Supporting Information

Gerster et al. 10.1073/pnas.0907654107

SI Text

Assembling the Bipartite Graph. All experimentally identified peptides [in the given examples, we only considered peptides with a PeptideProphet (1) score above 0.9 if nothing else is mentioned] are present in the graph. In a first stage, the graph also holds all the protein sequences that match to at least one experimentally identified peptide sequence. An example for such an initial graph is shown in Fig. S1*A*.

In a first pruning step, we remove proteins if their set of matching peptides is a strict subset of another protein's set of matching peptides. In the given example, we remove proteins 2 and 6 because of this rule. The resulting graph is shown in Fig. S1*B*.

In the second pruning step, from Fig. S1 *B* to *C*, we remove proteins without unique evidence if all their matching peptides belong to at least one protein with unique evidence. A protein is said to have unique evidence if (*i*) at least one of its matching peptides is unambiguous or (*ii*) at least one of its matching peptides belongs only to proteins having the exact same set of matching peptides. In Fig. S1*B*, protein 1 is said to have unique evidence because of peptides 1 and 2. Proteins 4 and 5 are said to have unique evidence because of peptide 5. The resulting (final) graph, as it would be used as input for Markovian Inference of Proteins and Gene Models (MIPGEM), is shown in Fig. S1*C*. We removed protein 3, because it has no unique evidence and both of its peptides also match to proteins having unique evidence. In contrast, no protein is removed from a connected component such as the one illustrated in Fig. S2, because we have, at this stage, no further information that would allow us to decide which protein to neglect.

The proposed pruning of the graph is conceptually very simple and might be improved. However, the graphs resulting from our pruning approach are very similar to the ones obtained by ProteinProphet (2) by weighting the edges (which leads to implicit pruning, because some edge weights are too low to be taken into consideration by ProteinProphet).

Effect of the pruning. We carried out some analyses to assess how strongly the two pruning steps affect the final graph. Most of the pruning is done in the first step where proteins are removed from the graph if their set of matching peptides is a strict subset of another protein's set of matching peptides. This is illustrated in Table S1 for all datasets used in our study.

The second pruning step has very little effect on the graph of the analyzed datasets. However, we expect that it will help to split large connected components into smaller ones when dealing with higher eukaryotes.

Fig. S3 illustrates the effect of the pruning on the protein inference results. We show the results for the three datasets with known (or approximate) ground truth. In general, next to the effect of speeding up the computations, pruning helps to focus on a more promising set of proteins. The coverage of the true positives is not strongly affected by the pruning. The effect of pruning varies for the different datasets. In the case of *Saccharomyces cerevisiae* (Fig. S3 *E* and *F*) and the mixture of 18 proteins (Fig. S3 *A* and *B*), there is no substantial effect on the identification accuracy for low numbers of false positives, although pruning reduces the set of all considered proteins by about 4% or 59%, respectively. For the Sigma49 dataset (Fig. S3 *C* and *D*), pruning has a more pronounced effect. The set of considered proteins is reduced by about 78%. For all these figures, we should be aware that the compared sets of proteins (pruned and unpruned) are (very) different and that the comparison is not always meaningful.

Shared Peptides. In our model, shared peptides contribute to the inference of proteins. Their influence is split between the different matching proteins. The following two examples show how our new model deals with shared (degenerate) peptides. The formula to compute the protein probabilities is given in Eq. **S10**. In the first example, one peptide is matching a single protein (see Fig. S4*A*):

$$A_1(1) = \mathrm{p}(\{p_i\}|Z_1 = 1) \cdot \mathrm{p}(Z_1 = 1)$$
$$A_1(0) = \mathrm{p}(\{p_i\}|Z_1 = 0) \cdot \mathrm{p}(Z_1 = 0)$$

In the second example, the peptide is shared between two proteins (see Fig. S4*B*):

$$\begin{aligned} A_2(1) = {} & \mathrm{p}(\{p_i\}|Z_1 = 1,Z_2 = 0) \cdot \mathrm{p}(Z_1 = 1) \\ & \cdot \mathrm{p}(Z_2 = 0) + \mathrm{p}(\{p_i\}|Z_1 = 1,Z_2 = 1) \cdot \mathrm{p}(Z_1 = 1) \\ & \cdot \mathrm{p}(Z_2 = 1) \\ A_2(0) = {} & \mathrm{p}(\{p_i\}|Z_1 = 0,Z_2 = 0) \cdot \mathrm{p}(Z_1 = 0) \\ & \cdot \mathrm{p}(Z_2 = 0) + \mathrm{p}(\{p_i\}|Z_1 = 0,Z_2 = 1) \cdot \mathrm{p}(Z_1 = 0) \\ & \cdot \mathrm{p}(Z_2 = 1) \end{aligned}$$

$A_1(1)$ is equal to $A_2(1)$ (this still holds if the peptide matches more than two proteins). $A_1(0)$ is not equal to $A_2(0)$:

$$A_1(0) > A_2(0) \quad \text{for } p_i < \text{median (peptide scores)}$$
$$A_1(0) \leq A_2(0) \quad \text{for } p_i \geq \text{median (peptide scores)}$$

This leads to the following results for the probabilities:

$$\begin{aligned} & \mathbb{P}_1[Z_1 = 1|\{p_i;i \in \mathcal{I}\}] \geq \mathbb{P}_2[Z_1 = 1|\{p_i;i \in \mathcal{I}\}] \\ & \quad \text{for } p_i \geq \text{median (peptide scores)}; \\ & \mathbb{P}_1[Z_1 = 1|\{p_i;i \in \mathcal{I}\}] < \mathbb{P}_2[Z_1 = 1|\{p_i;i \in \mathcal{I}\}] \\ & \quad \text{for } p_i < \text{median (peptide scores)}; \\ & \mathbb{P}_1[Z_1 = 0|\{p_i;i \in \mathcal{I}\}] \leq \mathbb{P}_2[Z_1 = 0|\{p_i;i \in \mathcal{I}\}] \\ & \quad \text{for } p_i \geq \text{median (peptide scores)}; \\ & \mathbb{P}_1[Z_1 = 0|\{p_i;i \in \mathcal{I}\}] > \mathbb{P}_2[Z_1 = 0|\{p_i;i \in \mathcal{I}\}] \\ & \quad \text{for } p_i < \text{median (peptide scores)}. \end{aligned}$$

Note that $A_1(1) = A_2(1)$ because $\mathrm{p}(\{p_i\}|Z_1 = 1) = \mathrm{p}(\{p_i\}|Z_1 = 1,Z_2 = 0) = \mathrm{p}(\{p_i\}|Z_1 = 1,Z_2 = 1)$. On the other hand, $\mathrm{p}(\{p_i\}|Z_1 = 0) \neq \mathrm{p}(\{p_i\}|Z_1 = 0,Z_2 = 1)$ and hence $A_1(0) \neq A_2(0)$.

Note as well that $\mathbb{P}_2[Z_1 = 1|\{p_i;i \in \mathcal{I}\}] = \mathbb{P}_2[Z_2 = 1|\{p_i;i \in \mathcal{I}\}]$ and $\mathbb{P}_2[Z_1 = 0|\{p_i;i \in \mathcal{I}\}] = \mathbb{P}_2[Z_2 = 0|\{p_i;i \in \mathcal{I}\}]$.

Log-Likelihood. Our assumptions are used to write down the log-likelihood of the probability distribution of the peptide scores:

$$\ell = \log(\mathrm{p}(\{p_i; i \in \mathcal{J}\})) = \log\left(\prod_{r=1}^{R} \mathrm{p}(\{p_i; i \in \mathcal{J}_r\})\right)$$
$$= \sum_{r=1}^{R} \log(\mathrm{p}(\{p_i; i \in \mathcal{J}_r\}))$$
$$= \sum_{r=1}^{R} \log\left(\sum_{\substack{z_j \in \{0,1\} \\ j \in \mathcal{R}(\mathcal{J}_r)}} \prod_{i \in \mathcal{J}_r} \mathrm{p}(p_i | \{z_j; j \in Ne(i)\}) \cdot \prod_{j \in \mathcal{R}(\mathcal{J}_r)} \mathrm{p}(z_j)\right). \quad \textbf{[S1]}$$

Assuming that the prior probabilities are given, the log-likelihood becomes a function of the single unknown parameter b_1, i.e., $\ell = \ell(b_1)$, using the constraints

$$\int_l^u f_1(x)dx = 1, \qquad b_1 > 0,\ b_2 \geq 0. \quad \textbf{[S2]}$$

The optimal values for the parameters of the probability density function are then given by

$$\hat{b}_1 = \arg\min_{b_1} - \ell(b_1) \quad \textbf{[S3]}$$

$$\hat{b}_2 = \frac{2 - \hat{b}_1(u-l)^2}{(u-m)^2}, \quad \textbf{[S4]}$$

where $l = \min_i(p_i)$, $m = \mathrm{median}_i(p_i)$, and $u = \max_i(p_i)$.

The minimization to obtain $\hat{b}_1$ is done by using the R-function `optimize`. The following arguments were used in `optimize`: `lower` $= 0$, `upper` $= 0.1$, and `tol` $= 10^{-3}$. A list of the used R packages is provided in the section *Computational Details*. The upper constraint 0.1 is without loss of generality, because we considered only peptides with PeptideProphet (1) scores above 0.9.

As a default, we estimate a "prior" probability $p(z_j) \equiv \pi$ by choosing π such that the negative log-likelihood $-\ell$ is minimized [as a function of b_1 and π, i.e., $\ell = \ell(b_1, \pi)$]. We (approximately) pursue this task by considering some candidate values for π on a grid with grid-points from 0.05 to 0.95 by steps of 0.05.

Protein Probabilities. Our goal is to compute the probability that a protein j is present given the peptide scores $\mathbb{P}[Z_j = 1 | \{p_i; i \in \mathcal{J}\}]$. The property

$$\mathbb{P}[Z_j = 0 | \{p_i; i \in \mathcal{J}\}] + \mathbb{P}[Z_j = 1 | \{p_i; i \in \mathcal{J}\}] = 1 \quad \textbf{[S5]}$$

must hold. The probability of a protein being present in the sample given the peptide scores can then be computed as follows. Denote by $d(j)$ the index of the connected component holding the protein j. Then

$$\mathbb{P}[Z_j = 1 | \{p_i; i \in \mathcal{J}\}] = \mathbb{P}[Z_j = 1 | \{p_i; i \in \mathcal{J}_{d(j)}\}]$$
$$= \sum_{\substack{z_k \in \{0,1\} \\ k \in \mathcal{R}(\mathcal{J}_{d(j)}) \\ k \neq j}} \mathbb{P}[Z_j = 1, Z_k = z_k | \{p_i; i \in \mathcal{J}_{d(j)}\}]$$
$$= \sum_{\substack{z_k \in \{0,1\} \\ k \in \mathcal{R}(\mathcal{J}_{d(j)}) \\ k \neq j}} \left(\frac{1}{\mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\})} \cdot \mathrm{p}(Z_j = 1, Z_k = z_k)\right.$$
$$\left. \cdot\, \mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\} | Z_j = 1, Z_k = z_k)\right)$$
$$= \frac{A(1)}{\mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\})}, \quad \textbf{[S6]}$$

with the function $A(z)$ defined as

$$A(z) = \sum_{\substack{z_k \in \{0,1\} \\ k \in \mathcal{R}(\mathcal{J}_{d(j)}) \\ k \neq j}} [\mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\} | Z_j = z, Z_k = z_k) \cdot \mathrm{p}(Z_j = z)$$
$$\cdot \prod_{\substack{k \neq j \\ k \in \mathcal{R}(\mathcal{J}_{d(j)})}} \mathrm{p}(Z_k = z_k)]. \quad \textbf{[S7]}$$

The probability $\mathbb{P}[Z_j = 0 | \{p_i; i \in \mathcal{J}\}]$ can be computed analogously to $\mathbb{P}[Z_j = 1 | \{p_i; i \in \mathcal{J}\}]$:

$$\mathbb{P}[Z_j = 0 | \{p_i; i \in \mathcal{J}\}] = \frac{A(0)}{\mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\})}.$$

With the property in Eq. **S5**, we can write

$$\frac{A(0) + A(1)}{\mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\})} = 1 \Rightarrow \frac{1}{\mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\})} = \frac{1}{A(0) + A(1)} \quad \textbf{[S8]}$$

and hence

$$\frac{A(1)}{\mathrm{p}(\{p_i; i \in \mathcal{J}_{d(j)}\})} = \frac{A(1)}{A(0) + A(1)}, \quad \textbf{[S9]}$$

which leads to the formula

$$\mathbb{P}[Z_j = 1 | \{p_i; i \in \mathcal{J}\}] = \frac{A(1)}{A(0) + A(1)}. \quad \textbf{[S10]}$$

Sampling for Large Connected Components. The computational effort for the maximum likelihood parameter estimation and for the computation of the protein probabilities for connected components $\mathcal{J}_r$ with many proteins is considerable. We have to sum over all possible protein configurations, accounting for the two possible states of each protein, namely absent or present in the sample (see formulas **S7** and **S1**). For n proteins, this means that we have 2^n summands. This summation is reasonably fast for connected components with up to $n \approx 10$ proteins. A workaround is needed if there are more proteins.

In all datasets presented in the manuscript, most connected components do not hold more than 10 proteins. The presented approximation is actually only used for one connected component in the Sigma49 dataset, one in the yeast dataset and two in the *Arabidopsis thaliana* dataset. The concerned connected components hold 12, 11, 17, and 18 proteins, respectively.

The expressions to be computed by using a workaround are of the form $\sum f(x)\mathrm{p}(X = x)$, where the sum goes over all x, see Eqs. **S7** and **S1**, which is equal to $\mathbb{E}[f(x)]$ since x is discrete. Therefore, these sums can be estimated by random sampling.

As an example, we will look at

$$\sum_{\substack{z_j \in \{0,1\} \\ j \in \mathcal{R}(\mathcal{J}_r)}} \mathrm{p}(\{p_i; i \in \mathcal{J}_r\} | \{z_j; j \in \mathcal{R}(\mathcal{J}_r)\}) \cdot \mathrm{p}(\{z_j; j \in \mathcal{R}(\mathcal{J}_r)\})$$
$$= \mathbb{E}[f(\{z_j; j \in \mathcal{R}(\mathcal{J}_r)\})],$$

with $f(\{z_j; j \in \mathcal{R}(\mathcal{J}_r)\}) = \mathrm{p}(\{p_i; i \in \mathcal{J}_r\} | \{z_j; j \in \mathcal{R}(\mathcal{J}_r)\})$. To compute this expectation, we proceed as follows:

1. Sample $\{z_j; j \in \mathcal{R}(\mathcal{J}_r)\}$. This gives one possible protein configuration. $z_j \in \{0,1\}$ with $\mathbb{P}[Z_j = 1] = \pi$ and $\mathbb{P}[Z_j = 0] = 1 - \pi$ (π stands for the protein prior).
2. Compute $S^{(1)} = f(\{z_j; j \in \mathcal{R}(\mathcal{J}_r)\})$.
3. Repeat steps 1 and 2 B times ($B = 2^{10}$).

4. Approximate the expectation:

$$\mathbb{E}[f(\{z_j; j \in \mathcal{R}(\mathcal{I}_r)\})] \approx \frac{1}{B}\sum_{b=1}^{B} S^{(b)}$$

This approach is used for the parameter optimization and to compute the probabilities of the proteins and of the gene models. The only change from case to case is the function $f(\cdot)$.

Gene Model Probabilities. A distinguishing feature of MIPGEM is that it also considers the relationship between the gene models and the protein sequences in addition to the relation between peptide and protein sequences. It can thus be seen as a special form of a tripartite graph (see Fig. 2).

First we consider the case where a gene model has only neighboring protein sequences belonging to the same connected component of the peptide-protein graph. To compute the probability of the gene model X being present in the sample we use

$$\mathbb{P}[X = 1|\{p_i; i \in \mathcal{I}\}] = 1 - \mathbb{P}\left[\bigcap_{j\in\mathcal{R}(X)} \{Z_j = 0\}|\{p_i; i \in \mathcal{I}_{r(X)}\}\right], \quad \textbf{[S11]}$$

where $\mathcal{R}(X)$ is the range of X (all the proteins j such that there exists an edge between j and X) and $\mathcal{I}_{r(X)}$ stands for all the peptides in the same connected component as the proteins belonging to X. Note that $\mathcal{R}(X) \subseteq \mathcal{R}(\mathcal{I}_{r(X)})$. Then,

$$\begin{aligned}\mathbb{P}&\left[\bigcap_{j\in\mathcal{R}(X)} \{Z_j = 0\}|\{p_i; i \in \mathcal{I}_{r(X)}\}\right] \\ &= \sum_{\substack{z_k\in\{0,1\}\\ k\in\mathcal{R}(\mathcal{I}_{r(X)})\setminus\mathcal{R}(X)}} \mathbb{P}[\{\{Z_j = 0\,\forall\, j \in \mathcal{R}(X)\} \\ &\quad \cap \{Z_k = z_k\}\}|\{p_i; i \in \mathcal{I}_{r(X)}\}]. \quad \textbf{[S12]}\end{aligned}$$

Generalization. If the gene model X corresponds to proteins from different connected components of the peptide-protein graph, we can proceed as follows:

$$\begin{aligned}\mathbb{P}&\left[\bigcap_{j\in\mathcal{R}(X)} \{Z_j = 0\}|\{p_i; i \in \mathcal{I}\}\right] \\ &= \prod_{\ell=1}^{m} \mathbb{P}\left[\bigcap_{j\in\mathcal{R}_\ell(X)} \{Z_j = 0\}|\{p_i; i \in \mathcal{I}_\ell(X)\}\right], \quad \textbf{[S13]}\end{aligned}$$

where m is the number of peptide-protein connected components having neighboring proteins to the gene model X and $\mathcal{R}_\ell(X)$ are the neighboring proteins of X in the connected component ℓ. Note that $\mathcal{R}(X) = \mathcal{R}_1(X) \cup \mathcal{R}_2(X) \cup \ldots \cup \mathcal{R}_m(X)$. The factors in the product can be computed as shown in Eq. **S12**.

Implementation. Our model assumptions and Bayes' law are used to compute the gene model probabilities. Eq. **S12** (or the generalization in **[S13]**) can be rewritten as

$$\begin{aligned}&\sum_{\substack{z_k\in\{0,1\}\\ k\in\mathcal{R}(\mathcal{I}_{\ell(X)})\setminus\mathcal{R}(X)}} \mathbb{P}[\{\{Z_j = 0\,\forall\, j \in \mathcal{R}_\ell(X)\} \\ &\quad \cap \{Z_k = z_k\}\}|\{p_i; i \in \mathcal{I}_{\ell(X)}\}] = \frac{1}{\mathrm{p}(\{p_i; i \in \mathcal{I}_{\ell(X)}\})} \\ &\quad \cdot \sum_{\substack{z_k\in\{0,1\}\\ k\in\mathcal{R}(\mathcal{I}_{\ell(X)})\setminus\mathcal{R}(X)}} \Big\{\mathrm{p}(\{p_i; i \in \mathcal{I}_\ell(X)\}|\{\{Z_j = 0\,\forall\, j \in \mathcal{R}_\ell(X)\} \\ &\quad \cap \{Z_k = z_k\}\}) \cdot \prod_{j\in\mathcal{R}_\ell(X)} \mathrm{p}(Z_j = 0) \prod_{k\in\mathcal{R}(\mathcal{I}_{\ell(X)})\setminus\mathcal{R}(X)} \mathrm{p}(Z_k = z_k)\Big\}\end{aligned}$$

which can then be written as

$$\begin{aligned}&\frac{1}{A(0)+A(1)} \cdot \sum_{\substack{z_k\in\{0,1\}\\ k\in\mathcal{R}(\mathcal{I}_{\ell(X)})\setminus\mathcal{R}(X)}} \Big\{\mathrm{p}(\{p_i; i \in \mathcal{I}_\ell(X)\}|\{Z_j = 0, \\ &\quad Z_k = z_k\,\forall\, j,k \in Ne(i)\}) \cdot \prod_{j\in\mathcal{R}_\ell(X)} \mathrm{p}(Z_j = 0) \\ &\quad \cdot \prod_{k\in\mathcal{R}(\mathcal{I}_{\ell(X)})\setminus\mathcal{R}(X)} \mathrm{p}(Z_k = z_k)\Big\},\end{aligned}$$

where $A(0)$ and $A(1)$ are computed according to Eq. **S7**. Any protein from the gene model can be chosen to compute $A(0)$ and $A(1)$.

For large connected components the same sampling idea is used as for the protein probabilities (see *Sampling for Large Connected Components*).

Additional Figures and Tables. Figs. S5 *A–C* present the same curves as Fig. 3 in our manuscript, except that the identified single hits are discarded [in MIPGEM as well as in the reference methods: ProteinProphet (2) and MSBayesPro (3)].

Figs. S6 *A* and *B* illustrate the distribution of the computed protein scores for MIPGEM and for ProteinProphet for the dataset from *Drosophila melanogaster*. We sorted the protein scores and plotted the score against the index. The distribution illustrates one of the major differences between ProteinProphet and MIPGEM. ProteinProphet gets a bulk of proteins with a "perfect" score of one. This implies that ProteinProphet cannot differentiate or rank among the top inferred proteins. MIPGEM, on the other hand, provides a fine ranking among the protein scores and can thus be used to find a conservative cutoff for the protein probabilities.

Fig. S7 illustrates the influence of the cutoff for the peptide scores on the protein inference. We show this influence for the three datasets with known (or approximate) ground truth. No trend is recognizable. We cannot say that with increasing/decreasing cutoff the inference gets better or worse. Fig. S7 *A* and *B* show the results for the mixture of 18 proteins, Fig. S7*C* and *D* illustrate the results for the Sigma49 dataset, and Fig. S7 *E* and *F* are from the *S. cerevisiae* dataset. Fig. S7 *A*, *C*, and *E* present the results when keeping the single hits. In Fig. S7 *B*, *D*, and *F*, the single hits were discarded.

Fig. S8 illustrates the use of proteomics data to identify different protein splice isoforms that are encoded by one gene model. Compared to ProteinProphet, our approach, which relies on a tripartite graph, has the advantage to compute not only probabilities for the proteins, but also for their encoding gene model.

Table S2 shows the overlap between the n best scoring proteins from MIPGEM with (*i*) the set of 167 proteins identified with a score of 1 by both reference methods, (*ii*) the set of 217 sequences from ProteinProphet with a score of one, and (*iii*) the set of 194 proteins which got a score of one in MSBayesPro. The identified single hits were discarded in all methods. Note that some of our top-scoring proteins are neither identified by ProteinProphet nor by MSBayesPro with top scores.

Comparison with Other Protein Inference Models. We conceptually compare our model with three other methods for protein inference: a nested mixture model (4), a hierarchical statistical model (5), and MSBayesPro (3) (for the latter, we also include empirical comparisons). The first two approaches reassess peptide scores and estimate protein scores, with a strong focus on peptides, whereas MIPGEM and MSBayesPro (mainly) deal with protein inference.

Nested Mixture Model for Protein Inference. The modeling approach by Li et al. (4) is tailored to simultaneously reassess the peptide scores and infer the proteins. Their focus and results are mostly on better peptide identification, as pointed out in ref. 4 (sections 1 and 3). For this purpose, they make use of the fact that the presence or absence of a protein has implications on all its matching peptides (without taking into consideration the issue of shared peptides). Li et al. (4) (section 2.5) incorporate a few additional features of identified peptide sequences such as the number of tryptic termini and the number of missed tryptic cleavages. Such features (e.g., number of tryptic termini) are also integrated in the PeptideProphet probabilities which we used as input to MIPGEM.

For peptide inference with simpler organisms such as yeast (where there is less "degeneracy," see below), the model in ref. 4 seems to perform very well. For protein inference, which is the goal in our paper, the findings in ref. 4 are less conclusive, as the authors themselves point out in their abstract.

From the modeling perspective, the main difference to our approach is the treatment of shared peptides: Li et al. (4) acknowledge that they are not really dealing with this issue (called the "degeneracy problem"), whereas MIPGEM is tailored to address this important problem which occurs frequently in higher organisms. Unlike other models, ours is incorporating a much more flexible structure for dependence of observed peptide sequences, using a Markov assumption on graphs. Li et al. (4, section 2.2) describe the crucial issue of modeling dependence, and our approach goes a substantial step further in this respect. As a consequence, the stronger the degree of shared peptides (or the degree of degeneracy), the stronger our model and its results will differ from others. For example, Li et al. (4, section 3.3) use an ad hoc rule (to match groups from ProteinProphet) for dealing with the issue of shared peptides. In addition, our third layer for inferring gene models is motivated by identifiability problems which are particularly present in organisms with many shared peptides: For example, our *A. thaliana* dataset exhibits many more shared peptides than, say, yeast which has been analyzed by ref. 4.

Hierarchical Statistical Model (HSM) for Protein Inference. Shen et al. (5) present a four-layer hierarchical model for peptide and protein inference by considering also additional layers for assignment of peptide scores. They use an expectation-maximization algorithm to infer the parameters of their model over all connected components. In contrast, our model is structured as a k-partite graph with a Markov assumption and the optimization is performed on the level of clearly defined connected components.

In contrast to the approach of Li et al. (4), the model proposed by Shen et al. (5) accounts for degenerate peptides. However, this seems to be modeled/implemented in a computationally inefficient way. Li et al. (4) report that they were not able to compare their results with HSM because of computation and memory problems and argue why their approach is an improvement over ref. 5.

Analogously to what we wrote in the previous section, Shen et al. (5) have a much simpler model for dependence than our Markovian framework on graphs. Their paper also presents results on simpler organisms only exhibiting a low amount of shared peptides.

MSBayesPro. In contrast to MIPGEM and to ProteinProphet (2), MSBayesPro (3) includes peptide detectabilities to infer proteins.

Technicalities.
We used MSBayesPro according to the README file provided under http://darwin.informatics.indiana.edu/yonli/proteininfer/.

We followed the procedure below:

1. Crawl the predicted peptide detectabilities from http://darwin.informatics.indiana.edu/applications/PeptideDetectabilityPredictor/.
2. Run MSBayesPro a first time to estimate the protein priors.
3. Run MSBayesPro a second time including the computed priors to estimated the probabilities for each protein being in the sample.
4. Analyze the results: Each protein is identified with a probability of Positive_Probability_by_memorizing if and only if MAP_state_by_Memorizing is one.

The experimental data contain some nontryptic peptides. Because the tool to compute the peptide detectabilities only predicts scores for tryptic peptides, we added the nontryptic ones by hand to the detectability file (generated in the first step of the procedure above). We assigned arbitrary low detectability scores to these peptides [median(predicted detectability scores)/3].

Differences Between MSBayesPro and MIPGEM.
Li et al. (3) develop another approach, called MSBayesPro, for modeling the posterior distribution of presence/absence of proteins given the peptide scores within a connected component of a bipartite graph (see, e.g., Fig. 1). This basic step is similar to ProteinProphet's and our approach. There are, however, two main differences between MSBayesPro and MIPGEM. (*i*) The model underlying MSBayesPro does not allow for the flexibility of unknown parameters, whereas our method involves estimation of two parameters (differing) for each experiment. (*ii*) MSBayesPro uses peptide detectabilities as an additional source of data, whereas MIPGEM does not involve peptide detectabilities. We remark that the inclusion of peptide detectabilities in MSBayesPro is essentially noninformative: We show in Fig. S9 that we obtain almost exactly the same results when using MSBayesPro *without* inclusion of peptide detectabilities. To use MSBayesPro without detectabilities, we set all d_{ij}s to a constant value.

MSBayesPro and our method both have to deal with conditional probability distributions for all peptide scores given presence or absence of all matching proteins in a connected component of the bipartite graph as illustrated in Fig. 1. In our notation, this conditional distribution is

$$\mathrm{p}(\{p_i; i \in \mathcal{I}_r\} | \{z_j; j \in \mathcal{R}(\mathcal{I}_r)\}).$$

Both modeling approaches break up this conditional probability assuming conditional independence of the peptides given all corresponding proteins, i.e.,

$$\mathrm{p}(\{p_i; i \in \mathcal{I}_r\} | \{z_j; j \in \mathcal{R}(\mathcal{I}_r)\}) = \prod_{i \in \mathcal{I}_r} \mathrm{p}(p_i | \{z_j; j \in \mathcal{R}(\mathcal{I}_r)\}).$$

Both methods then proceed with some specific modeling of

$$\mathrm{p}(p_i | \{z_j; j \in \mathcal{R}(\mathcal{I}_r)\}),$$

which is in general a very high-dimensional quantity because the number of different states in the conditioning set is $2^{|\mathcal{R}(\mathcal{I}_r)|}$.

Li et al. (3) assume that

$$\mathrm{p}(p_i | \{z_j; j \in \mathcal{R}(\mathcal{I}_r)\}) = 1 - \prod_{j \in \mathcal{R}(\mathcal{I}_r)} (1 - z_j d_{ij}), \qquad \textbf{[S14]}$$

where $d_{ij} \in [0,1)$ are parameters, see formulas 4 and 5 in Li et al., which we rewrote to correspond to our notation. The form of the distribution in **[S14]** *cannot* be derived assuming some independence assumptions (as claimed in Li et al. before their equation 4). We have to view it (at best) as a (unusual and not clearly motivated) model simplifying the more general term $p(p_i | \{z_j; j \in \mathcal{R}(\mathcal{I}_r)\})$. An unusual property of the formula in **[S14]** is that all proteins in the connected component contribute to the peptide probability: In particular, a protein j contributes to a peptide i's probability even if there is no corresponding edge

between i and j in the bipartite graph. The parameters d_{ij} are then determined via (normed) peptide detectabilities which is a very pragmatic approach. These parameters are *not* estimated from fitting the data to the proposed model. We show in Fig. S9 that trivial choices of the parameters such as $d_{ij} \equiv 0.5$ for all i, j lead to almost the same results as compared to using the predicted peptide detectabilities. This surprising fact is likely due (*i*) to the difficulty to predict the parameters d_{ij}, and (*ii*) to the fact that the model in MSBayesPro is not efficiently incorporating this additional source of information. In contrast, our method reduces the high-dimensional state space by assuming a Markov assumption saying that

$$\mathrm{p}(p_i|\{z_j; j \in \mathscr{R}(\mathscr{I}_r)\}) = \mathrm{p}(p_i|\{z_j; j \in Ne(i)\},$$

see formula **3**. Then, this quantity is further modeled by using a two-component mixture model with parameters b_1 and b_2 (see *Probability Mixture Distribution for the Peptide Scores* in the manuscript) which are estimated by maximum likelihood estimation, fitting the data to the model.

In our approach, it is important that the two-component mixture model is a reasonable approximation. However, the flexibility to choose two parameters b_1 and b_2 (i.e., estimating them from data) makes such an approximation more realistic and powerful: b_1 and b_2 are not global parameters but vary among different datasets (and they are much more identifiable than the d_{ij} parameters in MSBayesPro). Finally, our method is based on peptide scores only and *not* relying on some other source of data, like peptide detectabilities for determining or estimating the model parameters (but see also Fig. S9 showing that peptide detectabilities are essentially uninformative when using them in MSBayesPro).

Additional Information About the Datasets. All data used in our examples have been previously published and are available at the sources mentioned in Table S3. Note that the *A. thaliana* data we tested our method with is part of a larger group of experiments available under the given accession numbers. However, the corresponding data repository associates one peptide only with one protein. Therefore, the shared peptides could not be uploaded. For convenience and to make sure that there is no confusion regarding the used data, we provide our input data files to all three models (ProteinProphet, MSBayesPro, and MIPGEM) for each of the analyzed datasets upon request.

The MS/MS data for the datasets were searched with TurboSEQUEST (6) against the respective protein database. Peptide validation was done with PeptideProphet (1) [Trans-Proteomic Pipeline (TPP) ver. 4.0]. The often used and highly cited ProteinProphet (2) (TPP ver. 4.0) was used as the reference method to infer proteins from the scored peptides. In addition, we also included the results from MSBayesPro (3) in the empirical comparison.

Mixture of 18 purified proteins. The first test dataset is a mixture of 18 highly purified proteins from different species including bovine (*Bos taurus*), chicken (*Gallus gallus*), rabbit (*Oryctolagus cuniculus*), *Escherichia coli*, horse (*Equus caballus*), yeast (*S. cerevisiae*) and *Bacillus licheniformis*. For more details about this synthetic sample we refer to ref. 7.

The MS/MS data was searched with SEQUEST by Keller et al. (7), using a database consisting of 88,377 sequences representing the 18 searched proteins as well as human protein sequences. We did the postprocessing with PeptideProphet.

For MIPGEM, the generated bipartite graph holds 265 peptides and 60 matching proteins (after the pruning steps). The nodes are connected by 332 edges and the graph decomposes into 33 connected components.

A prior probability of 0.35 (the same for all proteins) was estimated for our model.

The used list of true positives, as described in the original publication, includes also the alternative protein identifiers for rabbit myosin. For *B. lichenformis* α-amylase both `SW:AMY_BACLI` and `sp|Q04977|AMYM_BACLI` are included in the list of true positives, although the observed peptide hits are from `SW:AMY_BACLI`. For true proteins, see Table S4.

The contaminants include three casein proteins flagged as contaminants by the authors of the dataset as well as a few keratins and other well-known contaminants; see Table S5.

Not all proteins in the synthetic samples were detected by the experimentally identified peptides. For the mixture of 18 purified proteins, only 19 out of 27 proteins can be inferred. Therefore, neither the reference methods nor MIPGEM are able to find all the proteins in the sample. These undetected proteins are not counted as false negatives. The fact that we could not identify them may be due to a problem of peptide detectability (see, for example, ref. 8), or it might be due to the low concentration of some proteins in the samples.

Sigma49. Sigma49 is a mixture of 49 human proteins from Sigma Aldrich. We refer to refs. 9 and 10 for more details.

The output from the MS/MS pipeline is available online. We searched the data with SEQUEST ($pep\_mass\_tol = 3$, $mass\_type = 1$ (monoisotopic), $max\_cleavages = 2$) using release 51.0 (Oct. 31, 2006) of UniProtKB/Swiss-Prot containing 241,242 sequences. We did the postprocessing with PeptideProphet.

For MIPGEM, the generated bipartite graph holds 508 peptides and 169 matching proteins (after the pruning steps). The nodes are connected by 888 edges and the graph decomposes into 73 connected components.

A prior probability of 0.3 (the same for all proteins) was estimated for our model.

The list of true proteins is given in Table S6. The contaminants include keratins and other known contaminants, classified as such based on their protein accession description or their sequence; see Table S7.

As mentioned in the previous section, not all proteins in the synthetic samples were detected by the experimentally identified peptides. In the Sigma49 dataset 47 out of the 49 protein sequences include at least one experimentally identified peptide sequence.

Drosophila melanogaster dataset. These data originate from a Golgi fraction prepared from the embryonal Kc 167 cell line from *D. melanogaster*. For details we refer to ref. 11.

The output from the MS/MS was searched with TurboSEQUEST (ver. 27, rev. 12) with the following parameters: $pep\_mass\_tol = 3$, $mass\_type = 0$ (average), $mass\_cleavages = 1$, using the release 5.2 from Flybase with 20,726 entries as well as their reverse decoy sequences and 256 well-known contaminants. Peptide validation was done with PeptideProphet.

The generated tripartite graph of MIPGEM holds 1,831 peptides, 863 matching proteins, and 687 gene models (after the pruning steps). The peptide and protein nodes are connected by 2,642 edges. The proteins are connected to the gene models by 908 additional edges. The graph decomposes into 621 connected components. A prior probability of 0.65 (the same for all proteins) was estimated for our model.

For this dataset, the true proteins are not known. The set of contaminants was composed of 256 proteins including human keratins and other contaminants. It was used for the peptide identification. For the protein inference, only identified peptides matching to a *D. melanogaster* protein sequence were used.

Saccharomyces cerevisiae dataset. Several proteomics datasets are available for wild-type yeast cells that were grown in rich medium to log-phase. A compilation of eight experiments (contributed by different groups) is provided at http://www.marcottelab.org/MSdata/gold_yeast.html.

Intersections of proteins identified by several experiments are provided. These intersections can be used as an approximate reference dataset as to which proteins are expressed in *S. cerevisiae* under this specific condition.

For our analysis, we considered proteins belonging to either at least two of the four MS-based datasets (excluding the yeast Orbitrap data) or to any of the three non-MS-based datasets to be true identifications. This leads to a set of 4,265 proteins, corresponding to 4,230 unique protein sequences, for which experimental evidence has been accumulated. Based on this information, we assume that these 4,230 sequences represent true positives if they are identified. Conversely, the remaining 2,401 unique protein sequences in the yeast database are assumed to represent false positives. No contaminants were taken into consideration.

A dataset of wild-type yeast grown in rich medium and harvested in log-phase is also available on this Web page (yeast Orbitrap data). We used the provided data, already postprocessed with PeptideProphet (TPP ver. 4.0), as testing set, i.e., input for MIPGEM. For details about the data, we refer to the Web page mentioned above.

The SEQUEST search for the peptides was performed against the yeast database (Saccharomyces Genome Database; 6,714 proteins corresponding to 6,331 unique sequences; April 2006) without including any contaminants. Therefore, we did not consider contaminants for the protein inference step either.

The bipartite graph used for MIPGEM holds 6,988 peptides and 1,542 matching proteins (after the pruning steps). The nodes are connected by 7,809 edges and the graph decomposes into 1,436 connected components (all of them being very small in terms of numbers of proteins).

A prior probability of 0.5 (the same for all proteins) was estimated for our model.

We considered this *S. cerevisiae* dataset, because working only on the two small synthetic samples seemed to be too far away from reality. The main criticism toward these control datasets are (*i*) their size (small number of proteins) and (*ii*) the discrepancy between the sample size and the size of the database used for the sequence matching (already on the peptide identification level). However, larger datasets with a reliably known ground truth do not exist. Thus, we opted for the yeast dataset with the approximate ground truth from the intersection of other experiments. Nevertheless, there are some shortcomings to this validation as well:

- There is no certainty that the 4,230 protein sequences used as ground truth correspond to the "absolute truth." This set is a combination of the results from several experiments. It could very well contain wrong identifications or not include all truly expressed proteins.
- Although the set of assumed true positives is large (4,230 sequences), we can only identify up to 1,400 of them with the given set of identified peptides (no matching peptides were found for the other sequences).
- The amount of shared peptides is quite low in this dataset. A statistical model is especially needed if there are many shared peptides. Thus, this validation dataset has also a "toy" character, namely, in terms of difficulties dealing with many shared peptides.

Arabidopsis thaliana dataset. The aim is to be able to use MIPGEM on organisms with higher percentages of shared peptides, namely, on higher eukaryotes (including plants) where a large percentage of the genome arose from genome duplication events. The *A. thaliana* pollen dataset belongs to this category of data.

Several published proteomics datasets are available for *A. thaliana* pollen. They can be used to build an approximate ground truth for the gene models that are actively expressed in *A. thaliana* pollen. In our case, our approximate ground truth relies upon seven (out of eight) proteomics experiments, several transcriptomics datasets from different laboratories, one non-MS 2D-gel proteomics experiment, and a literature mining dataset of roughly 100 genes that, when mutated, are known to affect pollen development. For details, we refer to ref. 12. As a testing set, we used the eighth proteomics experiment.

For our analysis, we considered gene models to be true identifications if they fulfilled at least one of the two following rules: (*i*) the gene model was identified by at least two of the seven MS-based datasets; (*ii*) the gene model was identified by at least one non-MS-based dataset and at least one MS-based dataset. Based on this experimental evidence, we assume that these 4,580 gene models represent true positives. Conversely, identified gene models that do not belong to this list are assumed to represent false positives (conservative approach).

The testing set was searched with TurboSEQUEST (ver. 27, rev. 12) with the following parameters: *pep_mass_tol* = 3, *mass_type* = 0 (average), *mass_cleavages* = 1, using release TAIR7 from TAIR with 31,921 entries as well as their reverse decoy sequences and 256 well-known contaminants. Peptide validation was done with PeptideProphet. For details we refer to Grobei et al. (12)

The generated tripartite graph of MIPGEM holds 7,351 peptides, 2,057 matching proteins, and 1,863 gene models (after the pruning steps). The peptide and protein nodes are connected by 9,722 edges. The proteins are connected to the gene models by 2,087 additional edges. The graph decomposes into 1,508 connected components. Among the 1,863 gene models, 1,690 are true positives according to our approximate ground truth. A prior probability of 0.85 (the same for all proteins) was estimated for our model.

This dataset is interesting, because it allows us to show the possibilities and efficiency of MIPGEM in a domain where neither ProteinProphet nor MSBayesPro can compete, because they are not designed to infer gene model probabilities. Nevertheless, there are some shortcomings to this validation to keep in mind:

- There is no certainty that the 4,580 gene models in the ground truth correspond to the absolute truth. This set is a combination of the results from several experiments. It could contain wrong identifications or not include all truly expressed gene models.
- Although the set of assumed true positives is large (4,580 gene models), we can only identify up to 1,877 of them with the given set of identified peptides (no matching peptides were found for the other gene models).

Computational Details. The code is written in R (13). The following R packages are used: *Rgraphviz* (14) to plot the bipartite and tripartite graphs and *RBGL* (15) to compute the connected components of undirected graphs.

1. Keller A, Nesvizhskii AI, Kolker E, Aebersold R (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by ms/ms and database search. *Anal Chem* 74:5383–5392.
2. Nesvizhskii AI, Keller A, Kolker E, Aebersold R (2003) A statistical model for identifying proteins by tandem mass spectrometry. *Anal Chem* 75:4646–4658.
3. Li YF, et al. (2009) A Bayesian approach to protein inference problem in shotgun proteomics. *J Comput Biol* 16(8):1183–1193.
4. Li Q, MacCoss M, Stephens M (2010) A nested mixture model for protein identification using mass spectrometry. *Ann Appl Statist* (preprint).
5. Shen C, et al. (2008) A hierarchical statistical model to assess the confidence of peptides and proteins inferred from tandem mass spectrometry. *Bioinformatics* 24(2):202–208.
6. Eng JK, McCormack AL, Yates JR, III (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J Am Soc Mass Spectr* 5:976–989.

7. Keller A, et al. (2002) Experimental protein mixture for validating tandem mass spectral analysis. *OMICS* 6:207–212.
8. Alves P, et al. (2007) Advancement in protein inference from shotgun proteomics using peptide detectability. *Pacific Symposium on Biocomputing* 409–420.
9. Tabb DL, McDonald WH, Yates JR, III (2002) DTASelect and contrast tools for assembling and comparing protein identifications from shotgun proteomics. *J Proteome Res* 1:21–26.
10. Tabb DL, Fernando CG, Chambers MC (2007) Myrimatch: Highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis. *J Proteome Res* 6:654–661.
11. Brunner E, et al. (2007) A high-quality catalog of the Drosophila melanogaster proteome. *Nat Biotechnol* 25:576–583.
12. Grobei MA, et al. (2009) Deterministic protein inference for shotgun proteomics data provides new insights into Arabidopsis pollen development and function. *Genome Res* 19:1786–1800.
13. R Development Core Team (2008) *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna).
14. Gentry J, et al. (2009) Rgraphviz: Provides plotting capabilities for R graph objects. R package ver. 1.20.0, http://www.bioconductor.org/packages/release/bioc/html/Rgraphviz.html.
15. Carey V, Long L, Gentleman R (2009) RBGL: An interface to the BOOST graph library. R package ver. 1.16.0, http://www.bioconductor.org/packages/release/bioc/html/Rgraphviz.html.

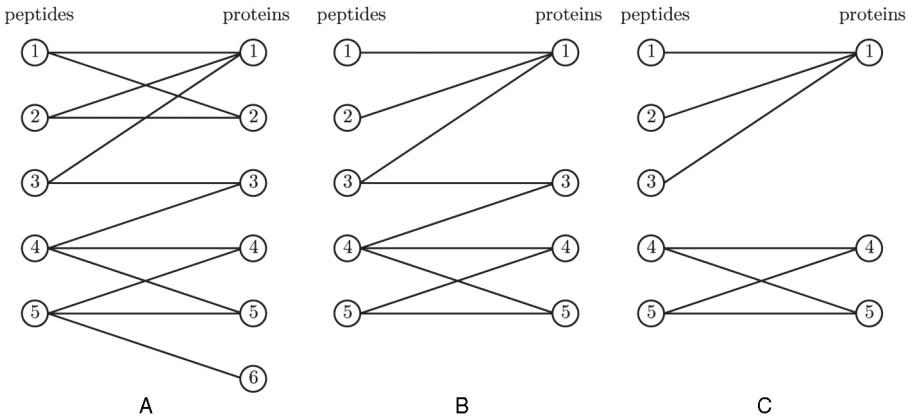

Fig. S1. The three steps to generate our bipartite graph holding the peptide and the protein sequences.

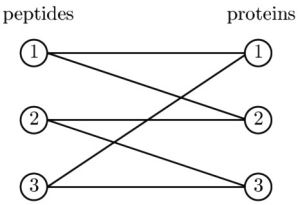

Fig. S2. Example of a connected component with only shared peptides where none of the proteins is removed by our pruning procedure.

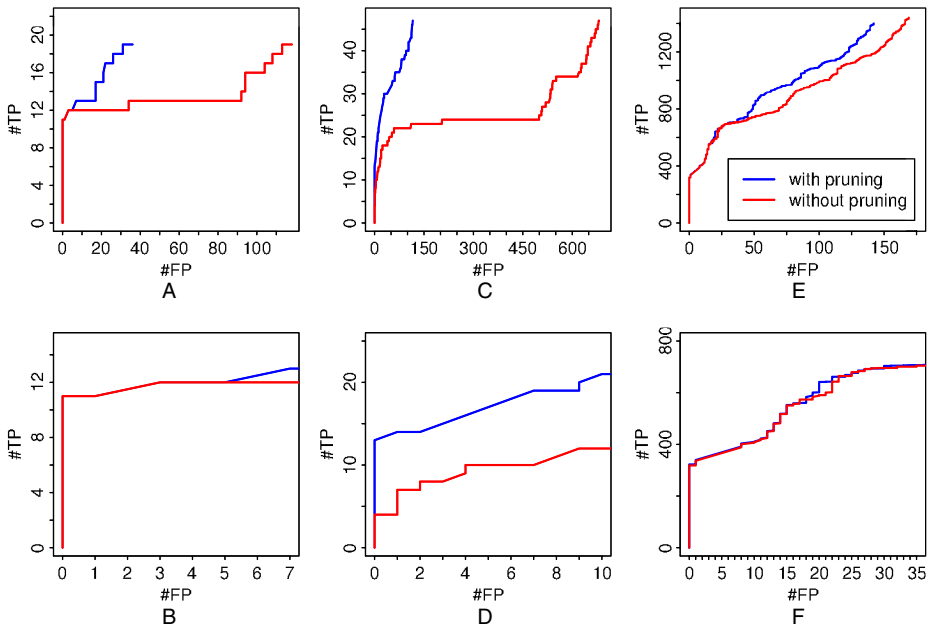

Fig. S3. Plots of the receiver operating characteristics (ROCs) to compare the performance of MIPGEM on the pruned and unpruned sets of proteins. Plots for the three datasets with known (or approximate) ground truth on protein level are provided. *A* and *B* illustrate the results for the mixture of 18 proteins, *C* and *D* the outcome for the Sigma49 dataset, and *E* and *F* correspond to the *S. cerevisiae* dataset. The curves in *A*, *C*, and *E* show the global view. In *B*, *D*, and *F*, we focus on the left region of the curves, where the number of false positive hits is low. The used protein priors were 0.35 (pruned and unpruned) for the mixture of 18 proteins, 0.3 (pruned) and 0.2 (unpruned) for Sigma49, and 0.5 (pruned and unpruned) for *S. cerevisiae*.

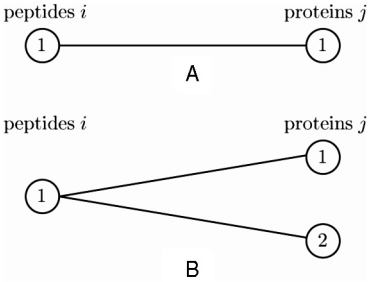

Fig. S4. Examples of connected components. (*A*) A peptide matching to a single protein. (*B*) The peptide is shared between two proteins.

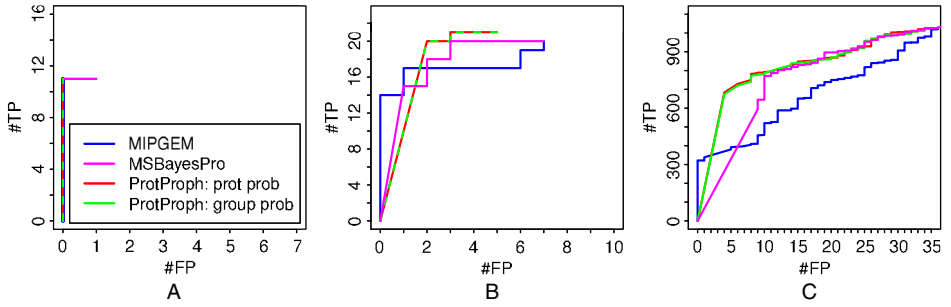

Fig. S5. Comparisons between MIPGEM and the two reference models when discarding single hits. The three plots show the number of true positives (#TP) versus number of false positives (#FP) for (*A*) the mixture of 18 purified proteins (protein prior was set to 0.35 for MIPGEM), (*B*) the Sigma49 dataset (protein prior was set to 0.3 for MIPGEM), and (*C*) the *S. cerevisiae* dataset (protein prior was set to 0.5 for MIPGEM). The identified single hits were discarded in all methods.

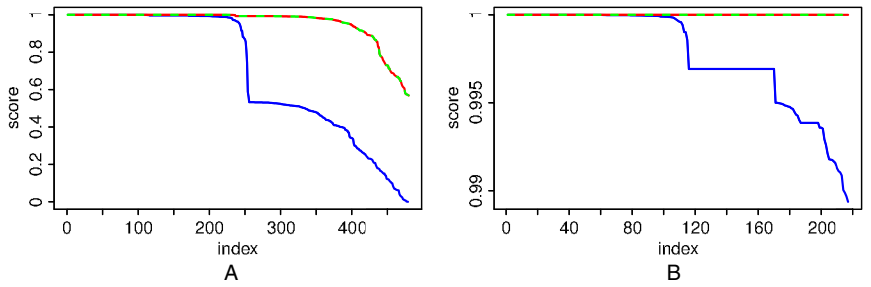

Fig. S6. Comparison of the score distribution in ProteinProphet and in MIPGEM. (*A*) Sorted protein scores for ProteinProphet and MIPGEM for all identified proteins in the *D. melanogaster* dataset indicating ProteinProphet's tendency of returning many proteins with a score of exactly one. (*B*) A zoom on the top-scoring 217 proteins of both methods. Unlike ProteinProphet, our model allows to rank these top protein identifications.

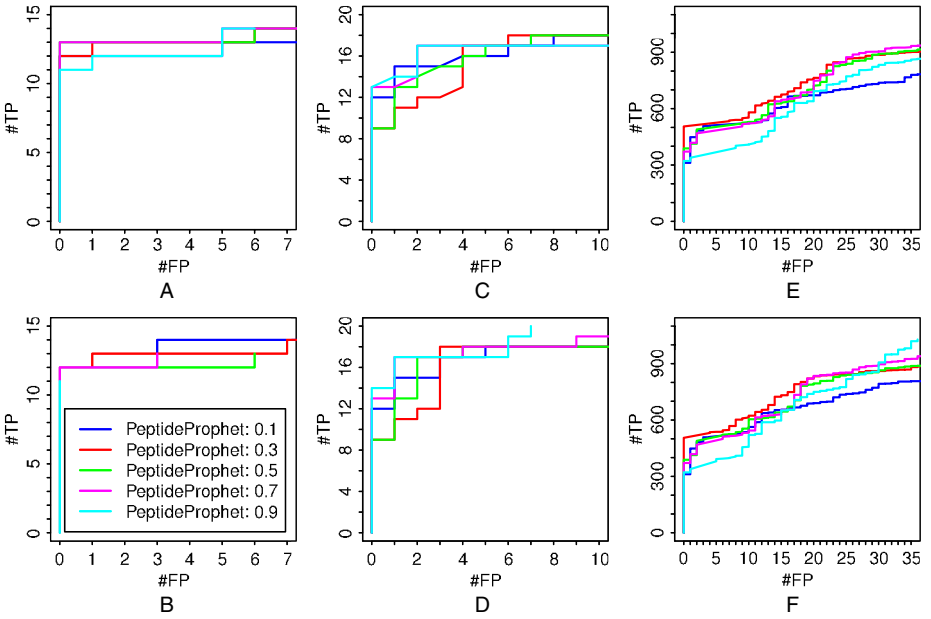

Fig. S7. Plots of the ROC curves of MIPGEM's results for varying cutoffs of the peptide scores. Results on the three datasets with known (or approximate) ground truth are displayed. *A* and *B* illustrate the results for the mixture of 18 proteins, *C* and *D* the outcome for the Sigma49 dataset, and *E* and *F* correspond to the *S. cerevisiae* dataset. The curves in *A*, *C*, and *E* include the single hits. In *B*, *D*, and *F*, the single hits were discarded. The used protein priors were, in increasing order of the peptide score cutoffs, 0.05, 0.1, 0.15, 0.2, 0.35 for the mixture of 18 proteins, 0.05, 0.05, 0.1, 0.2, 0.3 for Sigma49, and 0.2, 0.35, 0.45, 0.55, 0.5 for *S. cerevisiae*.

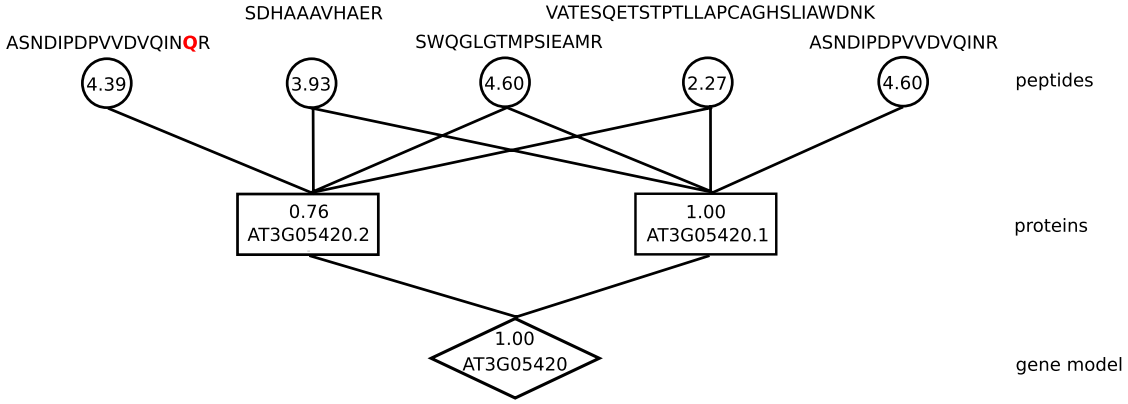

Fig. S8. Distinction of alternatively spliced protein isoforms with our tripartite graph model. As an example, we show one connected component from the tripartite graph of the *A. thaliana* dataset. Here, the experimental peptide evidence can unambiguously identify and distinguish two alternatively spliced protein isoforms (AT3G05420.1, 668 amino acids; AT3G05420.2, 669 amino acids) that are encoded by the same gene model (AT3G05420). The two protein sequences differ by only one amino acid (see the red amino acid in the left-most peptide). Both (almost identical) peptide sequences on the left (ASNDIPDPVVDVQINQR) and on the right (ASNDIPDPVVDVQINR) are specific to one of the protein isoforms. The other three peptides are shared between both proteins. ProteinProphet assigns these proteins to two different protein groups and cannot, in contrast to MIPGEM, provide a probability for the encoding gene model.

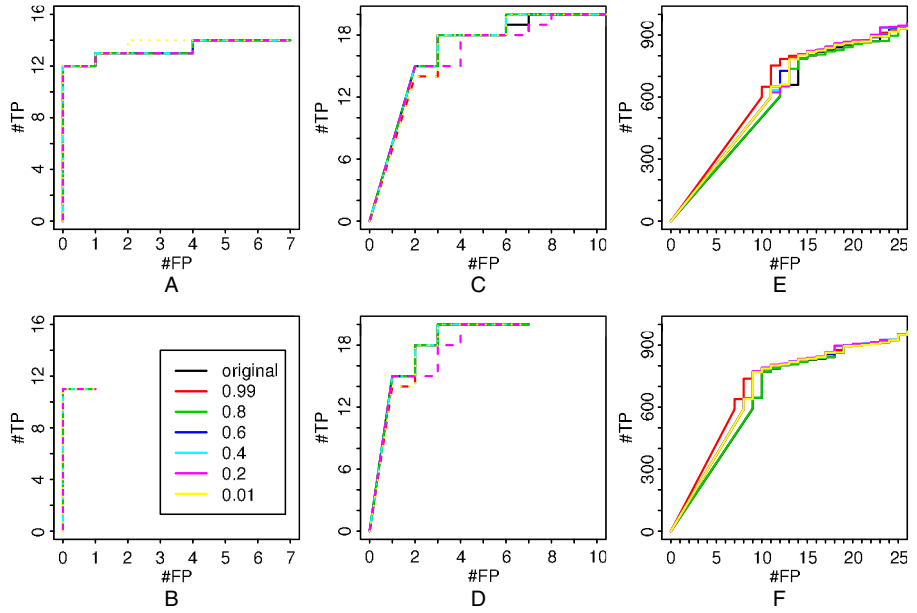

Fig. S9. Comparison of the effect on protein inference when varying the input peptide detectabilities for MSBayesPro. The black line corresponds to the run with predicted peptide detectabilities. The colored lines correspond to runs where all the peptide detectabilities were set to a common constant number (given by the legend). *A* and *B* illustrate the results for the mixture of 18 proteins, *C* and *D* the outcome for the Sigma49 dataset, and *E* and *F* correspond to the *S. cerevisiae* dataset. The curves in *A*, *C*, and *E* include the single hits. In *B*, *D*, and *F*, the single hits were discarded. The differences in performance are very small between the different runs. It does not seem to be worth predicting peptide detectabilities as input to MSBayesPro, because a similar performance can be reached by setting all these values to a common constant.

Table S1. Effects of the graph pruning on the protein inference

	mix. of 18 prot.	Sigma49	*D. melanogaster*	*S. cerevisiae*	*A. thaliana*
No. proteins before pruning	145	755	993	1,609	2,465
No. proteins after first pruning step	60	170	865	1,542	2,067
No. proteins after second pruning step	60	169	863	1,542	2,057

Table S2. Overlap of protein identifications without single hits

n	25	50	78	100	150	200	217
Reference (*i*)	25	45	72	94	116	154	163
Reference (*ii*)	25	50	78	100	123	169	185
Reference (*iii*)	25	45	72	94	143	181	190

Table S3. List of repositories for the five datasets used in the evaluation

Dataset	Source
Mixture of 18 purified proteins	http://www.systemsbiology.org/extra/protein_mixture.html
Sigma49	http://www.mc.vanderbilt.edu/root/vumc.php? site=msrc/bioinformatics&doc=21164
S. cerevisiae	http://www.marcottelab.org/MSdata/Data_02/
D. melanogaster	http://www.peptideatlas.org/repository/ We worked with Dm_Kc_Golgi_exp_045.
A. thaliana	http://www.ebi.ac.uk/pride/ Accessions: 8743, 8744, 8745, 8746, 8747, 8748, 8749, and 8750

Table S4. List of considered true positives in the mixture of 18 proteins

sp|P02666|CASB_BOVIN	sp|P00489|PHS2_RABIT	sp|P02603|MLE3_RABIT	sp|Q29443|TRFE_BOVIN
sp|P00921|CAH2_BOVIN	sp|P00722|BGAL_ECOLI	sp|P24732|MLRT_RABIT	sp|P46406|G3P_RABIT
sp|P00006|CYC_BOVIN	sp|ATBOG|actin	sp|P04461|MYH7_RABIT	sp|P35748|MYHB_RABIT
sp|P02754|LACB_BOVIN	sp|P00432|CATA_BOVIN	sp|Q99105|MYSU_RABIT	sp|Q28641|MYH4_RABIT
sp|P00711|LCA_BOVIN	sp|P02562|MYSS_RABIT	sp|P00634|PPB_ECOLI	SW:AMY_BACLI
sp|P02769|ALBU_BOVIN	sp|P02602|MLE1_RABIT	sp|P02188|MYG_HORSE	sp|P29952|MANA_YEAST
sp|P01012|OVAL_CHICK	sp|P04460|MYH6_RABIT	sp|Q04977|AMYM_BACLI	

Table S5. List of considered contaminants for the mixture of 18 proteins

SW:CAS1_BOVIN	SW:K22O_HUMAN	SW:PHS2_HUMAN	SW:K1CI_HUMAN	SW:K2C7_HUMAN
SW:CAS2_BOVIN	SW:K2C1_HUMAN	SW:PHS3_HUMAN	SW:K22E_HUMAN	SW:G3P2_HUMAN
SW:CASK_BOVIN	SW:K2C3_HUMAN	SW:ACTA_HUMAN	SW:CATA_HUMAN	

Table S6. List of considered true positives in the Sigma49 protein mixture

O00762	P01127	P02768	P08263	P15559	P62988	P00918	P02144	P06396	P10599
P00167	P01133	P02787	P08311	P16083	P63165	P01008	P02741	P06732	P10636
P00441	P01343	P02788	P08758	P41159	P63279	P01031	P02753	P07339	P12081
P00709	P01344	P04040	P09211	P51965	P68871	P01112	P99999	P61626	P62937
P00915	P01375	P05413	P10145	P55957	P69905	Q15843	Q06830	P61769	

Table S7. List of considered contaminants for the Sigma49 protein mixture

P02446	Q29463	P00711	Q5XQN5	P08727	P48666	O76013	P12763	Q14533	P04264
P02445	P19013	Q01546	P02448	P19012	P02538	O77727	P02666	O43790	P50446
P02444	P00760	P02663	Q29426	P13645	P04259	P00791	P35908	P30879	Q92764
P02439	P00761	P0C1U8	Q14525	P00792	P15241	P35900	P02769	P35527	P02534
P02440	P48667	Q15323	Q9NSB4	P25691	P00767	Q99456	P02770	P78386	Q28580
P02438	Q7M135	P04745	P02443	O76011	P00766	Q10735	P02441	Q9NSB2	O76014
P08131	Q07627	P02662	O76009	P25690	Q02958	P26371	P12035	Q14532	
P48668	P15636	P78385	P05783	P02539	P26372	P02668	P13647	O76015	