ADILSON MENDES RICARDO

MELHORIA DA SENSIBILIDADE EM DADOS DE PROTEÔMICA *SHOTGUN* USANDO REDES NEURAIS ARTIFICIAIS SENSÍVEIS AO CUSTO E O ALGORITMO *THRESHOLD SELECTOR*

Dissertação apresentada a Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

Viçosa
Minas Gerais-Brasil
2015

Ficha catalográfica preparada pela Biblioteca Central da Universidade Federal de Viçosa - Câmpus Viçosa

T
R488m
2015

Ricardo, Adilson Mendes, 1966-
Melhoria da sensibilidade em dados de proteômica Shotgun usando redes neurais artificiais sensíveis ao custo e o algoritmo Threshold Selector / Adilson Mendes Ricardo. – Viçosa, MG, 2015.
xv, 84f. : il. (algumas color.) ; 29 cm.

Inclui anexos.
Orientador: Fábio Ribeiro Cerqueira.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f. 58-62.

1. Bioinformática. 2. Redes neurais (Computação). 3. Mineração de dados (Computação). 4. Algoritmos. 5. Peptídeos - Identificação - Processamento de dados. 6. Proteômica. I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-graduação em Ciência da Computação. II. Título.

CDD 22. ed. 570.285

ADILSON MENDES RICARDO

MELHORIA DA SENSIBILIDADE EM DADOS DE PROTEÔMICA SHOTGUN USANDO REDES NEURAIS ARTIFICIAIS SENSÍVEIS AO CUSTO E O ALGORITMO THRESHOLD SELECTOR

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 07 de dezembro de 2015.

Sabrina de Azevedo Silveira

Humberto Josué de Oliveira Ramos

Fábio Ribeiro Cerqueira
Orientador

Para Eni, Lucas, Tiago e Letícia pela parceria e dedicação.

"Entretando, se descobrirmos de fato uma teoria completa, ela deverá, ao longo do tempo, ser compreendida, grosso modo, por todos e não apenas por alguns poucos cientistas. Então devemos todos, filósofos, cientistas, e mesmo leigos, ser capazes de fazer parte das discussões sobre a questão de porque nós e o universo existimos. Se encontrarmos a resposta para isto teremos o triunfo definitivo da razão humana: porque, então, teremos atingido o conhecimento da mente de Deus."

(Stephen W. Hawking)

AGRADECIMENTOS

Agradeço primeiramente a Deus pela benção da oportunidade e da coragem de iniciar, assim como pela saúde para conduzir todo este período de dedicação ao mestrado. Muitos foram os amigos que incentivaram-me neste desafio e agradeço a cada um. Aos professores orientadores Fábio e Alcione, sou grato pela paciência, dedicação e competência. Agradeço também ao coordenador e professor Jugurta, aos demais professores, funcionários e colegas do programa de mestrado de Ciência da Computação da UFV. Um agradecimento especial aos órgãos de financiamento CAPES, CNPq e FAPEMIG pelo apoio essencial durante este período. Por fim, à minha família, Eni, Lucas, Tiago e Letícia, agradeço pelo apoio, companheirismo, dedicação e determinação. À todos que construíram comigo esta realização, obrigado.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

ANN - *artifical neural network* - rede neural artificial

AUC - *area under curve* - área sob a curva

BD - banco de dados

CID - *collision-induced dissociation* - dissociação induzida por colisão

DM - *data mining* - mineração de dados

DNA - ácido desoxirribonucleico

EI - *electron impact* - impacto de elétrons

EM - *expectation Maximization* - algoritmo expectativa maximização

ETD - *electron transfer dissociation* - dissociação por transferência de elétron

eV - elétron-volt

FDR - *false discovery rate* - taxa de descobertas falsas

FN - *false negatives* - falsos negativos

FP - *false positives* - falsos positivos

FPR - *false positive rate* - taxa de falsos positivos

HCD - *higher-energy collision dissociation* - dissociação por colisão de alta energia

KDD - *knowledge discovery in databases* - descoberta do conhecimento em banco de dados

LC - *liquid chromatography* - cromatografia líquida

LC-MS/MS - *liquid chromatography - mass spectrometry/mass spectrometry* - cromatografia líquida acoplada a espectrometria de massa em sequência

m/z - razão massa/carga

MLE - *maximum likelihood estimation* - estimativa por máxima verossimilhança

MLP - *multilayer perceptron* - algoritmo perceptron de múltiplas camadas

mRNA - RNA mensageiro

MS - *mass spectrometry* - espectrometria de massa

MS/MS - *mass spectrometry/mass spectrometry* - espectrometria de massa em sequência

MUDE - *multivariate decoy database analysis*

MUMAL - *multivariate analysis in shotgun proteomics using machine learning techniques*

MUMAL2 - *Improving sensitivity in shotgun proteomics using cost sensitive artificial neural networks and a threshold selector algorithm*

PSM - *peptide-spectrum matches* - correspondência espectro-peptídeo

RNA - ácido ribonucléico

ROC - *receiver operating characteristic*

TDDB - *target-decoy database search* - pesquisa em banco de dados alvo-isca

TN - *true negatives* - verdadeiros negativos

TP - *true positives* - verdadeiros positivos

TPR - *true positive rate* - taxa de verdadeiros positivos

TSA - *threshold selector algorithm*

ULPAN - *analysis in complex proteomics data using unsupervised learning techniques preceding the ANN*

Resumo

RICARDO, Adilson Mendes, M.Sc., Universidade Federal de Viçosa, dezembro de 2015. **Melhoria da Sensibilidade em dados de proteômica *Shotgun* usando redes neurais artificiais sensíveis ao custo e o algoritmo *threshold selector***. Orientador: Fábio R. Cerqueira. Coorientador: Alcione P. Oliveira.

Antecedentes: Este trabalho apresenta uma estratégia de aprendizagem de máquina para aumentar sensibilidade na análise de dados de espectrometria de massa para identificação de peptídeos / proteínas. A espectrometria de massa em *tandem* é uma técnica de química analítica amplamente utilizada para identificar as proteínas em misturas complexas, dando origem a milhares de espectros em uma única corrida que são depois interpretados por software. A maioria destas abordagens computacionais usam bancos de dados de proteínas para realizar a interpretação dos espectros, ou seja, para cada um, obter a melhor correspondência entre o mesmo e a sequência de um peptídeo obtido computacionalmente, a partir das sequências de proteínas do banco de dados. As correspondências espectro-peptídeo (PSM - *peptide-spectrum matches*) também devem ser avaliadas por ferramentas computacionais já que a análise manual não é possível em função do volume. A estratégia do banco de dados *target-decoy* é largamente utilizada para avaliação de PSMs. No entanto, em geral, o método não considera a sensibilidade, apenas a estimativa de erro. Resultados: Em trabalho de pesquisa anterior, o método MUMAL aplica uma rede neural artificial para gerar um modelo para classificar PSMs usando a estratégia do banco de dados *target-decoy* para o aumento da sensibilidade. Entretanto, o presente trabalho de pesquisa mostra que a sensibilidade pode ser melhorada com a utilização de uma matriz de custo associada com o algoritmo de aprendizagem. Demonstra-se também que a utilização do algoritmo *threshold selector* para o ajuste de probabilidades conduz a valores mais coerentes de probabilidade atribuídos para os PSMs, o que afeta positivamente a etapa de inferência de proteínas. Portanto, a abordagem aqui proposta, denominada MUMAL2, fornece

duas contribuições para proteômica *shotgun*. Em primeiro lugar, o aumento no número de espectros corretamente interpretados no nível de peptídeo aumenta a chance de identificar mais proteínas. Em segundo lugar, os valores mais adequados de probabilidade dos PSMs produzidos pelo algoritmo *threshold selector* impactam de forma positiva a fase de inferência de proteínas, realizada por programas que levam em conta estas probabilidades, tais como o ProteinProphet. Os experimentos demonstraram que o MUMAL2 fornece um maior número de verdadeiros positivos em comparação com métodos convencionais para avaliação de PSMs. Esta nova abordagem atingiu cerca de 15% de melhoria na sensibilidade em comparação com o melhor método atual. Além disso, a área sob a curva ROC obtida foi de 0,93, o que demonstra que as probabilidades geradas pelo MUMAL2 são, de fato, apropriadas. Finalmente, diagramas de Venn comparando o MUMAL2 com o melhor método atual mostram que o número de peptídeos exclusivos encontrado pelo MUMAL2 foi quase quatro vezes superior, o que impacta diretamente a cobertura do proteoma. Conclusões: A inclusão de uma matriz de custos e do algoritmo *threshold selector* na tarefa de aprendizagem melhora, ainda mais, a análise pela estratégia banco de dados *target-decoy* para identificação dos peptídeos, e contribui de forma eficaz para a difícil tarefa de identificação no nível de proteínas, resultando em uma poderosa ferramenta computacional para a proteômica *shotgun*.

Abstract

RICARDO, Adilson Mendes, M.Sc., Universidade Federal de Viçosa, December of 2015. **Improving sensitivity in shotgun proteomics using cost sensitive artificial neural networks and a threshold selector algorithm**. Advisor: Fábio R. Cerqueira. Co-advisor: Alcione P. Oliveira.

Background: This work presents a machine learning strategy to increase sensitivity in mass spectrometry data analysis for peptide/protein identification. Tandem mass spectrometry is a widely used analytical chemistry technique used to identify proteins in complex mixtures, yielding thousands of spectra in a single run which are then interpreted by software. Most of these computer programs use a protein database to match peptide sequences to the observed spectra. The peptide-spectrum matches (PSMs) must also be assessed by computational tools since manual evaluation is not practicable. The target-decoy database strategy is largely used for PSM assessment. However, in general, the method does not account for sensitivity, only for error estimate. Results: In a previous study, we proposed the method MUMAL that applies an artificial neural network to effectively generate a model to classify PSMs using decoy hits with increased sensitivity. Nevertheless, the present approach shows that the sensitivity can be further improved with the use of a cost matrix associated with the learning algorithm. We also demonstrate that using a threshold selector algorithm for probability adjustment leads to more coherent probability values assigned to the PSMs. Our new approach, termed MUMAL2, provides a two-fold contribution to shotgun proteomics. First, the increase in the number of correctly interpreted spectra in the peptide level augments the chance of identifying more proteins. Second, the more appropriate PSM probability values that are produced by the threshold selector algorithm impact the protein inference stage performed by programs that take probabilities into account, such as ProteinProphet. Our experiments demonstrated that MUMAL2 provides a higher number of true positives compared with standard methods for PSM evaluation.

This new approach reached around 15% of improvement in sensitivity compared to the best current method. Furthermore, the area under the ROC curve obtained was 0.93, demonstrating that the probabilities generated by our model are in fact appropriate. Finally, Venn diagrams comparing MUMAL2 with the best current method show that the number of exclusive peptides found by our method was nearly 4-fold higher, which directly impacts the proteome coverage. Conclusions: The inclusion of a cost matrix and a probability threshold selector algorithm to the learning task further improves the target-decoy database analysis for identifying peptides, which optimally contributes to the challenging task of protein level identification, resulting in a powerful computational tool for shotgun proteomics.

Capítulo 1

Introdução

1.1 O problema e sua importância

O termo proteoma é usado para designar o conjunto de proteínas expressas por um genoma. Nos projetos de proteoma um dos objetivos é caracterizar o máximo de proteínas possível de uma amostra, permitindo que sejam catalogadas computacionalmente e estudadas, de modo a atribuí-las um papel nas atividades celulares, incluindo casos de ocorrência da doença grave devido a mau funcionamento destas [Kim et al., 2014] [Kumar et al., 2014]. A cromatografia líquida acoplada a espectrometria de massa em sequência (ou em *tandem*) (LC-MS/MS - *liquid chromatography - mass spectrometry/mass spectrometry*) é a abordagem mais usada para este fim [Marcotte, 2007] [Wilhelm et al., 2014] [Lleo et al., 2014].

A LC-MS/MS é uma técnica analítica poderosa que pode ser usada para identificar materiais desconhecidos e elucidar as propriedades químicas e estruturais das células. É um método que permite a determinação da massa molecular de compostos com alta acurácia. Com equipamentos cada vez mais especializados em proteínas, a LC-MS/MS tornou-se uma ferramenta poderosa para a identificação e o estudo dessas moléculas [Silverstein et al., 2014]. No LC-MS/MS as moléculas da amostra são ionizadas e os ions são separados na razão massa/carga (m/z). Basicamente o processo implica em: digestão das proteínas de uma mistura complexa em peptídeos, a separação destes peptídeos por cromatografia líquida (LC), a aquisição em sequência dos espectros de massa do peptídeo por fragmentação espectrométrica e a aplicação de softwares, tais como *Sequest* e *Mascot*, para interpretar cada espectro MS/MS, o que resulta na identificação de proteínas presentes na amostra [Silverstein et al., 2014].

Uma corrida de LC-MS/MS gera milhares de espectros, em que cada um pode representar um peptídeo. O próximo passo é atribuir uma sequência peptídica a cada

espectro com base no seu padrão de pico espectral. A necessidade é avaliar automaticamente a correspondência espectro-peptídeo (PSMs - *peptide-spectrum matches*) resultantes, dada a enorme quantidade normalmente produzida em uma única corrida e pelo fato de haver um número potencialmente alto de falsos positivos [Cerqueira et al., 2012] [Swan et al., 2013].

O objetivo é identificar o máximo de proteínas possível de uma amostra. É importante notar que a quantidade potencialmente alta de falsos positivos torna a estimativa da taxa de descobertas falsas (FDR - *false discovery rate*) das identificações uma importante questão em proteômica.

1.2 Estratégias e trabalhos relacionados

Existem basicamente duas técnicas de interpretar espectros MS/MS. Uma delas é a denominada abordagem de novo que analisa os padrões de pico sem usar qualquer informação externa [Ma et al., 2003]. A técnica mais comum, no entanto, utiliza conjuntos de dados de sequências de proteínas, que é o caso de programas computacionais tais como Sequest e Mascot [Eng et al., 1994] [Perkins et al., 1999]. Estes programas executam uma digestão *in silico* das proteínas presentes no banco de dados (DB - *database*) e geram espectros virtuais a partir dos peptídeos resultantes. Desta forma, para cada espectro observado, o programa encontra a sua melhor correspondência a um espectro virtual e a respectiva sequência peptídica é atribuída ao dado espectro MS. Os programas normalmente relatam as dez melhores pontuações. Vários escores são atribuídas a um PSM para medir a sua qualidade [Cerqueira et al., 2010] [Cerqueira et al., 2012] [Söderholm et al., 2014]. Esta estratégia pode ser utilizada para identificar e quantificar os peptídeos/proteínas [Silverstein et al., 2014].

No entanto, um problema importante neste processo é que uma única execução LC-MS/MS normalmente leva a milhares de espectros, em que menos de 20% são interpretados corretamente [Cerqueira et al., 2012]. Neste contexto, a FDR é importante para estimar as identificações [Ivanov et al., 2014].

Uma das estratégias usadas para estimar a FDR, usando a técnica do conjuntos de dados de sequência de proteínas, é realizar a busca por sequências de aminoácidos, para associar a espectros, em bancos de dados alvo-isca (TDDB - *target-decoy database search*) [Walzthoeni et al., 2012] [Cerqueira et al., 2012] [He et al., 2015]. Nesta abordagem, as sequências de proteínas falsas (*decoys*) são geradas para serem usadas juntamente com sequências de proteínas-alvo (*target*) para a pesquisa, que pode ser realizada usando um DB *target-decoy* ou em duas rodadas, ou seja, uma busca para

cada DB (*decoy* e *target*). Os métodos comuns para a geração de sequências de *decoys* são reverter ou embaralhar as sequências alvo, mantendo a distribuição de aminoácidos. As sequências *decoys* devem ser produzidas de uma forma que seja razoável supor que a sequência de um PSM incorreto tenha igual probabilidade de vir de ambos os conjuntos de proteínas, alvo ou *decoy*. A estratégia TDDB baseia-se na premissa de que os PSMs *decoys* são bons modelos para os PSMs alvo incorretos. Desta forma, a estimativa do número de PSMs alvo incorretos será o número de PSMs *decoy* [Cerqueira et al., 2012] [Granholm et al., 2012].

A estratégia TDDB tem sido utilizado com sucesso para estimar FDR, porém em geral, não é aplicada adequadamente para otimizar a sensibilidade, isto é, combinações mais sofisticadas de escores não são totalmente exploradas visando aumentar a sensibilidade de PSMs [Li & Radivojac, 2012]. Além disso, as pontuações importantes são deixados de fora do processo de estimativa FDR [Ivanov et al., 2014].

Outra estratégia utilizada para avaliar PSMs é implementada pelo *PeptideProphet* que considera uma mistura de distribuições estatísticas dos escores dos PSMs para predizer os corretos e os incorretos. O método aprende a distinguir atribuições de peptídeos corretas de incorretas e calcula a probabilidade de cada atribuição estar correta. Para o caso do *Sequest*, por exemplo, os parâmetros de distruibuições gausiana e gama dos escores incorretos e corretos respectivamente são estimados pelo algoritmo EM (*Expectation Maximization*) [Keller et al., 2002] [Nesvizhskii et al., 2003] [Cerqueira et al., 2009]. Por outro lado, em certos conjuntos de dados, principalmente no caso de fosfopeptídeos, os escores podem apresentar distribuições completamente diferentes, pois o processo de fragmentação de peptídeos tem uma tendência para os grupos fosfato, levando à supressão de fragmentos de íons importantes [Imanishi et al., 2007] [Jiang et al., 2008] [Cerqueira et al., 2009].

A estratégia TDDB, por sua vez, funciona sem qualquer suposição preliminar sobre a distribuição dos dados. Duas pesquisas recentes que exploram a estratégia TDDB com o sofware *Sequest* foram base para este trabalho de pesquisa: o *multivariate decoy database analysis* (MUDE) [Cerqueira et al., 2010] e o *multivariate analysis in shotgun proteomics using machine learning techniques* (MUMAL) [Cerqueira et al., 2012].

O MUDE considera que outros parâmetros de qualidade fornecidos pelo *Sequest*, que normalmente não são utilizados quando se emprega a estratégia TDDB, passem a ser considerados visando o aumento da sensibilidade. O problema de encontrar os valores de corte (*threshold*) para os parâmetros do *Sequest* que conduzam a uma FDR desejada é tratado como um problema de otimização, em contraste com os procedimentos empregados anteriormente [Cerqueira et al., 2010]. A abordagem MUDE fornece

limites de decisão lineares para separar falsos de verdadeiros positivos. Além disso, a heurística utilizada para resolver o problema de otimização tem que ser executada várias vezes, sendo o resultado final uma junção de vários resultados obtidos [Cerqueira et al., 2010].

O trabalho de pesquisa MUMAL foi desenvolvido visando as melhorias propostas pelos autores do MUDE [Cerqueira et al., 2012]. O método MUMAL para avaliação dos PSMs baseia-se em técnicas de aprendizagem de máquina. Este método pode estabelecer limites de decisão não-lineares, levando a uma maior chance de recuperar mais verdadeiros positivos. Este é um aperfeiçoamento do método MUDE, onde o procedimento de otimização é substituído pela aplicação de rede neural artificial (ANN - *artifical neural network*) para encontrar melhores limites de decisão e a curva ROC (*receiver operating characteristic*) resultante é analisada para verificar linhas de decisão alternativas (probabilidades de corte variadas), com o intuito de explorar diversos valores de FDR [Cerqueira et al., 2012].

No MUMAL todos os PSMs alvo são considerados classe 1, enquanto que os PSMs *decoys* são considerados classe 0. A maioria dos PSMs reais representa uma identificação incorreta e têm, portanto, as mesmas características ou pontuações dos PSMs *decoy*, que são incorretos. Ou seja, se por um lado, todos os PSMs *decoys* são obviamente errados, por outro lado, apenas uma pequena parte dos PSMs alvos estão corretos. Isto torna o modelo de classificação uma tarefa desafiadora e, por esta razão, usando métodos clássicos de avaliação, o modelo resultante é considerado insatisfatório. No entanto, a análise ROC é uma ferramenta ideal para encontrar probabilidades discriminantes adequadas que proporcionem a FDR desejada com boa sensibilidade [Lasko et al., 2005]. A curva ROC é uma representação gráfica de uma curva onde cada ponto constituinte representa a taxa de verdadeiros positivos (eixo y) pela taxa de falsos positivos (eixo x) para um determinado valor de probabilidade discriminante [Cerqueira et al., 2012] [Cheung, 2014]. Com isto, as várias FDRs para cada valor discriminante são reportadas, tendo o usuário a possibilidade de escolher o valor que mais lhe convém [Cerqueira et al., 2012]. No MUMAL os autores abrem margem para novos projetos de pesquisa visando o aumento da sensibilidade, ou seja, maiores identificações de verdadeiros positivos.

1.3 Hipótese

1) Associar uma matriz de custo ao modelo de rede neural artificial aumenta a

sensibilidade na avaliação de PSMs.

2) O uso do algoritmo *Threshold Selector* produz probabilidades mais coerentes para cada PSM.

1.4 Objetivos

O objetivo deste trabalho de pesquisa é proporcionar uma abordagem que aumente as identificações de verdadeiros positivos (a sensibilidade) na avaliação automática de PSMs, já que uma enorme quantidade normalmente é produzida em uma única corrida LC-MS/MS e há um número potencialmente alto de falsos positivos. Para isto, os objetivos específicos são: investigar o comportamento de algoritmos de aprendizagem não supervisionada para o agrupamento dos dados de proteínas obtidos no processo LC-MS/MS; investigar algoritmos complementares à estrutura da ANN já adotada na abordagem MUMAL; desenvolver ferramentas computacionais que permitam a aplicação dos algoritmos investigados aos dados em questão.

1.5 Abordagens desenvolvidas nesta pesquisa

Duas abordagens são desenvolvidas neste trabalho usando a técnica de aprendizagem de máquina supervisionada ANN, utilizada no MUMAL, adicionando-se técnicas de agrupamento, matriz de custos e o algoritmo TSA (*threshold selector*). A abordagem que apresentou melhor resultado é aqui denominada Melhoria da sensibilidade em dados de proteômica *shotgun* usando redes neurais artificiais sensíveis ao custo e o algoritmo *threshold selector* (MUMAL2 - *Improving sensitivity in shotgun proteomics using cost sensitive artificial neural networks and a threshold selector algorithm*). Baseia-se no princípio que um aspecto dos modelos que utilizam técnicas de classificação é o custo associado com as classificações erradas produzidas pelo modelo. O uso da técnica *cost-sensitive classification*, que consiste na inserção de uma matriz de custo padrão para cada tipo de erro falso positivo e falso negativo associado [Witten & Frank, 2005], aumentou o desempenho do classificador e, consequentemente, a quantidade de verdadeiros positivos. O TSA por sua vez faz com que os valores de probabilidade sejam mais confiáveis [Kapp et al., 2005], favorecendo futuros trabalhos que envolvam a inferência de proteínas.

Os experimentos deste trabalho de pesquisa foram realizados em 11 conjuntos de dados. O MUMAL2 demonstra um aumento da sensibilidade de forma mais consistente para todos os conjuntos de dados, em uma comparação com métodos de re-

ferência para a avaliação PSMs. O MUMAL2 alcançou um aumento médio de 15% na sensibilidade comparado ao melhor método atual, para FDRs variando de 0% a 5%. Usando diagramas de Venn com peptídeos identificados, para uma FDR de 1%, o MUMAL2 encontrou quase 4 vezes mais peptídeos exclusivos, o que impacta diretamente a cobertura do proteoma. Além disso, em um experimento adicional com o gráfico ROC, usando um conjunto de dados com proteínas conhecidas, a área sob a curva (AUC - *area under curve*), calculado após o ajuste de probabilidades por TSA, foi de 0,93, o que demonstra que as probabilidades geradas pelo MUMAL2 são, de fato, apropriadas. Vale notar a demonstração do poder preditivo do MUMAL2 para fosfopeptídeos. Nestes casos, a distribuição da pontuação pode ser muito diferente de PSMs não-fosfopeptídeos, o que complica a análise por ferramentas computacionais tradicionais, tais como PeptideProphet.

1.6 Organização do trabalho

O capítulo 1 deste trabalho mostra os fundamentos e abordagens que motivaram o desenvolvimento desta pesquisa, bem como apresenta previamente as abordagens usadas para o cumprimento do objetivo.

O capítulo 2 descreve a pesquisa bibliográfica para o desenvolvimento do tema, apresentando os fundamentos teóricos envolvidos no estudo de bioinformática e proteômica, e os princípios das abordagens computacionais usadas para determinação e avaliação de dados de proteômica *shotgun*.

No capitulo 3 há a descrição do desenvolvimento da pesquisa, envolvendo os conjuntos de dados utilizados, os critérios técnicos considerados em cada uma das ferramentas computacionais analisadas e as abordagens desenvolvidas a partir destas ferramentas.

O capítulo 4 apresenta uma descrição dos resultados comparativos entre as abordagens propostas neste trabalho e os trabalhos anteriores relacionados, discutindo a contribuição de cada abordagem para o aumento da sensibilidade na avaliação dos PSMs.

Finalmente, o capítulo 5 mostra as conclusões e os avanços para novos trabalhos de pesquisa sobre o tema.

Capítulo 2

Referencial teórico

2.1 Bioinformática

A bioinformática surgiu da necessidade por recursos computacionais cada vez mais eficientes para, a partir das pesquisas desenvolvidas em biologia, fornecer o suporte e a interpretação dos resultados de processamento do enorme volume de informações geradas. O objetivo é, a princípio, a criação e manutenção dos bancos de dados para armazenar informações biológicas, tais como sequências de DNA e proteínas e, em seguida, a análise, a interpretação e a visualização destes dados [Russell, 2010]. As pesquisas desenvolvidas na década de 50 desvendaram a estrutura química do DNA, Ácido Desoxirribonucleico, um composto orgânico que contém as instruções genéticas e cuja função principal é armazenar informações necessárias para a construção das proteínas e RNAs - ácido ribonucléico. Posteriormente, a biologia molecular passou a constituir um dos principais objetos de estudos da área, dedicada aos métodos de sequenciamento, principalmente do DNA, que permitiram a investigação de suas sequências constituintes, gerando uma enorme quantidade de informações. A evolução dos métodos de sequenciamento, ocorrida nos anos 1990 e 2000, proporcionou uma explosão da quantidade de dados a serem armazenados, processados e analisados [Prosdocimi et al., 2002].

Os primeiros trabalhos e pesquisas na bioinformática eram desenvolvidos por profissionais de diferentes áreas como a biologia, a estatística e a computação. A demanda natural passou a ser por um perfil profissional com habilidade de desenvolver um elo entre estas áreas, com conhecimento suficiente para saber quais os problemas reais da biologia e quais as soluções viáveis de abordagem computacional para solucionar os problemas em questão. Conforme ilustra a Figura 2.1, essa nova ciência envolveria a

união de diversas linhas de conhecimento como a engenharia de software, a matemática, a estatística, a ciência da computação, a química, a bioquímica e a biologia molecular. A bioinformática é, portanto, a área da ciência onde a biologia e tecnologia da informação se unem com objetivos de tornar possível a descoberta de novos conhecimentos em biologia por meio de desenvolvimento de ferramentas computacionais, facilitando novas descobertas por meio da elaboração de algoritmos, da realização de análise matemática e estatística para tarefas como a identificação de um gene em uma sequência de DNA, a predição da estrutura e a função de um proteína, a inferência das proteínas presentes em uma análise de proteômica, dentre outros [Prosdocimi et al., 2002].

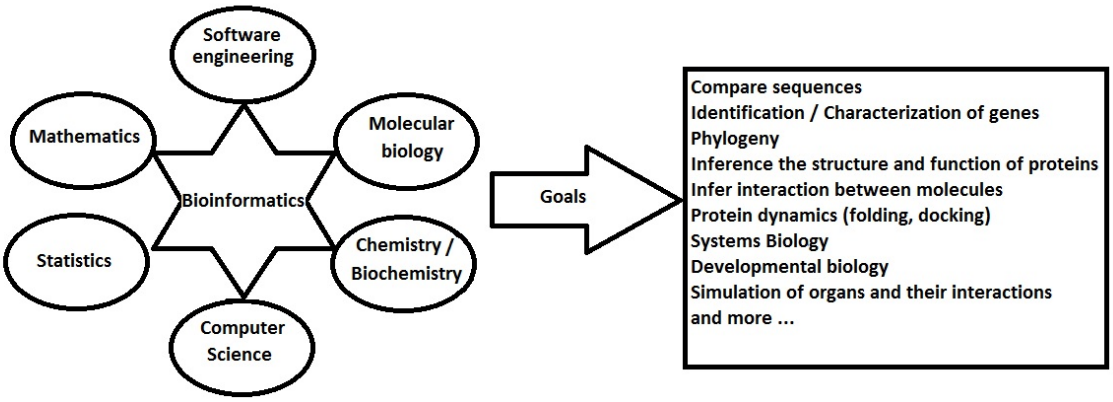


Figura 2.1. Linhas de conhecimento e objetivos da Bioinformática.

A bioinformática compreende análise, interpretação e visualização de vários tipos de dados, incluindo sequências, estrutura de proteínas, interações moleculares, etc. A maior parte dos esforços refere-se à biologia molecular, relacionando conhecimentos da bioquímica e da genética, investigando os mecanismos de replicação, transcrição e tradução do material genético. Grande parte dos trabalhos envolvem dados de projetos genoma. Genoma é toda a informação codificada no DNA de um organismo, incluindo tanto os genes como as regiões inter-gênicas. Os genes são segmentos de DNA que são responsáveis por carregar a informação genética. O DNA é um composto orgânico cujas moléculas possuem as instruções genéticas. Sua função principal é armazenar as informações necessárias para a construção das proteínas e RNAs. O RNA é o responsável pela síntese de proteínas da célula. Em projetos genoma de procariotos, onde as células não possuem núcleo e o DNA fica disperso pela membrana, é realizada a quebra do DNA inteiro do organismo desejado em fragmentos pequenos, através da técnica de sequenciamento (*shotgun*). Já em projetos genoma com organismos eucariotos, que

possuem uma enorme quantidade de DNA, normalmente usa-se a técnica conhecida como sequenciamento hierárquico ou *shotgun* hierárquico [Prosdocimi et al., 2002].

Um fluxo de informações celular importante nas pesquisas de bioinformática recebe o nome de dogma central da biologia e está representado na Figura 2.2. A transcrição gênica é o processo onde um gene faz a codificação para a produção de uma molécula específica de RNA. A tradução gênica é o processo de produção de uma proteína: um molde de um RNA mensageiro (mRNA) produz uma sequência de aminoácidos que ligam-se através das ligações peptídicas, formando os polipeptídeos. As proteínas participam da maioria dos processos bioquímicos dentro e fora das células, sendo substâncias essenciais da estrutura das células vivas. Portanto, o DNA é capaz de originar uma fita de RNA, que, por sua vez, possui o código para a síntese de aminoácidos para formar a cadeia polipeptídica [Snustad et al., 2008].

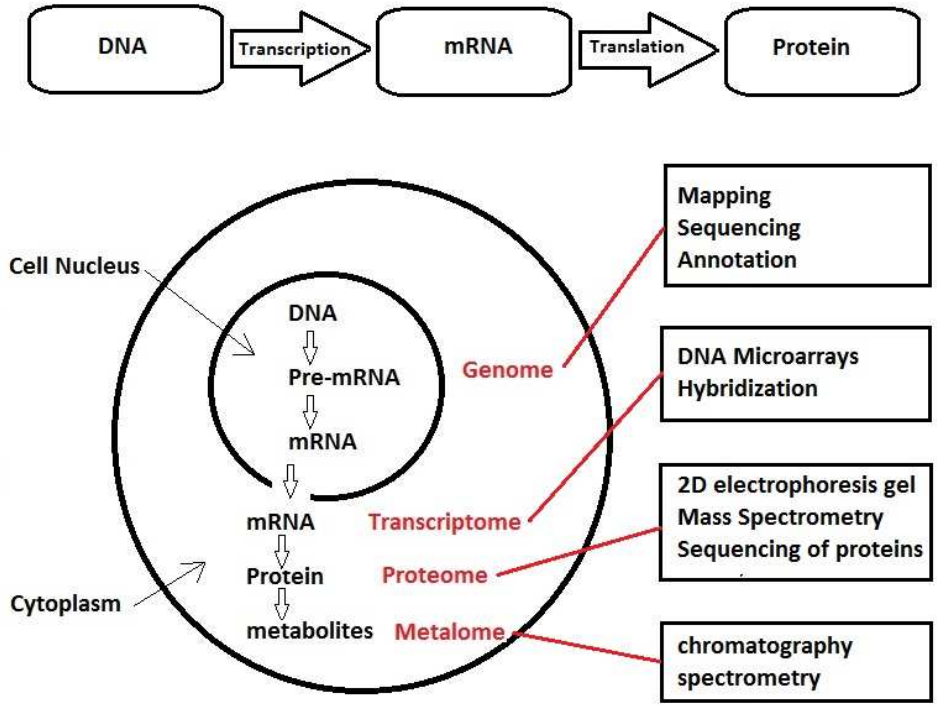


Figura 2.2. Visão geral do dogma central da biologia, do fluxo de informações celular e as abordagens usadas em cada sequenciamento.

Para um organismo eucarioto outra abordagem é, ao invés de ser realizado o sequenciamento genômico, realiza-se o sequenciamento só das regiões gênicas, utilizando informações oriundas do mRNA. Um problema com essa abordagem, de avaliação da expressão gênica a partir da análise dos mRNAs transcritos, é que nem sempre a quantidade de um mRNA consegue determinar a quantidade da proteína correspondente expressa na célula. Assim, não se pode relacionar

diretamente essa proteína a uma função nas células. Por isto, uma outra abordagem usada para avaliar a expressão gênica é a proteômica: análise das proteínas expressas em uma célula [Prosdocimi et al., 2002], que é a partida para este trabalho de pesquisa.

2.2 Proteômica

A proteômica é o conjunto de métodos analíticos empregados para caracterizar, em termos qualitativos e quantitativos, um proteoma. É de uma área de estudos interdisciplinar da ciência que agrega conhecimentos principalmente em química, biologia e tecnologia da informação. Proteoma é o conjunto de proteínas expressas por um genoma. Já o termo proteína, vem do grego *proteios*, significa a mais importante, podendo agir como enzimas, anticorpos, hormônios, receptores celulares, componentes estruturais, entre outras funções. Nos animais perfazem cerca de 90% do sangue seco, 80% do peso dos músculos desidratados e de 70% da pele [Leninger, 2002]. Além de formarem várias estruturas da célula, as proteínas controlam a entrada e a saída de substâncias pela membrana plasmática. Há muitas proteínas diferentes e cada uma especializada em uma função biológica específica: estrutural, enzimática, contrátil, hormonal, de defesa, de transporte, de proteção contra agentes externos, dentre outras [Lodish et al., 2014].

As proteínas são formadas por aminoácidos ligados entre si por ligações peptídicas, pertencendo à classe dos peptídeos. Os vegetais produzem todos os aminoácidos de que necessitam a partir de cadeias de carbono e nitrato retirado do ambiente. No ser humano os chamados aminoácidos essenciais não podem ser formados para suprir suas necessidades e devem ser, obrigatoriamente, consumidos na alimentação. Conforme ilustra a Figura 2.3, o aminoácido possui um átomo de carbono ao qual estão ligados uma carboxila *(-COOH)*, uma amina *(R-NH2)* e um hidrogênio. A quarta ligação é a porção variável e pode ser ocupada por um hidrogênio, ou por um metil ou por outro radical e está representada por R na Figura 2.3. Por analogia, os aminoácidos representam o alfabeto da estrutura proteica e determinam muitas das importantes propriedades da proteína. O composto orgânico formado é um peptídio. Se forem duas moléculas de aminoácido é um dipeptídio, se forem três é tripeptídio, e assim por diante, até a formação de filamentos longos, os polipeptídios. A ligação peptídica é a única ligação entre os aminoácidos na cadeia estrutural linear das proteínas. Com apenas vinte tipos de aminoácidos formam-se uma quantidade praticamente infinita de

proteínas diferentes e cada ser vivo tem a sua coleção proteica característica [Leninger, 2002].

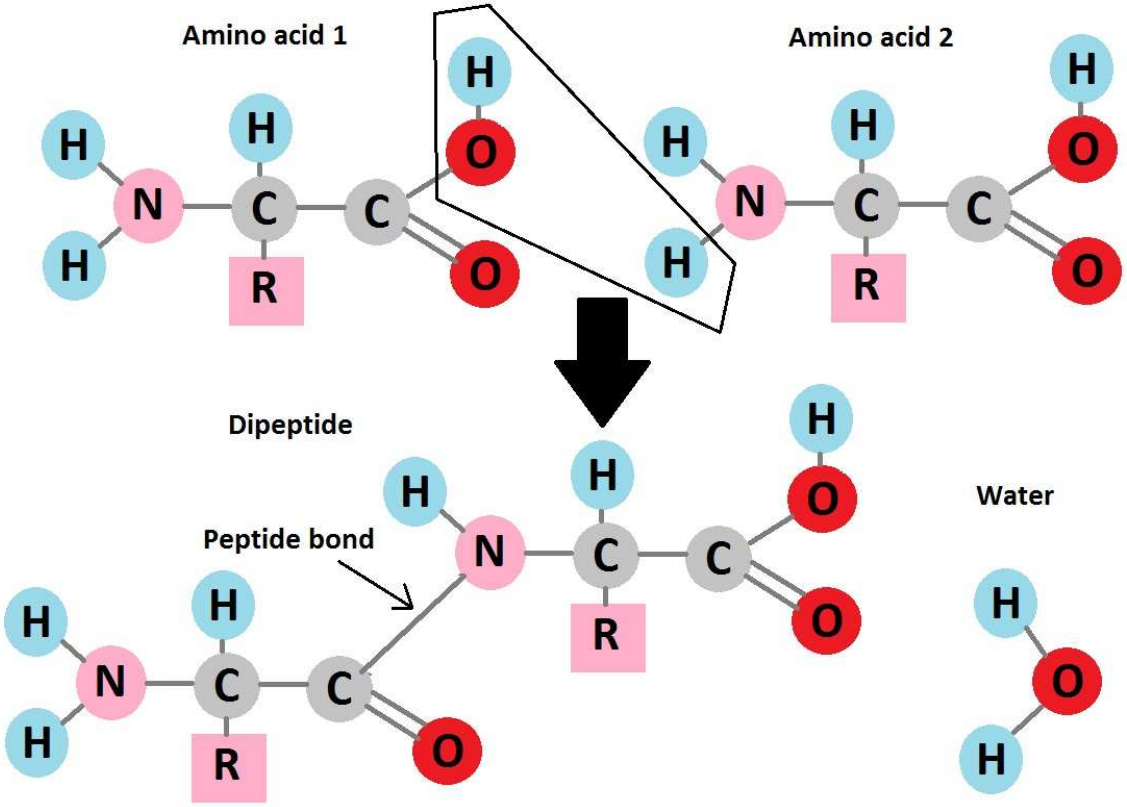


Figura 2.3. Formação de uma proteína - ligação peptídica de dois aminoácidos, onde *H, N, C* e *R* representam respectivamente as moléculas de Hidrogênio, Nitrogênio, Carbono e uma porção variável.

A estrutura primária de uma proteína é dada pela sequência de aminoácidos e ligações peptídicas da molécula, sendo o nível estrutural mais simples e importante, derivando dele todo o arranjo espacial da molécula. Já a estrutura secundária é formada pelo arranjo espacial, na sequência primária da proteína, de aminoácidos próximos entre si. A estrutura terciária, por sua vez, é a forma tridimensional como a proteína se entrelaça na sequência polipeptídica por meio do arranjo espacial de aminoácidos distantes entre si. Por fim a estrutura quaternária, complexa estrutural e funcionalmente, é formada pela distribuição espacial de várias cadeias polipeptídicas no espaço [Russell, 2010]. Estas estruturas são ilustradas na Figura 2.4.

Nos projetos de proteoma, um dos objetivos é separar e visualizar o máximo de proteínas possível em uma amostra para, posteriormente, permitir a catalogação computacionalmente e os estudos por técnicas analíticas. Uma abordagem moderna

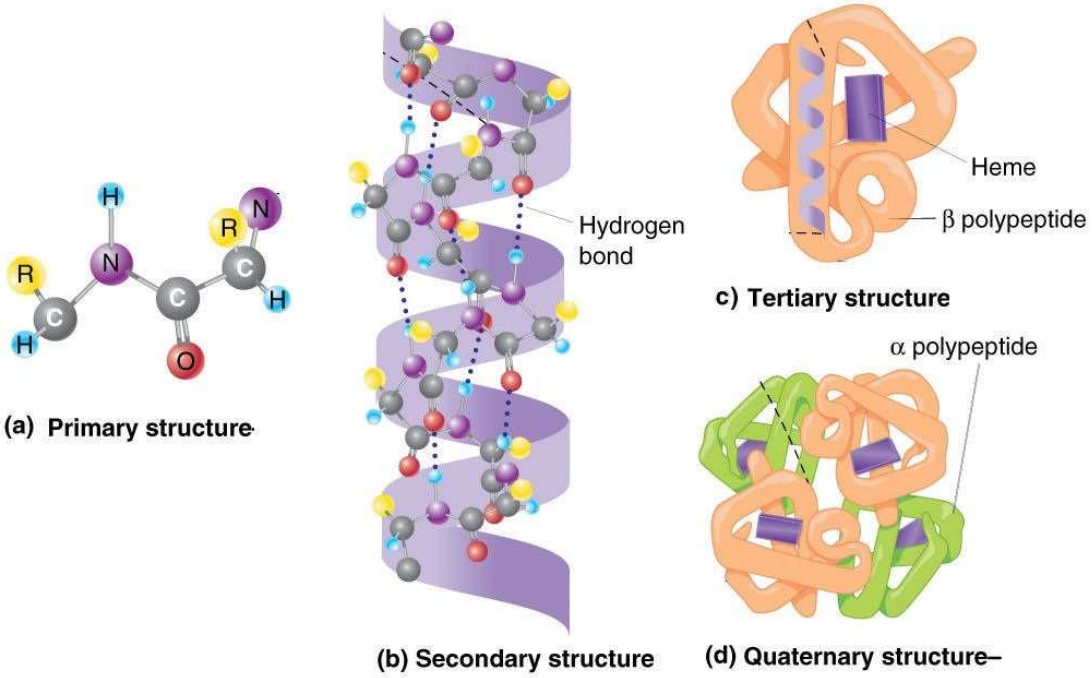


Figura 2.4. Estruturas primária, secundária, terciária e quartenária da proteína. Fonte: [Russell, 2010].

e sensível faz uso da espectrometria de massa por meio da abordagem LC-MS/MS, uma técnica analítica poderosa que pode ser usada para identificar materiais desconhecidos e elucidar as propriedades químicas e estruturais das células [Cantú et al., 2008]. Os fundamentos e conceitos associados à espectrometria de massa incluindo a espectrometria de massa em sequência e da abordagem LC-MS/MS são descritos na Seção 2.3.

2.3 Espectrometria de massa

A espectrometria de massa (MS - *mass spectrometry*) é uma técnica analítica que pode ser usada para a determinação da composição elementar ou da estrutura de uma molécula [Marcotte, 2007]. Os espectrômetros de massa, equipamentos usadas na MS, são úteis na análise tanto de compostos cujo espectro de massa é conhecido como em compostos com a estrutura completamente desconhecida. Para os compostos conhecidos uma busca computadorizada compara o espectro de massas gerado com uma biblioteca de espectros de massa. A coincidência entre os espectros indica a identificação. No caso de compostos desconhecidos a sequência de fragmentações e evidências de outros tipos de espectrometria podem levar a identificação de novos compostos [Silverstein

et al., 2014].

Um diagrama de um espectrômetro de massa é apresentado na Figura 2.5. Algum tipo de cromatografia usualmente precede a introdução da amostra no espectrômetro de massa. A cromatografia líquida (LC - *liquid chromatography*) consiste na separação em fase líquida de misturas complexas, entretanto dificilmente consegue fornecer a identificação positiva de componentes individuais e, normalmente, é associada a uma técnica de detecção como a MS, tornando-se uma poderosa ferramenta analítica. Todos os espectrômetros de massa têm métodos de ionização da amostra e separação dos íons na base da relação massa/carga m/z. Após a separação, os íons são detectados e quantificados. Toda a operação do instrumento é controlada por um computador, que também recolhe e armazena os dados obtidos, além de fornecer os resultados (espectros) na forma de gráficos ou tabelas [Silverstein et al., 2014].

O método de impacto de elétrons (EI - *electron impact*) é a técnica mais usada na geração de íons para a MS, onde as moléculas são bombardeadas com elétrons de alta energia, geralmente 70eV (elétron-volt), ainda na fase gasosa. Este bombardeio remove um elétron da molécula da amostra para produzir o íon molecular. Como o potencial de ionização dos compostos orgânicos é normalmente menor que 15eV, os elétrons que estão bombardeando as moléculas-alvo acrescentam energia da ordem de 50eV ao íon molecular. A quebra das ligações é muito previsível e reprodutível, tornando-a característica do composto e permitindo o aproveitamento da capacidade de elucidação de estruturas que há na MS. Um porém é que, com frequência, a energia adicionada ao íon molecular é tão grande que leva a um espectro de massa em que não é possível reconhecer o fragmento do íon molecular. As principais bibliotecas e bancos de dados de espectros de massas são formados por espectros de massas EI. Alguns dos bancos de dados têm mais de 390.000 espectros de massas EI em que a procura por meio de programas de computadores é muito rápida [Silverstein et al., 2014].

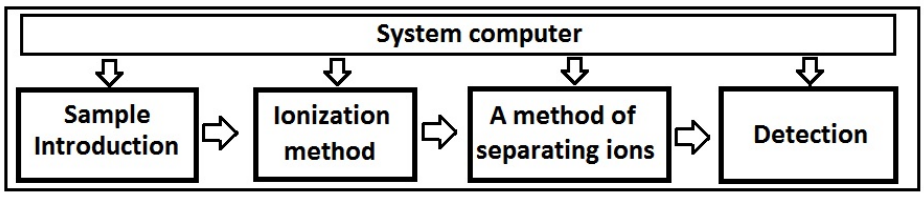


Figura 2.5. Visão esquemática de um espectrômetro de massas.

A espectrometria de massas em sequência ou em *tandem* (MS/MS - *mass spectrometry/mass spectrometry*) seleciona um íon principal da fragmentação inicial e fragmenta-o novamente para gerar íons filhos. Em misturas complexas, estes íons filhos

são a evidência da presença de um composto conhecido e, para o caso de compostos desconhecidos, esses permitem a obtenção de informações estruturais. Um modo de obtenção do MS/MS é ligar em série dois ou mais analisadores de massas para produzir um instrumento capaz de selecionar um único íon e analisar a sua fragmentação [Silverstein et al., 2014]. A Figura 2.6 exibe a visão esquemática do MS/MS com duas sequências de MS. Teoricamente são possíveis até 9 sequências de MS, mas na prática raramente ultrapassa 2 ou 3 [Silverstein et al., 2014].

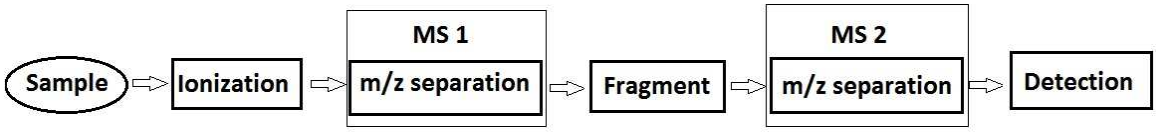


Figura 2.6. Visão esquemática da espectrometria de massas em sequência (MS/MS) com duas sequências de MS.

2.3.1 Abordagem LC-MS/MS

A abordagem de espectrometria de massa em sequência precedida pela cromatografia líquida (LC-MS/MS - *liquid chromatography - mass spectrometry/mass spectrometry*) prevê que a amostra seja fracionada pela cromatografia líquida e aplicada a dois estágios de espectrometria de massa. As moléculas são ionizadas e suas massas são detectadas, sendo reportadas na forma de espectro. Cada pico de um espectro representa a relação m/z de um íon. No primeiro estágio de MS os íons são os peptídeos. Tem-se então um único espectro, onde cada pico representa um peptídeo da amostra. No segundo estágio, peptídeos selecionados do primeiro são fragmentados por métodos como a dissociação induzida por colisão (CID - *collision-induced dissociation*), a dissociação por colisão de alta energia (HCD - *higher-energy collision dissociation*), a dissociação por transferência de elétron (ETD - *electron transfer dissociation*), dentre outros [Cantú et al., 2008]. Gera-se um espectro por peptídeo, onde os picos representam fragmentos do peptídeo. O passo seguinte é a identificação dos peptídeos baseado no padrão dos picos espectrais.

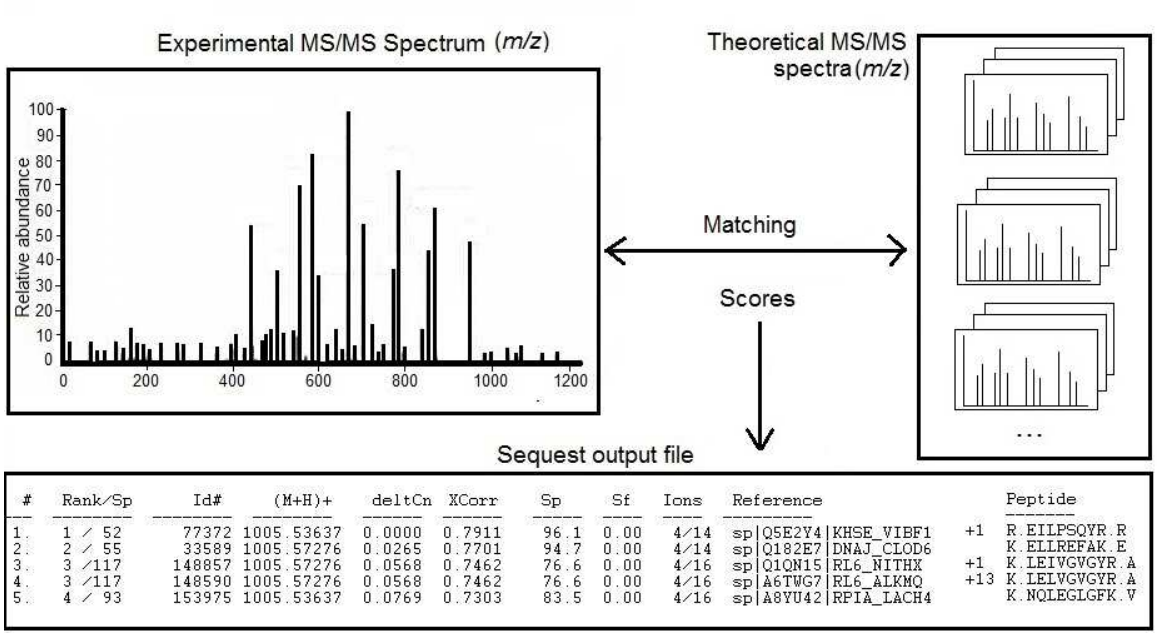
2.3.2 Interpretação do espectro LC-MS/MS

Em função do elevado número de espectros produzidos no LC-MS/MS, é essencial o uso de ferramentas computacionais para a atribuição de uma sequência peptídica de cada espectro, sendo que as principais abordagens utilizadas para interpretação do

espectro LC-MS/MS são: interpretação *de novo* e a busca em banco de dados (DB - *data base*). O sequenciamento de peptídeos *de novo* depende apenas de informações presentes no espectro, só verificando o padrão dos picos sem utilizar qualquer outra informação externa. O problema é que alguns aminoácidos e suas combinações podem apresentar valores de massa idênticos ou quase idênticos [Eng et al., 1994] [Perkins et al., 1999] [Ma et al., 2003] [Peng et al., 2003] .

A busca em DB é adequada se peptídeos de interesse são conhecidos e estão incluídos em um banco de dados de proteínas. Neste caso, as sequências proteicas são computacionalmente digeridas de acordo com as regras de clivagem da enzima. Os espectros teóricos são então calculados a partir das sequências peptídicas resultantes e combinados com os espectros observados. A melhor combinação é dada como a solução para cada espectro LC-MS/MS. O Sequest e o Mascot são os mais conhecidos e utilizados na abordagem de pesquisa DB [Ma et al., 2003] [Peng et al., 2003]. Como resultado, fornecido um espectro, os melhores candidatos peptídicos são atribuídos àqueles que têm massa próxima teórica (dentro de uma faixa de tolerância) à massa relatada pelo espectrômetro de massa. Os íons fragmentos teóricos dos candidatos obtidos são, em seguida, comparados com os picos observados. O Sequest e o Mascot geram várias pontuações (escores) que são atribuídas a esta associação de modo a medir a qualidade da mesma [Cerqueira et al., 2010] [Söderholm et al., 2014]. Este processo é ilustrado na Figura 2.7 e permite uma determinação qualitativa e quantitativa de compostos. A massa molecular é medida com uma precisão suficiente para detectar as suas menores variações, incluindo aminoácidos e peptídeos [Silverstein et al., 2014]. Por outro lado, um único experimento LC-MS/MS gera normalmente milhares de espectros e normalmente menos de 20% são interpretados corretamente [Cerqueira et al., 2010].

Uma técnica de processamento chamada autocorrelação é usada no Sequest visando determinar, matematicamente, a sobreposição entre o espectro teórico, derivado de uma sequência obtida no banco de dados, e o espectro experimentalmente obtido. O resultado de tal sobreposição é dado, quantitativamente, por duas pontuações mais conhecidas ($Xcorr$ e $\triangle Cn$) para cada peptídeo. O $Xcorr$ é um parâmetro que depende de diversos fatores, como o estado de carga do peptídeo, e o maior $Xcorr$ indica a melhor correspondência entre o espectro gerado no experimento e o teórico. A avaliação de uma segunda pontuação, classificado como $\triangle$Cn também é utilizada para que a confiabilidade do resultado obtido seja melhor estimada, e é definida como sendo a diferença entre os valores de $Xcorr$ obtidos para a sequência de aminoácidos que obteve o maior $Xcorr$ e a sequência de segundo melhor $Xcorr$ ($\triangle Cn = (Xcorr_{1st} - Xcorr_{2st})/Xcorr_{1st}$). O Sequest fornece as 10 sequências que obtiveram melhor associação a um determinado espectro observado. São uados diferentes critérios para classificar um peptídeo obtido



#	Rank/Sp	Id#	(M+H)+	deltCn	XCorr	Sp	Sf	Ions	Reference		Peptide
1.	1 / 52	77372	1005.53637	0.0000	0.7911	96.1	0.00	4/14	sp|Q5E2Y4|KHSE_VIBF1	+1	R.EILPSQVR.R
2.	2 / 55	33589	1005.57276	0.0265	0.7701	94.7	0.00	4/14	sp|Q182E7|DNAJ_CLOD6		K.ELLREFAK.E
3.	3 /117	148857	1005.57276	0.0568	0.7462	76.6	0.00	4/16	sp|Q1QN15|RL6_NITHX	+1	K.LEIVGVGYR.A
4.	3 /117	148590	1005.57276	0.0568	0.7462	76.6	0.00	4/16	sp|A6TWG7|RL6_ALKMQ	+13	K.LELVGVGYR.A
5.	4 / 93	153975	1005.53637	0.0769	0.7303	83.5	0.00	4/16	sp|A8YU42|RPIA_LACH4		K.NQLEGLGFK.V

Figura 2.7. Ilustração da abordagem utilizando banco de sequências de peptídeos para interpretação de espectros MS/MS, como a utilizada pelo Sequest. Primeiro são selecionados banco de peptídeos que têm massa de íon próxima ao do espectro MS/MS. Posteriormente o espectro MS/MS do experimento é comparado com o espectro teórico.

pelo espectro como satisfatório ou não, normalmente baseados em valores fixos: *Xcorr* maior que 3.75 para peptídeos com carga +3; *Xcorr* maior que 2.2 para peptídeos com carga +2 e *Xcorr* maior que 1.9 para peptídeos com carga +1 [Cantú et al., 2008]. Em todos os casos descritos o $\triangle Cn$ maior que 0.10 é exigido para que a correlação do peptídeo obtido pelo espectro (PSM - *peptide-spectrum matches*) seja considerado suficientemente confiável. As demais pontuações do Sequest são os valores $\triangle m$, *SpRank* e *PercIons*. O *RTp-value* é fornecido pelo algoritmo OpenMS para predição do tempo de retenção, valor calculado como o desvio entre os tempos de retenção observados e previstos [Pfeifer et al., 2007] [Cerqueira et al., 2010]. O Sequest tem se mostrado uma ferramenta eficiente, inclusive para análises de espectros com baixa relação sinal ruído [Eng et al., 1994] [Williams et al., 2014]. A abordagem LC-MS/MS, culminando na interpretação por software, é ilustrada na Figura 2.8.

2.4 Mineração de dados

Considerando-se que um único experimento LC-MS/MS gera milhares de espectros dos quais menos de 20% estão corretamente interpretados, é importante o desenvolvimento

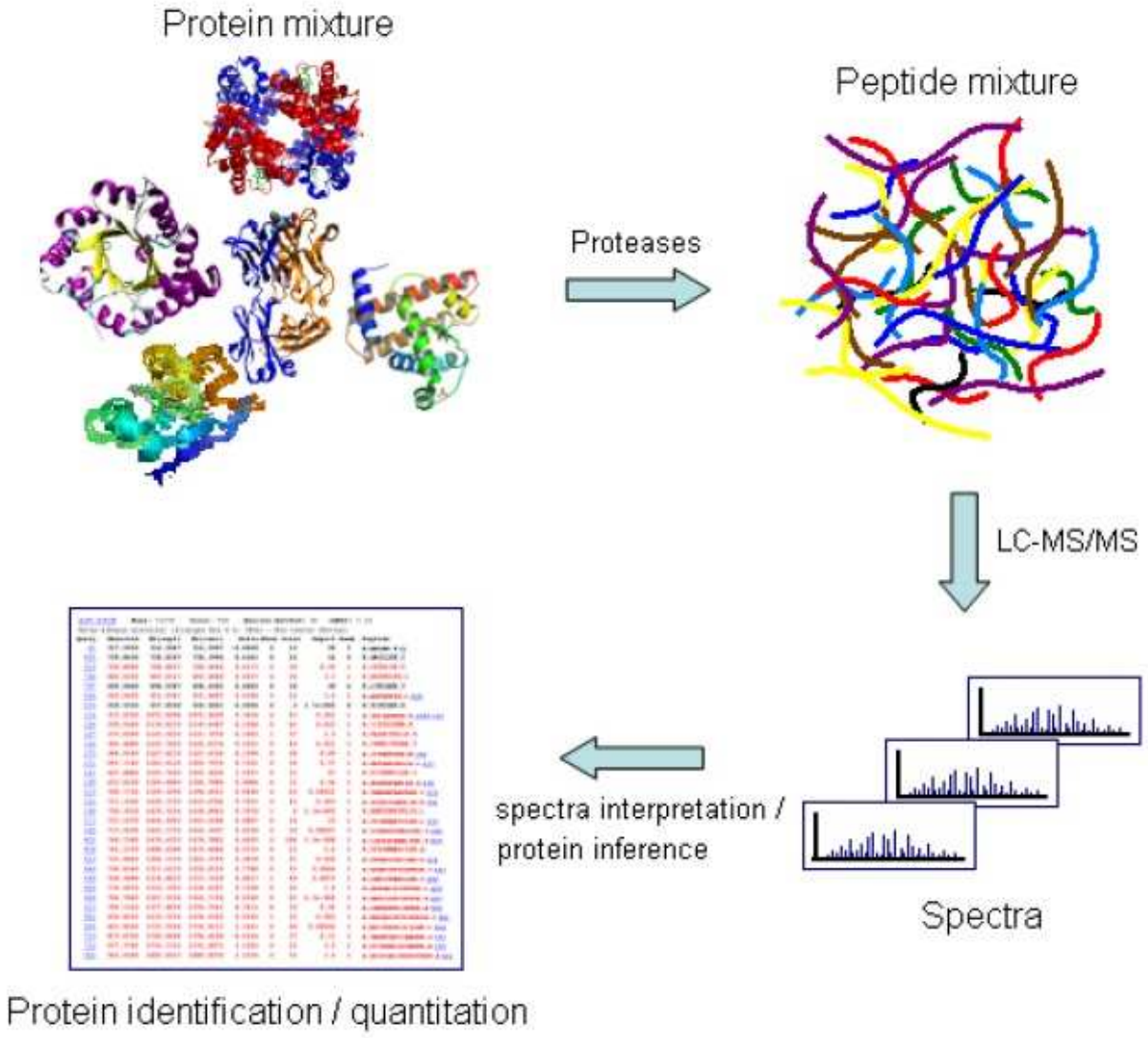


Figura 2.8. Processo de cromatografia líquida acoplada a espectrometria de massa em tandem - LC-MS/MS. Os espectros gerados são interpretados por softwares como o Sequest e o Mascot. Fonte: [Cerqueira et al., 2012].

de procedimentos de avaliação dos dados obtidos. A busca em BD, através do Sequest e do Mascot, é a abordagem mais utilizada para interpretação do espectro LC-MS/MS [Cerqueira et al., 2012]. Como resultado, as principais ferramentas computacionais para avaliação são construídas para extrair informações úteis dos dados gerados por estes softwares. Para tratar esse desafio, de extrair automaticamente informação útil em grandes volumes de dados, há o processo de mineração de dados. A mineração de dados (DM - *data mining*) é a tecnologia para processar grande volume de dados combinando métodos tradicionais de análise com algoritmos sofisticados, permitindo explorar e analisar dados novos e/ou antigos, objetivando um processo de descoberta

automática de informações úteis diante de grandes repositórios de dados [Tan et al., 2006].

A DM é composta basicamente por três tarefas: extração de regras de associação, agrupamento ou clusterização, e classificação/predição. As regras de associação associam as transações e os relacionamentos em bases de dados, identificando suas regras de ocorrências, além de determinar os itens ou conjuntos de itens que ocorrem com certa frequência. A classificação/predição utiliza um modelo de dados para predizer classes de objetos que ainda não foram classificadas permitindo prever valores desconhecidos ou futuros. O agrupamento ou clusterização trabalha sobre dados onde as classes não estão previamente definidas e a tarefa consiste em formar grupos de objetos, ou *clusters*, que sejam semelhantes entre si, sendo assim possível analisar e identificar suas características [Lan et al., 2011].

A DM é uma parte da descoberta de conhecimento em banco de dados (KDD - *Knowledge Discovery in Databases*), que visa a conversão de dados brutos em informações úteis. O KDD consiste de uma série de passos: entrada dos dados, pré-processamento, mineração dos dados, pós-processamento, obtenção das informações úteis [Tan et al., 2006]. O pré-processamento visa a transformação dos dados brutos da entrada em formatos adequados para as análises posteriores, incluindo a fusão de dados de diferentes fontes, a remoção de ruídos e dados duplicados, dentre outras características que sejam relevantes para a mineração dos dados. A DM é muitas vezes aplicada sobre dados que foram coletados para outro propósito. Em função disto a qualidade dos dados é tratada pela DM por meio da detecção e correção de problemas de qualidade de dados, chamado de limpeza dos dados, e pelo uso de algoritmos que possam tolerar a baixa qualidade dos dados. Normalmente é o passo mais demorado do KDD pela diversidade de formas através das quais os dados podem ser coletados e armazenados [Tan et al., 2006]. Já o pós-processamento tem como objetivo assegurar que somente resultados válidos e úteis vindos da DM sejam incorporados às informações para a tomada de decisão [Tan et al., 2006].

A DM é a confluência de técnicas como a aprendizagem de máquina, estatística, inteligência artificial, computação distribuída, computação paralela e banco de dados. O objetivo deste trabalho de pesquisa e a utilização das técnicas de classificação com aprendizagem de máquina e agrupamento, empregando-as no processamento de dados obtidos pelo processo LC-MS/MS.

2.5 Rede neural artificial e aprendizagem de máquina

Classificação é a tarefa de organizar objetos que estão entre diversas categorias pré-definidas em uma categoria. Os dados de entrada são conhecidos como registros ou instâncias, caracterizado por uma tupla (x,y), onde x é o vetor de atributos e y é um atributo especial conhecido como rótulo da classe. A classificação consiste em aprender uma função alvo f, conhecida como modelo de classificação, que mapeie cada vetor de atributos x para um dos rótulos de classes y pré-definidos [Tan et al., 2006]. A tarefa de classificação está ilustrada na Figura 2.9.

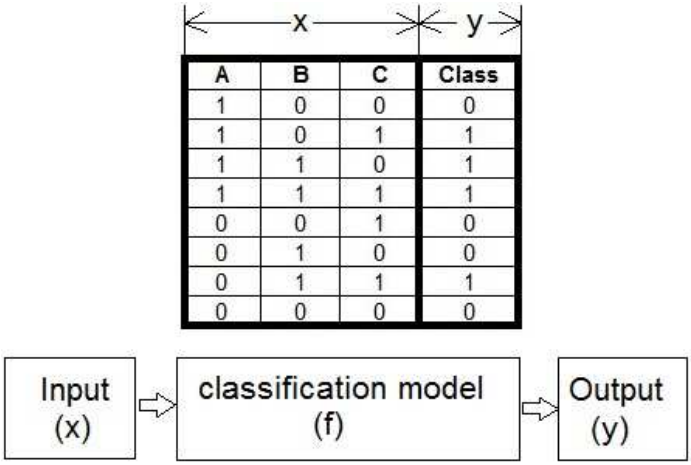


Figura 2.9. Ilustração da tarefa de classificação: definir um modelo f para mapear um vetor de atributos x (identificados como *A, B, C* na tabela) no seu rótulo de classe y pré-definido (identificado como *class* na tabela).

Um modelo de classificação pode ser usado para prever o rótulo da classe de registros não conhecidos. O modelo transforma-se em uma ferramenta para atribuir automaticamente um rótulo de classe quando recebe um vetor de atributos de um registro desconhecido [Tan et al., 2006]. Entre as técnicas de classificação, abordagem sistêmica para construir modelos de classificação a partir de um conjunto de dados de entrada, há os classificadores de árvores de decisão, classificadores baseados em regras, redes neurais artificiais, máquinas de vetores de suporte e classificadores Bayesianos. Cada técnica emprega um algoritmo de aprendizagem de máquina para identificar um modelo que seja mais adequado para o relacionamento entre o vetor de atributos e o rótulo da classe dos dados de entrada. O modelo gerado pelo algoritmo de aprendizagem deve ter boa capacidade de generalização, ou seja, deve adaptar-se aos dados de entrada e antecipar, corretamente, os valores de classe para os novos registros [Tan et al., 2006].

Tabela 2.1. Matriz de Confusão para um classificador de 2 classes - classe 0 e classe 1

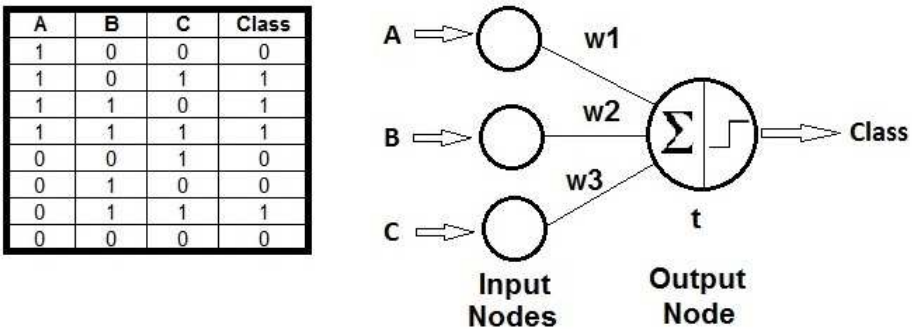
		Predited Class	
		0	1
Actual Class	0	TN	FP
	1	FN	TP

Um conjunto de treinamento é formado por registros cujos valores são conhecidos, inclusive a classe, e é usado para construir o modelo de classificação. Para avaliar o modelo, o mesmo é aplicado ao conjunto de testes, cujos rótulos dos registros são conhecidos, mas não são usados no processo de aplicação do modelo, de modo que se possa avaliar o mesmo posteriormente. A avaliação do desempenho de um modelo é baseada nas contagens de registros de testes previstos correta e incorretamente pelo modelo. Estas contagens são tabuladas em uma tabela chamada matriz de confusão, ilustrada na Tabela 2.1, onde os verdadeiros negativos (TN - *true negatives*) representam o número de registros previstos corretamente como classe 0 e os verdadeiros positivos (TP - *true positives*) denotam o número de registros previstos corretamente na classe 1. Em falsos negativos (FN - *false negative*) e falsos positivos (FP - *false positive*) estão representados respectivamente o número de registos previstos erroneamente como 0 e 1 [Tan et al., 2006].

O estudo das redes neurais artificais (ANN - *artificial neural networks*) foi inspirado em tentativas de simular os sistemas neurais biológicos, onde, de forma análoga à estrutura do cérebro humano, uma ANN é um conjunto interconectado de nodos (neurônios) e ligações direcionadas [Tan et al., 2006]. Um neurônio ou nodo é uma unidade de processamento de informação que é essencial para a operação de uma ANN. Há três elementos básicos de ligações entre os nodos para o modelo: um conjunto de sinapses ou elos de conexão, cada um caracterizado por um peso, que pode estar em um intervalo que inclui valores positivos e negativos; um somador para somar elementos de entrada da ANN, ponderados pelas respectivas conexões do nodo; e uma função de ativação com o objetivo de limitar a amplitude de saída de um nodo que é, normalmente, representada por um intervalo unitário fechado como 0 e 1 ou -1 e 1 [Haykin, 1998].

Uma ANN simples é conhecida como *perceptron* e está ilustrada na Figura 2.10. A tabela à esquerda ilustra um conjunto de dados de treinamento com um vetor de atributos *A, B, C* e um atributo de classe *Class*. O *perceptron* consiste de dois tipos de nodos: os que representam os atributos de entrada (*input nodes*) e o que representa

o nodo de saída (*output node*). Cada nodo de entrada é conectado por uma ligação ponderada (com diferentes níveis de importância) com o nodo de saída, identificados como *w1, w2, w3*, usada para emular a força da conexão sináptica entre os nodos [Tan et al., 2006]. Um *perceptron* calcula o seu valor de saída *Class* pela soma ponderada das entradas e pela subtração de um fator de tendência da soma *t*, examinando então o resultado. Durante a fase de treinamento os parâmetros de peso *w* são ajustados até que o resultado fique consistente com as saídas reais de exemplos de treinamento [Tan et al., 2006].



A	B	C	Class
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

Figura 2.10. Função com um *perceptron* simples. A tabela ilustra um conjunto de dados de treinamento com um vetor de atributos *A, B, C*, que formarão os nós de entrada, e um atributo de classe *Class* que representa o nó de saída. A força de conexão sináptica entre os nós é representada pelos diferentes níveis de importância e estão identificados como *w1, w2 e w3*. Fonte: adaptada de [Tan et al., 2006]

.

Uma função de ativação define a saída de um neurônio e três tipos básicos são bastante conhecidos: a função limiar produzindo um valor de saída binário (0 ou 1, por exemplo), a função linear parcial e a função sigmóide, sendo a última a forma mais comum de função de ativação [Haykin, 1998]. As funções de ativação estão representadas na Figura 2.11. A função de ativação é representada por $f(v)$ e o valor que induz a função de ativação é representado por v. Para a função limiar, $f(v)=1$ caso $v \geq 0$ ou f(v)=0 caso $v<0$. Na função limiar por partes, $f(v)=1$ caso $v \geq +1/2$, $f(v)=v$ caso $+1/2>v>-1/2$ e $f(v)=0$ caso $v \leq -1/2$. A função sigmóide, cujo gráfico tem forma de *s*, assume um intervalo contínuo de valores entre 0 e 1, e não somente os valores 0 e 1 como a função limiar. A função sigmóide é dada por $f(v)=1/(1+exp(-av))$, onde a é o parâmetro de inclinação da função sigmóide, usado para obter-se diferentes inclinações na curva [Haykin, 1998].

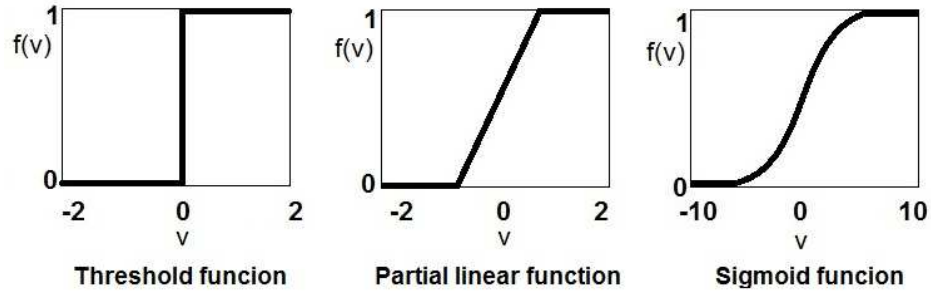


Figura 2.11. Tipos de função de ativação: (a) função Limiar, (b) função limiar parcial e (c) função simóide. A função de ativação é representada por *f(v)* no eixo y e o valor que induz a função de ativação é representado por v no eixo do x. Fonte: adaptado de [Haykin, 1998].

Complexidades adicionais podem ser adicionadas em uma rede *perceptron*. Uma delas é que a rede pode conter diversas camadas intermediárias entre as suas entradas e saídas. As camadas intermediárias são conhecidas como camadas ocultas e os nodos internos destas camadas são conhecidos como nodos ocultos, resultando em uma ANN multicamadas [Tan et al., 2006]. Esta estrutura está ilustrada na Figura 3.4 com 6 nós na camada de entrada e 4 nós na camada oculta, conforme utilizada neste trabalho de pesquisa. A função dos nodos ocultos é atuar entre a entrada externa e a saída da ANN, aperfeiçoando a eficácia da rede para de extrair estatísticas mais elaboradas, importante quando o tamanho da camada de entrada é grande [Haykin, 1998].

Uma propriedade importante para uma ANN é a habilidade de aprender a partir de seu conjunto de dados e melhorar o seu desempenho com o que aprendeu. A aprendizagem é por um processo interativo de ajustes aplicados aos pesos sinápticos da ANN e, após cada iteração desse processo, a rede conhece melhor o seu ambiente [Haykin, 1998].

2.6 Agrupamento

Agrupamento é a divisão dos dados em grupos (*clusters*) que tenham significado e/ou sejam úteis. É baseada em informações encontradas nos dados que descrevem os objetos e seus relacionamentos. Diferente das tarefas de classificação, o agrupamento é um problema não supervisionado, já que não há os rótulos de classe dos dados. O objetivo é que os dados dentro de um grupo sejam semelhantes ou relacionados entre si e diferentes ou não relacionados com outros objetos de outros grupos. Quanto maior o homogeneidade dentro de um grupo e maior a diferença entre os grupos, melhor será

o agrupamento [Tan et al., 2006]. O agrupamento de dados contribui para áreas de pesquisa que incluem a mineração de dados, estatística, aprendizagem de máquina, tecnologia de banco de dados espacial, biologia e *marketing* [Han, 2006].

A formação de agrupamentos não-supervisionados tem como maior problema a distinção entre várias categorias em uma coleção de objetos. Conforme a Figura 2.12, a formação de agrupamentos não-supervisionados começa com dados onde o problema de agrupamento é recuperar um modelo de estrutura como o da Figura 2.12(b) a partir dos dados brutos como os da Figura 2.12(a) [Stuart & Norvig, 2004].

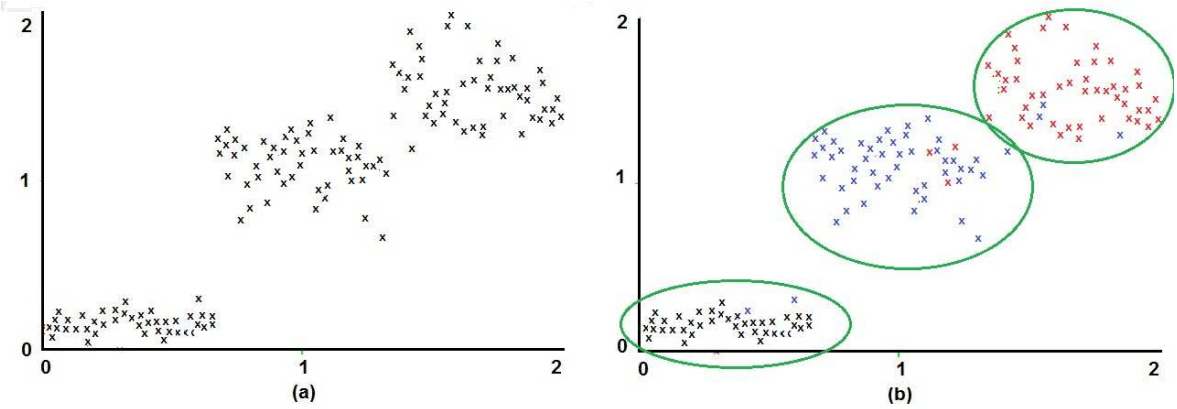


Figura 2.12. Pontos de dados em duas dimensões sugerindo a presença de 3 agrupamentos, onde em (a) estão os dados brutos e em (b) os agrupamentos reconstruídos a partir de um algoritmo de agrupamento. Fonte: adaptado de [Stuart & Norvig, 2004].

Há diversos tipos diferentes de agrupamento que se provam úteis na prática. Um dos tipos de grupo é conhecido como bem separados, onde um grupo é um conjunto de objetos no qual cada objeto está mais próximo a cada um dos outros objetos do grupo do que de qualquer outro objeto fora deste grupo, ou seja, a distância entre dois pontos quaisquer em grupos diferentes é maior do que a distância entre dois pontos quaisquer de um grupo. Um segundo tipo de grupo é o baseado em protótipos, que define o grupo como um conjunto de objetos que estão mais próximos ao protótipo ou objeto que caracteriza o grupo. O terceiro tipo é o baseado em densidades que define um grupo como uma região densa de objetos que seja rodeada por uma região de baixa densidade. Por fim o quarto tipo de grupo é conhecido como propriedades compartilhadas em que o grupo é definido como um conjunto de objetos que compartilham alguma propriedade [Tan et al., 2006].

Quatro algoritmos são normalmente usados para agrupamento: o K-means baseado em protótipos, o agrupamento hierárquico que envolve um conjunto de técnicas relacionadas em termos de agrupamento baseado em grafos e baseada em protótipo, o

DBSCAN que é um algoritmo baseado em densidade e o EM (*Expectation–maximization algorithm*) que é um método para estimar funções de máxima verossimilhança a partir de dados incompletos, conforme descrito na Seção 2.6.2 [Mitchell, 1997] [Han, 2006]. Neste trabalho de pesquisa foram considerados os algoritmos K-means e EM para o tratamento dos dados gerados no processo LC-MS/MS, pois o algoritmo de clusterização baseado em densidade não é adequado para grupos de densidades desiguais, como é o caso, e a clusterização hierárquica tem desempenho inferior no que refere-se ao tempo de resposta e uso de memória [Han, 2006].

2.6.1 K-means

O K-means é uma técnica de agrupamento baseada em protótipos onde um centroide, que é geralmente a média de um grupo de pontos, é definido como o protótipo. É um dos algoritmos de agrupamento mais antigos e amplamente usados [Tan et al., 2006].

O algoritmo K-means é parametrizado para a escolha de K centroides iniciais, onde K é o parâmetro especificado pelo usuário que define o número de grupos desejados. Cada ponto é então atribuído ao centroide mais próximo. Cada coleção de pontos atribuídos a um centroide forma um grupo. O centroide é então atualizado, conforme os pontos atribuídos ao seu grupo. Há uma iteração dessa fase do processo, onde os passos de atribuição e a atualização são repetidos, até que os centroides permaneçam inalterados ou nenhum ponto mude de grupo [Tan et al., 2006]. A Figura 2.13 ilustra o algoritmos K-means. Em (a) os pontos são atribuídos aos 4 centroides iniciais, usando-se a média para a definição do centroide. Em (b) os pontos são atribuídos aos centroides atualizados e os próprios centroides são atualizados novamente. Em (c) a atualização final dos centroides e dos pontos.

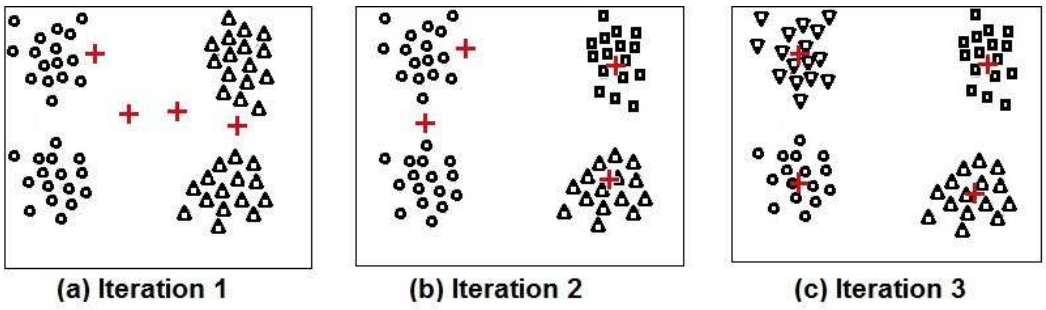


Figura 2.13. Ilustração do algoritmo K-means para encontrar quatro grupos de dados.

Como o K-means calcula iterativamente a semelhança de cada ponto com o centroide, as medidas para semelhança usadas pelo algoritmo são normalmente simples.

A distância Euclidiana é frequentemente usada para pontos de dados no espaço Euclidiano, enquanto que a semelhança do coseno é mais apropriado para documentos [Tan et al., 2006].

2.6.2 EM - *expectation–maximization*

O algoritmo maximização de expectativa (EM - *expectation–maximization*) baseia-se no princípio que, se alguma variável foi algumas vezes observada e outras não, pode-se utilizar os casos onde ela foi observada para aprender a predizer seus valores quando não observados. O algoritmo EM também pode ser usado para variáveis cujos valores nunca foram observados, desde que seja conhecida a forma geral da distribuição de probabilidade das variáveis. O algoritmo EM é um método para estimar funções de máxima verossimilhança a partir de dados incompletos [Mitchell, 1997].

A estimativa por máxima verossimilhança (MLE - *maximum likelihood estimation*) é um método para estimar os parâmetros de um modelo estatístico que, a partir de um conjunto de dados e fornecido um modelo estatístico, prevê valores para os diferentes parâmetros deste modelo que descrevem as características de uma população. O algoritmo EM, a partir de uma amostra dessa população, faz uma estimativa MLE dos parâmetros, ou seja, estima parâmetros que sejam os mais consistentes como os dados da amostra no sentido de maximizar a função de verossimilhança. A Figura 2.14 ilustra uma função linear alvo *(f)* representado pela linha sólida, e um conjunto de exemplos de treinamento ruidosos desta função alvo. A linha pontilhada corresponde à hipótese do MLE (h_{MLE}). A (h_{MLE}) não é necessariamente igual à hipótese correta *(f)* pois é inferida a partir de apenas uma pequena amostra de dados de treinamento ruidosos [Mitchell, 1997].

A ideia do EM é pressupor conhecer os parâmetros do modelo, e depois deduzir a probabilidade de cada ponto de dados pertencer a cada componente. Posteriormente readapta-se os componentes aos dados, onde cada componente é ajustado ao conjunto de dados inteiro, com cada ponto ponderado pela probabilidade de pertencer a esse componente. Este processo se repete até a convergência. Os dados são completados deduzindo distribuições de probabilidades sobre as variáveis ocultas - o componente ao qual pertence cada ponto de dados - com base no modelo atual. Para a mistura de distribuições gaussianas, inicia-se arbitrariamente os parâmetros do modelo de mistura, e repete-se as etapas E (*expectation*) e a M (*maximization*) [Stuart & Norvig, 2004].

A etapa E é o cálculo das probabilidades $p_{ij} = P(C = i|x_j)$, que é a probabilidade de que o dado x_j tenha sido componente de i. A etapa M é o cálculo da nova média, a co-variância e os pesos de componentes, encontrando os novos valores dos parâmetros

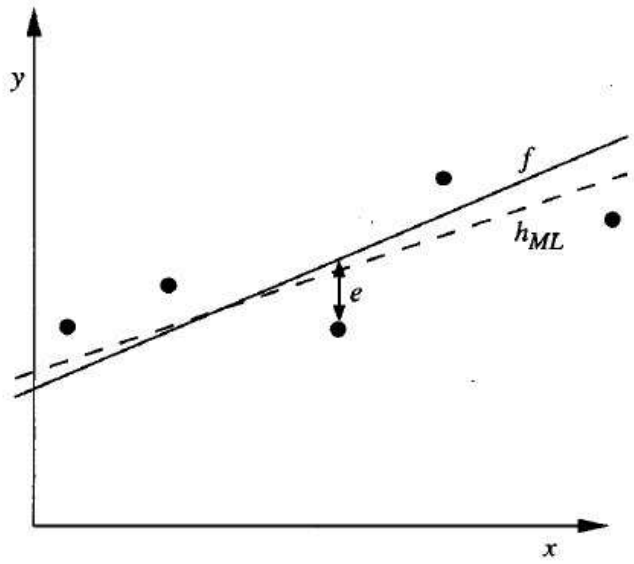


Figura 2.14. Função linear alvo e função de verossimilhança: a função alvo f corresponde à linha sólida. Os exemplos de treinamento (x_i) são assumidos e e_i é o ruído. A linha tracejada corresponde à função linear que minimiza a soma dos quadrados dos erros. Portanto, é a hipótese de máxima verossimilhança tendo em conta estes cinco exemplos de treinamento. Fonte: [Mitchell, 1997].

que maximizam a probabilidade logarítmica dos dados, fornecidos os valores esperados das variáveis indicadoras ocultas [Stuart & Norvig, 2004].

Capítulo 3

Materiais e Métodos

3.1 Conjuntos de dados

Os mesmos conjuntos de dados usadas nos projetos de pesquisa MUDE e MUMAL foram os conjuntos originais para este trabalho. A Figura 3.1 ilustra os 11 conjuntos usados com as respectivas quantidades de PSMs totais (coluna à esquerda) e PSMs alvo ou não-*decoys* (coluna à direita), de cada uma. Em todos os casos o número de PSMs alvo é ligeiramente maior do que a metade do número total de PSMs. Isto é esperado, uma vez que a expectativa é de que menos de 20% das instâncias reais estejam corretas. Portanto, a quantidade total de PSMs é composta por este pequena porcentagem, mais o resto de PSMs incorretos, onde, aproximadamente, metade destes é composta de instâncias *decoys* e a outra metade de falsos positivos (PSMs alvos incorretos).

Os arquivos referentes a estes conjuntos de dados foram obtidos conforme método descrito no MUDE [Cerqueira et al., 2010], onde três conjuntos foram inicialmente produzidos a partir de três amostras independentes fosfoenriquecidas. Os espectros MS/MS foram convertidos para arquivos *dta*, o formato de arquivo de texto do Sequest para espectros MS/MS, resultando em 24405 (S1), 23668 (S2) e 18996 (S3) espectros, respectivamente. Em seguida, o Sequest é executado para atribuir sequências de peptídeos para cada espectro. Cada conjunto de dados (com o seu respectivo Sequest *output*) foi dividido em duas partes, uma contendo espectros cujo melhor resultado foi relatado como um fosfopeptídeo, e outra composta pelos espectros cuja a melhor indicação foi atribuída a um não-fosfopeptídeo. Cada parte foi dividida com base no estado de carga do precursor, onde somente cargas +2 e +3 foram consideradas. Como resultado, os três conjuntos de dados iniciais geraram doze conjuntos de dados que foram rotuladas como S1_P_CH2, S1_P_CH3, S1_NP_CH2,

S1_NP_CH3, S2_P_CH2, S2_P_CH3, S2_NP_CH2, S2_P_CH3, S3_P_CH2, S3_P_CH3, S3_NP_CH2 e S3_NP_CH3, onde "P"e "NP"denotam *phosphodata* e *no-phosphodata*, respectivamente, enquanto que "CH2"e "CH3"representam os estados de carga 2 e 3, respectivamente. O conjunto de dados S3_NP_CH3 foi removido dos experimentos, uma vez que mostrou conter menos de 10 atribuições corretas. Finalmente os arquivos contendo as atribuições produzidas pelo Sequest de cada conjunto foi convertido em um arquivo $IdXML$, formato utilizado pelo algoritmo (OpenMS v1.4) para a predição do tempo de retenção (*RTp-value*) [Pfeifer et al., 2007] e [Cerqueira et al., 2012]. Detalhes do conteúdo de cada conjunto e o método de obtenção das mesmas podem ser consultados em [Cerqueira et al., 2010] e [Cerqueira et al., 2012].

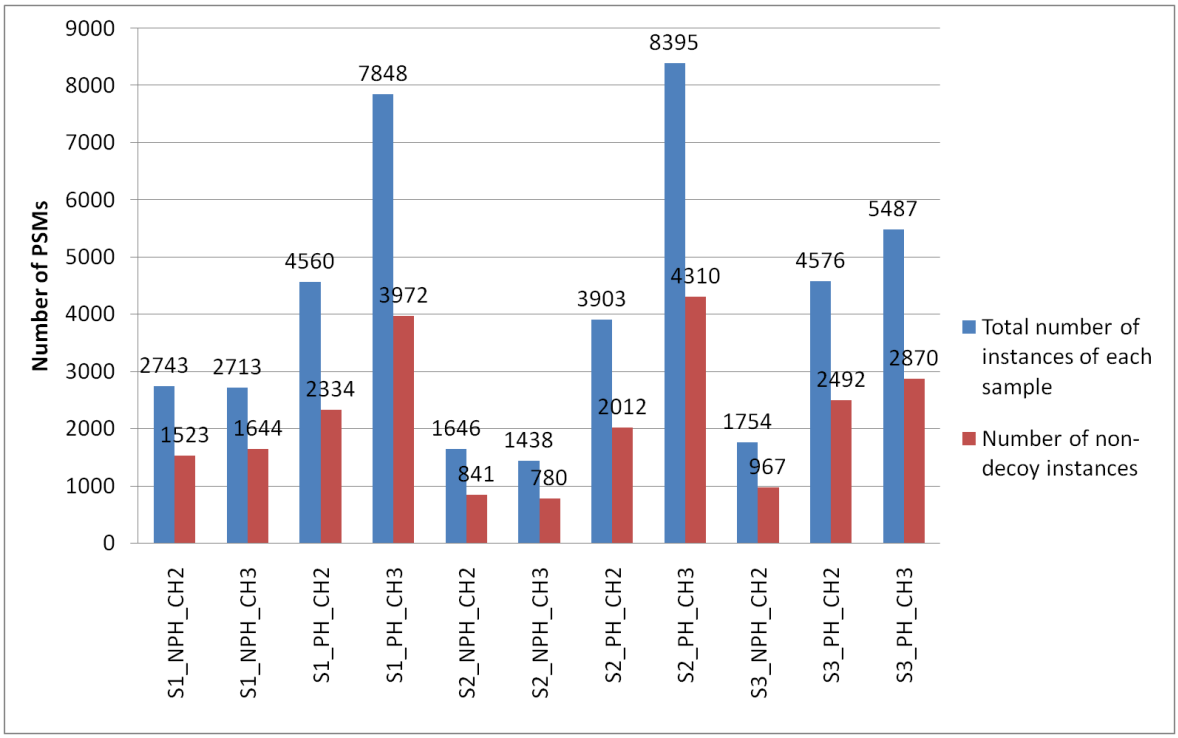


Figura 3.1. Representação gráfica dos 11 conjuntos de dados usadas para os experimentos do projeto. A coluna à esquerda representa a quantidade total de instâncias de cada amostra e a coluna à direita representa a quantidade de instâncias alvo (não-*decoys*).

Um aspecto importante a ser observado na análise da eficiência da abordagem é a correlação entre o número de corretos preditos e o número de corretos que realmente há na amostra, para os vários limites de decisão. Para verificar esta correlação foi usada um outro conjunto contendo dados de uma mistura constituída por proteínas previamente identificadas [Pfeifer et al., 2007]. Este conjunto de dados foi denominada como

M123 e foi usada nas abordagens MUDE e MUMAL. O M123 foi obtida a partir de três conjuntos de dados adicionais conforme descrito em [Pfeifer et al., 2007] e [Cerqueira et al., 2010], cujas proteínas constituintes são conhecidas previamente. As proteínas identificadas na mistura que gerou o conjunto M123 são: β-caseína (leite bovino), conalbumina (clara de ovo - frango), proteína básica de mielina (bovino), hemoglobina (humanos, divididos em subunidades alfa e beta), leptina (humano), creatinofosfoquinase (tecido muscular de coelho), α1-glicoproteína ácida (plasma humano, que aparece em duas versões distintas do M123), albumina (soro bovino), citocromo C (coração bovino), β-lactoglobulina A (bovino), anidrase carbônica (eritrócitos bovinos), catalase (fígado bovino), mioglobina (coração de cavalo), lisozima (ovo de galinha branca), ribonuclease A (pâncreas bovino), transferrina (bovino), β-lactalbumina (bovino), albumina (soro bovino), tireoglobulina (tireoide bovina) e albumina (soro bovino). Para a avaliação da eficiência na identificação de peptídios nas abordagens MUDE, MUMAL e MUMAL2, os arquivos de saída de cada conjunto de dados produzidos pelo Sequest foram convertidos para um único arquivo com extensão IdXML M123 [Cerqueira et al., 2010]. Maiores detalhes sobre as proteínas presentes nos dados desta mistura e sobre o método de obtenção podem ser consultados em [Pfeifer et al., 2007], [Cerqueira et al., 2010] e [Cerqueira et al., 2012].

3.2 A estratégia TDDB

Cada experimento LC-MS/MS pode gerar milhares de espectros MS/MS que são posteriormente comparados com sequências peptídicas em bancos de dados. Devido à extensão do espaço de pesquisa, um número alto de PSMs são provavelmente falso-positivos, ou seja, a sequência peptídica que foi associada ao espectro não é correta. Em função disto, a medida estatística FDR é importante no processo de identificação de dados de proteômica. Essa abordagem envolve a comparação de espectros MS/MS experimentais com um banco de dados falso e equivalente em tamanho ao banco de dados de sequência de proteínas usado. Este BD falso contém sequências *decoy* de proteínas geradas a partir das sequências alvo presentes no BD de sequências reais de proteínas, do mesmo organismo de onde a amostra foi extraída [Elias & Gygi, 2007].

Considerando que um PSM falso, que surgiu ao acaso, tem 50% de chance de vir do BD de sequências *decoy* e 50% de vir do BD sequências alvo, o número de PSMs reais incorretos pode ser estimado pelo número de PSMs *decoys*. Desta maneira o FDR é determinado por $D_T/(N_T - D_T)$, onde D_T é o número de PSMs *decoys* encontrados com escore acima de um valor de corte predeterminado T (*threshold*), e

N_T é o número total de identificações de peptídeos (*decoys* e reais), utilizando o mesmo *threshold* T [Elias & Gygi, 2007]. Esta é a estratégia TDDB (*target-decoy database search*) e está ilustrada na Figura 3.2 extraída de [Cerqueira et al., 2012] e tem como vantagem funcionar sem uma suposição preliminar sobre a distribuição dos dados. No entanto, em sua forma original, a estratégia TDDB não considera a sensibilidade, ou seja, nenhuma estratégia computacional e métricas de desempenho são aplicadas para encontrar conjuntos alternativos de PSMs com o mesmo FDR, porém com um número maior de PSMs [Cerqueira et al., 2012].

O MUDE e MUMAL são dois trabalhos de pesquisa que exploram a estratégia TDDB. Ambos os métodos utilizam os escores PSM de forma mais abrangente para aumentar a sensibilidade. O MUDE considera, além do $Xcorr$ e do $\triangle Cn$, normalmente usados no TDDB, outras quatro contagens alternativas: $\triangle M$, $SpRank$, porcentagem de íons encontrados (todos eles calculados pelo Sequest), e /textitRTp-value (calculado pelo OpenMS) [Pfeifer et al., 2007] [Cerqueira et al., 2010]. Além disso, o problema de encontrar valores de *threshold* para os resultados que conduzem a um desejado FDR é tratado como um problema de otimização, ao contrário dos procedimentos simplistas descritos anteriormente. Mesmo proporcionando um aumento significante na sensibilidade, a abordagem MUDE é capaz de produzir apenas limites de decisão lineares para separar os falsos positivos a partir de verdadeiros positivos. Assim como no MUDE, o MUMAL e as abordagens deste trabalho de pesquisa também aplicam a estratégia TDDB para avaliar PSMs utilizando a análise multivariada, com os seis escores citados. No entanto, isso é feito com técnicas de aprendizagem de máquina, visando proporcionar limites de decisão mais flexíveis para aumentar ainda mais a sensibilidade no processo de estimativa de FDR [Cerqueira et al., 2012] [Swan et al., 2013].

3.3 A utilização da aprendizagem de máquina

A estratégia LC-MS/MS para identificação de proteínas pode gerar milhares de espectros em uma única execução, resultando em milhares de sinais para serem comparados e analisados. Dentre estes sinais uma grande quantidade é gerada pela presença de sinais de fundo como ruídos químicos e eletrônicos. Em função disto, os algoritmos tradicionais para interpretação destes sinais geram uma alta taxa de falsos positivos. As técnicas de aprendizagem de máquina são um meio viável para alcançar uma maior taxa de verdadeiros positivos na classificação destes dados, pois faz uso de algoritmos capazes de analisar dados complexos e detectar padrões que seriam impossíveis de

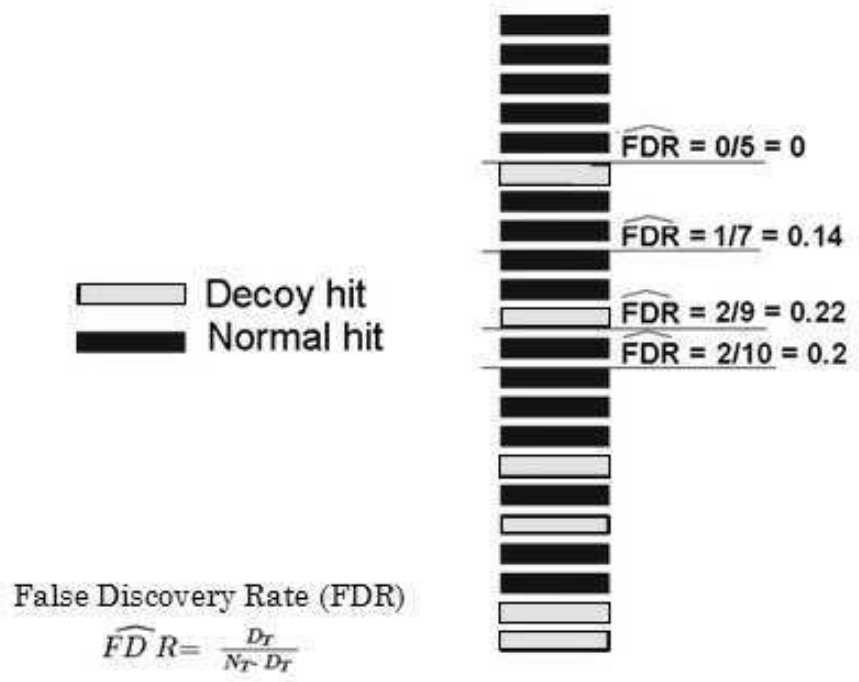


Figura 3.2. FDR (*False Discovery Rate* - Taxa de Descobertas Falsas)

FDR é determinado por $D_T/(N_T - D_T)$, onde D_T é o número de PSMs *decoys* encontrados com escore acima de um valor de corte predeterminado (*threshold*)T, e N_T é o número total de identificações de peptídeos (*decoys* e reais), utilizando o mesmo *threshold* T. Fonte: [Cerqueira et al., 2012].

serem determinados manualmente ou por algoritmos tradicionais.

O MUMAL utiliza a estratégia TDDB e a aprendizagem de máquina para avaliar os PSMs, em particular, a aprendizagem supervisionada que prevê a aprendizagem a partir de dados de treinamento conhecidos. No caso do uso da estratégia TDDB, os PSMs estão definidos como classes 0 (*decoy*) e 1 (não-*decoy*) e servem como o conjunto de treinamento. De uma forma geral, a tarefa é estabelecer generalizações que possam predizer a saída para dados válidos, baseado no que foi aprendido com os dados de treinamento. No MUMAL o modelo gerado é aplicado nos próprios dados usados na fase de aprendizagem, conforme ilustrado na Figura 3.3. O autores ainda observaram que PSMs rotulados como *decoy* classe 0 e PSMs alvos como classe 1 levam a uma tarefa de classificação difícil, porque a maioria dos PSMs classe 1 são incorretos, ou seja, eles são semelhantes aos *decoys*. No entanto, o objetivo não é fornecer uma separação perfeita entre as instâncias classes 0 e 1. Uma vez que um modelo é criado, diferentes probabilidades são testadas para se obter uma que resulte em um FDR procurado. A contagem de *decoys* é a chave para separar o que realmente importa, ou seja, corretos de incorretos. Vale destacar também que a aprendizagem

de máquina é uma primeira etapa do MUMAL e é complementada por uma análise de custo/benefício para diferentes limites de probabilidades por meio da curva ROC, conforme está descrito na seção 3.5.

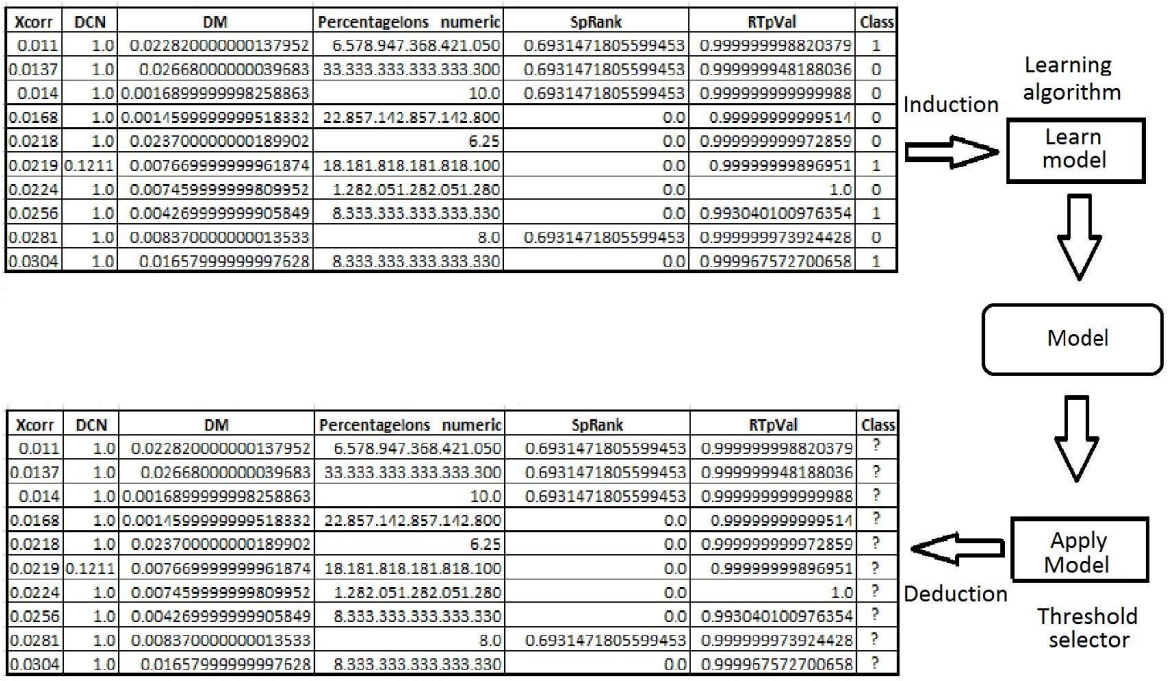


Figura 3.3. Ilustração da estratégia de classificação usando a aprendizagem de máquina supervisionada.

É importante notar que o método de otimização usado no MUDE produz apenas limites de decisão lineares, quando limites de decisão não-lineares poderiam proporcionar uma sensibilidade ainda mais elevada para o mesmo FDR. Os algoritmos não-lineares de aprendizagem podem estabelecer um limite de decisão mais apropriado, conduzindo a uma sensibilidade mais elevada. A técnica de aprendizagem de máquina ANN pode propiciar uma função mais complexa de combinar os escores visando a definição de limites de decisão não-lineares [Cerqueira et al., 2012]. A abordagem MUMAL usa a ANN para este fim. A ANN, conforme usada no MUMAL, pode conter vários nodos em uma camada intermediária chamadas camada oculta, que pode ser observada na Figura 3.4. A fim de construir limites de decisões mais apropriados entre corretos e incorretos, a ANN é usada para que os seis escores (entradas da ANN) mencionados na Seção 3.2 sejam aplicados em combinação para produzir uma pontuação final no intervalo [0, 1] (a saída da ANN usando uma função sigmóide) que pode ser interpretado como uma probabilidade valor.

Neste trabalho a proposta é melhorar a abordagem MUMAL para aumentar ainda mais a sensitividade na avaliação de PSMs.

3.4 A utilização da matriz de Custos

A avaliação do desempenho de um algoritmo que usa a técnica ANN para um modelo de classificação é baseada na análise da habilidade de prever a correta separação das classes. O objetivo é minimizar os erros de classificação de um determinado modelo. Normalmente o classificador monta um modelo em que os custos de FP e FN são os mesmos, porém há situações em que o custo da predição incorreta pode ser relevante. No contexto da avaliação de dados de espectrometria de massas, a questão está na percepção do custo associado ao avaliar um PSM como correto (classe 1) e este esteja incorreto (falso positivo) ou de avaliá-lo como incorreto (classe 0) e o mesmo esteja correto (falso negativo) [Elkan, 2001] [Lan et al., 2011]. O MUMAL, usando a técnica ANN, não considera este custo associado às predições incorretas e às possivelmente corretas. Desta forma, o custo de uma predição incorreta, seja para um falso positivo ou para um falso negativo, é a mesma. Porém o desafio é que a classe 1 contém, em sua maioria, dados incorretos e que deveriam migrar para a classe 0.

Os custos das condições incorretas podem ser incorporados no processo de construção do modelo de aprendizagem [Lan et al., 2011]. No caso de duas classes, os custos podem ser sintetizadas na forma de uma matriz de 2 x 2, em que os elementos da diagonal principal representam os dois tipos de classificação corretas e os elementos fora desta diagonal representam os dois tipos de classificação erradas [Lan et al., 2011].

Neste trabalho de pesquisa, a abordagem MUMAL2 mostrou-se superior ao MUMAL ao aumentar ainda mais a sensibilidade na avaliação PSMs. A estratégia escolhida foi usar as instâncias *decoy* do conjunto de dados de modo a melhorar a capacidade do modelo para separar corretos de incorretos. Para este efeito, uma matriz de custo foi introduzido para a tarefa classificação. Pode-se associar custos a verdadeiros positivos(cTP), a verdadeiros negativos (cTN), a falsos positivos (cFP) e a verdadeiros negativos (cTN) e um melhor desempenho pode ser obtido se o classificador for forçado pelo algoritmo de aprendizagem a usar a matriz de custo [Elkan, 2001] [Tan et al., 2006] [Lan et al., 2011].

No caso das duas classes usadas no classificador da abordagem MUMAL2 (classes 0 e 1) a ideia é gerar dados de treinamento com diferentes proporções para os casos de FP e FN (elementos fora da diagonal principal da matriz de confusão). Com o esquema de aprendizagem esforçando-se para minimizar o número de erros e associando-se, por

Tabela 3.1. Matriz de custo para um classificador de 2 classes - classe 0 e classe 1

		Predited Class	
		0	1
Given Class	0	cTN=0	cFP=10
	1	cFN=1	cTP=0

exemplo, um custo alto para os erros da classe 1 e um custo mínimo para erros da classe 0, o classificador chegará a uma estrutura de decisão que está inclinado para evitar os erros na classe 0 [Elkan, 2001] [Lan et al., 2011]. Os erros da classe 0 são penalizados justamente pela certeza que estes são errados, pois são *decoy*. Ou seja, o classificador deve gerar um modelo que acerte a classe 0 e que possa predizer melhor os dados de classe 1 que estejam errados. Este cenário é ilustrado na Tabela 3.1. Neste exemplo, uma matriz de custos força o modelo para favorecer instâncias *decoy*. A idéia é provocar um modelo favorável aos casos de *decoy*, levando à reclassificação de instâncias alvo erradas para classe 0, resultando em melhores limites de decisão para separar PSMs corretos de incorretos.

No MUMAL2 a matriz de custo é introduzida na tarefa de classificação, considerando-se os não-*decoys* como positivos e que o custo de um falso positivo é maior do que o custo de um falso negativo. Por conseguinte, o modelo final tenderá a classificar casos de classe-0 corretamente, enquanto que instâncias classe-1 serão "erroneamente classificada". As aspas são para chamar a atenção para o fato de que o objetivo final é o de construir um modelo de separação entre PSMs corretos dos incorretos, e não separar *decoys* de não-*decoys*. Consequentemente, a maioria das instâncias classe 1 serão classificados como classe 0 pelo modelo, que são, na verdade, corretamente remarcadas para classe 0, pois as suas sequências de peptídeos foram incorretamente atribuídas. Esta classificação sensível custo foi implementado na linguagem de programação Java usando o API v3.7.8 Weka [Lan et al., 2011] [Hall et al., 2009].

Conforme os experimentos realizados neste projeto de pesquisa, o processo de rotulagem das instâncias, usando a ANN sensível ao custo, conduz a resultados satisfatórios no aumento da sensibilidade. No entanto, após a fixação cFN=1 e tentando valores diferentes para cFP, percebeu-se que um certo cFP que leva a um bom modelo para um determinado conjunto de dados, não é necessariamente a melhor escolha para um outro conjunto de dados. Como resultado, para cada conjunto de dados fornecido como entrada para o MUMAL2, dez modelos diferentes são criados, variando-se

o cFP com valores inteiros no intervalo [1, 10]. São geradas muitas probabilidades discriminantes, sendo que cada um dos 10 modelos modelos gera uma probabilidade discriminante diferentes para cada FDR. Em seguida, o modelo com o maior número médio de PSMs corretos, para FDRs que variam de 1 a 5%, é selecionado para a classificação final.

3.5 A utilização da curva ROC

Conforme ilustrado na Figura 3.4, tanto no MUMAL quanto no MUMAL2, os valores gerados pelo *Sequest* (*Xcorr,*$\triangle$*Cn,* $\triangle$*M, PercentIons, SpRank*), além do *RTp-Value* formam os atributos dos dados de classificação. É importante notar que muitos classificadores binários podem gerar valores de probabilidades como função de ativação. Usando a *sigmoid* o valor real é mapeado no intervalo entre 0 e 1, podendo ser interpretado como uma probabilidade. Neste caso, o modelo pode ser construído de tal maneira que a probabilidade de 0.5 seja determinada, por exemplo, como o valor limite para decidir a qual classe um dado pertence [Haykin, 1998].

Na abordagem MUMAL o treinamento e a aplicação da ANN é apenas o primeiro estágio do processo. Em uma segunda fase há uma análise de custo/benefício para diferentes *thresholds* de probabilidades, visando alcançar um melhor valor de decisão para um FDR máximo predefinido. Após a construção do modelo, há a possibilidade de variar o *threshold* até obter um valor que conduza a um FDR não superior ao previamente definido. Este procedimento é possível com o uso da curva ROC (*receiver operating characteristic*)[Tan et al., 2006] [Cerqueira et al., 2012]. A curva ROC permite estudar a variação da sensibilidade e especificidade para diferentes valores de corte em um modelo de aprendizagem. É uma representação gráfica da taxa de verdadeiros positivos (TPR - *true positive rate*) no eixo das ordenadas pela taxa de falsos positivos (FPR) no eixo das abscissas para vários *thresholds* discriminantes distintos [Tan et al., 2006]. O TPR é a quantidade de verdadeiros positivos (TP) dividido pelo total de positivos, ou seja, verdadeiros positivos (TP) somados aos falsos negativos (FN), sendo representado pela fórmula $TPR = TP/(TP + FN)$ [Lan et al., 2011]. O FPR é a quantidade de falsos positivos (FP) dividido pelo total de negativos da amostra, ou seja, falsos positivos (FP) e verdadeiros negativos (TN) e é representado pela fórmula FPR=FP/(FP+TN) [Lan et al., 2011]. A Figura 3.5 mostra a curva ROC gerada a partir de um modelo de ANN para o conjunto de dados S2_NP_CH2 onde o cálculo FDR aqui equivale ao resultado da quantidade de instâncias classificadas como *decoy* dividido pela quantidade de valores identificados como reais para um determinado valor

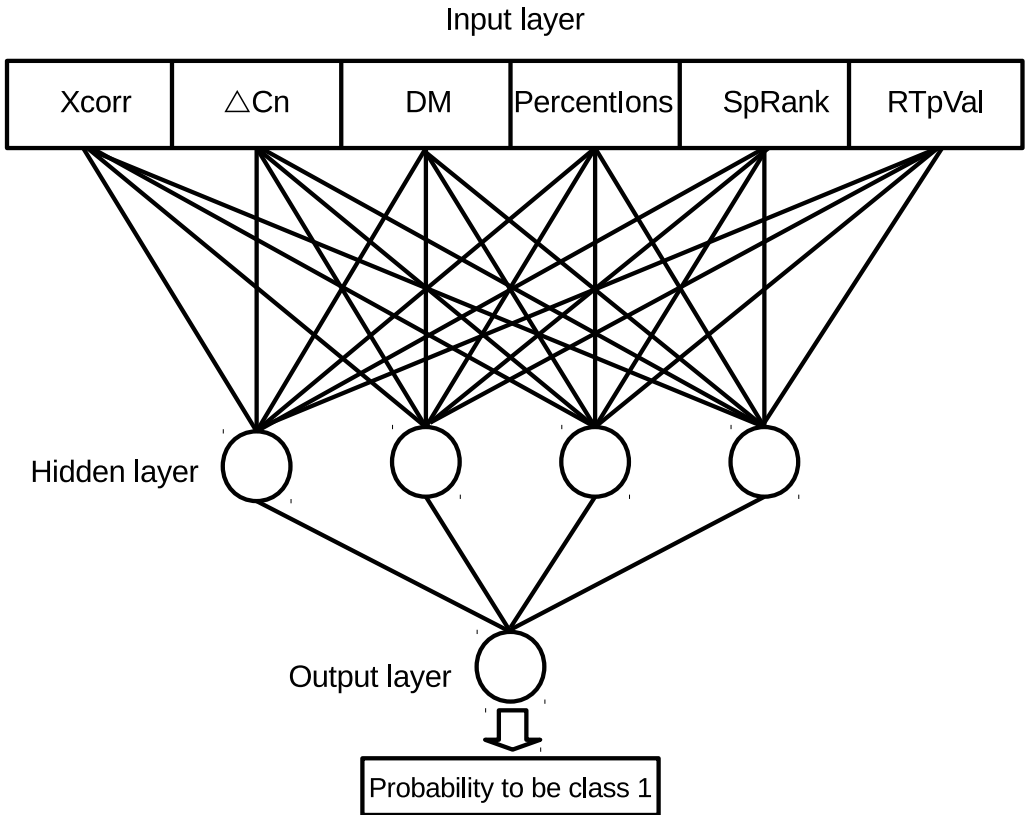


Figura 3.4. Arquitetura de uma Rede Neural ilustrando as camadas de entrada (6 nós), escondida (4 nós) e de saída (1 nó), com a função de ativação que pega o resultado da combinação linear das conexões e mapeia este resultado num valor final de saída segundo a função sigmoid.

de *threshold*. O software usado foi o Weka 3.7.8, algoritmo de classificação ANN de múltiplas camadas (MLP - *multilayer perceptron*) com 4 nós na camadas oculta e taxa de aprendizagem (*learning rate*) 0.2 [Hall et al., 2009]. A maior aproximação da curva ao eixo Y, bem como a maior da área sob à curva (AUC - *area under the curve*) determinam a melhor avaliação do modelo. A AUC com valor mais próximo a 1 (o valor máximo) é o melhor. Valores de AUC próximos a 0.5 mostram que a capacidade do modelo para discriminar entre a classificação correta ou errada é devido ao acaso [Lan et al., 2011].

No MUMAL várias probabilidades discriminantes são exploradas após a construção do modelo, de modo que a contagem de *decoys* considerados positivos sirva como uma estimativa para o número de alvos positivos errados. Esta tarefa é realizada analisando a curva ROC resultante [Tan et al., 2006]. Para cada ponto da curva, a respectiva probabilidade discriminante t é usada para contar o número de *decoys* com probabilidade (gerada pela ANN) igual ou superior a t. Esta contagem é a estimativa para o

número de PSMs reais com P>t que estão incorretos. Conforme já descrito, as classes nos conjuntos de dados são: 0 para *decoys* , e 1 para reais. Espera-se que uma pequena parte de classe 1 esteja correta. Por esta razão, através da utilização de métodos de avaliação clássicos tais como exatidão, precisão e área sob a curva (AUC), é esperado um modelo de classificação pobre. No entanto, o objetivo não é separar *decoys* de reais, mas incorretos de corretos, e a análise ROC estabelece probabilidades discriminantes adequadas que proporcionam o FDR desejado para uma melhor sensibilidade quando comparado com as abordagens TDDB clássicas.

Nessa abordagem a AUC apresenta valores próximos a 0.6, sendo que especificamente para o conjunto S2_NP_CH2 ilustrada na Figura 3.5, AUC=0.5525. Isto acontece justamente pela dificuldade de distinção entre *decoys* e alvos errados, explicada na Seção 3.3. A curva ROC gerada para a ANN complementa a abordagem, sendo utilizada para visualizar o ponto que pode ser selecionado como o melhor para obter uma quantidade de falsos positivos aceitável. Isto também é ilustrado na Figura 3.5 onde para um FDR escolhido é possível visualizar quantos PSMs classificados pela ANN foram identificados como verdadeiros positivos e falsos positivos. Para cada ponto na curva (Figura 3.5a), a probabilidade t discriminante (*threshold*) e as respectivas estatísticas (Figura 3.5b) são conhecidas. A estimativa FDR (Figura 3.5c) é feita dividindo-se o número de falsos positivos (FP) pelo número de verdadeiros positivos (TP) porque FP e TP são, respectivamente, os *decoys* e alvos com $P > t$.

3.6 Algoritmo *Threshold Selector*

Na abordagem MUMAL2, deste trabalho de pesquisa, a ANN é associada, além da matriz de custos e da curva ROC, ao algoritmo TSA (*Threshold Selector*) implementado pelo Weka 3.7.8. Este algoritmo modifica a probabilidade discriminante de um classificador com o objetivo de otimizar alguma medida de desempenho. Conforme explicado na Seção 3.5, a análise ROC dá chance de selecionar um valor limite de probabilidade adequado que conduz a um FDR desejado. Isto é o suficiente se o objetivo é apenas a seleção de um conjunto de PSMs com baixo FDR. Entretanto, tal conjunto é normalmente utilizado na fase de inferência de proteína, em que as proteínas da amostra são, em última análise, identificadas. Para este fim, a probabilidade associada a cada PSM é de grande importância, principalmente para ferramentas computacionais que utilizam este valor como uma medida essencial para inferir proteínas como, por exemplo, ProteinProphet. Por outro lado, os valores de probabilidade originalmente atribuídas

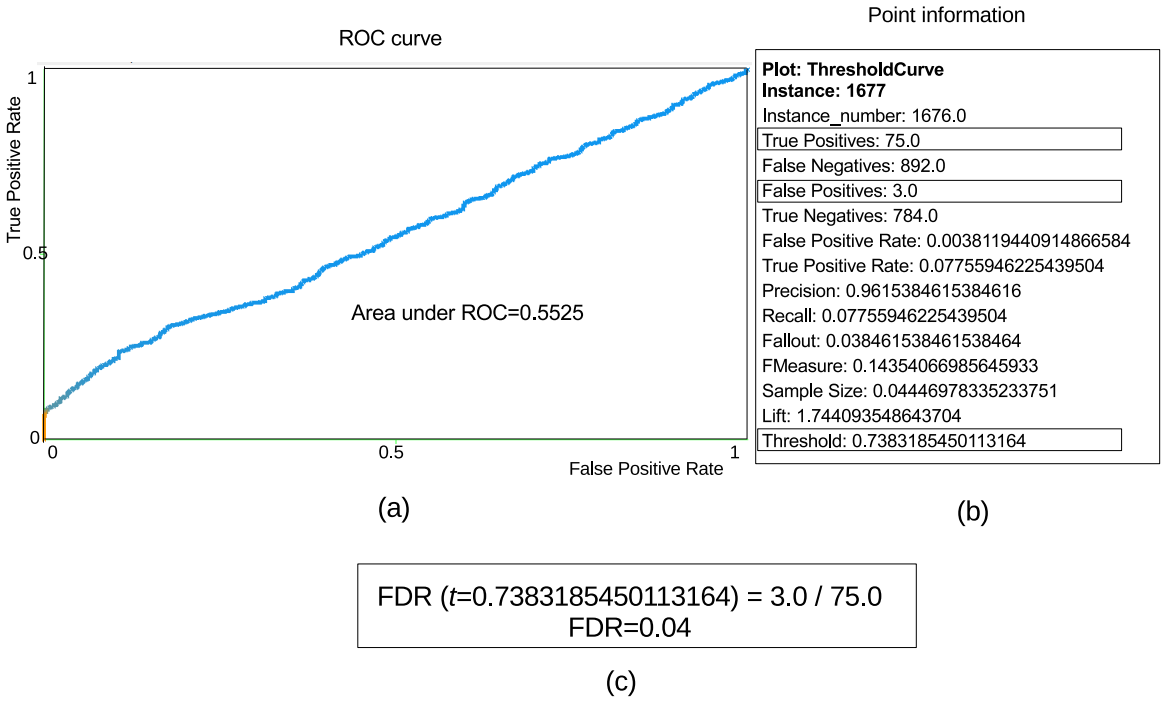


Figura 3.5. (a) Modelo da curva ROC obtida do conjunto dados S2_NPH_CH2. No eixo X encontra-se a taxa de valores falsos positivos e no eixo Y está a taxa de verdadeiros positivos identificadas pelo modelo. (b) O quadro ao lado do gráfico exibe para um FDR máximo escolhido, qual o *threshold*, a quantidade de falsos positivos e verdadeiros positivos contados. (c) O FDR é dado pelo resultado da quantidade de falsos positivos dividido pela quantidade de verdadeiros positivos.

pela ANN às instâncias no conjunto de dados não refletem o acerto dos PSMs. Estes valores indicam, pelo contrário, se os PSMs são *decoys* ou alvos porque os rótulos de classe são definidos dessa maneira. Considerando-se que a maioria dos alvos (aqueles incorretos) são semelhantes aos *decoys*, a própria distinção de *decoys* e alvos não está muito bem caracterizado por essas probabilidades. A AUC = 0,55 visto na Figura 3.6, mostra claramente esta situação.

O algoritmo TSA pode funcionar de duas maneiras. No primeira configuração o TSA encontra automaticamente uma probabilidade discriminante que otimiza alguma dada medida como a *F-measure, accuracy, precision* e *recall*. Na segunda configuração, é fornecido ao TSA um valor de *threshold* fixo. Em seguida, a TSA força o classificador para prever como positivas todas as ocorrências com probabilidade maior ou igual ao *threshold* dado, ou como negativa, de outra forma. A abordagem MUMAL2 usa a última opção junto com a correção de probabilidade que o TSA fornece. Neste

processo de correção o TSA substitui as probabilidades iguais ao *threshold* fornecido para 0,5 e expande os demais valores de modo que a probabilidade mínima observada seja mapeada para 0, enquanto a máxima seja mapeada para 1.

Na abordagem MUMAL2, após construir um modelo com uma ANN sensível ao custo, com a melhor matriz de custos e análise da curva ROC para realizar estimativas de FDR, a probabilidade discriminante que resulta em FDR=1% é selecionado para ser o *threshold t* que separa os PSMs corretos dos incorretos. A opção por 1% é por ser este o melhor *trade-off* entre a sensibilidade e precisão [Elias et al., 2004] [Balgley et al., 2007]. Em seguida, o valor *threshold t* é determinado pelo TSA para ajustar as probabilidades geradas pela ANN, produzindo novos valores que são mais apropriadas para indicar o PSM exato. Para PSMs com $P = t$, o TSA substitui suas probabilidades para 0,5. Todos os outros valores de probabilidade PSM são modificados, tal como descrito acima. Como resultado, obtém-se uma classificação que separa os PSMs corretos e incorretos (e não alvos de *decoys*) com o valor de probabilidade de ponto médio 0,5 como ponto de separação entre negativos e positivos.

A figura 3.6 ilustra esta situação para o conjunto de dados M123, onde o primeiro quadro é obtido na execução da estratégia MUMAL2 sem o algoritmo TSA. O valor do trhreshold para FDR=1% é selecionado e parametrizado como de precisão 0.5 para o último pipeline. O segundo quadro ilustra o mesmo cenário, porém com a execução da estratégia MUMAL2 completa, incluindo o TSA. Os novos valores de probabilidade são reorganizados e ficam mais coerentes para maior precisão. Isto aumentará as chances de inferência corretas de proteínas para um trabalho futuro.

3.7 Clusterização e a abordagem ULPAN

Visando aumentar a sensibilidade para dados complexos de proteomica resultante da análise de espectrometria de massas, uma das proposta deste trabalho é a utilização de um algoritmo de clusterização precedendo a ANN, abordagem aqui denominada de análise em dados complexos de proteômica utilizando técnicas de aprendizagem não supervisionadas precedendo a ANN (ULPAN - *analysis in complex proteomics data using unsupervised learning techniques preceding the ANN*). A clusterização prevê o agrupamento de dados onde haja objetos semelhantes ou que possuam afinidade de um para outro. Com o mesmo conceito formam-se agrupamentos distinguindo objetos diferentes ou não relacionados com os objetos de outros grupos [Han, 2006]. O objetivo é identificar, ainda que de forma aproximada, os PSMs 1 que são incorretos e que são semelhantes aos PSMs 0, de modo a distribuí-los para a classe 0, juntos aos *decoys*,

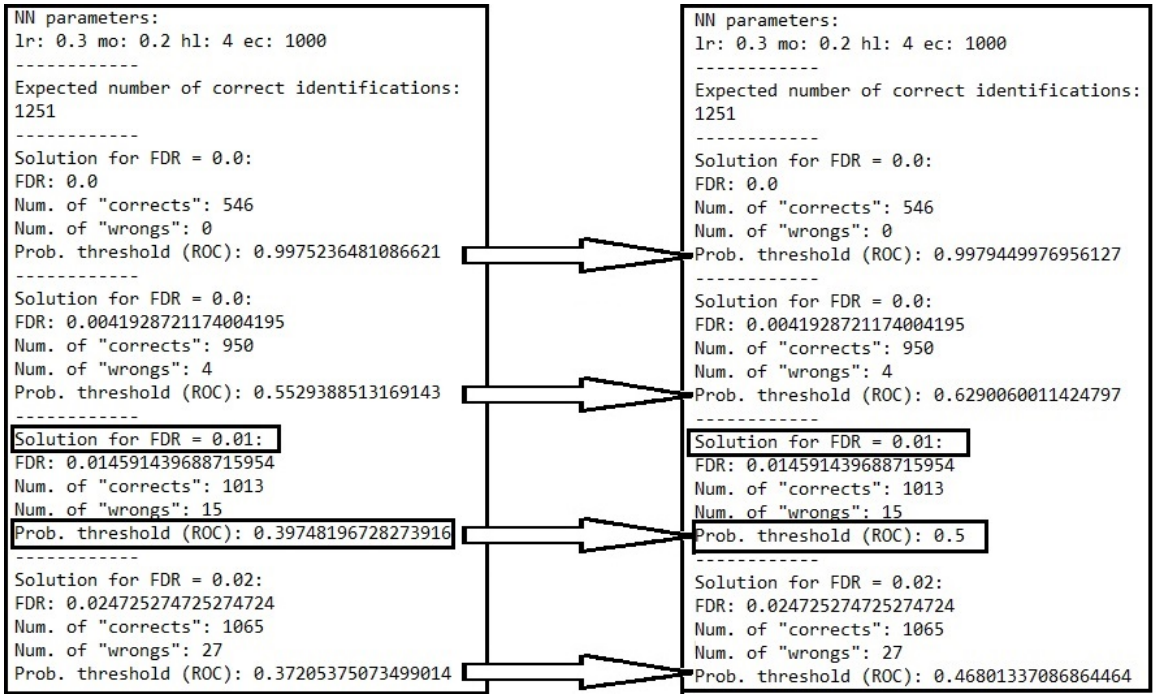


Figura 3.6. Representação dos valores de probabilidade antes e após a execução o algoritmo TSA para o conjunto de dados M123. No primeiro quadro estão os valores obtidos após a execução da estratégia MUMAL2 sem o algoritmo TSA. O valor de *threshold* para o FDR=1% é selecionado para a parametrização manual do TSA. O quadro 2 ilustra os novos valores de probabilidade já reorganizadas tendo o *threshold* selecionado anteriormente como novo ponto médio 0.5.

obtendo-se um conjunto de dados mais acurada para a posterior execução da ANN conforme a abordagem MUMAL.

Dois algoritmos bem conhecidos de clusterização foram testados para realizar experimentos no método aqui proposto: k-means e EM (*expectation maximization*). Isto porque o algoritmo de clusterização baseado em densidade não é adequado para grupos de densidades desiguais, como é o caso, e a clusterização hierárquica tem desempenho inferior no que se refere a tempo de resposta e uso de memória [Han, 2006], conforme descrito na Seção 2.6.

O uso da clusterização na abordagem aqui denominada ULPAN é ilustrado na Figura 3.7. O conjunto de dados original (BO) é dividido em dois conjuntos para o agrupamento: um conjunto *decoy* (BD) somente com as falsas sequências protéicas e um conjunto não-*decoy* (BND) somente com os dados alvos. Em BND é eliminado o atributo *class*, valor referente à classificação da instância como *decoy* ou *no decoy* (classes 0 e 1 respectivamente). Considerando-se que em BND ficam somente os dados *no decoy*, este atributo torna-se dispensável. O segundo passo é usar um

algoritmo de agrupamento em BND objetivando a formação de 2 grupos de dados: o menor contendo as instâncias possivelmente corretas e o grupo com maior quantidade de instâncias com as possivelmente incorretas, já que a maior parte dos PSMs são incorretos. No terceiro passo retorna-se com o atributo class em BND, porém com o valor conforme o agrupamento feito pelo algoritmo de clusterização. As instâncias do grupo onde possivelmente estão os valores corretos recebem o mesmo valor de *class* que as instâncias não-*decoy* de BO. As instâncias do grupo onde possivelmente estão os valores incorretos assumem o mesmo valor de *class* que as instâncias *decoy* de BO. O passo seguinte é a união dos conjuntos BND e BD formando o novo conjunto de treinamento (BT) para a aplicação da ANN. O modelo gerado é aplicado sobre a Base Original (BO), explorando a curva ROC para análise da curva de sensibilidade.

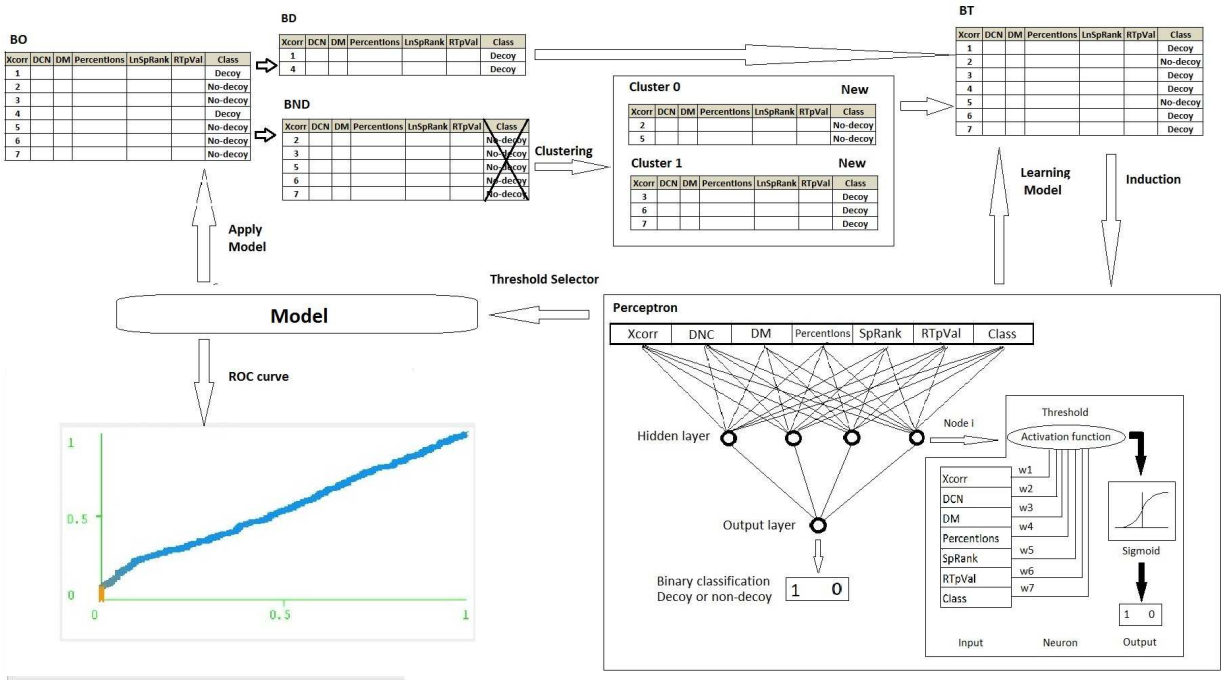


Figura 3.7. Abordagem usando a clusterização antecedendo a ANN (ULPAN), onde BO é o conjunto de dados original, BD é a divisão da BO contendo somente instâncias decoy, BND é a divisão da BO contendo somente instâncias não-decoy e BT é o conjunto de treinamento. Nesta abordagem o BO é dividido em BD e BND. BND passa pela clusterização e as classes são redefinidas conforme o agrupamento das instâncias. Posteriormente BND e BD se juntam novamente formando o BT. A ANN gera o modelo sobre BT, que é aplicado em BO. A curva ROC gera os limites de decisão.

3.8 A abordagem MUMAL2 - Pipeline

Para o desenvolvimento do *framework* MUMAL2 foram usados os classificadores da API do Weka: *CostSensitiveClassifier* para a definição da matriz de custos, *ThresholdSelector* para o algoritmo TSA, o *MultilayerPercetron* para a implementação da ANN e o *ThresholdCurve* para a análise da curva ROC [Hall et al., 2009]. À princípio o MUMAL2 executa dez iterações do *CostSensitiveClassifier* sobre a ANN com as definições e parâmetros propostos como *default* na abordagem MUMAL. Para detalhes do *framework* MUMAL, consultar [Cerqueira et al., 2012]. Em cada iteração o valor da matriz de custo associado a previsão de classificação errada para a classe 1 varia entre 1 e 10. Como a classe 0 é de instâncias *decoy*, que são sabidamente incorretos, deseja-se um modelo gerado pelo classificador que os acerte, conforme descrito na Seção 3.4. Para obtenção da maior quantidade de valores verdadeiros positivos são executadas as 10 iterações propostas e a matriz de custo que apresentar a maior quantidade de valores verdadeiros positivos, na média entre FDR=0 e 0,05, é escolhida, conforme descrito na Seção 3.4. Uma última iteração é feita para selecionar o valor do *trhreshold* para o FDR=1% e parametrizá-lo no TSA com o valor 0,5, objetivando obter limites de probabilidade mais coerentes. Este procedimento completo é ilustrado na Figura 3.8.

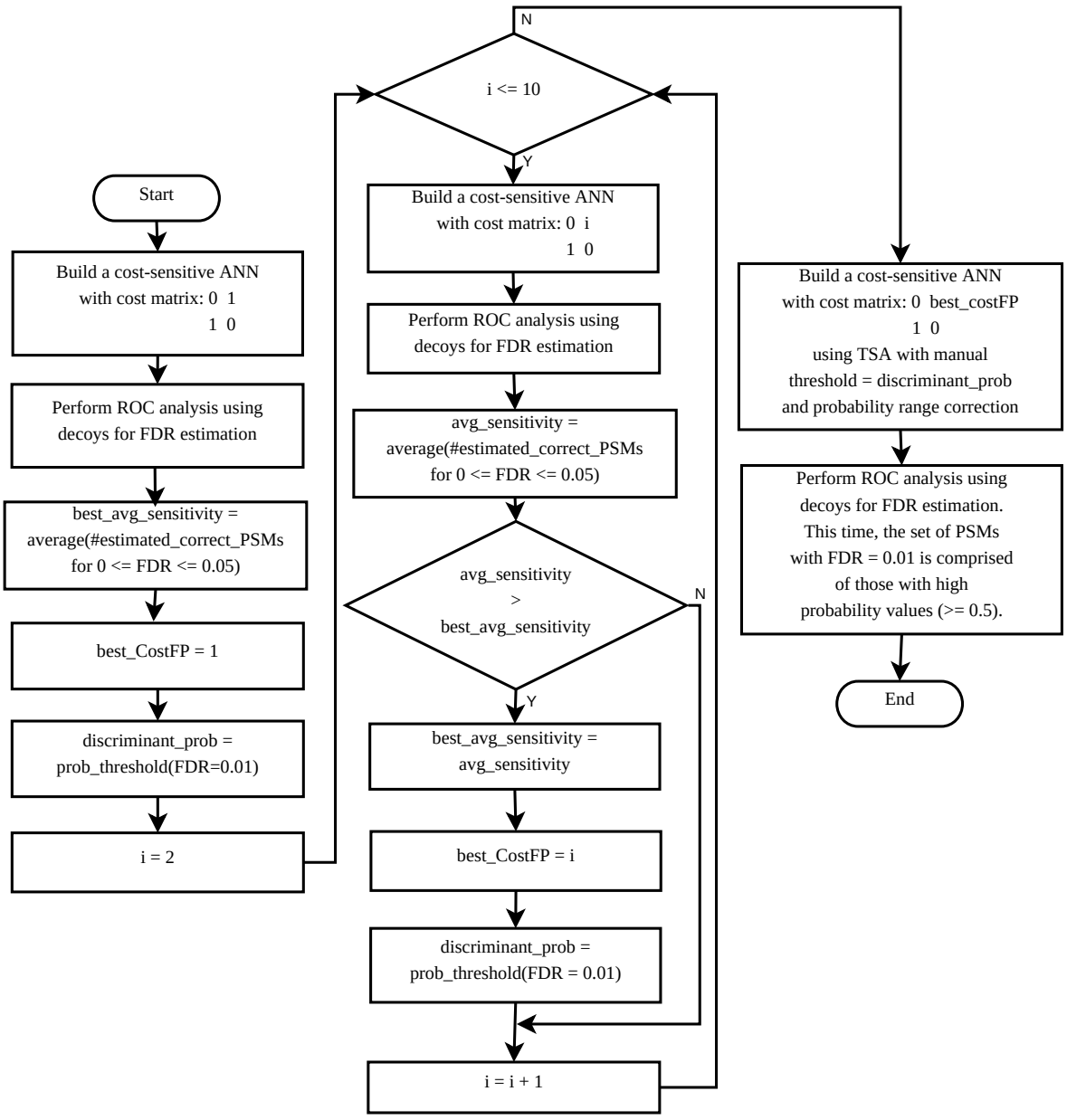


Figura 3.8. Representação das fases da abordagem MUMAL2.

Capítulo 4

Resultados e discussão

Como demonstrado no trabalho do MUMAL, as melhorias nos níveis de identificação de peptídeo levam também a melhorias no nível de identificação de proteína, possivelmente levando a uma maior cobertura do proteoma, ou seja, identificação de mais proteínas [Cerqueira et al., 2012]. Assim como no MUMAL, as comparações no presente trabalho foram feitas sobre o nível de peptídeos, assumindo-se que isto automaticamente se refletirá em melhoria no nível de proteínas [Cerqueira et al., 2012].

Os 11 conjuntos de dados usados neste trabalho, descritos na Seção 3.1, foram submetidos às três abordagens visando a comparação da sensibilidade, ou seja a quantidade de verdadeiros positivos identificados: MUMAL [Cerqueira et al., 2012], ULPAN descrito em 3.7 e MUMAL2 descrito em 3.8. As comparações foram feitas usando a interface API do Weka v3.7.8. O MLP é utilizado nas duas abordagens deste trabalho e são usados os parâmetro *default* definidos no MUMAL, com o uso de 4 nós na camada oculta, *learning rate* 0.3 e *epochs* 1000. Maiores detalhes dos parâmetros default do MUMAL são encontrados em [Cerqueira et al., 2012]. No MUMAL2 e no ULPAN, assim como no MUMAL, os parâmetros podem ser modificados objetivando maior sensibilidade.

Os experimentos foram realizados em uma máquina Windows 8.2 equipado com Intel R Celeron R CPU N2830 2.16 GHz, 2 núcleos e 4 GB de RAM. Isto demonstra que as abordagens fornecem uma resposta rápida mesmo em computadores pessoais. Uma execução do ULPAN leva em média 30 s. Uma iteração do MUMAL2, ou seja, uma execução da matriz de custos mais a ANN, leva 20 s em média. Com onze execuções para produzir o modelo final são necessários 220 s em geral. Esse tempo não é significativamente diferente do tempo de execução do MUMAL e do ULPAN, pois ambos têm também de executar um certo número de iterações para produzir os melhores resultados. Diante disto as análises dos experimentos concentraram-se na

capacidade do MUMAL2 em avaliar PSMs.

4.1 Comparação entre as abordagens MUMAL e ULPAN

Variações dos algoritmos de clusterização (EM e *Kmeans*) precederam o algoritmo MLP e foram testados nos mesmos conjuntos de dados usadas na abordagem MUMAL. Para cada combinação de algoritmos aplicada a cada conjunto de dados foram comparados em sensibilidade, ou seja, valores preditos corretos, para os limites de FDR de 0%, 1%, 2%, 3%, 4% e 5%. Os resultados estão apresentados na Figura 4.1. O MUMAL apresentou resultados superiores à abordagem ULPAN, independente do algoritmo de clusterização usado, para 4 conjuntos de dados. A abordagem ULPAN apresentou sensibilidade melhor em 7 conjuntos de dados, porém em 3 destes conjuntos os valores foram muito próximos ao do MUMAL. A abordagem ULPAN apresentou melhores resultados com o uso do algoritmo *Kmeans*. Pelos experimentos realizados, a combinação de algoritmos de agrupamento, principalmente o *Kmeans*, precedendo o uso da ANN e da análise da curva ROC pode atingir maiores números de verdadeiros positivos dentro dos limites de decisão definidos no MUMAL. Porém os resultados não demonstram contundentemente a eficiência da estratégia ULPAN sobre o MUMAL para a sensibilidade, conduzindo a necessidade de novos experimentos com uso de outras técnicas de DM diferentes da clusterização.

4.2 Comparação entre as abordagens MUMAL e MUMAL2

4.2.1 Medição do poder preditivo do MUMAL2

Um aspecto importante a ser observado é a correlação entre o número de corretos preditos pela abordagem proposta e o número de corretos que realmente há na amostra para os vários valores de corte. Para esta análise o conjunto contendo os dados de uma mistura constituída por proteínas previamente identificadas (M123) foi submetida à abordagem MUMAL2 para valores de corte de FDR de 0% até 20%. O gráfico da Figura 4.2 apresenta no eixo x o número de identificações erradas contadas na mistura (proteínas identificadas previamente) e no eixo y o número de identificações erradas que a abordagem MUMAL2 predisse. Usando-se o Microsoft Excel 2007 foi estabelecida a regressão linear para adequar uma linha nos pontos e verificar se há uma

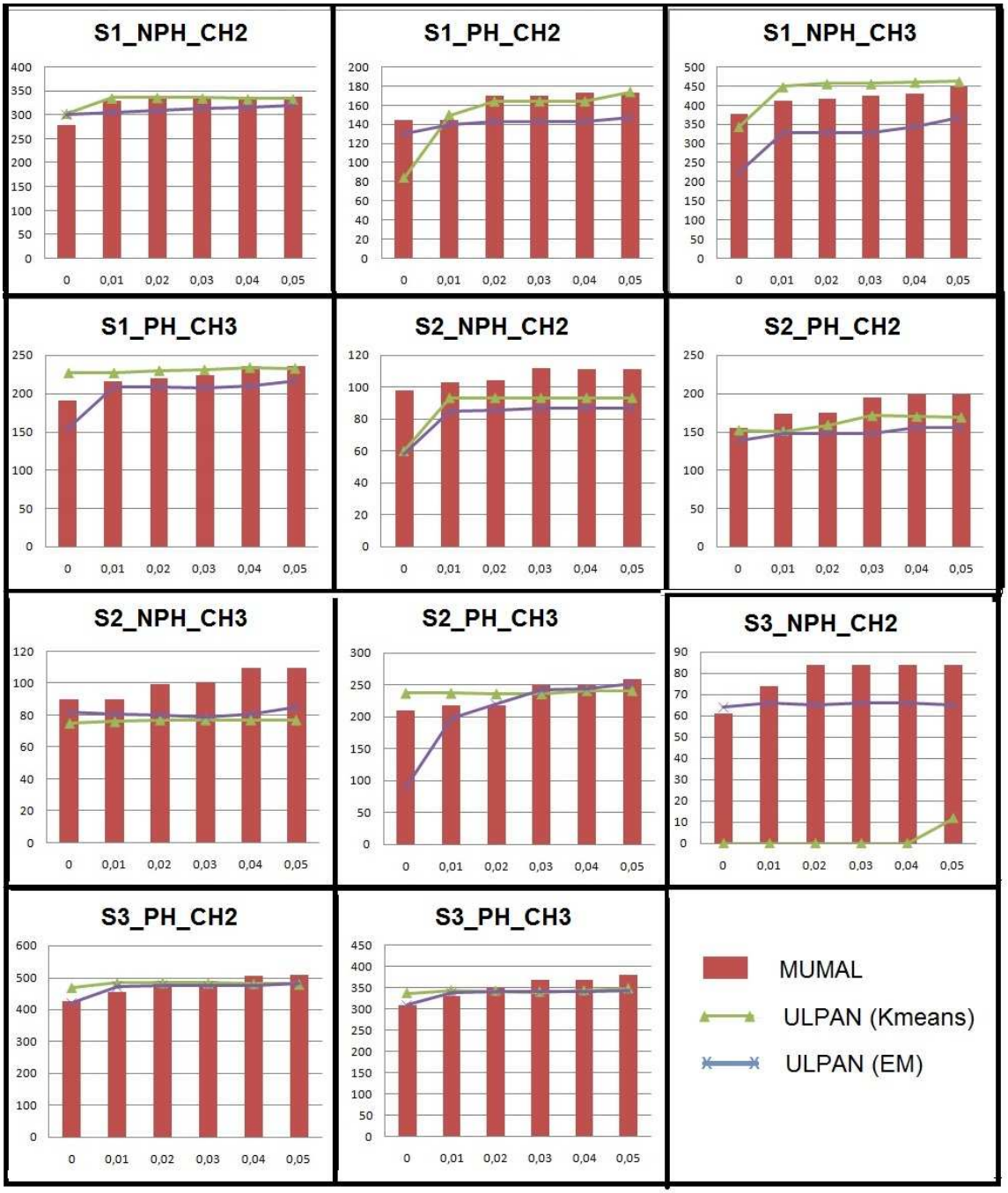


Figura 4.1. Valores de FDR (x) por quantidade de PSMs corretos (y). Comparação entre as abordagem ULPAN, usando o algoritmo *Kmeans* ou EM, e MUMAL para identificação de PSMs corretos para valores de FDR entre 0% e 5% nas 11 conjunto de dados.

forte correlação entre a quantidade identificada predita e a quantidade real. Observa-se boa concordância entre os resultados previstos e observados, com R^2 superior a 0.99, ratificando a precisão na estimativa FDR pelo uso da abordagem TDDB.

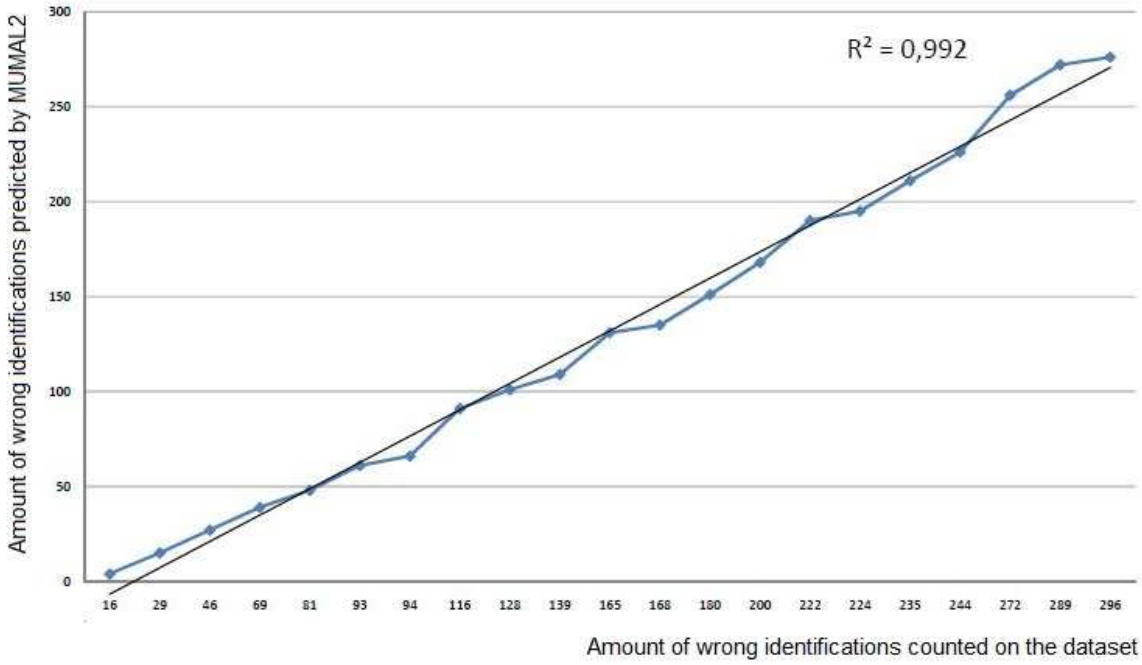


Figura 4.2. Correlação linear para FDR de 0 até 20% (21 valores) entre o número de identificações erradas contadas (eixo X) e o número de identificações erradas que a abordagem da matriz de custos predisse (eixo Y), extraído da mistura M123 com proteínas previamente identificadas.

Quanto ao poder preditivo do método, analisa-se se a reclassificação de PSMs errados da classe-1 para a classe-0 foi satisfatória, ou seja, se o estabelecimento de um limite de decisão adequado entre corretos e incorretos pôde ser realizado satisfatoriamente. A Figura 4.3 contém valores de $\triangle$Cn por Xcorr para a conjunto de de dados M123 antes (Figura 4.3a) e depois (Figura 4.3b) da execução do MUMAL2. Os PSMs da Classe-0 são representadas em azul, enquanto os PSMs classe-1 são mostrados em vermelho. Uma nuvem densa de pontos na Figura 4.3a pode ser vista composta por aproximadamente de 50% *decoys* e 50% não-*decoys*. De acordo com o princípio da abordagem TDDB, esta nuvem densa representa o conjunto de PSMs errados. Portanto, o interesse é pela parte que é composta majoritariamente de pontos vermelhos, pois são os prováveis corretos. Como já mencionado na Seção 3.4, a utilização de uma matriz de custos para a construção de um modelo preciso para a classe 0 é uma tentativa de manter essas instâncias como tal, uma vez que PSMs dessa classe são obrigatoriamente errados, e ao mesmo tempo reclassificar corretamente as erradas da classe 1 para a classe 0. A Figura 4.3b mostra que a fronteira de decisão produzida pelo MUMAL2 parece proporcionar uma boa separação da mistura de *decoys* e não-*decoys* da parte homogênea composta por pontos vermelhos. A matriz de confusão na Tabela 4.1 torna

Tabela 4.1. Matriz de confusão para o conjunto de dados M123 com a aplicação da abordagem MUMAL2 mostrando o enorme número de instâncias de classe 1 que foram classificados como classe 0.

		Predited Class	
		0	1
Given Class	0	4451	12
	1	4466	1047

evidente a enorme quantidade de PSMs alvo que foram classificados como classe 0, o que era esperado porque a maioria destes são PSMs são errados. O modelo construído como resultado da reclassificação das instâncias mostra a grande maioria dos casos na nuvem densa transformado em azul.

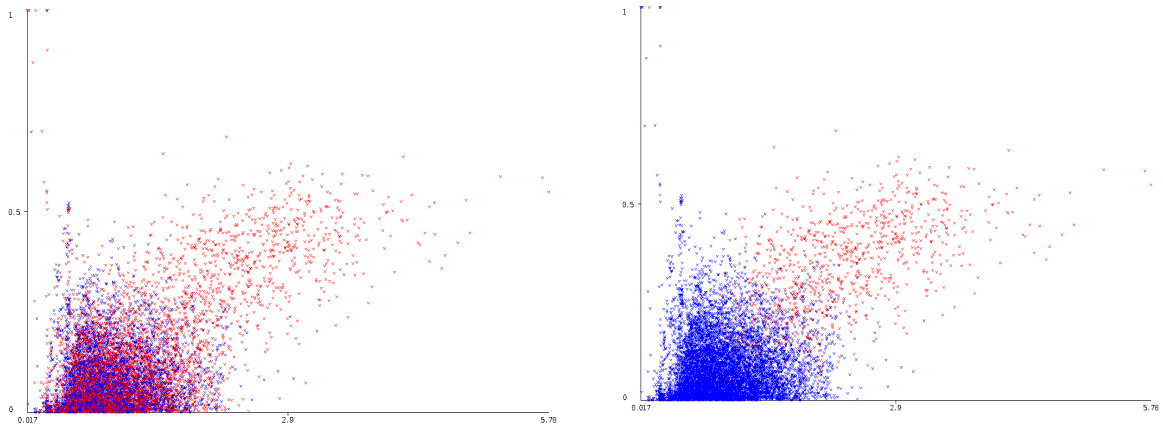


Figura 4.3. É exibido um conjunto de valores de △Cn por Xcorr para o conjunto de dados M123 em (a) e (b), antes e depois de aplicar MUMAL2, respectivamente. As instâncias classificadas como classe-0 casos são mostrados em azul, enquanto as classe-1 são mostradas em vermelho.

Embora a Figura 4.3b indique que a região de interesse (a parte na Figura 4.3a composta principalmente de pontos vermelhos) poderia ser identificada, é possível fornecer medições mais precisas da qualidade da solução apresentada pelo MUMAL2, porque as proteínas do conjunto de dados M123 são conhecidos. A matriz de confusão na Tabela 4.1 mostra que 12 instâncias *decoys* foram erroneamente classificados como classe 1. Provavelmente, algumas corretas da classe 1 também foram incorretamente remarcadas para classe 0. Assim, para proporcionar mais uma avaliação da estratégia MUMAL2, as 8917 instâncias previstas como classe 0 e as 1059 previstas como classe 1 tiveram suas sequências peptídicas inspecionadas para verificar se elas vieram ou não do conjunto de proteínas esperados. Como resultado, conforme ilustra a Tabela

4.2, construiu-se uma matriz de confusão mais útil, considerando positivas ou negativas as instâncias cujas sequências peptídicas vieram da lista de proteínas conhecidas ou das proteínas aleatórias, respectivamente. Como consequência, pode-se usar métricas clássicas que expressam o poder preditivo de um modelo de classificação, como mostra a Tabela 4.2b. Observa-se que a classificação do MUMAL2 é altamente precisa para todas os valores medidos. Os valores de exatidão (*accuracy*), sensibilidade (*sensitivity*), especificidade (*specificity*) e precisão (*precision*) são calculados conforme [Fawcett, 2006] [Powers, 2011]:

- *accuracy*= $ACC = (TP + TN)/(P + N)$;
- *sensitivity* ou *True Positive Rate*= $TPR = TP/P$;
- *specificity* ou *True Negative Rate*= $TNR = TN/N$;
- *precision* ou *Positive Predictive Value*= $PPV = TP/(TP + FP)$;

Onde TP são os verdadeiros positivos (*True Positive*), TN são os verdadeiros negativos (*True Negative*), FP são os falsos positivos (*False Positive*), FN são os falsos negativos (*False Negative*); P=TP+FN e N=FP+TN conforme [Fawcett, 2006] [Powers, 2011].

Conforme demonstrado na Tabela 4.2a, apenas 25 casos (12 negativos e 13 positivos) foram classificados erroneamente, levando a elevados valores de sensibilidade, especificidade e precisão. No entanto, é importante salientar que as instâncias são conhecidas como PSMs 0 (*decoys*) estão erradas. Portanto, os 12 *decoys* mal classificadas mostradas na Tabela 4.1 não são relevantes, ou seja, só os casos não-*decoys* são mais importantes após a classificação. É evidente, assim, que a aplicação aqui de um método de classificação supervisionado não utiliza os passos clássicos que sao: construir o modelo de aprendizagem utilizando um conjunto de treinamento, e aplicar o modelo resultante para casos desconhecidos. Neste caso, a ANN é treinada e aplicada aos mesmos dados. As instâncias não-*decoys* são as de interesse. O objetivo final é separar não-*decoys* errados de corretos. Para este fim, usa-se um custo mais elevado para o FN visando forçar o modelo a aprender a classificar corretamente *decoys* cujos rótulos estão corretos. É por isso que as instâncias *decoys* ajudam na seleção das não-*decoys* erradas. Em seguida, aplica-se o modelo final sobre os mesmos dados para rotular novamente as instâncias não-*decoys* erradas, ou seja, aquelas com características semelhantes às *decoys*, a classe 0. Ou seja, verifica-se se as instâncias corretas e incorretas estão sendo identificados.

É importante também analisar se as probabilidades produzidas pelo MUMAL2 são coerentes. Esta é uma questão relevante se a intenção é usar um método para a

Tabela 4.2. Avaliação do MUMAL2 de acordo com as proteínas conhecidas no conjunto de dados M123. Em (a), é exibida uma matriz de confusão onde as instâncias 1 e 0 não representam não-*decoy* e *decoy*. Neste caso, uma instância é considerada 1 se a sua sequência de péptido veio a partir da lista de proteínas conhecidas. Caso contrário, a instância é considerada 0. Em (b), medidas estatísticas conhecidas são apresentados para avaliar o poder preditivo de MUMAL2.

		Predicted Class	
		0	1
Actual Class	0	8904	12
	1	13	1047

(a)

Statistical measures	
Accuracy	0.9975
Sensitivity	0.9877
Specificity	0.9987
Precision	0.9887

(b)

inferência de proteínas que leve em consideração as probabilidades de PSM, como é o caso do ProteinProphet. Como mostrado na Figura 3.5, o AUC e outras medidas são baixas porque as instâncias não-*decoys* erradas que são corretamente remarcados para classe 0 são contados como erro de classificação. Na verdade, a aplicação do MUMAL2 no conjunto de dados M123 também proporcionou uma AUC próxima de 0,60. No entanto, como as proteínas no conjunto de dados M123 são conhecidas, pode-se produzir uma curva ROC real para avaliar as probabilidades geradas pelo MUMAL2. Figura 4.4 mostra a curva ROC construída para o conjunto de dados M123 contando como TPRs as instâncias cujos peptídeos vieram a partir da lista de proteínas conhecidas, e contando como FPRs, caso contrário. Como pode ser visto, a AUC é muito satisfatória (cerca de 0,93), mostrando que os valores de probabilidade são apropriados, e corroborando o alto poder preditivo do MUMAL2. Para a obtenção da curva ROC os PSMs e as respectivas probabilidades obtidas com a abordagem foram tabulados usando-se o Microsoft Excel 2007. As probabilidades foram ordenadas e os PSMs avaliados como corretos ou não, já que o conjunto M123 possui as proteínas previamente identificadas. Posteriormente cada probabilidade serviu como valor discriminante para a determinação do FPR e do TPR. Para o cálculo do TPR baseou-se na fórmula TPR=(TP/(TP+FN)) e o FPR baseou-se na fórmula FPR=(FP/(FP+TN)) [Witten & Frank, 2005].

O MUMAL2 também foi aplicado nos outros conjuntos de dados, onde as proteínas não são conhecidas a priori, ou seja os onze conjuntos de dados conforme descrito em 3.1. São utilizados os valores de $\triangle Cn$ e $Xcorr$, como demonstrado anteriormente para o conjunto M123, para realizar uma inspeção visual. A Figura 4.5 mostra a análise para conjunto de dados S1_PH_CH2. Pode-se ver na matriz de confusão (Figura 4.5b) que um número significativo de instâncias de classe 1 foram remarcado para classe 0, como esperado. Ainda da Figura 4.5b, as cores dos pontos indicam que a classificação

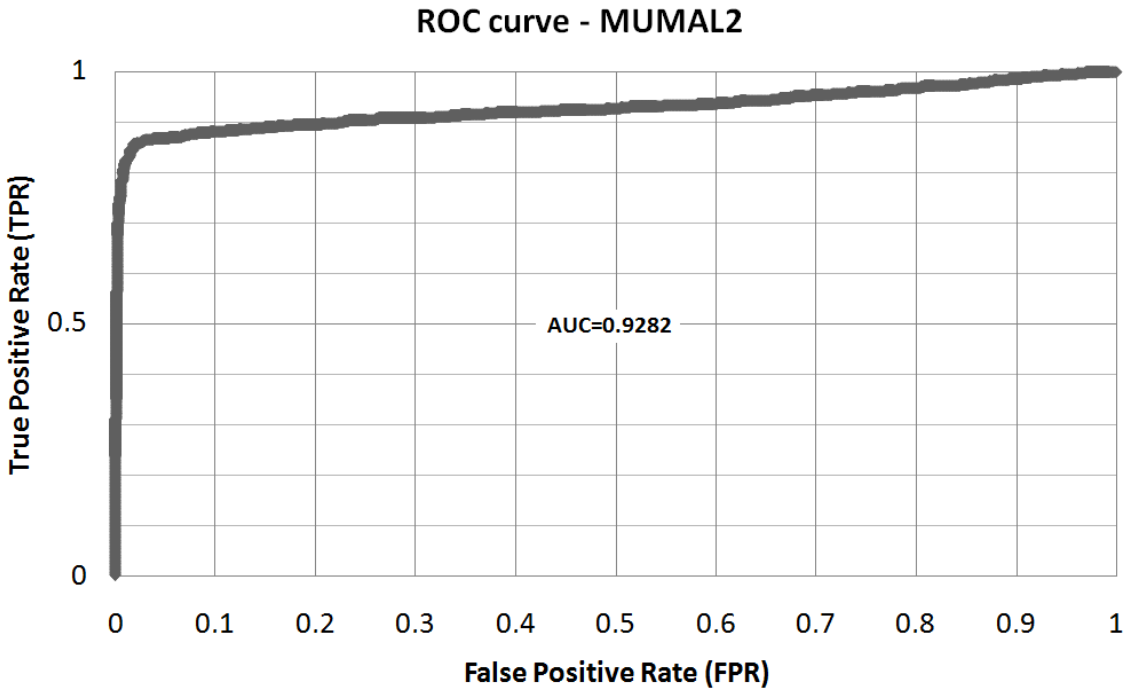


Figura 4.4. Curva ROC obtida na abordagem MUMAL2 para o conjunto de dados M123, com o FPR no eixo X e o TPR no eixo Y.

parece ser adequada, pois os pontos azuis correspondem aos da Figura 4.5a composta de uma mistura de valores *decoy* e não-*decoy*, ou seja, a parte em que os não-*decoys* são provavelmente errados, de acordo com o princípio do TDDB. Realizando a mesma análise para os outros conjuntos de dados levou-se a resultados muito semelhantes (não mostrados). Ou seja, o MUMAL2 promoveu uma expressiva migração de não-*decoys* para a classe 0, ou seja, a classe 0 corresponde à mistura de instâncias *decoys* e instâncias não-*decoys* erradas.

4.2.2 Comparação do MUMAL2 com outros métodos previamente propostos

Os experimentos seguintes demonstram a sensibilidade superior do MUMAL2 em relação a métodos importantes para a avaliação do PSM: MUMAL, MUDE, PeptideProphet, e as análises baseadas na estratégia TDDB, onde os limites de duas pontuações, muitas vezes $\triangle Cn$ e $Xcorr$, conduzem a um FDR desejado. Para comparações

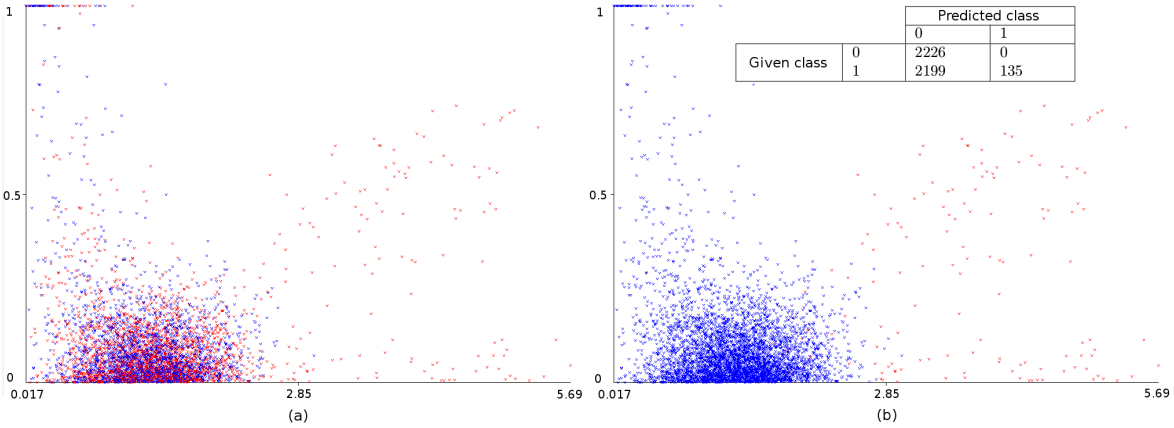


Figura 4.5. É exibido um conjunto de valores de $\triangle Cn$ por *Xcorr* para o conjunto de dados S1_PH_CH2 em (a) e (b), antes e depois de aplicar MUMAL2, respectivamente. As instâncias classificadas como classe-0 são mostrados em azul, enquanto as instâncias classe-1 são mostradas em vermelho. Em (b) ainda apresenta no topo a matriz de confusão que demonstra o número significativo de instâncias classe 1 que foram classificadas como classe 0.

com *phosphodata*, incluiu-se uma análise bivariada com $\triangle M$ e Xcorr, seguindo as recomendações de [Jiang et al., 2008] [Villén et al., 2007]. De acordo com estes autores, a pontuação $\triangle Cn$ é muitas vezes suprimida quando fosfopeptídeos têm mais de um potencial local de fosforilação. Portanto, o $\triangle$M deve ser usado em vez disso. Para uma descrição detalhada de como o os métodos utilizados para a comparação foram executados para produzir os resultados mostrados a seguir, consultar as obras [Cerqueira et al., 2010] [Cerqueira et al., 2012].

As Figuras 4.6 e 4.7 mostram as curvas do número de PSMs identificados pela estimada FDR para todos os métodos mencionados, aplicados aos onze conjuntos de dados descritos, cujas proteínas são desconhecidas. É possível construir tais curvas pois todas essas abordagens fornecem uma maneira eficaz para estimar o valor do FDR para um determinado conjunto de PSMs selecionados. As curvas mostram valores FDR variando de 0 a 5%. Pode ser observado que em todos os casos o MUMAL2 é superior ao MUDE, ao PeptideProphet e às análises bivariadas.

Em relação ao MUMAL, o MUMAL2 tem uma sensibilidade igual ou superior. Isso é esperado porque MUMAL2 executa 10 iterações com o custo de falso positivo (cFP) variando de 1 a 10. A execução com cFP = 1 é equivalente à execução MUMAL. Portanto, o MUMAL2 não pode ser pior, mas pode, eventualmente, apresentar a mesma sensibilidade que MUMAL. No entanto, o número de casos em que o MUMAL2 tem uma sensibilidade maior é mais elevada do que os casos de desempenho igual. O MUMAL2 pode fornecer um ganho em média de 16,5% e 7,2% em relação ao MUMAL

para não-fosfopeptídeos e fosfopeptídeos, respectivamente. Isso significa cerca de 24 e 20 peptídeos a mais em média, respectivamente.

Em particular para um FDR 1%, que é um valor FDR geralmente desejado, o MUMAL2 demonstra a superioridade em todos os casos. Para proteínas não fosforilados, a avaliação dos PSM fornecida pelo MUMAL2 para este FDR levou a uma melhoria na sensibilidade em média de 16,8% em comparação com MUMAL, ou seja, cerca de 23 peptídeos adicionais. Para as proteínas fosforiladas, por sua vez, o aumento foi de aproximadamente 12%, resultando em quase 31 péptidos a mais.

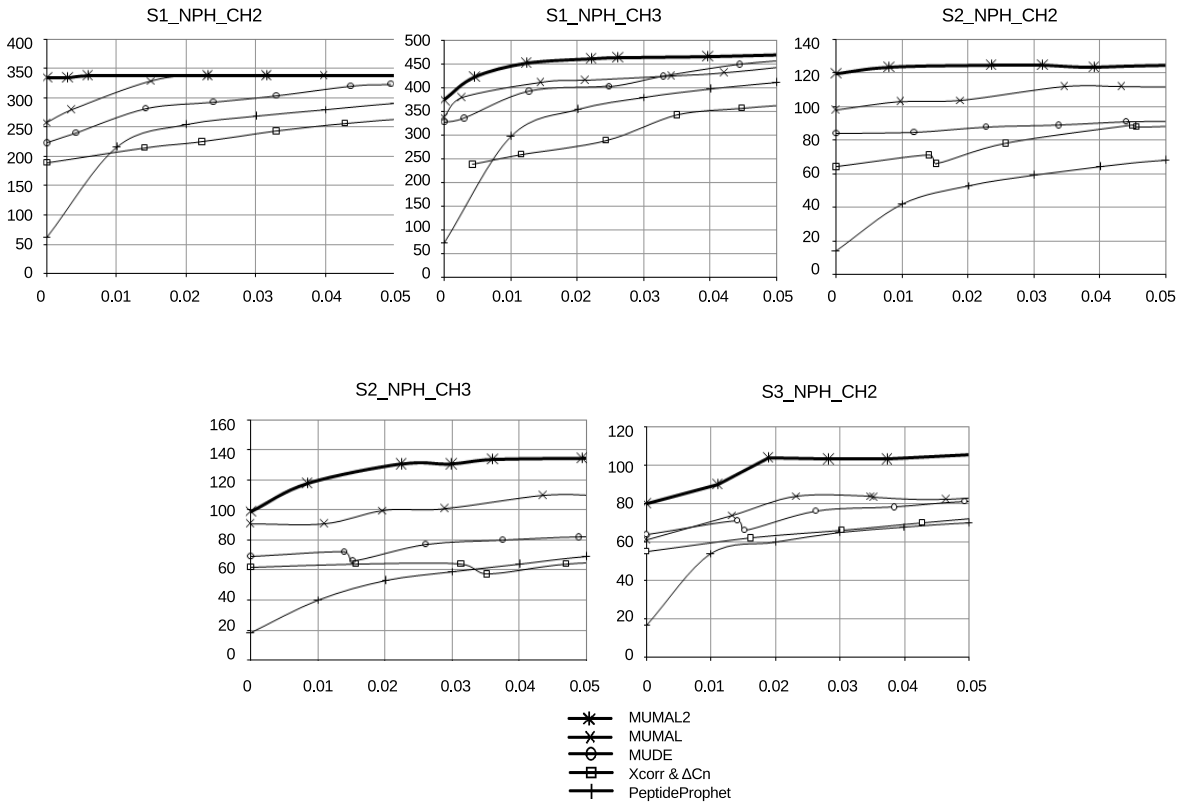


Figura 4.6. FDR (x) por quantidade de PSMs corretos (y) - Comparação entre as abordagem MUMAL2, MUMAL, MUDE e análise bivariada (Xcorr&△Cn) para identificação de PSMs corretos para valores de FDR entre 0% e 5% nos conjuntos de dados - *no-phosphodata*.

As Figuras 4.6 e 4.7 demonstram que o MUMAL apresentou melhor desempenho entre os os métodos que estão sendo comparados com o MUMAL2. Por esta razão, foram realizados experimentos adicionais para comparar o MUMAL com o MUMAL2 por meio de diagramas de Venn visando chamar a atenção para o maior número de identificações de peptídeos exclusivos. Nesta comparação, para ambas abordagens e para cada conjunto de dados, reservou-se o valor de *threshold* para o FDR=0.01 e

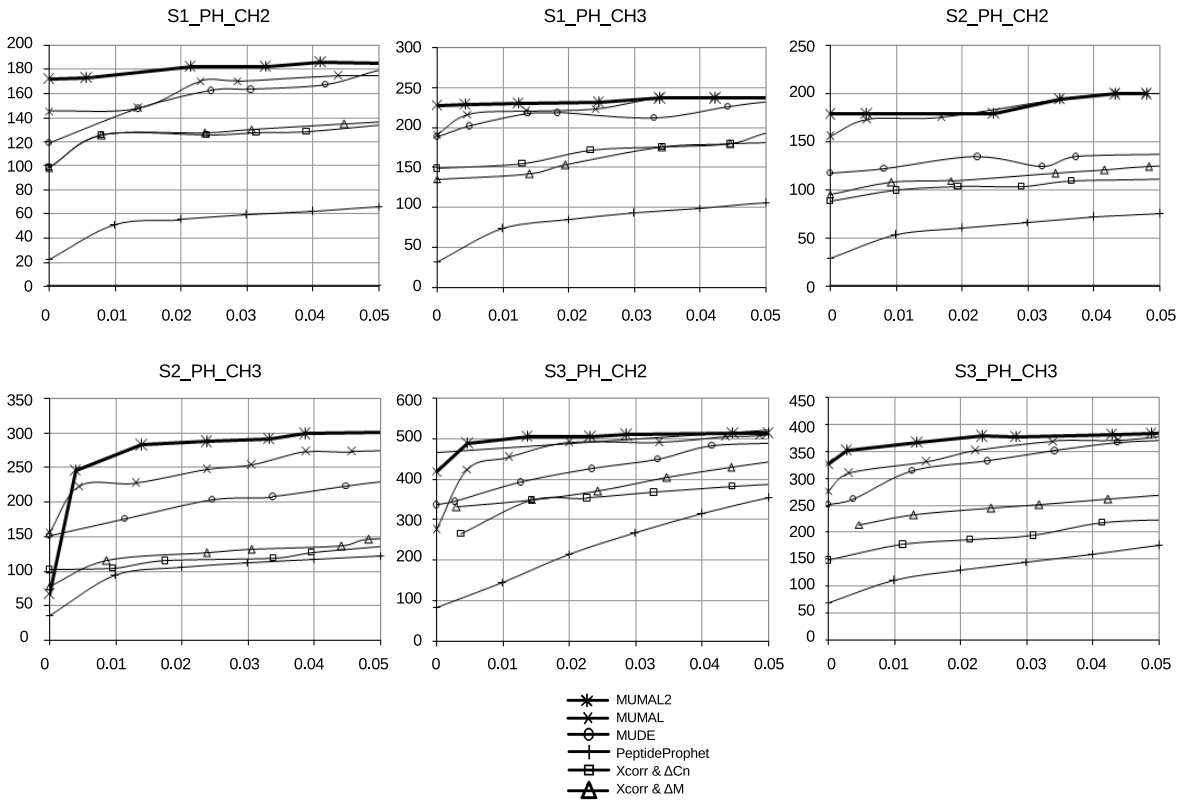


Figura 4.7. FDR (x) por quantidade de PSMs corretos (y) - Comparação entre as abordagem MUMAL2, MUMAL, MUDE e análises bivariadas (Xcorr&△Cn e Xcorr&△M) para identificação de PSMs corretos para valores de FDR entre 0% e 5% nas bases de dados - *phosphodata*.

selecionou-se todas as instâncias com *threshold* superior ao valor reservado. Posteriormente comparou-se os PSMs encontrados, identificando quantos eram iguais e quantos foram identificados de forma exclusiva por cada abordagem. A Figura 4.8 mostra esta contagem para um FDR 1% em ambos métodos. Para cada diagrama, o conjunto da esquerda representa as atribuições recuperadas exclusivamente pela abordagem MUMAL2, o conjunto do centro as recuperadas em ambas as abordagens, enquanto que o conjunto da direita indica a quantidade de identificações encontradas exclusivamente pelo MUMAL. Em todos os casos, uma superioridade expressiva de MUMAL2 pode ser notado. Em média, o número de PSMs exclusivos que MUMAL2 encontrou é quase 4 vezes maior. Este é um resultado importante, pois mais peptídeos podem implicar em mais proteínas identificadas e uma cobertura superior do proteoma.

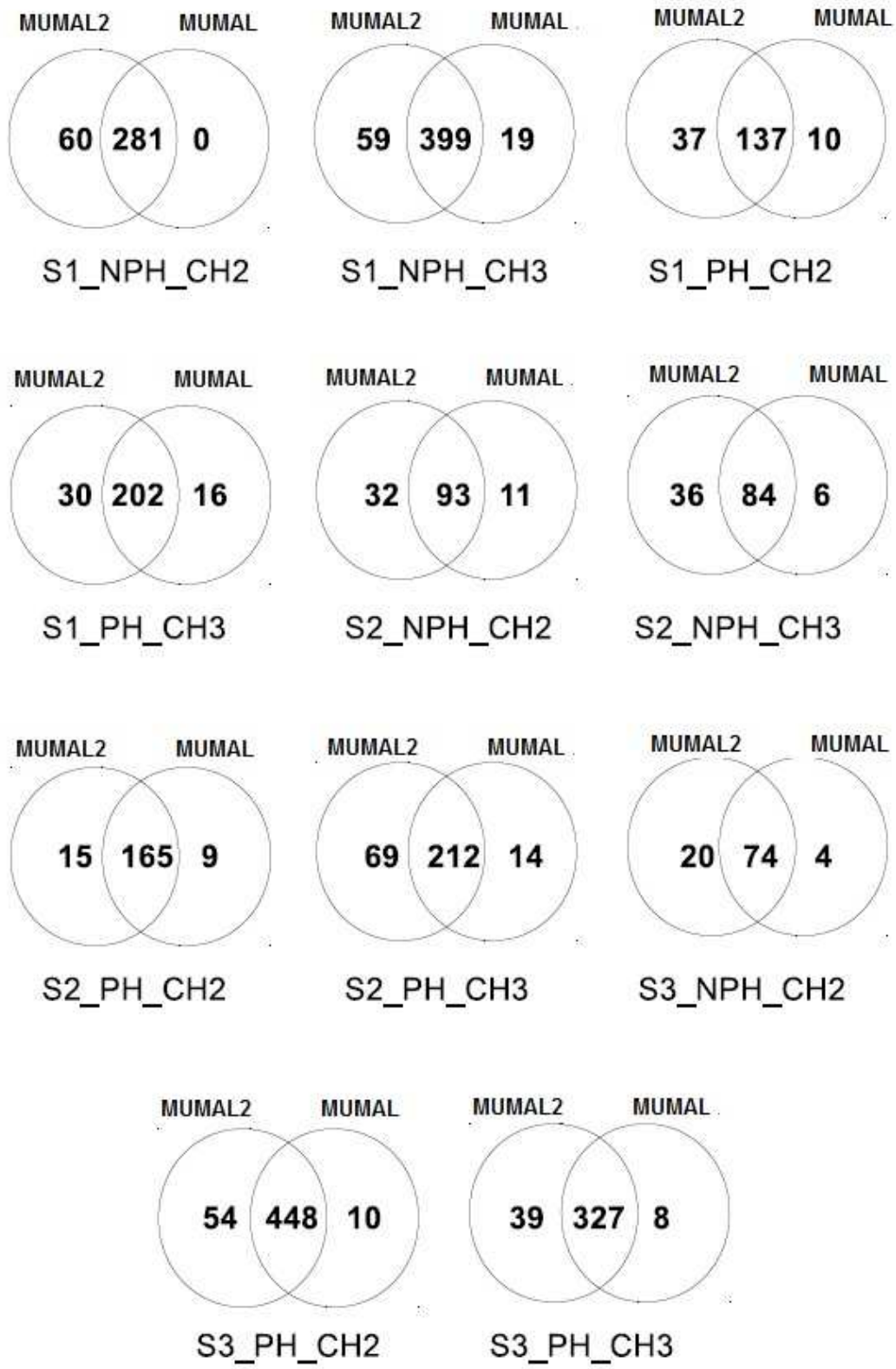


Figura 4.8. Diagramas de Venn para dados identificados para um FDR 1% entre as abordagens MUMAL2 e MUMAL. Para cada diagrama o conjunto de esquerda representa atribuições recuperados pela abordagem deste projeto, enquanto o conjunto da direita indica identificações encontrados pelo MUMAL. Ao centro estão as atribuições recuperadas por ambas abordagens.

Capítulo 5

Conclusões

As estratégias MUDE e MUMAL, usadas como base para este trabalho de pesquisa, já haviam demonstrado promover um aumento significativo da sensibilidade quando comparado a outras abordagens que usam a técnica TDDB na avaliação de PSMs, nas corridas MS/MS. Entretanto, a melhoria na identificação dos verdadeiros positivos pode ser obtida por meio da abordagem MUMAL2 usando uma rede neural sensível ao custo. Embora a hipótese do uso de técnicas não supervisionadas de aprendizagem (agrupamento) precedendo a abordagem MUMAL não demonstrou ganho na sensibilidade, a inserção da técnica da matriz de custos ao MUMAL apresentou melhorias para todos os conjuntos de dados testadas.

A utilização dos algoritmos EM e Kmeans para agrupamento, precedendo a abordagem MUMAL, não conseguiu aumentar a sensibilidade em todas as bases de dados testadas. O propósito básico da abordagem ULPAN foi usar as características ruidosas dos dados, com a maioria das instâncias alvos de valores possivelmente semelhantes aos *decoys*, como fator para uma nova rotulagem de classe e posterior submissão à ANN. Entretanto, não houve eficiência quando comparado à separação simples de *decoy* e não-*decoy* da própria técnica TDDB.

Já o MUMAL2 demonstrou, durante os experimentos desenvolvidos neste trabalho, um aumento da sensibilidade apresentando resultados superiores ao MUMAL para todas as bases de dados testadas. Acrescentando a matriz de custos e o algoritmo TSA ao MUMAL, o MUMAL2 permite um significativo ganho em cobertura de proteoma, permitindo a identificação de maiores quantidades de PSMs verdadeiros positivos presentes em uma amostra, para um determinado FDR, tornando-o uma alternativa mais consistente para a análise de dados complexos de proteoma.

O MUMAL2 demonstrou também impactar positivamente a inferência correta de proteínas em futuros trabalhos de pesquisas em proteômica *shotgun*. Proporcio-

nando melhor distribuição e reorganização mais coerente das probabilidades para a classificação de um PSM, tornam maior a precisão do poder preditivo do método, conforme demonstrado nos experimentos. Esta é uma questão fundamental para melhorar a inferência de proteínas quando a abordagem aplicada depende de tais valores de probabilidade, como no caso do ProteinProphet.

O MUMAL2 tem grande potencial para fornecer importantes melhorias na identificação de proteínas, o que terá um impacto em estudos futuros em que busca-se uma compreensão mais ampla das atividades celulares. Felizmente, pesquisas futuras na descoberta de medicamentos, doenças, e muitos outros estudos em biologia serão positivamente afetados por esta nova estratégia computacional para a identificação de peptídeos/proteínas.

Como trabalhos futuros, propõem-se a utilização da estratégia MUMAL2 para o desenvolvimento de ferramenta para a inferência das proteínas presentes em bases de dados de proteômica que usa a estratégia TDDB, bem como a verificação da viabilidade na aplicação deste *pipeline* para outras bases de dados complexas de bioinformática.

Referências Bibliográficas

Balgley, B. M.; Laudeman, T.; Yang, L.; Song, T. & Lee, C. S. (2007). Comparative evaluation of tandem MS search algorithms using a target-decoy search strategy. *Mol. Cell. Proteomics*, 6:1599–1608.

Cantú, M. D.; Carrilho, E.; Wulff, N. A. & Palma, M. S. (2008). Peptide sequencing using mass spectrometry: a practical guide. *Química Nova*, 31(3):669--675.

Cerqueira, F. R.; Ferreira, R. S.; Oliveira, A. P.; Gomes, A. P.; Ramos, H. J.; Graber, A. & Baumgartner, C. (2012). Mumal: Multivariate analysis in shotgun proteomics using machine learning techniques. *BMC genomics*, 13(Suppl 5):S4.

Cerqueira, F. R.; Graber, A.; Schwikowski, B. & Baumgartner, C. (2010). Mude: a new approach for optimizing sensitivity in the target-decoy search strategy for large-scale peptide/protein identification. *Journal of proteome research*, 9(5):2265--2277.

Cerqueira, F. R.; Morandell, S.; Ascher, S.; Mechtler, K.; Huber, L. A.; Pfeifer, B.; Graber, A.; Tilg, B. & Baumgartner, C. (2009). Improving phosphopeptide/protein identification using a new data mining framework for ms/ms spectra preprocessing. *J Proteomics Bioinform*, 2:150--164.

Cheung, M. R. (2014). Receiver operating characteristic curve analysis of seer medulloblastoma and primitive neuroectodermal tumor (pnet) outcome data: Identification and optimization of predictive models. *Asian Pacific journal of cancer prevention: APJCP*, 15(16):6781.

Elias, J. E.; Gibbons, F. D.; King, O. D.; Roth, F. P. & Gygi, S. P. (2004). Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nat. Biotechnol.*, 22:214–219.

Elias, J. E. & Gygi, S. P. (2007). Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods*, 4(3):207–214.

Elkan, C. (2001). The foundations of cost-sensitive learning. Em *International joint conference on artificial intelligence*, volume 17, pp. 973--978. Citeseer.

Eng, J. K.; McCormack, A. L. & Yates, J. R. (1994). An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.*, 5:976–989.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861--874.

Granholm, V.; Noble, W. S. & Käll, L. (2012). A cross-validation scheme for machine learning algorithms in shotgun proteomics. *BMC bioinformatics*, 13(Suppl 16):S3.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P. & Witten, I. H. (2009). The WEKA data mining software: An update. Em *SIGKDD Explorations*, volume 11.

Han, J., K. M. (2006). *Data Mining: Concepts and Techniques, 2006 Elsevier*. Elsevier.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundatio, 1998 Bookman*. Prentice Hall.

He, K.; Fu, Y.; Zeng, W.-F.; Luo, L.; Chi, H.; Liu, C.; Qing, L.-Y.; Sun, R.-X. & He, S.-M. (2015). A theoretical foundation of the target-decoy search strategy for false discovery rate control in proteomics. *arXiv preprint arXiv:1501.00537*.

Imanishi, S. Y.; Kochin, V.; Ferraris, S. E.; Thonel, A.; Pallari, H. M.; Corthals, G. L. & Eriksson, J. E. (2007). Reference-facilitated phosphoproteomics: Fast and reliable phosphopeptide validation by μLC-ESI-Q-TOF MS/MS. *Mol. Cell. Proteomics*, 6:1380–1391.

Ivanov, M. V.; Levitsky, L. I.; Lobas, A. A.; Panic, T.; Laskay, U. A.; Mitulovic, G.; Schmid, R.; Pridatchenko, M. L.; Tsybin, Y. O. & Gorshkov, M. V. (2014). Empirical multidimensional space for scoring peptide spectrum matches in shotgun proteomics. *Journal of proteome research*, 13(4):1911--1920.

Jiang, X.; Han, G.; Feng, S.; Jiang, X.; Ye, M.; Yao, X. & Zou, H. (2008). Automatic validation of phosphopeptide identifications by the MS2/MS3 target-decoy search strategy. *J. Proteome Res.*, 7:1640–1649.

Kapp, E. A.; Schütz, F.; Connolly, L. M.; Chakel, J. A.; Meza, J. E.; Miller, C. A.; Fenyo, D.; Eng, J. K.; Adkins, J. N.; Omenn, G. S. & Simpson, R. J. (2005).

An evaluation, comparison, and accurate benchmarking of several publicy available MS/MS search algorithms: Sensitivity and specificity analysis. *Proteomics*, 5:3475–3490.

Keller, A.; Nesvizhskii, A. I.; 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(20):5383–5392.

Kim, M.-S.; Pinto, S. M.; Getnet, D.; Nirujogi, R. S.; Manda, S. S.; Chaerkady, R.; Madugundu, A. K.; Kelkar, D. S.; Isserlin, R.; Jain, S. et al. (2014). A draft map of the human proteome. *Nature*, 509(7502):575--581.

Kumar, A.; Rajendran, V.; Sethumadhavan, R.; Shukla, P.; Tiwari, S. & Purohit, R. (2014). Computational snp analysis: current approaches and future prospects. *Cell biochemistry and biophysics*, 68(2):233--239.

Lan, H.; Frank, E. & Hall, M. (2011). Data mining: Practical machine learning tools and techniques.

Lasko, T. A.; Bhagwat, J. G.; Zou, K. H. & Ohno-Machado, L. (2005). The use of receiver operating characteristic curves in biomedical informatics. *Journal of biomedical informatics*, 38(5):404--415.

Leninger, A. L. (2002). *Bioquímica*. São Paulo, 2 edição.

Li, Y. F. & Radivojac, P. (2012). Computational approaches to protein inference in shotgun proteomics. *BMC bioinformatics*, 13(Suppl 16):S4.

Lleo, A.; Zhang, W.; McDonald, W. H.; Seeley, E. H.; Leung, P. S.; Coppel, R. L.; Ansari, A. A.; Adams, D. H.; Afford, S.; Invernizzi, P. et al. (2014). Shotgun proteomics: identification of unique protein profiles of apoptotic bodies from biliary epithelial cells. *Hepatology*, 60(4):1314--1323.

Lodish, H.; Berk, A.; Kaiser, C. A.; Krieger, M.; Bretscher, A.; Ploegh, H. & Amon, A. (2014). *Biologia celular e molecular*. Artmed Editora.

Ma, B.; Zhang, K.; Hendrie, C.; Liang, C.; Li, M.; Doherty-Kirby, A. & Lajoie, G. (2003). Peaks: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry*, 17(20):2337--2342.

Marcotte, E. M. (2007). How do shotgun proteomics algorithms identify proteins? *Nat. Biotechnol.*, 25(7):755–757.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Singapore.

Nesvizhskii, A. I.; Keller, A.; kolker, E. & Aebersold, R. (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem.*, 75(17):4646–4658.

Peng, J.; Elias, J. E.; Thoreen, C. C.; Licklider, L. J. & Gygi, S. P. (2003). Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the Yeast proteome. *J. Proteome Res.*, 2:43–50.

Perkins, D. N.; Pappin, D. J. C.; Creasy, D. M. & Cottrell, J. S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567.

Pfeifer, N.; Leinenbach, A.; Huber, C. G. & Kohlbacher, O. (2007). Statistical learning of peptide retention behavior in chromatographic separations: a new kernel-based approach for computational proteomics. *BMC bioinformatics*, 8(1):468.

Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

Prosdocimi, F.; Coutinho, G.; Ninnecw, E.; Silva, A. F.; dos Reis, A. N.; Martins, A. C.; dos Santos, A. C. F.; Júnior, A. N. & Camargo Filho, F. (2002). Bioinformática: manual do usuário. *Biotecnologia Ciência & Desenvolvimento*, 29:12--25.

Russell, P. J. (2010). *IGenetics - A molecular approach 3rd ed.* Pearson Benjamin Cummings San Francisco.

Silverstein, R. M.; Webster, F. X.; Kiemle, D. & Bryce, D. L. (2014). *Spectrometric identification of organic compounds*. John Wiley & Sons.

Snustad, D. P.; Simmons, M. J. & Motta, P. A. (2008). *Fundamentos de genética*. Guanabara koogan.

Söderholm, S.; Hintsanen, P.; Öhman, T.; Aittokallio, T. & Nyman, T. A. (2014). Phosfox: a bioinformatics tool for peptide-level processing of lc-ms/ms-based phosphoproteomic data. *Proteome science*, 12(1):36.

Stuart, R. & Norvig, P. (2004). *Artificial Inteligence*. Pearson Education.

Swan, A. L.; Mobasheri, A.; Allaway, D.; Liddell, S. & Bacardit, J. (2013). Application of machine learning to proteomics data: classification and biomarker identification in postgenomics biology. *Omics: a journal of integrative biology*, 17(12):595--610.

Tan, P. N.; Steinbach, M. & Kumar, V. (2006). *Introduction to data mining*. Addison-Wesley, Boston.

Villén, J.; Beausoleil, S. A.; Gerber, S. A. & Gygi, S. P. (2007). Large-scale phosphorylation analysis of mouse liver. *Proc. Natl. Acad. Sci. U S A*, 104(5):1488–1493.

Walzthoeni, T.; Claassen, M.; Leitner, A.; Herzog, F.; Bohn, S.; Förster, F.; Beck, M. & Aebersold, R. (2012). False discovery rate estimation for cross-linked peptides identified by mass spectrometry. *Nature methods*, 9(9):901--903.

Wilhelm, M.; Schlegl, J.; Hahne, H.; Gholami, A. M.; Lieberenz, M.; Savitski, M. M.; Ziegler, E.; Butzmann, L.; Gessulat, S.; Marx, H. et al. (2014). Mass-spectrometry-based draft of the human proteome. *Nature*, 509(7502):582--587.

Williams, V.; Reading, B.; Hiramatsu, N.; Amano, H.; Glassbrook, N.; Hara, A. & Sullivan, C. (2014). Multiple vitellogenins and product yolk proteins in striped bass, morone saxatilis: molecular characterization and processing during oocyte growth and maturation. *Fish physiology and biochemistry*, 40(2):395--415.

Witten, I. H. & Frank, E. (2005). *Data Mning: Practical machine learning tools and techniquesi*. Morgan Kaufmann.

Anexo A

Artigo - X-Meeting 2015

RESEARCH

MUMAL2: Improving sensitivity in shotgun proteomics using cost sensitive artificial neural networks and a threshold selector algorithm

Fabio R Cerqueira[1*], Adilson M Ricardo[1,2†], Alcione P Oliveira[1,3], Amin Graber[4] and Christian Baumgartner[5]

[*]Correspondence:
fabio.cerqueira@ufv.br
[1] Department of Informatics, Universidade Federal de Viçosa, 36570-000 Viçosa, Brazil
Full list of author information is available at the end of the article
[†]Adilson M Ricardo and Fabio R Cerqueira contributed equally to this work.

Abstract

Background: This work presents a machine learning strategy to increase sensitivity in mass spectrometry data analysis for peptide/protein identification. Tandem mass spectrometry is a widely used analytical chemistry technique used to identify proteins in complex mixtures, yielding thousands of spectra in a single run which are then interpreted by software. Most of these computer programs use a protein database to match peptide sequences to the observed spectra. The peptide-spectrum matches (PSMs) must also be assessed by computational tools since manual evaluation is not practicable. The target-decoy database strategy is largely used for PSM assessment. However, in general, the method does not account for sensitivity, only for error estimate.

Results: In a previous study, we proposed the method MUMAL that applies an artificial neural network to effectively generate a model to classify PSMs using decoy hits with increased sensitivity. Nevertheless, the present approach shows that the sensitivity can be further improved with the use of a cost matrix associated with the learning algorithm. We also demonstrate that using a threshold selector algorithm for probability adjustment leads to more coherent probability values assigned to the PSMs. Our new approach, termed MUMAL2, provides a two-fold contribution to shotgun proteomics. First, the increase in the number of correctly interpreted spectra in the peptide level augments the chance of identifying more proteins. Second, the more appropriate PSM probability values that are produced by the threshold selector algorithm impact the protein inference stage performed by programs that take probabilities into account, such as ProteinProphet. Our experiments demonstrated that MUMAL2 provides a higher number of true positives compared with standard methods for PSM evaluation. This new approach reached around 15% of improvement in sensitivity compared to the best current method. Furthermore, the area under the ROC curve obtained was 0.93, demonstrating that the probabilities generated by our model are in fact appropriate. Finally, Venn diagrams comparing MUMAL2 with the best current method show that the number of exclusive peptides found by our method was nearly 4-fold higher, which directly impacts the proteome coverage.

Conclusions: The inclusion of a cost matrix and a probability threshold selector algorithm to the learning task further improves the target-decoy database analysis for identifying peptides, which optimally contributes to the challenging task of protein level identification, resulting in a powerful computational tool for shotgun proteomics.

Keywords: artificial neural network; cost sensitive classification; peptide/protein identification; phosphoproteomics; shotgun proteomics; data mining

Background

The goal in proteome studies is to characterize as many proteins as possible in the samples being analyzed, in order to assign to these proteins a role in cellular activities, including cases of severe disease occurrence due to protein malfunction [1, 2]. For this purpose, liquid chromatography coupled with tandem mass spectrometry (LC-MS/MS) is the most commonly used approach [3, 4, 5].

An LC-MS/MS run generates thousands of spectra, where each one represents a peptide. The next step is to assign a peptide sequence to each spectrum based on its spectral peak pattern [6]. There are basically two techniques to interpret MS/MS spectra. One is the so-called de novo approach that analyzes the peak patterns without using any external information [7]. The most common technique, however, uses protein sequence databases, which is the case of computational programs such as Sequest and Mascot [8, 9]. These programs perform an in silico digestion of proteins present in the database (DB) and generate virtual spectra from the resulting virtual peptides. Thus, for each observed spectrum, the program finds its best match to a virtual spectrum and the respective peptide sequence is assigned to the given MS spectrum. The programs normally report the ten best matches. Several scores are attributed to a peptide-spectrum matching (PSM) to measure its quality [10]. This strategy can be used to identify and quantify peptides/proteins [11]. Nevertheless, a major issue in this procedure is that a single LC-MS/MS run usually leads to thousands of spectra, where fewer than 20% are interpreted correctly [12].

In this work, we are primarily interested in the identification task. In particular, we aim at performing a computational curation of Sequest PSMs, given the enormous volume of spectra that is usually produced and the potentially large number of false positive hits. In this context, it is important to efficiently estimate the false discovery rate (FDR) of the identifications [13].

A common strategy to FDR estimation is the use of a target-decoy database (TDDB) [14, 15]. In this approach, decoy protein sequences are generated to be used along with target protein sequences for the search, which can be performed using a composite target-decoy DB or in two rounds, i.e., one search for each DB (decoy and target). Common methods for generating decoy sequences are to reverse or shuffle the target sequences, keeping the amino acid distribution. The TDDB strategy relies on the premise that the decoy PSMs are good models of the incorrect target PSMs. Hence, for a wrong PSM, the probability of the assigned peptide sequence to pertain to the target DB is assumed to be the same probability of the sequence to pertain to the decoy DB. As a result, a good estimate for the number of wrong spectrum interpretations among target PSMs is simply the number of decoy PSMs [16]. However, even though the TDDB strategy has been used successfully for FDR estimation, it has not been, in general, suitably applied to optimize sensitivity, i.e., more sophisticated combinations of the PSM scores are not fully explored to increase sensitivity [17]. Furthermore, important scores are left out from the FDR estimation process [13].

PeptideProphet is another known approach used to PSM assessment. This method considers mixed statistical distributions of PSM scores to predict correct and incorrect spectrum interpretations [18, 19]. For Sequest PSMs, e.g., the Gaussian and gamma distribution parameters for incorrect and correct PSMs, respectively,

are estimated by the Expectation-Maximization algorithm [20]. When the dataset presents the assumed distributions, PeptideProphet can provide an accurate probability that a PSM is correct. On the other hand, in certain datasets the scores might present completely different distributions. Particularly in the case of phosphopeptides, the peptide fragmentation process in the MS/MS run is biased towards phosphate groups, which suppresses important ions and leads to odd spectra [21, 22, 23].

MUDE and MUMAL are two more recently introduced methods proposed by our group that explore the TDDB strategy without assuming and relying on a data distribution [12, 3]. Both methods describe a more comprehensive use of PSM scores to enhance sensitivity.

MUDE considers in addition to Xcorr and $\triangle$Cn, normally used in TDDB analyses, four alternative scores: $\triangle$m, SpRank, and PercIons, provided by Sequest, and RTp-value, provided by the OpenMS proteomics tool [24, 12]. Furthermore, the problem of finding threshold values for the scores that lead to a desired FDR is treated as an optimization problem, unlike simplistic procedures previously described. Even though it provides a significant increase in sensitivity, the MUDE approach is capable of producing only linear decision boundaries to separate false positives from true positives because the score thresholds are defined individually.

As in MUDE, the MUMAL method to assess PSMs applies a TDDB analysis using a multivariate approach. However, this is accomplished with machine learning techniques, aimed at providing more flexible decision boundaries to further increase sensitivity in the FDR estimation process [3, 6]. MUMAL replaces the optimization procedure in MUDE with an Artificial Neural Network (ANN) algorithm to perform PSM classification. The resulting ROC (receiver operating characteristic) curve is analyzed according to the decoy count idea, i.e., for each point in the curve, the respective discriminant probability threshold t is used to count the number of decoy hits with probability (generated by the ANN) equal or greater than t. This count is the estimate for the number of target hits with $P > t$ that are incorrect. For the ANN model construction, the training set is the data to be analyzed itself, i.e., all Sequest hits, where the attributes are composed by the six scores mentioned above, and the class labels are: 0 for decoy hits, and 1 for target hits. If, on one hand, all of the decoy hits are obviously wrong, on the other hand, only a minor part of target hits are correct. For this reason, by using classical evaluation methods such as accuracy, precision, and recall, the resulting model is regarded as unsatisfactory because most target hits have characteristics similar to decoy hits. However, the ROC analysis is an optimal tool to find appropriate discriminant probabilities that provide the desired FDR with good sensitivity [25]. Nevertheless, there is room for improving sensitivity even further, particularly concerning the classification procedure, because other techniques could be applied for using decoy hits to characterize the wrong interpretations among target PSMs.

Therefore, we again propose to use ANNs as in MUMAL to keep delineating good decision boundaries. However, two important approaches are included in the PSM assessment procedure. The first one is the use of a cost matrix for making the cost of misclassifying an instance of class 0 (decoy hit) higher than the cost of misclassifying an instance of class 1 (target hit) [26]. It provides a bias toward the

correct classification of decoy hits, for which the class labels are definitely correct (decoy hits are obviously wrong). In this way, the incorrect target hits, i.e., the ones with the same characteristics of decoy PSMs, but with different label (class 1), tend to be correctly classified as class 0 by the model. Therefore, decoy hits help to pin down incorrect target hits, providing better decision boundaries, which leads to a higher sensitivity.

The second technique we use for improving the MUMAL approach is to apply a threshold selector algorithm (TSA) [27]. After building the model with an ANN with a cost matrix, and analyzing the ROC curve to see which discriminant probabilities provide suitable FDRs, the discriminant probability that leads to a 1% FDR is selected to be the final threshold value t that separates correct from incorrect PSMs. Next, threshold t is set to TSA, which replaces the probabilities generated by the ANN approach with probabilities that make more sense in terms of indicating the PSM correctness. For hits with probability = t, TSA replaces their probabilities with 0.5, and all other PSM probability values are proportionally normalized, keeping the range [0, 1], so that 0.5 is the point of separation between a set of PSMs with high FDR (those with P $<$ 0.5) and a set of PSMs with low FDR (those with P $\geq$ 0.5). Note that the previous version of MUMAL provides a good approach for separating PSMs with low FDR. However, due to the model problem caused by the fact that many class-1 instances have similar characteristics to class-0 instances, the probability value generated by the ANN approach for a target PSM is not appropriate for its individual evaluation. With the probability value adjustment provided by TSA, in turn, individual assessment of PSMs is now possible, which is very important for the protein inference stage, such as the one performed by ProteinProphet [18].

In this work, we performed experiments with 11 datasets, in a comparison with standard methods for PSM assessment, to demonstrate that our method, named MUMAL2, could achieve an average increase of 15% in sensitivity concerning the best current method, for FDRs varying from 0% to 5%. Still, by using Venn diagrams with peptides identified for a 1% FDR, we demonstrated that almost 4-fold more exclusive peptides were found. Furthermore, in an additional experiment using a dataset with known proteins, the ROC area calculated after the adjustment of probabilities by TSA was 0.93, showing coherent probability values. It is worth noting the demonstration of the predictive power of our method for phosphopeptides. In these cases, the score distribution might be very different from non-phosphopeptide PSMs, which complicates the analysis of traditional computational tools such as PeptideProphet.

Material and methods

Datasets

Eleven datasets used in the validation of MUDE and MUMAL were again utilized in our experiments [12, 3]. Figure 1 illustrates the datasets with their respective amounts of PSMs. Note that in all cases the number of target PSMs is slightly higher than half the total number of PSMs. This is reasonable because it is expected that less than 20% of target hits are correct. Therefore, the total amount of PSMs is composed by this small percentage plus the rest of incorrect PSMs, where, roughly, one half is composed of decoy hits and the other half contains wrong target hits.

These datasets were obtained from three LC-MS/MS runs with three independent phospho-enriched mouse samples. For each resulting set of spectra, Sequest was used for peptide sequence assignment. Each PSM dataset produced as output was split into two parts: The first part contained spectra whose best result was reported as a phosphopeptide, and the second part was made up of spectra whose best hit was attributed to a non-phosphopeptide. These sets were further split based on the precursor charge state, where only +2 and +3 charges were considered. As a consequence, the three Sequest outputs turned into 12 datasets that were labeled S1_PH_CH2, S1_PH_CH3, S1_NPH_CH2, S1_NPH_CH3, S2_PH_CH2, S2_PH_CH3, S2_NPH_CH2, S2_PH_CH3, S3_PH_CH2, S3_PH_CH3, S3_NPH_CH2, and S3_NPH_CH3, where PH and NPH denote phosphodata and non-phosphodata, respectively, while CH2 and CH3 represent +2 and +3 charge states, respectively. The dataset S3_NPH_CH3 was removed from the experiments since it had fewer than ten correct assignments. Finally, for each of these datasets, the Sequest files in the "out" format were converted into a single IdXML file, which is the format used by OpenMS, the computational toolkit we applied to predict retention time (RT) [28]. The details on the protocol and chemicals in sample preparation, MS technology applied, versions of programs and formats, parameters and database used in the Sequest search, etc., can be found in the previous works of Cerqueira et al. [23, 12, 3].

Another dataset we used in our experiments was taken from the work of Pfeifer et al. [24]. They used three samples containing known proteins. In our work, the PSMs of each mixture were also generated by Sequest and were joined in a single IdXML file that we refer to as M123. The proteins present in the mixtures are: β-casein (bovine milk), conalbumin (chicken egg white), myelin basic protein (bovine), hemoglobin (human, divided in subunits alpha and beta in the DB), leptin (human), creatine phosphokinase (rabbit muscle), α1-acid-glycoprotein (human plasma, appearing in two distinct versions in the DB), albumin (bovine serum), cytochrome C (bovine heart), β-lactoglobulin A (bovine), carbonic anhydrase (bovine erythrocytes), catalase (bovine liver), myoglobin (horse heart), lysozyme (chicken egg white), ribonuclease A (bovine pancreas), transferrin (bovine), β-lactalbumin (bovine), and thyroglobulin (bovine thyroid). Knowing the proteins we are supposed to identify facilitates the development of experiments to validate the performance of our method to appropriately curate PSMs. The details to produce this dataset can be found in the papers of Pfeifer et al. and Cerqueira et al. [24, 12].

Target-decoy database strategy

As recommended by Elias et al. [29], we used a composite target-decoy DB for the searches, where decoys were produced by reversing the target sequences. In this way, the peptide sequence of an incorrect PSM has an equal chance of coming from either a target or a decoy sequence. As a result, to estimate the number of wrong target hits, it suffices to count the number of decoy hits, i.e., the FDR estimate for target hits is given by: D_t /(N_t - D_t), where D_t is the number of decoy PSMs found with a score equal or greater than a predetermined threshold t, and N_t is the total number of PSMs (decoys and targets) according to the same threshold t.

In order to enhance sensitivity, as proposed previously [12, 3], the TDDB strategy is used here in a multivariate manner, taking into consideration six key PSM scores:

$\triangle C_n$, Xcorr, $\triangle$M, SpRank, percentage of ions found (all of them calculated by Sequest), and RT p-value (calculated by OpenMS). Furthermore, machine learning techniques are applied to promote a better separation between correct and incorrect hits, using decoy PSMs as a key part in this procedure, as described in the next sections.

Cost sensitive artificial neural network

Classical decoy approaches typically use no more than two scores that have their threshold values analyzed individually, which lead to linear decision boundaries. In order to construct more appropriated decisions boundaries between correct and incorrect hits, an ANN is used so that the six scores (the ANN's inputs) mentioned previously are applied in combination to produce a final score in the range [0, 1] (the ANN's output using a sigmoid function) that can be interpreted as a probability value. Then, using the decoy counting idea, a suitable threshold for this score is pursued to reach a desired FDR. Figure 2 illustrates the ANN architecture.

In the MUMAL work, the authors compared support vector machines with ANNs, and proved that the latter approach was capable of delivering higher sensitivity. The authors still observe that labeling decoy PSMs as class 0 and target PSMs as class 1 leads to a difficult classification task because most target PSMs are incorrect, i.e., they are similar to decoy hits. However, the goal is not providing a perfect separation between class-0 and class-1 instances. Once a model (even supposed of low quality using traditional metrics such as accuracy) is created, different discriminant probabilities are tested to obtain one that results in a sought FDR. This threshold exploitation using decoy counting is the key to separate what really matters, i.e., correct from incorrect hits.

In this work, we improve the MUMAL approach to further increase the sensitivity in PSM assessment. The selected strategy is to use the decoy instances in the dataset to pin down wrong target instances, so that the model's capacity to separate correct from incorrect hits is improved. For this purpose, a cost matrix is introduced to the classification task [30, 26, 27], where, considering target instances as positives, the cost of a false positive (CFP) is set higher than the cost of a false negative (CFN). Therefore, the final model will tend to classify class-0 instances correctly, while class-1 instances will be mostly "misclassified". The double quotes are to call the attention to the fact that the final goal is to construct a model to separate correct from incorrect PSMs, not separating target from decoys. Hence, when most target instances are classified as class 0 by the model, they are being, actually, correctly relabeled to class 0 because their peptide sequences were incorrectly assigned. Table 1 shows an example of a cost matrix that forces the model to favor decoy instances. The idea is to provoke a model bias toward decoy instances, leading to the relabeling of wrong target instances to class 0, resulting in better decision boundaries to separate correct from incorrect PSMs.

As can be seen in the results of our experiments, this relabeling process could be successfully accomplished. However, after fixing CFN = 1 and trying different values for CFP, we have realized that a certain CFP that leads to a good model for a given dataset is not necessarily the best choice for another dataset. As a result,

for each dataset given as input to our pipeline, ten different models are created varying CFP with the integer values in the range [1, 10], and many discriminant probabilities resulting in different FDR values for each case are reported. Next, the model with the highest average number of correct PSMs, for FDRs varying from 1 to 5%, is selected as the final classifier. This cost sensitive classification was implemented in the Java programming language using the Weka API v3.7.8 [31, 27].

ROC Curve

As already mentioned, several discriminant probabilities are explored after the model construction, so that the count of decoys considered as positives serves as an estimate to the number of wrong positive targets. This task is accomplished by analyzing the resulting ROC curve [26]. For each point in the curve, the respective discriminant probability t is used to count the number of decoy hits with probability (generated by the ANN) equal or greater than t. This count is the estimate for the number of target hits with $P > t$ that are incorrect. As described before, class labels in the datasets are: 0 for decoy hits, and 1 for target hits. It is expected that a minor part of target hits are correct. For this reason, by using classical evaluation methods such as accuracy, precision, recall, and area under the curve (AUC), a very poor classification model is expected. However, we stress the fact that the goal is not separating decoys from targets, but incorrect from correct hits, and the ROC analysis suffices to establish appropriate discriminant probabilities that provide the desired FDR with a better sensitivity when compared to classical target-decoy approaches.

Figure 3 illustrates such an analysis. For each point in the curve (Figure 3a), the discriminant probability t (threshold) and the respective statistics (Figure 3b) are known. The FDR estimation (Figure 3c) is made by dividing the number of FPs by the number of TPs because FPs and TPs are, respectively, the decoys and targets with $P > t$.

Threshold Selector

We could see that the ROC analysis gives us the chance of selecting an appropriate probability threshold value leading to a desired FDR. This is enough if the goal is just the selection of a low-FDR set of PSMs. Nonetheless, such a set is normally used in the protein inference stage, where the proteins in the sample being analyzed are ultimately identified. To this end, the probability associated to each PSM is of great importance, mainly for computational tools that use this value as a key measure to infer proteins, e.g., ProteinProphet. On the other hand, the probability values originally assigned by the ANN to the instances in the dataset do not reflect the correctness of PSMs. These values indicate, instead, whether PSMs are decoys or targets because the class labels where defined this way. Considering that most targets (the incorrect ones) are similar to decoys, even the distinction of decoys and targets is not very well characterized in these probabilities. The AUC = 0.55 seen in Figure 6 clearly shows this fact.

In order to obtain appropriate probability values indicating PSM correctness, we came up with another improvement by using the threshold selector algorithm (TSA) implemented in the Weka API. This algorithm can work in two ways. In the first setting, TSA automatically finds a discriminant probability that optimizes some given

measure such as F-measure, accuracy, precision and recall. In the second setting, TSA is given a fixed threshold value. Then, TSA forces the classifier to predict as positive all instances with probability greater or equal to the given threshold, or as negative, otherwise. Our pipeline uses the latter option along with the probability range correction that TSA provides. In this correction procedure, TSA replaces the probabilities that are equal to the given threshold with 0.5 and expands the other values so that the minimum probability observed maps to 0, while the maximum maps to 1.

The threshold given to TSA is defined as follows. After building a model with a cost-sensitive ANN, with the best cost matrix, and analyzing the ROC curve to perform FDR estimations, the discriminant probability that results in 1% FDR is selected to be the final threshold value t that separates correct from incorrect PSMs. We have chosen 1% because this is the best trade-off between sensitivity and precision, as described by Elias et al. and Balgley et al. [32, 33]. Next, the threshold t is given to TSA that adjusts the probabilities generated by the ANN, producing new values that are more appropriate to indicate the PSM correctness. For PSMs with P = t, TSA replaces their probabilities with 0.5. All other PSM probability values are modified as described above. As a result, we finally obtain a classifier that separates correct from incorrect PSMs (not target from decoys) with the usual mid-point probability value 0.5 as the point of separation between negatives and positives.

Framework

Figure 4 shows a flowchart that summarizes MUMAL2's framework. Ten cost-sensitive ANNs are built for CFP varying from 1 to 10. Then, the best CFP is selected according to the execution with the highest sensitivity. Furthermore, the probability threshold leading to FDR=1% of the best execution is saved. A final model is thus built with the best CFP and using TSA with the saved threshold. TSA makes a range correction, where PSMs with probability equal to the saved threshold have their probabilities replaced with 0.5. Additionally, the other probability values are expanded, such that the minimum probability is set to 0 and the maximum is set to 1. As a consequence, the PSMs with FDR=1% are assigned high probabilities (≥ 0.5).

Results and discussion

To evaluate MUMAL2, the parameters were kept with default values, i.e., number of nodes in the hidden layer = 4, momentum = 0.2, learning rate = 0.3, and epochs = 1000. The experiments were performed on a Linux machine equipped with Intel® Celeron® CPU N2830 2.16 GHz × 2, and 4 GB of RAM. Our intention was to prove that MUMAL2 can provide a quick answer even on personal computers. In fact, one iteration of MUMAL2, i.e., one execution of the strategy cost matrix + ANN, takes 20 s on average. Because eleven executions to produce the final model are needed, the total time taken is 220 s, in general. We realized that it is not significantly different from MUMAL's running time because MUMAL has also to execute a number of iterations to produce its best results. Therefore, we concentrate the analyses of our experiments on the capacity of our approach to assess PSMs.

Measuring the predictive power of MUMAL2

First, dataset M123, whose proteins are known, was used to measure the predictive power of our method. We analyzed whether the relabeling of wrong PSMs in class-1 to class-0, i.e., the establishment of a suitable decision boundary between correct and incorrect hits, could be satisfactorily accomplished. Figure 5 contains plots of $\triangle$Cn vs Xcorr for dataset M123 before (Figure 5a) and after (Figure 5b) running MUMAL2. Class-0 PSMs are represented in blue, whereas class-1 PSMs are shown in red. A dense cloud of points in Figure 5a can bee seen composed of approximately of 50% decoys and 50% targets. According to the target/decoy principle, this dense cloud represents the set of wrong PSMs. Therefore, we are interested in the part that is comprised mostly of red points (likely correct targets). As already mentioned, the use of a cost matrix to construct an accurate model for the class-0 instances is an attempt to keep these instances as such, since decoy PSMs are obviously wrong, whereas correctly relabeling the wrong class-1 instances, the ones mixed with decoys in the dense cloud, to class 0. Figure 5b shows that the decision boundary produced by MUMAL2 seems to provide a good separation of the mixture of decoys/targets from the homogeneous part composed of red points. Notice that the confusion matrix on the top makes evident the huge amount of target PSMs that were classified as class 0, which was expected because most of these PSMs are known to be wrong. The plot built as a result of the instance relabeling shows that the vast majority of instances in the dense cloud turned into blue.

Even though the plot in Figure 5b indicates that the region of interest (the part in Figure 5a composed mostly of red points) could be identified, it is possible to provide more precise measurements of the quality of our solution because the proteins of dataset M123 are known. The confusion matrix on the top of Figure 5b shows that 12 decoy hits were mistakenly classified as class 1. Probably, some correct target hits were incorrectly relabeled to class 0 as well. Thus, to provide a more precise assessment of our strategy, the 8917 instances predicted as class 0 and the 1059 predicted as class 1 had their peptide sequences inspected to check whether or not they came from the set of expected proteins. As a result, we could build a more useful confusion matrix, considering positive or negative those instances whose peptide sequence came from the list of known proteins or from a random protein, respectively. As a consequence, we could use classical metrics that express the predictive power of an ML model, as shown in Table 2. It can be seen that MUMAL2's classification was highly accurate. Only 25 instances (12 negatives and 13 positives) were misclassified, also leading to very high values of sensitivity, specificity, and precision. However, it is important to highlight that the decoy hits are known to be wrong PSMs. Therefore, the 12 misclassified decoys shown in Figure 5b have no importance, i.e., only target instances are further considered after the classification.

It is clear, thus, that the application of a supervised ML method here does not follow the classic steps: Build the learning model using a training set, and apply the resultant model to unknown instances. In our case, the ANN is trained and applied to the same data. Notice that the target instances are the ones of interest. Our final aim is to separate wrong targets from correct targets. to this end, we use a higher cost for FPs to force the model to learn how to correctly classify decoys whose labels are correct. That is why we say that decoys help to pin down wrong

targets. Next, we apply the final model on the same data to relabel the wrong target hits, i.e., the ones with similar features to decoy hits, to class 0. Thus, it does not make sense to talk about cross-validation to evaluate the model. Instead, we have to verify whether correct and incorrect targets are being identified.

Another important aspect to analyze is whether the probabilities produced by our model are coherent. This is a very relevant matter if the intention is to use a method for protein inference that takes PSM probabilities into account, such as ProteinProphet. As shown in Figure 3, the AUC and other measures are low because the wrong target hits that are correctly relabeled to class 0 are counted as misclassification. In fact, the blind application of MUMAL2 to dataset M123 leads to an AUC of 0.60. However, as we know the proteins of dataset M123, we can produce the real ROC curve to evaluate the probabilities generated by TSA. Figure 6 shows the ROC curve built for dataset M123 counting as TPs those instances whose peptides came from the list of known proteins, and counting as FPs, otherwise. As can be seen, the AUC is very satisfactory (nearly 0.93), showing that the probability values are appropriate, and corroborating the high predictive power of MUMAL2.

MUMAL2 was also applied to the other datasets for which the proteins are not known a priori. However, it is possible to use plots of $\triangle$Cn vs Xcorr, as previously shown for dataset M123, to perform a visual inspection. Figure 7 shows this analysis for dataset S1_PH_CH2. It can be seen in the confusion matrix (Figure 7b) that a significant number of class-1 instances were relabeled to class 0, as expected. In the plot of Figure 7b, the colors of the points indicate that the classification seem to be appropriate because the blue points correspond to those in Figure 7a composed of a mixture of targets and decoys, i.e., the part where targets are probably wrong, according to the target/decoy principle. Performing the same analysis for the other datasets led to very similar outcomes (not shown), i.e., MUMAL2 promoted an expressive migration of target hits to class 0, resulting in a plot where class-0 instances correspond to the mixture of class-0 and class-1 instances before the application of MUMAL2.

Comparing MUMAL2 with previously proposed methods

The next experiments demonstrate the superior sensitivity of MUMAL2 in relation to important methods for PSM assessment: MUMAL, MUDE, PeptideProphet, and bivariate decoy/target analyses, where the thresholds of two scores, often $\triangle$Cn and Xcorr, leading to a desired FDR are pursued. For comparisons with phosphodata, we included a bivariate analysis with $\triangle$M and Xcorr, following Beausoleil et al. and Jiang et al. recommendation [34, 22]. According to them, $\triangle$Cn scores are often suppressed when phosphopeptides have more than one potential phosphorylation site. Therefore, $\triangle$M should be used instead. For a detailed description of how the methods used in the comparison were run to produce the results shown next, refer to the works of Cerqueira et al. [35, 3].

Figures 8 and 9 show the curves of the number of identified PSMs vs estimated FDR for all above-mentioned methods applied to the eleven datasets of unknown proteins. It is possible to build such curves because all those approaches provide an effective way to estimate the FDR value for a given set of selected PSMs. The

curves show FDR values varying from 0 to 5%, which are the error rates commonly accepted. It can be seen in all cases that MUMAL2 is superior than MUDE, PeptideProphet, and the bivariate analyses.

Regarding MUMAL, MUMAL2 has an equal or greater sensitivity. It is expected because MUMAL2 performs 10 executions with CFP varying from 1 to 10. The execution with CFP = 1 is equivalent to the MUMAL execution. Therefore, MUMAL2 cannot be worst, but it can eventually present the same sensitivity as MUMAL. Notice, however, that the number of cases where MUMAL2 has a greater sensitivity is higher than the cases of equal performance. Our method could provide an average increase of 16.5% and 7.2% in relation to MUMAL for non-phosphodata and phosphodata, respectively. It means about 24 and 20 more peptides, on average, respectively.

In particular for a 1% FDR, which is a commonly pursued FDR value, MUMAL2 demonstrates superiority in all cases. For non-phosphorylated proteins, the PSM evaluation provided by MUMAL2 for this FDR led to an average improvement in sensitivity of 16.8% compared with MUMAL, meaning about 23 additional peptides. For phosphorylated proteins, in turn, the increase was approximately 12%, resulting in nearly 31 more peptides.

Figures 8 and 9 demonstrate that MUMAL presented the best performance among the methods being compared with our approach. For this reason, we performed an additional experiment to compare MUMAL2 with MUMAL by means of Venn diagrams to call attention to the higher number of exclusive peptide identifications provided by the former. Figure 10 shows this counting for a 1% FDR in both methods. In all cases, an expressive superiority of MUMAL2 can be noted. On average, the number of exclusive PSMs that our method could find is almost 4-fold greater. This is an important result because more peptides may imply more identified proteins and a higher proteome coverage.

Conclusions

The target-decoy database strategy is widely used for data analysis in shotgun proteomics. Many previous studies have demonstrated the effective capacity of this approach for FDR estimation. However, the classical TDDB procedure does not take sensitivity into account. Fortunately, this fact has been changing since the introduction of MUDE and MUMAL.

In this work, we have further improved sensitivity in MS/MS-based peptide/protein identification by using advanced machine learning methods that use decoys to establish more appropriate decision boundaries. Furthermore, the probabilities assigned to PSMs by our method are proven to be highly accurate. This is a fundamental matter to improve protein inference when the applied approach depends on such probability values, as in the case of ProteinProphet.

We could demonstrate that our new approach has great potential to provide important improvements in protein identification, which will impact future studies that seek a broader understanding of notable cell activities. Hopefully, future research on drug discovery, diseases, and many other studies in life sciences will be positively affected by this new computational strategy for peptide/protein identification.

Addendum: URL for software download
The software is open-source and is available under the URL: `http://sourceforge.net/projects/mumal/`

Competing interests
The authors declare that they have no competing interests.

Author's contributions
AMR and FRC contributed equally to this work. FRC, AG, and CB designed all analyses; AMR, FRC, and APO were responsible for carrying out the analyses; AMR, FRC, and CB wrote the initial draft of the manuscript; APO and AG contributed to posterior revisions to the final draft. All authors read and approved the final paper.

Acknowledgements
This work is supported by FAPEMIG, CNPq, and CAPES.

Author details
[1] Department of Informatics, Universidade Federal de Viçosa, 36570-000 Viçosa, Brazil. [2] Department of Computing and Construction, Centro Federal de Educação Tecnológica de Minas Gerais, Rua 19 de Novembro, 121, 35180-008 Timóteo, Brazil. [3] Department of Computer Science, University of Sheffield, Western Bank, S10 2TN, Sheffield, UK. [4] Research and Product Development of Genoptix, a Novartis company, 2110 Rutherford Rd, 92008 Carlsbad, USA. [5] Institute of Health Care Engineering with European Notified Body of Medical Devices, Graz University of Technology, Stremayrgasse 16/II, A-8010 Graz, Austria.

References

1. Kim, M.-S., Pinto, S.M., Getnet, D., Nirujogi, R.S., Manda, S.S., Chaerkady, R., Madugundu, A.K., Kelkar, D.S., Isserlin, R., Jain, S., *et al.*: A draft map of the human proteome. Nature **509**(7502), 575–581 (2014)
2. Kumar, A., Rajendran, V., Sethumadhavan, R., Shukla, P., Tiwari, S., Purohit, R.: Computational SNP analysis: Current approaches and future prospects. Cell biochemistry and biophysics **68**(2), 233–239 (2014)
3. Cerqueira, F.R., Ferreira, R.S., Oliveira, A.P., Gomes, A.P., Ramos, H.J., Graber, A., Baumgartner, C.: MUMAL: Multivariate analysis in shotgun proteomics using machine learning techniques. BMC genomics **13**(Suppl 5), 4 (2012)
4. Wilhelm, M., Schlegl, J., Hahne, H., Gholami, A.M., Lieberenz, M., Savitski, M.M., Ziegler, E., Butzmann, L., Gessulat, S., Marx, H., *et al.*: Mass-spectrometry-based draft of the human proteome. Nature **509**(7502), 582–587 (2014)
5. Lleo, A., Zhang, W., McDonald, W.H., Seeley, E.H., Leung, P.S., Coppel, R.L., Ansari, A.A., Adams, D.H., Afford, S., Invernizzi, P., *et al.*: Shotgun proteomics: Identification of unique protein profiles of apoptotic bodies from biliary epithelial cells. Hepatology **60**(4), 1314–1323 (2014)
6. Swan, A.L., Mobasheri, A., Allaway, D., Liddell, S., Bacardit, J.: Application of machine learning to proteomics data: Classification and biomarker identification in postgenomics biology. Omics **17**(12), 595–610 (2013)
7. Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., Lajoie, G.: PEAKS: Powerful software for peptide de novo sequencing by tandem mass spectrometry. Rapid communications in mass spectrometry **17**(20), 2337–2342 (2003)
8. Eng, J.K., McCormack, A.L., Yates, J.R.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J. Am. Soc. Mass Spectrom. **5**, 976–989 (1994)
9. Perkins, D.N., Pappin, D.J.C., Creasy, D.M., Cottrell, J.S.: Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis **20**(18), 3551–3567 (1999)
10. Söderholm, S., Hintsanen, P., Öhman, T., Aittokallio, T., Nyman, T.A.: PhosFox: A bioinformatics tool for peptide-level processing of LC-MS/MS-based phosphoproteomic data. Proteome science **12**(1), 36 (2014)
11. Silverstein, R.M., Webster, F.X., Kiemle, D., Bryce, D.L.: Spectrometric Identification of Organic Compounds, 8th edn., USA (2014)
12. Cerqueira, F.R., Graber, A., Schwikowski, B., Baumgartner, C.: MUDE: A new approach for optimizing sensitivity in the target-decoy search strategy for large-scale peptide/protein identification. Journal of proteome research **9**(5), 2265–2277 (2010)
13. Ivanov, M.V., Levitsky, L.I., Lobas, A.A., Panic, T., Laskay, U.A., Mitulovic, G., Schmid, R., Pridatchenko, M.L., Tsybin, Y.O., Gorshkov, M.V.: Empirical multidimensional space for scoring peptide spectrum matches in shotgun proteomics. Journal of proteome research **13**(4), 1911–1920 (2014)
14. Walzthoeni, T., Claassen, M., Leitner, A., Herzog, F., Bohn, S., Förster, F., Beck, M., Aebersold, R.: False discovery rate estimation for cross-linked peptides identified by mass spectrometry. Nature methods **9**(9), 901–903 (2012)
15. He, K., Fu, Y., Zeng, W.-F., Luo, L., Chi, H., Liu, C., Qing, L.-Y., Sun, R.-X., He, S.-M.: A theoretical foundation of the target-decoy search strategy for false discovery rate control in proteomics. arXiv preprint arXiv:1501.00537 (2015)
16. Granholm, V., Noble, W.S., Käll, L.: A cross-validation scheme for machine learning algorithms in shotgun proteomics. BMC bioinformatics **13**(Suppl 16), 3 (2012)
17. Li, Y.F., Radivojac, P.: Computational approaches to protein inference in shotgun proteomics. BMC bioinformatics **13**(Suppl 16), 4 (2012)
18. Keller, A., Nesvizhskii, A.I., Kolker, E., Aebersold, R.: Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. Anal. Chem. **74**(20), 5383–5392 (2002)
19. Nesvizhskii, A.I., Keller, A., kolker, E., Aebersold, R.: A statistical model for identifying proteins by tandem mass spectrometry. Anal. Chem. **75**(17), 4646–4658 (2003)
20. Mitchell, T.M.: Machine Learning. McGraw-Hill, Singapore (1997)
21. Imanishi, S.Y., Kochin, V., Ferraris, S.E., Thonel, A., Pallari, H.M., Corthals, G.L., Eriksson, J.E.: Reference-facilitated phosphoproteomics: Fast and reliable phosphopeptide validation by μLC-ESI-Q-TOF MS/MS. Mol. Cell. Proteomics **6**, 1380–1391 (2007)

22. Jiang, X., Han, G., Feng, S., Jiang, X., Ye, M., Yao, X., Zou, H.: Automatic validation of phosphopeptide identifications by the MS2/MS3 target-decoy search strategy. J. Proteome Res. **7**, 1640–1649 (2008)
23. Cerqueira, F.R., Morandell, S., Ascher, S., Mechtler, K., Huber, L.A., Pfeifer, B., Graber, A., Tilg, B., Baumgartner, C.: Improving phosphopeptide/protein identification using a new data mining framework for MS/MS spectra preprocessing. J. Proteomics Bioinform. **2**, 150–164 (2009)
24. Pfeifer, N., Leinenbach, A., Huber, C.G., Kohlbacher, O.: Statistical learning of peptide retention behavior in chromatographic separations: A new kernel-based approach for computational proteomics. BMC bioinformatics **8**(1), 468 (2007)
25. Lasko, T.A., Bhagwat, J.G., Zou, K.H., Ohno-Machado, L.: The use of receiver operating characteristic curves in biomedical informatics. Journal of biomedical informatics **38**(5), 404–415 (2005)
26. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Boston (2006)
27. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, Burlington (2011)
28. Pfeifer, N., Leinenbach, A., Huber, C.G., Kohlbacher, O.: Statistical learning of peptide retention behavior in chromatographic separations: a new kernel-based approach for computational proteomics. BMC Bioinformatics **8**(1), 468 (2007)
29. Elias, J.E., Gygi, S.P.: Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nat. Methods **4**(3), 207–214 (2007)
30. Elkan, C.: The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence, vol. 17, pp. 973–978 (2001)
31. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD explorations newsletter **11**(1), 10–18 (2009)
32. Elias, J.E., Gibbons, F.D., King, O.D., Roth, F.P., Gygi, S.P.: Intensity-based protein identification by machine learning from a library of tandem mass spectra. Nat. Biotechnol. **22**, 214–219 (2004)
33. Balgley, B.M., Laudeman, T., Yang, L., Song, T., Lee, C.S.: Comparative evaluation of tandem MS search algorithms using a target-decoy search strategy. Mol. Cell. Proteomics **6**, 1599–1608 (2007)
34. Beausoleil, S.A., Villén, J., Gerber, S.A., Rush, J., Gygi, S.P.: A probability-based approach for high-throughput protein phosphorylation analysis and site localization. Nat. Biotechnol. **24**, 1285–1292 (2006)
35. Cerqueira, F.R., Graber, A., Schwikowski, B., Baumgartner, C.: MUDE: A New Approach for Optimizing Sensitivity in the Target-Decoy Search Strategy for Large-Scale Peptide/Protein Identification. J. Proteome Res. **9**(5), 2265–2277 (2010)

Figures

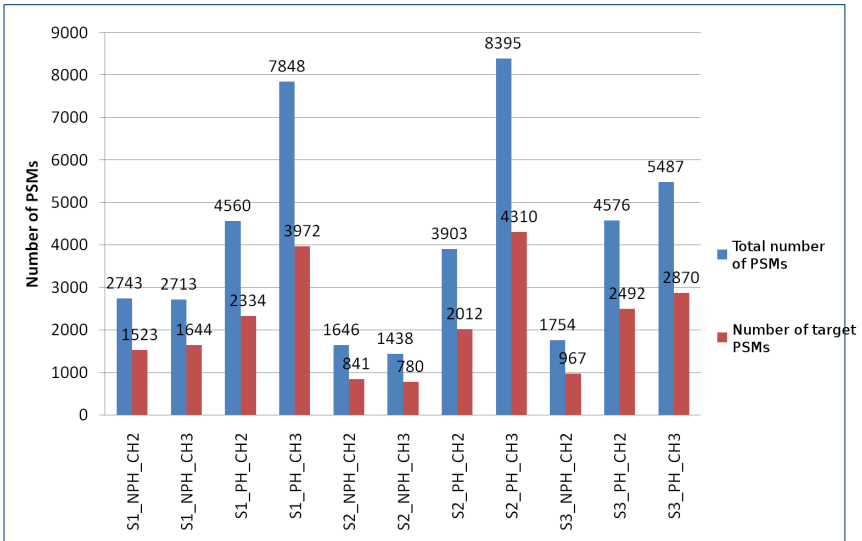


Figure 1 The eleven datasets used in most of our experimental work and evaluation.

Tables

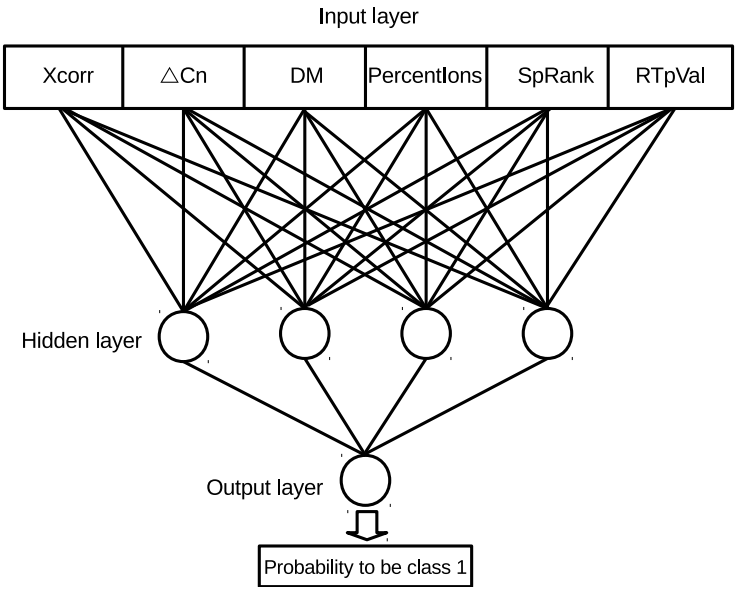


Figure 2 Architecture of the artificial neural network used in MUMAL and in this work. It illustrates the input layer (six nodes), the hidden layer (four nodes by default), and the output layer (one node), where the activation function is the sigmoid to map the ANN's output into a value in the range [0, 1].

Table 1 Cost matrix for a 2-class (class 0 and class 1) classifier. In this case, the cost of a false positive is 10 times higher than the cost of a false negative. CTN = cost of a true negative, CFP = cost of a false positive, CFN = cost of a false negative, and CTP = cost of a true positive.

		Predited Class	
		0	1
Given Class	0	CTN=0	CFP=10
	1	CFN=1	CTP=0

Table 2 Assessing MUMAL2 according to the known proteins of datset M123. In (a), a confusion matrix is shown, where positive and negative instances are not target and decoys anymore. Instead, an instance is considered positive if its peptide sequence came from the list of known proteins. Otherwise, the instance is considered negative. In (b), known statistical measures are presented to evaluate the predictive power of MUMAL2.

		Predicted Class	
		0	1
Actual Class	0	8904	12
	1	13	1047

(a)

Statistical measures	
Accuracy	0.9975
Sensitivity	0.9877
Specificity	0.9987
Precision	0.9887

(b)

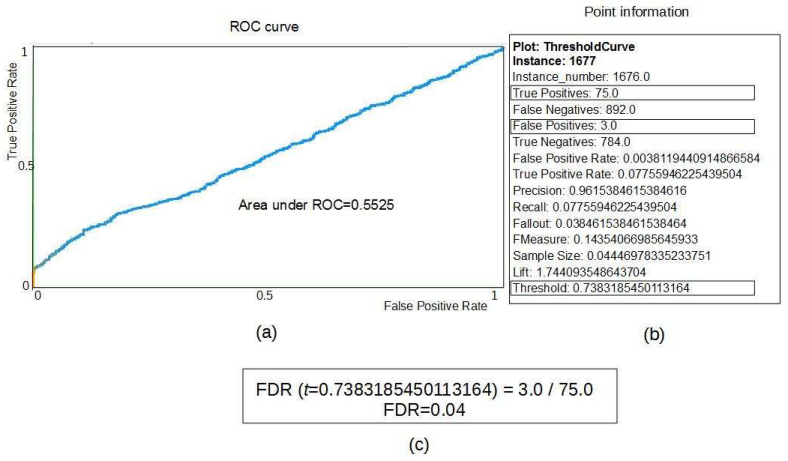


Figure 3 Analysis of a ROC curve obtained from a model built with dataset S2_NPH_CH2. For each point of the curve (a), the threshold and several statistical measures (b) related to that point are known. False and true positives are, respectively, instances of class 0 (decoys) and class 1 (targets) that were considered positives (P > threshold) by the model. In (c), it is shown how to estimate FDR among target hits for a given discriminant probability (threshold) of the model. It is simply the number of FPs over the number of TPs.

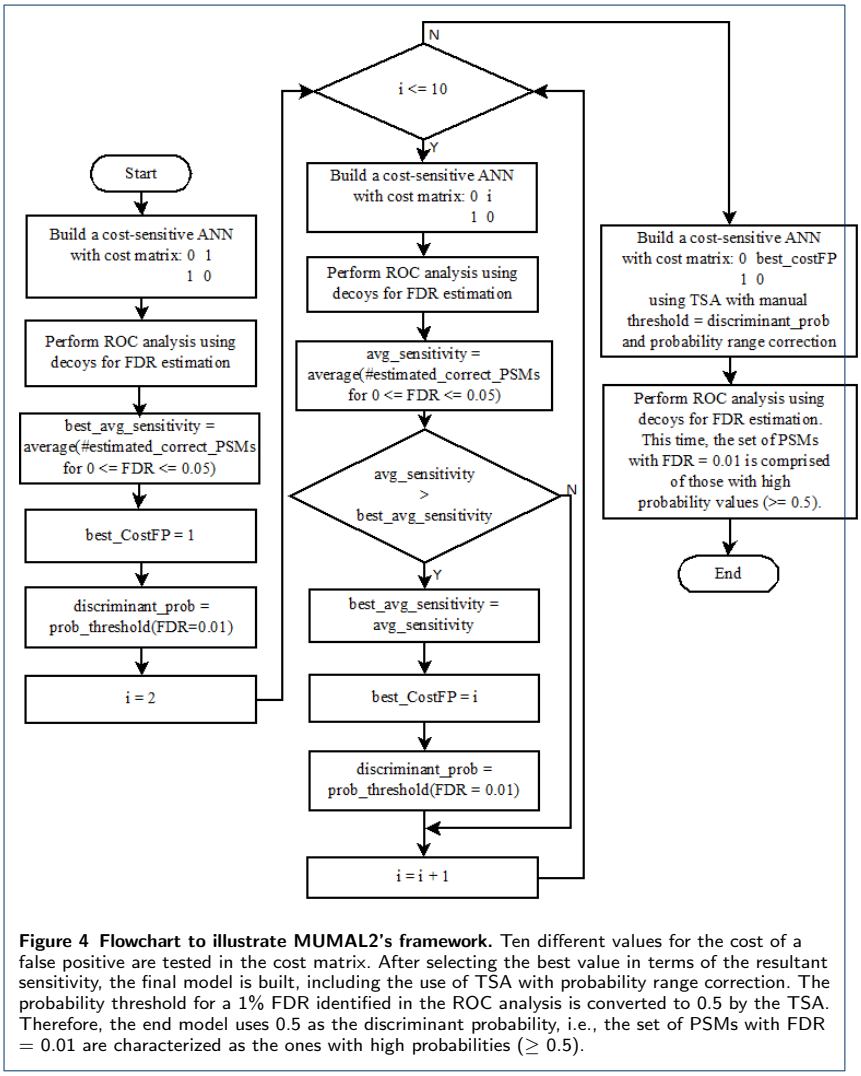


Figure 4 Flowchart to illustrate MUMAL2's framework. Ten different values for the cost of a false positive are tested in the cost matrix. After selecting the best value in terms of the resultant sensitivity, the final model is built, including the use of TSA with probability range correction. The probability threshold for a 1% FDR identified in the ROC analysis is converted to 0.5 by the TSA. Therefore, the end model uses 0.5 as the discriminant probability, i.e., the set of PSMs with FDR = 0.01 are characterized as the ones with high probabilities ($\geq$ 0.5).

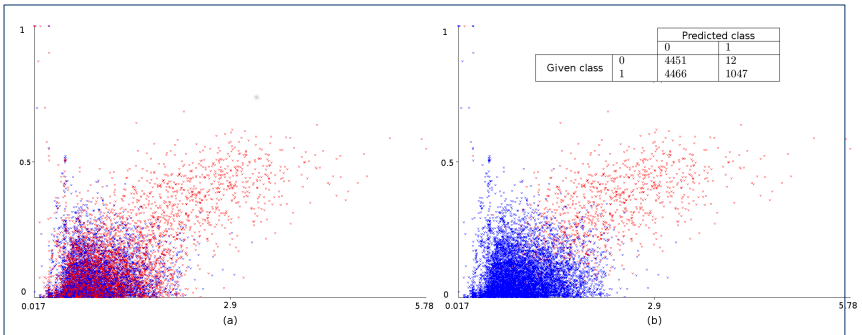


Figure 5 Instance relabeling provided by MUMAL2. A plot of △Cn vs Xcorr for dataset M123 is shown in (a) and (b), before and after applying MUMAL2, respectively. Class-0 instances are shown in blue, while class-1 instances are shown in red. Part (b) includes also the confusion matrix on the top to show the huge number of class-1 instances that were classified as class 0.

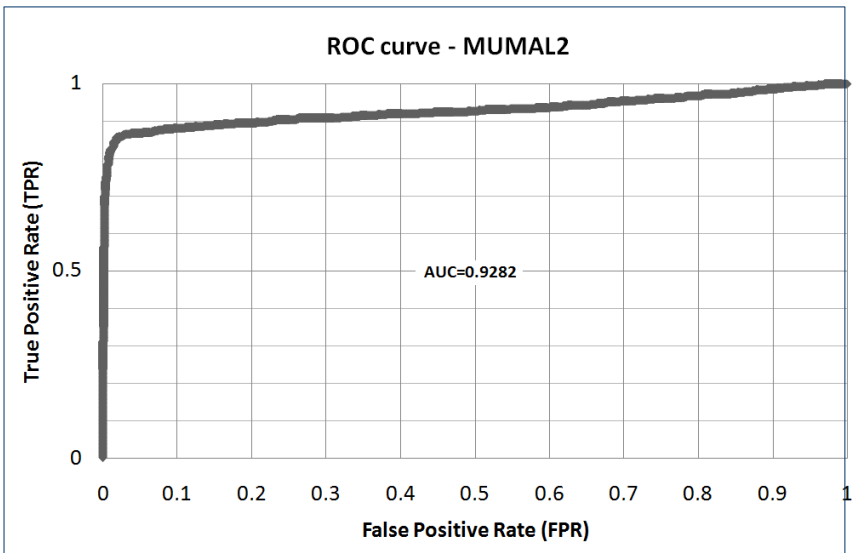


Figure 6 ROC curve of MUMAL2's model for dataset M123. An instance is regarded as TP if its peptide sequence came from the list of expected proteins. Otherwise, the instance is considered FP.

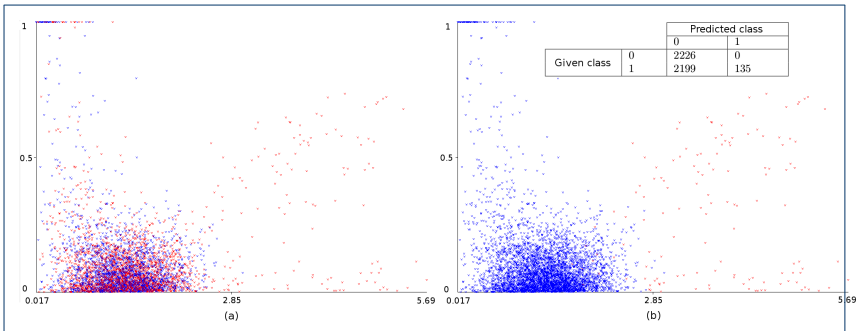


Figure 7 Instance relabeling performed by MUMAL2 for dataset S1_PH_CH2. A plot of △Cn vs Xcorr for the dataset is shown in (a) and (b), before and after using MUMAL2, respectively. Class-0 hits are shown in blue, whereas class-1 hits are shown in red. Part (b) still presents the confusion matrix on the top, demonstrating the significant number of class-1 examples that were classified as class 0.

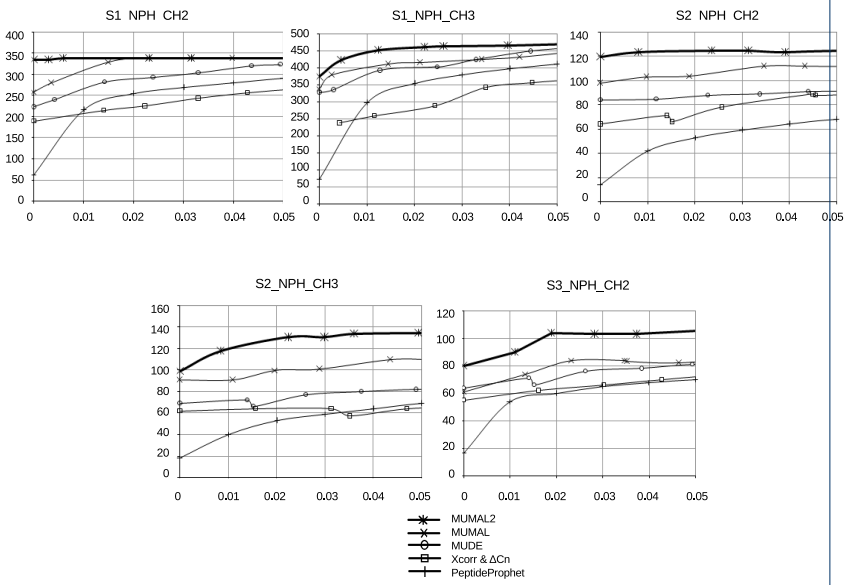


Figure 8 Comparing the sensitivity of MUMAL2 and other approaches with data of non-phosphorylated proteins. For each dataset, a curve of the number of identified PSMs vs estimated FDR is shown. FDR values vary from 0% and 5%.

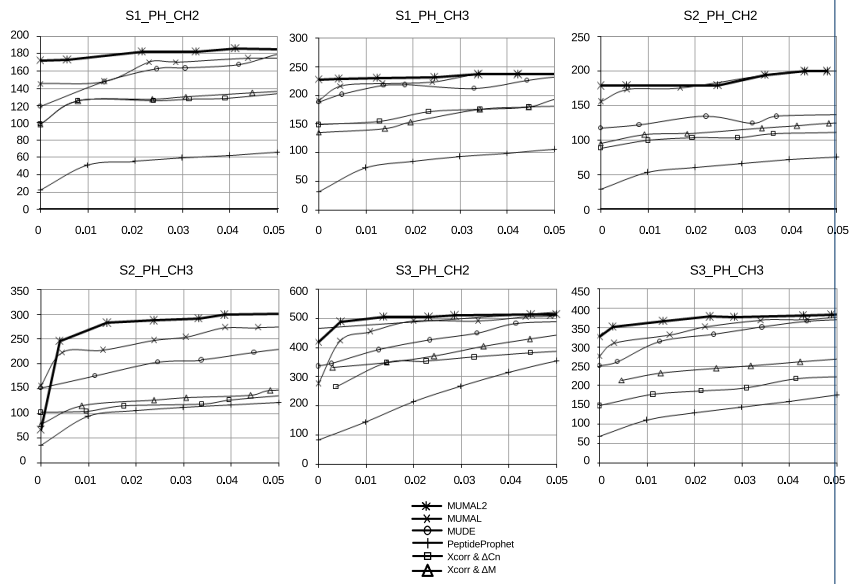


Figure 9 Comparing the sensitivity of MUMAL2 and other approaches with data of phosphorylated proteins. For each dataset, a curve of the number of identified PSMs vs estimated FDR is shown. FDR values vary from 0% and 5%.

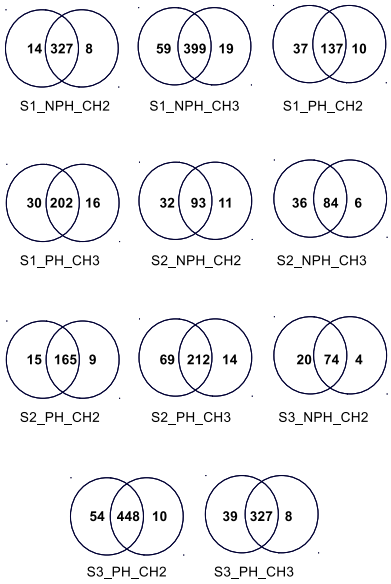


Figure 10 Comparison of MUMAL2 with MUMAL by Venn diagrams containing the number of PSMs selected for FDR=1%. For each diagram, the left-hand side contains the number of peptides identified exclusively by our method, while the right-hand side shows the number of identifications found exclusively by MUMAL.

Anexo B

Poster - X-Meeting 2015

MUMAL2: Improving sensitivity in shotgun proteomics using cost sensitive artificial neural networks and a threshold selector algorithm

Fabio R Cerqueira[1], Adilson M Ricardo[1,2], Alcione P Oliveira[1,3], Armin Graber[4], and Christian Baumgartner[5]

[1] Departamento de Informática, Universidade Federal de Viçosa, Brazil, [2] Departamento de Computação e Construção Civil, Centro Federal de Educação Tecnológica de Minas Gerais, Brazil, [3] Department of Computer Science, University of Sheffield, UK, [4] Research and Product Development of Genoptix, a Novartis company, USA, and
[5] Institute of Health Care Engineering with European Notified Body of Medical Devices, Graz University of Technology. Austria.

Universidade Federal de Viçosa

Introduction

This work presents a machine learning (ML) strategy to increase sensitivity in mass spectrometry data analysis for peptide/protein identification. Tandem mass spectrometry is a widely used analytical chemistry technique to identify proteins in complex mixtures, yielding thousands of spectra in a single run which are then interpreted by software. Most of these computer programs use a protein database to match peptide sequences to the observed spectra. The peptide-spectrum matches (PSMs) must also be assessed by computational tools since manual evaluation is not practicable. The target-decoy database strategy is largely used for PSM assessment. However, in general, the method does not account for sensitivity, only for error estimate. In a previous study, we proposed the method MUMAL that applies an artificial neural network to effectively generate a model to classify PSMs using decoy hits with increased sensitivity.

Nevertheless, the present approach (termed MUMAL2) shows that the sensitivity can be further improved with the use of a cost matrix associated with the learning algorithm. We also demonstrate that using a threshold selector algorithm for probability adjustment leads to more coherent probability values assigned to the PSMs.

Material and methods

The key ML techniques we use in our pipeline for improving sensitivity are shown in Figure 1. The classification problem here considers decoy hits as class 0 and target hits as class 1. We use the artificial neural network architecture shown in Figure 1 associated with a cost matrix that penalizes false positives (FPs). FPs are decoy hits that are classified as positives. Notice that decoy hits are obviously wrong, while target hits are mostly wrong. Therefore, the intention is to build an ML model that classifies well decoy hits, so that these instances are kept in this class, while wrong target hits migrate to the class 0, i.e., it is possible to effectively identify wrong target hits. In the example shown in Figure 1 the cost of a false positive is 10 times higher than the cost of a false negative.

The ROC curve analysis helps to identify a discriminant probability that leads to an acceptable false discovery rate (FDR). The FDR estimate is calculated by counting the number of decoys that are considered as positives and dividing this number by the number of true positives (target hits considered as positives).

The threshold selector (TSA) technique is used to adjust the generated probabilities. For a certain discriminant probability p selected with the ROC curve, p is given to TSA. Then, TSA replaces the probabilities that are equal to the given threshold with 0.5 and expands the other values so that the minimum probability observed maps to 0, while the maximum maps to 1. This procedure generates more coherent probabilities that represent the quality of a PSM. Coherent probabilities are very important for the protein inference stage, such as the one performed by the method ProteinProphet.

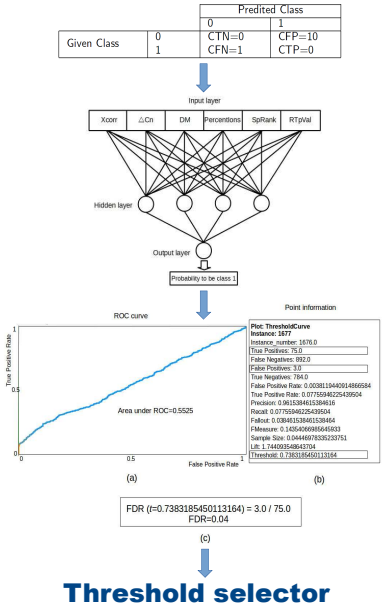


Figure 1: *Key ML techniques that compose our pipeline for improving sensitivity.*

Results

For the experiments, we used one dataset whose proteins are known (M123), and eleven datasets where proteins are unknown, i.e., the evaluation was based on the estimated FDR.

Figure 2 shows an experiment with dataset M123 where it can be seen a cloud composed mostly of wrong hits (decoys + wrong targets). This cloud is successfully detected by MUMAL2. The confusion matrix clearly shows that most target hits (which are expected to be wrong) migrated to class 0.

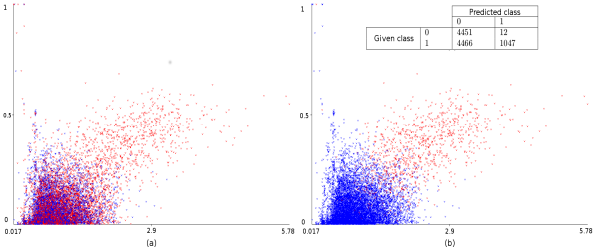


Figure 2: *Experiment with dataset M123 showing the successful relabeling of wrong target hits to class 0. At left, decoys are represented in blue and targets in red.*

However, as we know the proteins present in M123, it is possible to build the real confusion matrix and use important metrics to measure the predictive power of the method, which are shown in Table 1.

		Predicted Class	
		0	1
Actual Class	0	8904	12
	1	13	1047

(a)

Statistical measures	
Accuracy	0.9975
Sensitivity	0.9877
Specificity	0.9987
Precision	0.9887

(b)

Table 1: *(a) The real confusion matrix can be constructed, as the proteins present in dataset M123 are known. (b) Also, accuracy, sensitivity, specificity, and precision can be measured.*

Finally, Figure 3 compares MUMAL2 with standard methods, using the eleven datasets of unknown proteins. It can be seen that MUMAL2 can achieve higher sensitivity for the same FDRs, even for datasets of phosphorylated proteins.

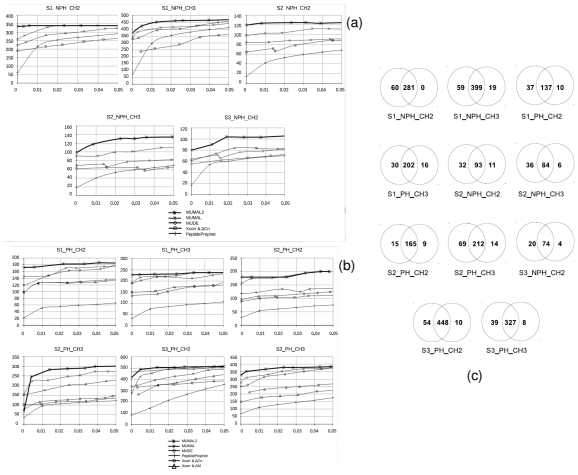


Figure 3: *Comparing MUMAL2 to other approaches using non-phosphorylated (a) and phosphorylated (b) proteins. For each dataset, a curve of the number of identified PSMs vs estimated FDR is shown. (c) Comparison of MUMAL2 with MUMAL by Venn diagrams containing the number of PSMs selected for FDR=1%. For each diagram, the left-hand side contains the number of peptides identified exclusively by our method, while the right-hand side shows the number of identifications found exclusively by MUMAL.*

Acknowledgements

This work is supported by CAPES, CNPq, FAPEMIG.