \ldots, 10$. Based on model (1), μ_i is the overall mean for protein i, PEP_{ij} is the mean offset for the jth peptide in protein i, and GRP_{ik} is the effect of diabetes on mean protein expression. The following sum-to-zero constraints apply: $\sum_{j=1}^{m_i} \mathrm{PEP}_{ij} = 0$ and $\sum_{k=1}^{2} \mathrm{GRP}_{ik} = 0$. Note that we assume a common group effect for each peptide in the same protein.

In matrix form, the model is

$$
\boldsymbol{Y}_i = \boldsymbol{X_i}\boldsymbol{\beta_i} + \boldsymbol{\epsilon_i},
$$

where, for example, the model matrix for a protein with 3 peptides is

$$\boldsymbol{X_i} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{-1} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{-1} & \mathbf{-1} & \mathbf{1} \\ \mathbf{1} & \mathbf{-1} & \mathbf{-1} & \mathbf{-1} \end{pmatrix},$$

where each component of the matrix is a vector of length 10. There are therefore $3 + 4 = 7$ parameters to estimate numerically: $\boldsymbol{\delta_i} = (\delta_{i1}, \delta_{i2}, \delta_{i3})'$ and $\boldsymbol{\beta_i} = (\mu_i, \mathrm{PEP}_{i1}, \mathrm{PEP}_{i2}, \mathrm{GRP}_{i1})'$. The first derivative of the log-likelihood is a vector of length 7, equal to

$$\frac{\partial l_i}{\partial \boldsymbol{\theta_0}} = \left(\frac{\partial l_i}{\partial \boldsymbol{\delta_i}}, \frac{\partial l_i}{\partial \boldsymbol{\beta_i}}\right)' = \left(\frac{\partial l_i}{\partial \delta_{i1}}, \frac{\partial l_i}{\partial \delta_{i2}}, \frac{\partial l_i}{\partial \delta_{i3}}, \frac{\partial l_i}{\partial \mu_i}, \frac{\partial l_i}{\partial \mathrm{PEP}_{i1}}, \frac{\partial l_i}{\partial \mathrm{PEP}_{i2}}, \frac{\partial l_i}{\partial \mathrm{GRP}_{i1}}\right)'.$$

The second derivative is a 7×7 matrix equal to

$$\frac{\partial^2 l_i}{\partial \boldsymbol{\theta_0} \partial \boldsymbol{\theta_0}'} = \begin{pmatrix} \frac{\partial^2 l_i}{\partial \boldsymbol{\delta_i} \partial \boldsymbol{\delta_i}'} & \frac{\partial^2 l_i}{\partial \boldsymbol{\delta_i} \partial \boldsymbol{\beta_i}'} \\ \frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \boldsymbol{\delta_i}'} & \frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \boldsymbol{\beta_i}'} \end{pmatrix},$$

where $\frac{\partial^2 l_i}{\partial \boldsymbol{\delta_i} \partial \boldsymbol{\delta_i}'}$ is the 3×3 diagonal matrix with jth diagonal entry equal to $\frac{\partial^2 l_i}{\partial \delta_{ij}}$, $\frac{\partial^2 l_i}{\partial \boldsymbol{\delta_i} \partial \boldsymbol{\beta_i}'}$ is the 3×4 matrix with jth row equal to $\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \delta_{ij}}$, and $\frac{\partial^2 l_i}{\partial \boldsymbol{\beta_i} \partial \boldsymbol{\beta_i}'}$ is a 4×4 matrix.

2 Preprocessing

2.1 Rough Parameter Estimates

In both the filtering and imputation steps outlined below, we require rough estimates of the model parameters for making decisions on information content and generating random realizations under the estimated probability model. We do not employ the full numerical estimation routine of the censored likelihood for this purpose for several reasons: (1) Rough estimates are sufficient for the filtering and imputation steps, (2) The full censored likelihood model may not be identifiable if there are too many missing values, (3) Even if the full censored likelihood model is identifiable, with very little information content, the numerical algorithm may fail, and (4) The numerical algorithm is relatively slow.

Reviewing our model statement, our assumptions are: (1) there is a common group difference for each peptide from the same protein, (2) for any one peptide, the variances are equal in the difference comparison groups, although this variance may change from peptide to peptide, and (3) peak intensities for a peptide are Normally distributed. Consider a hypothetical protein with multiple peptides. If peptide j, say, is complete, with no missing values in any comparison group, we can use simple averaging of this peptide to obtain unbiased estimates of the overall peptide-level mean $\mathrm{PROT}_i + \mathrm{PEP}_{ij}$ and the corresponding group differences GRP_{ik} from model (1), $k = 1, 2, \ldots, K$. Similarly, we can obtain an unbiased estimate of δ_{ij} from the sample variances of the residuals. Since we assume common group differences for each peptide, the $\hat{\mathrm{GRP}}_{ik}$ estimates can be applied to all sibling peptides. For example, suppose peptide j' is complete in at least one of the comparison groups but also missing values in at least one comparison group. An unbiased estimate of the peptide-level mean for a group k, $\text{PROT}_i + \text{PEP}_{ij'} + \text{GRP}_{ik}$, in which the peptide is complete can be obtained by simple averaging as before. Similarly, we can estimate $\delta_{ij'}$ from the sample variance of the resulting residuals. We can then estimate all remaining group-specific means for peptide j' by shifting this estimate according to the $\hat{\text{GRP}}_{ik}$.

These parameter estimates are used in the filtering and imputation steps as described below.

2.2 Filtering

Our interest is in the protein-level group difference estimates, contained in the vector-valued parameter $\boldsymbol{\beta}_i$. Let $\boldsymbol{\beta}_{i,\texttt{GRP}}$ be the subset of $\boldsymbol{\beta}_i$ corresponding to these group difference estimates; this is just the result of removing the protein-level overall mean and peptide effect estimates from $\boldsymbol{\beta}_i$. Let $\boldsymbol{I}_0(\hat{\boldsymbol{\beta}}_{i,\texttt{GRP}})$ be the result of plugging $\hat{\boldsymbol{\beta}}_{i,\texttt{GRP}}$ into the corresponding sub-block of $\boldsymbol{I}(\boldsymbol{\beta}_i)$. We then quantify the information content $\mathcal{C}_i$ for protein i as the determinant of this matrix:

$$\mathcal{C}_i = |\boldsymbol{I}_0(\hat{\boldsymbol{\beta}}_{i,\texttt{GRP}})|.$$

Larger values of $\mathcal{C}_i$ correspond to smaller standard errors for the protein-level group effects and hence greater information content. Zero determinants correspond to non-identifiable models, as happens for example when no observations occur at all in one or more comparison groups. With this measure of information content, we can evaluate a protein, as well as any combination of its peptides, in terms of its ability to provide useful information about the protein-level group differences of interest. Proteins are first filtered if no combination of their peptides results in an identifiable model. For each protein that remains, a greedy search routine is employed to find the minimal subset of peptides required to achieve 90% of the total information content present for the protein. Specifically, the first peptide is chosen to individually maximize $\mathcal{C}_i$. Then, each remaining peptide is paired with the already-selected peptide, and the second peptide is chosen to maximize $\mathcal{C}_i$ in conjunction with the already-selected peptide. This is continued until 90% of the total information content, computed using all peptides for the protein, is attained.

2.3 Imputation

Imputation is carried out by simply generating missing values as random draws from our estimated model. Each missing value is chosen to have been censored with probability

$$P(Y_{ijs} < \hat{c}_{ij} | W_{ijs} = 1) = \frac{\Phi(\hat{\zeta}_{ijs})}{\hat{\pi}_s + (1 - \hat{\pi}_s)\Phi\big(\hat{\zeta}_{ijs}\big)},$$

where the denominator comes from the overall missingness probability in equation (2). If censored, the missing value is imputed with a random draw from the relevant Normal distribution, truncated at $\hat{c}_{ij}$. If not censored, the missing value is imputed with a random draw from the same Normal distribution, but with not truncation at the estimated censoring point. When there are no complete peptides for a protein, we randomly select the peptide that has the fewest missing values and use similar ad-hoc rules to make decisions when fleshing out the sibling peptides.

The above routine is an example of *single imputation.* Single imputation is known to suffer from overfitting, in that the uncertainty associated with using the data to estimate a model has not been taken into account. This means that we will end up underestimating standard errors of parameters as well as p-values. To address this, we could implement a *multiple imputation* routine. In multiple imputation, the above procedure would be repeated several times, each time with different random numbers generated from the model distributions. These repeated samples can be used to estimate the variability due to using the observed data for imputation.

Instead of multiple imputation, we directly adjust the p-values from our single imputation routine to appropriately reflect all sources of variability. This amounts to inflating the nominal p-values computed on the imputed data. The choice of how much to inflate the p-values is done automatically, based on the following argument. The p-values for null proteins should be uniformly distributed between 0 and 1 (Lehmann [1997]). Furthermore, in high-throughput significance testing applications, the overall distribution of p-values can be viewed as a mixture of this uniform distribution and a right-skewed distribution for the alternative features (genes, proteins, *etc.*) (Efron et al. [2001], Storey [2002]). Overfitting will generally result in underestimated p-values, causing among other things the null p-value distribution to be right-skewed and non-uniform (Dabney and Storey [2006]). If we knew how much the null p-value distribution has been skewed, we could just apply an appropriate scale factor, κ, to all p-values. That is, we could replace p_i, the p-value for the ith protein, with $p_i \times \kappa$ (restricted to not exceed 1). Since we can not separate the null and alternative p-values in practice, we can only hope to estimate κ. If we underestimate κ, we will continue to have underestimated p-values. If we overestimate κ, we will have *left*-skewed null p-values, with a peak on the right side of the overall p-value distribution. Based on this last observation, we consider a range of scale factors and choose the largest one that does not cause a peak on the right side of the overall p-value distribution. Specifically, we let κ range from 1 to 10, and use each value in turn as the scale factor for inflating the p-values. We then fit a least-square regression line to the resulting p-value distributions, over the range $[0.7, 1.0]$ (an arbitrary definition for the "right side" of the p-value distribution), and record its slope. A slope greater than one is indicative of our having inflated too much. We then choose the largest value of κ in our range of possible values that does not have a positive slope to this regression curve, as measured by a one-sided hypothesis test at level 0.05.

This routine produces approximately uniformly-distributed null p-values, as measured by Kolmogorov-Smirnoff tests, in the simulation studies reported in the main paper.

3 Simulation Study

The simulation data mimicked the diabetes data considered in the main manuscript. There were 174 proteins, and each simulated protein had the same number of peptides as in the diabetes data. The first 90 proteins were made to be not differentially expressed (with group differences terms set to zero), and the remaining 84 were differentially expressed (with group differences randomly drawn from the sequence $\{-1.1, -1, \ldots, -0.6, 0.6, 0.7, \ldots, 1.1\}$). The random missingness π_s parameters were set to 5%, and the censoring thresholds were selected such that a total of 40% of all measurements were missing. Protein means were randomly generated from the $N(20, 0.2)$ distribution (Normal distribution with mean 20 and standard deviation 0.2). Peptide effects were randomly generated from the $N(0, 0.6)$ distribution. Residual standard deviations were randomly generated from the Uniform distribution on the interval [0.6, 1.1]. These steps were done one time, creating all the parameters to be used repeatedly in five simulations. In each simulation, residual error was randomly generated and added to the model determined by the above parameters. Missingness was induced by censoring the lowest intensities and randomly selecting entries regardless of intensity to be missing. For each method, we computed a set of (1-specificity, sensitivity) values corresponding to a sequence of p-value cutoffs. We then averaged the values for the same p-value cutoff across simulations. These were plotted in the ROC curve figure in the main manuscript. All remaining simulations were carried out similarly and summarized in Table 1 in the main manuscript.

4 Analysis of Mutant Virulence Data

4.1 Model

Let Y_{ijks} be the intensity for protein i, peptide j, mutant group k, and replicate s, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, m_i$, $k = 1, 2, \ldots, K$, $s = 1, 2, \ldots, n$. Assume the model:

$$Y_{ijks} = \mathrm{WT}_i + \mathrm{PEP}_{ij} + \mathrm{MUT}_{ik} + \epsilon_{ijks}, \tag{3}$$

where WT_i is the protein-level mean for the WT group, the PEP_{ij} are the peptide effects (assumed to be constant across comparison groups), and the MUT_{ik} are the mutant effects. After filtering, we have 13 mutants, all of which need to be compared to WT. As usual, we require the sum-to-zero constraints on the PEP terms: $\sum_{j=1}^{m_i} \mathrm{PEP}_{ij} = 0$.

In matrix form, the model is $\boldsymbol{Y}_i = \boldsymbol{X}_i \boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i$. The covariance matrix of $\boldsymbol{\epsilon}_i$ is diagonal but with different variance parameters for each peptide. That is, $\mathrm{Var}(\boldsymbol{\epsilon}_i) = \Sigma_i$, where

$$\Sigma_i = \mathrm{BlockDiagonal}(\sigma_{i1}^2 I_{45\times 45},\ \sigma_{i2}^2 I_{45\times 45},\ \ldots,\ \sigma_{im_i}^2 I_{45\times 45}),$$

where $I_{45\times 45}$ is the 45×45 identity matrix. The "45" comes from the fact that we have 6 WT replicates and 3 replicates each for the WT and 13 mutant groups.

4.2 Estimation

Given the model matrix, we can estimate the model parameters using standard least squares

$$\hat{\boldsymbol{\beta}}_i = (\boldsymbol{X}_i' \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i' \boldsymbol{Y}_i.$$

Note that, since each peptide is allowed to have its own error variance, we would ideally do something more like a weighted regression, using an estimate of Σ_i. The above estimates are still unbiased, though. To estimate their standard errors, we can use the estimated covariance matrix of $\hat{\boldsymbol{\beta}}_i$:

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}_i) = (\boldsymbol{X}_i' \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i' \hat{\Sigma}_i \boldsymbol{X}_i (\boldsymbol{X}_i' \boldsymbol{X}_i)^{-1},$$

where

$$\hat{\Sigma}_i = \mathrm{BlockDiagonal}(\hat{\sigma}_{i1}^2 I_{45\times 45},\ \hat{\sigma}_{i2}^2 I_{45\times 45},\ \ldots,\ \hat{\sigma}_{im_i}^2 I_{45\times 45}),$$

and the $\hat{\sigma}^2_{ij}$ are computed from the peptide-specific model residuals:

$$\hat{\sigma}^2_{ij} = \frac{\sum_{k=1}^{14}\sum_{s=1}^{3}(y_{ijks} - \hat{y}_{ijks})^2}{45 - 1 - 14/45},$$

$j = 1, 2, \ldots, m_i$.

4.3 Inference

The interest in this experiment is in finding proteins that differ from WT in the majority of the mutants. That is, we would ideally like to find those proteins for which there is a significant difference in protein-level intensity for most of mutant 1, mutant 2, …, mutant 13. For protein i, we compute the mutant-specific test statistics:

$$T_{ik} = \frac{\hat{\text{MUT}}_{ik}}{\hat{\text{s.e.}}(\hat{\text{MUT}}_{ik})},$$

$k = 1, 2, \ldots, 13$, and then use

$$T_i = \# \left\{ |T_{ik}| \geq 2 : k = 1, 2, \ldots, 13 \right\},$$

the number of mutant-specific test statistics exceeding 2 in absolute value, as the protein-level test statistic.

To put a p-value on this statistic, we construct bootstrap samples from the null distribution as follows:

1. Compute the residuals from the fitted model in equation (1): e_{ijks}.
2. For each of $B = 500$ bootstrap iterations:
   (a) Take a random sample with replacement from the residuals: e^b_{ijks}.
   (b) Add the bootstrapped residuals to the estimated null model ($\hat{Y}_{ijks} = \hat{\text{PRO}}_i + \hat{\text{PEP}}_{ij}$) to get a bootstrapped sample under the null: y^b_{ijks}.
   (c) Compute the test statistic using the bootstrapped sample and the full model in equation (1) to obtain a bootstrapped null statistic: T^b_i.
3. Compute a p-value as the proportion of bootstrap statistics that exceed the observed statistic: $p_i = \# \left\{ T^b_i > T_i \right\} / B$.

References

A. R. Dabney and J. D. Storey. A reanalysis of a published affymetrix genechip control dataset. *Genome Biology*, 7:401, 2006.

B. Efron, R. Tibshirani, J. D. Storey, and V. Tusher. Empirical Bayes analysis of a microarray experiment. *J. Am. Statist. Assoc.*, 96: 1151–1160, 2001.

E.L. Lehmann. *Testing Statistical Hypotheses.* Springer, 1997.

J.D. Storey. A direct approach to false discovery rates. *J. R. Statist. Soc. B*, 64:479–498, 2002.