MGI

MESTRADO
Gestão de Informação

PROGRAMAÇÃO GENÉTICA
Aplicada à previsão de parâmetros farmacocinéticos

Dizando Norton António Mvemba

Trabalho de projeto apresentado como requisito parcial para obtenção do grau de Mestre em Gestão de Informação

UNIVERSIDADE NOVA
ISEGI Estatística e Gestão de Informação

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

PROGRAMAÇÃO GENÉTICA APLICADA A PREVISÃO DE PARÂMETROS FARMACOCINÉTICOS

por

Dizando Norton António Mvemba (M2012341)

Trabalho de projeto apresentado como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão do Conhecimento e Business Intelligence

Orientador: Leonardo Vanneschi, Ph.D

Co-Orientador: Mauro Castelli, Ph.D

Abril de 2014

``Essentially, all models are wrong…but some are useful''

George E. P. Box

AGRADECIMENTOS

Agradeço ao criador Jeová pela vida e pelas bençãos. Toda boa obra, conhecimento e sabedoria que o homem produz é somente porque fomos feitos à sua imagem.

Endereço o meu enorme obrigado ao meu orientador e professor, Dr. Leonardo Vanneschi, pelo conhecimento, inspiração e desafios que forneceu ao longo das aulas e dos encontros para a elaboração do presente trabalho. A forma como transmitiu conhecimentos que pareciam ser complexos foi fundamental para que eu desse os primeiros passos no tópico que trata este trabalho. Agradeço também ao seu companheiro de trabalho e meu co-orientador Dr. Mauro Castelli, pela disponibilidade e enorme interesse em ajudar-me quando em dificuldades. Estendo o agradecimento a Dr. Sara Silva, pelo código-fonte que forneceu e pelas dicas ao utilizar a sua excelente ferramenta de Programação Genética.

Ao meus pais, razão da minha existência, agradeço pelos conselhos e força que me transmitiram mesmo estando distantes. Ao Bozé Donadoni, por cumprir em pleno o papel de irmão mais velho e amigo, e as minhas irmâs, Kelani e Cristina, pelas conversas sempre oportunas que me permitiam ``tirar'' a mente do trabalho. À Jeorgina, agradeço por seres meu alicerce e porto seguro. Obrigado pelo amor e pelo suporte, mesmo com as minhas conversas e explicações intermináveis sobre Programação Genética. Agradeço ao Dr. João Silva, que mostrou-se amigo, e transmitiu-me a coragem necessária para enfrentar este desafio com profissionalismo e responsabilidade; espero ter atingido as suas expectativas. Ao Alberto Lourenço, grande amigo, agradeço pela amizade e companheirismo.

À todas as pessoas que direta ou indiretamente contribuiram para a conclusão do presente trabalho…o meu sincero agradecimento!

RESUMO

A Programação Genética (PG) é uma técnica de Aprendizagem de Máquina (*Machine Learning* (ML)) aplicada em problemas de otimização onde pretende-se achar a melhor solução num conjunto de possíveis soluções. A PG faz parte do paradigma conhecido por Computação Evolucionária (CE) que tem como inspiração à teoria da evolução natural das espécies para orientar a pesquisa das soluções.

Neste trabalho, é avaliada a performance da PG no problema de previsão de parâmetros farmacocinéticos utilizados no processo de desenvolvimento de fármacos. Este é um problema de otimização onde, dado um conjunto de descritores moleculares de fármacos e os valores correspondentes dos parâmetros farmacocinéticos ou de sua atividade molecular, utiliza-se a PG para construir uma função matemática que estima tais valores. Para tal, foram utilizados dados de fármacos com os valores conhecidos de alguns parâmetros farmacocinéticos. Para avaliar o desempenho da PG na resolução do problema em questão, foram implementados diferentes modelos de PG com diferentes funções de *fitness* e configurações.

Os resultados obtidos pelos diferentes modelos foram comparados com os resultados atualmente publicados na literatura e os mesmos confirmam que a PG é uma técnica promissora do ponto de vista da precisão das soluções encontradas, da capacidade de generalização e da correlação entre os valores previstos e os valores reais.

PALAVRAS-CHAVE

Programação Genética, parâmetros farmacocinéticos, previsão.

ABSTRACT

Genetic Programming (GP) is a ML technique used in optimization problems where one tries to find the best solution on a set of possible solutions. GP is part of the *Evolutionary Computation* (EC) paradigm inspired by the theory of natural evolution of species to guide the search of solutions.

In this work, we evaluated the performance of GP on the problem of prediction of pharmacokinetic parameters used in the drug development process. This is an optimization problem where, given a set of drug molecular descriptors and the corresponding values of the pharmacokinetic parameters or the molecular activity, GP is then used to build a mathematical model that estimates such values. To this end, we used data of drugs with known values of some pharmacokinetic parameters. To evaluate GP performance in solving the problem at hand, several GP models were implemented with different fitness functions and configurations.

The results from the different GP models were compared with the results currently published in the literature, and they confirm that GP is a promising technique from the point of view of the accuracy of the solutions, their generalization ability and the correlation between the predicted and the actual values.

KEYWORDS

Genetic programming, pharmacokinetics parameters, prediction.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ÍNDICE DE ALGORITMOS

LISTA DE SIGLAS E ABREVIATURAS

%F Biodisponibilidade Oral

%PPB *Plasma Protein Binding*

ADMET Absorção, Distribuição, Metabolismo, Excreção e Toxicidade

AE Algoritmos Evolutivos

AG Algoritmo Genético

ANN *Artificial Neural Networks*

CE Computação Evolucionária

DELOS *Discovery and Lead Optimization Systems*

EC *Evolutionary Computation*

EE Estratégias de Evolução

FDA *Food and Drug Administration*

GP *Genetic Programming*

GPLAB *Genetic Programming toolbox for MATLAB*

HTS *High Throughput Screening*

LD50 *Median Lethal Dose*

LISP *List Processing*

LS *Linear Scaling*

ML *Machine Learning*

MAPE *Mean Absolute Percentage Error*

MASE *Mean Absolute Scaled Error*

MCNS *Minimal Criteria Novelty Search*

MMFF94 *Merck Molecular Force Field 94*

MOE *Molecular Operating Environment*

MSE *Mean Squared Error*

NCI *National Cancer Institute*

NS *Novelty Search*

PCC Coeficiente de Correlação de *Pearson*

PCCN Coeficiente de Correlação de *Pearson* Normalizado

PE Programação Evolutiva

PG Programação Genética

QSAR *Quantitative Structure Activity Relationship*

RCSB-PDB *Research Collabolatory for Structural Bioinformatics - Protein Data Bank*

RE Robótica Evolutiva

RF *Random Forests*

RMSE *Root Mean Squared Error*

SMILES *Simplified Molecular Input Line Entry Specification*

SOM *Self Organizing Maps*

SVM *Support Vector Machines*

1

INTRODUÇÃO

Conteúdo

1.1. COMPUTAÇÃO EVOLUCIONÁRIA

A Computação Evolucionária (CE) é um paradigma na área de Inteligência Artificial que tem como inspiração a teoria da evolução biológica de Charles Darwin. A teoria de Darwin afirma que a evolução dos seres vivos resulta dos processos de seleção, recombinação, reprodução e mutação onde a cada geração apenas sobrevivem os indivíduos mais "aptos" da população (**?**).

Tradicionalmente a CE é composta pelas seguintes áreas ou Algoritmos Evolutivos (AE):

- Estratégias de Evolução (EE) (**?**)
- Programação Evolutiva (PE) (**?**)
- Algoritmo Genético (AG) (**?**)
- Programação Genética (PG) (**?**)

Todas estas técnicas partilham do comum objetivo de produzir sistemas automáticos para a resolução de problemas de otimização ou de pesquisa num espaço de soluções possíveis. A Tabela 1.1 mostra a analogia de alguns dos termos mais utilizados em CE.

Evolução darwiniana	**CE**
Ambiente	Problema
Indivíduo	Solução candidata
Aptidão (*fitness*)	Qualidade da solução

Tabela 1.1 – Analogia entre a evolução darwiniana e a CE

Os AGs são o tipo de AE mais antigo e mais estudado. Num AG, um indivíduo é representado por uma cadeia fixa de caracteres. Cada indivíduo representa uma solução para o problema em questão e a sua capacidade de resolver tal problema é determinada por uma função de aptidão[1]. Ao longo das gerações (ou iterações) do algoritmo, uma nova população é criada contendo os melhores indivíduos da geração anterior e outros

[1]Também chamada de função de *fitness*, função-objetivo ou função de custo. Ao longo deste relatório será utilizado o termo: função de *fitness*.

indivíduos que são produzidos por cruzamento e mutação (**?**). A Figura 1.1 ilustra o esquema de funcionamento geral de um AG.

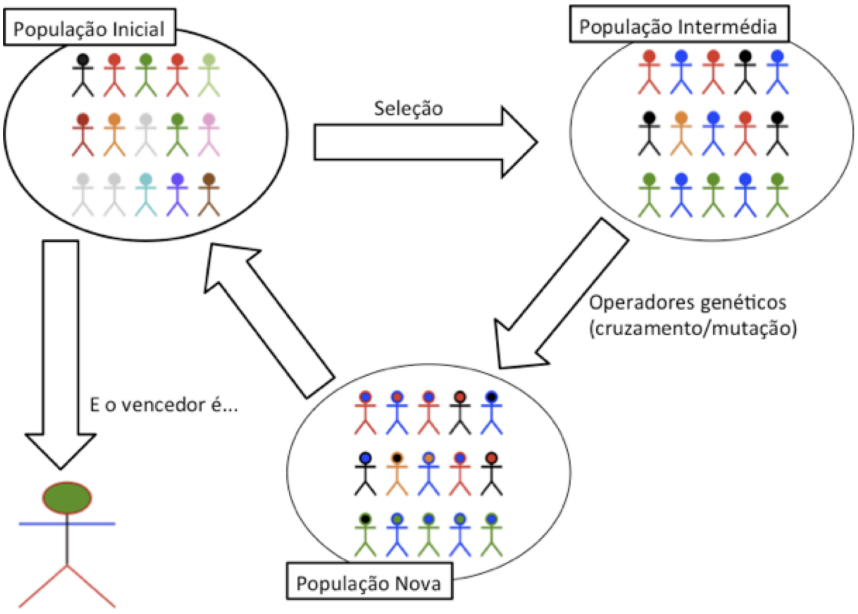

Figura 1.1 – Funcionamento geral de um AG padrão

Para a execução de um AG deve ser definido à partida o conjunto de caracteres que representam os indivíduos. De seguida, é definido também o tamanho L de cada indivíduo, a função de *fitness* e outros parâmetros fundamentais (e.g. quantidade de indivíduos na população, número máximo de gerações, etc.) (**?**). Ao representar os indivíduos com os caracteres binários 0 e 1, o tamanho do espaço de pesquisa (ou o conjunto de soluções possíveis) será 2^L ou seja, existirão 2^L indivíduos diferentes.

O processo, tal como ilustrado na Figura 1.1, começa com a criação aleatória de uma população de tamanho p. Posteriormente o algoritmo entra num ciclo (geração) onde a cada iteração todos os indivíduos são avaliados e recebem um valor de *fitness*. Alguns indivíduos são selecionados e copiados para a geração intermédia. Este processo é repetido p vezes para que no final a população intermédia tenha exatamente p indivíduos. Alguns indivíduos da população intermediária são escolhidos para reprodução, cruzamento ou mutação. O processo termina quando é satisfeito o critério de paragem: um indivíduo da população possui um valor de *fitness* satisfatório ou o número máximo de gerações definido a princípio, foi atingido. Os operadores de seleção são utilizados para escolher os indivíduos que irão fazer parte da reprodução, cruzamento ou mutação.

A reprodução é a cópia exata de um indivíduo da geração intermediária para a nova geração (**?**). Numa operação de cruzamento dois indivíduos são selecionados e combinados gerando dois filhos geralmente diferentes entre si e diferentes dos seus progenitores. Estes novos indivíduos são introduzidos na nova população. O método de cruzamento mais conhecido é o *one point crossover* (**?**). A mutação é uma operação unária, ou seja, requer apenas um indivíduo para a executar. Existem vários métodos de mutação mas o mais conhecido é o *point mutation* em que uma posição do indivíduo é escolhida aleatoriamente e o carácter naquela posição é substituído por outro carácter escolhido aleatoriamente (**?**).

1.1.1. O problema da representação nos Algoritmos Genéticos

Os AGs diferenciam-se de uma pesquisa aleatória em parte pelo facto de a cada geração ser gerada nova informação para orientar a pesquisa. Apesar da sua simplicidade e utilidade, os AGs possuem algumas limitações importantes. Uma delas, por exemplo, é a impossibilidade de representar soluções hierárquicas, uma vez que as cadeias de caracteres são de tamanho fixo. Outra limitação é a impossibilidade de representar estruturas de repetição ou estruturas condicionais que a maior parte dos problemas reais exigem (**??**). Esta habilidade é encontrada nos programas de computador.

1.2. INTRODUÇÃO À PROGRAMAÇÃO GENÉTICA

John Koza introduziu o conceito de PG onde os indivíduos são representados por programas de computador (e.g: expressões matemáticas, expressões lógicas, etc.). Desta forma, em vez de a pesquisa ser feita num conjunto de soluções, são gerados programas que automaticamente resolvem o problema sem a necessidade de se saber à partida a estrutura da solução (**?**).

Uma forma comum de representar os programas de computador em PG é por intermédio de árvores de sintaxe que podem ser facilmente transformadas em programas nas linguagens de programação conhecidas. Na implementação original da PG, o *List Processing* (LISP) foi a linguagem de programação utilizada (**?**).

Os indivíduos da população inicial são criados aleatoriamente e ao longo das ge-

rações são produzidos novos indivíduos pela aplicação dos operadores genéticos. Os indivíduos mais aptos (com maior valor de *fitness* em problemas de maximização ou com menor valor de *fitness* em problemas de minimização) são copiados para as novas gerações ou selecionados para cruzamento ou mutação.

Tal como acontece com os AGs, as operações de seleção, reprodução, cruzamento e mutação também se aplicam ao indivíduos em PG (que são programas de computador). Estes e outros aspetos mais avançados sobre a PG tradicional são o assunto do capítulo 2.

1.3. OBJETIVOS

Neste trabalho pretende-se aplicar a PG para a previsão de parâmetros farmacocinéticos[2] utilizados no desenvolvimento de medicamentos. Posteriormente, será feita uma avaliação da performance da PG padrão, comparando-a com a de outras variantes da PG, sobre os mesmos dados.

1.3.1. Geral

Avaliar a PG como uma ferramenta computacional bem estabelecida aplicada à resolução de problemas reais e de natureza diferente.

1.3.2. Específicos

- Aplicar a PG para a previsão de parâmetros farmacocinéticos utilizados no processo de desenvolvimento de medicamentos
- Comparar a performance da PG padrão com outras variantes de PG que utilizam diferentes funções de *fitness*
- Comparar a PG padrão com as outras variantes tendo como critério a complexidade e usabilidade das soluções encontradas
- Comparar a qualidade dos resultados obtidos pelas diferentes versões de PG com os resultados encontrados na literatura sobre os problemas em questão

[2] A farmacocinética é o estudo da variação das concentrações plasmáticas dos medicamentos

1.4. IMPORTÂNCIA E RELEVÂNCIA

A PG tem sido aplicada largamente na resolução de problemas de otimização, (*Machine Learning* (ML)) e programação automática com reconhecido sucesso. A PG apresenta muitas vantagens sobre os métodos de otimização convencionais, uma vez que pode lidar com vários conjuntos de estruturas no espaço de pesquisa, não requerer informação adicional, exceto a definição do objetivo (através da função de *fitness*) e pode lidar com problemas que possuem muitos ótimos locais[3] e outros (**??**). Desde a sua formalização, experimentos foram realizados em várias áreas com destaque para as seguintes:

- Ajustamento de curva e regressão simbólica (**??**)
- Processamento de imagens e sinais (**?**)
- Negociação financeira, séries temporais e modelação económica (**?**)
- Medicina, Biologia e Bioinformática (**?**)
- Jogos de computador e entretenimento (**?**)
- Arte (**?**)

A pesquisa recente e novas implementações tem feito crescer a aplicabilidade da PG em várias áreas do saber apresentando resultados comparáveis e muitas vezes melhores que os obtidos por humanos utilizando métodos tradicionais ou computacionais. Genericamente a PG tem sucesso em problemas complexos, de domínios pouco conhecidos e em que não se conhece a estrutura e o tamanho da solução (**?**). Para o presente trabalho, utilizaremos a regressão simbólica[4], que é a técnica de PG mais utilizada em trabalhos empíricos publicados nos últimos anos nas principais conferências (**?**).

[3] Num problema de otimização, um ótimo local (mínimo ou máximo) é a melhor solução num conjunto vizinho de soluções candidatas. Em contraste, um ótimo global é a melhor solução entre todas as soluções possíveis, não apenas entre as soluções vizinhas.

[4] A regressão simbólica consiste em induzir (ou descobrir) expressões matemáticas a partir de um conjunto dados numéricos multivariados.

1.4.1. Previsão de parâmetros farmacocinéticos

O sucesso de um tratamento médico está fortemente correlacionado com a capacidade que uma molécula tem em atingir o seu alvo no organismo do paciente sem induzir efeitos tóxicos. Além disso, a redução do custo e o tempo relacionado com a descoberta e desenvolvimento de medicamentos é uma exigência cada vez mais crucial para a indústria farmacêutica. Portanto, métodos computacionais que permitam fazer previsões confiáveis das propriedades dos compostos recém-sintetizados são de extrema relevância (**?**).

Neste trabalho será avaliado o papel da PG sobre o problema da previsão de parâmetros farmacocinéticos, considerando a estimativa dos processos de Absorção, Distribuição, Metabolismo, Excreção e Toxicidade (ADMET) a que é submetido um medicamento no organismo do paciente.

Será estabelecida uma comparação com outras variantes da PG de acordo com a sua capacidade de prever os seguintes parâmetros farmacocinéticos: Biodisponibilidade Oral (%F), Dose Oral Letal Mediana (*Median Lethal Dose* (LD50)) e os níveis de ligação às proteínas do plasma (*Plasma Protein Binding* (%PPB)). Uma vez que estes parâmetros caracterizam respetivamente a percentagem de dose inicial da droga que alcança efetivamente o sistema de circulação sanguínea, os efeitos nocivos e a distribuição do fármaco no organismo, eles são essenciais para a seleção de moléculas potencialmente boas (**?**).

1.5. CONTRIBUIÇÃO

Para a elaboração das previsões dos parâmetros farmacocinéticos foi utilizado o *Genetic Programming toolbox for MATLAB* (GPLAB)[5] que é uma ferramenta de PG para o MATLAB[6] (**?**). Apesar do GPLAB ser uma ferramenta robusta e suficiente para a execução de PG padrão, ele permite a integração de novas funcionalidades (e.g: funções de fitness, operadores genéticos, etc.) na forma *plug-and-play*. Sendo assim, no presente trabalho foram acrescentadas as seguintes novas funções *fitness* ao GPLAB:

- *Linear scaling* (**?**) (ver secção 4.3.2)

[5] `http://gplab.sourceforge.net`

[6] O MATLAB é um *software* de alta performance dedicado ao cálculo numérico desenvolvido pela MathWorks: `www.mathworks.com`

- *Mean Absolute Scaled Error* (MASE) (**?**) (ver secção 4.3.4)

- GPBoost (**?**) (ver secção 4.3.6)

A pesquisa por novidade (*Novelty Search* (NS)), tal como definida em (**?**), é uma técnica utilizada em Robótica Evolutiva (RE), que simplesmente substitui a função de *fitness* por uma medida de novidade de formas a explorar o espaço de comportamentos dos indivíduos ao longo das gerações da PG. A pesquisa por novidade tem apresentado resultados promissores em vários experimentos envolvendo RE, Neuro Evolução e outras ``novas sub-áreas'' da CE, tal como descrito em (**?**) e (**?**). Para o presente trabalho foi desenvolvida uma nova medida de novidade que implementa o conceito da pesquisa de novidade para problemas de regressão simbólica (ver secção 4.3.7). Até ao momento, esta é a segunda tentativa para alcançar tal objetivo, sendo a primeira apresentada em (**?**).

1.6. ESTRUTURA

Para além do presente capítulo de introdução, este trabalho de projeto está dividido pelos seguintes outros capítulos:

- O capítulo 2 fornece uma introdução a PG padrão, apresentado os principais conceitos

- O capítulo 3 descreve a importância da previsão dos parâmetros farmacocinéticos bem como os principais resultados encontrados na literatura

- O capítulo 4 apresenta a metodologia, técnicas, configurações utilizadas para a recolha e análise dos dados, configuração do ambiente de execução dos testes e a ferramenta utilizada durante o processo de previsão

- Os resultados do presente trabalho são apresentados e discutidos no capítulo 5

- Finalmente, o capítulo 6 conclui o presente relatório e traça o caminho para a elaboração de trabalhos futuros

2

PROGRAMAÇÃO GENÉTICA

Conteúdo

2.1. INTRODUÇÃO

A PG é uma técnica utilizada em CE para a resolução de problemas de pesquisa e otimização. Em PG, diferentemente de uma pesquisa aleatória, o objetivo é fazer com que as soluções melhorem ao longo da execução do algoritmo através da aplicação de uma função de *fitness*. Durante este processo, os indivíduos mais aptos (ou parte das suas características), são preservados através da utilização de operadores genéticos (**?**). No geral, a PG cria programas de computador para resolver problemas executando os seguintes passos:

1. Cria uma população de programas de computador (soluções, indivíduos).
2. Executa iterativamente os seguintes passos até que um critério de paragem seja satisfeito:
    (a) Executa cada programa na população e atribui um valor de *fitness* de acordo a sua capacidade de resolver o problema.
    (b) Cria uma nova população aplicando as seguintes operações:
        i. Seleciona, probabilísticamente, um conjunto de programas de computador para serem reproduzidos com base na sua *fitness* (seleção).
        ii. Copia alguns dos programas selecionados, sem modificá-los, para a nova população (reprodução).
        iii. Cria novos programas de computador por combinar genéticamente partes de dois indivíduos selecionadas aleatoriamente (cruzamento).
        iv. Cria novos programas de computador por substituir partes selecionadas aleatoriamente de um indivíduo por novos indivíduos criados aleatoriamente (mutação).
3. Os melhores programas de computador encontrados numa geração são o resultado do processo de PG para tal geração. Este resultado pode ser uma solução (ótima ou aproximada) para o problema.

Nas próximas secções, cada um destes passos é apresentado em mais detalhes.

2.2. REPRESENTAÇÃO DOS INDIVÍDUOS

Os AGs diferem da PG essencialmente pela forma como os indivíduos são representados e codificados. Nos AGs, os indivíduos são representados por uma cadeia fixa de caracteres (geralmente binários), tal como ilustrado na Figura 2.1.

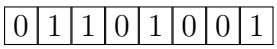

Figura 2.1 – Representação de um indivíduo em cadeia de caracteres binários

A representação em cadeia fixa de caracteres carece de uma propriedade importante e usualmente encontrada nas soluções de problemas complexos: a organização hierárquica das soluções em tarefas e subtarefas (**?**). Para além desta deficiência, outras foram levantadas em (**?**) e (**?**).

Na PG os indivíduos são programas de computador. Estes indivíduos são geralmente representados em árvores de sintaxe, mas existem outras formas de representação como por exemplo: linear (**??**), em grafo (**??**) e cartesiana (**??**). Neste trabalho adoptou-se a representação mais comum: árvores de sintaxe (**?**).

As árvores de sintaxe são construídas a partir de um conjunto de funções $F = \{f_1, \ldots, f_n\}$ que representam os nós da árvore e um conjunto de símbolos terminais $T = \{t_1, \ldots, t_n\}$ que representam as folhas da árvore. Assim, o espaço de procura de soluções é constituído por todas as expressões que podem ser construídas recursivamente com as funções em F e os símbolos terminais em T.

Cada função do conjunto F pode ser uma função aritmética, matemática, lógica ou outra, que aceite um determinado número de argumentos (aridade) (**?**). Um elemento do conjunto T pode ser uma variável ou uma constante definida sobre o domínio do problema. A Figura 2.2, apresenta uma árvore válida construída "aleatoriamente" a partir dos conjuntos $F = \{+, -, *, /\}$ e $T = \{x_1, x_2\}$.

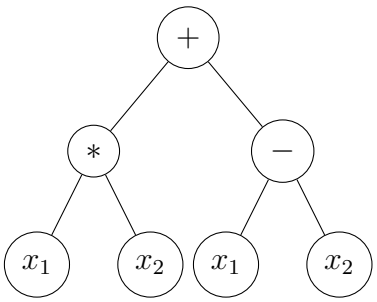

Figura 2.2 – Exemplo de um indivíduo em PG

Esta árvore corresponde a expressão matemática,

$$f(x_1, x_2) = (x_1 * x_2) + (x_1 - x_2) \tag{2.1}$$

As funções em F devem obedecer a propriedade do fechamento para garantir consistência nos tipos de dados e segurança na avaliação das expressões por elas criadas (**?**). Considerando a expressão da Equação (2.1) que representa o indivíduo da Figura 2.2, a consistência nos tipos de dados consiste em garantir que os valores para as variáveis x_1 e x_2 estejam definidos para os operadores de adição ($+$) e subtração ($-$). Mesmo respeitando esta propriedade algumas funções podem falhar ao serem executadas. Um caso comum, é a divisão por 0. Koza introduziu uma nova operação, a divisão protegida (%), que retorna o valor 1 sempre que o denominador for igual a 0 (**?**).

Os elementos dos conjuntos F e T devem ser suficientes para representar as soluções do problema dado. Esta propriedade de suficiência, nem sempre é satisfeita porque antes da execução do algoritmo a estrutura da solução não é conhecida, e logo, não há garantia que os elementos em F e T sejam suficientes para representar a solução do problema. Um exemplo de um conjunto suficiente é o conjunto $F = \{AND, OR, NOT\}$ que só por si é adequado para representar qualquer função *booleana* (**?**).

2.3. INICIALIZAÇÃO DA POPULAÇÃO

Os indivíduos na população inicial são criados aleatoriamente tal como nos AGs porém, os métodos utilizados para tal são diferentes. Em PG existem três métodos comuns de inicialização:

- Método completo (*full*)

- Método de crescimento (*grow*)
- Método em rampa meio-a-meio (*ramped half-and-half*)

No método completo, uma função é selecionada aleatoriamente do conjunto de funções F para ser o nó raiz. De seguida, outras funções são selecionadas também aleatoriamente em F para formar os outros nós da árvore. A profundidade[1] máxima da árvore é preenchida apenas por símbolos terminais selecionados aleatoriamente no conjunto de terminais T. Dessa forma, são criadas árvores completas em que todas as folhas se encontram à mesma profundidade. O Algoritmo 2.1 contém o pseudocódigo para o método *completo*.

```
1:  função COMPLETO(profMaxima)
2:      se profMaxima = 1 então
3:          no ← n ∈ ℕ, ARIDADE(n) = 0
4:      senão
5:          no ← n ∈ ℕ, ARIDADE(n) ≠ 0
6:          para i ← 1 : ARIDADE(n) faça
7:              ADDDESCENDENTE(no, COMPLETO(profMaxima − 1))
8:          fim para
9:      fim se
10:     retorne no
11: fim função
```

Algoritmo 2.1 – Método de Inicialização Completo

No Algoritmo 2.1, $profMaxima$ é a profundidade máxima da árvore, $\mathbb{N}$ é o conjunto formado pelas funções em F e os símbolos em T, e n é um elemento de $\mathbb{N}$. A função ARIDADE(n) determina o número de argumentos ou operandos de n. Quando a aridade de n é igual a zero (ARIDADE$(n) = 0$), isto significa que n é um símbolo terminal. A função ADDDESCENDENTE$(no,$ COMPLETO$(profMaxima - 1))$ conecta o nó-filho no ao seu nó-pai e acrescenta os outros nós à árvore de forma recursiva.

O método de *crescimento* funciona de forma quase semelhante ao método *completo*, exceto que não são criadas árvores completas ou cheias, pois os nós são selecionados aleatoriamente de um conjunto formado pelas funções e os símbolos terminais. Sempre que for selecionado um símbolo terminal, o crescimento da árvore para aquele

[1] A profundidade de uma árvore é a quantidade de nós que devem ser percorridos desde a raiz da árvore até ao nó mais profundo (folha).

nó termina mesmo que não tenha sido atingida a profundidade máxima. O Algoritmo 2.2 ilustra o pseudocódigo para o método de *crescimento*.

```
1: função CRESCIMENTO(profMaxima)
2:     se profMaxima = 1 então
3:         no ← n ∈ ℕ, ARIDADE(n) = 0
4:     senão
5:         no ← n ∈ ℕ
6:         para i ← 1 : ARIDADE(n) faça
7:             ADDDESCENDENTE(no, CRESCIMENTO(profMaxima − 1))
8:         fim para
9:     fim se
10:    retorne no
11: fim função
```

Algoritmo 2.2 – Método de Inicialização de Crescimento

O método *completo* assume a princípio uma estrutura completa para os indivíduos e o método de *crescimento* pode gerar árvores muito curtas, caso existam muitos elementos de aridade igual a 0 (símbolos terminais) (**?**). Numa inicialização em *rampa meio-a-meio*, metade da população é criada com o método *completo* e a outra metade é criada com o método de *crescimento*. Este procedimento permite a construção de árvores de tamanhos e configurações diferentes, garantindo assim a diversidade na população (**?**). O pseudocódigo para o método em *rampa meio-a-meio* é apresentado no Algoritmo 2.3.

```
1: função RAMPA(profMaxima, probCrescimento)
2:     profundidade ← ALEATORIO(1, profMaxima)
3:     se ALEATORIO(0, 1) < probCrescimento então
4:         retorne CRESCIMENTO(profundidade)
5:     senão
6:         retorne COMPLETO(profundidade)
7:     fim se
8: fim função
```

Algoritmo 2.3 – Método de Inicialização em Rampa Meio-a-Meio

No Algoritmo 2.3, $probCrescimento$ é um parâmetro que determina a probabilidade de selecionar o método *completo* ou o método de *crescimento* ao construir uma árvore. A variável $profundidade$ recebe um valor aleatório, entre 1 e $profMaxima$, que determina o tamanho para a árvore a ser construída.

2.4. FUNÇÃO DE *FITNESS*

A capacidade que um indivíduo tem em resolver um problema é quantificada pela função de *fitness*. A função de *fitness* avalia a performance do indivíduo executando-o num conjunto de casos de aptidão conhecidos. Num problema de regressão simbólica, em que se pretende ajustar uma expressão à um conjunto de dados, os casos de aptidão (ou casos de *fitness*) são os valores que as variáveis independentes[2] assumem nos diferentes pontos desse conjunto.

2.4.1. *Fitness* bruto

Considerando o valores para x_1 e x_2 (variáveis independentes) e y (variável dependente[3]) representados na Tabela 2.1, o trabalho da regressão simbólica consistirá em encontrar uma função $f(x_1, x_2)$ que produz valores de saída iguais ou aproximados aos de y. Uma possível solução é o indivíduo representado pela Figura 2.2 que origina a função $f(x_1, x_2) = (x_1 * x_2) + (x_1 - x_2)$.

x_1	x_2	y
1	−1	0
0	1	−1
−2	2	15
−1	2	−8
2	−1	7

Tabela 2.1 – Casos de *fitness*

Para avaliar a capacidade da função $f(x_1, x_2)$ se ajustar aos dados da Tabela 2.1, deve-se tradicionalmente calcular uma medida de erro. Uma medida de erro muito utilizada é o erro quadrático médio (*Mean Squared Error* (MSE)), dado pela fórmula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2 \tag{2.2}$$

onde n é o número de casos de *fitness* (linhas, registos ou exemplos), y_i são os valores de saída conhecidos e f_i são os valores gerados por Equação (2.1). Ao aplicar a Equação

[2]Também conhecida por: variável de entrada, característica, variável de previsão, atributo, etc.
[3]Também conhecida por: variável de saída, variável de resposta, valor de saída, etc.

(2.1) e a Equação (2.2) ao nosso exemplo, obtêm-se os resultados na Tabela 2.2.

x_1	x_2	y	$f(x_1, x_2)$	$(y - f)^2$
1	-1	0	1	1
0	1	-1	-1	0
-2	2	15	-8	49
-1	2	-8	-5	9
2	-1	7	1	36
			MSE	95

Tabela 2.2 – Valores para y, f e erro quadrático médio de f

O valor obtido (95), é o *fitness* bruto do indivíduo representado pela Figura 2.2, para os casos de aptidão na Tabela 2.1. O *fitness* bruto expressa a aptidão da solução numa terminologia natural do problema (**?**).

2.4.2. *Fitness* padronizado

O *fitness* padronizado é uma transformação ao *fitness* bruto de formas a que um valor menor de *fitness*, é considerado melhor. Em muitos casos é conveniente e desejável fazer com que o melhor valor de *fitness* padronizado seja igual a zero. Isto pode ser obtido através da soma ou subtração de uma constante. Num problema em que um valor de *fitness* bruto é melhor e o valor máximo de *fitness* bruto é conhecido, o *fitness* padronizado pode ser obtido pela fórmula:

$$f_S(i) = f_R^{max} - f_R(i) \tag{2.3}$$

onde $f_R(i)$ é o *fitness* bruto de i.

2.5. SELEÇÃO

Após a determinação da aptidão dos indivíduos da população numa geração, deve-se decidir se os indivíduos serão copiados ou selecionados para cruzamento ou mutação. Esta é a função dos operadores de seleção. Existem vários operadores de seleção mas, os mais utilizados são: a seleção proporcional à *fitness* (roleta russa), a seleção por classificação (*ranking*) e a seleção por torneio.

Na seleção proporcional à *fitness* um indivíduo é selecionado com base numa probabilidade que é dada por:

$$p_i = \frac{f_i}{\sum f} \tag{2.4}$$

onde p_i é a probabilidade de o indivíduo i ser selecionado e f_i é a *fitness* de i.

Na seleção por classificação os indivíduos são ordenados com base no seu *fitness*. De seguida é designada uma probabilidade a cada indivíduo em função da sua ordem na população. Normalmente são utilizadas classificações lineares e exponenciais.

A seleção por torneio, diferentemente das outras duas apresentadas anteriormente, não é baseada numa "competição" entre todos os indivíduos da população. Apenas um número de indivíduos (chamado tamanho do torneio) é selecionado aleatoriamente. O indivíduo com o melhor *fitness* nesse grupo é escolhido. Este procedimento é repetido N vezes, onde N é o tamanho da população. Este método é amplamente utilizado em PG principalmente porque não requer uma comparação de *fitness* centralizada entre todos os indivíduos. Este método também permite poupar o processamento computacional.

2.6. CRUZAMENTO

O operador de cruzamento (recombinação sexual) introduz diversidade na população por produzir novos indivíduos (filhos) que são compostos por partes retiradas de cada um dos pais. Os pais são escolhidos através dos métodos de seleção apresentados na secção 2.5. O método mais comum de cruzamento é o cruzamento de subárvore (**?**) que funciona da seguinte forma:

- Seleciona dois indivíduos (pais) utilizando um método de seleção
- Seleciona uma subárvore aleatória em cada um dos pais. A raíz dessa subárvore é o nó ou ponto de cruzamento[4]
- Cria dois novos indivíduos (filhos) por trocar as duas subárvores selecionadas entre os pais.

[4] É comum, e conveniente, que subárvores constituídas por símbolos terminais ou pelo nó-raiz sejam selecionadas com probabilidade baixa em relação as outras pois esta é provavelmente uma das causas do fenómeno conhecido por *bloat*, que é o crescimento excessivo das árvores sem uma correspondente melhoria da *fitness*.

A Figura 2.3 ilustra um exemplo de um cruzamento de subárvore.

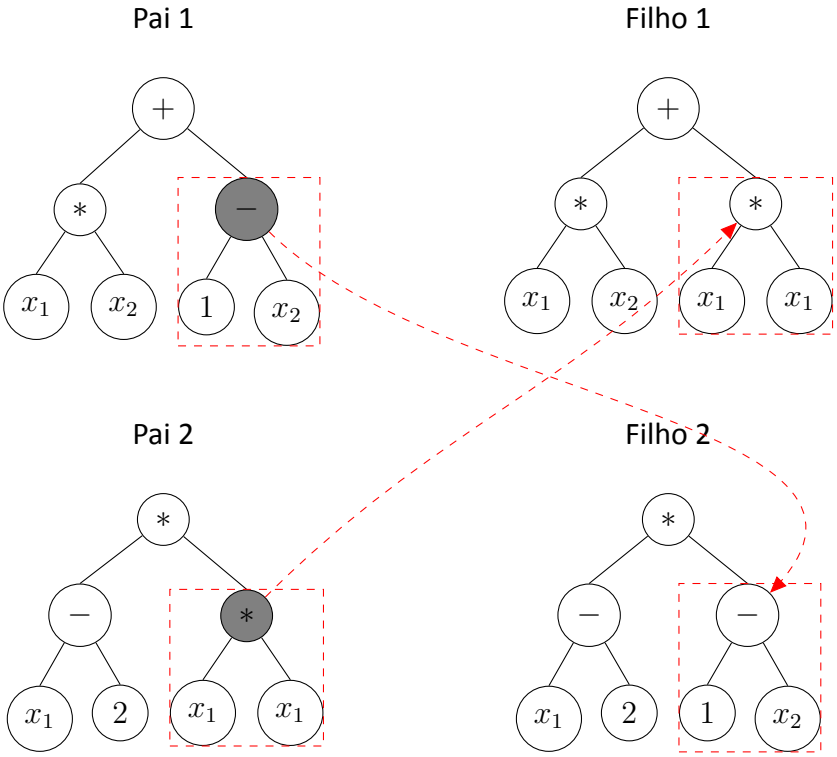

Figura 2.3 – Exemplo de cruzamento de subárvore. O nó cinzento é o ponto de cruzamento

2.7. MUTAÇÃO

Outro operador que altera a estrutura de um indivíduo é a mutação. Em PG o método mais comum de mutação é a mutação de subárvore que seleciona aleatóriamente um ponto (nó) numa árvore e substitui a subárvore com raíz nesse ponto por uma outra subárvore gerada aleatóriamente. A Figura 2.4 ilustra um exemplo de um mutação de subárvore.

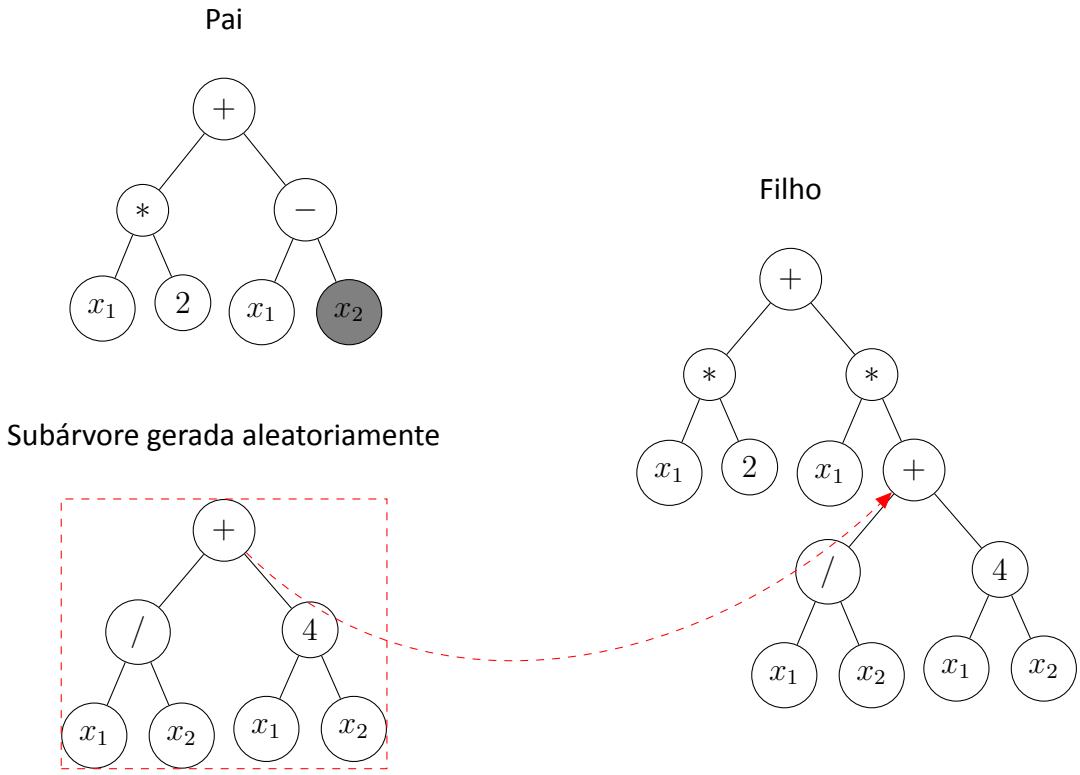

Figura 2.4 – Exemplo de mutação de subárvore. O nó cinzento é o ponto de mutação e é substituído pela subárvore gerada aleatoriamente

Outros operadores de mutação também muito utilizados em PG são: mutação de troca e mutação de ponto. A mutação por troca seleciona aleatoriamente duas subárvores e as troca. A mutação de ponto seleciona aleatóriamente um nó e o susbtitui com um nó aleatório de mesma aridade.

2.8. REPRODUÇÃO

A reprodução consiste simplesmente em copiar um indivíduo de uma geração para outra sem alterá-lo. Esta operação é geralmente associada a uma técnica de *elitismo*. O elitismo garante que um ou mais indivíduos são copiados, inalterados, de uma geração para a outra. A proporção $\frac{N}{M}$ entre o tamanho da elite N e o tamanho da população M, é chamada de *fracão da elite*.

2.9. PARÂMETROS

Antes de executar a PG, é necessário executar os seguintes passos preparatórios:

- Definir o conjunto de terminais T
- Definir o conjunto de funções F
- Escolher a função de *fitness*
- Definir os parâmetros para controlar execução
- Escolher o critério de paragem

Por outro lado, os parâmetros para controlar a execução da PG são os seguintes:

- Tamanho da população
- Técnica utilizada para a inicialização da população
- Algoritmo de seleção
- Método e taxa de cruzamento
- Método e taxa de mutação
- Profundidade máxima das árvores

O critério de paragem é o método que determina o resultado (fim) da execução. A execução pode ser terminada quando for atingido o número máximo de gerações ou quando é encontrado um indivíduo com uma *fitness* considerada aceitável. A escolha destes parâmetros é um passo muito importante uma vez que os mesmos determinam a performance da PG. A definição dos parâmetros escolhidos para o presente trabalho são apresentados no capítulo 4.

3

PARÂMETROS FARMACOCINÉTICOS

Conteúdo

3.1. INTRODUÇÃO

Os medicamentos[1] receitados para lidar com certas doenças podem por vezes produzir efeitos colaterais sobre o corpo humano[2]. Esses efeitos ou reações adversas podem acontecer nalgum ponto crítico do ciclo de vida do medicamento. Sendo assim, é importante perceber quais propriedades químicas dos medicamentos produzem o efeito desejado. Este objetivo é alcançado pelo processo de triagem de alta produtividade (*High Throughput Screening* (HTS)), que é um método computacional para a busca de moléculas relevantes em enormes quantidades de substâncias que compôem os medicamentos (**?**).

Os resultados obtidos pela triagem de alta produtividade são posteriormente utilizados para otimizar as propriedades de certas moléculas no processo de fabricação de medicamentos. Além disso, é necessário garantir que os fármacos percorram o caminho apropriado no corpo humano sem alterar a saúde do paciente. Por esta razão, o comportamento das moléculas é avaliado durante os processos de absorção, distribuição, metabolismo e excreção do medicamento no organismo. O estudo deste processo recebe o nome de farmacocinética (**?**).

Em suma, a farmacocinética dá uma resposta a pergunta sobre como o corpo lida com o fármaco e é composta pelas seguintes fases:

- **Absorção** - é o processso de entrada das substâncias na circulação sanguínea
- **Distribuição** - é a dispersão ou disseminaçao das substâncias através dos fluídos e tecidos do corpo
- **Metabolismo** - é o reconhecimento, pelo organismo, de que uma substância está presente e a transformação irreversível dos compontes desta substância em metabolitos
- **Excreção** - é a remoção das substâncias do corpo. Raramente alguns medicamentos acumulam-se nos tecidos do corpo

[1] No presente trabalho, os termos *medicamento*, *fármaco* e *droga* serão utilizados alternadamente.

[2] Por questões de simplicidade aqui nos referimos ao corpo humano. No entanto esses conceitos podem ser generalizados à outros animais.

- **Toxicidade** - representa o quanto uma substância pode prejudicar o organismo

Quase metade das falhas no desenvolvimento de compostos fármacológicos são relacionadas com a farmacocinética e a toxicidade dos fármacos, segundo estudos publicados nos anos de 1991 (**?**) e 2000 (**?**), tal como ilustrado na Figura 3.1.

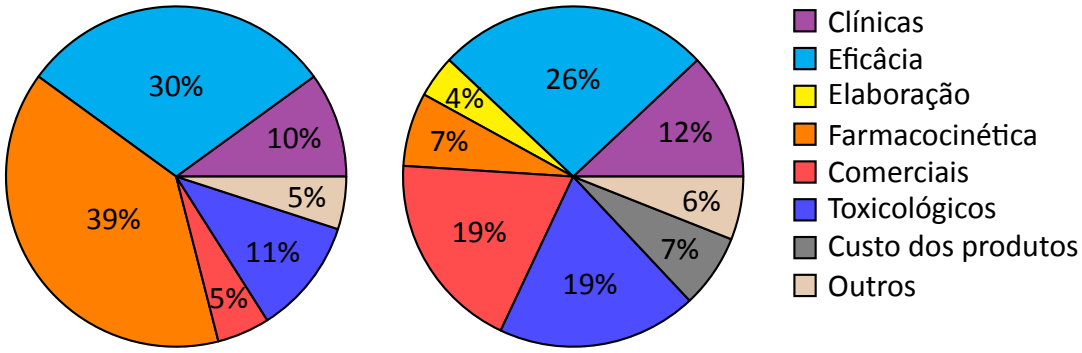

Figura 3.1 – Razões para falhas no desenvolvimento de fármacos (valores aproximados). O gráfico da direita ilustra os resultados de 2009 e o da esquerda os resultados de 2000

Alguns parâmetros farmacocinéticos diretamente correlacionados com o processo ADMET serão utilizados neste trabalho, nomeadamente: a Biodisponiblidade Oral %F, o nível de Ligação às Proteínas do Plasma (%PPB) e a Dose Letal Mediana LD50[3]. Outro parâmetro diretamente ligado ao comportamento dos fármacos no organismo, e que será utilizado no presente trabalho, é a energia de acoplamento molecular. Finalmente, também será utilizado o fármaco Fludarabina. As próximas secções apresentam uma breve descrição destes itens e como serão utilizados no presente trabalho.

3.2. PARÂMETROS FARMACOCINÉTICOS UTILIZADOS

3.2.1. %F

A %F indica a percentagem do medicamento administrado que chega ao sistema circulatório comparada com o método de administração intravenoso[4] após a passagem pelo fígado. A biodisponilidade oral é determinada pelos de processos farmacocinéticos de *absorção* e *metabolismo*. Após uma administração intravenosa, a biodisponibilidade

[3] Para uma lista mais extensa dos parâmetros farmacocinéticos, consulte (**?**).

[4] A administração intravenosa consiste na injeção de medicamentos por meio de agulhas nas veias periféricas dos membros superiores.

é 100% enquanto que numa administração oral a biodisponibilidade é geralmente inferior. Tipicamente, isto acontece devido a muitos fatores, sendo um deles a incompleta absorção intestinal (**?**).

3.2.2. %PPB

A distribuição de um medicamento do plasma para os tecidos de destino no corpo humano pode ser afetada por vários fatores e um dos mais importantes é a percentagem de ligação às proteínas do plasma (%PPB). Quanto menor for essa percentagem, mais eficientemente o fármaco atravessa as membranas celulares e se difunde. No sangue, uma proporção de um fármaco pode estar ligada ou não ligada, de acordo a sua afinidade às proteínas do plasma (**?**). A proporção não-ligada é a que possui efeitos farmacológicos e é também esta porção que será metabolizada e/ou excretada. Por exemplo, a proporção ligada do anticoagulante varfarina[5] é de 97%. Isto significa que a quantidade de varfarina no sangue, 97%, está ligada à proteínas do plasma. O restante 3% (fracção não-ligada) é a fracção não-ativa e poderá ser excretada (**?**). Substâncias que se ligam fortemente às proteínas do plasma têm grande impacto na eficâcia do fármaco uma vez que são as responsáveis pela acção do mesmo.

3.2.3. LD50

A LD50 é um teste feito para determinar o risco ou potencial de toxicidade de substâncias existentes ou novas. Os testes de LD50, geralmente realizados em ratos, são caros, morosos e ativamente combatidos por ativistas. Em particular, a informação sobre a toxicidade aguda de substâncias químicas é necessária como um dos critérios essenciais para avaliar sua segurança (**?**).

3.2.4. Energia de acoplamento molecular

O objetivo do acoplamento molecular no desenvolvimento de fármacos é de identificar drogas candidatas direcionadas às proteínas receptoras no organismo. Estas drogas candidatas podem ser encontradas utilizando um algoritmo de acoplamento que tenta

[5] A varfarina é um anticoagulante utilizado na prevenção de tromboses.

identificar uma ligação otimizada de uma pequena molécula (chamada de *ligante*) às proteínas receptoras no seu local ativo de formas que a energia livre de todo o sistema seja minimizada. Esta energia é chamada de: energia de acoplamento molecular (**?**).

3.2.5. Fludarabina

A fludarabina não é um parâmetro farmacocinético mas sim um fármaco utilizado no tratamento de leucemia linfocítica[6]. É um dos 118 fármacos que fazem parte da base de dados NCI-60 do *National Cancer Institute* (NCI)[7] (**?**) que é composta por 60 linhagens de células tumorais derivadas de pacientes com os seguintes 9 tipos de câncer: coloretal, renal, do ovário, da mama, da próstata, do sistema nervoso central, leucemias e melanomas (**?**).

A acção da fludarabina (e de outros fármacos) foi estimada através da medição da inibição do crescimento das linhagens de células tumorais 48 horas após tratamento e é definida como sendo a concentração logarítmica necessária para reduzir a taxa de crescimento para 50%. O nosso objetivo é encontrar uma relação matemática entre o perfil das expressões génicas e o padrão de atividade da fludarabina.

3.3. PREVISÃO DE PARÂMETROS FARMACOCINÉTICOS

As ferramentas de simulação computacional (*in silico*) para a previsão de parâmetros farmacocinéticos são de particular interesse na indústria farmacêutica. Para além de pouparem tempo e recursos, os modelos computacionais permitem também que moléculas inapropriadas sejam descartadas durante a fase inicial do processo de desenvolvimento de fármacos. O objetivo dos modelos *in silico* de absorção, distribuição, metabolismo e excreção é obter uma previsão *in vivo*, o mais precisa possível, do efeito das potenciais molêculas de um medicamento no organismo humano (**??**). Existem básicamente dois métodos de modelação computacional: modelação molecular e modelação de dados. Os métodos de modelação molecular utilizam cálculos intensivos nas estruturas proteícas. Os métodos baseados em modelação de dados são amplamente divulgados na literatura e pertencem a categoria de modelos de previsão chamados de

[6] A leucemia é uma neoplasia que afecta os glóbulos brancos (células imaturas da medula óssea).
[7] Instituto Nacional do Câncer dos Estados Unidos da América: `www.cancer.gov`.

``modelos da Relação Quantitativa Estrutura-Atividade'' (*Quantitative Structure Activity Relationship* (QSAR)).

Os modelos QSAR têm por objetivo definir uma relação quantitativa entre a estrutura de uma molécula e suas atividades biológicas. Para tal, é necessário obter um conjunto de dados de treino de fármacos para os quais são conhecidos os parâmetros de atividade biológica. Um enorme conjunto de características, ou descritores moleculares, são calculados apartir da estrutura molecular de cada composto. A construção de modelos QSAR geralmente envolve a execução dos seguintes três passos:

1. Adquirir, ou se possível, desenvolver um conjunto de dados de treino de compostos químicos para os quais se conhecem os parâmetros biológicos

2. Obter os descritores moleculares que relacionem-se adequadamente com a atividade biológica

3. Aplicar métodos para construir uma relação matemática que permite calcular a atividade biológica

A obtenção de modelos QSAR de qualidade, capazes de prever a atividade biológica de um composto químico fora de um conjunto de treino, depende de muitos fatores como a qualidade dos dados e as escolhas dos descritores mais significantes. Em modelos QSAR, utilizam-se geralmente duas categorias de desctirores moleculares: descritores químicos bidimensionais, baseados na representação bidimensional dos compostos, e descritores químicos tridimensionais (**?**).

As técnicas de ML têm sido muito aplicadas ao desenvolvimento de modelos QSAR. Por exemplo, o método dos mínimos quadrados adaptativos difusos (*fuzzy adaptive least squares*) é utilizado em (**?**) para treinar um modelo QSAR para classificar os fármacos em uma das 4 classes predifinidas de biodisponibilidade de acordo com a presença ou ausência de grupos funcionais típicos mais suscetíveis de estarem envolvidos em reações metabólicas. Em (**?**), AGs foram utilizados para selecionar os melhores descritores moleculares e mapas auto-organizáveis (*Self Organizing Maps* (SOM)) foram utilizados para atribuir uma classe de biodisponibilidade à cada um. Em (**?**) foram utilizadas árvores aleatórias (*Random Forests* (RF)), árvores de decisão e o método de médias móveis para a

previsão dos parâmetros farmacocinéticos da cefalosporina[8].

Máquinas de vetores de suporte (*Support Vector Machines* (SVM)) são utilizadas em (**?**) para a previsão de biodisponibilidade estimando a similaridade entre moléculas diferentes com comportamentos biológicos semelhantes.

As redes neuronais artificiais (*Artificial Neural Networks* (ANN)) são geralmente utilizadas para a construção de modelos QSAR (**?**) e são frequentemente integradas em pacotes de *software* comerciais de empresas envolvidas no ramo de modelação molecular. Entre os líderes neste ramo, as empresas Accelrys Inc. (**?**) e PharmaAlgorithms Inc. (**?**) provêm modelos matemáticos de caixa-preta[9] e ferramentas de *data mining* para a construção de novos indicadores.

Os AE têm sido utilizados com enorme sucesso na elaboração de modelos moleculares (**?**) e em outras tarefas no processo de desenvolvimento de fármacos (**?**). Por exemplo em (**?**), AGs foram utilizados para avaliar a velocidade de convergência e as propriedades de desvio na otimização da energia de acoplamento molecular produzidas pelo *software* de código-livre Autodock 3.05 do *Scripps Research Institute* (**?**). Em (**?**), os AGs foram utilizados para a redução/seleção de características de um modelo QSAR para a previsão da soludibilidade de um fármaco na água.

Nos últimos anos, a PG tem-se tornado uma escolha popular em modelação QSAR e em aplicações biomédicas relacionadas. Por exemplo, em (**?**), a PG foi utilizada para classificar moléculas em termos da sua biodisponibilidade; em (**???**) e (**?**) foi utilizada para a previsão da biodisponibilidade oral, do grau de ligação as proteínas do plasma e da toxicidade induzida em fármacos. Em (**?**), a PG foi utilizada para gerar uma relação funcional entre um conjunto de descritores moleculares e a sua energia de acoplamento molecular.

Em (**?**), a PG é aplicada a dados de perfis de expressões de câncer para selecionar os atributos (*features*) dos genes e construir classificadores moleculares, através da integração matemática dos mesmos genes. A PG foi também utilizada em conjunto com a base de dados NCI-60 para procurar uma relação entre as expressões génicas e a sua resposta

[8] A cefalosporina é um antibiótico, semelhante a penincilina, utilizado no tratamento de infeções bacterianas.

[9] Sistemas fechados de complexidade muito alta em que a sua estrutura interna (neste caso fórmula matemática) é desconhecida.

aos medicamentos oncológicos (fluoruracila, fludarabina, floxuridina e citarabina) com o objetivo de determinar a probalidade de resistência aos mesmos.

A previsão eficaz dos parâmetros farmacocinéticos e outros componentes do processo de desenvolvimento de medicamentos foi também tratada no presente trabalho, estabelecendo uma comparação com os resultados apresentados na bibliografia, com realce aos descritos em (**?????**) e (**?**). Para tal, foram aplicadas abordagens diferentes, descritas no capítulo 4, com a intenção de avaliar a performance da PG na resolução do problema em questão e descobrir o quanto os resultados podem ser melhorados.

4

METODOLOGIA

Conteúdo

4.1. INTRODUÇÃO

O presente trabalho está focado na aplicação prática da PG aos problemas apresentados no capítulo 3. Para tal, neste capítulo são apresentados os dados e a forma como foram adquiridos. De seguida são apresentadas as diferentes versões de PG, os parâmetros de execução e a ferramenta utilizada.

4.2. DADOS

4.2.1. %F, %PPB e LD50

Os dados utilizados para a previsão de %F, LD50 e %PPB são os mesmos utilizados em (**??**) e (**?**). Originalmente os dados foram obtidos através de um conjunto de estruturas moleculares com os valores correspondentes de %F, LD50 e %PPB, e de uma base dados pública de fármacos e compostos semelhantes aprovados pela *Food and Drug Administration* (FDA)[1] (**?**). As estruturas químicas dos compostos são expressas em códigos (*Simplified Molecular Input Line Entry Specification* (SMILES)), que são caracteres que representam as estruturas moleculares bidimensionais dos compostos, os átomos e as suas ligações de forma concisa (**?**).

As bibliotecas de dados resultantes contêm 359 (respetivamente 234 e 131) moléculas com os valores de %F, LD50 e %PPB respetivamente. Os caracteres em código SMILES pertencentes a %F foram utilizados para calcular 241 descritores moleculares bidimensionais utilizando o *software* ADMET Predictor[2]. Os descritores moleculares da LD50 e %PPB foram calculados utilizando o *software* DRAGON[3], o que resultou em 627 e 626 descritores moleculares bidimensionais respetivamente. Sendo assim, os dados formam matrizes compostas por 359 (respetivamente 234 e 131) linhas e 242 (respetivamente 628 e 627) colunas. Cada linha é um vector com os valores dos descritores moleculares que identificam um fármaco; cada coluna representa um descritor molecular, exceto a última que contem os valores de %F, LD50 e %PPB conhecidos.

Os conjuntos de dados para treino e teste foram obtidos através de uma divisão ale-

[1] `www.fda.gov`
[2] `www.simulationplus.com`
[3] `www.talete.mi.it/products/dragon_description.htm`

atória. Para cada conjunto de dados, 70% das moléculas (ou linhas) foram selecionadas aleatoriamente com probabilidade uniforme e inseridas no conjunto de treino enquanto que o restante 30% ficou para o conjunto de teste.

4.2.2. Energia de acoplamento molecular

Os dados para a previsão da energia de acoplamento, previamente utilizados em (**?**), são um conjunto pequeno de moléculas virtuais de estrogênio-genisteína obtidos da base de dados do *Research Collabolatory for Structural Bioinformatics - Protein Data Bank* (RCSB-PDB) (**?**). Tal como explicado em (**?**), foram definidos alguns pontos de cruzamento com uma pequena base de dados de substituentes, obtendo um conjunto de 992 moléculas virtuais de genisteína. Posteriormente, as estruturas químicas resultantes foram otimizadas por meio de mecânica molecular utilizando o *software Molecular Operating Environment* (MOE) (**?**) e o campo de força molecular Merck (*Merck Molecular Force Field 94* (MMFF94)) (**?**) para calcular 267 descritores moleculares. Finalmente, para cada molécula (ligantes), foi calculado o valor da energia de acoplamento através do *software Discovery and Lead Optimization Systems* (DELOS), que é um ambiente para triagem virtual e simulações de acoplamento produzido pela empresa DELOS S.r.l (**?**).

O conjunto de dados resultante está composto por 992 moléculas de genisteína, cada uma com 267 descritores moleculares e os valores da energia de acoplamento. Assim, os dados finais formam uma matriz com 992 linhas e 268 colunas, onde cada linha representa uma molécula com 267 descritores e o seu valor da energia de acoplamento (última coluna).

Antes da construção dos modelos de PG, foi feita uma partição aleatória dos dados dando origem a um conjunto de dados de treino e outro de teste: 70% das moléculas foram selecionadas aleatoriamente com probabilidade uniforme e inseridas no conjunto de treino enquanto que o restante 30%, para o conjunto de teste.

4.2.3. Fludarabina

A fludarabina, é um fármaco utilizado no tratamento de leucemia linfocítica[4] e é um dos 118 fármacos que fazem parte da base de dados NCI-60 do NCI. O NCI-60 consiste

[4]A leucemia linfolítica é um tipo de câncer que afeta o sangue e a medula óssea.

em 60 linhagens de células tumorais derivadas de pacientes com os seguintes 9 diferentes tipos de câncer: colo-retal, renal, do ovário, da mama, da próstata, do sistema nervoso central, leucemias e melanoma (**?**).

A expressão génica da fludarabina foi medida em 9703 genes mas apenas 1375, que apresentaram forte variação entre as linhagens celulares, foram retidos para análise. Os dados da expressão génica para os p genes formam uma matriz X $[n$ x $p]$, onde n é a quantidade de amostras (60). Cada elemento da matriz x_{ij}, $i = 1, 2, \ldots, n$ e $j = 1, 2, \ldots, p$ representa o nível de expressão do gene j na amostra i. Dos 1400 compostos químicos testados em cada uma das linhages celulares, apenas em 118 foi possível conhecer o mecanismo de acção do fármaco (**?**). A acção dos fármacos foi estimada através da medição da sua capacidade de inibir o crescimento do cancêr 48 horas após tratamento, utilizando o teste de *sulforrodamina B*[5].

O NCI-60 é composto por duas matrizes: a matriz de atividade (A) e a matriz alvo (T). A matriz T contém os dados das expressões génicas e a matriz A contém as respostas aos tratamentos fármacológicos. Particularmente, a matriz A representa o padrão de atividade dos 118 fármacos nas 60 linhagens de celulas cancerígenas. A atividade de um fármaco para uma dada linhagem é definida como sendo a concentração logarítmica necessária para reduzir a taxa de crescimento para 50%. A resposta à um fármaco em qualquer linhagem celular é denotado por R_j com $j = 1, 2, \ldots, 60$ e é calculado por $R_j = -log_10GI_50$, onde GI_50 é a concentração do fármaco que causa uma inibição de 50% ao crescimento da célula. A matriz T mostra as expressões (ou concentrações) dos 1375 genes sobre as mesmas linhagens celulares (**?**).

O conjunto de dados final consiste na matriz T do NCI-60 mais uma linha da matriz A, correspondente ao padrão de acção de um fármaco em particular. O nosso objetivo é encontrar uma relação matemática entre o perfil das expressões génicas e o padrão de atividade da fludarabina (**?**). Os conjuntos de dados foram repartidos aleatoriamente e com probabilidade uniforme em 70% para o conjunto de treino e o restante 30% para o conjunto de teste.

[5] Sulforrodamina B é um teste ou ensaio utilizado na avaliação da citoxicidade e proliferação de células (**?**).

4.3. CONFIGURAÇÕES DE PROGRAMAÇÃO GENÉTICA

Os parâmetros utilizados nos experimentos são indicados na Tabela 4.1.

Parâmetro	**Valor**
Conjunto de funções	$\{+, -, *, \%\}$
Tamanho da população	100
Inicialização	*Ramped half-and-half*
Profundidade máxima da árvore	17 (**?**)
Seleção	Torneio de tamanho 10
Cruzamento	Cruzamento de subárvore
Mutação	Mutação de subárvore
Taxa de cruzamento	0.95
Taxa de mutação	0.1
Número máximo de gerações	200
Número de execuções	30

Tabela 4.1 – Configuração utilizada nas diferentes versões de PG

O conjunto de funções F utilizado é composto pelas operações matemáticas primitivas $F = \{+, *, -, \%\}$, onde % representa a divisão protegida, ou seja, retorna 1 quando o denominador é igual a 0. Por sua vez, o conjunto de terminais T é composto por n variáveis reais, onde n é o número de colunas no conjunto de dados, isto é o número de descritores moleculares de cada composto. O melhor indivíduo k, retornado a cada geração pelas funções de *fitness* descritas abaixo, será utilizado para calcular a raíz quadrada do erro quadrático médio (*Root Mean Squared Error* (RMSE)) no conjunto de teste. Posteriormente, utilizaremos este valor para estabelecer uma comparação com as outras versões de PG.

4.3.1. Programação Genética padrão (RMSE-GP)

A primeira configuração de PG utilizada é a versão padrão tal como definida por Koza (**?**). O *fitness* de cada indivíduo foi definido como sendo o RMSE calculado sobre o conjunto de treino. Por exemplo, dado o indivíduo k que produz a previsão ou estimativa $\hat{y}$ do valor real esperado y na i-gésima molécula do conjunto de treino, definimos o seu *fitness* $fit(k)$ como sendo:

$$fit(k) = \sqrt{\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{m}} \tag{4.1}$$

onde m é o número de moléculas no conjunto treino (casos de *fitness*) e y_i é o valor correto observado para a molécula representada pelos dados na i-gésima linha do conjunto de treino.

4.3.2. Escalonamento linear (LS-GP)

A segunda configuração de PG difere da anterior na função de *fitness* aplicada. Neste caso, a *fitness* é obtida a partir do escalonamento linear (*Linear Scaling* (LS)) aplicado ao RMSE tal como detalhado em (**?**). O escalonamento linear consiste em calcular o ``declive'' e a ``ordenada'' da fórmula codificada pelo indivíduo da PG. Por exemplo, dado que $\hat{y}_i$ é o resultado da previsão para o indivíduo k na i-gésima linha do conjunto de treino, uma regressão linear pode ser aplicada aos valores de y utilizando a Equação (4.2) e a Equação (4.3):

$$b = \frac{\sum_{i=1}^{m}[(y_i - \overline{y})(\hat{y}_i - \overline{\hat{y}})]}{\sum_{i=1}^{m}(\hat{y}_i - \overline{\hat{y}})} \tag{4.2}$$

$$a = \overline{y} - b\overline{\hat{y}} \tag{4.3}$$

onde m é o número de casos de *fitness*, e $\overline{\hat{y}}$ e $\overline{y}$ são a média dos valores estimados e a média dos valores reais, respetivamente. Estas expressões calculam o declive e a ordenada dos valores de $\overline{\hat{y}}$ de formas que o RMSE de y e $a + b\hat{y}$ é minimizada. Os operadores definidos pela Equação (4.2) e pela Equação (4.3) podem ser calculados em tempo linear $O(N)$. Depois disso, qualquer medida de erro pode ser calculada com a fórmula escalonada $a + b\hat{y}$. Para o presente trabalho foi utilizado o RMSE:

$$fit(k) = RMSE(y, a + b\hat{y}) = \sqrt{\frac{\sum_{i=1}^{m}(a + b\hat{y}_i - y_i)^2}{m}} \tag{4.4}$$

Se $a \neq 0$ e $b \neq 1$, o processo definido acima reduz o RMSE para qualquer $\hat{y}$ (**?**). Por calcular eficientemente o declive e a ordenada para cada indivíduo, a sobrecarga na procura destas duas constantes é eliminado da execução. Assim, a PG pode pesquisar

pela expressão cuja forma é mais similar a função alvo. A eficácia do escalonamento linear em problemas de regressão simbólica é amplamente demonstrada em (**?**) e foi utilizada com sucesso em (**?**) e (**?**).

4.3.3. Coeficiente de correlação (PCCN-GP)

O Coeficiente de Correlação de *Pearson* (PCC) (ou r) é uma medida do grau de relação linear entre duas variáveis quantitativas. Este coeficiente varia entre os valores -1 e 1. O valor 0 (zero) significa que não há relação linear, o valor 1 indica uma relação linear perfeita e o valor -1 indica uma relação linear perfeita inversa, ou seja, quando uma das variáveis aumenta a outra diminui. Quanto mais próximo o coeficiente estiver de 1, mais forte é a associação linear entre as duas variáveis (**?**).

Como exemplo, dado o valor previsto $\hat{y}$ e os valores reais y para um determinado indivíduo k, o coeficiente de correlação entre as duas variáveis é dado pela Equação (4.5):

$$r(k) = \frac{\sum_{i=1}^{m}(\hat{y}_i - \overline{\hat{y}})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m}(\hat{y}_i - \overline{\hat{y}})^2}\sqrt{\sum_{i=1}^{m}(y_i - \overline{y})^2}} \tag{4.5}$$

onde m é o número de casos de *fitness*. Caso o indíviduo k faça boas previsões dos valores de y, $r(k)$ estará o mais aproximado possível de 1. Este comportamento é o ideal para avaliar a correlação entre as previsões de k e os valores reais de y, exceto pelo facto de que na configuração de PG consideram-se melhores indivíduos aqueles com valores menores de *fitness*. Sendo assim, utilizou-se como função de *fitness* o Coeficiente de Correlação de *Pearson* Normalizado (PCCN), dado pela Equação (4.6):

$$fit(k) = PCCN(k) = \frac{-r(k) + 1}{2} \tag{4.6}$$

onde $r(k)$ é o PCC calculado pela Equação (4.5). Dessa forma, o intervalo de valores ficará entre 0 e 1, e os melhores indivíduos terão o valor de *fitness* mais próximo de 0.

4.3.4. Erro médio absoluto dimensionado (MASE-GP)

O erro médio absoluto dimensionado (MASE) é uma medida da precisão de previsões em séries temporais e difere das outras medidas de erro dependentes de escala e

sensíveis à *outliers* (por exemplo: o RMSE e o *Mean Absolute Percentage Error* (MAPE)). Uma descrição mais completa das vantagens do MASE sobre as outras medidas de erro é discutida em (**?**). Originalmente a MASE é calculada pela Equação (4.7):

$$MASE = \frac{1}{m}\sum_{t=1}^{m}\left(\frac{|e_t|}{\frac{1}{m-1}\sum_{i=2}^{m}|y_i - y_{i-1}|}\right) = \frac{\sum_{t=1}^{m}|e_t|}{\frac{m}{m-1}\sum_{i=2}^{m}|y_i - y_{i-1}|} \tag{4.7}$$

em que,

$$e_t = y_t - \hat{y}_t \tag{4.8}$$

onde e_t é o erro de previsão, y_t é o valor real e $\hat{y}_t$ é o valor da previsão num determinado período t. O denominador na Equação (4.7) é a média do erro de previsão do ``método de previsão ingênua'' de uma etapa, que utiliza o valor real do período anterior como previsão (**?**).

Considerando que $\hat{y}_i$ é o resultado da previsão de y para um dado indivíduo k na i-gésima linha do conjunto de treino, podemos calcular o MASE pela Equação (4.9):

$$fit(k) = \frac{\sum_{i=1}^{m}|y_i - \hat{y}_i|}{\frac{m}{m-1}\sum_{i=2}^{m}|y_i - y_{i-1}|} \tag{4.9}$$

4.3.5. Coeficiente de determinação (R2-GP)

O coeficiente de determinação R^2 (ou coeficiente de correlação múltipla) é uma medida da proporção de variância de um modelo de regressão e é muito utilizada na construção de modelos de previsão em estatística (**?**). De forma geral, o coeficiente de determinação é dado pela seguinte equação:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{4.10}$$

onde y_i são os valores observados, $\hat{y}_i$ são os valores estimados e $\overline{y}$ é a média dos valores observados. O valor de R^2 indica a percentagem com que os valores de $\hat{y}_i$ aproximam-se aos de y_i e portanto, R^2 varia entre 0 e 1. No entanto, para a construção da função de *fitness*, interessa-nos transformar a Equação (4.10) de formas a que os melhores valo-

res estejam mais próximos de 0. Esta transformação, denotada por ${R_N}^2$, é obtida pela Equação (4.11):

$$ {R_N}^2 = -R^2 + 1 = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{4.11} $$

logo, para um indivíduo k que retorna $\hat{y}_i$ como o resultado da previsão de y_i temos que:

$$ fit(k) = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{4.12} $$

4.3.6. *Boosting* (B-GP)

O *boosting* é uma técnica de ML utilizada para melhorar a performance de um modelo de classificação por combinar, num só modelo, vários modelos de classificação de baixa performance no conjunto de treino (**?**). Dado um conjunto de treino $(x_1, y_1, \ldots, (x_m, y_m))$ onde cada x_i pertence ao conjunto de variáveis de entrada e y_i pertence ao conjunto de variáveis de saída, a técnica de *boosting* é executada T vezes, onde a cada execução é mantida uma distribuição (ou conjunto de pesos) sobre o conjunto de treino. O peso para o exemplo de treino i na execução t (onde $t = 1 \ldots T$) é denotado por $D_t(i)$. Inicialmente, os pesos são repartidos uniformemente entre os exemplos de treino ($D_1(i) = 1/m$), mas posteriormente os pesos dos exemplos classificados incorretamente são incrementados de formas a receberem mais ênfase nas execuções seguintes (**?**). Foi teoricamente provado em (**?**) que o erro no conjunto de treino da hipótese final é melhor que os das outras hipóteses sem *boosting*.

O *AdaBoost* foi uma das primeiras implementações do *boosting* para problemas de classificação binária, proposta em (**?**). Baseado em (**?**), Iba propôs uma versão do *AdaBoost* para problemas de regressão utilizando a PG (**?**). Iba mantém a função de *fitness* como na PG padrão e a distribuição é utilizada para selecionar os exemplos (casos de *fitness*) para gerar um novo conjunto de treino, a cada execução do *boosting*. A probabilidade de um exemplo ser selecionado é proporcional ao seu peso e qualquer exemplo pode ser selecionado uma ou mais vezes, até que o conjunto de treino esteja completo. A PG padrão é então executada com o novo conjunto de treino para calcular a função associada com o *boosting* atual.

A implementação de *boosting* utilizada no presente trabalho (ver Algoritmo 4.1) é uma adaptação do algoritmo *GPBoost* introduzido em (**?**), onde foi utilizada como função de *fitness* a soma da diferença absoluta ponderada pelos valores da distribuição. Dessa forma, a avaliação do *fitness* leva em consideração a contribuição de cada exemplo (**?**).

entrada: Conjunto de treino $S = \{(x_1, y_1, \ldots, (x_m, y_m))\}; x_i \in \mathbb{X}, y_i \in \mathbb{R}$
saída: Hipótese final: F$(x) = min\{y \in \mathbb{R} : \sum_{t:\hat{y}_t \leq y} \log(1/\beta_t) \geq \frac{1}{2} \sum_{t=1}^{m} \log(1/\beta_t)\}$

1: Seja D_1 a distribuição para a iteração $t = 1$
2: $D_1(i)$ é o peso para o exemplo (x_i, y_i)
3: Inicializa $D_1(i) \leftarrow 1/m$ para todos $(x_i, y_i) \in S$
4: **para** $t \leftarrow 1 : T$ **faça**
5: Executa a PG em D_t com a seguinte função de *fitness*:
6: $fit = \sum_{i=1}^{m}(|\hat{y}_i - y_i| * D_t(i)) * m$
7: O melhor indivíduo é denotado por $\hat{y}_t$
8: Calcula a perda para cada exemplo: $L_i = \frac{|\hat{y}_t - y_i|}{max_{i=1\ldots m}|\hat{y}_t - y_i|}$
9: Calcula a perda média: $\overline{L} = \sum_{i=1}^{m} L_i D_i$
10: Seja $\beta_t = \frac{\overline{L}}{1-\overline{L}}$, a confiança dada a $\hat{y}_t$
11: Atualiza a distribuição $D_{t+1}(i) = \frac{D_t(i) * \beta_t^{1-L_i}}{Z_t}$
12: onde Z_t é um fator de normalização de formas a que D_{t+1} seja uma distribuição
13: **fim para**

Algoritmo 4.1 – Algoritmo de *Boosting*

4.3.7. Pesquisa de novidade (NS-GP)

Na PG tradicional (e na CE no geral) a pesquisa pelos melhores indivíduos é orientada por uma funcão de *fitness* (ou função-objetivo) explicitamente definida. No entanto, as funções de *fitness* geralmente sofrem do problema da deceção (**?**). A deceção ocorre quando o objetivo a alcançar é muito ambicioso ou a paisagem de *fitness*, construída a partir da função de *fitness*, contém ótimos locais que podem ser retornados como solução (**?**). Adicionalmente, as funções de *fitness* tradicionais não recompensam os indivíduos com características importantes para se chegar a solução ótima (*building blocks*) (**?**).

A pesquisa por novidade minimiza o problema da deceção e das paisagens de *fitness* enganosas por substituir a pesquisa do *objetivo* pela pesquisa de *novidades*. O comportamento de um indivíduo é comparado com o de outros anteriormente avaliados, armazenados num arquivo. Cada indivíduo recebe um valor de novidade que encapsula

o seu comportamento quando comparado com o de outros indivíduos. Os comportamentos novos ou diferentes são premiados e o espaço de comportamentos diferentes torna-se maior e mais complexo até ser satisfeito um critério de paragem (e.g.: uma solução é encontrada, etc.) (**?**).

Para executar uma pesquisa por novidade é necessário definir um *descritor comportamental* e uma medida de dispersão. Em (**?**), a medida de novidade definida utiliza um arquivo de comportamentos e premeia os indivíduos com comportamentos em áreas mais dispersas. Para calcular a dispersão ρ, calcula-se a distância média de um ponto x aos k-vizinhos mais próximos através da equação:

$$\rho(x) = \frac{1}{k} \sum_{i=0}^{k} dist(x, \mu_i) \tag{4.13}$$

onde x é o indivíduo, ρ é o valor da dispersão, k é um número inteiro arbitrário encontrado através de experimentação e μ_i é o i-gésimo vizinho mais próximo de x (ou o que apresenta o comportamento mais semelhante a x). Por sua vez, $dist$ é uma medida que determina a diferença de comportamentos entre x e μ_i no espaço de pesquisa, dependente do domínio do problema. Geralmente, para se definir o *descritor comportamental* e a medida da diferença de comportamentos é necessário algum conhecimento especializado sobre o problema a resolver (**?**).

Doncieux e Mouret desenvolveram alguns *descritores comportamentais* independentes do problema (**?**). No entanto, estes descritores fazem parte do contexto de problemas de RE e não são adaptáveis à problemas de regressão simbólica. Trujillo et al. apresentaram o único *descritor comportamental* para problemas de regressão simbólica publicados até ao presente (**?**). O descritor implementado em (**?**) não descreve uma medida de comportamento completamente livre da pesquisa por *fitness*, uma vez que utiliza uma medida do erro de cada indivíduo ao calcular a medida de novidade. Segundo os autores, uma vez que o objetivo da regressão simbólica é simplesmente minimizar o erro entre o valor de saída real e um valor estimado, o descritor comportamental deve considerar o conceito de ``erro'' de alguma forma.

Para o presente trabalho, foi criado um novo descritor comportamental diferente do introduzido em (**?**), pois é implementado sem levar em conta uma medida de erro re-

lacionada com a *fitness* dos indivíduos, onde se deseja que os melhores indivíduos possuam um valor de *fitness* menor. O descritor comportamental ρ de um indivíduo K_i é calculado levando em consideração os indivíduos das gerações anteriores e os casos de *fitness* $X = \{(x_1, f(x_1)), \ldots, (x_L, f(x_L))\}$.

Na primeira geração, o descritor comportamental de um indivíduo é simplesmente o seu *fitness* bruto calculado tal como definido em 2.4.1. O indivíduo com maior *fitness* bruto é considerado o que ``apresenta maior novidade'' e é armazenado no ficheiro. Nas gerações posteriores, calcula-se a distância média do *fitness* bruto de $K_i$ aos indivíduos da geração anterior. O indivíduo com a maior distância média é considerado o que ``apresenta maior novidade'' e é armazenado no ficheiro. Este processo é descrito no Algoritmo 4.2.

entrada: $x \in \mathbb{R}^n$, $f(x)$: Função simbólica, K: Função PG
saída: ρ: descritor comportamental

se $gen = 0$ **então**
  para $i = 1 : pop$ **faça**
    $fit(K_{i,0}) \leftarrow \sum_{j=1}^{L} |f(x_j) - K_i(x_j)|$
    $\rho(i) \leftarrow fit(K_{i,0})$
  fim para
  ARMAZENANOFICHEIRO($max(\rho(i))$)
fim se
para $t \leftarrow 1 : gen$ **faça**
  para $i \leftarrow 1 : pop$ **faça**
    $fit(K_{i,t}) \leftarrow \sum_{j=1}^{L} |f(x_j) - K_i(x_j)|$
    $\rho(i) \leftarrow \frac{1}{L} \sum_{j=1}^{L} [fit(K_{i,t}) - fit(K_{i,t-1})]^2$
  fim para
  ARMAZENANOFICHEIRO($max(\rho(i))$)
fim para

Algoritmo 4.2 – Descritor Comportamental para a Pesquisa de Novidade

Esta abordagem é ligeiramente diferente da ideia geral do algoritmo apresentado em (**?**) e (**?**) uma vez que o cáculo do descritor comportamental incorpora também o cálculo de uma medida de dispersão. No final da execução, é retornado como solução o indivíduo com maior *fitness* presente no arquivo. A performance dos indivíduos é avaliada sobre o conjunto de dados de teste utilizando o método padrão apresentado em 4.3.1.

4.4. GPLAB - UMA FERRAMENTA DE PROGRAMAÇÃO GENÉTICA

O GPLAB é um pacote de *software* para PG escrito em MATLAB. A sua arquitetura é extramente modular, o que permite que vários utilizadores possam facilmente acrescentar funcionalidades ao pacote na forma *Plug and Play* (ligar e utilizar). Além disso, o GPLAB pode ser utilizado por pessoas com pouca experiência em PG desde que possuam algum conhecimento de MATLAB (**?**).

Para além dos ficheiros que compõem o núcleo do aplicativo, estão disponíveis outros ficheiros de demonstração em que são resolvidos quatro problemas de referência em PG, nomeadamente: a regressão simbólica, a formiga artificial no trilho de Santa Fé, o problema da paridade e o problema do multiplexador. Estes problemas são devidamente descritos em (**?**).

O GPLAB incorpora alguns dos últimos avanços em PG tais como: técnicas para controle de *bloat*, adaptação em tempo real das probabilidades dos operadores e outros (**?**). As funções de *fitness* descritas na secção 4.3 (exceto a função RMSE gentilmente cedida pela Dr. Sara Silva) foram desenvolvidas no âmbito do presente trabalho e acrescentadas ao GPLAB.

4.4.1. Ambiente computacional

Para a execução das diferentes versões da PG descritas na secção 4.3 foi utilizada a versão 3 do GPLAB. Atualmente, esta é a ultima versão disponível no site principal da ferramenta e incorpora algumas alterações significativas. A descrição destas alterações está fora do âmbito do presente trabalho. As versões de PG foram executadas sobre a versão R2003a do Matlab de 64-bit, divididas entre 3 computadores portatéis com as seguintes especificações:

- **MacBook Air**
  - **Processador**: 1.6 GHz, Intel Core i5
  - **Memória**: 4 GB DDR3
  - **Sistema operativo**: OS X 10.9.1

- **Sony Vaio**
  - **Processador**: 2.26 GHz, Intel Core 2 Duo CPU
  - **Memória**: 4 GB DDR2
  - **Sistema operativo**: Ubuntu 12.04.4 LTS
- **Lenovo Thinkpad**
  - **Processador**: 2.50 GHz, Intel Core i5
  - **Memória**: 4 GB
  - **Sistema operativo**: Windows 7 Ultimate SP1

5

RESULTADOS E DISCUSSÃO

Conteúdo

5.1. INTRODUÇÃO

Nesta secção apresentamos os resultados obtidos pela execução das diferentes versões de PG implementadas para os diferentes problemas de previsão apresentados no capítulo 4. Para cada versão de PG foram feitas 30 execuções sobre os conjuntos de dados de treino. Os melhores indivíduos encontrados nas 200 gerações para cada execução, foram posteriormente utilizados para calcular o *fitness* no conjunto de dados de teste utilizando a medida do RMSE. As próximas secções reportam: a mediana do *fitness* de teste dos melhores indivíduos, o *fitness* dos melhores indivíduos no conjunto de teste, a mediana do tamanho dos melhores indivíduos e a percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos durante as 30 execuções independentes.

5.2. RESULTADOS

5.2.1. %F

A Figura 5.1 ilustra a mediana do *fitness* de teste dos melhores indivíduos num total de 30 execuções independentes. Pode-se verificar que as versões RMSE-GP, MASE-GP e R2-GP apresentam as melhores medianas ao longo das 200 gerações. Por sua vez, a versão B-GP apresenta uma mediana que piora durante as primeiras 125 gerações e depois mantêm-se constante até ao final. O PCCN-GP e LS-GP apresentam as piores medianas do *fitness* de teste e é de notar que se mantêm constantes durante as 200 gerações.

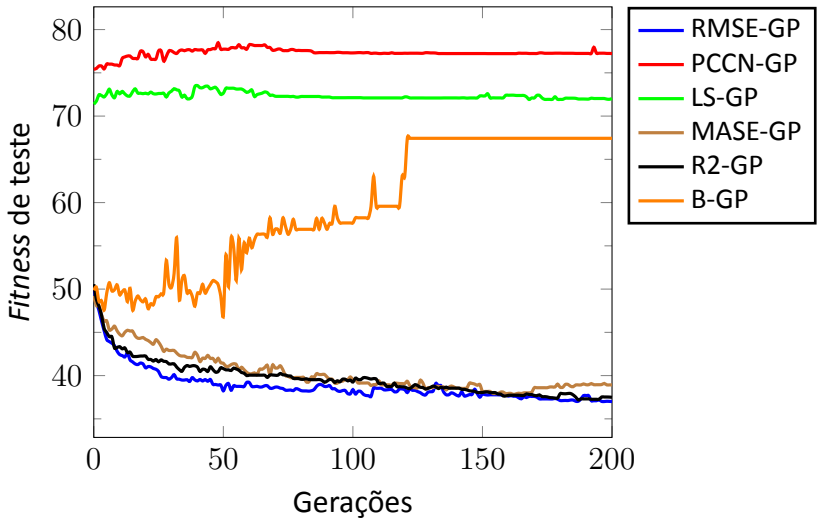

Figura 5.1 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino

Na Figura 5.2, são ilustrados à parte os resultados para o NS-GP devido a enorme diferença de escala (na ordem de 10^4) comparando com as versões anteriores (na ordem das dezenas). Nesta figura, a mediana no *fitness* de teste decresce rapidamente nas primeiras gerações, e depois mantém-se constante até a última geração. Este comportamento ``esquisito'' poderá dever-se a natureza do algoritmo apresentado na secção 4.3.7, em que se substitui a procura do melhor indivíduo (com *fitness* mais baixa) pela procura do indíviduo com maior novidade.

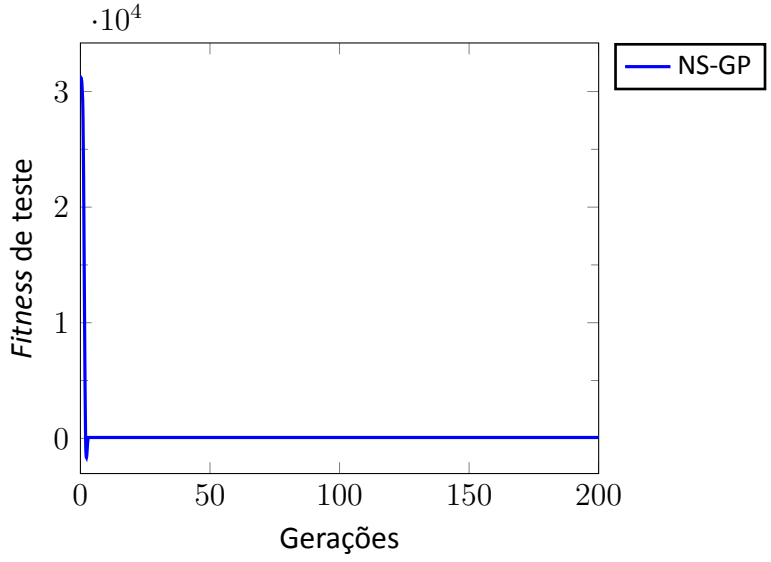

Figura 5.2 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino (NS-GP)

A Tabela 5.1 apresenta o *fitness* de teste, a média do *fitness* de teste e o desvio padrão do *fitness* do melhor indivíduo encontrado ao longo das 200 gerações em 30 execuções independentes no conjunto de treino. O melhor *fitness* de teste é apresentado pelo melhor indivíduo do MASE-GP, seguido pelo de RMSE-GP e R2-GP. De maneira geral, estas versões de PG também apresentam os melhores *fitness* médio e desvio padrão comparando com as outras versões. Os valores altos do PCCN-GP indicam a fraca correlação linear entre as soluções encontradas e as soluções reais. Ainda assim, os valores de PCCN-GP são piores que os apresentados por NS-GP.

	Fitness	***Fitness* médio**	**Desvio padrão**
RMSE-GP	28.5968	39.0236	8.7232
LS-GP	61.2560	90.7707	103.5719
PCCN-GP	69.5299	80.9041	10.4046
MASE-GP	28.5469	38.9985	6.4013
R2-GP	29.9587	37.8471	4.7335
B-GP	35.7834	66.5439	19.7646
NS-GP	54.1312	78.6152	38.6170

Tabela 5.1 – Melhor *fitness* de teste, média do *fitness* de teste e desvio padrão do *fitness* de teste

Na Figura 5.3 é ilustrada a mediana da quantidade de nós (ou tamanho) dos melhores indivíduos encontrados no conjunto de treino ao longo das 200 nas 30 execuções independentes. No geral, os melhores indivíduos possuem um tamanho mediano entre 50 a 150 nós ao final das 200 gerações. O B-GP apresenta um comportamento curioso uma vez que a mediana do tamanho dos indivíduos produzidos por si pára de crescer na 50 geração e mantém-se constante desde a 120 geração até a última. Curiosamente, a mediana do tamanho dos melhores indivíduos de NS-GP mantém-se constante ao longo das 200 gerações.

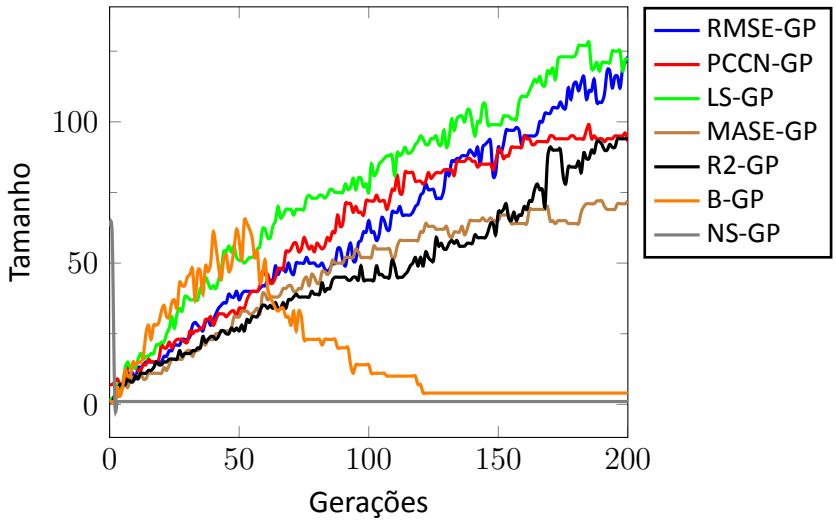

Figura 5.3 – Mediana do tamanho (nº de nós) do melhor indivíduo no conjunto de treino

Finalmente, reportamos a percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos nas 30 execuções para as diferentes versões de PG. Pelas tabelas 5.2 à 5.4 pode-se ter uma noção da combinação de variáveis que caracterizam os melhores indivíduos. Por exemplo, as variáveis x_{19}, x_{235}, x_4, x_{215}, x_{231}, x_{230}, x_{221} e x_{226} são utilizadas por 3 ou mais versões de PG o que indica a importância destes descritores moleculares para a previsão da %F.

B-GP	
Variável	**%**
x_{19}	3.85
x_{59}	3.15
x_2	3.15
x_{235}	2.8
x_4	2.8
x_{215}	2.1
x_{231}	2.1
x_{164}	1.75
x_{230}	1.75
x_{177}	1.75

LS-GP	
Variável	**%**
x_{221}	3.53
x_{215}	2.26
x_{96}	2.15
x_{200}	2.04
x_{223}	1.82
x_{240}	1.77
x_{105}	1.66
x_{219}	1.66
x_{226}	1.66
x_{175}	1.6

MASE-GP	
Variável	**%**
x_{25}	9.43
x_{30}	4.39
x_{27}	4.13
x_{230}	4.13
x_{231}	3.6
x_4	2.69
x_{215}	2.69
x_{19}	2.62
x_{203}	2.55
x_{235}	2.49

Tabela 5.2 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

PCCN-GP		RMSE-GP		R2-GP	
Variável	**%**	**Variável**	**%**	**Variável**	**%**
x_{220}	5.55	x_{177}	6.16	x_{228}	8.27
x_{213}	3.28	x_{19}	5.46	x_{19}	7.69
x_{9}	2.33	x_{180}	3.91	x_{38}	3.81
x_{71}	2.27	x_{38}	3.27	x_{226}	3.52
x_{215}	2.14	x_{230}	3.1	x_{235}	2.99
x_{149}	2.08	x_{24}	3.1	x_{29}	2.52
x_{22}	2.08	x_{231}	2.73	x_{241}	2.35
x_{221}	2.02	x_{5}	2.36	x_{57}	2.35
x_{105}	1.95	x_{226}	2.3	x_{94}	2.11
x_{51}	1.64	x_{17}	2.25	x_{220}	2.05

Tabela 5.3 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

NS-GP	
Variável	**%**
x_{82}	6.67
x_{94}	3.33
x_{33}	3.33
x_{114}	3.33
x_{221}	3.33
x_{44}	3.33
x_{233}	3.33
x_{206}	3.33
x_{13}	3.33
x_{4}	3.33

Tabela 5.4 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

5.2.2. %PPB

A Figura 5.4 ilustra a mediana do *fitness* de teste dos melhores indivíduos num total de 30 execuções independentes. Pode-se verificar que as versões RMSE-GP, MASE-GP e R2-GP apresentam as melhores medianas ao longo das 200 gerações. Por sua vez, a versão B-GP apresenta uma mediana que oscila sútilmente nas primeiras 10 gerações e posteriomente mantêm-se constante até ao final. O PCCN-GP e LS-GP apresentam as piores medianas do *fitness* de teste e é de notar que se mantêm ligeiramente constantes durante as 200 gerações.

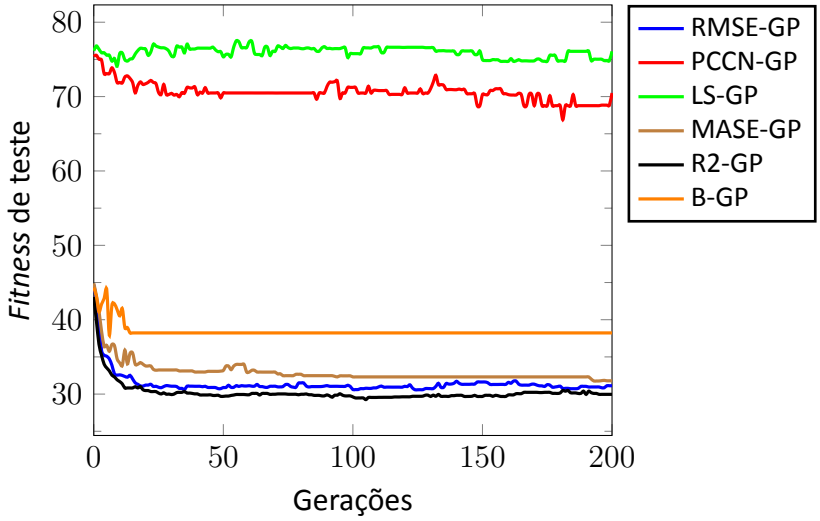

Figura 5.4 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino

Na Figura 5.5, são ilustrados à parte os resultados para o NS-GP devido a enorme diferença de escala (na ordem de 10^9) comparando com as versões anteriores (na ordem das dezenas). Nesta figura, os resultados da mediana no *fitness* de teste indicam que o *fitness* de teste decresce rapidamente nas primeiras gerações, e depois mantém-se constante até a última geração. Este comportamento ``esquisito'' poderá dever-se a natureza do algoritmo apresentado na secção 4.3.7 em que se substitui a a procura do melhor indivíduo (com *fitness* mais baixa) pela procura do indíviduo com maior novidade.

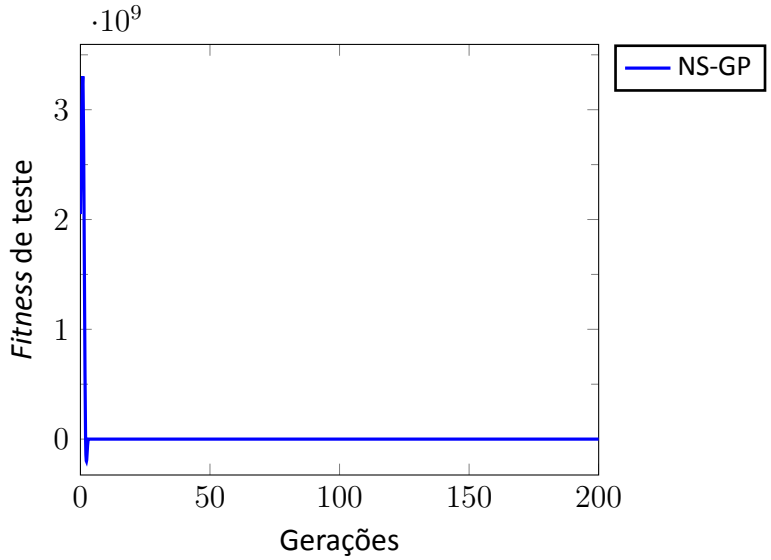

Figura 5.5 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino (NS-GP)

Na Tabela 5.5 são ilustrados o *fitness* de teste, a média do *fitness* de teste e o desvio padrão do *fitness* do melhor indivíduo encontrado ao longo das 200 gerações em 30 execuções independentes no conjunto de treino. O melhor *fitness* de teste é apresentado pelo melhor indivíduo do RMSE-GP e R2-GP. De maneira geral, estas versões de PG também apresentam os melhores *fitness* médio e desvio padrão comparando com as outras versões.

	Fitness	***Fitness* médio**	**Desvio padrão**
RMSE-GP	23.7581	31.1069	4.4390
LS-GP	45.7567	87.6411	42.6657
PCCN-GP	28.2897	77.6803	59.2576
MASE-GP	25.5349	51.9989	68.7031
R2-GP	23.9621	31.7268	6.1346
B-GP	30.6554	42.3080	9.5520
NS-GP	55.6753	92.4462	47.4815

Tabela 5.5 – Melhor *fitness* de teste, média do *fitness* de teste e desvio padrão do *fitness* de teste

A Figura 5.6 reporta a mediana da quantidade de nós (ou tamanho) dos melhores indivíduos encontrados no conjunto de treino ao longo das 200 nas 30 execuções independentes. Os valores dos tamanhos dos melhores indivíduos encontrados por RMSE-GP, MASE-GP e R2-GP, são muito aproximados ao longo das gerações. O B-GP e o NS-GP apresentam um comportamento curioso uma vez que a mediana do tamanho dos indivíduos produzidos por estes mantém-se constante a partir das primeiras gerações até a última geração.

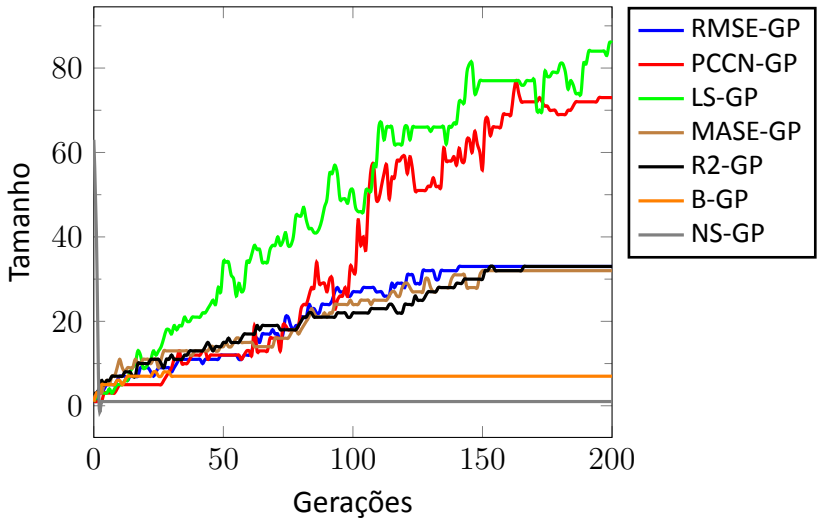

Figura 5.6 – Mediana do tamanho (nº de nós) do melhor indivíduo no conjunto de treino

Finalmente, as tabelas 5.6, 5.7 e 5.8 reportam a percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos, em 30 execuções independentes para as diferentes versões de PG. As variáveis x_{95}, x_{497} e x_{105} são utilizadas por 2 ou mais versões de PG.

B-GP	
Variável	**%**
x_{462}	5.33
x_{480}	2.96
x_{471}	2.37
x_{495}	2.37
x_{497}	2.37
x_{502}	2.37
x_{449}	2.37
x_{496}	1.78
x_{467}	1.78
x_{465}	1.78

LS-GP	
Variável	**%**
x_{211}	4.77
x_{47}	2.62
x_{332}	2.46
x_{375}	2.38
x_{400}	1.92
x_{383}	1.54
x_{402}	1.54
x_{105}	1.46
x_{263}	1.46
x_{17}	1.46

MASE-GP	
Variável	**%**
x_{325}	3.3
x_{267}	3.3
x_{363}	3.04
x_{353}	2.77
x_{326}	2.64
x_{95}	2.64
x_{212}	2.38
x_{563}	2.25
x_{497}	1.98
x_{228}	1.85

Tabela 5.6 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

PCCN-GP		RMSE-GP		R2-GP	
Variável	**%**	**Variável**	**%**	**Variável**	**%**
x_{24}	5.61	x_{213}	4.71	x_{536}	3.32
x_{351}	4.11	x_{344}	4.42	x_{131}	3.08
x_{614}	3.95	x_{349}	3.83	x_{220}	2.84
x_{263}	3.87	x_{343}	3.68	x_{95}	2.13
x_{309}	3.72	x_{479}	3.09	x_{405}	2.01
x_{335}	3.56	x_{76}	2.36	x_{530}	2.01
x_{221}	2.92	x_{71}	2.06	x_{494}	1.78
x_{302}	2.92	x_{492}	1.91	x_{379}	1.78
x_{404}	2.69	x_{93}	1.91	x_{317}	1.66
x_{105}	2.61	x_{568}	1.77	x_{589}	1.66

Tabela 5.7 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

NS-GP	
Variável	**%**
x_{286}	6.67
x_{344}	3.33
x_{523}	3.33
x_{238}	3.33
x_{409}	3.33
x_{307}	3.33
x_{161}	3.33
x_{585}	3.33
x_{36}	3.33
x_{451}	3.33

Tabela 5.8 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

5.2.3. LD50

A Figura 5.7 ilustra a mediana do *fitness* de teste dos melhores indivíduos, num total de 30 execuções independentes. Pode-se verificar que as versões RMSE-GP, MASE-GP e R2-GP apresentam as melhores medianas ao longo das 200 gerações. O PCCN-GP e LS-GP apresentam as piores medianas do *fitness* de teste e é de notar que se mantêm ligeiramente constantes durante as 200 gerações. Por sua vez, a versão B-GP apresenta uma mediana que oscila bastante nas primeiras 100 gerações e mantêm-se sútilmente constante até a última geração.

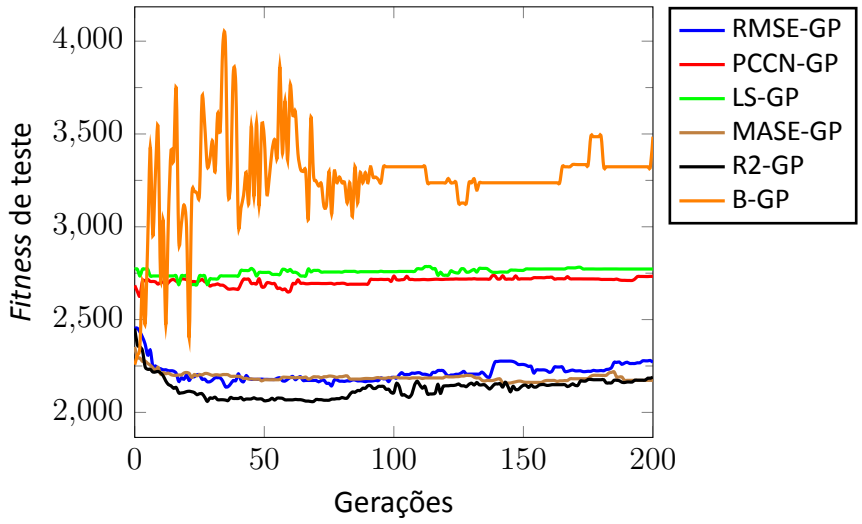

Figura 5.7 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino

Na Figura 5.8, são ilustrados à parte os resultados para o NS-GP devido a enorme diferença de escala (na ordem de 10^{10}) comparando com as versões anteriores (na ordem das unidades de milhar). Nesta figura, os resultados da mediana no *fitness* de teste apresentam um comportamento esquisito pois decrescem rapidamente na geração inicial, e de seguida mantém-se constante até a última geração.

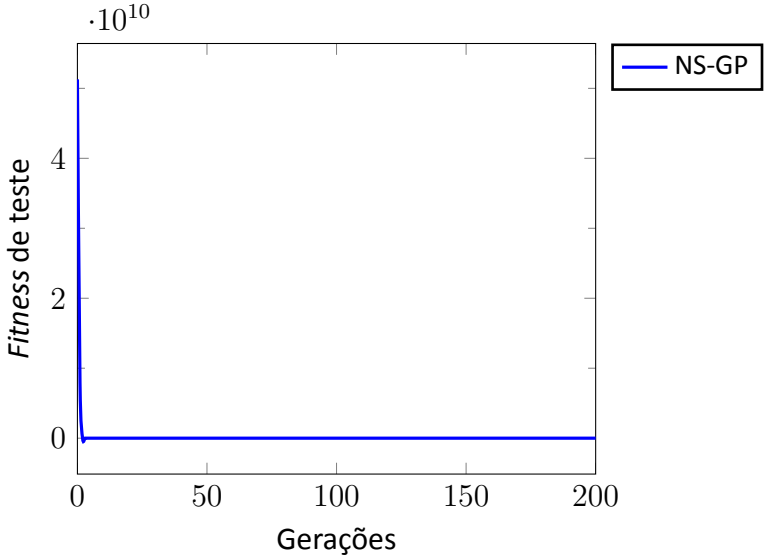

Figura 5.8 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino (NS-GP)

Na Tabela 5.9 são ilustrados o *fitness* de teste, a média do *fitness* de teste e o desvio padrão do *fitness* do melhor indivíduo encontrado ao longo das 200 gerações em 30

execuções independentes no conjunto de treino.

O melhor *fitness* de teste é apresentado pelo melhor indivíduo do R2-GP, seguido por MASE-GP e B-GP. Por outro lado, o RMSE-GP apresenta os melhores *fitness* médio e desvio padrão comparando com as outras versões. Curiosamente, o *fitness* encontrando por NS-GP é melhor que o de LS-GP. O PCCN-GP apresentou os piores resultados e isto indica a fraca correlação linear entre as soluções encontradas e as soluções reais.

	Fitness	***Fitness* médio**	**Desvio padrão**
RMSE-GP	1825.2780	2324.0526	397.5173
LS-GP	2274.3069	2940.2850	919.9910
PCCN-GP	1975.6890	$6.0401e+22$	$3.3083e+23$
MASE-GP	1726.8142	2358.5252	721.8147
R2-GP	1716.3409	2338.8859	639.3385
B-GP	1771.9797	4512.8091	3371.1339
NS-GP	2070.3043	4203.5076	8905.5740

Tabela 5.9 – Melhor *fitness* de teste, média do *fitness* de teste e desvio padrão do *fitness* de teste

A Figura 5.9 reporta a mediana do tamanho do melhores indivíduos encontrados no conjunto de treino ao longo das 200 nas 30 execuções independentes. O tamanho dos melhores indivíduos encontrados por RMSE-GP, PCCN-GP, LS-GP, MASE-GP e R2-GP varia entre 60 à 100 nós. O B-GP e o NS-GP apresentam um comportamento diferente das outras versões de PG. Para o B-GP, a mediana do tamanho dos indivíduos cresce até aproximadamente a 50^{a} geração, e permanece relativamente constante (entre 20 à 25 nós) até a última geração. Quanto ao NS-GP, os valores decrescem rápidamente nas gerações iniciais e permanecem constantes até à última geração.

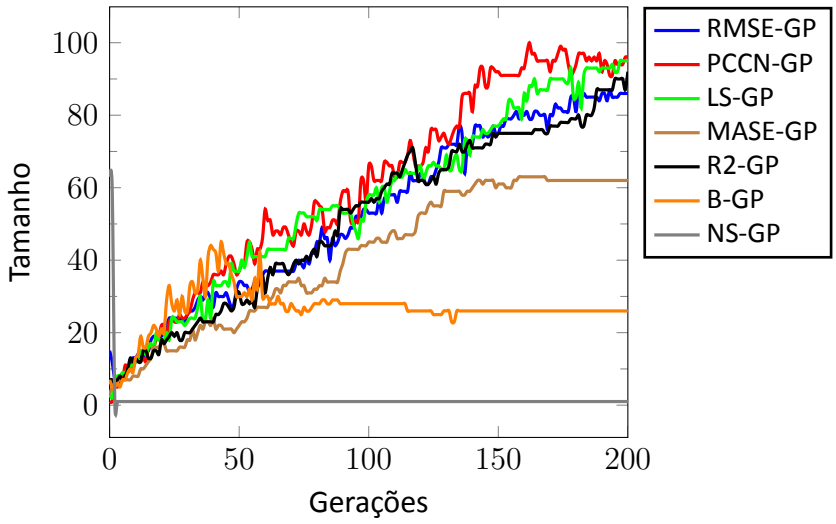

Figura 5.9 – Mediana do tamanho (nº de nós) do melhor indivíduo no conjunto de treino

Finalmente, reportamos a percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos nas 30 execuções para as diferentes versões de PG, sobre o conjunto de dados da LD50. Pelas tabelas 5.10, 5.11 e 5.12 pode-se notar a presença repetida das variáveis x_{11} (em LS-GP, MASE-GP, PCCN-GP), x_{151} (em MASE-GP, RMSE-GP, NS-GP) e x_{51} (em MASE-GP, RMSE-GP, R2-GP).

B-GP	
Variável	**%**
x_{476}	8.62
x_{179}	6.27
x_{494}	6.05
x_{493}	4.48
x_{2}	4.48
x_{468}	4.26
x_{467}	3.81
x_{463}	2.58
x_{374}	2.24
x_{112}	1.9

LS-GP	
Variável	**%**
x_{11}	2.97
x_{551}	1.78
x_{593}	1.72
x_{557}	1.65
x_{110}	1.65
x_{321}	1.52
x_{107}	1.39
x_{263}	1.25
x_{581}	1.25
x_{31}	1.19

MASE-GP	
Variável	**%**
x_{145}	7.74
x_{151}	3.73
x_{138}	2.29
x_{51}	2.29
x_{594}	2.2
x_{143}	2.1
x_{11}	1.82
x_{142}	1.72
x_{128}	1.53
x_{93}	1.34

Tabela 5.10 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

PCCN-GP		RMSE-GP		R2-GP	
Variável	**%**	**Variável**	**%**	**Variável**	**%**
x_{11}	4.04	x_{481}	4.99	x_{446}	4.21
x_{31}	3.98	x_{466}	3.52	x_{483}	4.21
x_{320}	2.53	x_{51}	3.13	x_{179}	2.65
x_{410}	2.46	x_{531}	2.3	x_{389}	2.07
x_{474}	2.27	x_{135}	1.98	x_{2}	1.88
x_{331}	1.83	x_{92}	1.79	x_{365}	1.81
x_{482}	1.71	x_{1}	1.79	x_{51}	1.81
x_{263}	1.64	x_{478}	1.79	x_{466}	1.75
x_{265}	1.64	x_{151}	1.79	x_{464}	1.62
x_{591}	1.58	x_{127}	1.73	x_{477}	1.42

Tabela 5.11 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

NS-GP	
Variável	**%**
x_{151}	3.33
x_{276}	3.33
x_{464}	3.33
x_{293}	3.33
x_{527}	3.33
x_{427}	3.33
x_{423}	3.33
x_{143}	3.33
x_{22}	3.33
x_{536}	3.33

Tabela 5.12 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

5.2.4. Energia de acoplamento molecular

A Figura 5.10 ilustra a mediana do *fitness* de teste dos melhores indivíduos num total de 30 execuções independentes. Pode-se verificar que as versões RMSE-GP, MASE-GP e R2-GP apresentam as melhores medianas ao longo das 200 gerações. Por sua vez, a versão B-GP apresenta uma mediana que oscila sútilmente nas primeiras 100 gerações e que mantêm-se sútilmente constante até ao final, tal como acontece também com o LS-GP. O PCCN-GP apresenta as piores medianas do *fitness* de teste durante as 200 gerações.

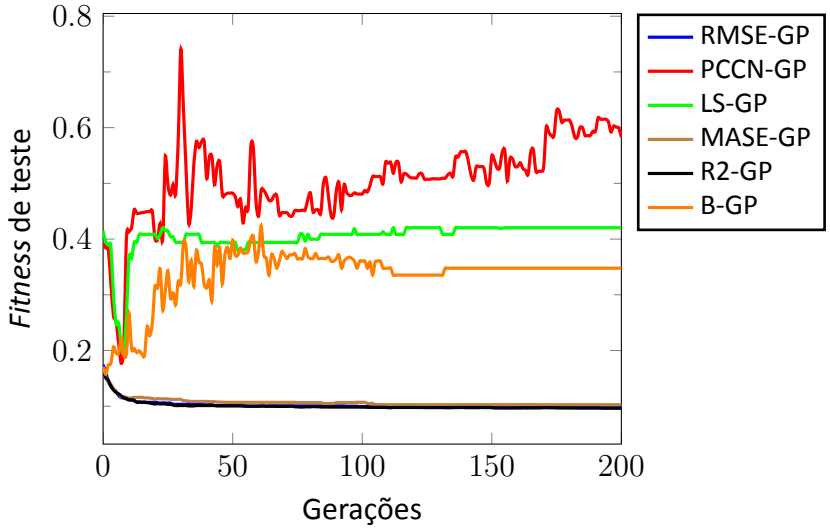

Figura 5.10 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino

Na Figura 5.11, são ilustrados à parte os resultados para o NS-GP devido a enorme diferença de escala (na ordem dos milhares) comparando com as versões anteriores (na ordem das décimas). Nesta figura, os resultados da mediana no *fitness* de teste decresce rapidamente nas primeiras gerações, e depois mantém-se constante até a última geração.

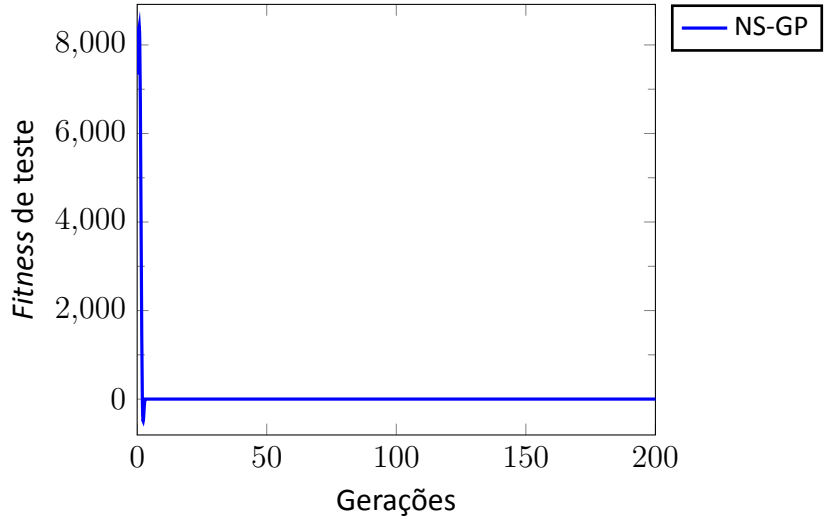

Figura 5.11 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino (NS-GP)

A Tabela 5.13 apresenta o *fitness* de teste, a média do *fitness* de teste e o desvio padrão do *fitness* do melhor indivíduo encontrado ao longo das 200 gerações em 30 execuções independentes no conjunto de treino para todas versões de PG utilizadas. Os

melhores valores do *fitness* de teste, do *fitness* médio e do desvio padrão foram retornados pelo PCCN-GP, R2-GP, MASE-GP e RMSE-GP com valores muito semelhantes entre si. No entanto os melhores valores do *fitness* médio e o desvio padrão do *fitness* foram apresentados por RMSE-GP e R2-GP.

	Fitness	***Fitness* médio**	**Desvio padrão**
RMSE-GP	0.0869	0.0985	0.0095
LS-GP	0.0915	4.0918	16.8311
PCCN-GP	0.0804	5.7765	11.3202
MASE-GP	0.0813	0.1039	0.0125
R2-GP	0.0811	0.0963	0.0055
B-GP	0.1576	0.3501	0.0923
NS-GP	0.2463	0.3870	0.0713

Tabela 5.13 – Melhor *fitness* de teste, média do *fitness* de teste e desvio padrão do *fitness* de teste

A Figura 5.12 reporta a mediana do tamanho dos melhores indivíduos encontrados no conjunto de treino ao longo das 200 nas 30 execuções independentes. O tamanho dos melhores indivíduos encontrados por RMSE-GP, LS-GP, MASE-GP, R2-GP, B-GP e NS-GP são muito semelhantes e mantêm-se relativamente constantes durante as 200 gerações. Diferentemente das outras versões de PG, a mediana do tamanho das melhores soluções encontradas pelo PCCN-GP PCCN-GP aumenta progressivamente da primeira à última geração durante as 30 execuções independentes.

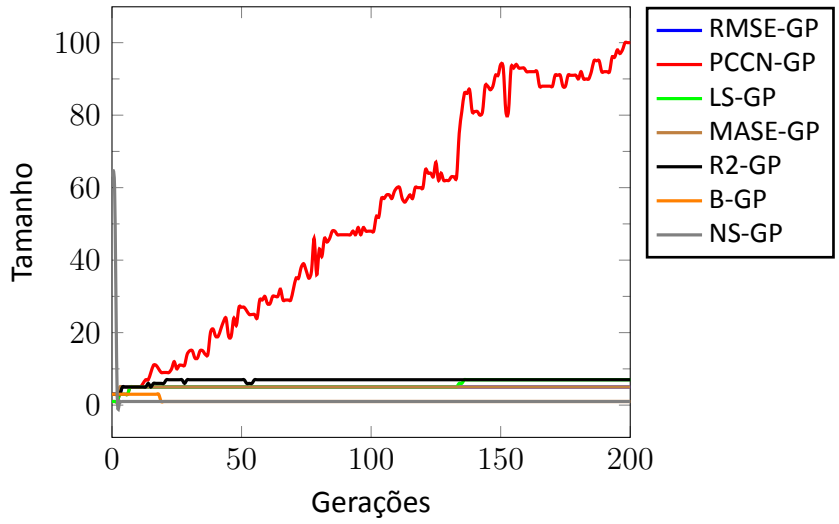

Figura 5.12 – Mediana do tamanho (nº de nós) do melhor indivíduo no conjunto de treino

De seguida, reportamos a percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos nas 30 execuções para as diferentes versões de PG, sobre o conjunto de dados. Pelas tabelas 5.14, 5.15 e 5.16 pode-se notar a presença repetida das variáveis x_{189} (em B-GP, LS-GP, MASE-GP, PCCN-GP, RMSE-GP e R2-GP), x_{227} (em LS-GP, R2-GP, NS-GP), x_{229} (em LS-GP, PCCN-GP, RMSE-GP e R2-GP), x_{225} (em LS-GP, RMSE-GP e R2-GP) e x_{48} (em MASE-GP, PCCN-GP e RMSE-GP), o que revela a importância de tais variáveis para a previsão da energia de acoplamento molecular.

B-GP		LS-GP		MASE-GP	
Variável	**%**	**Variável**	**%**	**Variável**	**%**
x_{169}	4.08	x_{189}	10.22	x_{189}	6.25
x_{88}	4.08	x_{227}	8.15	x_{190}	5
x_{189}	4.08	x_{164}	8.03	x_{219}	2.5
x_{65}	4.08	x_{47}	5.84	x_{199}	2.5
x_{133}	4.08	x_{49}	5.47	x_{48}	2.5
x_{46}	2.04	x_{229}	5.23	x_{218}	2.08
x_{237}	2.04	x_{35}	3.28	x_{187}	2.08
x_{147}	2.04	x_{225}	2.68	x_{158}	2.08
x_{115}	2.04	x_{218}	2.55	x_{172}	2.08
x_{187}	2.04	x_{18}	2.31	x_{159}	2.08

Tabela 5.14 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

PCCN-GP		RMSE-GP		R2-GP	
Variável	**%**	**Variável**	**%**	**Variável**	**%**
x_{189}	9.16	x_{189}	6.45	x_{189}	14.91
x_{48}	7.72	x_{190}	4.84	x_{229}	8.07
x_{229}	4.42	x_{11}	4.52	x_{62}	4.97
x_{52}	3.38	x_{57}	3.55	x_{225}	4.35
x_{47}	3.3	x_{22}	2.9	x_{103}	4.35
x_{8}	2.17	x_{229}	2.9	x_{227}	3.73
x_{101}	2.09	x_{225}	2.58	x_{223}	2.48
x_{51}	2.01	x_{29}	2.58	x_{121}	2.48
x_{165}	1.85	x_{48}	2.26	x_{67}	2.48
x_{240}	1.85	x_{15}	2.26	x_{58}	1.86

Tabela 5.15 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

NS-GP	
Variável	**%**
x_{157}	3.33
x_{95}	3.33
x_{227}	3.33
x_{61}	3.33
x_{20}	3.33
x_{209}	3.33
x_{222}	3.33
x_{74}	3.33
x_{136}	3.33
x_{240}	3.33

Tabela 5.16 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

5.2.5. Fludarabina

A Figura 5.13 ilustra a mediana do *fitness* de teste dos melhores indivíduos num total de 30 execuções independentes. Pode-se verificar que as versões RMSE-GP, MASE-GP, R2-GP e B-GP apresentam as melhores medianas ao longo das 200 gerações. Por sua vez, as versões LS-GP e PCCN-GP apresentam uma mediana que piora ao longo das gerações e apresentam as piores medianas comparando com as outras versões de PG, exceto o NS-GP ilustrado na Figura 5.14.

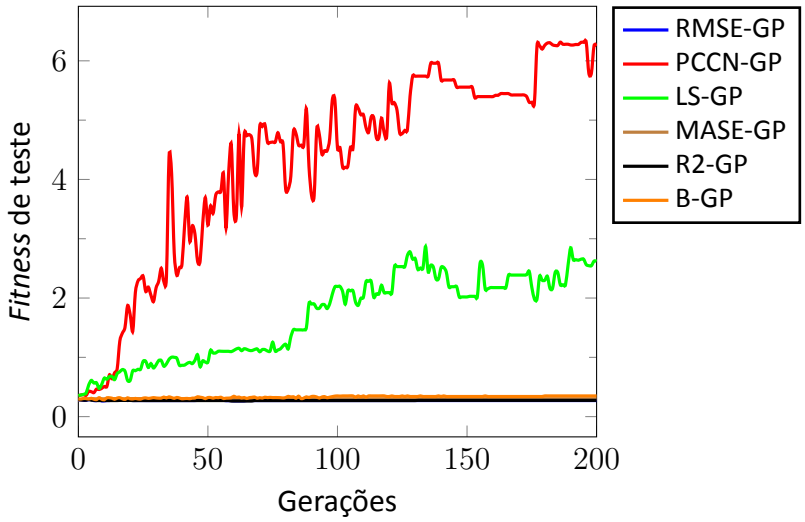

Figura 5.13 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino

Na Figura 5.14 são ilustrados a parte os resultados para o NS-GP, devido a enorme diferença de escala (na ordem dos milhares) comparando com as versões anteriores (na ordem das décimas). Nesta figura, os resultados da mediana no *fitness* de teste decresce rapidamente nas primeiras gerações, e depois mantém-se constante até a última geração.

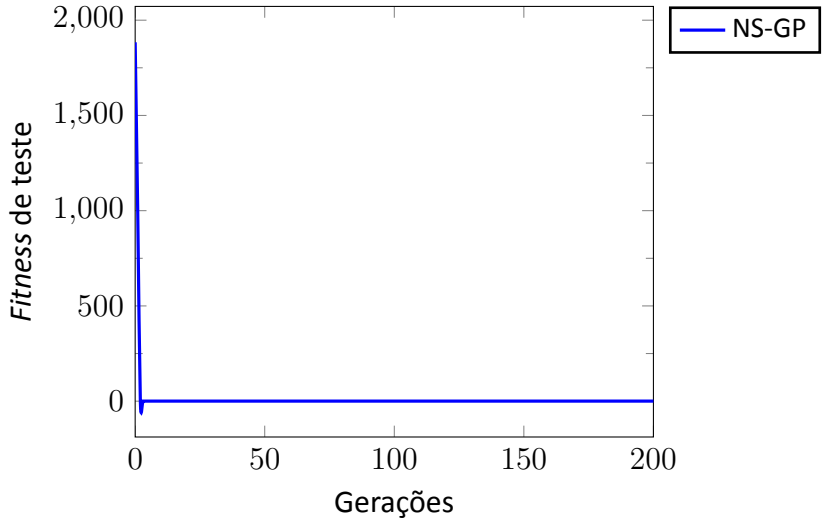

Figura 5.14 – Mediana do *fitness* de teste dos melhores indivíduos encontrados no conjunto de treino (NS-GP)

A Tabela 5.17 apresenta o *fitness* de teste, a média do *fitness* de teste e o desvio padrão do *fitness* do melhor indivíduo encontrado ao longo das 200 gerações em 30 execuções independentes no conjunto de treino para todas versões de PG utilizadas. O melhor valor do *fitness* de teste foi retornado por MASE-GP, sendo tal valor muito aproximado aos retornados por R2-GP, RMSE-GP, B-GP e NS-GP. De forma geral, estas versões de PG apresentaram também os melhores valores do *fitness* médio e do desvio padrão. É de notar que o *fitness* de teste retornado por NS-GP é melhor que os retornados por LS-GP e PCCN-GP.

	Fitness	*Fitness* médio	Desvio padrão
RMSE-GP	0.2054	0.2812	0.0438
LS-GP	0.2353	29.4928	129.1916
PCCN-GP	0.3707	47.2741	174.7481
MASE-GP	0.1926	0.3024	0.0473
R2-GP	0.1958	0.2812	0.0520
B-GP	0.2188	0.3331	0.0595
NS-GP	0.2278	0.3541	0.0620

Tabela 5.17 – Melhor *fitness* de teste, média do *fitness* de teste e desvio padrão do *fitness* de teste

A Figura 5.15 reporta a mediana da quantidade de nós (ou tamanho) dos melhores indivíduos encontrados no conjunto de treino ao longo das 200 nas 30 execuções independentes. O tamanho dos melhores indivíduos encontrados por RMSE-GP, MASE-GP, R2-GP, B-GP e NS-GP são muito semelhantes e mantêm-se relativamente constantes durante as 200 gerações. Diferentemente das outras versões de PG, a mediana do tamanho das melhores soluções encontradas pelo PCCN-GP e LS-GP aumenta progressivamente ao longo das 200 gerações, em 30 execuções independentes.

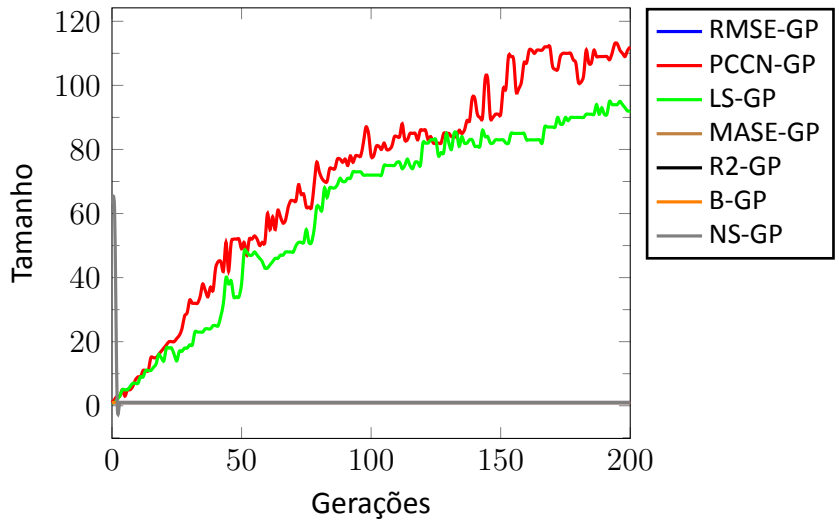

Figura 5.15 – Mediana do tamanho (nº de nós) do melhor indivíduo no conjunto de treino

De seguida, reportamos a percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos nas 30 execuções para as diferentes versões de PG, sobre o conjunto de dados da Fludarabina. Pelas tabelas 5.18, 5.19 e 5.20 pode-se notar a presença repetida das variáveis x_{520}, x_{629} e x_{283} que são utilizadas por R2-GP e RMSE-GP,

que apresentam os melhores resultados de *fitness* sobre o conjunto de teste.

B-GP		LS-GP		MASE-GP	
Variável	**%**	**Variável**	**%**	**Variável**	**%**
x_{1096}	6.67	x_{984}	2.59	x_{1371}	8.82
x_{1277}	6.67	x_{742}	1.41	x_{16}	8.82
x_{342}	3.33	x_{782}	1.33	x_{590}	5.88
x_{155}	3.33	x_{16}	1.18	x_{1106}	5.88
x_{520}	3.33	x_{992}	1.11	x_{444}	4.41
x_{1320}	3.33	x_{1242}	1.11	x_{791}	4.41
x_{911}	3.33	x_{752}	1.11	x_{1333}	4.41
x_{1013}	3.33	x_{997}	1.04	x_{970}	4.41
x_{352}	3.33	x_{887}	1.04	x_{333}	2.94
x_{865}	3.33	x_{69}	1.04	x_{520}	2.94

Tabela 5.18 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

PCCN-GP		RMSE-GP		R2-GP	
Variável	**%**	**Variável**	**%**	**Variável**	**%**
x_{145}	1.88	x_{629}	9.3	x_{520}	11.11
x_{16}	1.62	x_{283}	9.3	x_{1236}	7.78
x_{812}	1.49	x_{883}	4.65	x_{755}	6.67
x_{742}	1.43	x_{318}	4.65	x_{504}	5.56
x_{752}	1.3	x_{682}	4.65	x_{1286}	5.56
x_{984}	1.23	x_{502}	2.33	x_{629}	4.44
x_{771}	1.17	x_{1242}	2.33	x_{420}	4.44
x_{3}	1.1	x_{1203}	2.33	x_{771}	4.44
x_{744}	1.04	x_{405}	2.33	x_{283}	3.33
x_{477}	0.91	x_{141}	2.33	x_{526}	3.33

Tabela 5.19 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

NS-GP	
Variável	%
x_{430}	3.33
x_{653}	3.33
x_{645}	3.33
x_{56}	3.33
x_{683}	3.33
x_{499}	3.33
x_{986}	3.33
x_{1323}	3.33
x_{910}	3.33
x_{398}	3.33

Tabela 5.20 – Percentagem de ocorrência das 10 variáveis mais utilizadas pelos melhores indivíduos

5.3. DISCUSSÃO

Investigou-se a performance da PG sobre o problema de previsão de parâmetros farmacocinéticos utilizados durante o processo de descoberta e desenvolvimento de fármacos. Este processo é muito importante uma vez que permite poupar o tempo e os custos associados aos ensaios clínicos, e evitar os efeitos adversos de certos compostos no organismo humano. Para tal, desenvolveram-se diversos modelos quantitativos QSAR utilizando a PG sobre dados de descritores moleculares de alguns parâmetros farmacocinéticos, tal como descritos no capítulo 4.

Foram utilizadas 7 versões de PG, com diferentes funções de *fitness*, para construir modelos de previsão a partir de um conjunto de dados de treino. As diferentes versões de PG foram executadas 30 vezes sobre cada um dos 5 problemas de previsão. Em cada execução, os conjuntos de dados foram repartidos aleatoriamente em 70% para o conjunto de treino, e o restante 30% para o conjunto de teste. De seguida, foi estabelecida uma comparação entre a PG padrão (RMSE-GP) e as diferentes variações, de acordo as seguintes características:

- Performance e capacidade de generalização das soluções (indivíduos) sobre o conjunto de teste
- Complexidade das soluções: tamanho ou quantidade de nós

- Usabilidade das soluções: seleção automática das variáveis mais relevantes

No geral, foi possivel notar que a RMSE-GP, MASE-GP e R2-GP, apresentaram os melhores resultados no que confere a mediana do *fitness* do teste dos melhores indivíduos, sobre os diferentes conjunto de dados. Pelo contrário, a PCCN-GP, a LS-GP e NS-GP apresentaram sempre as piores medianas. Quanto a B-GP, o seu comportamento variou com relação ao conjunto de dados em questão, por vezes apresentando valores melhores que os de PCCN-GP e LS-GP, mas nunca melhores do que RMSE-GP, MASE-GP e R2-GP. O NS-GP apresentou medianas muito altas e geralmente fora da escala das outras versões. Este comportamento é justificado pela natureza do algoritmo de pesquisa de novidade apresentado na secção 4.3.7, onde o indivíduo que apresenta mais novidade é aquele com a maior distância média em relação aos indivíduos da geração anterior.

Quanto ao *fitness* de teste, o *fitness* médio de teste e o desvio padrão do *fitness* de teste do melhor indivíduo encontrado nas 30 execuções independentes, de forma geral a RMSE-GP apresentou os melhores resultados nos diferentes conjuntos de dados. Estes resultados são relativamente semelhantes aos encontrados por R2-GP e MASE-GP. Os melhores indvíviduos de PCCN-GP e LS-GP retornaram os piores resultados, exceto no conjunto de dados da *Energia de acoplamento molecular* onde a PCCN-GP retornou o menor *fitness* no conjunto de teste. Curiosamente, a NS-GP apresentou uma capacidade de generalização aceitável nos diferentes conjuntos de teste, mas com valores muito altos do *fitness* médio de teste e o desvio padrão do *fitness* de teste, comparado com as outras versões.

Para avaliar a complexidade das soluções encontradas pelas diferentes versões de PG, foi calculada a mediana do tamanho dos melhores indivíduos retornados ao longo de 30 execuções independentes. O tamanho de um indivíduo consiste na quantidade de nós da árvore de sintaxe que representa este indivíduo. Em todos os conjuntos de dados a NS-GP apresentou um comportamento muito diferente das outras versões de PG, uma vez que a mediana do tamanho dos melhores indivíduos por si encontrados manteve-se aproxidamanente igual a zero ao longo das 200 gerações. Outro comportamento interessante foi revelado por B-GP, em que a mediana oscilava ligeiramente nas gerações iniciais e permanecia constante até a última geração, apresentado sempre uma mediana menor do que as outras versões de PG. A elaboração de um estudo mais profundo para

determinar as razões deste comportamento estranho do NS-GP e do B-GP, não fez parte do âmbito do presente trabalho mas, é um objetivo para trabalhos futuros. No geral, a RMSE-GP, a MASE-GP e R2-GP apresentaram valores muito semelhantes e melhores que os de PCCN-GP e LS-GP. O PCCN-GP e o LS-GP, tal como descritos acima, retornaram os piores *fitness* de teste, o que revela que estas versões estão sujeitas ao fénomemo do *bloat* ou seja, o crescimento do tamanho das soluções não implica a melhoria do *fitness*.

As melhores soluções encontradas pelas diferentes versões de PG, efetuaram uma seleção automática das variáveis mais importantes. No geral, as melhores soluções não utilizaram mais de 10% do total de descritores moleculares. Foi possível notar também que, ao longo das 30 execuções independentes, algumas variáveis foram utilizadas regularmente pelas melhores soluções nas diferentes versões de PG. Isto indica que a PG é capaz de identificar os descritores moleculares mais importantes para a construção de um modelo de previsão. No entanto, é necessário combinar o conhecimento especializado na área com os resultados obtidos pela PG para identificar o quão relevantes são estes descritores moleculares para o processo de previsão de parâmetros farmacocinéticos.

Embora que no geral os resultados apresentados pelo NS-GP são piores com relação aos do RMSE-GP, MASE-GP e R2-GP ainda assim são promissores e abrem caminho para futuras investigações sobre a aplicação desta técnica em problemas de regressão simbólica. O método de escalonamento linear implementado pela configuração LS-GP pode apenas apresentar resultados significativamente melhores que a PG-padrão (RMSE-GP) nos dados do conjunto de treino mas não superiores nos dados do conjunto de teste, tal como descrito em (**?**). Esta afirmação foi constatada no presente trabalho uma vez que, no geral os resultados encontrados pelo LS-GP foram muito piores que os obtidos pelo RMSE-GP no conjunto de teste.

6

CONSIDERAÇÕES FINAIS

Conteúdo

6.1. CONCLUSÃO

Os resultados obtidos demonstram a capacidade da PG em resolver problemas de previsão de parâmetros farmacocinéticos e construção de modelos QSAR através da definição apropriada de uma função de *fitness*. Além disso, os resultados encontrados são muito semelhantes, e na maior parte das vezes melhores, do que os encontrados na literatura, por exemplo em (**??**) e (**?**), onde foi demostrado que a PG supera outros métodos de previsão amplamente aplicados em ML.

Um estudo completo da performance da PG no problema de previsão de parâmetros farmacocinéticos é uma atividade de extrema complexidade e que carece da aplicação de conhecimento especializado. No entanto, foi feita neste trabalho uma tentativa de perceber até que ponto modelos QSAR confiáveis podem ser construídos utilizando a PG.

Apesar da PG ser uma técnica de ML nova comparando com as técnicas tradicionais (e.g: ANN, regressão linear, etc.), as suas vantagens abrem caminhos para o desenvolvimento de soluções robustas em problemas reais. Tal como demostrado em trabalhos anteriores (e.g: (**?**), (**?**) e (**?**)), a PG, com relação a outros métodos de previsão, tem a vantagem de efetuar uma seleção automática dos atributos (ou características) mais relevantes no conjunto de dados. Esta vantagem, pode ser combinada com o conhecimento especializado na área para a construção de funções de *fitness* mais apropriadas ao desenvolvimento de modelos QSAR, uma vez que foi demostrado no presente trabalho que a definição da função de *fitness* influencia na capacidade de generalização, a performance e a complexidade das soluções encontradas pela PG.

A investigação na área de PG tem vindo a evoluir substancialmente e novos métodos para aperfeiçoar a regressão simbólica têm sido desenvolvidos. Um exemplo recente é apresentado em (**?**) onde é introduzido o conceito de operadores semânticos geométricos que operam sobre a semântica/comportamento dos indivíduos em vez da sua sintaxe. Em (**?**) e (**?**), uma implementação desta técnica é aplicada para a previsão de alguns parâmetros farmacocinéticos, com resultados melhores que os obtidos no presente trabalho.

6.2. TRABALHOS FUTUROS

A pesquisa por novidade apresentou alguns resultados interessantes nos problemas de regressão simbólica discutidos neste trabalho. Sendo assim, pretende-se como trabalho futuro aprofundar a aplicação desta técnica PG à problemas de regressão simbólica por desenvolver um descritor comportamental que combine os benefícios da pesquisa por novidade e a pesquisa por objetivo (Critérios Mínimos de Pesquisa de Novidade - *Minimal Criteria Novelty Search* (MCNS) (**??**)) e avaliar o sua performance nos conjuntos de dados utilizados neste trabalho e noutros problemas de referência em PG (**?**).