Refatoração do ambiente *MaigesPack*

Trabalho Supervisionado de Formatura

Emerson Takeshi Hassegawa

Fábio Yoshio Sato

Ricardo Issao Shimanuki

Orientador: Eduardo Jordão Neves

Instituto de Matemática e Estatística – Universidade de São Paulo

1. Introdução

O homem com a sua curiosidade vem se aprofundando nas pesquisas em diversas áreas desde tempos remotos. Hoje, estamos em um nível em que apenas uma área de conhecimento não é o suficiente para nos prover ferramentas para explicar os fenômenos que nos despertam curiosidade. A Bioinformática é uma nova área de pesquisa que une os conhecimentos dos estudos da biologia, como estudos genômicos e a medicina, com conhecimentos da área de exatas, como computação, estatística e física. Com a ajuda da Bioinformática, podemos estudar a influência dos genes na produção de proteínas em células, entre tantas outras possibilidades de estudos.

2. Objetivos

2.1. Proposta Anterior

Fazer uma refatoração do sistema de análise de dados de microarray R, redesenhando os processos de análise para que as dependências mostrem a realidade, melhorar a flexibilidade do sistema para que os usuários consigam ter maior domínio sobre o processo analítico realizado sobre os dados, aumentar a visibilidade para que o usuário consiga saber cada etapa do processo e entender a sua funcionalidade. Como refatoração, visávamos não mudar nenhuma funcionalidade e não ter que escrever muito código, apenas quebrar métodos em métodos menores visando às melhoras já citadas.

2.2. Proposta Atual

Não se difere muito nos objetivos e sim no método. Depois de estudar o código do sistema, percebemos que ele já estava bem modularizado e não havia muitos espaços para mudança sem modificar o código dos métodos. Como para os usuários do sistema objetivos do projeto seriam de grande utilidade, procuramos outra maneira de realizá-los. Começaremos pela função relNetworkM, modificando alguns parâmetros e possivelmente criando algumas funções para adequação dos objetos de entrada e saída dos métodos usados na construção da rede de relevância, para dar maior flexibilidade para o usuário.

3. Microarray

3.1. Expressão gênica

Todo ser humano tem o ácido desoxirribonucléico (ADN, em português, ou mais, por convenção; DNA, do inglês, deoxyribose nucleic acid) em suas células. Ele carrega a informação genética, que é o que determina as características de cada pessoa, e conseqüentemente coordena o seu desenvolvimento e funcionamento, pois é usando essa informação que se produzem as proteínas, que são as biomoléculas funcionais indispensáveis para a manutenção do ciclo de vida celular.

O processo de produção de uma proteína a partir de um gene (DNA) é o que se chama de expressão gênica, e é um dos assuntos mais estudados na Biologia Molecular. Simplificando, a partir de um gene (seqüência de DNA) um mRNA (RNA Mensageiro) é formado através do processo de transcrição, esse mRNA tem como função levar a informação sobre a cadeia protéica para o RNAt (RNA Transportador) que opera formação da proteína.

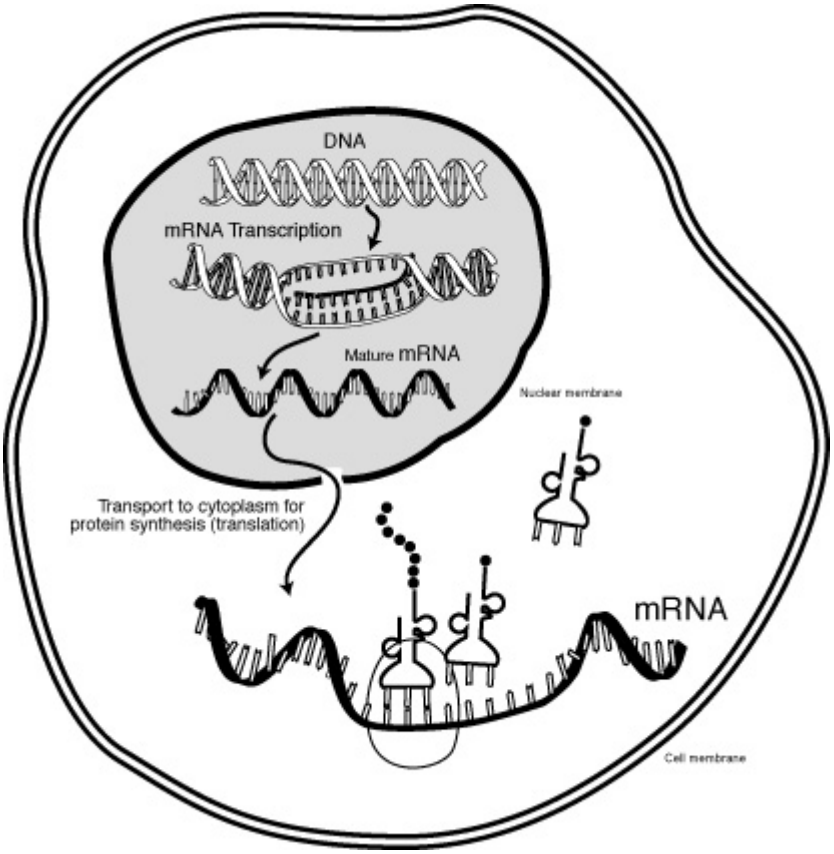


Figura 1: Transcrição, processo onde um mRNA leva a informação de um gene para ser expresso como uma proteína [8].

A Regulação gênica é a habilidade da célula de controlar quando e em que quantidade uma proteína irá ser produzida, e é a base da diferenciação celular, da versatilidade e adaptabilidade de qualquer organismo. Logo, entender em que momento um gene é expresso e o que tal expressão irá desencadear é um dos grandes desafios da biologia, medicina e bioinformática atualmente.

3.2. Análise Microarray

Em 2003, o “Projeto Genoma” foi considerado concluído ele seqüenciou 94% do DNA do genoma humano. Este foi um grande passo na evolução dos estudos sobre a genética [4], pois evidenciou o potencial do que a Biologia aliada à alta tecnologia pode fazer pela humanidade. Um genoma completo permite que se tenha uma visão global dos genes, porém não só evidencia a seqüência dos genes,

mas como mostra suas colocalizações e relações com fatores regulatórios. (quem escreveu??? Ficou em vermelho e sem formatar)

Com tantas descobertas e estudos genômicos, houve uma necessidade de métodos para analisar essas informações. Assim, a Bioinformática como área de pesquisa ganhou mais espaço e força, uma vez que tem como meta a utilização, integração e interpretação dos dados de genômica.

Nesse cenário, surgiu a plataforma de Microarray em meados da década de 90 (DeRisi et al., 1996; Schena et al., 1995; Shalon et al., 1996) capaz de analisar a expressão gênica de milhares de genes simultaneamente (como uma "foto" da expressão de milhares de genes de um determinado instante), com a vantagem de ainda permitir a comparação entre duas populações de mRNA (RNA mensageiro- ácido ribonucléico mensageiro), com o objetivo de entender os mecanismos de transformação molecular dos tecidos analisados. Catalogando em banco de dados as diferenças entre a expressão gênica das amostras estudadas, pode-se comparar a expressão dos genes em diversas condições biológicas, como por exemplo, em um tecido sadio versus em um patológico.

3.2.1. Processo de Análise de Microarray

Existem diversos métodos diferentes para cada etapa da análise de Microarray, porém no nosso estudo não era interessante se prender a detalhes de cada metodologia. Por isso, descreveremos agora o processo de análise de uma forma genérica, apenas citando algumas particularidades.

3.2.1.1. Obtenção do Microarray

A análise consiste em colocar amostras específicas de DNA que representam genes, que podem ser cDNAs ou oligonucleotídeos, em uma lâmina de

vidro ou uma membrana de nylon, de dimensões mínimas. Quando se utiliza uma lâmina de vidro é necessário para garantir a precisão, o auxilio de uma estrutura robótica que usando agulhas de impressão depositam o material nas posições específicas da lâmina, conhecida como spots. As fitas de DNA são colocadas em pequenos poros da lâmina com a ajuda de adesivos de superfície como aminosilanos ou poli-l-lisina para maior adesão. mRNA são retirados dos tecidos a serem estudados, transformados em cDNA e marcados com fluocromos, em um experimento típico de duas cores, são usados o cy5, corante vermelho, e o cy3, corante verde. A lâmina é depositada em uma solução de hibridização, neste processo caso ocorre a complementaridade entre o cDNA fixado na lâmina e a amostra, assim temos a proporção da expressão gênica de cada gene referenciado em um determinado spot.

3.2.1.2. Escaneamento da lâmina

A lâmina é escaneada usando dois comprimentos de ondas para diferenciar os fluocromos usados na marcação. Essa imagem é trabalhada em um programa que processa imagens para ficar apta a ser estatisticamente analisada. Diversos processos são usados.

- O background (fundo da imagem) tem sua intensidade diminuída para não se confundir com os marcadores usando algumas transformações padrão. (Ishi et al., 2000; Schuchhardt et al., 2000; Kepler et al., 2002)
- Elimina-se spots (cada ponto da imagem que representa a hibridização do RNAm com a fita de DNA) que tenham problemas de qualidade
- É feita a identificação e o alinhamento dos spots.
- A expressão de cada spot é quantificada. Essa quantificação pode ser feita de diversas maneiras, com seu valor podendo variar muito dependendo da escolha. O método a ser usado depende das suposições feitas sobre o modelo.
- Muitas vezes um mesmo gene está referenciado em mais de um spot para garantir que o valor estimado de expressão seja confiável.

- Invertem-se os canais atribuídos às amostras dos tecidos para compensar qualquer interferência indesejada.

Como resultado final, se terá uma matriz de números reais que refletem o valor da expressão dos genes, cujas linhas são os genes e as colunas são as amostras.

3.2.2. Métodos estatísticos aplicados nos dados de Microarray

Com os dados da expressão gênica dos tecidos a serem estudados em mãos, existem diversas técnicas a serem aplicadas. Citaremos algumas explicando brevemente sua funcionalidade.

3.2.2.1. Normalização

Transformação usada para ajustar os valores medidos tirando interferências não relacionadas propriamente com as amostras como, quantidade de mRNA inicial, diferenças causadas pelos fluorescentes usados e possíveis erros de medição da expressão gênica.

O método de normalização usado é escolhido levando em consideração o modelo estatístico utilizado. Indiferentemente do método escolhido deve-se escolher um conjunto de genes que servirá de referência na normalização. Existem três principais critérios para essa escolha.

3.2.2.1.1. Utilizando todos os genes da lâmina

As análises geralmente são muito específicas. Logo, não é esperada diferença de expressão entre todos os genes apenas entre um grupo pequeno.

Tendo essa expressão constante na maioria dos genes podemos usá-los como indicadores da intensidade relativa dos dois canais.

3.2.2.1.2. Genes expressados constantemente

Identifica-se um grupo de genes, chamados housekeeping, que pode ser considerado com expressão constante indiferente das condições. Existe uma grande de dificuldade nessa identificação, mas pode ser feita para condições experimentais particulares.

3.2.2.1.3. Genes expressados controles

Controle Spiked-in consiste em adicionar genes previamente escolhidos na lâmina, se eles forem colocados em quantidades iguais eles não apresentaram diferença na intensidade e poderão ser usados na normalização.

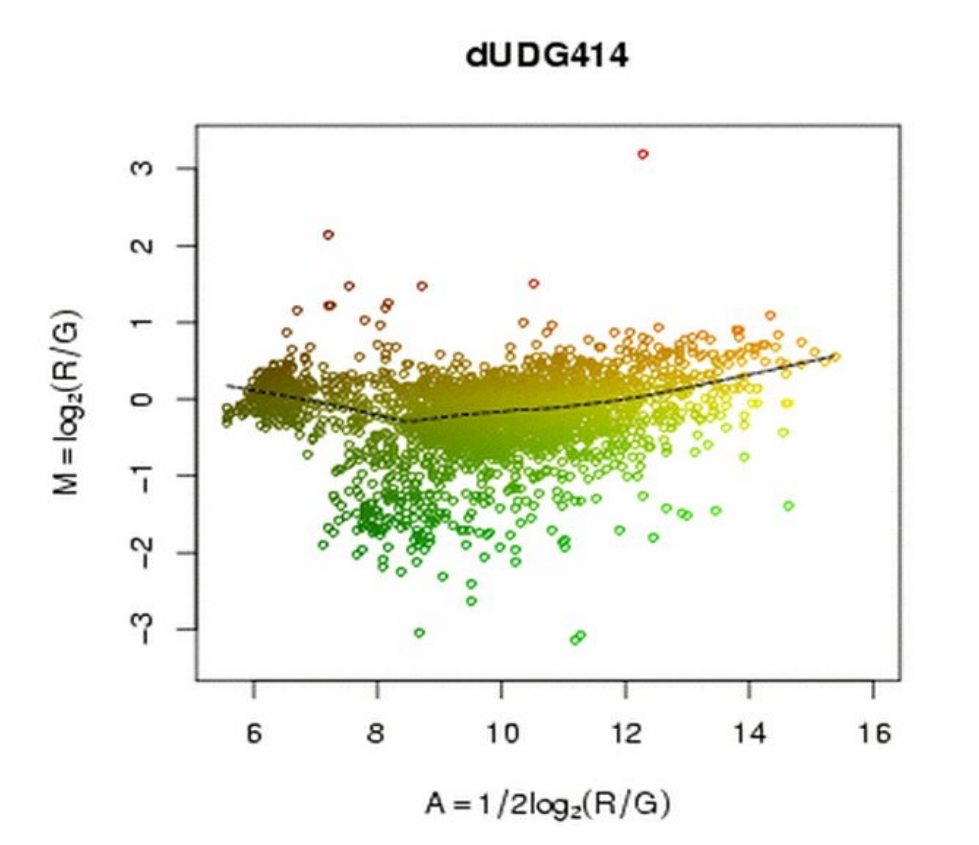


Figura 2: Dado sem Normalização.

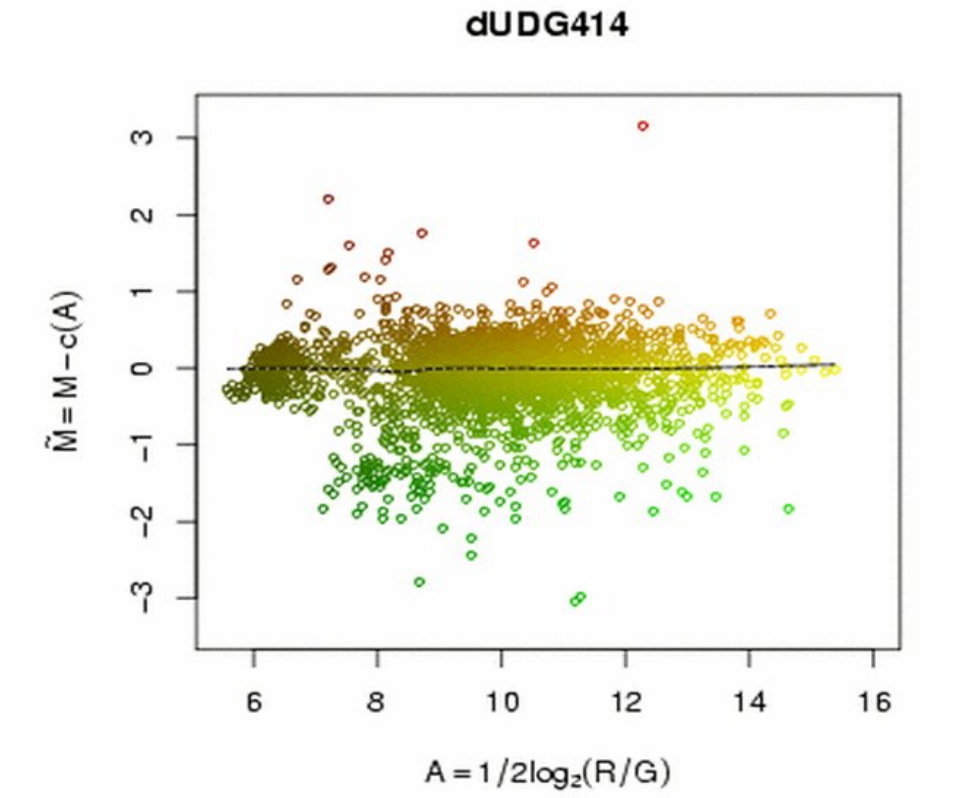


Figura 3: Dado com normalização global (lowess).

3.2.3. Agrupamento

Técnica usada para dividir os dados em grupos usando critérios tanto quantitativos quanto qualitativos. Primeiro define-se quais serão os objetos (pontos) a serem considerados, podendo ser os genes ou amostras de pacientes. Após o objeto definido resta escolher a função que será usada para calcular a distância entre os pontos, quais pontos serão usados para medir a distância entre os grupos e quantos grupos serão formados.

Um dos métodos de agrupamento muito usado é o Agrupamento Hierárquico, esse método não pressupõe que cada objeto pertence somente a um grupo. Os objetos são ordenados de forma hierárquica onde a proximidade entre todos os objetos é mostrada. No primeiro nível agrupa-se os objetos com similaridade máxima e depois a cada nível junta-se os grupos segundo o critério de distância escolhido.

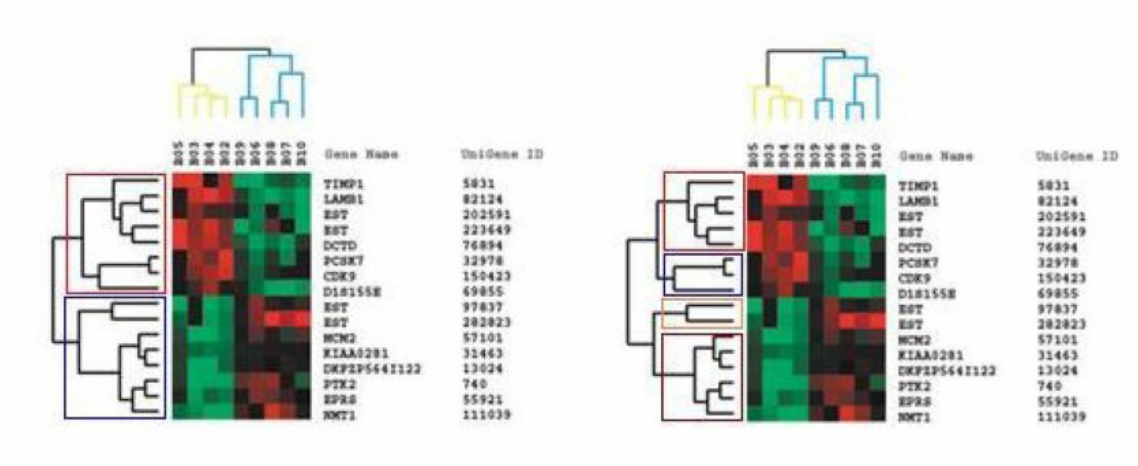


Figura 4: Duas árvores hierárquicas completas. A árvore pode ser cortada em diversos níveis para se obter diferentes números de grupos.A imagem da esquerda mostra dois grupos enquanto a da direta está divida em 4 grupos.

3.2.4. Classificação

É de interesse do estudo determinar genes ou grupos de genes que possam distinguir bem amostras de tipo diferentes. É necessário um grande número de amostras para se definir um critério de classificação eficiente.

4. MaigesPack

4.1. Introdução

No capítulo anterior, foi apresentado o processo de análise de microarray, o qual se mostrou complexo e constituído de diversas etapas experimentais de natureza diferente.

Os dados de expressão gênica obtidos desta análise podem sofrer modelagem estatística com o intuito de extrair informações genéticas mais precisas e detalhadas. Para isso, é necessário um ambiente computacional que integre métodos matemáticos e estatísticos de análise já disponíveis na literatura e que mantenha confiabilidade, reprodutibilidade e robustez às análises efetuadas.

É preciso também que este ambiente seja simples, modularizado e bem documentado, pois cada etapa precisa ser bem compreendida e controlada pelo usuário que irá utilizar este ambiente.

Neste contexto, surgiu o pacote denominado MaigesPack.

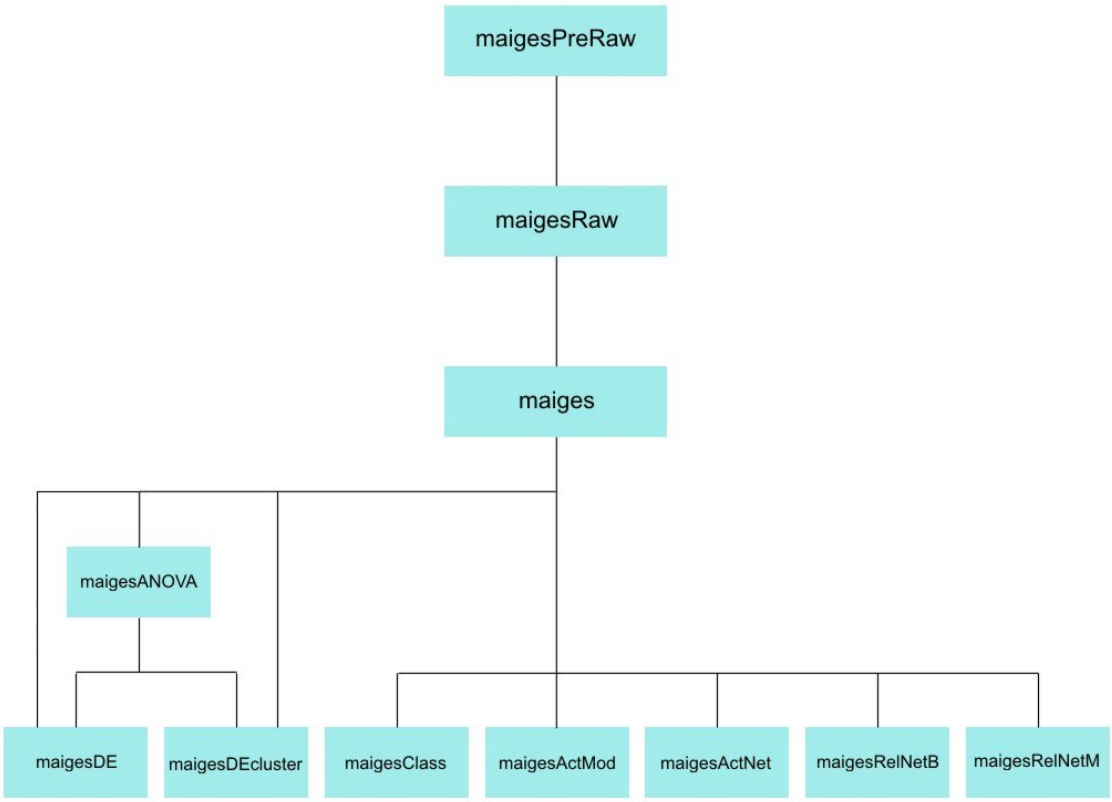
4.2. Definição

O MaigesPack é um pacote de arquivos com métodos matemáticos e estatísticos, escritos em linguagem de programação estatística R e em C, que desenvolve um ambiente computacional para análise de dados de microarray, integra estes com diversos algoritmos já desenvolvidos pelo projeto Bioconductor e implementa outros métodos de análise. Ele foi criado por Gustavo Henrique Esteves e está disponível no site do projeto Bioconductor ().

4.3. Estrutura

Neste novo ambiente, foram definidas classes de objetos para armazenar desde conjuntos de dados de microarray bruto até os resultados das análises específicas.

A figura abaixo contém um grafo representando a estrutura e as classes, que serão definidas na próxima seção.



Todos os dados numéricos e informações de interesse para o conjunto a ser analisado encontram-se na primeira etapa do processo, na classe maigesPreRaw. Aqui, os dados sofrem análises exploratórias em busca de possíveis problemas com a carga dos dados e mesmo problemas experimentais. Quando verificado que algum spot não cumpre critérios de qualidade exigidos atualiza-se os campos referentes aos spots ruins.

Através de um método específico denominado create.maigesRaw, que recebe como parâmetro um objeto desta classe, variáveis de caracteres que especificam os campos numéricos a serem utilizados e dois argumentos de texto contendo a especificação dos rótulos de genes que serão usados para mapear os

grupos e a rede gênica, o objeto da classe maigesPreRaw é transformado em um objeto da classe seguinte: maigesRaw.

Nesta classe, são aplicadas funções de normalização que utilizam os pacotes limma, marray e OLIN, já implementados no Bioconductor, seja para criar novas funções como o Mnorm.scaleLimma e o Mnorm.scaleMarray, ou apenas simplesmente para utilizar algum método já pronto como Mnorm.OLIN.

Após a aplicação das normalizações, são obtidos os dados da classe maiges, onde os objetos sofrem análises específicas para: genes diferencialmente expressos (maigesDE e maigesDEcluster), classificação (maigesClass, maigesActMod e maigesActNet) e redes de relevância (maigesRelNetB e maigesRelNetM).

A classe maigesANOVA é uma extensão da classe maiges.

4.4. Classes de objetos

Nesta seção, são apresentadas as características de cada classe do MaigesPack.

4.4.1. MaigesPreRaw

Como dito anteriormente, nesta classe estão todos os dados e informações para o conjunto a ser analisado. Ela possui listas que recebem campos numéricos a partir da análise de imagens, vetores de texto que especificam os grupos gênicos a serem avaliados, grafos que definem as redes de regulação de interesse, configuração das lâminas e rótulos para os genes e amostras de estudo.

Nesta primeira etapa, caso existam problemas, eles são informados em um campo lógico especificando spots ruins e campo de texto para informações adicionais.

4.4.2. MaigesRaw

Esta classe possui matrizes que: armazenam os valores de intensidade de sinal, indexam os spots a serem utilizados para a normalização e indexam genes pertencentes a diferentes grupos gênicos.

4.4.3. Maiges

Os objetos contidos nesta classe são gerados a partir da normalização aplicada sobre os objetos da classe maigesRaw. Aqui, são armazenados valores de razão de intensidade média dos spots em escala logarítmica e desvio padrão e intervalos de confiança para os valores normalizados.

- **MaigesANOVA**: é uma extensão da classe maiges que armazena matrizes de planejamento e de contrastes utilizadas para o ajuste de um modelo de análise de variância e para estimação de parâmetros.
- **MaigesDE**: contém resultados das análises de busca de genes diferencialmente expressos (genes DE)
- **MaigesDEcluster**: trata-se de uma extensão da classe anterior porém com matriz de valores de razão
- **MaigesClass**: possui resultados de análises de discriminação ou de classificação
- **MaigesActMod**: armazena as saídas das análises de classificação funcional de grupos gênicos
- **MaigesActNet**: contém resultados de análises de classificação funcional de redes de regulação gênica
- **MaigesRelNetB**: possui resultados de análises de redes de relevância
- **MaigesRelNetM**: armazena resultados de análises de redes de relevância usando o modelo aprimorado discutido

5. Refatoração

5.1. Definição

Refatoração [10] é o processo de alterar a estrutura interna de um código sem que o comportamento externo seja mudado. A idéia é que as alterações internas melhorem a estrutura do código através de mudanças pequenas, mas constantes, que transformem para melhor a estrutura interna do código. Porque as mudanças tendem a ser pequenas, as chances de introduzir erros são minimizadas. Uma conseqüência é a melhora do design, e a partir daí da legibilidade, de porções de código que sem refatoração se deteriorariam. Essa definição pode ser associada a qualquer linguagem de programação tais como Java, C, C++, Ruby, R, entre outras.

Para que seja possível compreender a refatoração em uma aplicação R é necessário conhecer antes algumas ferramentas como o ambiente de análise estatística R e o RUnit. Ambas as ferramentas serão apresentados em seguida.

5.2. Ambiente R

R[11] é um ambiente de análise estatística que, por ser de código aberto [14], confere um grande poder de análise, contribuição de código e aprendizado. Ele é constituído de um pacote base (necessário) e de vários outros pacotes (vários contribuintes, por ser código aberto) que podem ser obtidos em dois repositórios: CRAN e Bioconductor. O projeto Bioconductor é uma iniciativa que visa implementar funcionalidades para análise de dados de genômica, logo, a versão presente do MaigesPack é parte integrante do mesmo.

5.3. RUnit

RUnit [14] é um pacote que modela um framework de testes unitários em programas codificados em R e disponibiliza funcionalidades para conseguir rastrear resultados após a execução de casos de testes e gerar um relatório resumido. Esse pacote também disponibiliza ferramentas para inspecionar código e dessa forma, também para análise de cobertura de casos de testes.

5.4. Refatoração em uma aplicação em R

A refatoração numa aplicação em R agrupa um conjunto de técnicas de melhorias pequenas na estrutura interna do código. Esse conjunto de técnicas faz parte de um conjunto maior que é o conjunto das técnicas de melhorias para a linguagem R e para as demais linguagens de programação existentes. Não existe um número exato de técnicas, pois algumas são aplicáveis a algumas linguagens e não a outras. Alguns procedimentos foram adotados para garantir que após a refatoração da aplicação, o comportamento do mesmo não se altere. Os procedimentos são os seguintes [12]:

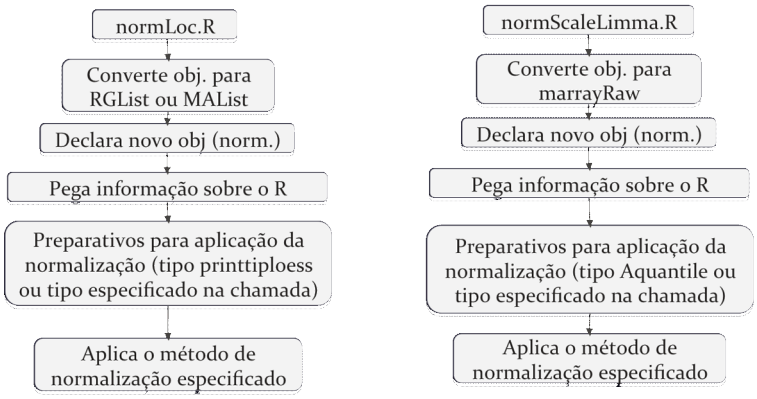
5.4.1. Identificar refatorações

A decisão de refatorar um trecho de código, ou determinar que aquele ponto seja passível de sofrer refatoração, depende apenas do julgamento do indivíduo que está analisando o código. Algumas dessas refatorações podem ser identificadas facilmente, sem que haja dependência da linguagem que está sendo usado, ou seja, aplicável a códigos em R. Um exemplo seria a identificação de duplicação de código.

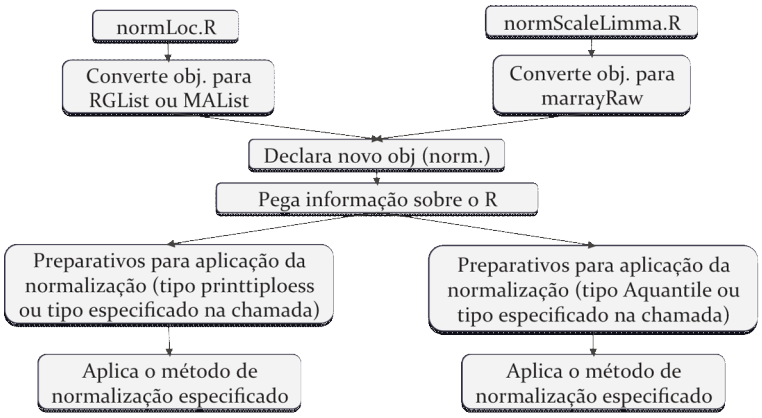
Exemplo:

Esse exemplo foi encontrado com a análise do processo de normalização dos dados para o maigesPack.

5.4.1.1. Antes da refatoração



5.4.1.2. Depois da refatoração



5.4.2. Determinar quais refatorações deve ser aplicada aos trechos identificados

Pode ser, conceitualmente, realizado a partir do procedimento anterior, mas, na prática, isso se torna difícil de distinguir. No exemplo anterior, a refatoração deve visar a remoção do código duplicado.

5.4.3. Aplicar a refatoração

Após identificar os trechos que podem ser refatorados e identificados as técnicas a serem aplicadas é necessária aplicar as técnicas aos trechos.

5.4.4. Garantir que o comportamento será preservado após a refatoração

A principal ferramenta para garantir que a refatoração irá preservar o comportamento do programa é manter um conjunto rico de testes unitários durante o processo de desenvolvimento. Alguns instrumentos que podem auxiliar são os subdiretórios de testes do pacote R e a integração com o RUnit.

5.4.5. Avaliar os efeitos da refatoração com relação a características de qualidade

Atualmente não existem ferramentas para realizar métricas em aplicações codificadas em R que sejam específicas para essa linguagem. Muitos dos

programas para realizar métricas, no entanto, devem ser adaptados e implementados.

5.4.6. Manter consistência com os outros artefatos do programas

As documentações devem ser atualizadas considerando as modificações realizadas. A documentação do R pode ser realizada usando o Roxygen [13]. O Roxygen é um sistema que permite unificar a manipulação do código e da documentação, mas essa documentação pode ser feita manualmente sem auxílio de uma ferramenta específica.

6. Avaliações Subjetivas

6.1. Emerson Takeshi Hassegawa

6.1.1. Desafios e Frustrações

O processo de estudo e pesquisa de bioinformática foi um dos desafios que tive que enfrentar, pois é uma área nova, especialmente pra mim. Os conhecimentos que eu tinha dessa área, que foram obtidas das aulas de Biologia no ensino médio, não eram suficientes compreender os processos de análise gênica que o maigesPack envolvia. O aprendizado foi baseado em buscas sobre o assunto na internet e na tese de doutorado de Gustavo Henrique Esteves em bioinformática. Além dos conhecimentos em bioinformática foi necessário realizar um estudo sobre a linguagem R e sobre RUnit.O primeiro devido a linguagem base do pacote maigesPack e o segundo pelo fato de que o processo de refatoração prega a execução de testes unitários e como o maigesPack não possui testes em seu código foi necessário fazer um planejamento e aplicação dos testes.

6.1.2. Disciplinas relevantes

Dentre as disciplinas que eu cursei até o momento eu posso citar algumas que eu considero relevante para realizar esse trabalho são:

6.1.2.1. MAC0110 - Introdução à Computação

Essa disciplina me proporcionou o primeiro contato com alguma linguagem de programação, pois antes de entrar no curso de Bacharelado em Ciências da Computação no Instituto de Matemática e Estatística na Universidade de São Paulo eu nunca havia programado, nem sequer um "Hello World". A disciplina foi baseada em Java, desta forma foi introduzido lógica de computação e conceitos de orientação objetos, sendo que este último foi compreendido melhor posteriormente. Logo, eu considero que essa disciplina foi o meu primeiro passo na faculdade com relação a programação.

6.1.2.2. MAC0122 - Princípios de Desenvolvimento de Algoritmos

Eu considero essa disciplina importante, pois nela foi onde comecei a aprender lógicas mais complexas e desenvolver programas que exigiam mais de mim. Os algoritmos eram mais complexos comparados com os aprendidos em MAC0110, tais como algoritmos de busca (mergesort, quicksort, heapsort), algoritmos de encontrar padrões. Além dos algoritmos, a disciplina introduziu alguns conceitos de estrutura de dados (pilhas, filas e listas ligadas), busca de erros em programas, alocação de memória.

Todo esse conteúdo dado foi base para as demais disciplinas da área da computação. Sem elas seria muito difícil dar continuidade ao curso.

6.1.2.3. MAC0323 – Estrutura de Dados

O MaigesPack possui muitas estruturas de dados para armazenar os dados gênicos e muitas dessas estruturas são bem complexas. Com o estudo das estruturas de dados mostradas na disciplina, o processo de entendimento do pacote melhor aproveitado.

6.1.2.4. MAC0211 - Laboratório de Programação I

Essa disciplina foi importante, pois foi a primeira disciplina a oferecer aos alunos um projeto mais longo e não como nas demais disciplinas onde os programas tinham um prazo de um mês (em média) e eram dois ou três programas disjuntos. O projeto nessa disciplina foi um jogo que tinha algumas fases e no final obtivemos um jogo completo. Então tínhamos a preocupação de que o que foi feito na primeira fase poderia impactar no desenvolvimento das demais fases. Nesse caso o planejamento das fases foi importante, mas sem seguir alguma metodologia.

6.1.2.5. FLC0474 – Língua Portuguesa

Essa disciplina foi importante para manter contato com textos e escritas. Como o curso de Bacharelado em Ciências da Computação é uma área exata, fiquei muito tempo sem escrever textos e dissertações mais formais. Nos trabalhos e listas de exercícios não eram muito freqüentes, pois a resolução era muito mais técnica do que descritiva.

6.1.2.6. MAC0316 - Conceitos Fundamentais de Linguagens de Programação

Essa disciplina foi importante para o entendimento da linguagem R, pois na disciplina aprendi conceitos de linguagens que puderam ser usadas na melhor compreensão. Com esse estudo antecipado ficou mais fácil compreender o funcionamento da linguagem R, pois elas possuem características em comum.

6.1.2.7. MAC0342 - Laboratório de Programação Extrema

A importância dessa disciplina para a elaboração desse trabalho está relacionado as ferramentas e práticas utilizadas nessa metodologia de desenvolvimento de programas. A refatoração que é uma das práticas do XP (Extreme Programming) e a execução e criação de testes para garantir as funcionalidades devido as constantes mudanças que o código pode sofrer por causa da metodologia foram as mais relevantes.

6.1.3. Expectativas e planos futuros

Com a mudança de escopo do projeto, eu espero que possamos entregar as mudanças necessárias para atender aos novos requisito até o prazo limite, em fevereiro.

6.2. Fábio Yoshio Sato

6.2.1. Desafios e Frustrações

Neste projeto tivemos um grande de desafio que foi sair da nossa área de conhecimento, a computação. Apesar dos conhecimentos sobre biologia, análise microarray não serem essenciais para o desenvolvimento do projeto como ferramenta, acredito que seja de extrema importância conhecer a área de aplicação de um sistema a ser desenvolvido para que o desenvolvedor tenha uma visão mais abrangente e consiga entender a s necessidades dos usuários. Tive muita dificuldade particularmente em entender as técnicas estatísticas utilizadas na análise de dados microarray. Tivemos a experiência de tratar com um “cliente” no caso nosso orientador, vimos as dificuldades de se fazer o levantamento de requisitos, e tantas outras dificuldades da relação “cliente” X desenvolvedor. Uma grande

frustração foi não ter conseguido aplicar nossa primeira proposta de projeto que seria bastante interessante para nós, tratando de uma análise menos detalhada do sistema Maigespack e de mais alto nível, sem depender de peculiaridades da linguagem R.

6.2.2. Disciplinas relevantes

6.2.2.1. MAC0110 - Introdução à Computação

6.2.2.2. MAC0323 – Estrutura de Dados

Muito importante para entender os códigos do Maigespack que trabalham com estruturas bem complexas.

6.2.2.3. MAE0121 – Introdução à Probabilidade e à Estatística I

A familiaridade com métodos estatísticos me ajudou bastante para entender os códigos e a função das técnicas na análise microarray.

6.2.2.4. FLC0474 – Língua Portuguesa

6.2.2.5. MAC0332 – Engenharia de Software

Principal disciplina para o projeto na minha opinião, pois mesmo que não tenhamos aplicado nenhuma metodologia formal no desenvolvimento do projeto essa matéria nos ensinou as práticas usadas em desenvolvimento de softwares.

6.2.2.6. MAC0316 – Conceitos Fundamentais de Linguagens de Programação

O contato com linguagens diferentes principalmente as interpretadas nos ajudou muito no aprendizado da linguagem R.

6.2.3. Expectativas e planos futuros

Estamos no meio de uma mudança de escopo do projeto, espero que possamos fazer a modularização de forma a ajudar o usuário do Maigespack a ter maior flexibilidade nas suas análises.

6.3. Ricardo Issao Shimanuki

6.3.1. Desafios e Frustrações

Neste trabalho de formatura, certamente, os meus maiores desafios foram: a interdisciplinaridade e a modelagem do trabalho.

A interdisciplinaridade pelo fato de integrar três disciplinas: biologia, estatística e computação. Na parte biológica, meus conhecimentos se restringiam ao

conteúdo aprendido no ensino fundamental e médio, ou seja, eram apenas superficiais. Foi necessário um maior estudo, nem tão superficial e nem tão profundo, para compreender e desenvolver a proposta. Na área estatística, relembramos a parte de normalização e correlação, entre outros processos. Por fim, na parte computacional, para somar aos nossos conhecimentos de C, tivemos que aprender mais sobre a linguagem estatística de programação R, na qual está baseada maior parte deste projeto.

Outro desafio encontrado foi a modelagem da proposta. Devido ao primeiro desafio, tivemos dificuldade de formular os objetivos do trabalho. Quando escolhemos e o colocamos em prática, percebemos que nosso desafio poderia ser aprimorado. Por isso, decidimos acrescentar objetivos à nossa proposta inicial e apresentar este novo trabalho de formatura em fevereiro.

A frustração que tive foi de não ter elaborado e aplicado este novo escopo para o projeto anteriormente. Acreditamos que o que havíamos decidido inicialmente fosse o suficiente para desenvolver um ótimo trabalho, mas nossa auto-crítica nos fez repensar e decidir reestruturar nossa proposta.

6.3.2. Disciplinas relevantes

6.3.2.1. MAC0110 – Introdução à Computação

Esta matéria foi o princípio de todo o aprendizado de computação. Antes de entrar no curso, nunca havia programado. Por isso, tive que aprender todos os conceitos, lógicas e técnicas desde o início. Graças a essa base, tive maior facilidade para aprender, entender e desenvolver algoritmos posteriormente.

6.3.2.2. MAE0121 – Introdução à Probabilidade e à Estatística I

Esta disciplina foi bastante importante, pois através dela tive o primeiro contato com análises estatísticas e ela me proporcionou uma base sólida. Por se tratar de um projeto interdisciplinar, muitas dessas análises e técnicas aprendidas nesta matéria é utilizada no nosso projeto. Por exemplo, no MaigesPack, para transformar um objeto da classe maigesRaw pra classe maiges são utilizados diversos tipos de normalização, que foi aprendida nesta matéria.

6.3.2.3. MAE0212 – Introdução à Probabilidade e à Estatística II

Esta matéria é uma continuação da Introdução à Probabilidade e à Estatística I. Nela, também foram apresentadas análises estatísticas novas para mim e que foram muito úteis para o desenvolvimento do trabalho. Como exemplo, posso citar o conceito e a técnica de correlação lecionada nesta disciplina e aplicada nas análises de redes de relevância do MaigesPack.

6.3.2.4. MAC0122 – Princípios de Desenvolvimento de Algoritmos

Como o próprio nome diz, esta matéria foi fundamental para começar a desenvolver algoritmos. Herdadas algumas idéias da Introdução à Computação, aqui se iniciou todo o raciocínio e aprendizado de técnicas para elaborar algoritmos. Comecei a ver pontos importantes como a complexidade dos algoritmos através de estudos empíricos.

6.3.2.5. FLC0474 – Língua Portuguesa

Considero esta uma matéria muito importante para qualquer profissão, trabalho ou disciplina. Para este projeto, esta matéria contribuiu muito para todos os relatórios, como a avaliação das monografias anteriores, formulação da proposta, elaboração da apresentação e a criação desta monografia.

6.3.2.6. MAC0316 – Conceitos Fundamentais de Linguagem de Programação

Como neste trabalho de formatura, utilizamos uma linguagem estatística de programação chamada R, que em certos pontos se assemelha às linguagens aprendidas nesta disciplina, utilizei diversos conhecimentos aprendidos nesta matéria para poder aplicar no nosso projeto.

6.3.2.7. MAC0332 – Engenharia de Software

Essa matéria foi fundamental para encarar segundo desafio explicado acima: modelagem do projeto. Muitos conhecimentos adquiridos nesta matéria foram aplicados, tais como gerenciamento de projeto, análise e especificação de requisitos, modelagem de dados, projeto estruturado e testes.

6.3.3. Expectativas e planos futuros

Devido à alteração do nosso objetivo final, espero que na entrega da versão final do nosso projeto tenhamos alcançados três metas: atingir nossa

proposta, desenvolver um ótimo trabalho e fazer com que nosso trabalho auxilie ou aperfeiçoe o pacote MaigesPack.

7. Bibliografia

[1] CRISTO, Elier Broche - Métodos Estatísticos de Análise de Experimentos Microarray, 2003.

[2] BEISSBARTH, Tim; RUSCHHAUPT, Markus; JACKSON, David; LAWRENZ, Chris; Mansmann, Ulrich - Recommendations for normalization of microarray data [online]. Disponível na internet via WWW. URL: http://www.science.ngfn.de/dateien/Recommendations_for_normalization_of_microarray_data.pdf. Arquivo capturado em 12 de novembro de 2008**.**

[3] BILBAN, Martin; BUEHLER, Lukas K.; HEAD, Steven; DESIYE, Gernot; QUARANTA, Vito – Normalizing DNA Microarray Data

[4] WIKIPEDIA. Projeto Genoma [online]. Disponível na internet via WWW. URL: http://pt.wikipedia.org/wiki/Projeto_Genoma. Arquivo consultado em 4 de novembro de 2008.

[5] IRWIN, Richard D.; BOORMAN, Gary A.; CUNNINGHAM, Michael L.; HEINLOTH, Alexandra N.; MALARKEY, David E. - Application of Toxicogenomics to Toxicology: Basic Concepts in the Analysis of Microarray Data [online]. Disponível na internet via WWW. URL: http://tpx.sagepub.com/cgi/content/abstract/32/1_suppl/72. Arquivo consultado em 24 de setembro de 2008.

[6] GHANEM, Moustafa - Introduction to Bioinformatics - Microarrays2: Microarray Data Normalization

[7] TURNER, Heather. Clustering Microarray Data [online]. Disponível na internet via WWW. URL: http://www2.warwick.ac.uk/fac/sci/statistics/staff/research/turner/turnerchapter3.pdf. Arquivo capturado em 24 de setembro de 2008**.**

[8] Prof. Abraham B. Korol - Microarray cluster analysis and applications, 2003

[9] NCBI – National Center for Biotechnology Information. Molecular Biology Review [online]. Disponível na internet via WWW. URL: http://www.ncbi.nlm.nih.gov/Class/MLACourse/Modules/MolBioReview/images/mrna.gif. Arquivo capturado em 18 de novembro de 2008**.**

[10] FOWLER, Martin; BECK, Kent; BRANT, John; OPDYKE, William; ROBERTS, Don - Refactoring: Improving The Design of Existing Code. Addison-Wesley Object Technology Series, 1999.

[11] Sem Autor. Refactoring – Definição [online]. Disponível na internet via WWW. URL: http://www.javafree.org/wiki/Refactoring. Arquivo consultado em 20 de setembro de 2008.

[12] R PROJECT ORG. R - The R Projectfor Statistical Computing [online]. Disponível na internet via WWW. URL: http://www.r-project.org. Arquivo consultado em 3 de novembro de 2008.

[13] LANG, Jean-Phillipe. Refactoring R [online]. Disponível na internet via WWW. URL: http://www.r-developer.org/wiki/refactoring/RefactoringActivities. Arquivo consultado em 3 de novembro de 2008.

[14] EUGSTER, Manuel. Roxygen - Literate programming in R [online]. Disponível na internet via WWW. URL: http://roxygen.org. Arquivo consultado em 3 de novembro de 2008.

[15] BURGER, Matthias; JUENEMANN, Klaus; KOENIG, Thomas. RUnit Package [online]. Disponível na internet via WWW. URL: http://cran.r-project.org/web/packages/RUnit/RUnit.pdf. Arquivo capturado em 3 de novembro de 2008.

[16] ESTEVES, Gustavo H. - Métodos estatísticos para a análise de dados de cDNA microarray em um ambiente computacional integrado, Universidade de São Paulo, 2007