UNIVERSIDADE FEDERAL FLUMINENSE

Alvaro Sergio Di Lauro Pedreira

Predição de interações proteína-proteína através de cobertura de conjuntos utilizando uma metaheurística GRASP

NITERÓI
2012

UNIVERSIDADE FEDERAL FLUMINENSE

Predição de interações proteína-proteína através de cobertura de conjuntos utilizando uma metaheurística GRASP

Alvaro Sergio Di Lauro Pedreira

Dissertação de Mestrado submetida ao corpo docente do Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como parte dos requisitos necessários para obtenção do grau de Mestre em Computação.

Orientador:

Carlos Alberto de Jesus Martinhon

NITERÓI
2012

UNIVERSIDADE FEDERAL FLUMINENSE

Predição de interações proteína-proteína através de cobertura de conjuntos utilizando uma metaheurística GRASP

Alvaro Sergio Di Lauro Pedreira

Dissertação de Mestrado submetida ao corpo docente do Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como parte dos requisitos necessários para obtenção do grau de Mestre em Computação.

Aprovada por:

_____

Prof. Carlos Alberto de J. Martinhon / IC-UFF
(Presidente)

_____

Prof. Luiz Satoru Ochi / IC-UFF

_____

Prof. Carlile Campos Lavor / IMECC-UNICAMP

Resumo

A identificação das interações de pares de domínios é um passo importante para a compreensão das interações proteína-proteína. Nos últimos anos, vários métodos para predizer interações protéicas foram propostos. Compreender a capacidade e as limitações desses métodos é fundamental para o desenvolvimento de abordagens melhores, e para a maior compreensão da natureza dessas interações.

Neste trabalho, propomos um algoritmo GRASP para cobertura do conjunto de pares de proteínas com o objetivo de encontrar o menor conjunto de pares de domínios responsáveis pelas interações do conjunto de proteínas com maior especificidade. Estabelecemos a hipótese de que as interações entre proteínas evoluíram de forma parcimoniosa, ou seja, que o conjunto correto de interações domínio-domínio é bem aproximado por um conjunto mínimo de interações de domínio necessário para justificar uma determinada rede de interações proteína-proteína.

Palavras-chaves: interação proteína-proteína, interação domínio-domínio, problema da cobertura de conjuntos, metaheurística GRASP

Abstract

The identification of interacting domain pairs is an important step towards understanding protein interactions. In recent years, several methods to predict proteins interactions have been proposed. Understanding the capabilities and limitations of these methods is crucial to the development of improved approaches and better understanding of the nature of these interactions.

We also propose a GRASP algorithm to cover the set of pairs of proteins with the goal of finding the smallest set of pairs of domains responsible for interactions of all proteins with greater specificity. We established the hypothesis that interactions between proteins evolved sparingly and that the correct set of domain-domain interactions is well approximated by a minimal set of domain interactions necessary to justify a specific network of protein-protein interactions.

Keywords: protein-protein interactions; domain-domain interactions; set cover problem; GRASP metaheuristic

Agradecimentos

Gostaria de expressar minha profunda gratidão ao meu orientador, Prof. Dr. Carlos de Jesus Martinhom, por sua orientação e aconselhamento. Sem o seu incentivo, eu não teria terminado essa dissertação.

Agradeço também aos meus colegas atuais e passados.

Finalmente, gostaria de expressar minha profunda gratidão aos meus pais e família pelo apoio durante toda minha vida.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

1. Introdução

Elucidar redes de interações de proteínas é um dos principais objetivos da genômica funcional para organismos plenamente sequenciados [1].

As proteínas interagem umas com as outras de uma maneira altamente específica, e as interações proteína-proteína desempenham um papel fundamental em muitos processos celulares, em particular, a distorção de interfaces de proteína podem levar ao desenvolvimento de várias doenças. Para entender os mecanismos de reconhecimento das proteínas em nível molecular e para desvendar o quadro global de interações entre proteínas na célula, diferentes técnicas experimentais têm sido desenvolvidas. Algumas delas caracterizam as interações individuais de proteínas, enquanto outras avançaram para o rastreamento de interações em escala genômica [2]. Ainda que, várias destas abordagens experimentais, tais como sistema de duplo híbrido de levedura [3], [4], e métodos de espectroscopia de massa [5] tenham produzido uma enorme quantidade de informações sobre interações proteína-proteína (PPIs - *Protein-Protein Interactions*), os bancos de dados que armazenam estas PPIs têm uma reduzida quantidade de experimentos de confirmação e elevados níveis de falsos positivos [6], [7]. Além disto, estas técnicas experimentais são custosas, demoradas e trabalhosas.

A escassa cobertura de interações, juntamente com os vieses experimentais em relação a certos tipos de proteínas e localizações celulares relatados pela maioria das

técnicas experimentais motivaram o desenvolvimento de métodos computacionais para predizer se duas proteínas interagem. Estes métodos podem ser muito úteis para a escolha de potenciais alvos para rastreamento experimental ou para validação de dados experimentais e podem fornecer informações sobre detalhes de interação (no caso de métodos de predição de domínio), que podem não ser aparentes a partir das técnicas experimentais. Outros métodos utilizam combinações de técnicas experimentais e computacionais de maneira diferente (por exemplo, os métodos de co-expressão de genes e de letalidade sintética, estão entre abordagens experimentais da revisão de Shoemaker, B. e Panchenko, A [8]) e não predizem interações físicas diretamente, mas sim, inferem as associações funcionais entre potenciais interações entre proteínas [8] (veja Tabela 1).

Em consequência, vários métodos computacionais foram propostos para inferência de interações proteína-proteína. Enright e colaboradores [9] e Marcotte [10] propuseram o método *gene fusion / Rosetta stone*. Seu método encontra pares de proteínas que supostamente interagem se cada um deles é codificado separadamente como um gene distinto em um organismo, e se encontram fusionados em outro organismo. Marcotte e colaboradores [11] também propuseram um método que combina múltiplas fontes de dados, tais como proteínas que evoluíram de forma correlacionada e RNA mensageiro correlacionado a padrões de expressão. Wojcik e Schachter [12] propuseram o método de perfil de interação de pares de domínios, Gomez e Rzhetsky [13] propuseram modelos probabilísticos para formar uma rede de interações proteína-proteína com base em probabilidades de interações (atrações e repulsões) entre os domínios.

Tabela 1: Diferentes métodos de predição.
Adaptado de [8]

Nome do Método	Interação Proteina (P) ou Domínio (D)	Interação Física ou Associação Funcional
Co-expressão de Genes	P	Funcional
Letalidade Sintética	P	Funcional
Perfil Filogenético	P,D	Funcional
Rosetta Stone	P	Funcional
Co-evolução de Sequencia	P,D	Funcional
Classificação	P,D	Física
Integrativo	P,D	Física
Domínios	D	Física
Redes Baysianas	P,D	Funcional, Física
Exclusão de Par de Domínios	D	Física
p-Valor	D	Física

Outros métodos foram propostos para inferir interações domínio-domínio (e/ou interações assinatura-assinatura) a partir de dados de interação proteína-proteína. Dados de interação de domínios são úteis, não só para a compreensão mais detalhada das interações proteína-proteína, mas também para predizer interações proteína-proteína. Sprinzak e Margalit [14] propuseram o método da associação (AM - *Association Method*) que atribui uma pontuação para cada par de domínios. Kim e colaboradores [15] propuseram "scores" semelhantes e aplicaram a pontuação dada aos pares de domínios para predizer interações proteína-proteína. Deng e colaboradores [16] propuseram um modelo probabilístico, o qual será adotado neste trabalho, que define a interação de duas proteínas se, e somente se, existe pelo menos um par de domínios interagindo. Além disso, propuseram um algoritmo de expectativa-maximização (EM-*Expectation-Maximization*) para estimar a probabilidade de interação para cada par de domínios. Eles compararam os métodos EM e AM utilizando dados de interação proteína-proteína obtidos por Uetz e colaboradores [17] e Ito e colaboradores [18], e os resultados mostraram que o Método EM foi superior ao Método da Associação. No entanto, a precisão (capacidade de predizer interações verdadeiras) alcançada na classificação foi considerada baixa.

Mais recentemente, Huang e colaboradores [19] introduziram um novo método para predição de interações proteína-proteína. A concepção deste procedimento probabilístico de cobertura de conjuntos, parte da generalização das complexas relações de interações entre proteínas e suas arquiteturas de domínio, e permite a determinação de um conjunto de interações de domínios de proteína que descreve a presença de interações protéicas com máxima especificidade, a ser discutida mais adiante.

Neste trabalho apresentamos uma heurística GRASP (*Greedy Randomized Adaptative Search Procedure*) visando à determinação de soluções aproximadas e de boa qualidade para a predição de interações de pares de proteínas baseada em domínios. Adotando o modelo do conjunto de máxima especificidade (MSSC) proposto por Huang e seus colaboradores [19], modelo este implementado por cobertura de conjuntos utilizando um algoritmo guloso puro, propomos um método de construção e busca local visando a determinação de soluções de maior especificidade (como discutido mais adiante, na seção 3.5). Os resultados computacionais obtidos foram similares a implementação gulosa de Huang e colaboradores [19] quanto à sensibilidade e especificidade.

Este trabalho de dissertação está organizado do seguinte modo: No Capítulo 2, descrevemos o problema da predição de interação de proteínas, a notação básica utilizada e os principais métodos de avaliação. No Capítulo 3, revisamos os métodos de Associação, de Estimativa de Máxima Verossimilhança, da Força de Interação utilizando Programação Linear, da Explicação Parcimoniosa, e da Cobertura de Conjuntos de Máxima Especificidade (MSSC). No Capítulo 4, descrevemos a heurística GRASP proposta. No Capítulo 5, descrevemos as instâncias que foram utilizadas como entradas para os algoritmos, os resultados computacionais obtidos e sua análise correspondente. Por fim,

no Capítulo 6, apresentamos as conclusões e propostas para futuras aplicações. No Anexo, descrevemos em detalhe as características dos bancos de dados utilizados na obtenção das instâncias aplicadas aos algoritmos.

CAPÍTULO 2

2. Problema da Predição de Interações Proteína-Proteína (PIP)

2.1 Problema da Predição de Interações Proteína-Proteína

A predição de interações proteína-proteína é um campo combinado da bioinformática e da biologia estrutural numa tentativa de identificar e catalogar interações entre pares ou grupos de proteínas. Compreender estas interações é essencial para investigar as vias de sinalização intracelular. A partir de dados produzidos por técnicas experimentais, vários métodos computacionais têm sido desenvolvidos com a finalidade de predição das interações proteína-proteína [20].

As investigações sobre a estrutura espacial da proteína sugerem que a unidade fundamental da estrutura da proteína é um domínio. Esta região de uma cadeia polipeptídica dobra-se em uma estrutura distinta que fornece a funcionalidade biológica da proteína. A maioria das proteínas de procariontes possui apenas um domínio [21], enquanto que as seqüências de eucariontes multicelulares aparecem como proteínas multi-domínios de até 130 domínios [22].

A Figura 1 ilustra essa situação. O nosso objetivo é selecionar pares de domínios (pares de formas geométricas na figura) que explicam a rede de interação de proteínas conhecida. Utilizando os domínios selecionados a partir dos pares, podemos inferir quais

pares de domínios (DDIs) podem ser responsáveis pela interação dos pares de proteínas (PPIs)

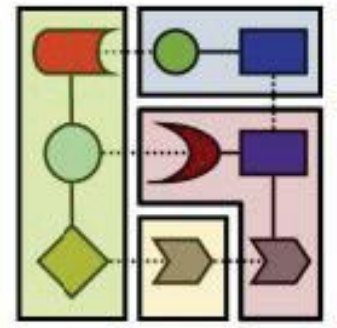


Figura 1: Esquema de Interação de Proteínas:
As figuras geométricas representam os domínios que formam as proteínas. (extraído de [16])

A Figura 2 ilustra o processo de predição de interação de proteínas com base nas informações de domínios. Neste caso, o autor não teve a preocupação de representar o "encaixe" geométrico, como na Figura 1.

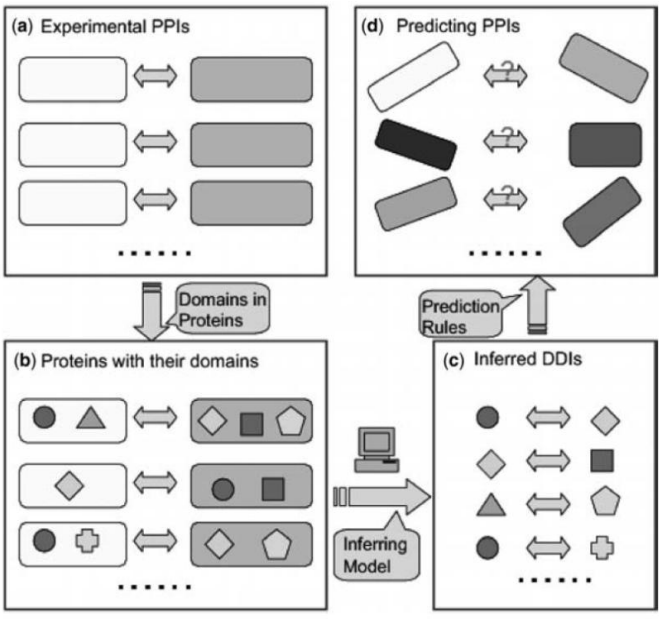


Figura 2: Representação esquemática do processo de predição de PPIs a partir de domínios
(a) Uma lista de PPIs obtidos experimentalmente. (b) Os domínios apropriados são atribuídos a cada uma das proteínas que interagem. (c) DDIs inferidos a partir de PPIs e suas informações de domínios. (d) PPIs preditos a partir dos DDIs inferidos. (extraído de [23])

2.2 Notação Utilizada

Seja P_1,..., P_n as proteínas no conjunto de dados de treinamento, e D_1,..., D_m os domínios. Representaremos por P_{ij} e D_{mn} um par de proteínas (P_i, P_j) e um par de domínios (D_m, D_n), respectivamente.

O modelo probabilístico idealizado por Deng e colaboradores [16] será utilizado neste trabalho. Neste modelo probabilístico de interações de proteínas as interações Proteína-Proteína e as interações Domínio-Domínio são representadas por variáveis aleatórias:

$$P_{ij} = \begin{cases} 1, se\ e\ somente\ se, P_i\ interage\ com\ P_j \\ 0, caso\ contrário \end{cases} \text{ e,}$$

$$D_{mn} = \begin{cases} 1, se\ e\ somente\ se, D_m\ interage\ com\ D_n \\ 0, caso\ contrário \end{cases}$$

É assumido que as interações Domínio-Domínio são independentes, e duas proteínas interagem se, e somente se, pelo menos um par de domínios, um de cada proteína, interage.

Sob estas condições, a probabilidade de P_i interagir com P_j é dada por:

$$\Pr(P_{ij} = 1) = 1 - \prod_{D_{mn} \epsilon P_{ij}} (1 - \lambda_{mn}) \qquad 2.1$$

onde $D_{mn} \epsilon P_{ij}$ significa que $D_m \epsilon P_i$ e $D_n \epsilon P_j$, e λ_{mn} denota a probabilidade que D_m interaja com D_n, ou seja, $\lambda_{mn} = Pr(D_{mn} = 1)$. Note que, se um par de domínios (D_m, D_n) interage com $Pr(D_{mn} = 1) = 1$ então $Pr(P_{ij} = 1) = 1$, ou seja, o par de proteínas contendo D_m, D_n respectivamente, também interage.

De modo geral, o resultado obtido é um novo conjunto de pares de proteínas que interage com probabilidades determinadas pelas "forças de interação".

2.3 Medidas Estatísticas de Avaliação de Desempenho

2.3.1 Aferição de Eventos

O tema da aferição de eventos ocupa substancial parcela dos compêndios da metodologia científica das pesquisas, e muitos textos da área da saúde, tratam pormenorizadamente do assunto. Uma das maneiras de abordá-lo consiste no estudo dos erros ou vieses que são introduzidos quando da realização de mensurações ou observação dos eventos. Brevemente, o viés de aferição é um erro sistemático introduzido no processo de aferição. Maiores detalhamentos podem ser obtidos em [24].

Em seguida são discutidos os aspectos fundamentais concernentes à mensuração dos eventos e os dois conceitos fundamentais na matéria: a *reprodutibilidade* e a *validade* da informação.

Definição dos termos

- *Reprodutibilidade* (confiabilidade ou precisão; em inglês *reproducibility*, *reliability*, *precision*) - é a consistência de resultados quando a medição ou o exame se repete.
- *Validade* (acuidade, acurácia ou exatidão; em inglês *validity*, *acuracy*) - refere-se ao grau em que o exame é apropriado para medir o verdadeiro valor daquilo que é medido ou observado. A validade informa se os resultados representam a "verdade" ou quanto se afasta dela.

Existe uma relação complexa entre reprodutibilidade e validade. Vejamos esta relação através de um exemplo.

Exemplo: Teste de uma arma por um exímio atirador [24].

- Se todos os tiros disparados acertaram o centro do alvo, a arma é confiável porque os tiros estão concentrados em um mesmo ponto. Ela também é de alta validade, por acertar a marca central.
- Se os tiros estão concentrados, mas fora do alvo, a arma é confiável, ou seja, repetindo-se a operação, se obtém os mesmos resultados. No entanto, ela é pouco válida; os tiros ficaram distantes da marca central.
- Se os tiros estão dispersos, a arma é pouco confiável; e ela é de validade limitada, por não acertar o alvo.

O exemplo da arma pode ser generalizado para os testes diagnósticos. Um teste de baixa reprodutibilidade forçosamente acarreta baixa validade, o que significa pouca utilidade. Um teste de alta reprodutibilidade, no entanto não assegura alta validade: observe-se que um exame pode ser reproduzível (dar resultados idênticos ou próximos quando o processo diagnóstico é repetido), mas não ser capaz de discriminar corretamente as diversas situações que se encontram na prática. Todos os resultados podem estar errados. Como este aspecto é fundamental para separar corretamente as classes (doentes de sadios ou proteínas que interagem ou não), a questão da validade tem de ser adequadamente esmiuçada, ao lado da reprodutibilidade, no sentido de avaliar a qualidade de um teste, e consequentemente, da informação por ele produzida.

2.3.2 Validade em relação a um padrão

Este tipo de validade é muito popular na área da saúde, em especial, em epidemiologia, farmacologia e patologia clínica. Refere-se a quanto, em termos quantitativos, um teste é útil para diagnosticar um evento (validade simultânea ou concorrencial) ou predizê-la (validade preditiva). Para tal, comparam-se os resultados do

teste com os de um padrão: esse pode ser o verdadeiro estado do paciente, se tal informação esta disponível, um conjunto de exames julgados mais adequados ou outra forma de diagnóstico que sirva como referencial. Os princípios de validade concorrente e preditiva são idênticos e serão apresentados a seguir. A estimativa da validade em relação a um determinado padrão referencial é feito pela aplicação do teste, separadamente, a um grupo ou classe (doentes e de sadios ou proteínas que interagem ou não), permitindo atestar o seu nível de validade. Conhecendo-se a proporção de acertos (verdadeiros positivos e verdadeiros negativos) e erros (falso-positivos e falso-negativos), estabelecem-se os diversos ângulos pelos quais a validade é expressa: sensibilidade, a especificidade, e os valores preditivos (veja os indicadores na Tabela 2).

Tabela 2: Modelo para a avaliação de um teste diagnóstico

Teste	Condição positiva	Condição negativa	Total
Resultado do teste positivo	Verdadeiro positivo (a)	Falso positivo (b)	a+b
Resultado do teste negativo	Falso negativo (c)	Verdadeiro negativo (d)	c+d
Total	a+c	b+d	N

N=quantidade total de examinados = a+b+c+d

Indicadores*

Sensibilidade = a/(a+c)
Especificidade = d/(b+d)
Prevalência real = (a+c)/N
Prevalência estimada (teste) = (a+b)/N
Valor preditivo positivo = a/(a+b)
Valor preditivo negativo = d/(c+d)
Classificação Correta = (a+d)/N
Classificação Incorreta = (b+c)/N

*Habitualmente expressos em porcentagens

Definição de Sensibilidade e Especificidade do Teste

A sensibilidade e a especificidade são medidas estatísticas do desempenho de um teste de classificação binário.

A sensibilidade é a capacidade que o teste apresenta de detectar os elementos verdadeiramente positivos, ou seja, diagnosticar corretamente os indivíduos doentes, no caso das pesquisas clínicas, ou classificar corretamente um par de proteínas que interage. Neste caso, quanto menor o total de falsos negativos, maior a sensibilidade.

A especificidade é a capacidade que o teste tem de detectar os verdadeiros negativos, isto é, identificar corretamente os indivíduos sadios, no caso dos estudos clínicos. Neste caso, quanto menor o total de falsos positivos, maior a especificidade.

Definição dos valores preditivos (ou diagnósticos) do Teste

O Valor preditivo positivo é a proporção de doentes entre os considerados positivos ao teste.

O valor preditivo negativo é a proporção de sadios entre os negativos ao teste.

Sensibilidade e especificidade são propriedades inerentes ao teste e não variam substancialmente a não ser por mudanças na técnica ou por erros na sua aplicação. O mesmo não ocorre com os valores preditivos do teste, que dependem da prevalência real do evento. Em conseqüência, a interpretação do valor preditivo deve ser cuidadosa. Trata-se de uma questão de probabilidade e não de certeza, em que assumem um papel crucial não somente a sensibilidade e especificidade do teste, mas também, a prevalência real do evento no conjunto. Observa-se que os valores preditivos positivos aumentam com a prevalência, enquanto os negativos diminuem.

Terminologia na recuperação da informação

Em reconhecimento de padrões e recuperação de informação, o valor preditivo positivo é denominado a precisão e representa a fração de instâncias recuperadas que são relevantes, enquanto a sensibilidade é denominada "*recall*" e é a fração de casos relevantes que são recuperados. Tanto a precisão e *recall* são, portanto, basedas no entendimento e na medida do que seja considerado como relevante.

CAPÍTULO 3

3. Trabalhos Relacionados

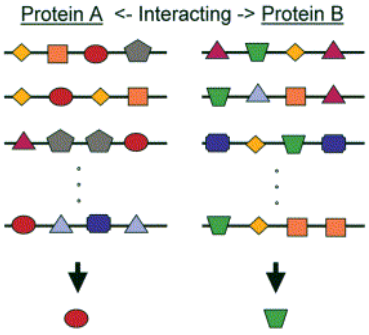
Neste capítulo, introduzimos alguns dos métodos desenvolvidos revisando suas características principais.

3.1 Método da Associação

O padrão de domínios que aparece em conhecidas proteínas que interagem também pode ajudar a predizer PPIs adicionais. Sprinzak e Margalit [14] propuseram o uso de pares de domínios, denominados assinaturas de sequências, que se repete com frequência em várias proteínas que interagem. Primeiramente caracterizaram as proteínas pelas suas assinaturas de sequências e derivaram uma tabela de contingência. Então, identificaram os pares de assinaturas de sequências com maior representação, comparando as freqüências observadas para aqueles que surgem aleatoriamente. Neste método simples denominado Método da Associação (AM - *Association Method*), uma pontuação é atribuída a cada par de domínios (D_m, D_n). Seja I_{mn} o número de pares de proteínas que interagem (no conjunto de treinamento) contendo o par de domínios (D_m, D_n). Seja N_{mn} o número de pares de proteínas (no conjunto de treinamento) contendo pares de domínios (D_m, D_n). Então, a pontuação (probabilidade da interação) para (D_m, D_n) é dada por

$$A(D_m, D_n) = \frac{I_{mn}}{N_{mn}} \tag{3.1}$$

Uma visão esquemática de assinaturas de sequências de pares de proteínas que interagem e da tabela de contingência correspondente são ilustrados na Figura 3. Esse método se baseia no pressuposto de que todas as interações ocorrem dentro de interações domínio-domínio bem definidas



	1							
	3	0						
	2	1	0					
	3	2	**4**	0				
	2	2	2	2	1			
	2	0	1	1	2	0		
	1	1	1	2	0	0	0	
	2	1	2	0	0	0	1	0

Figura 3: Visão esquemática do Método de Associação

No painel superior, cada linha contém as sequências de um par de proteínas (A, B), cuja interação foi determinada experimentalmente. Cada seqüência é caracterizada por suas assinaturas, denotado aqui esquematicamente por formas coloridas. Na parte inferior do painel, uma tabela de contingência das combinações de assinatura é descrito, em que cada entrada (i, j) na tabela mostra o número de pares de proteínas que contêm as assinaturas i e j combinadas, onde uma proteína contém a assinatura i e a outra proteína companheira de par contém a assinatura j. Por exemplo, o par de assinaturas de sequências representado por um retângulo laranja e um triângulo rosa aparece em dois pares de proteínas que interagem. O par mais abundante de assinatura de sequências é a de uma elipse vermelho e um trapézio verde que aparece em quatro pares diferentes de proteínas que interagem. No passo seguinte da análise, a probabilidade dos pares de assinaturas de sequência é avaliada. (Extraído de [14]).

3.2 Método de Estimativa de Máxima Verossimilhança

Deng e colaboradores [16] introduziram o modelo probabilístico (equação 2.1), e propuseram um algoritmo de Maximização de Expectativa (EM - *Expectation-Maximization*) para estimar a probabilidade de interação para cada par de domínios. No seu trabalho foram considerados dois tipos de erros experimentais: falsos positivos, nos quais duas proteínas não interagem na realidade, mas foram observadas interagindo nas experiências, e falsos negativos, nos quais duas proteínas interagem na realidade, mas não foram observadas interagindo nas experiências. Sejam fp e fn as variáveis que representam a taxa de falsos positivos e da taxa de falsos negativos, respectivamente. Seja $O_{ij} \in \{0,1\}$ a variável para o resultado observado para a interação P_i e P_j (O_{ij} = 1 se e somente se a interação é observada), então temos:

$$fp = \Pr(\, O_{ij} = 1 \mid P_{ij} = 0)$$

$$fn = \Pr(\, O_{ij} = 0 \mid P_{ij} = 1)$$

Em seguida, $\Pr(\, O_{ij})$ é dada pela expressão da probabilidade absoluta (vide [25])

$$\Pr(\, O_{ij} = 1) = \Pr(\, O_{ij} = 1,\, P_{ij} = 1) + \Pr(\, O_{ij} = 1,\, P_{ij} = 0),$$

$$= \Pr(\, P_{ij} = 1).(1 - fn) + (1 - \Pr(\, P_{ij} = 1)).fp$$

Definiram, ainda, a função de verossimilhança (probabilidade de observar todo os dados de interação do proteoma) por

$$L = \prod(\Pr\,(O_{ij} = 1))^{O_{ij}}\, .(1 - \Pr\,(O_{ij} = 1))^{1-O_{ij}}.$$

A verossimilhança L é função de (λ_{mn}, fp, fn). Uma vez que é difícil calcular diretamente (λ_{mn}, fp, fn) que maximize L, eles desenvolveram um algoritmo EM com a finalidade de localmente maximizar L, onde fp e fn foram fixados em determinados valores.

Os resultados obtidos pelos métodos EM e AM foram comparados, utilizando como entradas os dados de interação proteína-proteína obtidos pelos experimentos de Uetz e

colaboradores [17] e Ito e colaboradores [18], e mostraram que o método EM foi superior ao Método AM.

3.3 Método da Força de Interação Proteína-Proteína utilizando Programação Linear

Hayashida e colaboradores [26] propuseram um novo método para inferir interações domínio-domínio a partir de dados sobre a "força" das interações proteína-proteína. A maioria dos métodos existentes assume que os dados de interação proteína-proteína são apresentados como dados binários (isto é, se cada par proteína interage, ou não). No entanto, múltiplas experiências biológicas são realizadas para os mesmos pares de proteínas e, assim, a "força", em razão da quantidade de interações observadas para o número de experiências, está disponível para cada par de proteínas. Este método tenta minimizar os erros entre a "força" de interações observadas e as probabilidades preditas com os dados de treinamento.

O problema foi formulado como programação linear (PL) com base no modelo probabilístico da interação proteína-proteína (equação 2.1), e como proposto por Deng e colaboradores [16]. Apesar do uso deste modelo probabilístico, o método proposto é completamente diferente do método de Deng e colaboradores [16] que usa um algoritmo EM, e assume dados binários {0,1} como entrada. Já o método proposto que usa programação linear pressupõe dados de interação numéricos, com valores entre zero e um, como entrada. O método proposto tem outra vantagem: vários tipos de restrições podem ser facilmente, inseridas e, portanto, é fácil de combinar o método com outros métodos. O método foi comparado com o método de associação (AM), com método EM,

e com o método baseado em Máquina de Vetores de Suporte (SVM - *Support Vector Machine*) utilizando dados reais de interação proteína-proteína.

3.3.1 Método baseado em PL para dados de interação binários

Descrevemos inicialmente a versão básica do método proposto baseado em Programação Linear (denominado Método baseado em programação linear para dados de interação binários (LPBN - *Linear programming based method for binary interaction data*)).

Utilizando o modelo probabilístico para o método EM e um limiar Θ, podemos predizer interações proteína-proteína através da seguinte fórmula

$$P_i \; and \; P_j \; interagem \; <=> 1 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) \geq \Theta.$$

Esta condição pode ser transformada como segue

$$1 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) \geq \Theta,$$

$$\prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) \leq 1 - \Theta,$$

$$\ln \left(\prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) \right) \leq \ln (1 - \Theta),$$

$$\sum_{D_{mn} \in P_{ij}} \ln(1 - \lambda_{mn}) \leq \ln (1 - \Theta).$$

Onde "ln" denota o logaritmo natural. Seja $\gamma_{mn} = \ln(1 - \lambda_{mn})$, e $\beta = \ln (1 - \Theta)$. Então a condição acima pode ser escrita como:

$$\sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \beta.$$

Isto é uma inequação linear. Portanto, se pudermos encontrar $\gamma_{mn,}$ $(\gamma_{mn} \leq 0)$, satisfazendo

$$O_{ij} = 1 <=> \sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \beta.$$

(O_{ij} representa a variável para o resultado de interação observada entre P_i e P_j). Daí, para todos os dados observados O_{ij}, (isto é, os dados de treinamento), pode-se obter parâmetros consistentes com todos os dados de treinamento.

Embora seja bastante difícil de minimizar o número de restrições não satisfeitas, é possível minimizar a soma das distâncias [27]. Por isso, usamos a seguinte programação linear:

$$Minimizar \sum_{P_{ij}} \xi_{ij}$$

Sujeito à

$$\sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \beta - const + \xi_{ij}$$

Para $P_{ij} tal\ que\ O_{ij} = 1$,

$$\sum_{D_{mn} \in P_{ij}} \gamma_{mn} > \beta + const - \xi_{ij}$$

Para $P_{ij} tal\ que\ O_{ij} = 0$,

$\gamma_{mn} \leq 0\ para\ todo\ \gamma_{mn}$,

$\xi_{mn} \geq 0\ para\ todo\ \xi_{mn}$,

$\beta < 0$,

Onde *const* é uma pequena constante escolhida convenientemente (utilizou-se *const* = 0,01). Uma vez que γ_{mn} e β foram determinadas pode-se obter λ_{mn} e Θ por $\lambda_{mn} = 1 - \exp(\gamma_{mn})$ e $\Theta = 1 - \exp(\beta)$, respectivamente.

3.3.2 Método baseado em PL para dados de interação numéricos

Aqui, descrevemos o método baseado em programação linear para os dados de interação numéricos (denominado LPNM - *Linear programming based method for*

numeric interaction data), que é a variante mais importante do método baseado em programação linear.

No LPNM, usamos um limite Θ para predizer as interações proteína-proteína. Por outro lado, em LPNM, estabelecemos Θ_{ij} como a razão entre as interações entre proteínas P_i e P_j em uma série de experiências, isto é,

$$\Theta_{ij} = \frac{N(O_{ij})}{Z},$$

onde $N(O_{ij})$ é o número de vezes que uma interação entre proteínas P_i e P_j é observada nas experiências, e Z é o número total de experiências.

Uma vez que, Θ_{ij} é a razão entre as interações entre P_i e P_j, consideramos minimizar a diferença entre $\Pr(P_{ij} = 1)$ e Θ_{ij}, em outras palavras, entre a probabilidade de observar um interação no modelo probabilístico acima e a razão das interações observadas nas experiências.

Quando $\Pr(P_{ij} = 1)$ e Θ_{ij} são equivalentes, podemos afirmar:

$$\sum_{D_{mn} \in P_{ij}} \ln(1 - \lambda_{mn}) = \ln(1 - \Theta_{ij})$$

A partir da equação acima, temos a seguinte equação linear

$$\sum_{D_{mn} \in P_{ij}} \gamma_{mn} = \beta_{ij}$$

Para todos P_{ij} definindo-se $\gamma_{mn} = \ln(1 - \lambda_{mn})$ e $\beta_{ij} = \ln(1 - \Theta_{ij})$. Se tivermos γ_{mn} para todos m e n satisfazendo as equações acima, podemos obter parâmetros consistentes para as interações domínio-domínio com o conjunto de dados numérico.

No entanto, estas equações nem sempre se mantém. É razoável, portanto, tentar minimizar a soma das diferenças $\sum_{P_{ij}} |\sum_{D_{mn} \in P_{ij}} \gamma_{mn} = \beta_{ij}|$. Portanto, utilizamos a seguinte programação linear para minimizar a diferença:

$$Minimizar \sum_{P_{ij}} \alpha_{ij}$$

Sujeito à

$$\sum_{D_{mn} \in P_{ij}} \gamma_{mn} - \beta_{ij} \leq \alpha_{ij}$$

$$\beta_{ij} - \sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \alpha_{ij}$$

$\gamma_{mn} \leq 0 \; para \; todo \; \gamma_{mn}$,

$\alpha_{ij} \geq 0 \; para \; todo \; \alpha_{ij}$,

$\beta_{ij} < 0$,

Foi mostrado que o método é comparável aos métodos existentes quando aplicados a dados binários e mostrou-se melhor que os outros métodos comparados quando aplicada aos dados numéricos (isto é, dados de força de interação).

3.4 Método de explicação parcimoniosa

Guimarães e colaboradores [28] estabeleceram a hipótese de que as interações entre proteínas evoluíram de forma parcimoniosa e que o conjunto correto de interações domínio-domínio é bem aproximado por um conjunto mínimo de interações de domínio necessário para justificar uma determinada rede de interação proteína-proteína.

A esta abordagem nos referimos como o método de "Explicação Parcimoniosa" (PE - *Parsimonious Explanation*). O PE foi formulado como um problema de otimização de programação linear, onde cada potencial contato domínio-domínio é uma variável que pode receber um valor chamado "*Score*" de programação linear (LP-score), variando entre

0 e 1, e cada par da rede de interação proteína-proteína corresponde a uma restrição linear.

Esta formulação permite uma nova maneira de lidar com ruídos nos dados de interação da proteína. Ou seja, foi construído um conjunto de instâncias de programação linear aleatoriamente, em que a probabilidade de incluir uma restrição de LP é igual à probabilidade com a qual a correspondente interação proteína-proteína é assumida como correta, e média dos resultados para obter o LP-*score* para cada par.

Para controlar possíveis excessos de predições de interações entre pares de domínio freqüentes, foi atribuída uma pontuação para o *testemunho* versus *promiscuidade* (pw-*score*), para cada interação domínio-domínio predita. O pw-*score* deriva de duas observações, medindo a confiança na predição.

Primeiro, as interações domínio-domínio que têm muitos testemunhos (pares que interagem através de interação de proteínas de domínio único) são mais prováveis de serem corretas do que aquelas têm pouco ou nenhum testemunho.

Em segundo lugar, existem interações domínio-domínio que têm pontuação alta, devido à elevada freqüência de seu aparecimento (denominadas "promíscuas") e não relacionadas com a topologia específica da rede de interação proteína-proteína. Tendo em conta estas observações, a formulação do pw-*score* recompensa as interações de domínio que têm muitos testemunhos e penaliza interações promíscuas.

Formalmente, dada uma rede de interação proteína-proteína I = (P, E), onde P = P_1, P_2,..., P_n é o conjunto de proteínas na rede e E é o conjunto de interações protéicas, e um conjunto de pares não ordenados denotando todas as possíveis interações de domínio do domínio D = {{i, j} | i ∈ P_n, j ∈ P_m, e P_n e P_m interagem}, resolver o programa linear (LP):

$$Minimizar \sum_{\{ij\} \epsilon D} x_{ij},$$

$$Sujeito \text{ à: } \sum_{i \epsilon P_m,\ j \epsilon P_n} x_{ij} \geq 1,$$

Para todos os pares de proteínas interagindo (P_m,P_n)

O método, abreviado como PE, é baseado no princípio de parcimônia: parceiros de domínio de domínio de interação são previstos, identificando o conjunto mínimo ponderada dos pares de domínio que possam apresentar uma rede de interação dada proteína-proteína. O problema de otimização correspondente é formulado usando programação linear. Os resultados obtidos mostraram que o método de PE supera, consideravelmente, os métodos AM e EM. Foi calculado o valor preditivo positivo (precisão) e a sensibilidade (recall) do método em 75,3% e 76,9%, respectivamente.

O conjunto de teste para este experimento faz a suposição de que as interações domínio-domínio que foram provados para mediar uma interação proteína-proteína específicos também são susceptíveis de mediar outras interações protéicas que contêm os pares de domínio.

Neste caso, é razoável supor que não pares de domínio no conjunto padrão de ouro não interagem no contexto dos pares dada proteína. No entanto, pode haver casos em que não é verdade, portanto, os números divulgados devem ser considerados como estimativas.

O método fornece uma maneira única para representar as incertezas das interações proteína-proteína em rede de interação proteína-proteína de em alto rendimento. O trabalho assume que a probabilidade de erro para cada interação proteína-proteína

representados na rede é o mesmo. No entanto, a abordagem pode também ser aplicada quando a probabilidade de correção de cada interação é avaliada individualmente, com base no tipo de experiência utilizado para a sua detecção outras informações suplementares. Por exemplo, os valores de confiança, com base na regressão logística. O método de PE é um afastamento significativo do pressuposto subjacente do método EM. Enquanto o método EM funciona bem para o problema da identificação de pares de proteínas que interagem com base na sua composição de domínios, ele não fornece uma abordagem eficaz para a detecção de interação entre os domínios.

Foi mostrado que o método de PE tem desempenho significativamente melhor do que os outros métodos anteriores. Estes resultados fornecem um argumento por trás da correção do princípio da parcimônia na detecção de domínio do domínio de interações baseadas na topologia da rede de interação proteína-proteína.

3.5 Método da Cobertura de Conjuntos de Máxima Especificidade

Huang e colaboradores [19] propuseram um novo método para predição de interações de proteínas a partir das complexas interações entre proteínas e suas estruturas de domínios o qual denominaram procedimento de Cobertura de Conjuntos de Máxima Especificidade (MSSC - *Maximum Specificity Set Cover*).

Nota importante: Huang e colaboradores [19] definem a especificidade como (|P∩T|/|P|) e esta definição equivale estatisticamente ao valor preditivo positivo (valor preditivo positivo = verdadeiros positivos / verdadeiros positivos + falsos positivos). A "máxima especificidade" decorre de encontrar uma cobertura de conjuntos com o menor

número de falsos positivos. Com a redução máxima de falsos positivos, se obtêm uma "especificidade máxima" (|P∩T|/|P|) ou o valor preditivo positivo máximo. No entanto, ao reduzir os falsos positivos a especificidade estatística (verdadeiros negativos / verdadeiros negativos + falso positivos) também se reduz, independentemente de desconhecermos o total de verdadeiros negativos.

Huang e colaboradores [19] desenvolveram um algoritmo guloso que utiliza uma abordagem de cobertura de conjuntos, outro bem conhecido problema NP - difícil de otimização combinatória, cuja importância decorre da sua grande aplicabilidade em problemas do mundo Real [29].

O Problema da Interação Proteína-Proteína é transformado em um Problema de Cobertura de Conjuntos Generalizado. Ao se resolver este, se obtém uma resposta para o problema original de interação proteína-proteína. Basicamente, a idéia é associar pares de domínios a conjunto de arestas de um grafo, cujas extremidades (ou vértices) representam, respectivamente, pares de proteínas contendo estes domínios. O critério de avaliação das soluções presente na função objetivo será descrito mais adiante. Desta forma, trata-se de encontrar um conjunto de pares de domínios para representar as interações proteína-proteína que foram fornecidas. Idealmente, o conjunto de pares de domínios deve conter, tanto quanto possível, poucos falsos positivos. Falsos positivos são interações proteína-proteína preditas, mas não incluídas na rede de interação de entrada. O algoritmo guloso descrito em [19] escolhe pares de domínios para cobrir, satisfazendo ao critério de especificidade descrito mais a adiante, as interações fornecidas de proteínas. É dito que um par de domínios cobre uma interação proteína-proteína se a duas proteínas que interagem contém os dois domínios, respectivamente.

Apresentamos o Problema da Cobertura de Conjuntos clássico e o Problema da Cobertura de Conjuntos Generalizado, e em seguida, a descrição da transformação do Problema de Interação de Proteínas (PIP) em um Problema de Cobertura de Conjuntos Generalizado(PCCG).

3.5.1 Critério de Avaliação de Qualidade

Formalmente, Huang e colaboradores [19] definem a especificidade como a razão entre a quantidade de interações que tem correspondência entre as predições P e o conjunto de teste T, em relação à quantidade total de interações preditas em P,

Especificidade = |P ∩ T| / |P|.

Note que o critério de especificidade utilizado em Huang e colaboradores [19] difere daquele discutido na Seção 2.3 e corresponde ao valor preditivo positivo.

Por sua vez, a sensibilidade é definida como a razão entre a quantidade de interações que tem correspondência entre as predições P e o conjunto de teste T em relação a quantidade de interações observadas no conjunto de teste T,

Sensibilidade = |P ∩ T| / |T|.

Assim, essas métricas são dependentes da escolha do conjunto de teste, bem como do limiar do "*score*" de predição.

3.5.2 Problema da Cobertura de Conjuntos (PCC) clássico

O PCC pode ser formulado como se segue.

Seja X um conjunto finito e $\mathcal{F}$ é uma família de todos os subconjuntos de X, e seja $\mathcal{F}$ = {Si, 1≤ i ≤ t} para t>0, uma família de subconjuntos de X que podem cobrir X, ou seja, $X = \bigcup_{S \in \mathcal{F}} S$, como mostrado na Figura 4, a seguir.

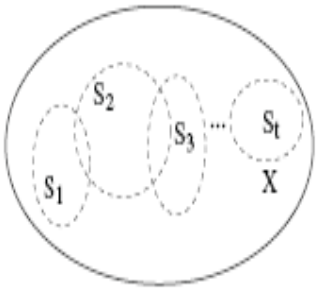


Figura 4: Esquematização do problema da cobertura de conjuntos:
X é o conjunto a ser coberto e $\mathcal{F}$ = {S_i, 1 ≤ i ≤ t} é a família de subconjuntos de X que pode cobrir X.
(Extraído de [19])

O problema da cobertura de conjuntos (PCC) é encontrar um subconjunto C de $\mathcal{F}$ para cobrir X, tal que

$$X = \bigcup_{S \in C} S \tag{3.3}$$

onde C necessita satisfazer determinadas condições de acordo com a especificidade de cada problema. No caso do Problema da Cobertura de Conjuntos Mínima (PCCM) o objetivo é encontrar C com cardinalidade mínima (|C|), enquanto no caso do Problema da Cobertura de Conjuntos Mínima Exata (PCCME) se exige que $\sum_{S \in C} |S|$ seja minimizado.

3.5.3 Problema da Cobertura de Conjuntos Generalizado (PCCG)

Huang.e colaboradores [19] generalizaram o PCC colocando X em um conjunto Y maior (Figura 5).

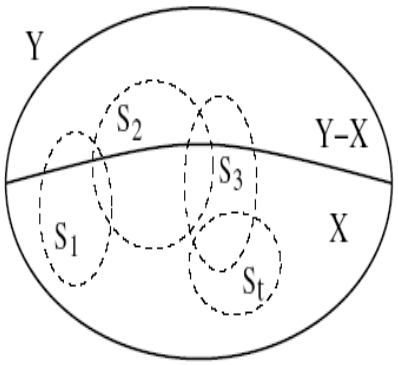


Figura 5: Esquematização do problema da cobertura de conjuntos generalizado:
X é um subconjunto de Y, e $\mathcal{F}$ = {Si, 1 ≤ i ≤ t}, t>0, é uma família de subconjuntos de Y, que pode cobrir X.
(Extraído de [19])

Suponha que Y é um conjunto finito, $X \subseteq Y$, e $\mathcal{F}$ é uma família dos subconjuntos de Y que podem cobrir X, ou seja, $X \subseteq \cup_{s\in\mathcal{F}} S$

O Problema da Cobertura de Conjuntos Generalizado (PCCG) é encontrar um subconjunto C de $\mathcal{F}$ para cobrir X, ou seja:

$$X \subseteq \cup_{s\in C} S \tag{3.4}$$

onde C deve satisfazer determinadas condições de acordo com as especificidades de cada problema.

3.5.4 Transformação do PIP em PCCG

Tanto o Problema da Interação de Proteínas (PIP) [30], quanto o Problema da Cobertura de Conjuntos (PCC) são problemas NP - difíceis. O PCCG é também NP - difícil, uma vez que ele é uma extensão do PCC clássico.

O PIP é resolvido transformando-o em um PCCG da seguinte forma:

Dada uma rede de interações proteína-proteína, obtida experimentalmente, ela pode ser modelada por um grafo G = (P, E), onde P é o conjunto de proteínas e E é o conjunto de arestas. As proteínas são os vértices de G. Existe uma aresta entre duas proteínas se, e somente se, elas interagem.

A Figura 6 mostra um típico par de domínios. O domínio D_1 está contido em 3 proteínas: P_1, P_2, e P_3, e o domínio D_2 esta contida em 4: P_4, P_5, P_6 e P_7. Existem 12 combinações de pares de proteínas entre D_1 e D_2. Alguns deste pares de proteínas interagem (na rede fornecida), e outras não. Note que o par (D_1, D_2) (associado a um subconjunto de $\mathcal{F}$ no PCCG) representa um subgrafo de G.

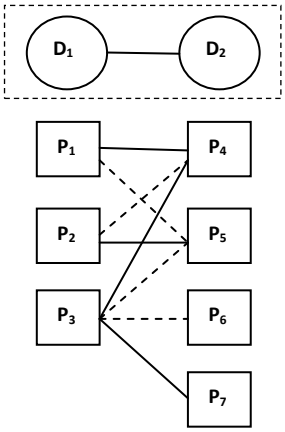


Figura 6: Relacionamento dos pares de proteínas de um par de domínios:
D_1 está contido em 3 proteínas: P_1, P_2, e P_3, e o domínio D_2 está contida em 4: P_4, P_5, P_6 e P_7.
Existem 12 combinações de pares de proteínas entre D_1 e D_2.
Alguns deste pares de proteínas interagem (linhas sólidas), e outras não (linhas tracejadas). Adaptado de [19].

Além disso, considere o conjunto de todos os possíveis pares de proteínas a partir da rede de proteínas fornecida. Na Figura 7, somente os segmentos sólidos representam os pares de proteínas que interagem, e os segmentos tracejados representam aqueles pares de proteínas que não interagem. Os pares de segmentos sólidos (interações) foram, então, colocados juntos formando um grupo.

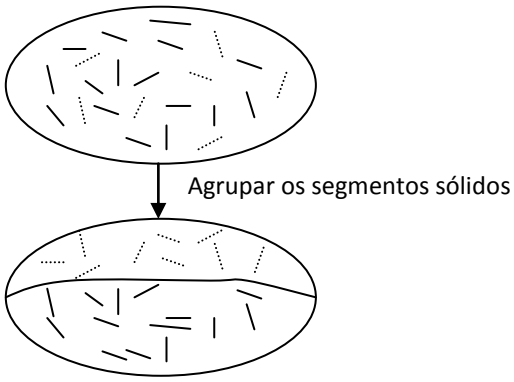


Figura 7: Conjunto de todos os pares de proteínas:
Um segmento sólido representa uma interação de proteína e um segmento tracejado representa um par de proteínas que não interagem. Em seguida, agrupamos os segmentos sólidos (interações)
(Adaptado de [19])

Voltemos a Figura 6, onde simbolicamente, podemos utilizar o par de domínios (D_1, D_2) para representar todos os 12 pares de proteínas da Figura 6, como na Figura 9

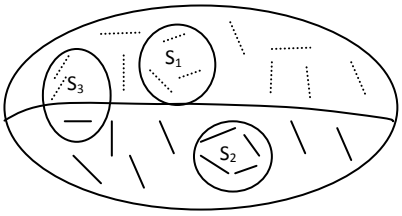


Figura 8: representação dos pares de domínios pelo conjunto par de domínios.
(Adaptado de [19])

O problema de cobertura de conjuntos generalizado é construído a partir da rede de interações de proteínas tendo

Y = {todos os pares de proteínas (P_i, P_j) | P_i, $P_j \in P$}

X = {pares de proteínas (P_i, P_j) | P_i interage com P_j em G}, e

$\mathcal{F}$ sendo o conjunto de todos os pares de domínios (D_m, D_n), onde cada (D_m, D_n) está associado um subconjunto de X, veja a Figura 10. É assumido que cada elemento de X aparece pelo menos uma vez em um dos subconjuntos de $\mathcal{F}$.

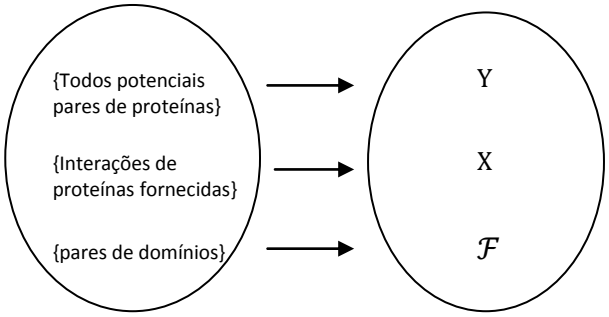


Figura 9: Transformação do problema de interação de proteínas para o PCC:
O grande conjunto Y é considerado como sendo o conjunto de todos os potenciais pares de proteínas, o subconjunto X é considerado como sendo o conjunto das interações de proteínas fornecido e, finalmente, a família $\mathcal{F}$ é considerado como sendo o conjunto de todos os pares de domínios (Adaptado de [19])

Um par de domínios (D_m, D_n) é visto como um subconjunto de Y. Especificamente, se um par de proteínas (P_i, P_j) (um elemento de X) contém (D_m, D_n), então (P_i, P_j) pertence ao subconjunto (D_m, D_n).

Suponha que encontramos um subconjunto C de $\mathcal{F}$ que cubra cada elemento de X. Um elemento em C corresponde a um par de domínios (D_m, D_n). Se (D_m, D_n) cobre (P_i, P_j), então as duas proteínas P_i e P_j contém D_m e D_n, respectivamente; logo (D_m, D_n) pode ser utilizada para representar a interação entre P_i e P_j. Portanto, temos um conjunto de pares de domínios que representa a rede de proteínas G.

Suponha que exista um conjunto D de pares de domínios que represente a rede G. Para cada elemento (P_i, P_j) em X, existe um par de domínios (D_m, D_n) de D que representa a interação ente P_i e P_j. Uma vez que (D_m, D_n) pode ser visto como um elemento de $\mathcal{F}$, a coleção C de todos os pares de domínios de D é um subconjunto de $\mathcal{F}$, e C cobre X.

Nesta transformação, o conjunto de interações proteína-proteína G corresponde ao conjunto X que necessita ser coberto, e um par de domínios corresponde a um elemento de $\mathcal{F}$ (um subconjunto de Y).

3.5.5 Abordagem gulosa para o MSSC

No MSSC estamos interessados em determinar um subconjunto de pares de domínios, de modo que tanto a especificidade (proporção de interações com resultados positivos que são corretamente identificados pelo método) quanto à sensibilidade (capacidade do teste para identificar os resultados positivos) sejam maximizados, assumido que o conjunto de treinamento e teste seja o mesmo. Esta é uma suposição razoável sempre que o conjunto de treinamento seja suficientemente representativo dos conjuntos de testes reais.

O problema do MSSC é encontrar é encontrar um subconjunto C de $\mathcal{F}$ para cobrir X tal que

$$m(C) = \sum_{S \in C} |S - X| \tag{3.7}$$

seja minimizado.

Podemos ver que o MSSC permite a subcobertura C cobrir a sobreposição com X, mas, a sobreposição com Y - X (elementos que pertencem a Y, mas não pertencem a X) é minimizada. Desta forma, *o MSSC escolhe uma cobertura que maximiza a especificidade, porque se considera que os falsos positivos apareçam apenas em Y - X.*

Como definida pelo autor, temos:

Especificidade = $\frac{|P \cap T|}{|P|}$

onde os Verdadeiros Positivos são aqueles elementos que foram preditos e observados e os Falso Positivos são aqueles elementos que foram preditos mas não observados

Daí, quando o MSSC reduz o denominador escolhendo os elementos com a menor quantidade de falso positivo estará aumentando a especificidade.

No algoritmo guloso para MSSC, mostrado abaixo, o conjunto U representa a parte não coberta de X, ε é o subconjunto de $\mathcal{F}$ que ainda não foi escolhido pelo algoritmo.

```
GREEDY_MSSC(Y, X, F)
1       U ← X
2       ε ← F
3       C ← ø
4       while U ≠ ø
5               do selecionar um S ∈ ε com |S−X| / |S∩U| mínimo
                (o desempate é feito por |S ∩ U|)
6               U ← U - S
7               ε ← ε -{ S}
8               C ← C ∪ { S}
9       return C
```

Neste algoritmo, a cada passo quando um subconjunto deve ser escolhido, é escolhido aquele cuja razão entre a parte de fora X e a parte dentro de U é minimizada, como mostrado na Figura 10.

A quantidade de interações do laço "**while**" é limitada por min(|X|, $|\mathcal{F}|$), e cada interação demanda o tempo de |X|.$|\mathcal{F}|$; portanto a complexidade deste algoritmo guloso é O(|X|.$|\mathcal{F}|$ min(|X|,$|\mathcal{F}|$)).

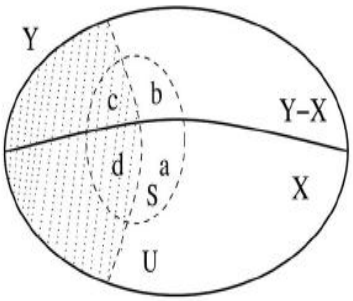


Figura 10: Funcionamento do algoritmo MSSC
A área sombreada já está coberta por C. U é a área de pleno sol no X. O conjunto de candidatos S é dividido em quatro partes: a, b, c, d. O MSSC escolhe um conjunto S, com o mínimo (b + c) / a. O algoritmo para MSSC permite a sobreposição de subcobertura dentro X, na verdade, aumentando a probabilidade de interação de um par de proteínas). (Extraído de [19])

O algoritmo guloso acima é apenas uma aproximação e a solução encontrada por ele tem a seguinte relação com a solução ótima do MSSC:

Teorema 1.

Suponha que Ca é a aproximação das MSSC encontrada pelo algoritmo guloso acima e Co é uma subcobertura ideal para MSSC.

Seja k = $\max_{S \in \mathcal{F}} |S|$.

Se m (Co) = 0, então m (Ca) = 0, caso contrário, temos:

$$\frac{m(Ca)}{m(Co)} \leq [\ln(k-1)+1] \qquad \square$$

A prova do Teorema 1 pode ser encontrada em material suplementar [19].

O teorema mostra a relação entre a aproximação por GREEDY_MSSC e uma solução ótima. Se k é pequeno, a diferença entre eles é pequena demais. Neste teorema, k é o número máximo de elementos que um subconjunto pode ter e que corresponde ao número máximo de pares de proteínas que contém um par de domínios na rede de proteínas.

Quando X = Y, o MSSC é reduzido a uma cobertura de conjuntos de mínima cardinalidade (MSC - *Minimum cardinality set cover*), que é bem conhecido por ser NP - difícil. No caso do MSC, a aproximação logarítmica é a aproximação mais conhecida.

3.5.6 Predição

Uma vez que os pares de domínios sejam escolhidos pelo MSSC, a cada par é atribuída uma probabilidade de interação, como calculada no método AM (Equação (3.1)). Aos pares de domínios não escolhidos é atribuída uma probabilidade de interação igual a zero. A Equação (3.2) é, então, usada para calcular a probabilidade de interação para cada suposto par de proteínas.

CAPÍTULO 4

4 Implementação do GRASP

4.1 Metaheurística GRASP

4.1.1 Introdução a metaheurística GRASP

Consideramos nesta seção um problema de otimização combinatória, definido por um conjunto finito E = {1,..., n}, um conjunto de soluções viáveis $\mathcal{F} \subseteq 2^E$, e uma função objetivo f: $2^E \rightarrow \mathbb{R}$. Na versão de minimização, buscamos uma solução ótima S* $\in \mathcal{F}$ tal que f(S*) ≤ f(S), ∀ S $\in \mathcal{F}$. O conjunto E, a função de custo f, e o conjunto de soluções viáveis $\mathcal{F}$ são definidos para cada problema específico. Por exemplo, no caso do problema do caixeiro viajante, o conjunto E é o conjunto de todas as arestas de ligação entre as cidades a serem visitadas, f(S) é a soma dos custos de todas as arestas e ∈ S, e $\mathcal{F}$ é formado por todos os subconjuntos de arestas que determinam um ciclo hamiltoniano.

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) [29] [31] é uma heurística iterativa gulosa adaptativa, no qual cada iteração consiste em duas fases: *construção* e *busca local*. A fase de construção constrói uma solução viável, cuja vizinhança é investigada até que um mínimo local seja encontrado durante a fase de busca local. A melhor solução global é mantida como o resultado. Um extenso levantamento da literatura é dado em Festa e Resende [32]. O pseudocódigo na Figura 11 ilustra os blocos principais de um procedimento GRASP para minimização, em que

Max_Iterations representa o total de iterações a serem realizadas e *Seed* é usada como a semente inicial para o gerador de números pseudo-aleatórios.

```
procedure GRASP(Max_Iterations, Seed)
1   Read_Input();
2   for k = 1,...,Max_Iterations do
3           Solution ← Greedy_Randomized_Construction(Seed);
4           Solution ← Local_Search(Solution);
5           Update_Solution(Solution, Best_Solution);
6   end;
7   return Best_Solution;
end GRASP.
```

Figura 11 Pseudocódigo da metaheurística GRASP.
Extraído de [33]

A Figura 12 ilustra a fase de construção com o seu pseudocódigo. Em cada iteração desta fase, seja o conjunto de elementos candidatos formado por todos os elementos do conjunto E que podem ser incorporados na solução parcial em construção, sem impedir a construção de uma solução viável com os elementos restantes do conjunto E. A seleção do próximo elemento a ser incorporado é determinada pela avaliação de todos os elementos candidatos de acordo com uma função de avaliação gulosa. Esta função gulosa geralmente representa o aumento incremental na função de custo devido à incorporação deste elemento na solução parcial em construção. A avaliação dos elementos por esta função leva à criação de uma Lista de Candidatos Restrita (LCR); em inglês *Restrict Candidate List* (RCL), formada pelos melhores elementos, isto é, aqueles cuja incorporação na solução parcial atual resulta em menores custos incrementais (este é o aspecto guloso do algoritmo). O elemento a ser incorporado na solução parcial é aleatoriamente escolhido dentre os elementos da LCR (este é o aspecto probabilístico da heurística). Uma vez que o elemento selecionado é incorporado à solução parcial, a lista de candidatos é atualizada e os custos incrementais são reavaliados (este é o aspecto adaptativo da heurística). Esta estratégia é semelhante à heurística *semi-greedy* proposto

por Hart e Shogan [34], que é também uma abordagem multi-partida baseada em construção gulosa aleatória, mas sem busca local.

As soluções geradas por uma construção gulosa aleatória não são necessariamente ótimas. A fase de busca local geralmente melhora a solução construída. Um algoritmo de busca local funciona de maneira iterativa através de sucessivas substituições da solução atual por uma solução melhor encontrada na vizinhança da solução atual. Ela termina quando nenhuma solução melhor é encontrada na vizinhança. O pseudocódigo de um algoritmo de busca local básico começa a partir da solução construída na primeira fase (Solution) e da utilização de uma vizinhança N(Solution), como mostrado na Figura 13.

```
procedure Greedy_Randomized_Construction(Seed)
1   Solution ← ∅;
2   Evaluate the incremental costs of the candidate elements;
3   while Solution is not a complete solution do
4           Build the restricted candidate list (RCL);
5           Select an element s from the RCL at random;
6           Solution ← Solution ∪ {s};
7           Reevaluate the incremental costs;
8   end;
9   return Solution;
end Greedy_Randomized_Construction.
```

Figura 12 Pseudocódigo da fase de construção.
Extraído de [33]

```
procedure Local_Search(Solution)
1   while Solution is not locally optimal do
2           Find s' ∈ N(Solution) with f(s') < f(Solution);
3           Solution ← s';
4   end;
5   return Solution;
end Local_Search.
```

Figura 13 Pseudocódigo da fase de busca local.
Extraído de [33]

A eficácia de um procedimento de busca local depende de vários aspectos, tais como a estrutura de vizinhança, a técnica de busca na vizinhança, a avaliação rápida da função

de custo dos vizinhos, e da solução de partida em si. A fase de construção desempenha um papel muito importante em relação a este último aspecto, a construção de soluções de partida de alta qualidade para a busca local. Vizinhanças simples são geralmente utilizadas. A busca na vizinhança pode ser implementada utilizando tanto uma estratégia de *best-improving* como uma estratégia de *first-improving*. No caso da estratégia de *best-improving*, todos os vizinhos são investigados e a solução atual é substituída pelo melhor vizinho. No caso da estratégia *first-improving*, a solução atual se move para o primeiro vizinho cujo valor da função de custo seja menor do que aquele da solução atual. Na prática, se observou, em várias aplicações distintas, que ambas as estratégias levam, com muita freqüência, a mesma solução final, mas com tempos menores de computação quando a estratégia *first-improving* é usada. Também se observou que uma prematura convergência a um mínimo local é mais provável de ocorrer com a estratégia *best-improving* [29].

4.1.2 Construção da Lista de Candidatos Restrita (LCR).

Uma característica especialmente interessante do GRASP é a facilidade com que pode ser implementado. Portanto, o desenvolvimento pode se concentrar na implementação de estruturas de dados eficientes para garantir iterações rápidas. O GRASP tem dois parâmetros principais: um relacionado com o critério de parada e outro com a qualidade dos elementos na lista de candidatos restrita.

Um critério de parada no pseudocódigo descrito na Figura 11 é determinado pela quantidade de iterações (Max_Iterations). Embora a probabilidade de encontrar uma nova solução melhor que a solução atual se reduza com a diminuição no número de iterações, a melhoria da qualidade da solução somente poderá ser encontrada através de

iterações. Uma vez que o tempo de computação não varia muito de iteração para iteração, o tempo de computação total é previsível e aumenta linearmente com o número de iterações. Por conseguinte, quanto maior o número de iterações, maior será o tempo de computação e melhor será a solução encontrada.

Para a construção da LCR utilizada na primeira fase, consideremos, sem perda de generalidade, um problema de minimização como o formulado na Secção 1.1. Denotemos por c(e) o custo adicional associado com a incorporação do elemento e ∈ E na solução em construção. Em qualquer iteração GRASP, tomemos c^{min} e c^{max} como, respectivamente, o menor custo e o maior custo incrementais

A lista de candidatos restrita (LCR) é composta de elementos e ∈ E com os melhores custos incrementais c(e) (ou seja, os menores). Esta lista pode ser limitada tanto pelo número de elementos (baseada em cardinalidade) ou pela sua qualidade (baseada em valor). No primeiro caso, é constituída pelos *p* elementos com os melhores custos incrementais. Nesta seção, a LCR está associada com um parâmetro de limiar α ∈ [0,1]. A lista de candidatos restrita é formada por todas os elementos e ∈ E, que podem ser inseridos na solução parcial em construção, sem destruir a viabilidade e cuja qualidade é superior ao valor de limiar, ou seja, $c(e) \in [c^{min}, c^{min} + \alpha (c^{max} - c^{min})]$. O caso de α = 0 corresponde a um algoritmo puramente guloso, enquanto. α = 1 é equivalente a uma construção aleatória. O pseudocódigo na Figura 14 é um refinamento do pseudocódigo da construção gulosa aleatória mostrada na Figura 12. Isto mostra que o parâmetro α controla a proporção entre o aspecto guloso e a aleatoriedade presentes no algoritmo.

```
procedure Greedy_Randomized_Construction(α, Seed)
1   Solution ← ∅;
2   Initialize the candidate set: C ← E;
3   Evaluate the incremental cost c(e) for all e ∈ C;
4   while C ≠ ∅ do
5           c^min ← min{ c(e) | e ∈ C};
6           c^max ← max{ c(e) | e ∈ C};
7           RCL ← { e ∈ C | c(e) ≤ c^min + α (c^max - c^min)};
8           Select an element s from the RCL at random;
9           Solution ← Solution ∪ {s};
10          Update the candidate set C;
11          Reevaluate the incremental costs c(e) for all e ∈ C;
12  end;
13  return Solution;
end Greedy_Randomized_Construction.
```

Figura 14 Pseudocódigo refinado da fase de construção.
Extraído de [33]

4.2 Método GRASP aplicado ao MSSC

A metaheurística GRASP implementada permite a obtenção de um subconjunto C de pares de domínios de $\mathcal{F}$ que cobre o conjunto de treinamento de pares de proteínas de X que interagem, com máxima especificidade, uma vez que se considera que os falsos positivos apareçam em Y - X, onde Y é o conjunto de todos os pares de proteínas (vide seção 3.5).

Como discutido anteriormente, cada iteração do algoritmo GRASP contém duas etapas: construção e busca local [31]. Na etapa de construção, uma solução viável é construída usando um algoritmo guloso randomizado. Na etapa seguinte uma heurística de busca local é aplicada na solução construída. Este esquema é repetido até um determinado número de iterações e a melhor solução encontrada é mantida como o resultado final.

Esta abordagem é eficiente se a heurística construtiva garante uma amostragem de diferentes regiões do espaço de busca, o que faz com que diferentes buscas locais gerem ótimos locais de "boa" qualidade. Este procedimento é implementado utilizando-se uma lista com os melhores candidatos a cada etapa.

As etapas do projeto GRASP são os procedimentos para a construção e de busca local:

• **Construção**: Na heurística construtiva, a cada iteração os elementos que podem ser incluídos na solução parcial são selecionados aleatoriamente de uma *Lista de Candidatos Restrita* (LCR). Na construção da LCR, o custo de seleção de um subconjunto de X ∈ $\mathcal{F}$ é dada pela razão entre a quantidade de elementos que estão fora de X e a quantidade de elementos comuns ao conjunto ainda não coberto U, ou seja, $\frac{|S-X|}{|S\cap U|}$. No caso em que elementos tenham o mesmo custo de seleção, a escolha é determinada pelo elemento com maior $|S\cap U|$.

A Figura 15 apresenta Pseudocódigo do algoritmo Construtivo. Nas linhas 1, 2 e 3, a cobertura C é inicializada com o conjunto vazio, U é inicializado com todos os elementos de X, e F é inicializado com todos os elementos de $\mathcal{F}$. A linha 4 descreve a estrutura de repetição que é executada até que a cobertura C esteja completa. Nas linhas 5 a 7, o custo incremental (Eval) de cada s é calculado, o Limite é estabelecido como o [min ($Eval_i$) + α (max($Eval_i$) - min($Eval_i$))], e a LCR é construída considerando os elementos cujos custos estão sejam menores ou iguais ao limite determinado. Na linha 8, um elemento da LCR é aleatoriamente selecionado. Uma vez que um elemento é incorporado na solução parcial, o elemento é retirado de F, e a parte não coberta (U) é reduzida dos elementos deste s incorporado (linhas 9-13). Após a inserção de um elemento na solução parcial é possível que outro elemento tenha a sua utilidade nulificada ($|S\cap U|=0$). Nas linhas 14-16, estes elementos são removidos, sem prejuízo na viabilidade da solução.

```
procedimento Construcao ( )
1       C ← ∅;
2       U ← X;
3       F ← 𝓕; //inicializa a lista de candidatos
4       enquanto (C não é cobertura ) faça
5               Eval_i ← |s \ X| / |s ∩ U| , ∀ s_i ∈ F
6               Limite ← min ( Eval_i ) + α (max( Eval_i ) - min(Eval_i ))
7               LCR ← { s_i ∈ F | Eval_i ≤ Limite};
8               s ← selecionar aleatoriamente de LCR;
9               se C ∪ s ∈ 𝓕 então
10                      C ← C ∪ s;
11              fim-se;
12              F ← F \ s;
13              U ← U - s
14              enquanto (existir redundantes) faça
15                      Remove redundantes;
16              fim-enquanto;
17      fim-enquanto;
18      retorna C;
fim Construcao.
```

Figura 15: Pseudocódigo do algoritmo Construtivo proposto

Utilizando uma instância de Bader (80% do conjunto para treinamento), avaliamos a influência do parâmetro α no tamanho da LCR da fase construtiva, e não identificamos uma variação importante no tempo da execução do GRASP. No entanto, este estudo nos auxiliou na fixação de alfa igual a 70%, uma vez que este valor proporciona um menor valor da função objetiva obtido após 100 iterações, como podemos observar na figura 16.

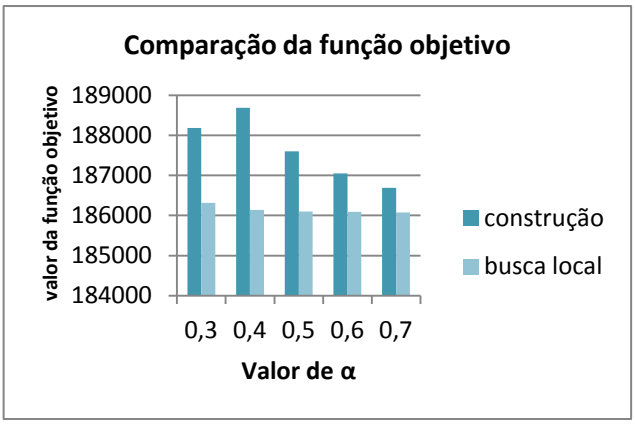


Figura 16: influência do valor de alfa no valor da função objetivo

Esta construção é similar ao algoritmo guloso, mas a cada etapa i, ao invés de selecionar a melhor entre todas m-i+1 opções (m corresponde ao tamanho de $\mathcal{F}$), escolhe aleatoriamente a partir de [α (m-i+1)] melhores opções, onde 0 < α ≤ 1 é o parâmetro de entrada. No nosso trabalho utilizamos α fixo para todas as iterações. Este algoritmo toma o tempo O(pmn), onde p corresponde ao tamanho da cobertura C, e n ao tamanho de X.

• **Busca Local**: Uma vez que não se pode garantir que as soluções encontradas pelo processo construtivo sejam ótimos locais, é necessário realizar uma etapa de busca local para que a solução construída possa ser melhorada. Tradicionalmente, um simples algoritmo de busca local é aplicado. No entanto, metaheurísticas como busca tabu, "*simulated annealing*", entre outras, podem ser utilizadas para busca local do GRASP.

O procedimento de Busca Local analisa a vizinhança da solução gerada pela fase construtiva em busca de alguma solução que, de acordo com a função objetivo estabelecida para o problema, apresente um resultado de qualidade superior ao obtido na fase anterior. A escolha da melhor solução da vizinhança pode ser realizada segundo diferentes estratégias. Em modo exaustivo, todas as soluções da vizinhança são analisadas em busca da que apresente a maior qualidade dentre elas. Uma outra maneira é escolher a primeira solução que tenha um custo melhor que a solução corrente sem analisar todas as outras possíveis soluções.

A Figura 17 apresenta o pseudocódigo do algoritmo de Busca Local. Na linha 1, selecionamos elementos S_i da cobertura C obtida na fase construtiva, cujo custo de seleção seja maior que zero ($|S_i - X| > 0$). Na linha 2, selecionamos elementos fora da cobertura C que possam substituir o elemento selecionado na linha 1. Nas linhas 3 a 9, avaliamos através da função objetivo se a substituição é vantajosa, e em caso afirmativo a solução C é atualizada.

Este procedimento de busca local toma o tempo O(p.(m-p)).

```
Procedimento Busca_Local (C, Best_C) {Vizinhança N(1,1)}
1       Para (cada S_i ∈ C | |S_i − X| > 0) faça
2               Para (cada S_k ∈ F \ C) faça
3                       C` ← (C \ { S_i }) U S_k;
4                       se C` é cobertura então
5                               Calcula m(C`) = ∑_{S ∈ C`} |S − X|;
6                               se m(C`) < m(C) então
7                                       C ← C`
8                               fim-se
9                       fim-se
10              fim-para
11      fim-para
12      Best_C ← C
13      retorna (Best_C)
fim Busca_Local
```

Figura 17: Pseudocódigo do algoritmo Busca_Local proposto

CAPÍTULO 5

5. Resultados Computacionais

Os algoritmos foram desenvolvidos utilizando a linguagem de programação Python versão 2.6 em ambiente Linux. Os testes foram efetuados em um computador Intel® Core I5™ com 4 GB de memória RAM.

- Alfa : 70% (referente ao tamanho da LRC na fase de construção).
- Critério de Parada do GRASP: Número Máximo de Iterações : 100.

Os dois conjuntos de dados da Tabela 2 (I, II) foram divididos em dois conjuntos disjuntos de treinamento e teste, usando uma razão de 80 por cento e 20 por cento, respectivamente. As figuras deste capítulo foram produzidas a partir destes dados. O detalhamento das bases de dados encontra-se no Anexo. Vamos nos referir a estes conjuntos de dados daqui em diante pelo número atribuído a cada uma dos conjuntos.

Tabela 3 Conjuntos de dados utilizados para aferir resultados.
Os conjuntos foram obtidos do material suplementar de Huang e colegas [19].

Número	Conjunto	Qt. interações	Qt. proteínas	Qt. domínios
I	Bader e colegas	6800	2003	1234
II	Interações físicas	1805	724	514

5.1 Comparações de Desempenho

Nós comparamos a capacidade do GRASP para predizer as interações proteína-proteína contra AM e MSSC utilizando o conjunto de dados I e II segundo o esquema geral

de predição apresentado na Figura 18 (veja, também, o anexo para o detalhamento das bases de dados de origem).

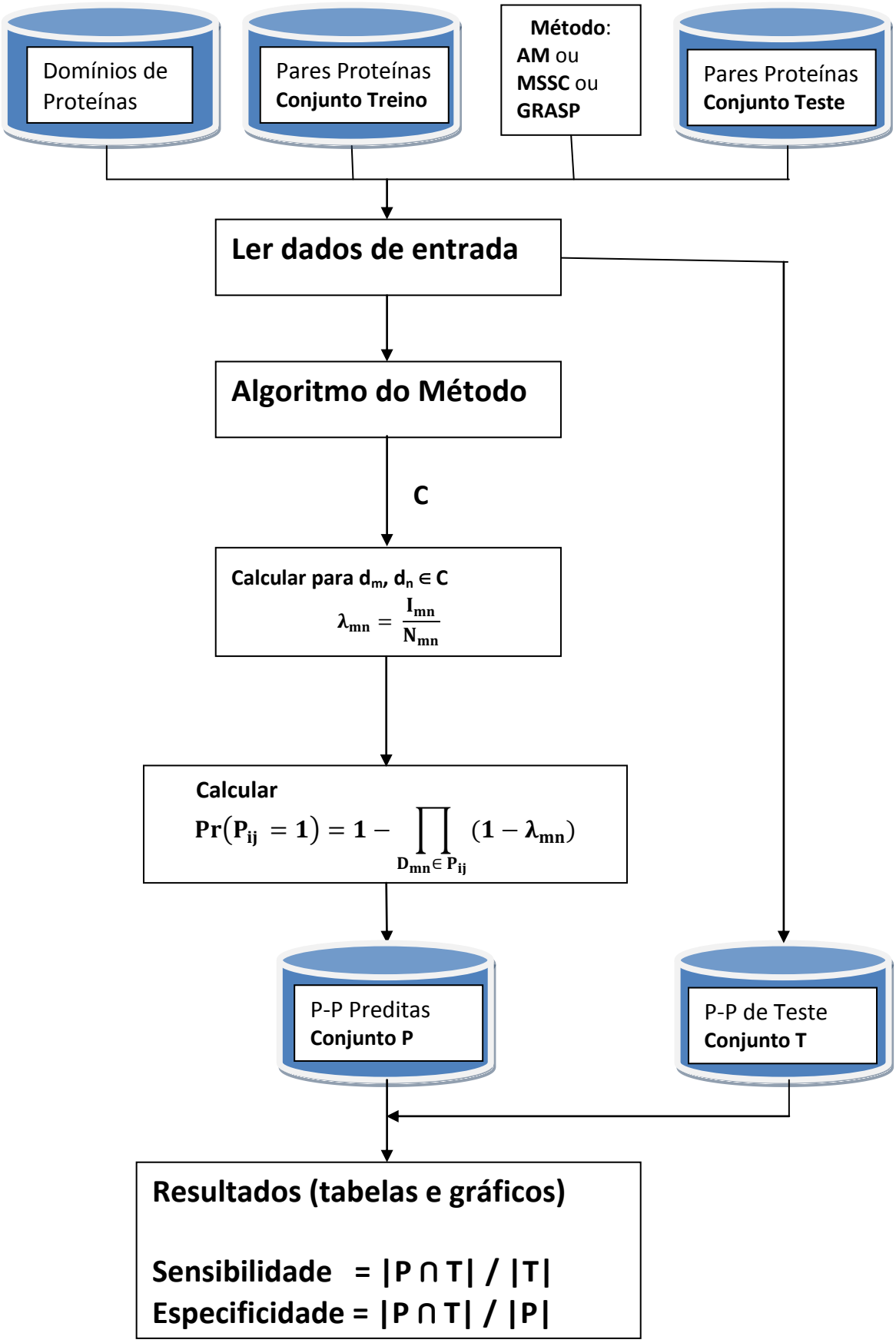
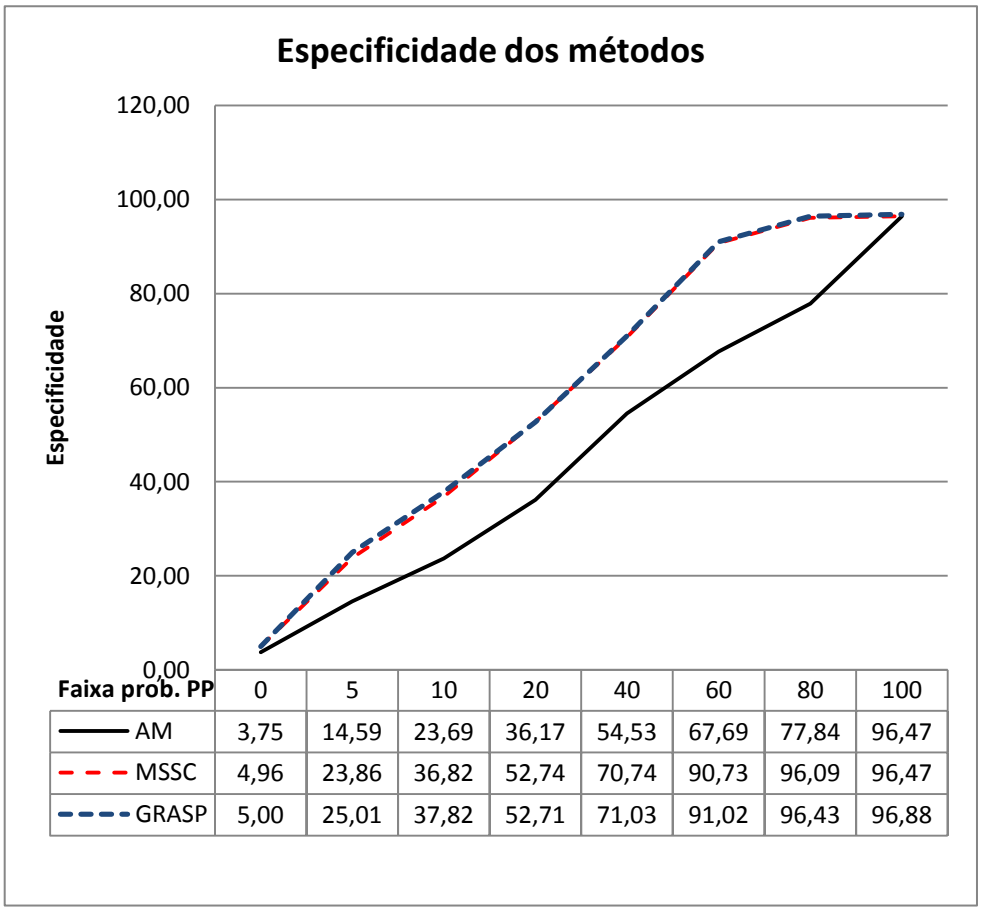


Figura 18: Fluxograma Geral do Processo de Predição de Proteínas

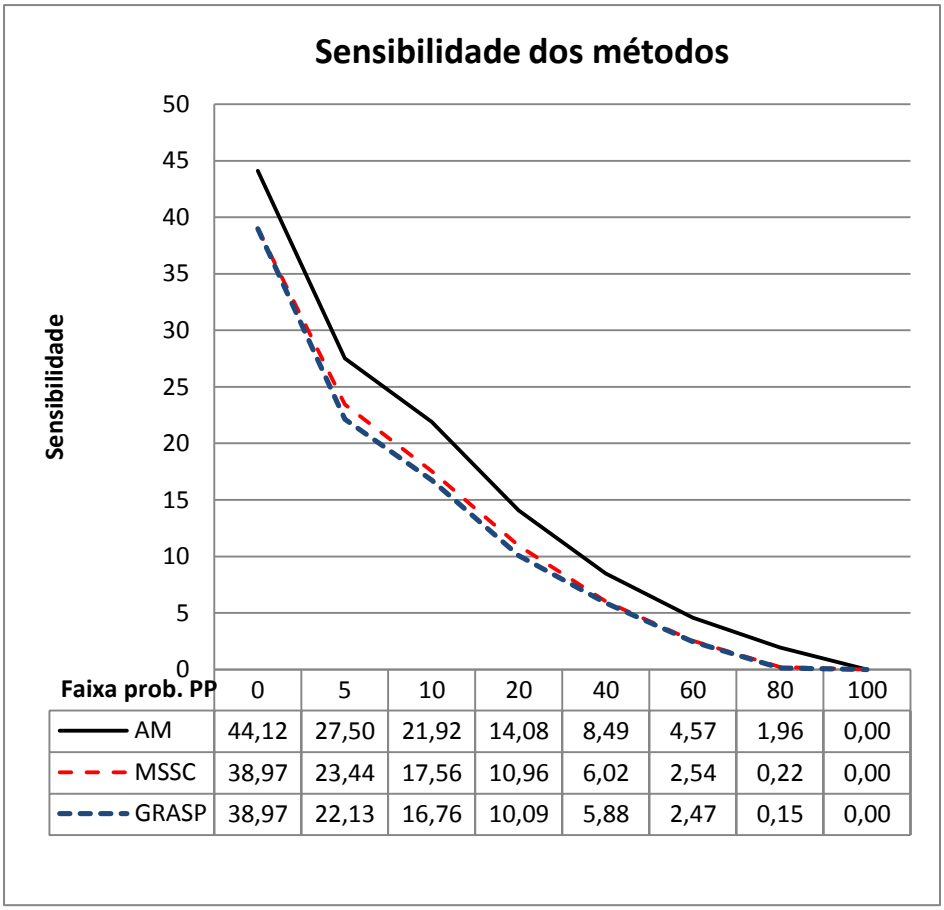
Utilizamos o conjunto de treinamento das interações de proteínas para determinar um conjunto de potenciais interações de domínio, as quais permitem designar as interações no conjunto de teste como de maior especificidade.

A Figura 19 mostra a especificidade dos métodos AM, MSSC e GRASP para as diversas faixas de probabilidade de interação dos pares de proteínas. Podemos observar que as predições utilizando o GRASP ou o algoritmo guloso para o MSSC são muito semelhantes, e que, ambos proporcionam predições de especificidade significativamente melhor do que o AM. A Figura 20 apresenta a sensibilidade dos métodos segundo a mesma distribuição de faixas de probabilidade de interação protéicas e corrobora a afirmação anterior, uma vez que o aumento da especificidade implica na redução da sensibilidade.



Faixa prob. PP	0	5	10	20	40	60	80	100
AM	3,75	14,59	23,69	36,17	54,53	67,69	77,84	96,47
MSSC	4,96	23,86	36,82	52,74	70,74	90,73	96,09	96,47
GRASP	5,00	25,01	37,82	52,71	71,03	91,02	96,43	96,88

Figura 19: Especificidade do GRASP, MSSC e AM versus Faixas de probabilidades de interação proteína-proteína O conjunto de treinamento é 80% do Conjunto I, e o conjunto de teste é o restante

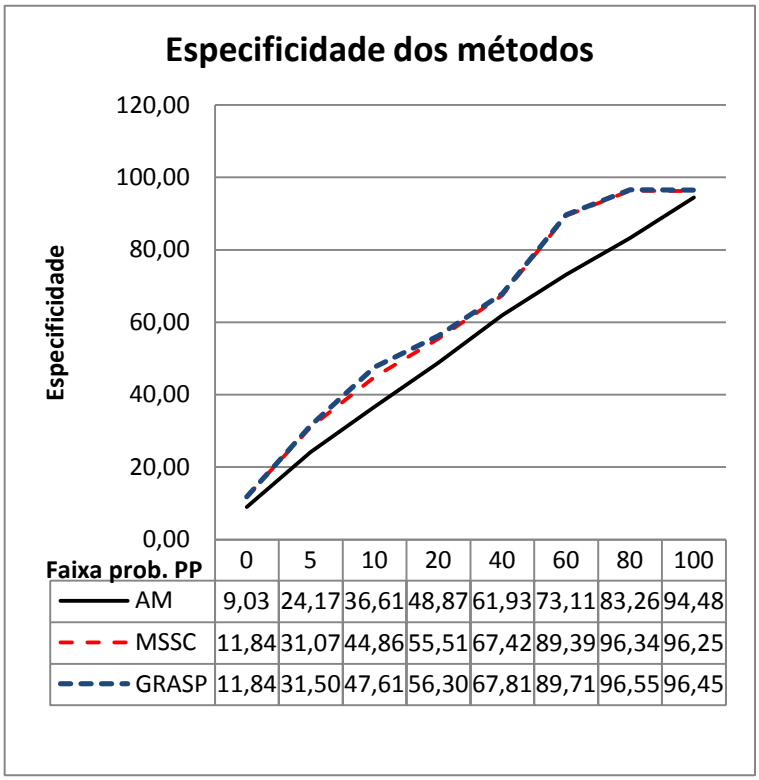


Faixa prob. PP	0	5	10	20	40	60	80	100
AM	44,12	27,50	21,92	14,08	8,49	4,57	1,96	0,00
MSSC	38,97	23,44	17,56	10,96	6,02	2,54	0,22	0,00
GRASP	38,97	22,13	16,76	10,09	5,88	2,47	0,15	0,00

Figura 20: Sensibilidade do GRASP, MSSC e AM versus Faixas de probabilidades de interação proteína-proteína O conjunto de treinamento é 80% do Conjunto I, e o conjunto de teste é o restante

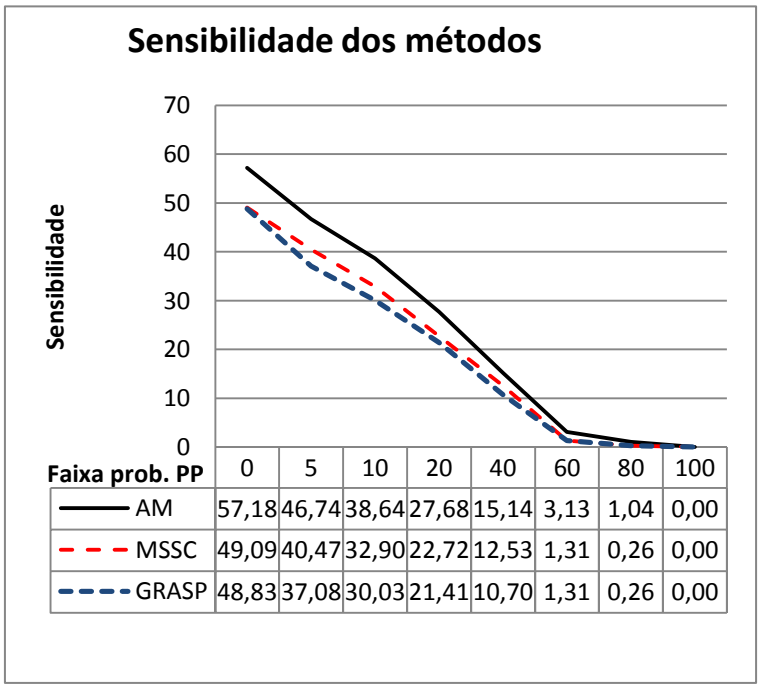
Em particular, nós retiramos amostras aleatórias do conjunto de dados I de interações para dois conjuntos disjuntos (80 por cento: 20 por cento). Tomando a amostra maior como o conjunto de treinamento, fizemos uma média de 100 execuções do algoritmo GRASP, obtendo-se pequenas flutuações nos comportamentos correspondentes a especificidade e a sensibilidade das predições (Figuras 19 e 20), quando comparados com os outros métodos. Os dados mostram que existe uma diferença importante entre os métodos AM e o GRASP ou MSSC, o que não ocorre entre o GRASP e o MSSC.

Os resultados, utilizando-se o conjunto II, também, não apresentaram diferenças significativas quanto à especificidade e sensibilidade, como pode ser visto na Figura 21 e 22, respectivamente. Este fato pode ser explicado pelas seguintes características dos dados e do algoritmo guloso implementado por Huang [19].



Faixa prob. PP	0	5	10	20	40	60	80	100
AM	9,03	24,17	36,61	48,87	61,93	73,11	83,26	94,48
MSSC	11,84	31,07	44,86	55,51	67,42	89,39	96,34	96,25
GRASP	11,84	31,50	47,61	56,30	67,81	89,71	96,55	96,45

Figura 22: Especificidade do GRASP, MSSC e AM versus Faixas de probabilidades de interação proteína-proteína O conjunto de treinamento é 80% do Conjunto II, e o conjunto de teste é o restante



Faixa prob. PP	0	5	10	20	40	60	80	100
AM	57,18	46,74	38,64	27,68	15,14	3,13	1,04	0,00
MSSC	49,09	40,47	32,90	22,72	12,53	1,31	0,26	0,00
GRASP	48,83	37,08	30,03	21,41	10,70	1,31	0,26	0,00

Figura 21: Sensibilidade do GRASP, MSSC e AM versus Faixas de probabilidades de interação proteína-proteína O conjunto de treinamento é 80% do Conjunto II, e o conjunto de teste é o restante

Cerca de 30% dos pares de domínios pertencem, obrigatoriamente, a solução, uma vez que existem pares de proteínas que somente podem ser cobertos por apenas um par de domínios que representar a interação.

Também, ocorre uma grande quantidade de elementos com características semelhantes, (por exemplo, a escolha entre pares de domínios para cobrir um par de proteínas com o mesmo valor da função de seleção), e neste caso, Huang introduzia uma escolha aleatória no processo de seleção. Desta forma, o algoritmo de Huang tem, como na etapa construtiva do GRASP, uma componente gulosa e uma componente aleatória.

Tabela 4 Comparação dos resultados e tempos médios dos métodos (conjunto I)

Método	Valor da função objetivo	Tempo médio (seg.)
AM	367513	12,89
MSSC	194517	29,23
GRASP	186079	38,52

CAPÍTULO 6

6. Conclusões

Nesta dissertação, estudamos algumas técnicas utilizadas na solução do problema da Predição de Interações de Proteínas - PIP. Em particular, focamos o problema de Recobrimento Generalizado utilizado na fase de construção de soluções juntamente com uma estratégia de busca local para o PIP (metaheurística GRASP).

O algoritmo GRASP implementado mostrou-se eficaz quando submetido a instâncias reais descritas na literatura. Quando comparado com os resultados mais bem sucedidos relatados, pode-se dizer que nossa abordagem é competitiva. Isto pode ser justificado facilmente já que ela é fortemente baseada na heurística MSSC, uma técnica recente e bastante eficiente quando comparada a outras técnicas importantes descritas na literatura. Em nosso trabalho, focamos essencialmente na avaliação da qualidade dos resultados sem levar em consideração os tempos de processamento. Obviamente, os tempos obtidos em nossa abordagem são maiores que o algoritmo de construção MSSC puramente guloso e não foram incluídos em nosso trabalho.

Como trabalhos futuros, ressaltamos a importância de investigação de novas estruturas de vizinhança e a utilização de estruturas de dados mais eficientes visando uma redução maior dos tempos de processamento obtidos em nossa implementação.

Anexo

Características das instâncias testadas

Base de Dados de Interação de Proteínas

Utilizamos os conjuntos de dados de interações proteína-proteína e de relacionamento de proteína-domínio usado por Huang e colaboradores [19]. Estes dados foram obtidos, por sua vez do trabalho de Bader e Hogue [35]. Ele introduziu um novo método para a avaliação da qualidade das interações, utilizando outras fontes de informação como a expressão de mRNA, interações genéticas e anotações do banco de dados. Em particular, um procedimento de regressão logística permite uma validação confiável de 47.783 interações proteína-proteína obtidas experimentalmente para 4.627 proteínas de levedura [36] com uma pontuação de confiança que varia de 0 a 1. Nós nos concentramos em 2973 proteínas de levedura incorporadas em 11.368 interações com pontuação maior ou igual a 0,50. O conjunto de dados DIP (*Database of Interactions Proteins*) [37] é um subconjunto de um conjunto maior que o apresentado em Bader e Hogue [35]. Com o limiar de confiança utilizado maior ou igual a 0,50 temos um conjunto de dados de tamanho e qualidade muito similar em relação ao DIP.

Base de Dados de Domínios de Proteínas

Para a nossa análise, nós nos concentramos em dados de domínios recuperados do banco de dados PFAM (*Proteins Families database*) [38], uma coleção confiável de múltiplos alinhamentos de seqüência de famílias de proteínas e que é composto de dois

subconjuntos com diferentes graus de precisão denominados PFAM-A e PFAM-B. (http://pfam.wustl.edu). A versão 10.0, utilizada por Hwuang [39], contém 6190 famílias PFAM-A totalmente anotadas. O PFAM-B fornece alinhamentos de agrupamentos de sequências adicionais gerados pelo PRODOM (*Protein Domains database*) [40] a partir do SWISSPROT e TrEMBL (*the Protein knowledgebase and its supplement TrEMBL*) [41] que não são modelados no PFAM-A. A fim de elucidar a arquitetura de domínio PFAM, nós pesquisamos o SWISSPFAM, uma compilação da estrutura de domínio de proteínas do SWISSPROT e TrEMBL de acordo com PFAM.

Referências bibliográficas

1. Eisenberg, D., E.M. Marcotte, I. Xenarios, and T.O. Yeates, *Protein function in the post-genomic era.* Nature, 2000. **405**(6788): p. 823-826.
2. Shoemaker, B.A. and A.R. Panchenko, *Deciphering Protein–Protein Interactions. Part I. Experimental Techniques and Databases.* PLoS Comput Biol, 2007. **3**(3): p. e42.
3. Ito, T., T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, *A comprehensive two-hybrid analysis to explore the yeast protein interactome.* Proc Natl Acad Sci U S A, 2001. **98**(8): p. 4569-74.
4. Uetz, P., L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson, J.R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J.M. Rothberg, *A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae.* Nature, 2000. **403**(6770): p. 623-7.
5. Ho, Y., A. Gruhler, A. Heilbut, G.D. Bader, L. Moore, S.L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A.R. Willems, H. Sassi, P.A. Nielsen, K.J. Rasmussen, J.R. Andersen, L.E. Johansen, L.H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B.D. Sorensen, J. Matthiesen, R.C. Hendrickson, F. Gleeson, T. Pawson, M.F. Moran, D. Durocher, M. Mann, C.W. Hogue, D. Figeys, and M. Tyers, *Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry.* Nature, 2002. **415**(6868): p. 180-3.
6. Jansen, R. and M. Gerstein, *Analyzing protein function on a genomic scale: the importance of gold-standard positives and negatives for network prediction.* Curr Opin Microbiol, 2004. **7**(5): p. 535-45.
7. Bader, G.D. and C.W.V. Hogue, *Analyzing yeast protein-protein interaction data obtained from different sources.* Nat Biotech, 2002. **20**(10): p. 991-997.
8. Shoemaker, B.A. and A.R. Panchenko, *Deciphering protein-protein interactions. Part II. Computational methods to predict protein and domain interaction partners.* PLoS Comput Biol, 2007. **3**(4): p. e43.

9. Enright, A.J., I. Iliopoulos, N.C. Kyrpides, and C.A. Ouzounis, *Protein interaction maps for complete genomes based on gene fusion events.* Nature, 1999. **402**: p. 86-90.
10. Marcotte, E.M., *Detecting protein function and protein-protein interactions from genome sequences.* Science, 1999. **285**: p. 751-753.
11. Marcotte, E.M., M. Pellegrini, M.J. Thompson, T.O. Yeates, and D. Eisenberg, *A combined algorithm for genome-wide prediction of protein function.* Nature, 1999. **402**: p. 83-86.
12. Wojcik, J. and V. Schachter, *Protein-protein interaction map inference using interacting domain profile pairs.* Bioinformatics, 2001. **17 Suppl 1**: p. S296-305.
13. Gomez, S.M. and A. Rzhetsky, *Towards the prediction of complete protein--protein interaction networks.* Pac Symp Biocomput, 2002: p. 413-24.
14. Sprinzak, E. and H. Margalit, *Correlated sequence-signatures as markers of protein-protein interaction.* J Mol Biol, 2001. **311**(4): p. 681-92.
15. Kim, W.K., J. Park, and J.K. Suh, *Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair.* Genome Inform., 2002. **13**: p. 42-50.
16. Deng, M., S. Mehta, F. Sun, and T. Chen, *Inferring domain-domain interactions from protein-protein interactions.* Genome Res, 2002. **12**(10): p. 1540-8.
17. Uetz, P., *A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae.* Nature, 2000. **403**: p. 623-627.
18. Ito, T., *A comprehensive two-hybrid analysis to explore the yeast protein interactome.* Proc. Natl. Acad. Sci. USA, 2001. **98**: p. 4569-4574.
19. Huang, C., F. Morcos, S.P. Kanaan, S. Wuchty, D.Z. Chen, and J.A. Izaguirre, *Predicting protein-protein interactions from protein domains using a set cover approach.* IEEE/ACM Trans Comput Biol Bioinform., 2007. **4**(1): p. 78-87.
20. Mirzarezaee, M., M. Sadeghi, and B. Araabi, *Dynamical analysis of yeast protein interaction network during the sake brewing process.* The Journal of Microbiology, 2011. **49**(6): p. 965-973.
21. Ekman, D., A.K. Bjorklund, J. Frey-Skott, and A. Elofsson, *Multi-domain proteins in the three kingdoms of life: orphan domains and other unassigned regions.* J Mol Biol., 2005. **348**(1): p. 231-43.

22. Li, W.H., Z. Gu, H. Wang, and A. Nekrutenko, *Evolutionary analyses of the human genome.* Nature., 2001. **409**(6822): p. 847-9.

23. Zhang, S., G. Jin, X.S. Zhang, and L. Chen, *Discovering functions and revealing mechanisms at molecular level from biological networks.* Proteomics, 2007. **7**(16): p. 2856-69.

24. Pereira, M.G., *Epidemiologia*. Vol. 1. 1995: Guanabara Koogan. 595.

25. Meyer, P.L., *Probabilidade. Aplicações à Estatística.* 2 ed1989: LTC.

26. Hayashida, M., N. Ueda, and T. Akutsu, *Inferring strengths of protein-protein interactions from experimental data using linear programming.* Bioinformatics, 2003. **19 Suppl 2**: p. ii58-65.

27. Bennett, K.P. and O.L. Mangasarian, *Robust linear programming discrimination of two linearly inseparable sets.* Optim Method Softw, 1992. **1**(1): p. 23-34.

28. Guimaraes, K.S., R. Jothi, E. Zotenko, and T.M. Przytycka, *Predicting domain-domain interactions using a parsimony approach.* Genome Biol, 2006. **7**(11): p. R104.

29. Feo, T.A. and M.G.C. Resende, *A probabilistic heuristic for a computationally difficult set covering problem.* Oper Res Lett., 1889. **8**(2): p. 67-71.

30. Ya, Z., Z. Hongyuan, C. Chao-Hisen, and J. Xiang, *Protein Interaction Inference as a MAX-SAT Problem*, in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops - Volume 03*2005, IEEE Computer Society.

31. Feo, T.A. and M.G.C. Resende, *Greedy Randomized Adaptive Search Procedures.* J Global Optim, 1995. **6**(2): p. 109-133.

32. Festa, P. and M.G.C. Resende, *GRASP: An annotated bibliography*. Essays and Surveys on Metaheuristics2002: Kluwer Academic Publishers.

33. Resende, M.G.C. and C.C. Ribeiro, *An Introduction to GRASP.* XXXIX Simpósio Brasileiro de Pesquisa Operacional, 2007.

34. Hart, J.P. and A.W. Shogun, *Semi-greedy heuristics: An empirical study. .* Oper Res Lett., 1987. **6**: p. 106-114.

35. Bader, G.D. and C.W. Hogue, *Analyzing yeast protein-protein interaction data obtained from different sources.* Nat. Biotechnol., 2002. **20**: p. 991-997.

36. Bader, J.S., A. Chaudhuri, J.M. Rothberg, and J. Chant, *Gaining confidence in high-throughput protein interaction networks.* Nat Biotechnol., 2004. **22**(1): p. 78-85. Epub 2003 Dec 14.

37. Xenarios, I., *DIP: the Database of Interacting Proteins.* Nucleic Acids Res., 2000. **28**: p. 289-291.

38. Sonnhammer, E.L., S.R. Eddy, and R. Durbin, *Pfam: a comprehensive database of protein domain families based on seed alignments.* Proteins., 1997. **28**(3): p. 405-20.

39. Hwang, Y.C., C.C. Lin, J.Y. Chang, H. Mori, H.F. Juan, and H.C. Huang, *Predicting essential genes based on network and sequence analysis.* Mol Biosyst, 2009. **5**(12): p. 1672-8.

40. Corpet, F., J. Gouzy, and D. Kahn, *The ProDom database of protein domain families.* Nucleic Acids Res., 1998. **26**(1): p. 323-6.

41. Boeckmann, B., A. Bairoch, R. Apweiler, M.C. Blatter, A. Estreicher, E. Gasteiger, M.J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider, *The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003.* Nucleic Acids Res., 2003. **31**(1): p. 365-70.