Programação Genética para Geração de Regras Usadas na Extração de Interações entre Proteínas em Textos

Ana Livia Rodrigues Queiroz[1], Evandro Eduardo Seron Ruiz[1], Renato Tinós[1]

[1]Departamento de Computação e Matemática, FFCLRP, Universidade de São Paulo, Av. Bandeirantes, 3900, 14040-901 Ribeirão Preto, SP, Brasil

lioncourt@aluno.ffclrp.usp.br, evandro@usp.br, rtinos@ffclrp.usp.br

Abstract. *In this work, a combination of syntax patterns used to extract protein-protein interactions from scientific text should be optimized. For this purpose, we present a system based on genetic programming (GP), an evolutionary algorithm that has symbolic expressions as individuals. GP allows the generation of new rules from a preliminary set of rules defined by an expert. The classification error obtained on a set of labeled examples is used as evaluation function. The training set used to evaluate the individuals is the BioCreAtIvE-PPI corpus, which contains textual information about interactions between proteins and /or genes.*

Resumo. *Este trabalho apresenta um método para otimização de um conjunto de regras sintáticas a fim de extrair interações entre proteínas de textos científicos. A técnica de otimização utilizada é a programação genética, um algoritmo evolutivo no qual os indivíduos são tratados como expressões simbólicas. A programação genética permite a geração de novas regras a partir de um conjunto preliminar de regras definidas por um especialista. O erro de classificação obtido sobre um conjunto de exemplos já rotulados é utilizado como função de avaliação. O conjunto de treinamento utilizado para avaliar os indivíduos é o corpus BioCreAtIvE-PPI, que contém informações textuais sobre interações entre proteínas e/ou genes.*

1. Introdução

Proteínas são macromoléculas vitais em processos presentes nos organismos vivos, sendo responsáveis por muitas tarefas, tais como catálise, transporte e armazenamento de substâncias, regulação gênica, entre outras [LEHNINGER *et al.*, 2005]. Desta forma, o estudo das proteínas e de suas interações é de vital importância para o entendimento dos processos biológicos e para a pesquisa de novos medicamentos. Como consequência, o número de artigos publicados que descrevem interações entre proteínas, e destas com genes, tem crescido exponencialmente nos últimos anos. Somente contando os cerca de 1 milhão de resumos de artigos publicados entre 1990 e 2007 contidos na base PubMed, Fundel *et al*. (2007) relatam que foram encontrados cerca de 150.000 artigos sobre relações proteínas/genes apenas em seres humanos.

Nem todas essas informações estão armazenadas de forma estruturada em bancos de dados. Ao contrário, grande parte do conhecimento atual está na forma de texto livre em artigos científicos. Dessa forma, a necessidade da extração automática de interações entre proteínas em textos tem aumentado. Uma das técnicas computacionais potencialmente interessante para a mineração de textos em bioinformática é a dos a*lgoritmos evolutivos* (AEs). AEs são meta-heurísticas populacionais inspiradas em

mecanismos encontrados em evolução natural e genética e que, devido às suas características intrínsecas como uso de operadores estocásticos e de populações de soluções, podem ser interessantes em diversas aplicações complexas.

De fato, AEs têm sido usados em diversas tarefas relacionadas à mineração de texto, e.g. [ALBA *et al*., 2006], incluindo a tarefa de localização de interações entre proteínas em bases de dados relacionadas à área de bioinformática [PLAKE *et al*., 2005]. Nessa última referência, os autores propõem o uso de algoritmos genéticos, um dos tipos mais comuns de AEs, para determinar parâmetros, como o tamanho de *gaps* de palavras, em um conjunto de 22 regras sintáticas usadas para extrair interações entre proteínas a partir de bases de textos. Cada regra contém termos fixos e entidades que podem ser nomes de proteínas ou referências às interações. Por exemplo, a regra: "**Protein** binds to **Protein**" contem entidades, as proteínas, e palavras fixas ("*binds to*").

Este trabalho tem como objetivo investigar o uso de técnicas de *programação genética* (PG) [POLI *et al.*, 2008], que é um AE usado para otimizar estruturas na forma simbólica, para extrair interações entre proteínas, genes ou ambas, a partir de bases de textos. De modo similar ao método apresentado em [PLAKE *et al*., 2005], a PG deve gerar novas regras sintáticas a serem utilizadas para indicar as interações entre proteínas em textos. No entanto, diferentemente do método apresentado em [PLAKE *et al*., 2005], que utiliza um conjunto de regras fixas no qual apenas os parâmetros, como o número de *gaps* entre palavras pré-definidas, são modificados, a PG permite alterar a forma das regras sintáticas através do processo de otimização.

Em nossa proposta metodológica, apresentada na Seção 2, os indivíduos da população são formados por combinações das regras definidas em [PLAKE *et al*., 2005], utilizando os operadores booleanos AND, OR e NOT. Os indivíduos (regras compostas) são avaliados através do erro de classificação obtido sobre um conjunto de exemplos (partes de textos) já rotulados. O conjunto de treinamento utilizado para avaliar os indivíduos é o corpus BioCreAtIvE-PPI [HAKENBERG *et al*., 2005], que contém informações textuais sobre interações entre proteínas e/ou genes. Os resultados alcançados são apresentados na Seção 3. Finalmente, o artigo é concluído na Seção 4.

2. Metodologia

Como em [PLAKE *et al*., 2005], este trabalho propõe otimizar as regras utilizando a abordagem de AEs, neste caso a PG. A PG possui expressões simbólicas como indivíduos, usualmente representados por árvores. No método proposto, cada indivíduo da população é composto por um conjunto de regras. As regras estão unidas através de operadores booleanos AND, OR e NOT. Desta forma, os nós não-terminais da árvore são os operadores booleanos e os nós terminais (folhas) são as regras previamente estabelecidas. A Figura 1 mostra um exemplo de um indivíduo formado por 3 regras. O conjunto de 22 regras utilizadas em [PLAKE *et al*., 2005] é usado como o conjunto de regras básicas aqui utilizadas (nós terminais). A programação genética permite a formação de novas regras a partir da composição das regras já existentes, diferentemente do método apresentado em [PLAKE *et al*., 2005].

Na população inicial, as regras compostas são aleatoriamente criadas utilizando-se as regras básicas. Esta população é então evoluída através de operadores de reprodução e seleção. A seleção é feita por torneio. Neste método, três indivíduos são selecionados aleatoriamente e aquele que possuir maior aptidão será o vencedor. Os

vencedores dos torneios são então submetidos aos operadores de reprodução para formar uma nova população.

Os operadores de reprodução utilizados são o crossover e a mutação. É utilizada a mutação de um ponto, ou seja, cada nó tem uma probabilidade p_m de ser modificado. No caso de um nó terminal, a regra básica representada é substituída por outra qualquer. No caso de um não-terminal, o operador booleano é que é substituído por outro aleatório. O crossover utilizado é o de subárvore, no qual duas subárvores definidas a partir de um dado nó são permutadas com probabilidade p_c calculada como (1- p_m). Em cada geração, os operadores de reprodução utilizados são mutuamente exclusivos, ou seja, um indivíduo sofrerá apenas um dos dois.

Para a avaliação dos indivíduos foi utilizado um arquivo com fragmentos de textos rotulados em duas classes: 0 se ela não contiver informações sobre as interações e 1 caso contrário. Os indivíduos (regras compostas) são avaliados de acordo com a taxa de acerto média produzida na classificação dos exemplos. Antes de executar a PG, a base de dados utilizada precisa sofrer um pré-processamento. Nesta etapa, as palavras são identificadas e recebem rótulos (*tags*) como, por exemplo, "*verbo*" e "*substantivo*". Neste trabalho em particular, é necessário que haja rótulos adicionais para as proteínas e os genes. Dessa forma, o corpus BioCreAtIvE-PPI [HAKENBERG *et al*., 2005] foi escolhido por ter essas características, semelhante ao que foi feito em [PLAKE *et al*., 2005]. O BioCreAtIvE-PPI é um corpus anotado manualmente com 1000 sentenças.

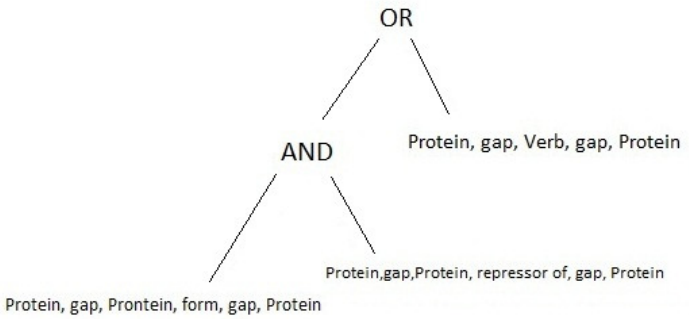

Figura 1. Exemplo de indivíduo da população.

3. Resultados

As sentenças foram divididas em dois conjuntos, de treinamento e teste. O conjunto de treinamento, utilizado para avaliar os indivíduos durante o processo de otimização, possui 750 sentenças, das quais 620 não possuem interações e 130 possuem. Já o conjunto de teste, utilizado para avaliar o melhor indivíduo obtido durante a execução da PG, é formado por 250 sentenças, das quais 207 não possuem interações e 43 possuem. O algoritmo de PG foi implementado em JAVA.

Os experimentos foram feitos inicializando-se a população com 300 indivíduos. A probabilidade de mutação é de 0,9, enquanto que a probabilidade de crossover é de 0,1. Nos experimentos, os formatos das regras básicas não foram alterados e para o gap foi utilizado um valor fixo (máximo) de oito. Foram realizadas cinco execuções (com diferentes sementes aleatórias) da PG com 100 gerações cada. A Tabela 1 mostra os resultados obtidos.

Tabela 1. Resultados obtidos para cinco execuções.

Fitness do melhor indivíduo	Fitness médio	Tamanho médio da árvore (nós)	Erro Médio	Menor Erro
0,907	0,859	19,116	0,141	0,093

O melhor indivíduo das cinco execuções foi aplicado ao conjunto de teste, obtendo fitness de 0,905. O resultado mostra que o indivíduo acertou 226 exemplos, classificando corretamente aproximadamente 50% das sentenças que possuem interações entre proteínas. Foi verificado que os indivíduos tendem a crescer indefinidamente ao longo das gerações, sem haver, contudo, uma melhora significativa no fitness. O fitness do melhor indivíduo convergiu entre as gerações 10 e 20. Também foi verificado que o operador NOT tende a ser excluído dos indivíduos ao longo das gerações.

4. Conclusão

Foi apresentado um sistema que otimiza um conjunto de regras a fim de extrair relações proteína/gene de artigos científicos. Esse sistema é baseado na técnica de programação genética. Nos testes realizados, o melhor fitness encontrado para o conjunto de treinamento foi de 0,907, enquanto que para o conjunto de teste o melhor indivíduo classificou corretamente cerca de 90% dos exemplos, apesar de ter errado a classificação de 50% dos exemplos da classe 1 no conjunto. Isto se deve principalmente ao desbalanceamento entre as classes. Futuramente serão realizados novos experimentos considerando o método de validação cruzada e funções de fitness modificadas, incluindo, por exemplo, o tamanho dos indivíduos na função de avaliação. Outra modificação será a possibilidade de modificar o formato das regras, alterando, por exemplo, o tamanho dos gaps em diferentes regras.

Agradecimentos: Os autores agradecem á FAPESP e ao CNPq pelo apoio financeiro.

Referências

ALBA, E.; LUQUE, G. & ARAÚJO, L. (2006). "Natural language tagging with genetic algorithms", *Information Processing Letters*, 100: 173–182.

FUNDEL, K.; KÜFFNER, R. & ZIMMER, R. (2007). "Relex - relation extraction using dependency parse trees" *Bioinformatics*, 23(3): 365-371

HAKENBERG, J.; BICKEL, S.; PLAKE, C.; BREFELD, U.; ZAHN, H.; FAULSTICH, L.; LESER, U. & SCHEFFER, T. (2005). "Systematic feature evaluation for gene name recognition", *BMC Bioinformatics*, 6(1): 1471-2105.

LEHNINGER, A. L.; NELSON, D. L. & COX, M. M. (2005). "*Lehninger Principles Of Biochemistry"*. New York: Freeman, 4th edition.

POLI, R.; LANGDON, W. B & MCPHEE, N. F. (2008). "*A field guide to genetic programming*". Published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk.

PLAKE, C.; HAKENBERG, J. & LESER, U. (2005). "Optimizing syntax patterns for protein-protein interactions", *In the Proc. of the 2005 ACM Symp. on Applied Computing*, 195-201.