Universidade de Lisboa
Faculdade de Ciências
Departamento de Estatística e Investigação Operacional

Aplicação de Redes Bayesianas na Ciência Forense

Mestrado em Bioestatística

Alberto Miguel Oliveira Santos

Dissertação orientada por Prof. Marília Antunes
e Dr.ª Cláudia Silva

2015

Para a Sónia...

Agradecimentos

Em primeiro lugar, quero agradecer à Professora Marília Antunes e à Dr.ª Cláudia Silva por toda a orientação, ajuda e inspiração que me dedicaram durante a realização desta dissertação.

Também quero agradecer à minha irmã, Emília Santos, por ser uma influência extremamente positiva na minha vida.

Finalmente, quero agradecer à Sónia Pinto, a minha musa e o meu único verdadeiro amigo, por ter sempre acreditado em mim, por me aturar durante as alturas menos boas e por me ter motivado desde o início a concluir esta dissertação.

Resumo

O estado da arte corrente na valorização de provas científicas levanta problemas para os cientistas forenses devido à complexidade das provas. As provas científicas estão, regra geral, incompletas, o que leva os cientistas forenses a terem que lidar com incerteza e tomada de decisões. De um ponto de vista jurídico, é essencial que o investigador forense consiga apresentar provas com a validação científica exigida. Regra geral, esta validação é apresentada na forma de uma Razão de Verosimilhanças (*Likelihood Ratio*, LR), cujo cálculo pode ser bastante difícil face a dados com maior complexidade.

Desde o final da década de 1980, as Redes Bayesianas (RB) atraem investigadores na área da ciência forense. RB são modelos probabilísticos que ajudam os cientistas forenses a qualificar e, caso seja possível, quantificar os estados de conhecimento usando Teoria da Probabilidade para obter resultados, considerando o cálculo do LR, para aconselhar os seus clientes relativamente a significância dos seus resultados.

Uma RB codifica uma distribuição de probabilidade conjunta, através de um grafo direcionado acíclico construído tendo em conta a condição de Markov: um nodo do grafo é condicionalmente independente dos nodos que não são seus descendentes, dados os seus nodos pais. Tendo em conta a independência condicional, a distribuição de probabilidade conjunta pode ser facilmente calculada usando a regra da cadeia.

Nesta dissertação foram avaliados alguns casos de paternidade disputada usando RB com o *software* R, através de um *package* adequado à construção destes modelos. No entanto, pela sua complexidade de utilização, não é, de momento, uma ferramenta acessível à maioria dos investigadores forenses. Este trabalho tem como objetivo a construção de um programa que permita ao investigador forense a utilização do referido *package* de R, através a construção dos modelos gráficos para diversos casos, simples e mais complexos, de paternidade disputada.

Os resultados obtidos com o programa criado foram comparados com resultados obtidos através do *software* **familias** e foram satisfatórios.

Palavras-Chave: Redes Bayesianas, Ciência Forense, testes de paternidade, Razão de Verosimilhanças.

Abstract

The current state of the art in scientific evidence evaluation does not allow scientists to cope adequately with the problems caused by the complexity of the evidence. Scientific evidence is usually incomplete to some degree, thus uncertainty and decision making is a prevalent issue which forensic scientists have to deal with. From a juridical point of view, it is essential that the forensic investigator can present solid scientific evidence to a jury. Usually, this validation assumes the shape of a Likelihood Ratio (LR), whose calculation can be quite troublesome in cases of big complexity.

Since the late 1980's, Bayesian Networks have attracted researchers in forensic science. Bayesian Networks are probabilistic models that help forensic scientists qualify and, if possible, quantify their states of knowledge using probability theory to obtain the results required, regarding the calculation of a LR, to advise their clients of the significance of their findings.

A Bayesian Network codifies a joint probability distribution, using a directional acyclic graph, built with the Markov property in mind: a node in the graph is conditionally independent from its descendant nodes given its parents. Thus, given the conditional independence, the joint probability distribution can be easily computed using the chain rule.

In this thesis, several cases of disputed paternity were addressed with Bayesian Networks using R software, with an adequate package to build this type of models. However, given its complexity, it is not a accessible tool for most forensic investigators. This work aims to build a program that allows forensic analysts to use R to build graphic models for cases of disputed paternity.

The results obtained with the created program were compared with results obtained with the **familias** *software* and the program proved to be efficient.

Keywords: Bayesian Networks, Forensic Science, paternity tests, Likelihood Ratio.

Acrónimos

cgt	*child genotype*
cmg	*child maternal gene*
cpg	*child paternal gene*
DAG	Grafo acíclico orientado
gfgt	*grandfather genotype*
gfmg	*grandfather maternal gene*
gfpg	*grandfather paternal gene*
gmgt	*grandmother genotype*
gmmg	*gramdmother maternal gene*
gmpg	*grandmother paternal gene*
mgt	*mother genotype*
mmg	*mother maternal gene*
mpg	*mother paternal gene*
NTP	Nodo de tabela de probabilidade
pfgt	*putative father genotype*
pfmg	*putative father maternal gene*
pfpg	*putative father paternal gene*
RB	Redes Bayesianas
tfmg	*true father maternal gene*
tfpg	*true father paternal gene*

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Estado da arte

A investigação criminal e a ciência forense estão em constante evolução. Um dos avanços mais importantes dos últimos anos aconteceu em 1985, quando Allec Jefreys descobriu que porções da estrutura do ácido desoxirribonucleico (DNA) de determinados genes eram únicas para cada indivíduo, tal como impressões digitais (este procedimento ficou conhecido como *fingerprinting*) (Thein, Jeffreys e Wilson 1985). Esta descoberta criou as fundações da ciência forense moderna.

Por ciência forense entende-se a aplicação de um conjunto de disciplinas científicas para responder a questões que surjam num tribunal. Assim, a ciência forense desempenha uma função essencial no sistema de justiça, ao fornecer informação científica fundamental para a investigação criminal e para os tribunais.

Em particular, a análise de evidência de DNA tornou-se importante nos sistemas legais de todo o mundo. Esta análise consiste na realização de exames periciais em amostras com origem conhecida, que consistem atualmente, na determinação de um perfil genético (ou perfil de DNA), contribuindo para o esclarecimento de relações biológicas de parentesco, paternidade ou maternidade na sua maioria, identificação de desconhecidos, ou identificação dos contribuintes de amostras biológicas relacionadas com processos de natureza criminal (Silva 2011).

Importante notar que o cientista forense analisa eventos passados e, em geral, este não acede à informação exata dos factos decorridos na cena do crime. Neste contexto, o cientista forense recorre às opiniões, testemunhos e provas físicas como principal fonte de informação, sendo agregado maior valor quanto mais conhecimento os especialistas conseguirem extrair do conjunto de provas disponíveis. Pode-se assim dizer que não é finalidade desta ciência chegar a uma verdade absoluta, mas antes, encontrar meios que apresentem a melhor justificação que otimize o acesso do júri às respostas sobre o que realmente terá acontecido num dado caso (Chadreque 2012).

Este facto coloca os cientistas forenses numa posição interessante. Por um lado, um cientista forense faz trabalho experimental de laboratório, puramente determinístico; por outro lado, tem de interpretar a evidência de casos singulares. Este último aspeto do trabalho de um cientista forense é explicitamente subjetivo visto que cada caso criminal (ou civil) acontece uma e só uma vez, e requer uma interpretação subjetiva da probabilidade de forma a interpretar probabilidades como graus de credibilidade (Lucy 2005).

De acordo com o paradigma bayesiano, o termo *probabilidade* refere-se sempre a um grau de credibilidade de um dado evento. Dado uma sequência de observações, os graus de credibilidade de cada evento são "reciclados", através do Teorema de Bayes, de forma a ter em conta a nova informação observada. Desta forma, uma abordagem bayesiana fornece aos cientistas forenses uma fórmula apropriada para calcular de que forma a evidência observada influência um caso e comunicar estes resultados como uma razão de verosimilhanças (*LR - likelihood ratio*) (Taroni et al. 2010).

Na análise forense de DNA em estudos de paternidade existe sempre um certo grau de incerteza já que apesar de cada ser humano possuir um perfil genético único, em estudos forenses é impossível analisarmos todo o genótipo dos indivíduos envolvidos e não existe uma base de dados com todos os indivíduos da população portuguesa, ou seja, existe a necessidade de se utilizar inferências a partir da análise de uma amostra.

Existem vários métodos formais para ajudar os cientistas forenses a entender todas as dependências que podem existir entre diferentes aspetos de evidências observadas e lidar com a tomada de decisões mediante essas mesmas evidências. Em particular, métodos gráficos, como as Redes Bayesianas (RB), têm sido bastante úteis para representar as relações entre as características de interesse em situações de incerteza, imprevisibilidade ou imprecisão (Taroni et al. 2006).

Abordagens gráficas com meios genuínos para lidar com incerteza, em particular através da teoria de probabilidade, começaram a estimular investigadores legais à cerca de duas décadas. Em particular, desde o início da década de 90, tanto advogados como cientistas forenses começaram a mostrar interesse em RB para estudar questões relacionadas com a avaliação de evidência. Enquanto os advogados se preocupavam mais com a estruturação dos casos no seu todo, os cientistas forenses estavam maioritariamente focados na avaliação de itens específicos de evidências científicas (Biedermann e Taroni 2012).

Ao contrário de outros métodos para lidar com a incerteza, é necessário uma grande introspeção teórica assim como conhecimento prático para explorar todas as oportunidades fornecidas pelas RB e grafos de decisão (Jensen 2001).

1.2 Estrutura da dissertação

Os seguintes parágrafos dão uma ideia geral dos tópicos de cada um dos capítulos desta dissertação.

No **Capítulo 2**, são descritos os principais conceitos de teoria de probabilidade que suportam este tipo de modelos.

No **Capítulo 3**, são descritas as componentes de uma RB em detalhe, começando por descrever a composição de um grafo acíclico direcionado (*Directional acyclic graph* - DAG em inglês). De seguida, é descrita a forma como as RB atualizam e propagam a informação com base em novas evidências.

No **Capítulo 4**, é descrito o objetivo dos testes de paternidade, os problemas que surgem no decorrer destes testes e como as RB podem ajudar a lidar com esses problemas e ajudar os analistas forenses a tirar conclusões nestes casos.

No **Capítulo 5**, são explorados três casos de paternidade disputada, de natureza diferente, utilizando o programa criado em R e comparado os valores obtidos de LR com os obtidos no *software* **familias**.

No **Capítulo 6**, são discutidos os resultados obtidos e tiradas as conclusões sobre o trabalho efetuado. São também feitas algumas sugestões para futuros projetos no âmbito desta dissertação.

1.3 Objetivos

Nesta dissertação o foco do estudo são casos de paternidade disputada e o objetivo é criar um programa em R para investigadores forenses, de simples utilização, que permita calcular a razão de verosimilhança de casos de paternidade disputada, através de modelos gráficos, nomeadamente Redes Bayesianas.

O programa será criado para acomodar três casos de paternidade disputada distintos. Um deles é o caso tipo com um trio familiar e os outros dois casos mais complexos. Será também capaz de abordar os problemas mais comuns em casos de paternidade disputada como a falta de informação sobre o pressuposto pai, mutação e também é tido em consideração a possibilidade de aparecimento de alelos inexistentes nos marcadores genéticos e para os quais é atribuída uma frequência relativa mínima.

O programa criado terá ainda em conta a inserção de dados, por parte do utilizador forense, de forma a facilitar a utilização da plataforma criada em R e permitir calcular rapidamente o valor do LR pretendido.

Capítulo 2

Conceitos de Probabilidade e Estatística

Neste capítulo serão abordados alguns conceitos de probabilidade indispensáveis à construção de Redes Bayesianas, assim como outras propriedades inerentes a este tipo de modelos gráficos e, posteriormente, mostrar argumentos a favor da utilização de Redes Bayesianas em testes de paternidade disputada.

Neste contexto, o termo *probabilidade* significa o grau de credibilidade que temos de um certo acontecimento ter ocorrido ou não. Não sabemos se este realmente aconteceu, apenas que acreditamos com um certo grau de certeza que se tenha dado o acontecimento.

De forma a conseguirmos 'medir' o nosso grau de credibilidade relativamente à ocorrência de certo acontecimento A, é necessário definir uma função de probabilidade que atribua um número à possibilidade da ocorrência do acontecimento A. Um grau de credibilidade numérico tem de satisfazer as regras fundamentais da Teoria da Probabilidade:

1. $0 \leq P(A) \leq 1$.
2. $P(\Omega) = 1$.
3. $P(A \cup B) \leq P(A) + P(B)$.

   pois se o número de casos favoráveis a A ou a B não excede a soma do número de casos favoráveis a A com o número de casos favoráveis a B. No caso de A e B serem disjuntos verifica-se igualdade.
4. $P(\overline{A}) = 1 - P(A)$.
5. A probabilidade de se verificar A sem se verificar B é

$$P(A - B) = P(A) - P(A \cap B).$$

6. Se $A_1, ..., A_n$ são acontecimentos disjuntos dois a dois, então é válida a *regra da adição*,
$$P(\bigcup_{k=1}^{n} A_k) = \sum_{k=1}^{n} P(A_k)$$
7. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
8. Se A e B são acontecimentos independentes, então é válida a *regra da multiplicação*,
$$P(A \cap B) = P(A) \times P(B)$$

Numa perspetiva bayesiana, a probabilidade depende da informação disponível e portanto, é natural assumir que a probabilidade de dado acontecimento deve ser reavaliada à medida que a informação muda (Pestana e Velosa 2010).

A inferência bayesiana tem em conta a informação *a priori*, anterior ou externa em relação à amostra ou experiência, que deriva normalmente dos conhecimentos acumulados no domínio em causa, bem como da intuição ou sensibilidade do investigador, e a informação obtida através da evidência observada, procedendo à sua combinação através do Teorema de Bayes (Murteira e Antunes 2012a).

2.1 Probabilidade Condicional

Sempre que um cientista forense avalia a probabilidade de um acontecimento, este tem de ter em conta toda a informação *a priori* sobre o caso. A nova evidência irá mudar a probabilidade *a priori* e consequentemente, reavaliar a probabilidade existente. Somos assim conduzidos ao conceito de probabilidade condicional.

Definição 2.1.1. Sejam A e B acontecimentos de um espaço Ω e I a informação *a priori*. A probabilidade condicional de A sabendo que B aconteceu, dado I, é definida por

$$P(A|B, I) = \frac{P(A \cap B|I)}{P(B|I)}.$$

A probabilidade $P(A|B, I)$ representa uma reavaliação da probabilidade de A utilizando a informação adicional de que B ocorreu, tendo em conta a informação *a priori* I. Por inversão, podemos calcular a probabilidade conjunta de dois acontecimentos, dado I:

$$P(A \cap B|I) = P(A|B, I)P(B|I),$$

que, através da regra da cadeia, podemos generalizar para n acontecimentos.

Definição 2.1.2 (Regra da cadeia)**.** Sejam $A_1, A_2, ..., A_n$ acontecimentos definidos em Ω tais que $P(A_1, A_2, ..., A_n) > 0$. Então,

$$P(A_1, A_2, ..., A_{n-1}, A_n, I) = P(A_n|A_1, ..., A_{n-1}, I)...P(A_2|A_1, I)P(A_1|I).$$

Caso este condicionamento não altere a avaliação da probabilidade dizemos que os acontecimentos são independentes. O conceito de independência desempenha um papel fundamental na Teoria da Probabilidade (Mello 2000).

2.2 Independência

Intuitivamente, dois acontecimentos dizem-se independentes se a realização de qualquer um deles não influencia nem é influenciada pela realização do outro.

Definição 2.2.1. Os acontecimentos A e B dizem-se independente se e só se,

$$P(A \cap B) = P(A) \times P(B).$$

Outra extensão do conceito de independência é a seguinte: dois acontecimentos A e B são condicionalmente independentes em relação a um acontecimento C, quando

$$P(A \cap B|C) = P(A|C)P(B|C).$$

A independência de acontecimentos permite uma simplificação notável das expressões que envolvem probabilidade conjunta (Pestana e Velosa 2010).

Generalizando a regra da cadeia para n acontecimentos, se cada um dos A_k acontecimentos for independente dos outros acontecimentos $A_1, A_2, ..., A_{k-1}$ o condicionamento torna-se irrelevante e portanto temos

$$P(A_1, A_2, ..., A_{n-1}, A_n) = P(A_n)P(A_{n-1})...P(A_2)P(A_1).$$

ou, de uma forma mais condensada.

$$P\left(\bigcap_{k=1}^{n} A_k\right) = \prod_{k=1}^{n} P(A_k).$$

2.3 Teorema de Bayes

A simples definição de probabilidade condicional, e a expressão da regra da cadeia, arrastam consigo um dos resultados mais produtivos de toda a Teoria da Probabilidade (Pestana), o Teorema da Probabilidade Total.

Teorema 2.3.1 (Teorema da Probabilidade Total)**.** Seja B um acontecimento, e $\{A_n\}_{n\in\mathbb{N}}$ uma partição do universo Ω em acontecimentos e I a informação *a priori*. Então,

$$P(B|I) = \sum_{n\in\mathbb{N}} P(B|A_n, I)P(A_n|I).$$

O teorema 2.3.1 relaciona as probabilidades marginais com as probabilidades condicionais, e a sua utilidade reside em permitir conhecer a probabilidade de um acontecimento complexo B através de uma partição de acontecimentos $\{A_n\}_{n\in\mathbb{N}}$ mais simples.

O Teorema de Bayes não é mais do que um corolário do Teorema da Probabilidade Total.

Teorema 2.3.2 (Teorema de Bayes)**.** Seja $\{A_n\}_{n\in\mathbb{N}}$ uma partição do universo Ω em acontecimentos, B um acontecimento de Ω e I a informação *a priori*. Então

$$P(A_i|B, I) = \frac{P(B|A_i, I)P(A_i|I)}{\sum\limits_{n\in\mathbb{N}} P(B|A_n, I)P(A_n|I)}, \quad i = 1, 2, ..., \mathbb{N}.$$

O teorema 2.3.2 permite reavaliar a probabilidade dos acontecimentos A_i, de uma partição $\{A_n\}_{n\in\mathbb{N}}$, quando se obtém a informação adicional de que o acontecimento B se realizou. Os acontecimentos A_i tomam o nome de "causas" e $P(A_i|B, I)$ dá então a probabilidade de que o acontecimento B que ocorreu seja o resultado da causa A_i dado a informação *a priori* I.

A probabilidade $P(A_i|I)$ toma o nome de probabilidade *a priori* da "causa" A_i, tendo em conta a informação I, e $P(A_i|B, I)$ é a probabilidade *a posteriori*, isto é, a probabilidade de A_i calculada após a informação de que B se realizou, tendo em conta a informação I (Mello 2000).

A importância deste teorema deve-se ao facto de se tratar de uma regra para atualizar o grau de credibilidade face a uma nova evidência observada (Taroni et al. 2006). Reescrevendo o teorema 2.3.2 na forma equivalente, e omitindo a informação I para simplificar a notação, temos

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)},$$

com $P(E) = P(H)P(E|H) + P(\overline{H})P(E|\overline{H})$, onde H representa a hipótese proposta e E a evidência observada. Se entretanto houver alguma evidência adicional E',

$$P(H|E \cap E') = \frac{P(H|E')P(E|H \cap E')}{P(H|E')P(E|H \cap E') + P(\overline{H}|E')P(E|\overline{H} \cap E')},$$

pondo em relevo a natureza recursiva do reajustamento processado com o Teorema de Bayes (Murteira e Antunes 2012b).

2.4 Razão de Verosimilhanças

Os conceitos de chances (*odds*) *a priori* e *a posteriori* são correntemente introduzidos no quadro do teorema de Bayes. Sejam E a evidência, H e $\overline{H}$ a hipótese proposta e a hipótese contrária, e $P(H)$ e $P(\overline{H})$ as respetivas probabilidades *a priori*. A razão

$$\mathcal{O}_0(H;\overline{H}) = \frac{P(H)}{P(\overline{H})},$$

traduz a chamada chance *a priori* de H *versus* $\overline{H}$. Considerando que, pelo Teorema de Bayes,

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)}, \qquad P(\overline{H}|E) = \frac{P(\overline{H})P(E|\overline{H})}{P(E)},$$

tem-se a razão

$$\mathcal{O}_0(H;\overline{H}|E) = \frac{P(H|E)}{P(\overline{H}|E)} = \frac{P(H)}{P(\overline{H})} \times \frac{P(E|H)}{P(E|\overline{H})} = \mathcal{O}_0(H;\overline{H}) \cdot L_E(H;\overline{H}),$$

a chamada chance *a posteriori* de H *versus* $\overline{H}$ face à evidência E. Como se verifica pela expressão acima, as chances *a posteriori* podem obter-se como o produto de dois fatores. O primeiro representa a chance *a priori*, o segundo a razão de verosimilhanças $L_E(H;\overline{H}) = \frac{P(E|H)}{P(E|\overline{H})}$, operador que modifica o primeiro fazendo intervir as verosimilhanças das hipóteses proposta e contrária, decorrentes da evidência (Murteira e Antunes 2012b).

2.5 Variáveis Aleatórias Discretas

Numa análise forense, é inevitável que um analista se depare com situações em que precisa de ter em conta uma variável numérica. Estas podem ser *discretas* ou *contínuas*. Nos casos práticos em estudo nesta dissertação, os marcadores genéticos analisados são *variáveis aleatórias discretas*, e por isso, apenas este tipo de variável aleatória será descrita. De forma a simplificar a notação, a informação *a priori*, I, será omitida.

Definição 2.5.1 (Variável aleatória discreta)**.** X é uma variável aleatória discreta se e só se existir um conjunto finito ou numerável de pontos $\mathcal{S} = \{x_k\}_{k\in\mathbb{K}}$, o suporte da variável aleatória, tal que

$$P(X = x_k) > 0 \quad e \quad P(X \in \mathcal{S}) = \sum_{k\in\mathbb{K}} P(X = x_k) = 1.$$

Geralmente, a distribuição de uma variável aleatória é explicitada na seguinte forma,

$$X = \begin{cases} x_k & k \in \mathbb{K} \\ p_k = P(X = x_k) & \end{cases}$$

onde na primeira linha se declara o suporte da variável, constituído pelos pontos em que assenta a probabilidade positiva, e na segunda linha a probabilidades desses pontos. É imediato que a função massa de probabilidade tem as seguintes características (Pestana e Velosa 2010):

$$1. \quad \forall\, k \in \mathbb{K},\ p_k \geq 0; \qquad 2. \quad \sum_{k\in\mathbb{K}} p_k = 1,$$

Por exemplo, num caso de paternidade disputada, $\mathcal{S} = \{x_k\}_{k\in\mathbb{K}}$, corresponderá ao conjunto dos alelos de determinado marcador genético X, com $\#\mathbb{K}$ alelos.

Para um conjunto de *variáveis aleatórias discretas*, $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ com suporte $\mathcal{S}$. A sua *distribuição de probabilidade conjunta* é dada por:

$$\mathbf{X} = \begin{cases} (k_1, ..., k_n) & (k_1, ..., k_n) \in \mathcal{S} \\ p_{(k_1,...,k_n)} = P(X_1 = k_1, ..., X_n = k_n) & \end{cases}$$

De notar que, através da regra da cadeia, a distribuição de probabilidade conjunta de duas variáveis aleatórias X e Y, pode ser escrita como o produto de duas outras

distribuições de probabilidade, sendo uma delas a distribuição de probabilidade da variável X e a outra a *distribuição de probabilidade condicional* da variável Y, condicional a X (Taroni et al. 2006):

$$\begin{aligned} P(X = x, Y = y) &= P(X = x) \times P(Y = y|X = x) \\ &= P(Y = y) \times P(X = x|Y = y) \end{aligned}$$

Nesta dissertação as distribuições de probabilidades condicionais serão apresentadas na forma de tabelas como na Tabela 2.1 (com $v = verdadeiro$ e $f = falso$).

Tabela 2.1: Distribuição de probabilidade condicional de X dado Y.

		$Y:$	v	f
$X:$	$P(X = v\|Y)$		0.3	0.9
	$P(X = f\|Y)$		0.7	0.1

Neste contexto, a distribuição de probabilidade de X pode ser obtida pela soma da sua distribuição de probabilidade conjunta com Y para todos os estados possíveis y tal que,

$$P(X = x) = \sum_{y} P(X = x, Y = y).$$

A esta expressão chamamos a *probabilidade marginal* de X.

Se considerarmos por exemplo que a variável Y da Tabela 2.1 tem $P(Y = v) = 0.8$ e $P(Y = f) = 0.2$, na Tabela 2.2 aparecem representadas as probabilidades marginais e distribuição de probabilidade conjunta de X e Y. Onde por exemplo,

$$P(Y = v, X = v) = P(Y = v) \times P(X = v|Y = v) = 0.8 \times 0.3 = 0.24.$$

Tabela 2.2: Distribuição de probabilidade conjunta de X dado Y e suas marginais.

		$Y:$	v	f	Marginal
$X:$	$P(X = v\|Y)$		0.24	0.18	0.42
	$P(X = f\|Y)$		0.56	0.02	0.58
		Marginal	0.8	0.2	1

Calcular probabilidades condicionais através das probabilidades marginais é bastante simples. Na Tabela 2.3 (arredondada à segunda casa decimal) está representada a distribuição de probabilidade condicional de Y dado X derivada da Tabela 2.2 onde, por exemplo,

$$P(Y = v|X = v) = \frac{P(Y = v, X = v)}{P(X = v)} = \frac{0.24}{0.42} = 0.57.$$

Tabela 2.3: Distribuição de probabilidade condicional de X dado Y

	$X:$	v	f
$Y:$	$P(Y = v\|X)$	0.57	0.97
	$P(Y = f\|X)$	0.43	0.03

As probabilidades marginais podem ser generalizadas para mais do que duas variáveis. A distribuição de probabilidade conjunta de um conjunto de variáveis aleatórias pode ser decomposto através da regra da cadeia num produto de distribuições de probabilidade para cada uma das variáveis individualmente.

Capítulo 3

Redes Bayesianas

Ao contrário do que o nome possa sugerir, uma RB não depende de um raciocínio bayesiano. O termo deriva do facto de uma RB utilizar inferência bayesiana para atualizar a probabilidade dos estados de variáveis não observadas através do conceito de independência condicional. Este conceito permite utilizar métodos computacionais bastante eficazes, e rápidos, para efetuar cálculos que de outra forma seriam bastante complexos. Uma RB é assim, uma caracterização explícita de dependências diretas entre um conjunto de variáveis. Esta caracterização toma a forma de um grafo acíclico orientado (*directed acyclic graph* - DAG em inglês) e um conjunto de nodos com tabelas de probabilidade (NTP) subjacentes (Mello 2000):

- *DAG* - Consiste num conjunto de nodos e arcos. Os nodos representam variáveis aleatórias e os arcos representam relações causais diretas ou de dependência entre variáveis.
- *NTP* - Para cada variável X com um conjunto de nodos pais $Y_1, Y_2, ..., Y_n$, existe uma tabela de probabilidade condicional associada $P(X|Y_1, Y_2, ..., Y_n, I)$ em que I representa toda a informação relevante que não pode ser explicitada no grafo.

3.1 Grafos Acíclicos Orientados

Um grafo consiste num conjunto V de vértices (ou nodos) e um conjunto A de arcos (ou arestas) que conectam um par de nodos. Cada nodo corresponde a uma variável e os arcos denotam relações entre cada par de variáveis. Cada arco de um grafo pode ser direcionado ou não direcionado. Em particular, grafos que são direcionados e não têm nenhum ciclo, denominam-se grafos acíclicos orientados (DAG). Todas as RB são representadas graficamente por um DAG.

Definição 3.1.1 (Grafo acíclico orientado)**.** Seja V um conjunto finito de nodos. Então um grafo $G = (V, A)$ onde os elementos de V representam os nodos de G e $A \subseteq \{a \to b | a, b \in V, a \neq b\}$ representa o conjunto dos arcos de G. Se existe uma ordem $v_1, ..., v_d$ de nodos, consistente com G, i.e., $v_i \to v_j \in A$ então, $i < j$, então G é um grafo acíclico orientado (DAG).

Se existe um arco dirigido a ligar um nodo X a um nodo Y, então diz-se que X é *pai* de Y e Y é *filho* de X, e designamos o conjunto de pais de um dado nodo Z por $pais(Z)$. Na Figura 3.1 (i) por exemplo, o conjunto de nodos pais do nodo D é: $pais(D) = \{B, C\}$. Um nodo sem nodos pais é designado por nodo raiz (por exemplo, o nodo A na Figura 3.1 (i)).

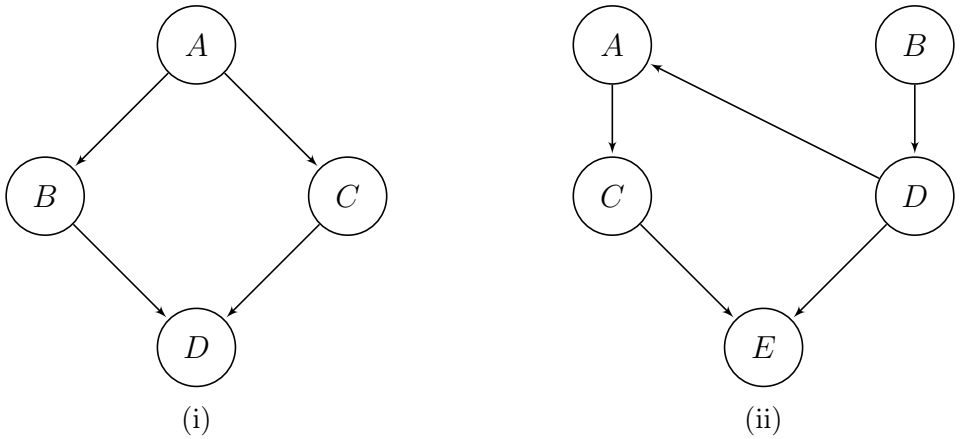


Figura 3.1: Exemplos de Grafos aciclícos orientados.

Outro conceito importante é o de *caminho* entre dois nodos. Um caminho entre os nodos X e Y, consiste numa sequência de arcos consecutivos, independentemente da direção destes, que conecta dois nodos X e Y. Por exemplo, na Figura 3.1 (ii) existem dois caminhos que conectam os nodos A e E: um é o caminho $A - C - E$ e o outro o caminho $A - D - E$ [10].

Um nodo pode ter nodos *ancestrais* e nodos *descendentes*. Um nodo X é ancestral de um nodo Y (e Y é descendente de X) caso exista um caminho entre X e Y, em que estes estão ligados por nodos intermédios (Taroni et al. 2006). Na Figura 3.1 (ii), o nodo B é pai de D e ancestral de todos os outros nodos do grafo.

3.2 Nodos com Tabelas de Probabilidade

Numa RB, a teoria de grafos é utilizada para criar uma estrutura qualitativa de um modelo e a teoria da probabilidade é usada para caracterizar a natureza e força das relações que governam esse modelo. As RB utilizam nodos e arcos da mesma forma que outros modelos gráficos. A característica que distingue as RB é os NTP. Estes permitem implementar formalmente a teoria de probabilidade para interpretar

as relações entre componentes do modelo gráfico e podem acomodar informação *a priori* ou informação subjetiva sobre determinado caso (Biedermann e Taroni 2012).

Seja X uma variável aleatória com n estados $x_1, ..., x_n$. Um *estado* é um valor possível de uma variável aleatória. Se X for um nodo raiz, então a tabela de probabilidade condicional $P(X|I)$ será uma tabela com n entradas que contém a distribuição de probabilidade $P(X = x_i), i = 1, ..., n$, com $\sum_{i=1}^{n} P(X = x_i) = 1$. Para simplificar a notação, a informação I foi omitida. Seja Y outra variável aleatória com m estados $y_1, ..., y_m$. Se Y é descendente de X, então a tabela de probabilidade condicional P(X | Y) será uma tabela $n \times m$ que contém todos os valores das probabilidades $P(X = x|Y = y)$ (Biedermann e Taroni 2012). Por exemplo, consideremos as variáveis B, C e D da Figura 3.1(i) e admitamos que são variáveis binárias, ou seja, todas têm apenas dois estados possíveis. Então a tabela de probabilidade condicional para o nodo D (representada na Tabela 3.1) tem 8 entradas $P(D = d_i|B = b_j, C = c_k) = p_{ijk}$, com $i = 1, 2; j = 1, 2; k = 1, 2$.

Tabela 3.1: Tabela de probabilidade condicional para o nodo D e seus pais B e C.

		$B:$	b_1		b_2	
		$C:$	c_1	c_2	c_1	c_2
$D:$	$P(D = d_1\|B = b_j, C = c_k)$		p_{111}	p_{112}	p_{121}	p_{122}
	$P(D = d_2\|B = b_j, C = c_k)$		p_{211}	p_{212}	p_{221}	p_{222}

As RB têm uma estrutura computacional incorporada para calcular o efeito de uma nova evidência nos estados das variáveis. Essa estrutura permite:

1. atualizar probabilidades dos estados das variáveis após inserção de nova evidência;
2. utilizar relações de independência probabilística, tanto as explicitamente como as implicitamente representadas no modelo gráfico, para tornar o cálculo computacional mais eficiente.

3.3 Definição de Rede Bayesiana

Como já vimos, a regra da cadeia permite decompor uma distribuição de probabilidade conjunta com n variáveis da seguinte forma:

$$P(X_1, ..., X_n) = \left[\prod_{i=2}^{n} P(X_i|X_1, ..., X_{i-1})\right] P(X_1).$$

Uma forma mais expedita de representar esta distribuição de probabilidade conjunta pode ser obtida se considerarmos as relações de dependência entre as variáveis do

grafo. Afinal, uma das grandes vantagens de uma representação gráfica deste tipo é tornar perfeitamente visível as relações de dependência entre as variáveis. Assim, uma das propriedades fundamentais de uma RB, conhecida na literatura por *Condição de Markov*, diz o seguinte:

1. Dada uma variável X_i de um DAG, esta é condicionalmente independente de todos as outras variáveis do DAG, dado o seu conjunto $pais(X_i)$.
2. Uma RB pode ser fatorizada como o produto, para todas as variáveis do DAG, das suas probabilidades condicionadas apenas pelos seus nodos pais:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|pais(X_i)).$$

Ou seja, qualquer nodo do DAG é condicionalmente independente dos seus não descendentes dado o conjunto dos seus nodos pais.

A equação em 2. designa-se por, *regra da cadeia para redes bayesianas*, e define formalmente o que significa uma RB: uma representação da distribuição de probabilidade conjunta para todas as variáveis (Biedermann e Taroni 2012). Aplicando esta regra no DAG (i) da Figura 3.1 temos:

$$P(A, B, C, D) = P(A)P(B|A)P(C|A)P(D|B, C).$$

À primeira vista, esta simplificação não altera muito o cálculo de uma probabilidade conjunta. No entanto, a nível computacional, a diferença é abismal. Em vez de termos de calcular e representar 2^n probabilidades de uma dada distribuição de probabilidade conjunta com n variáveis, as relações de independência condicional permitem-nos apenas ter em conta as probabilidades $P(X_i|pais(X_i))$ para cada variável X_i que, regra geral, são bastante menos do que 2^n (Thein, Jeffreys e Wilson 1985).

3.4 Critério de d-separação

Numa RB, determinamos que nodos serão atualizados com base numa evidência, através das estrutura das dependências (ou independências) condicionais da rede. Para tal, necessitamos ter em mente os diferentes tipos de relações entre variáveis e as várias formas como podem ser representadas num DAG (Fenton e Neil 2012). As relações de independência podem ser obtidas através da estrutura do DAG utilizando o critério designado por *d-separação* (o d vem de *direcional*).

Definição 3.4.1 (d-separação)**.** Seja $G = (V, A)$ um DAG. Diz-se que os nodos X e Y de V estão d-separados por um conjunto de nodos $Z \subseteq V$ quando, para todos os caminhos entre X e Y, se verifica uma das seguintes condições:

1. Existe um nodo em Z no caminho entre X e Y de forma que a conexão entre X e Y é em série;

2. Existe um nodo em Z no caminho entre X e Y de forma que a conexão entre X e Y é em divergente;

Suponhamos que a variável A está ligada às variáveis B e C. Existem três tipos diferentes de conexão entre estas variáveis como representado na Figura 3.2. Estas podem ser em *série* (i), *divergentes* (ii) ou *convergentes* (iii) (Pearl 2000).

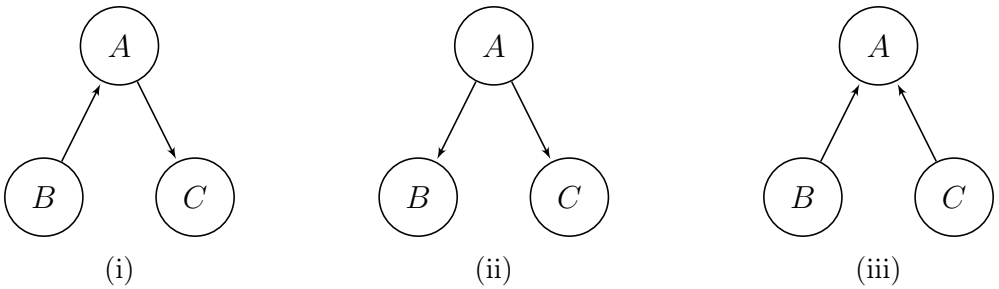


Figura 3.2: Variáveis relacionadas por diferentes tipos de conexão.

Analisando os grafos da Figura 3.2, no grafo (i), temos uma conexão em série, ou seja, a evidência acerca de B, é transmitida através de A para C. Suponhamos no entanto que temos informação sobre o estado de A. Qualquer informação acerca de B é irrelevante para C, pois qualquer informação sobre A sobrepõe-se. Ou seja, numa conexão em série B e C são condicionalmente independentes dado A, ou duma forma equivalente, B e C estão d-separados dado A (Fenton e Neil 2012).

No grafo (ii) temos uma conexão divergente. Neste caso, qualquer informação sobre A será transmitida para ambos os nodos B e C. Suponhamos no entanto que não temos informação sobre A e queremos saber se informação sobre C pode ser transmitida para B. Se tivermos alguma informação sobre C, esta pode ser transmitida para A através do teorema de Bayes, e consequentemente, transmitida para B. No entanto, se temos informação sobre A, qualquer nova informação sobre C, não altera a nossa crença em B, pois a certeza sobre A torna a informação sobre C irrelevante para B. Ou seja, numa conexão divergente, B e C são condicionalmente independentes dado A, ou duma forma equivalente, B e C estão d-separados dado A (Fenton e Neil 2012).

No grafo (iii) temos uma conexão convergente. Aqui, qualquer informação sobre B ou C irá influenciar A. Mas o que queremos saber é se podemos transmitir informação entre B e C. Se não temos informação sobre A, então é claro que B e C são independentes. Nestas condições, dizemos que B e C são condicionalmente dependentes dado A, ou por outras palavras, B e C são d-conectados dado A. Ou seja, numa conexão convergente, novas evidências só podem ser transmitidas entre os pais B e C quando o nodo convergente A recebeu alguma evidência (Fenton e Neil 2012).

Conexões convergentes descrevem um tipo de raciocínio mais sofisticado e um dos trunfos das RB é conseguir lidar com este tipo de raciocínio. Este tipo de conexões são apropriadas se sabemos que B e C são ambos relevantes para A, B não é relevante para C mas torna-se relevante perante a informação sobre A (e *vice-versa*) (Biedermann e Taroni 2012). Por exemplo, um dado genótipo gt é composto por dois alelos, um proveniente do lado materno, e outro do lado paterno como representado na Figura 3.3 (pg e mg representam os alelos herdados do lado paterno e materno respetivamente). Os alelos materno e paterno são independentes. No entanto, se tivermos informação sobre quais os alelos constituintes de gt, por exemplo se soubermos que gt é constituído pelos alelos {12,13}, sabemos que do lado paterno ou materno, estes alelos tem de estar presentes.

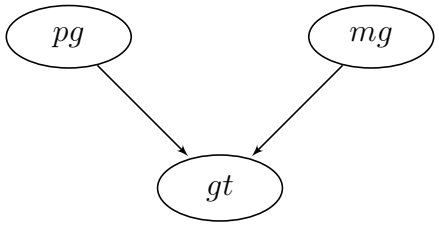


Figura 3.3: Rede bayesiana de um genótipo.

As conexões convergentes tem outra peculiaridade: para passar informação, não é necessário que o estado da variável para a qual se converge seja conhecido, é suficiente conhecer apenas alguma informação sobre este, mesmo que não nos dê certezas sobre o verdadeiro estado da variável. Na literatura sobre RB, é comum denominar à informação que fixa o estado duma variável como *evidência específica* e, caso contrário, *evidência virtual* (*hard evidence* e *soft evidence*, respetivamente, em inglês). Por outras palavras, evidência específica permite-nos definir o verdadeiro estado duma variável com probabilidade 1, e no caso de evidência virtual, temos de definir o estado da variável em causa com probabilidade menor que 1 (Biedermann e Taroni 2012).

3.5 Propagação de evidência em Redes Bayesianas

Uma vez propriamente definida a Rede Bayesiana, esta pode ser consultada para processar nova informação adquirida, ou seja, calcular as probabilidades condicionais dos nodos dado os valores que foram observados para alguns dos nodos da rede. Esta informação, num contexto forense, consiste em perceber qual o verdadeiro estado de uma dada suposição sobre um dado caso. O conhecimento sobre o verdadeiro estado de uma dada variável da rede iria permitir inferir sobre o estado das outras variáveis de interesse. Neste trabalho, esta informação corresponde a dados numéricos gerados pela análise de perfis genéticos (referentes aos alelos) dos intervenientes num caso de paternidade disputada. Estes dados servem de *input* na RB (Biedermann e Taroni 2012).

O processamento de informação é uma tarefa fundamental numa RB e é um dos

principais pontos de interesse no estudo de inferência probabilística na ciência forense. De facto, a principal característica de uma RB é, como o nome implica, ser capaz de reavaliar probabilidades, dada uma evidência em particular, através do teorema de Bayes. Uma ilustração gráfica conveniente desta característica consiste em considerar uma rede com dois nodos com variáveis binárias, como na Figura 3.4.

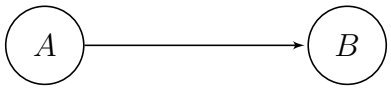


Figura 3.4: Rede bayesiana com dois nodos.

A probabilidade conjunta neste caso pode ser fatorizada como $P(A, B) = P(A)P(B|A)$. Se observarmos que B se verifica, interessa-nos calcular $P(A|B)$. De forma a aplicarmos o teorema de Bayes neste caso, temos de ter em conta as seguintes probabilidades:

1. $P(B) = P(B|A)P(A) + P(B|\overline{A})P(\overline{A})$;
2. $P(A, B) = P(A)P(B|A)$;
3. $P(A|B) = \dfrac{P(A, B)}{P(B)}$.

Neste caso, a evidência em B é "enviada" para A no sentido oposto da orientação dos arcos da rede. É importante notar que também é possível propagar informação no sentido dos arcos numa RB. A propagação no sentido $A \rightarrow B$ significa que teríamos de calcular a probabilidade de B dado A, ou seja, $P(B|A)$. O que sucede quando passamos de $P(B)$ para $P(B|A)$ pode ser observado na equação do ponto 1. Esta relação diz que para avaliarmos a incerteza de B, temos de ter em conta a incerteza sobre A. Quando é sabido que A "ocorre", ficamos a saber que $P(A) = 1$ e $P(\overline{A}) = 0$. Consequentemente, a equação do ponto 1. resume-se ao cálculo de $P(B|A)$ e este valor corresponde ao valor especificado na tabela de probabilidade condicional do nodo B da rede (Biedermann e Taroni 2012).

Este exemplo, embora extremamente simples, descreve a forma como o cálculo probabilístico é feito numa RB. Desta forma, dada uma RB é possível calcular a probabilidade *a posteriori* de qualquer evento definido sobre as variáveis da rede. Este procedimento é denominado por *atualização de crença*. Uma vantagem da atualização de crença em RB é que os algoritmos que as implementam são capazes de explorar as relações de independência expressas na rede para reduzir o esforço computacional do processo (Rocha, Guimarães e Campos 2011).

Capítulo 4

Testes de paternidade disputada

Hoje em dia, os testes de paternidade disputada são feitos com base em perfis de DNA. Estes consistem na análise de determinados marcadores genéticos, denominados como *short tandem repeats* (STR). STR são unidades de repetição de pequeno tamanho, variável, que correspondem na prática a fragmentos de DNA com grande dispersão no genoma humano (Silva 2011).

Para cada marcador, podemos observar o seu genótipo, que é composto por dois alelos, um proveniente do lado materno e outro do lado paterno (embora não seja possível saber qual é qual). Cada marcador tem um número finito de valores possíveis, correspondentes aos alelos (Dawid et al. 2002). Se os alelos localizados na mesma posição do cromossoma dos marcadores apresentarem uma estrutura igual, o indivíduo é *homozigótico* e, caso contrário, o indivíduo é *heterozigótico* (Kobilinsky, Liotti e Oeser-Sweat 2005).

Atualmente, são utilizados em laboratório, na rotina dos casos forenses, os marcadores CSF1PO, D2S1338, D3S1358, D5S818, D7S820, D8S1179, D13S317, D16S539, D18S51, D19S433, D21S11, FGA, Penta D, Penta E, TH01, TPOX e VWA, incluídos nos *kits* comerciais *Powerplex® Y System* (Promega Corporation) e *AmpFlSTR® Y FIler® PCR amplification kit* (Applied Biosystems™, Foster City California). São estes os marcadores considerados no âmbito desta dissertação.

Estes marcadores são escolhidos por se encontrarem em cromossomas diferentes e portanto, segregam-se de forma independente. Regra geral, é razoável assumir que as populações procriam de forma aleatória e que se verifica o equilíbrio de Hardy-Weinberg, ou seja, vamos assumir independência entre estes marcadores (Dawid et al. 2002).

As frequências alélicas de cada marcador foram estimadas tendo em conta a base de dados referente à população do Sul de Portugal, descrita em Vieira-Silva, T. Ribeiro e Espinheira (2006).

Nestes casos, interessa-nos inferir sobre a identificação de indivíduos com base nas evidências observadas após a análise do DNA. Geralmente, a abordagem para tal envolve o cálculo de uma *razão de verosimilhanças* (*likelihood ratio*, LR), tendo em conta os dados disponíveis para as várias hipóteses concorrentes. Porém, há situações em que podem não estar disponíveis amostras para um ou mais indivíduos de interesse. Nestes casos temos apenas informação indireta relevante, geralmente fornecida por familiares próximos. É nestas situações, com dados incompletos ou em falta, que o cálculo de LR se pode tornar conceptualmente e computacionalmente exigente, em particular se considerarmos a possibilidade de ocorrer mutação no processo de transmissão genética (Dawid et al. 2002).

4.1 Caso simples de paternidade disputada

O caso mais simples de paternidade disputada consiste num trio familiar (i.e., um pai, uma mãe e uma criança), onde é disputada a alegação de que um determinado homem é o pai de uma certa criança. A veracidade da maternidade nunca é posta em causa. Nestas circunstâncias, os analistas forenses têm disponíveis os perfis de DNA da mãe m, da criança c, e do pressuposto pai pf (*putative father*). Uma representação daquilo que chamamos *pedigree familiar*, aparece ilustrada na Figura 4.1, onde um quadrado representa um indivíduo do sexo masculino, um círculo um indivíduo do sexo feminino e tf (*true father*) denota o verdadeiro pai (Dawid et al. 2002).

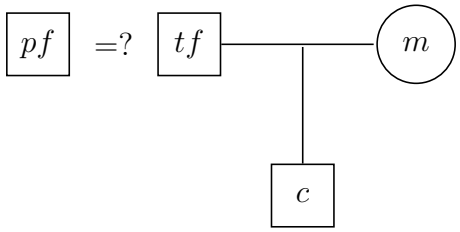


Figura 4.1: *Pedigree* familiar num caso de paternidade disputada simples.

Nestas circunstâncias, temos informação sobre os perfis de DNA de m, c e pf, que constituem a evidência ε. Interessa-nos, com base nestes dados, calcular o valor de LR para a hipótese de pf ser o verdadeiro pai, tf ($pf = tf$). Regra geral, a hipótese de interesse H_0 e a hipótese alternativa H_1 a testar são:

- H_0: O pai biológico é o pressuposto pai.
- H_1: O pai biológico é um indivíduo desconhecido da população.

Nesta dissertação serão sempre estas as hipóteses consideradas. O impacto da evidência ε é medido pelo valor de LR a favor da paternidade (também designado por *índice de paternidade*) (Dawid, Mortera e Vicard 2007):

$$LR = \frac{P(\varepsilon|H_0)}{P(\varepsilon|H_1)}.$$

Para encontrar os valores de LR temos de calcular a probabilidade conjunta da tripla de genótipos observados, sob cada uma das hipóteses tidas em conta. Para cada uma das hipóteses, o cálculo é simples: sob H_0, aplicamos as leis de segregação de Mendel; sob H_1 necessitamos das estimativas das frequências alélicas da população. Desta forma chegamos ao valor de LR descrito na equação 4.1. Utilizando o Teorema de Bayes, este LR pode ser combinado com a probabilidade *a priori* a favor da paternidade de *pf*, baseada em evidências externas, de forma a obtermos a probabilidade *a posteriori* a favor da paternidade (Dawid et al. 2002). Embora num panorama jurídico esta probabilidade possa variar, num dado teste de paternidade, o analista forense considera sempre que a probabilidade *a priori* a favor da paternidade é 0.5.

Sob o pressuposto de que todos os marcadores são independentes, é construída uma RB para cada um dos marcadores e calculado o respetivo LR. Estes valores podem depois ser multiplicados para obter o valor de LR global a favor da paternidade (Dawid et al. 2002). Transcrevendo um caso de um trio familiar para uma RB, e embora uma representação gráfica deste tipo seja flexível ao ponto de possibilitar várias representações, a RB da Figura 4.2 é baseada em Dawid et al. (2002), particularmente orientada para problemas do tipo em estudo nesta dissertação.

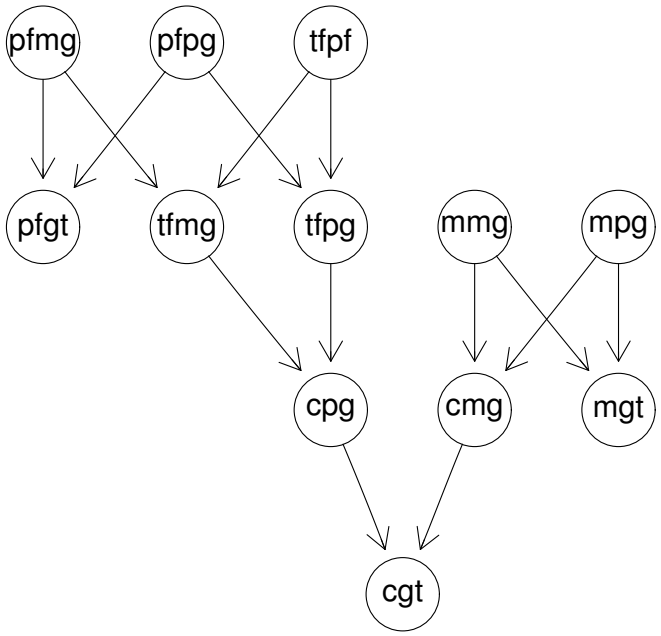


Figura 4.2: Rede bayesiana de um caso de paternidade disputada simples.

Os nodos $pfgt$, mgt e cgt são os nodos que dizem respeito à evidência observada e o nodo $tf = pf?$ (*true father = putative father ?*) representa um nodo adicional que incorpora as duas hipóteses concorrentes, H_0 e H_1, com os estados *yes* e *no*. Inicialmente estes estados são equiprováveis e após a propagação da evidência, a razão das suas probabilidades *a posteriori* pode ser interpretada como um valor de LR (Dawid, Mortera e Vicard 2007). Embora este nodo não seja obrigatório para a RB, após a inserção na RB da evidência observada, este proporciona uma forma

rápida e expedita de obter o valor de maior interesse neste tipo de casos: a razão de verosimilhanças (Dawid et al. 2002).

Para completar a representação de um caso de paternidade disputada referente a um trio familiar, temos de especificar as tabelas de probabilidade de cada nodo da RB. A título de exemplo, consideremos o marcador Penta D, de que observamos os alelos: $cgt = \{13, 13\}$, $mgt = \{13, 13\}$ e $pfgt = \{11, 13\}$ nos respetivos perfis genéticos. Nos nodos raiz (nodos sem pais), utilizamos as frequências alélicas da população, apresentadas na Tabela 4.1.

Tabela 4.1: Tabela de probabilidade para $pfpg$

$pfpg$:	11	13	x
	0.1616	0.1913	0.6471

Apenas os alelos observados são considerados na construção da tabela, assim como a agregação dos alelos não observados, representada por x.

O nodo referente à hipótese a testar, é inicializado com a probabilidade 0.5 para cada um dos seus estados, como apresentado na Tabela 4.2.

Tabela 4.2: Tabela de probabilidade para $tf = pf?$

$tf = pf?$:	*yes*	*no*
	0.5	0.5

As tabelas referentes à informação sobre o verdadeiro pai estão condicionadas à informação do pressuposto pai e ao nodo da hipótese, como representado na Tabela 4.3. Dependendo do estado (*yes* ou *no*) do nodo da hipótese $tf = pf?$, o gene paterno (ou materno se em vez de *tfpg* se considerar *tfmg*) do verdadeiro pai, ou corresponde ao alelo do pressuposto pai (são iguais com probabilidade igual a 1), ou então coincide com o alelo do verdadeiro pai com probabilidade igual à frequência observada na população.

Tabela 4.3: Tabela de probabilidade condicional para $tfpg$ dado $tf = pf?$ e $pfpg$

	$tf = pf?$	*yes*			*no*		
	$pfpg$	11	13	x	11	13	x
$tfpg$:	11	1	0	0	0.1616	0.1616	0.1616
	13	0	1	0	0.1913	0.1913	0.1913
	x	0	0	1	0.6471	0.6471	0.6471

As tabelas correspondentes a genótipos codificam de uma forma simples e determinística, as relações dos genes herdados dos lados materno e paterno, assim como os estados correspondentes a cada uma das combinações possíveis desses genes. Um exemplo está ilustrado na Tabela 4.4

Tabela 4.4: Tabela de probabilidade condicional de $pfgt$ dado $pfmg$ e $pfpg$

	$pfmg$	11			13			x		
	$pfpg$	11	13	x	11	13	x	11	13	x
$pfgt$:	11-11	1	0	0	0	0	0	0	0	0
	11-13	0	1	0	1	0	0	0	0	0
	11-x	0	0	1	0	0	0	1	0	0
	13-13	0	0	0	0	1	0	0	0	0
	13-x	0	0	0	0	0	1	0	1	0
	x-x	0	0	0	0	0	0	0	0	1

No caso do gene paterno (ou materno) herdado pela criança, e excluindo (para já) a possibilidade de ocorrer uma mutação, a criança recebe, aleatoriamente, um dos genes do pai com igual probabilidade (no caso homozigótico é certo qual o gene passado pelo pai (ou mãe)). A Tabela 4.5 corresponde a esta situação.

Tabela 4.5: Tabela de probabilidade condicional de cpg dado $tfmg$ e $tfpg$

	$tfmg$	11			13			x		
	$tfpg$	11	13	x	11	13	x	11	13	x
cpg :	11	1	0.5	0.5	0.5	0	0	0.5	0	0
	13	0	0.5	0	0.5	1	0.5	0	0.5	0
	x	0	0	0.5	0	0	0.5	0.5	0.5	1

Após a inserção da evidência observada nos nodos respetivos, esta informação é propagada pela rede com o uso do *software*. A distribuição de probabilidade atualizada condicionada à evidência pode ser obtida através de uma *query* no nodo da hipótese, podendo-se calcular o valor de LR para o marcador. Com base na evidência deste exemplo, obtemos a seguinte tabela:

Tabela 4.6: Tabela de probabilidade *a posteriori* para $tf = pf?$

$tf = pf?$:	*yes*	*no*
	0.7233	0.2767

O que corresponde a um $LR = \dfrac{0.7233}{0.2767} = 2.6140$ a favor da paternidade do pressuposto pai.

Nestes casos, o valor de LR pode ser calculado algebricamente sem recorrer a métodos computacionais (Dawid, Mortera e Vicard 2007). No entanto, é do nosso interesse aplicar este tipo de raciocínio em casos mais complexos, em particular casos com informação incompleta, nos quais este tipo de cálculo é bastante moroso.

4.2 Casos em que o pressuposto pai está ausente

Até agora, apenas considerámos situações em que a evidência está presente. No entanto, existem situações em que alguma da evidência pode estar ausente. Um dos objetivos ao usar uma rede probabilística passa também por acomodar este tipo de situações com informação genética de outros indivíduos com algum grau de parentesco com o pressuposto pai e permitir aos analistas forenses tirar conclusões sobre a paternidade (Dawid et al. 2002). É nestas situações, em que soluções puramente aritméticas se tornam consideravelmente mais complexas, que a RB representada na Figura 4.2 nos dá uma estrutura base importante, já que esta pode ser modificada e aumentada conforme a necessidade de cada caso (Taroni et al. 2006).

Consideremos agora um exemplo de um caso mais complexo em que temos os perfis genéticos de uma criança c e da sua mãe m, mas o perfil genético do pressuposto pai, pf, por alguma razão não está disponível. No entanto, temos os perfis genéticos dos seus pais, i.e., os pressupostos avô e avó (gf e gm respetivamente) da criança c. Na Figura 4.3 temos o *pedigree* deste caso e na Figura 4.4 a correspondente RB.

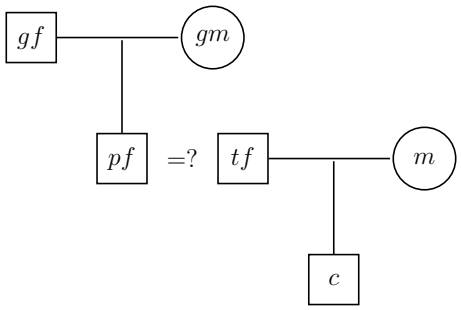


Figura 4.3: *Pedigree* familiar num caso de paternidade disputada sem informação sobre pf.

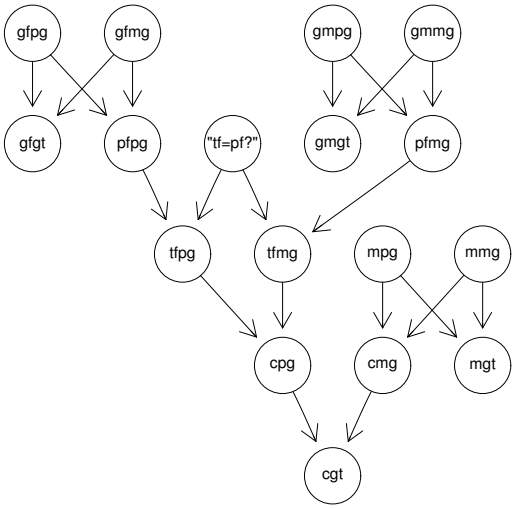
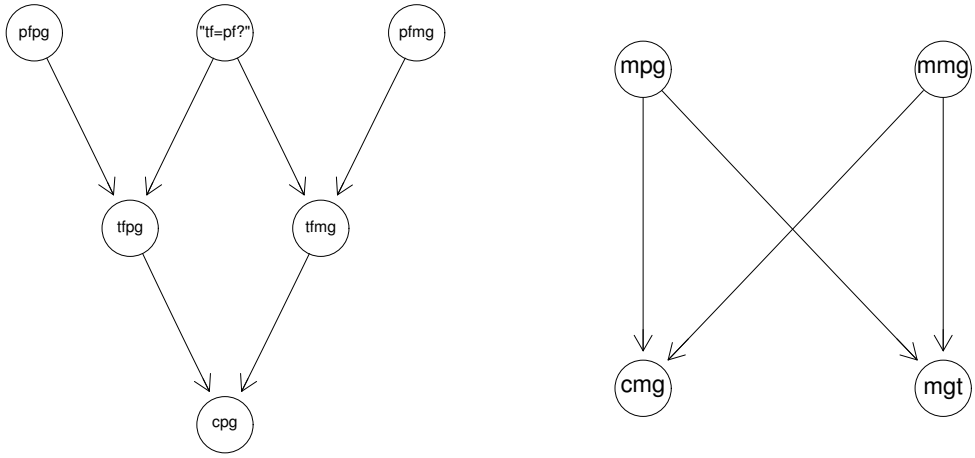


Figura 4.4: Rede bayesiana correspondente ao *pedigree* familiar da Figura 4.3.

Após a criação do DAG, as tabelas de distribuição de probabilidade correspondentes aos nodos são criadas e a evidência é inserida e propagada da mesma maneira que no caso simples de paternidade disputada discutido anteriormente. A inferência sobre o caso é obtida de forma análoga, analisando o nodo de hipótese após a propagação da evidência na rede. O software permite a realização destes cálculos de forma rápida e eficiente.

É de notar a capacidade de adaptação e elegância deste tipo de modelos nas situações referidas. Ao compararmos as redes nas Figuras 4.2 e 4.4, verificamos que estas têm vários *submodelos* de nodos comuns a ambas. O nodo de hipótese e os nodos que codificam se a informação genética do pressuposto pai se enquadra com o verdadeiro pai ou não estão sempre presentes nestes casos, assim como qualquer conjunto de nodos que represente a formação de um genótipo, como representado na Figura 4.5 em (i) e (ii) respetivamente.



(i) (ii)

Figura 4.5: Submodelos de uma Rede Bayesiana: (i) submodelo do nodo de hipótese e genótipo paterno herdado pela criança, *cpg*, em função da informação genética do seu pai; (ii) submodelo do genótipo materno da criança, *cmg*, construído em função da informação materna e paterna da sua mãe, assim como o genótipo materno, *mgt*.

Estes submodelos, desde que ligados entre si mediante a lógica do problema em mão, i.e., sem desrespeitar as relações de dependência (ou independência) entre variáveis, permitem criar RB, de uma forma flexível e intuitiva, para testes de paternidade disputada de uma forma bastante prática, mediante a informação que temos disponível para determinar a paternidade de determinado indivíduo.

4.3 Mutação

A possibilidade de ocorrer mutação numa determinada sequência genética entre gerações complica o processo de inferência forense. Num dado teste, um analista forense pode observar que o pressuposto pai apresenta uma incompatibilidade em um ou dois marcadores, quando todos os outros marcadores do perfil genético coincidem com os da criança. Um resultado deste género pode-se justificar pelo facto de ter ocorrido mutação, nesses marcadores, num gâmeta paternal. Um pressuposto pai pode assim ser excluído (a paternidade só é excluída quando aparecem pelo menos três marcadores incoerentes) com base na evidência do DNA quando, de facto, ele é o pai biológico mas uma mutação ocorreu e o pai transmitiu um alelo aparentemente impossível para a criança (Dawid, Mortera e Pascali 2001). Efetivamente, os marcadores tipicamente utilizados neste tipo de testes são particularmente propensos a sofrer mutação (Dawid et al. 2002). Por esta razão, é de extrema importância que testes de paternidade disputada tenham em conta a possibilidade da ocorrência de mutação na transmissão de genes.

Numa RB podemos incorporar todo o processo de mutação com um modelo adequado. Para tal, temos de considerar os alelos "originais" dos lados materno e paterno e, através de um processo específico de mutação, estes dão origem aos alelos "observados" da criança. Os alelos "observados" são, no fundo, os alelos que observamos ao analisar o genótipo da criança. Estendendo a RB para abranger este conceito, são criados dois nodos extra, *copg* (*child original paternal gene*) e *capg* (*child actual paternal gene*), que representam o "alelo paterno original" e "alelo paterno observado na criança" respetivamente (Dawid et al. 2002). Efetuando o mesmo raciocínio para o lado materno, obtemos a RB da Figura 4.6.

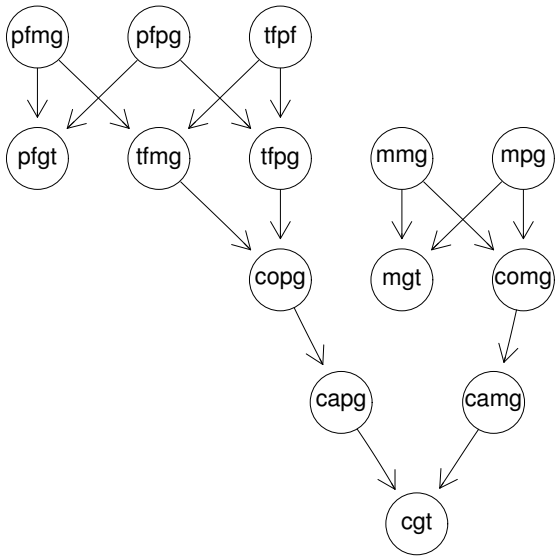


Figura 4.6: Rede bayesiana de um trio familiar com mutação.

Para acomodar totalmente o fenómeno da mutação, às tabelas de probabilidade já

definidas, temos de acrescentar agora uma matriz de transição aos nodos *capg* e *camg*, que irá representar a probabilidade de ocorrer mutação entre genes.

Há várias fontes onde as estimativas destas taxas podem ser obtidas. No entanto a informação sobre taxas de mutação entre alelos específicos é bastante escassa. Para esta dissertação foram consideradas as estimativas das taxas de mutação obtidas por Dauber et al. (2003), representadas na Tabela 4.7. Foi assumido que as taxas de mutação paternais e maternais são iguais.

Tabela 4.7: Estimativas das taxas de mutação em percentagem.

Marcador	Taxa de Mutação (μ)
CSF1PO	0.37
D2S1338	0.25
D3S1358	0.10
D5S818	0.13
D7S820	0.13
D8S1179	0.10
D13S317	0.13
D16S539	0.20
D18S51	0.10
D19S433	0.25
D21S11	0.29
FGA	0.22
Penta D	0.14
Penta E	0.16
TH01	0.04
TPOX	0.37
VWA	0.22

Para cada marcador, a matriz de transição, é representada por $Q = (q_{ij})$, onde q_{ij} denota a probabilidade de ocorrer uma mutação do gene "original" i, para o gene "observado" j. É razoável admitir que o vetor de frequências genéticas, $\pi = (\pi_1, ..., \pi_k)^T$, é constante ao longo do tempo. Então, π tem de ser uma distribuição estacionária para a matriz de transição Q associada, i.e. $\pi^T Q = \pi^T$. Dados π e a taxa de mutação μ, podemos construir a matriz de transição Q.

Existem vários modelos de mutação propostos na literatura. Nesta dissertação foi considerado o modelo de mutação estacionário descrito em Dawid, Mortera e Pascali (2001) e, como tal, foram tidos em conta os seguintes pressupostos:

1. Os *loci* dos STR têm um número finito de alelos;
2. Cada alelo pode mutar para um outro alelo qualquer;
3. A probabilidade de transição de um alelo para outro, diminui conforme a distância entre estes aumenta;

4. Estacionaridade: as frequências alélicas da população não se alteram com o processo da mutação.

Sejam $S = (s_{ij})$ uma matriz simétrica onde $s_{ij} \geq 0$ para $i \neq j$, $\sum_j s_{ij} = 0$ para todo o i e λ um parâmetro positivo ajustável. A matriz S é criada tendo em conta que é razoável, do ponto de vista biológico, assumir que um alelo tem maior probabilidade de mutar para um alelo seu vizinho do que para um mais distante. Especificamente, consideramos $s_{ij} = \alpha^{|i-j|}$, com $i \neq j$ e α uma constante fixa. Quanto menor o parâmetro α, maior a probabilidade que, caso uma mutação se verifique, esta seja num alelo próximo do alelo i.

Podemos definir o modelo de mutação da seguinte forma:

$$q_{ij} = \frac{\lambda s_{ij}}{\pi_i} \qquad (i \neq j), \qquad\qquad q_{ii} = 1 - \sum_{j \neq i} q_{ji} = 1 + \frac{\lambda s_{ii}}{\pi_i}.$$

Regra geral, a taxa de mutação μ é dada por

$$\mu = 1 - \sum_i \pi_i q_{ij} = \lambda \times \left(- \sum_i s_{ii} \right).$$

Logo, para o nosso modelo temos

$$\lambda = \frac{\mu}{- \sum_i s_{ii}}.$$

Para ilustrar um cenário em que, sem ter em conta a ocorrência de mutação, a paternidade do pressuposto pai seria excluída pelos dados, consideremos o seguinte exemplo de um caso de paternidade simples, descrito na Tabela 4.8, retirado de Dawid, Mortera e Pascali (2001).

De notar que, se a possibilidade de mutação não fosse sequer considerada, a paternidade do pressuposto pai seria excluída com base na informação dos marcadores VWA e FXIIIA1, mesmo que os marcadores restantes fossem aparentemente a favor da paternidade do pressuposto pai. Na Tabela 4.9 temos matriz de transição, com $\alpha = 0.5$, para o marcador VWA.

Os valores de LR dos marcadores concordantes com a paternidade de pf, com a introdução na rede da possibilidade de mutação, permanecem praticamente inalterados. Por outro lado, o efeito nos marcadores FXIIIA1 e VWA é considerável, alterando o valor de $LR = 0$ para um valor diferente de zero. Também na Tabela 4.9, foram considerados apenas os alelos observados e a agregação dos restantes alelos, x. Em princípio, uma agregação deste tipo poderia violar as propriedades de independência condicional de uma RB. No entanto, tendo em conta que as taxas de mutação são extremamente baixas, na prática, o efeito desta agregação é negligenciável (Dawid et al. 2002).

Tabela 4.8: Dados de um caso de paternidade disputada de um trio familiar.

Marcador	m		c		pf		LR
VWA	14	17	17	19	16	17	0.00277
FXIIIA1	7	16	5	7	7	7	0.000749
D21S11	28	34.2	32.2	34.2	32	32.2	5.01
TH01	6	7	6	7	6	6	2.36
FES	11	11	11	11	10	11	1.36
D13S317	11	12	12	12	12	12	3.24
D3S1358	16	16	15	16	15	15	3.74
TPOX	8	8	8	10	8	10	6.49
CSF1PO	10	10	10	12	9	12	1.77
D8S1179	10	12	10	13	11	13	1.44
D18S51	13	17	13	14	13	14	3.73
D7S820	9	11	9	10	10	12	1.95
FGA	22	24	22	24	23	24	1.57
Global							0.0763
Excluindo VWA							27.56
Excluindo FXIIIA1							101.87

Tabela 4.9: Tabela de probabilidade condicional para *capg* dado *copg* e *comg*.

	j				
i	14	16	17	19	x
14	0.997272	0.000391	0.000196	0.0000489	0.00209
16	0.000177	0.998652	0.000354	0.0000884	0.000729
17	0.000058	0.000234	0.999109	0.000117	0.000482
19	0.000065	0.00026	0.00052	0.996377	0.00278
x	0.000533	0.000412	0.000412	0.000533	0.99811

Em suma, utilizando o modelo de mutação proposto, o valor de LR global é 0.0763, ou seja, um valor que embora baixo não excluiria definitivamente a paternidade do pressuposto pai. Porém, considerando a exclusão de FXIIIA1, por exemplo, o valor de LR seria de 101.87, o que, assumindo uma probabilidade de paternidade *a priori* de 0.5, implicaria uma probabilidade *a posteriori* a favor da paternidade de 0.99. Concluímos assim que a simples exclusão de um ou dois marcadores, na realidade, não exclui efetivamente a paternidade se tivermos em conta a possibilidade de ocorrer mutação.

Capítulo 5

Casos Práticos

No âmbito desta dissertação, foram analisados três casos de paternidade disputada com o programa criado em R com o objetivo de obter um valor de LR que permita avaliar a favor (ou contra) a paternidade de um pressuposto pai.

Neste capítulo são resumidos alguns processos importantes para a correta utilização do programa criado, nomeadamente a inserção devida dos dados e são também revistas algumas propriedades das RB que precisaram ser tidas em conta na construção do programa em R. Para além disso são revistos os três casos trabalhados na dissertação.

Apesar de ter sido utilizado o *package* de R **gRain**, que cria as RB, foi necessário inserir no *software* as diferentes relações de parentesco entre os intervenientes e mediante a informação genética recolhida, para cada marcador, foi necessário criar toda uma estrutura subjacente que tem em conta todas as possibilidade combinatórias dos alelos de cada genótipo entre os intervenientes de cada caso para poder criar os estados de cada nodo do DAG da RB. Foi ainda necessário construir no R o modelo de mutação apropriado para cada um dos casos analisados.

Todos os casos estudados já foram tratados pelo Instituto de Medicina Legal (IML) e os resultados obtidos através do programa criado em R foram comparados com os valores obtidos pelo *software* utilizado pelos investigadores forenses no IML, o *software* **familias**.

Em todos os casos, foi assumida equiprobabilidade a favor da paternidade e foi ainda tido em conta a possibilidade de ocorrer mutação.

5.1 Introdução dos dados

Nos casos em estudo nesta dissertação, a nova evidência está resumida num *data frame* de forma a que o programa possa ler os dados e atualizar a crença sobre a

paternidade do pressuposto pai. A introdução dos dados sobre os alelos observados no R é feita através de um ficheiro .csv, criado pelo utilizador.

Para tal, inicialmente, o utilizador terá de ter um ficheiro .txt construído pelo analista forense após a análise laboratorial dos perfis genéticos dos intervenientes do caso, constituído por quatro colunas. Uma coluna para identificar a origem da amostra, outra para identificar o marcador analisado, e mais duas para cada um dos alelos observados (caso se verifique homozigotia apenas uma coluna é preenchida), tal como representado na Figura 5.1.

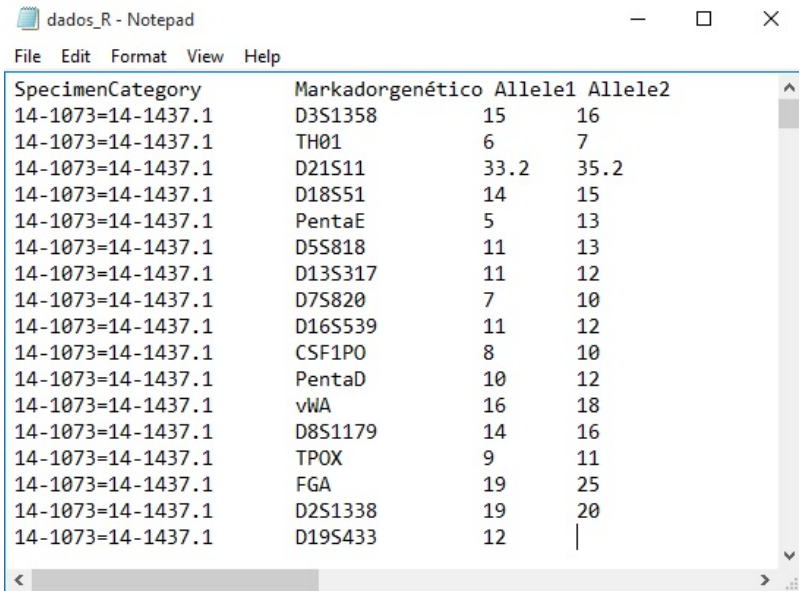


Figura 5.1: Ficheiro .txt com os alelos observados

Em seguida importamos esta informação para o Excel de forma a obter um ficheiro .csv para posteriormente podermos introduzir os dados no R.

Para criar o ficheiro .csv basta abrir o Excel, criar um novo livro (*book* na versão em inglês), ir ao menu dados (*data* em inglês) e importar um ficheiro .txt com as frequências alélicas, como descrito na Figura 5.2.

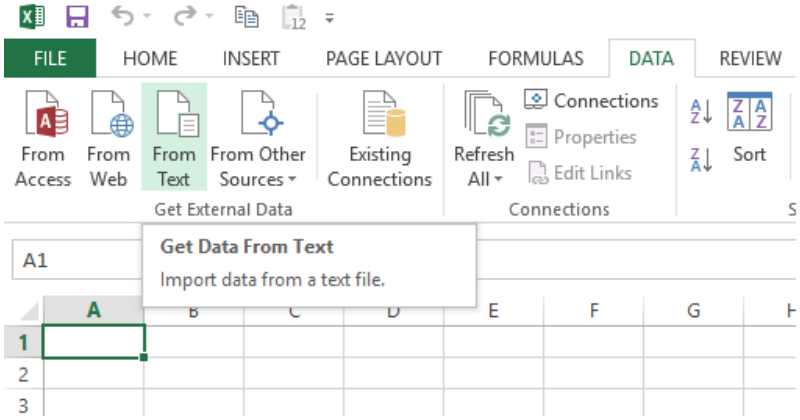


Figura 5.2: Como importar um ficheiro .txt para o Excel.

Para separar apropriadamente os dados no novo formato .csv temos que escolher as opções presentes nas Figuras 5.3 e 5.4.

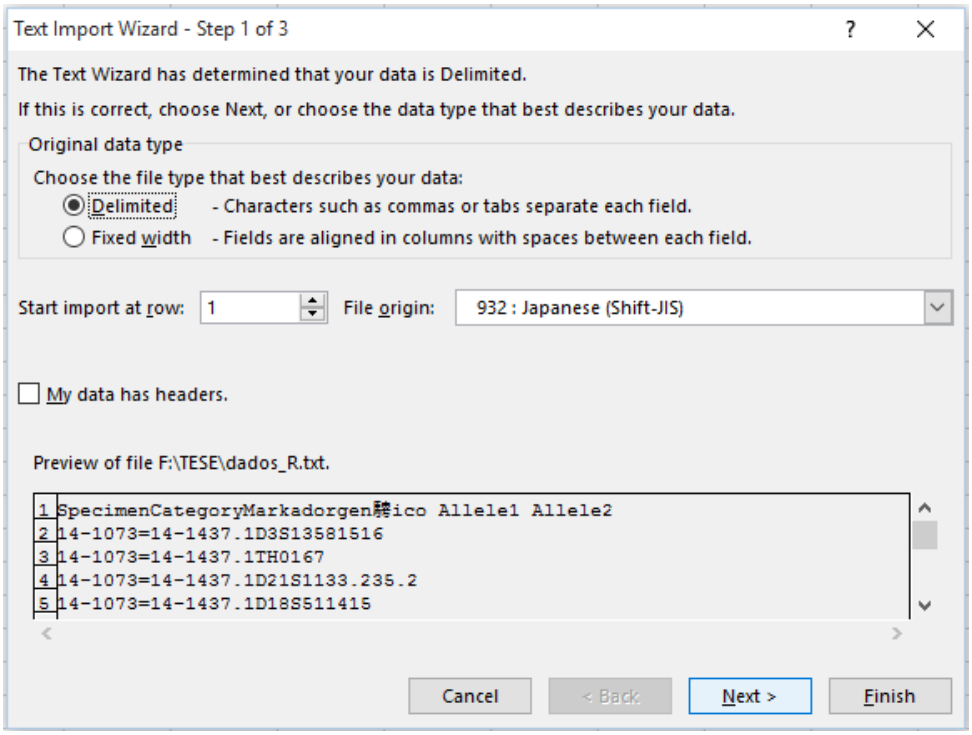


Figura 5.3: Opções a escolher para separar correctamente os dados no Excel I.

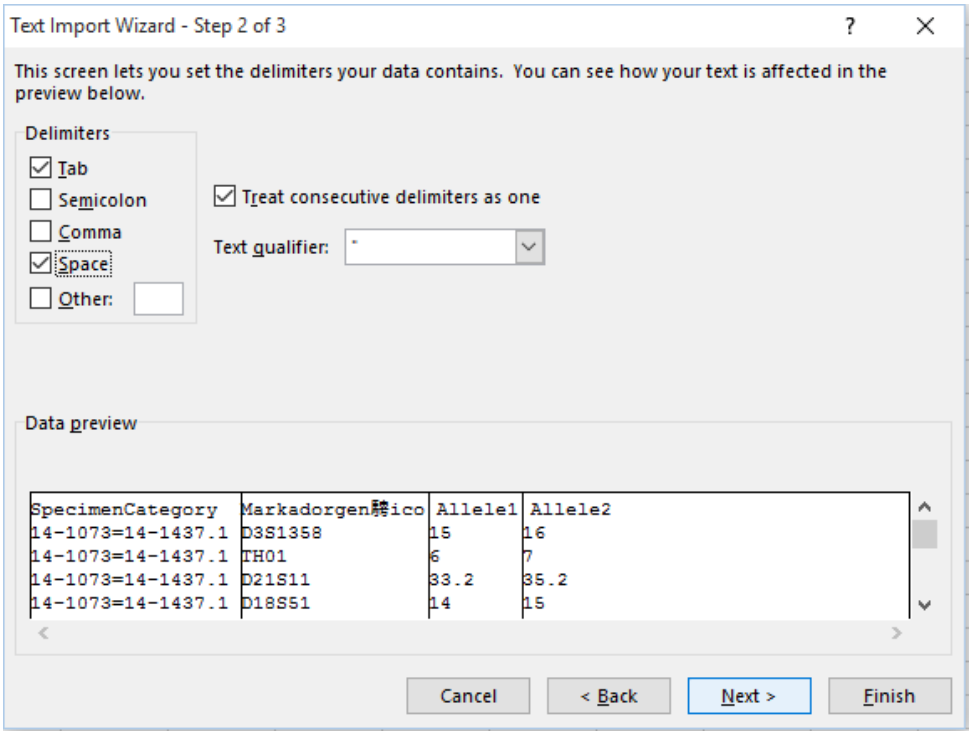


Figura 5.4: Opções a escolher para separar correctamente os dados no Excel II.

Finalmente temos de selecionar o local onde é colocada no Excel a informação importada do .csv (preferencialmente a célula $A1$).

Após a execução destes passos basta gravar o ficheiro Excel com o formato CSV (Comma Delimited) e a informação fica preparada para ser inserida no R através do seguinte comando:

```
read.csv("Dados_R.csv", header = T, sep = ";")
```

Desta forma criamos no R um *dataframe*, como o representado na Figura 5.5, com os alelos observados e as respetivas frequências alélicas, para cada marcador genético, de cada um dos intervenientes do caso em estudo.

	marker	pf	fr_pf	m	fr_m	c
1	D8S1179	14.0	0.2525	13.0	0.2901	14.0
2	D8S1179	14.0	0.2525	14.0	0.2525	14.0
3	D21S11	28.0	0.1577	30.2	0.0354	28.0
4	D21S11	30.0	0.2402	32.0	0.0088	32.0
5	D7S820	10.0	0.2896	10.0	0.2896	10.0
6	D7S820	11.0	0.2266	11.0	0.2266	10.0
7	CSF1PO	10.0	0.2657	12.0	0.3215	11.0
8	CSF1PO	11.0	0.3155	12.0	0.3215	12.0
9	D3S1358	16.0	0.2629	16.0	0.2629	16.0
10	D3S1358	16.0	0.2629	17.0	0.2007	17.0
11	TH01	7.0	0.1871	7.0	0.1871	7.0
12	TH01	9.0	0.1906	8.0	0.1449	9.0
13	D13S317	11.0	0.3256	12.0	0.2793	11.0
14	D13S317	11.0	0.3256	12.0	0.2793	12.0
15	D16S539	10.0	0.0602	9.0	0.1355	9.0
16	D16S539	11.0	0.3010	9.0	0.1355	11.0
17	D2S1338	17.0	0.2415	23.0	0.1034	17.0
18	D2S1338	19.0	0.1045	26.0	0.0127	26.0
19	D19S433	11.0	0.0112	13.0	0.2534	13.0
20	D19S433	13.0	0.2534	15.2	0.0455	15.2
21	vWa	16.0	0.2255	17.0	0.2611	16.0
22	vWa	16.0	0.2255	21.0	0.0026	17.0
23	TPOX	11.0	0.2870	8.0	0.4927	9.0
24	TPOX	12.0	0.0327	9.0	0.1065	11.0
25	D18S51	10.2	0.0006	18.0	0.0709	10.2
26	D18S51	15.0	0.1438	22.0	0.0024	18.0
27	D5S818	11.0	0.3349	12.0	0.3676	12.0
28	D5S818	12.0	0.3676	13.0	0.1724	13.0
29	FGA	22.0	0.1864	23.0	0.1484	23.0
30	FGA	23.0	0.1484	25.0	0.0817	25.0
31	PentaE	8.0	0.0288	8.0	0.0288	8.0
32	PentaE	12.0	0.1968	11.0	0.1254	8.0
33	PentaD	5.0	0.0033	7.0	0.0095	7.0
34	PentaD	10.0	0.1144	8.0	0.0206	10.0

Figura 5.5: *Data frame* no R com as frequências alélicas dos marcadores genéticos de um trio familiar (pf - pressuposto pai, m - mãe e c - criança).

As frequências alélicas encontram-se definidas no R e tendo em conta a natureza destes dados, as suas estimativas podem sofrer alterações nas suas estimativas ou podemos até querer utilizar estimativas de outras populações e como tal, os valores destas estimativas podem necessitar de ser alterados.

As estimativas das frequências alélicas utilizadas nesta dissertação foram obtidas em Vieira-Silva, T. Ribeiro e Espinheira 2006.

5.2 O *package* gRain

O *package* **gRain** (*gRaphical independence network*) (Hørjsgaard 2012) é um *package* do R desenhado para construir RB e propagar novas evidências através do algoritmo de propagação descrito em Lauritzen e Spiegelhalter (1988). As RB no **gRain** são restritas a variáveis discretas, cada uma com um conjunto finito de estados possíveis.

Neste *package* cada distribuição de probabilidade condicional satisfaz a condição de Markov e após criadas as tabelas de probabilidade, o **gRain** verifica se estas podem construir um DAG. Assim que o DAG fica definido, podemos consultar as tabelas ou inserir nova evidência na rede, que após ser propagada irá atualizar as tabelas de probabilidade do DAG.

Para melhor entender a utilidade do *package* **gRain**, recordemos a RB de um genótipo.

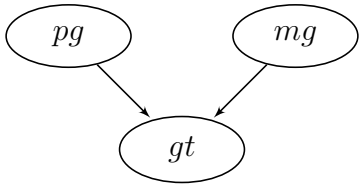


Figura 5.6: Rede bayesiana de um genótipo.

A principal utilidade do **gRain** no programa criado nesta dissertação é a construção da RB e a propagação da informação observada na rede. Computacionalmente, este processo pode ser resumido da seguinte forma (Højsgaard 2015):

1. Especificamos as tabelas de probabilidades condicionais:

```
pg<-cptable(~pg, values=allel_p, levels=p_levels)
mg<-cptable(~mg, values=allel_m, levels=m_levels)
gt<-cptable(~gt|pg:mg, values=c.prob_gt,levels=gt_lvls)
```

2. Compilamos as tabelas de probabilidades definidas em 1. numa lista:

```
plist<-compileCPT(list(pg,mg,gt))
```

3. Criamos a RB com base na lista criada em 2.:

```
net<-grain(plist)
```

4. Inserimos e propagamos na RB criada em 3. a evidência observada:

```
bnetE<-setEvidence(net,nodes=c("gt","mg","pg"),states=s)
```

5. Após propagada a nova evidência, podemos obter as novas probabilidades atualizadas na RB:

```
querygrain(bnetE)
```

Toda a informação que está nas tabelas (*values*, *levels* e *states*), assim como a estrutura de dependências da RB, não está definida no **gRain** e teve de ser definida no programa criado em R. Para um dado caso de disputa de paternidade o programa criado redefine toda a informação inserida nas tabelas com base na informação obtida de cada um dos intervenientes.

Para melhor apreciar a complexidade de todo este processo consideremos, a título de exemplo, a construção no R de um segmento de uma RB, como o da Figura 5.7, para um dado marcador genético assumindo que para cada marcador genético, a informação sobre os alelos observados dos intervenientes e respetivas frequências alélicas está representada num *dataframe* semelhante ao da Figura 5.5.

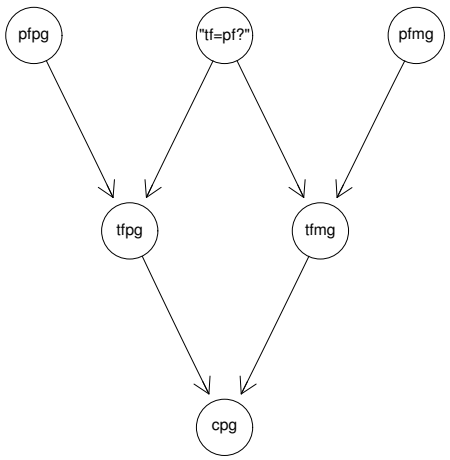


Figura 5.7: Segmento de uma RB de um caso de paternidade disputada.

Inicialmente é criada uma forma de atribuir uma variável identificadora se marcadores homozigóticos. Caso os alelos identificados num marcador sejam iguais, a variável identificadora toma o valor 1, caso contrário fica com o valor 0. Este passo é importante porque, para criar os *values* de tabelas como as Tabelas 4.4 e 4.5 por exemplo, é necessário ter em conta as formas possíveis da informação passar dos pais para a criança.

```
pH<-0;mH<-0;
if(identical(x[i,2],x[i+1,2])==TRUE)pH<-1;
if(identical(x[i,4],x[i+1,4])==TRUE)mH<-1;
```

O campo *levels* é simples de criar já que este corresponde, para um dado marcador, aos alelos presentes na população. Cada marcador tem vários alelos possíveis mas cingimos essa informação aos alelos observados e ao conjunto agrupado dos restantes alelos representado por "x" no R da seguinte forma.

```
pf_levels<-as.character(unique(c(x[i,2],x[i+1,2],"x")));
pfgt_levels<-unique(c(paste(x[i,2],x[i,2]),
    paste(x[i,2],x[i+1,2]),
            paste(x[i,2],"x"),paste(x[i+1,2],x[i+1,2]),
                paste(x[i+1,2],"x"),paste("x","x")));
cpg_levels<-unique(c(x[i,2],x[i+1,2]));
```

Para criar os *values* da informação herdada pelos seus pais, usamos simplesmente as frequências alélicas encontradas na população.

```
allel_pf<-unique(c(x[i,3],x[i+1,3],
    1-sum(unique(x[i:(i+1),3]))));
```

Já no caso da informação que o pressuposto pai passa para a criança é necessário criar todas as possibilidades possíveis em como os alelos podem combinar e como isso afecta a probabilidade de certo alelo ser herdado pela criança c do pressuposto pai *versus* um outro indivíduo aleatório.

```
if(pH+mH==2){
        cond.prob.pfgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
        cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,1,
                unique(x[i:(i+1),3]),
                    1-sum(unique(x[i:(i+1),3])),
                    unique(x[i:(i+1),3]),
                        1-sum(unique(x[i:(i+1),3])));
        cond.prob.cpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
        if(x[i,2]>x[i,4]){
                cond.prob.cgt<-c(1,0,0,0,0,1,0,0,0,
                    0,1,0,0,0,0,1);
        } else ifelse(equal==1,
            cond.prob.cgt<-c(1,0,0,0,1,0,0,1,0,0,0,1),
                cond.prob.cgt<-c(1,0,0,0,0,0,1,0,
                    0,1,0,0,0,0,0,1));
}    ...
```

O programa criado em R foi criado de forma a acomodar toda esta informação e após a inserção dos dados observados pelo utilizador, o programa irá automaticamente criar todos os campos necessários para o **gRain** criar a RB e propagar a evidência observada na rede tal como descrito anteriormente nos passos 1 a 5.

Com a ajuda do **gRain**, o programa construido cria uma rede para cada marcador à vez e para cada um destes marcadores atualiza os estados (e as respetivas probabilidades) das variáveis correspondentes à informação dos alelos observados e

propaga essa informação para toda a rede. Para cada marcador é calculado o LR do pressuposto pai ser o pai biológico e depois de obtidos todos os valores de LR é calculado um valor de LR global. Apenas utilizando o **gRain** não seria possível fazer estes cálculos.

5.3 Caso I: Trio familiar

O primeiro caso estudado resulta de uma disputa de paternidade comum, em que a paternidade de um individuo é posta em causa e temos como intervenientes no caso um pressuposto pai pf, uma mãe m e uma criança c. Neste caso temos disponíveis amostras de DNA de cada um dos intervenientes. Nas Figuras 5.8 e 5.9 estão representados o *pedigree* familiar e a correspondente RB.

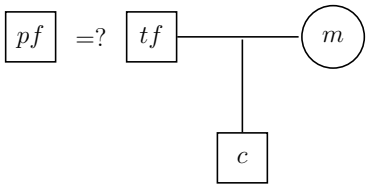


Figura 5.8: *Pedigree* familiar do Caso I onde c é a criança, pf é pressuposto pai, tf é o pai biológico, m a mãe da criança.

Foi importante começar por estabelecer um procedimento para construir RB no R para um caso sobre um trio familiar já que estes casos são extremamente comuns e constituem a base da grande maioria dos casos de paternidade. A partir desta rede, é relativamente fácil manipular e adaptar o código em R para criar uma RB conveniente para outros casos com intervenientes diferentes.

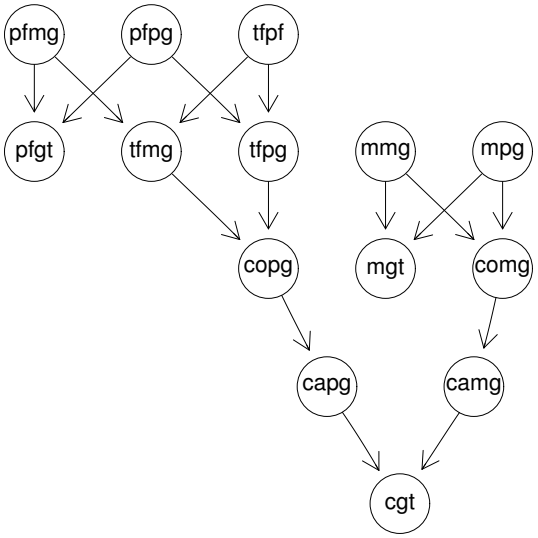


Figura 5.9: Rede bayesiana do Caso I.

No R, cada marcador é analisado de forma independente, à vez. Nesta fase, são criadas as tabelas de probabilidade dos nodos do DAG, mediante as possíveis combinações alélicas de cada marcador. Posteriormente a evidência observada é propagada na RB e por por fim, é calculado o respetivo valor de LR.

Assumindo independência entre os marcadores, o valor global de LR a favor da paternidade é dado pelo produto de todos os termos da última coluna da Tabela 5.1.

Tabela 5.1: Dados observados para o Caso I e respetivos LR.

Marcador	pf		m		c		LR
CSF1PO	8	10	10	10	10	10	1.874254
D2S1338	19	20	17	20	19	20	4.738196
D3S1358	15	16	15	19	15	16	1.900867
D5S818	11	13	11	11	11	11	1.492029
D7S820	7	10	11	12	7	12	19.618360
D8S1179	14	16	11	13	13	16	17.511142
D13S317	11	12	8	11	8	11	1.534885
D16S539	11	12	11	12	11	12	1.735688
D18S51	14	15	12	13	13	15	3.466782
D19S433	12	12	12	14	12	12	8.939042
D21S11	33.2	35.1	33.2	33.2	33.2	35.1	23.054759
FGA	19	25	22	28	19	22	7.244706
Penta D	10	12	9	14	9	10	4.348587
Penta E	5	13	15	21	5	15	7.634542
TH01	6	7	9	9.3	7	9	2.670630
TPOX	9	11	10	11	10	11	1.736193
VWA	16	18	15	18	16	18	2.211401
Global							40 616 703 591

O valor de LR global indica que é quarenta biliões de vezes mais provável o pressuposto pai ser o pai biológico da criança do que o pai biológico ser um individuo aleatório da população.

O valor elevado do LR global não é surpreendente já que todos os marcadores genéticos analisados do pressuposto pai têm pelo menos um alelo concordante com a informação genética da criança.

Este caso tem ainda uma particularidade interessante. Um dos alelos observados é desconhecido na população do marcador genético D21S11 (o alelo 35.1). Quando isto acontece é necessário atribuir uma frequência mínima que contemple estas situações.

Esta contingência foi tida em conta e incluída no programa criado. As frequências mínimas consideradas para cada marcador foram obtidas através das estimativas obtidas em Vieira-Silva, T. Ribeiro e Espinheira 2006.

Em seguida foi comparada a informação obtida no R com o *software* de referência, o **familias**. São comparados os valores de LR entre estes dois *softwares* a fim de determinar a eficácia do programa criado em R. Este procedimento foi transversal a todos os casos estudados nesta dissertação.

Tabela 5.2: LR obtidos no R para o Caso I comparados com os obtidos no *software* **familias**.

Marcador	LR no R	LR no **familias**
CSF1PO	1.874254	1.881821
D2S1338	4.738196	4.784688
D3S1358	1.900867	1.902587
D5S818	1.492029	1.492982
D7S820	19.618360	24.509803
D8S1179	17.511142	19.157088
D13S317	1.534885	1.087686
D16S539	1.735688	1.737619
D18S51	3.466782	3.475711
D19S433	8.939042	9.057971
D21S11	23.054759	1802.673772
FGA	7.244706	7.396449
Penta D	4.348587	4.370629
Penta E	7.634542	7.716049
TH01	2.670630	2.672367
TPOX	1.736193	1.418037
VWA	2.211401	2.217294
Global	40 616 703 591	2 699 732 329 859.79

Os valores de LR entre os dois *softwares* foram bastante consistentes. Na sua maioria encontram-se na mesma ordem de grandeza o que revela que o programa criado é consistente. A exceção foi o marcador D21S11 que apresenta um valor bastante maior no **familias**.

Naturalmente o alelo que estava ausente das estimativas populacionais do marcador genético D21S11 apresenta um valor de LR bastante elevado em ambos os casos, já que a criança apresenta o mesmo alelo que o pai. No **familias** esta informação tem de ser inserida *a posteriori* para lidar com um caso destes. No programa criado em R esta situação já está controlada e a frequência mínima é imediatamente atribuída caso surja num perfil genético um alelo que não está representado na população.

5.4 Caso II: Avós paternos, mãe e criança

Este é um caso mais complexo onde o pressuposto pai (pf) está ausente e temos de recorrer a seus familiares (neste caso o seu pai e a sua mãe) para mapear a sua

informação genética de forma a inferir sobre a probabilidade de ser o pai biológico (tf) da criança.

Neste caso temos disponíveis amostras de DNA da criança (c), da mãe da criança (m) e dos pressupostos avós paternos da criança (gf e gm).

Na Figura 5.10 está representado o *pedigree* familiar e na Figura 5.11 a correspondente RB.

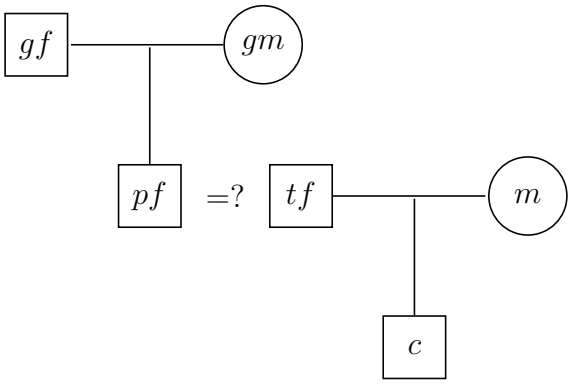


Figura 5.10: *Pedigree* familiar do Caso II onde c é a criança, pf é pressuposto pai, tf é o pai biológico, m a mãe da criança, gf o pai do pressuposto pai e gm a mãe do pressuposto pai.

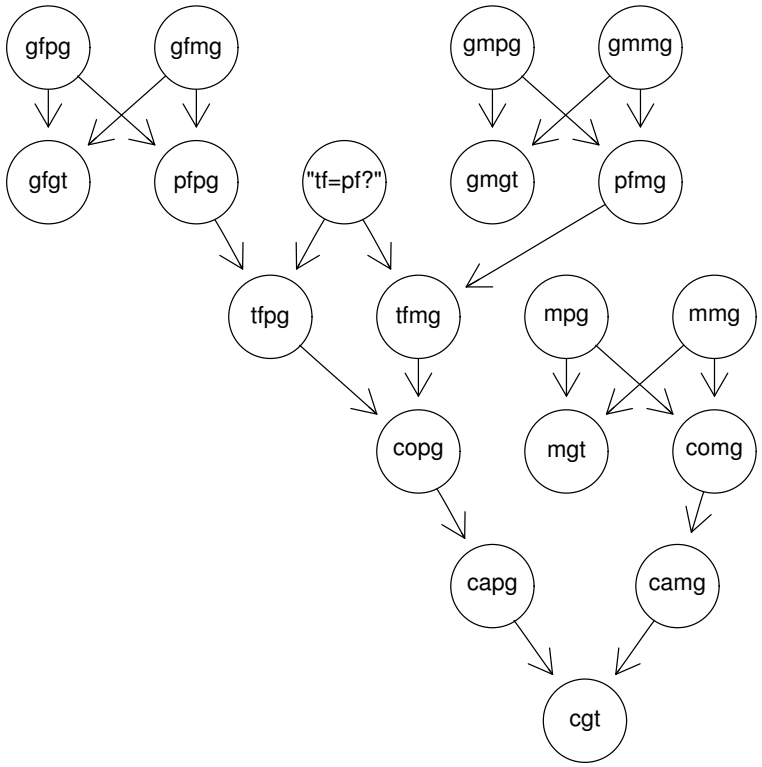


Figura 5.11: Rede bayesiana do Caso II.

Tendo esta rede como base, no R criamos uma rede com uma estrutura igual e propagamos os valores dos alelos observados, nos respetivos nodos, para cada um dos marcadores em análise de forma a obter uma probabilidade atualizada sobre a probabilidade a favor da paternidade do pressuposto pai da criança.

Mais uma vez, assumimos independência entre os marcadores e obtemos o valor global de LR a favor da paternidade multiplicando os LR obtidos para cada um dos marcadores. Estes valores, assim como o LR global, aparecem discriminados na última coluna da Tabela 5.3.

Tabela 5.3: Dados observados para o Caso II e respetivos LR.

Marcador	gf		gm		m		c		LR
CSF1PO	11	12	11	12	11	12	12	12	1.5541139
D2S1338	18	21	17	24	16	20	16	24	2.3247747
D3S1358	16	17	16	17	14	15	15	16	1.9004006
D5S818	12	14	11	11	12	12	11	12	1.4917000
D7S820	10	12	10	12	7	12	7	12	3.1918639
D8S1179	12	14	13	15	12	14	12	12	2.0132394
D13S317	8	9	8	9	8	11	9	11	8.4352459
D16S539	9	11	8	9	12	13	9	12	3.6706654
D18S51	13	17	14	16	15	16	15	16	1.4994183
D19S433	15	15.2	14	15	12	14	12	14	0.7579286
D21S11	30	31.2	31.2	31.2	28	29	29	31.2	4.5849629
FGA	22	23	20	25	19	25	25	25	3.0063442
Penta D	13	15	8	12	10	11	10	13	1.2356423
Penta E	7	12	5	11	13	14	5	14	0.8045988
TH01	6	9	8	9.3	6	9.3	8	9.3	1.7250685
TPOX	9	10	8	9	8	8	8	9	4.6833517
VWA	18	19	16	18	15	17	15	16	1.0396085
Global									266 565.20

Com um valor de LR tão elevado como o observado, $LRGlobal = 266565.20$, existem fortes evidências que suportam a paternidade do pressuposto pai (pf). Por outras palavras, é cerca de duzentos e sessenta e seis mil vezes mais provável observarmos estes perfis de DNA se o pressuposto pai for o pai biológico da criança do que o pai biológico da criança ser um individuo aleatório da população.

Ao observar os LR obtidos para cada marcador podemos ainda constatar que nenhum marcador foi claramente contra a paternidade do pressuposto pai. Apenas dois dos marcadores genéticos analisados apresentam um LR inferior a 1 (D19S433 e Penta E).

Ao comparar estes resultados com os obtidos no *software* **familias** verificamos que a grande maioria dos marcadores têm LR na mesma ordem de grandeza, embora o LR global obtido pelo **familias** seja bastante superior. A Tabela 5.4 tem estes valores.

Tabela 5.4: LR obtidos no R para o Caso II comparados com os obtidos no *software* **familias**.

Marcador	LR no **familias**	LR no R
CSF1PO	1.5547264	1.5541139
D2S1338	2.4015370	2.3247747
D3S1358	1.9025875	1.9004006
D5S818	1.2464915	1.4917000
D7S820	1.4835610	3.1918639
D8S1179	2.0226537	2.0132394
D13S317	8.9285714	8.4352459
D16S539	3.6900369	3.6706654
D18S51	0.8364001	1.4994183
D19S433	0.5910165	0.7579286
D21S11	6.8997240	4.5849629
FGA	3.0599755	3.0063442
Penta D	1.3068479	1.2356423
Penta E	3.8580247	0.8045988
TH01	1.7253278	1.7250685
TPOX	4.6948357	4.6833517
VWA	1.1086475	1.0396085
Global	946 109.04	266 565.20

Na maioria dos marcadores genéticos obtemos valores de LR bastante semelhantes em ambos os *softwares* e também se verificou que ambos os *softwares* têm somente dois marcadores genéticos com valor de LR abaixo de um (no **familias** D18S51 e D19S433 e no R Penta E e D19S433).

Os valores de LR mais distintos entre os dois *softwares* correspondem aos marcadores genéticos D18S51 e Penta E.

Apesar destas diferenças o valor global do LR é bastante elevado e a favor da paternidade do pressuposto pai em ambos os *softwares*.

5.5 Caso III: Avó paterna e criança

O último caso analisado nesta dissertação tem como intervenientes uma criança e a mãe do pressuposto pai. Este caso é difícil de analisar visto haver muito pouca informação disponível para determinar a paternidade do pressuposto pai. A dificuldade na análise deste caso deve-se ao facto de apenas termos disponíveis a informação genética da criança e da mãe do pressuposto pai, o que cria logo à partida um *pedigree* familiar bastante incompleto.

O *pedigree* em causa está na Figura 5.12 e a correspondente RB está na figura 5.13.

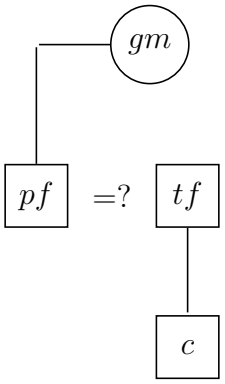


Figura 5.12: *Pedigree* familiar do Caso III onde c é a criança, pf é pressuposto pai, tf é o pai biológico e gm a mãe do pressuposto pai.

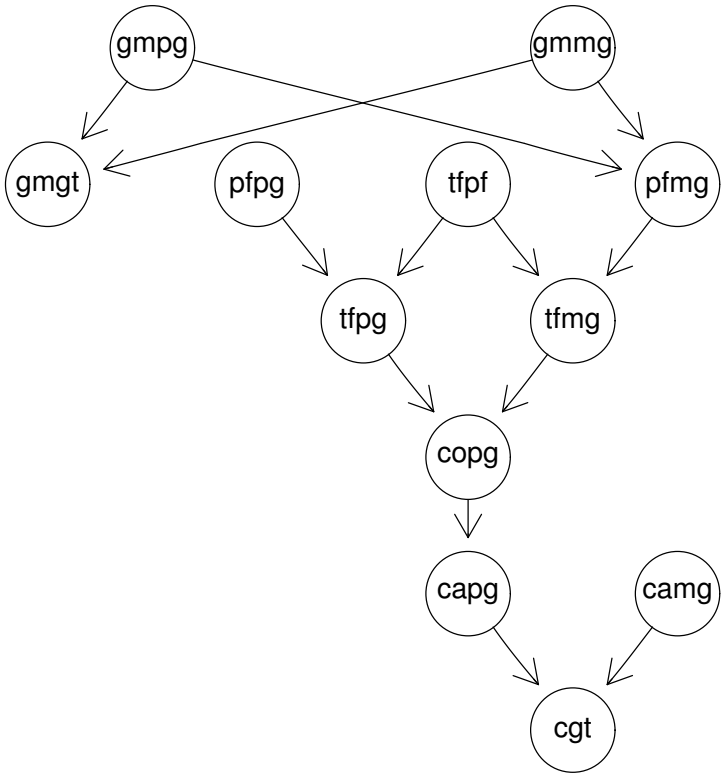


Figura 5.13: Rede bayesiana do Caso III.

Desta forma, como temos apenas informação sobre os genótipos de c e gm, a inferência feita sobre a informação genética materna do pf é feita com base na informação proveniente de gm e a inferência efetuada sobre informação paterna do pf tem como base as frequências alélicas da população.

Duma forma semelhante, toda a inferência sobre a informação genética materna da criança c tem por base as frequências alélicas da população já que não temos acesso à informação genética da mãe da criança.

Como usualmente, assumimos a independência entre marcadores genéticos e o valor global de LR a favor da paternidade é calculado multiplicando os valores de LR de cada um dos marcadores genéticos. Estes valores de LR estão representados na última coluna da Tabela 5.5.

Tabela 5.5: Dados observados para o Caso III e respetivos LR.

Marcador	gm		c		LR
CSF1PO	12	12	10	10	0.7591745
D2S1338	16	19	17	20	0.8674145
D3S1358	15	15	14	17	0.5027123
D5S818	10	11	11	11	1.2461234
D7S820	10	11	10	11	1.4825894
D8S1179	14	14	12	13	0.5016240
D13S317	12	13	11	12	1.2619033
D16S539	10	10	9	11	0.5044771
D18S51	16	19	12	16	1.7303313
D19S433	12	15	12	13	2.1478285
D21S11	32.2	34.2	28	32.2	2.5563511
FGA	23	27	26	27	9.9403555
Penta D	11	11	11	14	2.7057597
Penta E	7	10	12	12	0.5010481
TH01	7	9.3	6	7	1.5596439
TPOX	8	8	8	11	1.3421899
VWA	16	17	15	19	0.6847653
Global					35.84402

O valor de LR global diz-nos que é aproximadamente trinta e seis vezes mais provável o pressuposto pai, pf, ser o pai biológico da criança do que um individuo aleatório da população ser o pai da criança.

Apesar disto este valor é insuficiente, de um ponto de vista jurídico, para declarar pf como o pai biológico da criança. Para tal, seria preciso obter um valor de LR superior a mil.

Neste caso podemos visualizar uma das vertentes implementadas no programa criado que permite modificar a probabilidade inicial atribuída à paternidade do pressuposto pai.

De realçar que, não cabe ao investigador forense persuadir um júri de que o pressuposto pai dum dado caso é de facto o pai biológico de uma criança, mas se existem indicações fortes de que à partida o pressuposto pai é o biológico pai (ou no caso contrário), a probabilidade a favor da paternidade pode ser dada na forma de uma tabela com várias probabilidades *a priori* e respetivas probabilidades *a posteriori* para o júri ter em consideração o impacto da evidência obtida.

Na Tabela 5.6 vemos a evolução do valor de LR conforme a probabilidade atribuída a hipótese H_0: O pai biológico é o pressuposto pai.

Tabela 5.6: LR obtidos no R para o Caso III conforme a $P(H_0)$.

Probabilidade de H_0	LR
0.50	35.84402
0.55	1 086.35
0.75	4 628 902 623

É notório que uma ligeira mudança na probabilidade de H_0 muda radicalmente o valor de LR. Neste caso bastava que $P(H_0) = 0.55$ para que o valor de LR ultrapassasse o valor de referência (1000) e a paternidade do pressuposto pai já seria plausível.

Tabela 5.7: LR obtidos no R para o Caso III comparados com os obtidos no *software* **familias**.

Marcador	LR no R	LR no **familias**
CSF1PO	0.5	0.7591745
D2S1338	0.5	0.8674145
D3S1358	0.5	0.5027123
D5S818	1.2464915	1.2461234
D7S820	1.4835610	1.4825894
D8S1179	0.5	0.5016240
D13S317	0.9475474	1.2619033
D16S539	0.5	0.5044771
D18S51	1.3059316	1.7303313
D19S433	1.6322464	2.1478285
D21S11	1.9156285	2.5563511
FGA	17.623288	9.9403555
Penta D	2.0470297	2.7057597
Penta E	0.5	0.5010481
TH01	1.1680919	1.5596439
TPOX	1.0075112	1.3421899
VWA	0.5	0.6847653
Global	2.37324	35.84402

Neste caso temos muitos valores de LR abaixo de um o que explica o valor baixo do LR. Na maioria dos marcadores genéticos obtemos valores de LR na mesma ordem de grandeza em ambos os *softwares*. Existem vários marcadores genéticos discordantes com a paternidade nos dois *softwares* mas tendo em conta o quão incompleta está a informação disponível para este caso acaba por ser um resultado esperado.

Capítulo 6

Discussão e conclusão dos resultados

6.1 Discussão dos Resultados

Tendo em conta os casos estudados os programas produziram resultados positivos, obtendo valores de LR semelhantes aos valores de LR de referência do *software* **familias**.

Para o primeiro caso em estudo o programa criado correspondeu às expectativas. Os valores de LR gerados através do método aplicado nesta dissertação foram bastante semelhantes aos valores de LR de referência do *software* **familias**. Em ambos, o valor global de LR é bastante elevado e tendo em conta que todos os marcadores genéticos são concordantes a paternidade do pressuposto pai é bastante verosímil.

No Caso II o programa criado gerou valores de LR bastante semelhantes entre o R e o **familias** na maioria dos marcadores genéticos. A maior parte dos valores distintos estão na mesma ordem de grandeza e apenas dois marcadores genéticos apresentaram valores de grandeza diferente. Estas diferenças não comprometeram o valor de LR global que se apresentou bastante elevado em ambos os casos. Nenhum marcador genético apresentou discordância com a paternidade, ou seja, todos os marcadores genéticos que compunham o perfil genético do pressuposto pai tinham no mínimo um dos alelos presentes no perfil da criança.

Quanto ao Caso III os valores de LR gerados foram bastante distintos sempre que a informação não era concordante. O *software* **familias** assume sempre um LR de 0.5 a favor da paternidade enquanto que no programa criado em R e tendo em conta a possibilidade de mutação os valores de LR são geralmente superiores quando os marcadores são discordantes. Este facto tem impacto no valor global do LR. Foi também apresentada a possibilidade de alterar a probabilidade inicial a favor da paternidade e suas implicações no desfecho deste caso.

6.2 Conclusão

O foco desta dissertação foi criar programas em R que permitam investigadores forenses obter rápida e facilmente valores de LR para casos de paternidade disputada através do uso de Redes Bayesianas.

A criação de RB no software R foi possível através do *package* **gRain** que tem delineado nas suas funções formas de construir uma RB e propagar informação na rede após a inserção de nova evidência. Mediante os casos estudados, toda a estrutura da rede, assim como os possíveis estados das diferentes ligações entre os alelos dos intervenientes, tiveram de ser definidas nos programas criados. Foi ainda definido um modelo de mutação para cada um dos casos assim como a possibilidade do aparecimento de alelos não contidos na população.

Foram tidos em conta três casos, o caso mais comum que envolve um trio familiar composto por uma mãe, uma criança e um pressuposto pai e dois casos mais complexos, ambos sem o pressuposto pai presente, em que um tinha como intervenientes os avós paternos, a mãe da criança e a criança e no último apenas a avó paterna e a criança estavam presentes.

Os valores de LR obtidos com o novo programa criado em R foram, regra geral, coerentes com os valores de referência. O programa criado mostrou-se eficaz em resolver os tipos de casos com os *pedigrees* familiares estudados (Caso I, Caso II e o Caso III). Mostrou ainda ser capaz de resolver melhor algumas situações que o **familias**, nomeadamente a possibilidade de ocorrer mutação e o aparecimento de alelos que não estão na população estudada.

Todos os casos estudados apresentaram valores globais de LR concordantes com os observados no *software* **familias**.

6.3 Sugestões Futuras

Todo o trabalho efetuado nesta dissertação pode ser aprofundado no sentido de criar uma biblioteca no R que permita a um utilizador forense obter um valor de LR para qualquer caso de paternidade disputada. Nesse sentido ficam aqui algumas sugestões para trabalho futuro no âmbito do trabalho já criado nesta dissertação.

A inclusão de um método para tratar casos com alelos nulos (*silent alleles* em inglês). Estes casos acontecem quando um alelo não é capturado no equipamento que faz o perfil de DNA. Quando isto acontece, um genótipo que aparenta ser homozigótico num dado marcador pode não o ser na realidade. Uma explicação alternativa pode ser que visualizamos apenas uma banda de um genótipo heterozigótico, sendo a outra nula. Este fenómeno pode influenciar a interpretação da evidência de determinados perfis de DNA.

O trabalho feito nesta dissertação abordou alguns casos específicos que, embora comuns, não representam todos os casos possíveis de paternidade disputada. Nesse sentido, o trabalho efetuado cria uma *baseline* para criar programas em R para variados casos.

Indo mais longe ainda, e tendo em conta a natureza das RB, idealmente seria criado um programa adequado a um caso particular onde é tirado partido da forma estrutural como as RB são construídas e a única informação necessária seria a introdução dos intervenientes do caso em estudo. A partir daí o programa criaria a RB com as ligações adequadas para calcular o valor de LR de um caso em estudo.

Referências Bibliográficas

Biedermann, A. e F. Taroni (2012). «Bayesian Networks for Evaluating Forensic DNA Profiling Evidence: A Review and Guide to Literature». Em: *Forensic Science International: Genetics* 6, pp. 147–157.

Butler, J.M. e D.J. Reeder. *Short Tandem Repeat DNA Internet DataBase*. `http://www.cstl.nist.gov/strbase/mutation.htm`.

Chadreque, Manuela (2012). «Estatística na Investigação Forense». Tese de mestrado. Universidade de Lisboa.

Cooper, Gregory F. (1990). «The Computation Complexity of Probabilistic Inference Using Bayesian Belief Networks». Em: *Artificial Intelligence* 42, p. 395.

Dauber, E.M. et al. (2003). «Mutation rates at 23 different short tandem repeat loci». Em: *International Congress Series* 1239, pp. 565–567.

Dawid, A.P., J. Mortera e V.L. Pascali (2001). «Non-fatherhood or mutation? A probabilistic approach to parental exclusion in paternity testing». Em: *Forensic Science International* 124.1, pp. 55–61.

Dawid, A.P., J. Mortera e P. Vicard (2007). «Object-oriented Bayesian networks for complex forensic DNA profiling problems». Em: *Forensic Science International* 169, pp. 195–205.

Dawid, A.P. et al. (2002). «Probabilistic Expert Systems for Forensic Inference from Genetic Markers». Em: *Board of the Foundation of the Scandinavian Journal of Statistics* 29, pp. 577–595.

Eis, D. e V. Kostina (2009). *Foundations of Probabilistic Modeling*. `http://www.cs.princeton.edu/courses/archive/spr09/cos513/scribe/lecture05.pdf`.

Fenton, N. e M. Neil (2000). «The jury observation fallacy and the use of Bayesian Networks to present probabilistic legal arguments». Em: *Mathematics Today* 6.36, pp. 180–187.

— (2012). *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press. ISBN: 978-1-4398-0910-5.

Højsgaard, Søren (2015). *Bayesian networks in R with the gRain package*. Rel. téc. Aalborg University.

Hørjsgaard, Søren (2012). «Graphical Independence Networks with the gRain Package for R». Em: *Journal of Statistical Software* 46.10, pp. 1–26.

Jensen, Finn V. (2001). *Bayesian Networks and Decision Graphs*. Springer. ISBN: 0-387-95259-4.

Kobilinsky, Lawrence, Thomas F. Liotti e Jamel Oeser-Sweat (2005). *DNA: Forensic and Legal Applications*. 1ª ed. New Jersey: John Wiley & Sons. ISBN: 0-471-41478-6.

Lauritzen, S.L. e D. Spiegelhalter (1988). «Computations with Probabilities on Graphical Structures and their Application to Expert Systems». Em: *Journal of the Royal Statistical Society B* 50.2, pp. 157–224.

Lucy, David (2005). *Introduction to Statistics for Forensic Scientists*. John Wiley & Sons, Ltd.

Mello, F. Galvão de (2000). *Probabilidades e Estatística - conceitos e métodos fundamentais*. 2ª ed. Vol. 1. Escolar Editora. ISBN: 972-592-110-0.

Murteira, B. e M. Antunes (2012a). *Probabilidades e Estatística*. Escolar Editora. ISBN: 978-972-592-359-7.

— (2012b). *Probabilidades e Estatística*. Escolar Editora. ISBN: 978-972-592-355-9.

Pearl, Judea (2000). «Graphs and probabilities». Em: *Casuality*. 1ª ed. Cambridge University Press. Cap. 1, pp. 25–38.

Pestana, D. e S. Velosa (2010). *Introdução à Probabilidade e à Estatística*. 4ª ed. Vol. 1. Fundação Calouse Gulbenkian. ISBN: 978-972-31-1150-7.

Rocha, J.C.F., A.M. Guimarães e C.P. Campos (2011). «Integração de Evidências em Redes Credais e a Regra de Jeffrey». Em: *Revista de Informática Teórica e Aplicada* 18.2, pp. 251–265.

Silva, Claúdia Isabel Vieira da (2011). «Métodos Estatísticos para a análise de Y-STRs em Genética Forense». Tese de mestrado. Faculdade de Ciências da Universidade de Lisboa.

Taroni, F. et al. (2006). *Bayesian Networks and Probabilistic Inference in Forensic Science*. Jonh Wiley & Sons, Ltd. ISBN: 0-470-09173-8.

Taroni, F. et al. (2010). *Data analysis in forensic science - A Bayesian decision perspective*. 1ª ed. Wiley. ISBN: 978-0-470-99835-9.

Thein, S.W., A.J. Jeffreys e V. Wilson (1985). «Bayesian Networks for Evaluating Forensic DNA Profiling Evidence: A Review and Guide to Literature». Em: *Nature* 314, pp. 67–73.

Vieira-Silva, C., C. Cruz amd T. Ribeiro e R. Espinheira (2006). «South Portugal population genetic analysis with 17 loci STR». Em: *International Congress Series* 1288, pp. 367–368.

Wamelen, J. J. van (2011). «Bayesian Networks in Forensic DNA Analysis». Tese de mestrado. Universiteit Leiden.

Anexos

Instalação do *package* gRain

O *package* **gRain** usa uma combinação de outros *packages* já descontinuados e portanto é necessário instalar estes antes de poder correr o **gRain**.

Para tal é necessário executar o seguinte código no R:

```
source("http://bioconductor.org/biocLite.R");
biocLite(c("graph","RBGL","Rgraphviz"));
```

seguido de:

```
install.packages("gRbase", dependencies=TRUE);
install.packages("gRain", dependencies=TRUE);
install.packages("gRim", dependencies=TRUE)
```

Após a execução destes passos basta instalar o *package* **gRain** da forma usual.

Código de R para o Caso I

```
#Executar a biblioteca gRain

library(gRain)

##criar ficheiro com as frequencias alelicas para execucoes futuras

#lista de marcadores geneticos analisados
marker<-c("CSF1PO","CSF1PO","D2S1338","D2S1338","D3S1358","D3S1358","D5S818","D5S818",
"D7S820","D7S820","D8S1179","D8S1179","D13S317","D13S317","D16S539","D16S539",
"D18S51","D18S51","D19S433","D19S433","D21S11","D21S11","FGA","FGA","PentaD","PentaD",
"PentaE","PentaE","TH01","TH01","TPOX","TPOX","vWA","vWA")

#criacao da matrix de marcadores alelicos e respetivas frequencias
y<-matrix(0,nrow=28,ncol=34);
colnames(y)<-marker;

#insercao dos alelos de cada um dos marcadores
#CSF1PO
y[,1]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
```

```
#D2S1338
y[,3]<-c(13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D3S1358
y[,5]<-c(9,10,11,12,13,14,15,16,17,18,19,20,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D5S818
y[,7]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D7S820
y[,9]<-c(6,7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D8S1179
y[,11]<-c(8,9,10,11,12,13,14,15,16,17,18,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D13S317
y[,13]<-c(8,9,10,11,12,13,14,15,16,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D16S539
y[,15]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D18S51
y[,17]<-c(10,10.2,11,12,13,13.2,14,14.2,15,15.2,16,17,18,19,20,21,22,23,24,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D19S433
y[,19]<-c(10,11,11.2,12,12.2,13,13.2,14,14.2,15,15.2,16,16.2,17,17.2,18,18.2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D21S11
y[,21]<-c(21,24,24.2,25,25.2,26,26.2,27,28,28.2,29,29.2,30,30.2,31,31.2,32,32.2,33,33.2,34,34.1,34.2,35,35.2,36,37,-2)
#FGA
y[,23]<-c(16,17,18,18.2,19,20,20.2,21,21.2,22,22.2,23,23.2,24,24.2,25,25.2,26,27,27.2,28,29,30.2,31.2,43.2,-1,-1,-2)
#PentaD
y[,25]<-c(2.2,3.2,5,6,7,8,9,10,11,12,13,14,15,16,17,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#PentaE
y[,27]<-c(5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,-1,-1,-1,-1,-1,-1,-2)
#TH01
y[,29]<-c(3,4,5,6,7,8,9,9.3,10,11,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#TPOX
y[,31]<-c(6,7,8,9,10,11,12,13,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#vWa
y[,33]<-c(11,13,14,15,16,17,18,19,20,21,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)

#insercao das frequencias de cada um dos alelos de
#cada um dos marcadores e respetivas frequencias minimas
#CSF1PO
y[,2]<-c(0.0064,0.0114,0.0185,0.2657,0.3155,0.3215,0.05235,0.0079,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0010)
#D2S1338
y[,4]<-c(0.0004,0.0007,0.0015,0.0501,0.2415,0.0723,0.1045,0.1418,0.0468,0.0409,
0.1034,0.1041,0.0779,0.0127,0.0011,0.0004,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D3S1358
y[,6]<-c(0.0004,0.0002,0.0006,0.0039,0.0057,0.1036,0.2589,0.2629,0.2007,0.1491,
0.0134,0.0007,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D5S818
y[,8]<-c(0.0004,0.0145,0.0376,0.0617,0.3349,0.3676,0.1724,0.0101,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D7S820
y[,10]<-c(0.0002,0.0204,0.1478,0.1183,0.2896,0.2266,0.1561,0.0353,0.0053,0.0006,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D8S1179
y[,12]<-c(0.0095,0.0101,0.0762,0.0846,0.1236,0.2901,0.2525,0.1230,0.0261,0.0035,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D13S317
y[,14]<-c(0.1340,0.0560,0.0485,0.3256,0.2793,0.1144,0.0406,0.0015,0.0002,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D16S539
y[,16]<-c(0.0002,0.0211,0.1355,0.0602,0.3010,0.2745,0.1776,0.0292,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D18S51
y[,18]<-c(0.0145,0.0006,0.0083,0.1328,0.1269,0.0009,0.1381,0.0002,0.1428,0.0002,0.1553,
0.1272,0.0709,0.0426,0.0220,0.0112,0.0024,0.0017,0.0006,-1,-1,-1,-1,-1,-1,-1,-1,0.0012)
#D19S433
y[,20]<-c(0.0029,0.0112,0.0002,0.1104,0.0050,0.2534,0.0132,0.3127,0.0307,0.1460,0.0455,
0.0466,0.0152,0.0031,0.0017,0.0006,0.0017,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D21S11
y[,22]<-c(0.0002,0.0002,0.0024,0.0004,0.0009,0.0013,0.0004,0.0257,0.1577,0.0002,0.2264,0.0007,
0.2402,0.0354,0.0610,0.1087,0.0088,0.0883,0.0015,0.0305,0.0006,0.0002,0.0050,0.0024,0.0007,0.0002,0.0002,0.0011)
#FGA
y[,24]<-c(0.0004,0.0026,0.0156,0.0011,0.0676,0.1267,0.0002,0.1759,0.0018,0.1864,0.0059,0.1484,
0.0022,0.1357,0.0004,0.0817,0.0004,0.0347,0.0073,0.0002,0.0028,0.0013,0.0002,0.0006,0.0002,-1,-1,0.0011)
#PentaD
y[,26]<-c(0.0187,0.0015,0.0033,0.0007,0.0095,0.0206,0.1906,0.1144,0.1616,0.1855,0.1913,
0.0747,0.0202,0.0062,0.0011,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#PentaE
y[,28]<-c(0.0648,0.0013,0.1399,0.0288,0.0134,0.0854,0.1254,0.1968,0.1160,0.0628,0.0406,0.0380,
0.0382,0.0189,0.0134,0.0083,0.0037,0.0022,0.0015,0.0004,0.0002,-1,-1,-1,-1,-1,-1,0.001)
#TH01
y[,30]<-c(0.0002,0.0002,0.0006,0.2000,0.1871,0.1449,0.1906,0.2642,0.0121,0.0002,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#TPOX
y[,32]<-c(0.0101,0.0042,0.4927,0.1065,0.0656,0.2870,0.0327,0.0013,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0009)
#vWa
y[,34]<-c(0.0011,0.0020,0.1096,0.1328,0.2255,0.2611,0.1752,0.0760,0.0141,0.0026,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)

y<-data.frame(y) #criacao do dataframe com as frequencias alelicas da populacao

#obter a dimensao do ficheiro com as frequencias alelicas
n1<-dim(y)[1];n2<-dim(y)[2];

## marcadores observados num caso tipo (pai,mae e filho(a))

x<-read.csv("dados.csv", header= T ,sep = ";")
x[,2]<-as.character(x[,2])
```

```
#cria o vector com os alelos observados no pressuposto pai
pf<-rep(0,34);

j<-0;
for(j in 0:17){
      p<-1;j<-j+1;
      for(p in 1:17){
            l<-p*2-1;
            if(x[j,2]==marker[l]){
                  pf[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,pf[l+1]<-x[j,3],pf[l+1]<-x[j,4]);
            }
      }
}
#cria o vector com os alelos observados na mae
m<-rep(0,34);

j<-0;
for(j in 18:34){
      p<-1;j<-j+1;
      for(p in 18:34){
            l<-(p-17)*2-1;
            if(x[j,2]==marker[l]){
                  m[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,m[l+1]<-x[j,3],m[l+1]<-x[j,4]);
            }
      }
}

#cria o vector com os alelos observados na crianca
c<-rep(0,34);

j<-0;
for(j in 34:50){
      p<-1;j<-j+1;
      for(p in 35:51){
            l<-(p-34)*2-1;
            if(x[j,2]==marker[l]){
                  c[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,c[l+1]<-x[j,3],c[l+1]<-x[j,4]);
            }
      }
}

## este ciclo corresponde as frequencias alelicas aos respetivos alelos observados
## no caso de um trio familiar

fr_pf<-rep(0,n1);fr_m<-rep(0,n1);

i<-0;
for(i in 0:n2){
      k<-1;i<-i+1;
      for(k in 1:n1){
            ifelse(pf[i]==y[k,i],fr_pf[i]<-y[k,i+1],2)
            ifelse(pf[i+1]==y[k,i],fr_pf[i+1]<-y[k,i+1],2)
            ifelse(m[i]==y[k,i],fr_m[i]<-y[k,i+1],2)
            ifelse(m[i+1]==y[k,i],fr_m[i+1]<-y[k,i+1],2)
            #atribui frequencia minima caso o alelo nao exista
            ifelse(fr_pf[i]==0,fr_pf[i]<-y[28,i+1],2)
      }
}


# Dataframe final com os dados relevantes para o calculo do Likelyhood Ratio (LR)

x<-data.frame(marker,pf,fr_pf,m,fr_m,c);

# Criacao do modelo de mutacao

# Inserir os mutation rates conforme a ordem estipulada dos marcadores

m_rates<-c(0.0037,0.0025,0.0010,0.0013,0.0013,0.0010,0.0013,0.0020,0.0010,
0.0025,0.0029,0.0022,0.0014,0.0016,0.0004,0.0037,0.0022)

# Cria a matriz s

y1<-y[,seq(2, ncol(y), by = 2)]
y2<-y[,seq(1, ncol(y), by = 2)]

N<-length(y1)

s<-list();

i<-j<-k<-1
for(k in 1:N){
      k<-k*2-1;
      dim<-sum(as.numeric(y[,k]!=-1));
      s[[k]]<-matrix(0,nrow=dim,ncol=dim);
      for(i in 1:dim){
            sum<-0;
            for(j in i+1:dim-i){
                  s[[k]][i,j]<-0.5^abs(y[i,k]-y[j,k]);
                  s[[k]][j,i]<-s[[k]][i,j];
                  sum<-sum+s[[k]][i,j];
            }
            s[[k]][i,i]<--sum;
      }
```

```
}

s[sapply(s, is.null)] <- NULL #remove os NULL da lista

# Definicao do parametro lambda

lambda<-rep(0,N);i<-j<-0
for(i in 1:N){
      sum<-0;
      for(j in 1:dim(s[[i]])[1]){
            sum<-sum+s[[i]][j,j];
      }
      lambda[i]<-m_rates[i]/-sum;
}

## Criacao da matriz Q

Q<-list();

i<-j<-1
for(k in 1:N){
      dim<-sum(as.numeric(y1[,k]!=-1));
      names<-y2[,k][sapply(y2[,k], function(x) x!=-1)]
      Q[[k]]<-matrix(0,nrow=dim,ncol=dim,
            dimnames = list(as.character(names),
                  as.character(names)));

      for(i in 1:dim){
            j<-i+1;
            while(j<=dim){
                  Q[[k]][i,j]<-lambda[k]*s[[k]][i,j]/y1[i,k];
                  Q[[k]][j,i]<-y1[i,k]*Q[[k]][i,j]/y1[j,k];
                  j<-j+1;
            }
            Q[[k]][i,i]<-1+lambda[k]*s[[k]][i,i]/y1[i,k];
      }
}

# Criacao das matrizes de mutacao para os alelos provenientes do pai
# e para os provenientes da mae

qpf<-list();
qm<-list();

i<-K1<-K2<-J1<-J2<-k<-j<-p<-1;

for(k in 1:N){
      i<-k*2-1;K1<-K2<-1;
      mut.allel.pf<-unique(c(x[i,2],x[i+1,2])); #guarda os alelos do pai
      mut.allel.m<-unique(c(x[i,4],x[i+1,4])); #guarda os alelos da mae
      mut.allel.pf<-sort(mut.allel.pf)
      mut.allel.m<-sort(mut.allel.m)
      allel.pf<-vector();
      allel.m<-vector();
      for(j in 1:(n1-1)){
            if(sum(ifelse(y[j,i]==mut.allel.pf,s<-1,s<-0))==1){
                  allel.pf[K1]<-j; #guarda a posicao do alelo no dataframe das frequencias
                  K1<-K1+1;
            }
            if(sum(ifelse(y[j,i]==mut.allel.m,s<-1,s<-0))==1){
                  allel.m[K2]<-j; #guarda a posicao do alelo no dataframe das frequencias
                  K2<-K2+1;
            }

      }
      #define-se a dimensao das matrizes de mutacao

      dim1<-length(allel.pf)+1;dim2<-length(allel.m)+1;

      #contingencia para o caso de se veririficar um alelo ausente da populacao
      if(length(allel.pf)<length(mut.allel.pf)){
            dim1<-length(allel.pf)+2;
            allel.pf<-c(allel.pf,28);
      }

      qpf[[k]]<-matrix(0,nrow=dim1,ncol=dim1,
            dimnames = list(as.character(c(mut.allel.pf,"x")),
                  as.character(c(mut.allel.pf,"x"))));
      qm[[k]]<-matrix(0,nrow=dim2,ncol=dim2,
            dimnames = list(as.character(c(mut.allel.m,"x")),
                  as.character(c(mut.allel.m,"x"))));

      #este ciclo preenche as matrizes de mutacao com as probabilidades de
      # transicao dos alelos para os seus vizinhos

      for(p1 in 1:dim1-1){
            qpf[[k]][p1,p1]<-Q[[k]][allel.pf[p1],allel.pf[p1]]
            J1<-p1+1;
            while(J1<dim1){
                  qpf[[k]][p1,J1]<-Q[[k]][allel.pf[p1],allel.pf[J1]];
                  qpf[[k]][J1,p1]<-Q[[k]][allel.pf[J1],allel.pf[p1]];
                  J1<-J1+1;
            }
            qpf[[k]][p1,dim1]<-1-sum(qpf[[k]][p1,]);
            qpf[[k]][dim1,p1]<-sum(Q[[k]][,allel.pf[p1]])-sum(qpf[[k]][,p1]);
      }
      for(p2 in 1:dim2-1){
```

```
            qm[[k]][p2,p2]<-Q[[k]][allel.m[p2],allel.m[p2]]
            J2<-p2+1;
            while(J2<dim2){
                  qm[[k]][p2,J2]<-Q[[k]][allel.m[p2],allel.m[J2]];
                  qm[[k]][J2,p2]<-Q[[k]][allel.m[J2],allel.m[p2]];
                  J2<-J2+1;
            }
            qm[[k]][p2,dim2]<-1-sum(qm[[k]][p2,]);
            qm[[k]][dim2,p2]<-sum(Q[[k]][,allel.m[p2]])-sum(qm[[k]][,p2]);
      }
      qpf[[k]][dim1,dim1]<-1-sum(qpf[[k]][dim1,]);
      qm[[k]][dim2,dim2]<-1-sum(qm[[k]][dim2,]);
}

# funcao que cria as redes para calcular o LR

LR_trio<-function(x,alpha){

      lr<-list();
      lr[1]<-1;
      i<-1;j<-0;

      #este ciclo define toda a estrutura da rede para um trio familiar
      #executa-se para criar uma RB para cada marcador e calcular o seu LR

      for(i in 1:(n2/2)){ #inicio do ciclo
            i<-i*2-1;j<-j+1;
            pH<-0;mH<-0;equal<-0;

            #determina se existe homozigotia
            if(identical(x[i,2],x[i+1,2])==TRUE)pH<-1;
            if(identical(x[i,4],x[i+1,4])==TRUE)mH<-1;

            #verifica se existem alelos iguais e quantos sao
            equal<-length(unique(c(x[i,2],x[i+1,2],x[i,4],x[i+1,4])));

            #definicao dos niveis de cada nodo da rede

            mut_levels.pf<-unique(c(x[i,2],x[i+1,2]))
            mut_levels.pf<-sort(mut_levels.pf)
            mut_levels.pf<-c(mut_levels.pf,"x")
            mut_levels.m<-unique(c(x[i,4],x[i+1,4]))
            mut_levels.m<-sort(mut_levels.m)
            mut_levels.m<-c(mut_levels.m,"x")

            yn<-c("yes","no");
            allel.frq_pf<-unique(c(x[i,3],x[i+1,3],1-sum(unique(x[i:(i+1),3]))));
            allel.frq_m<-unique(c(x[i,5],x[i+1,5],1-sum(unique(x[i:(i+1),5]))));

            pf_levels<-as.character(unique(c(x[i,2],x[i+1,2],"x")));
            pfgt_levels<-unique(c(paste(x[i,2],x[i,2]),paste(x[i,2],x[i+1,2]),
                  paste(x[i,2],"x"),paste(x[i+1,2],x[i+1,2]),paste(x[i+1,2],"x"),paste("x","x")));
            m_levels<-as.character(unique(c(x[i,4],x[i+1,4],"x")));
            mgt_levels<-unique(c(paste(x[i,4],x[i,4]),paste(x[i,4],x[i+1,4]),
                  paste(x[i,4],"x"),paste(x[i+1,4],x[i+1,4]),paste(x[i+1,4],"x"),paste("x","x")));
            cpg_levels<-unique(c(x[i,2],x[i+1,2]));
            cmg_levels<-unique(c(x[i,4],x[i+1,4]));


            c_lvls<-c(cpg_levels,cmg_levels)
            c_lvls<-sort(c_lvls);c_lvls<-c(as.character(c_lvls),"x")
            cgt_levels<-vector();

            k1<-k2<-p<-1;
            for(k1 in 1:(length(c_lvls)-1)){
                  for(k2 in (k1+1):length(c_lvls)){
                        cgt_levels[p]<-paste(c_lvls[k1],c_lvls[k2]);
                        p<-p+1;
                  }
            }
            cgt_levels<-unique(cgt_levels);cgt_levels<-c(cgt_levels,paste("x","x"));

            ifelse(equal==4,remove<-c(paste(x[i,2],x[i+1,2]),paste(x[i,4],x[i+1,4])),
                  ifelse(pH+mH==1 && equal==3,remove<-c(paste(x[i,2],x[i+1,2]),paste(x[i,4],x[i+1,4])),remove<-0));
            cgt_levels<-cgt_levels[! cgt_levels %in% remove]

            cpg_levels<-c(cpg_levels,"x")
            cmg_levels<-c(cmg_levels,"x")

            #mediante as possiveis combinacoes alelicas dos intervenientes sao criados
            #os estados possiveis para cada tabela de probabilidade de cada nodo

            if(pH+mH==2){
                  cond.prob.pfgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
                  cond.prob.mgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
                  cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,1,
                        unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])),
                        unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3]))));
                  cond.prob.cpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
                  cond.prob.cmg<-c(1,0,0.5,0.5,0.5,0.5,0,1);

                  if(x[i,2]>x[i,4]){
                              cond.prob.cgt<-c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1);
                  } else ifelse(equal==1,cond.prob.cgt<-c(1,0,0,0,1,0,0,1,0,0,0,1),
                              cond.prob.cgt<-c(1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1));
            }
            if(pH==0 && mH==1){
```

```
        cond.prob.pfgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
              0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,
              0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
        cond.prob.mgt<-c(1,0,0,0,1,0,0,1,0,0,0,0,1);
        cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,0,1,0,0,0,1,
              x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),
              x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),
              x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3])));
        cond.prob.cpg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
              0.5,0.5,0,0,1,0,0,0.5,0.5,
              0.5,0,0.5,0,0.5,0.5,0,0,1);
        cond.prob.cmg<-c(1,0,0.5,0.5,0.5,0.5,0,1);

        if(equal==2){
              if(x[i+1,2]==x[i,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,
                          0,0,0,0,0,0,1,0,0,0,0,0,0,1);
              if(x[i+1,2]>x[i,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,
                          0,1,0,0,0,0,0,0,1,0,0,0,0,0,1);
        }
        if(equal==3){
              if(x[i,2]<x[i,4] && x[i+1,2]<x[i,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                          0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
              if(x[i,2]<x[i,4] && x[i+1,2]>x[i,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                          0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
              if(x[i,2]>x[i,4] && x[i+1,2]>x[i,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,
                          1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
        }
  }
  if(pH==1 && mH==0){
        cond.prob.pfgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
        cond.prob.mgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
              0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
              0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
        cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,1,
              unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])),
              unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])));
        cond.prob.cpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
        cond.prob.cmg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
              0.5,0.5,0,0,1,0,0,0.5,0.5,
              0.5,0,0.5,0,0.5,0.5,0,0,1);
        if(equal==2){
              if(x[i,2]==x[i,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,
                          0,0,0,1,0,0,0,1,0,0,0,0,0,0,1);
              if(x[i,2]>x[i,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,
                          0,0,1,0,0,0,0,1,0,0,0,0,0,0,1);
        }
        if(equal==3){
              if(x[i,2]>x[i,4] && x[i,2]>x[i+1,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                          0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
              if(x[i,2]>x[i,4] && x[i,2]<x[i+1,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                          0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1);
              if(x[i,2]<x[i,4] && x[i,2]<x[i+1,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,
                          1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1);
        }
  }
  if(pH+mH==0){
        cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,0,1,0,0,0,1,
              x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),
              x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),
              x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3])));
        cond.prob.pfgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
              0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
              0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
        cond.prob.mgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,
              0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
              0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
        cond.prob.cpg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
              0.5,0.5,0,0,1,0,0,0.5,0.5,
              0.5,0,0.5,0,0.5,0.5,0,0,1);
        cond.prob.cmg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
              0.5,0.5,0,0,1,0,0,0.5,0.5,
              0.5,0,0.5,0,0.5,0.5,0,0,1);
        if(equal==2){
              cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
                    0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
                    0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
        }
        if(equal==3){
              if(x[i,2]<x[i,4] && x[i+1,2]<x[i+1,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,
                          0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,
                          1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1);
              if(x[i,2]==x[i,4] && x[i+1,2]>x[i+1,4])
                    cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                          1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,
                          1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
              if(x[i,2]<x[i,4] && x[i+1,2]==x[i+1,4])
```

```
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
                  if(x[i,2]==x[i,4] && x[i+1,2]<x[i+1,4])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
                  if(x[i,2]>x[i,4] && x[i+1,2]>x[i+1,4])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
                  if(x[i,2]>x[i,4] && x[i+1,2]==x[i+1,4])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
            }
            if(equal==4){
                  if(x[i,2]<x[i,4] && x[i+1,2]>x[i+1,4])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,
                              0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
                  if(x[i,2]>x[i,4] && x[i+1,2]<x[i+1,4])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,
                              0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,
                              0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
                  if(x[i,2]<x[i,4] && x[i+1,2]<x[i+1,4])
                        ifelse(x[i+1,2]>x[i,4],cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,
                        0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1),
                              cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,
                              0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,
                              0,0,0,0,0,0,0,0,0,0,0,1));
                  if(x[i,2]>x[i,4] && x[i+1,2]>x[i+1,4])
                        ifelse(x[i,2]<x[i+1,4],cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,
                           0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,
                           0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1),
                              cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,
                              0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,
                              0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1));
            }
      }

      ##criacao das tabelas de probabilidade de cada nodo
      #parametro alpha define a probabilidade a priori a favor da paternidade

            Ttfpf<-cptable(~tfpf,values=c(alpha,1-alpha),levels=yn);
            Tpfpg<-cptable(~pfpg, values=allel.frq_pf, levels=pf_levels);
            Tpfmg<-cptable(~pfmg, values=allel.frq_pf, levels=pf_levels);
            Tmpg<-cptable(~mpg, values=allel.frq_m, levels=m_levels);
            Tmmg<-cptable(~mmg, values=allel.frq_m, levels=m_levels);
            Tmgt<-cptable(~mgt|mpg:mmg, values=cond.prob.mgt,levels=mgt_levels);
            Tpfgt<-cptable(~pfgt|pfpg:pfmg, values=cond.prob.pfgt,levels=pfgt_levels);
            Ttfpg<-cptable(~tfpg|pfpg:tfpf, values=cond.prob.tfpg,levels=pf_levels);
            Ttfmg<-cptable(~tfmg|pfmg:tfpf, values=cond.prob.tfmg,levels=pf_levels);
            Tcopg<-cptable(~copg|tfpg:tfmg,values=cond.prob.cpg,levels=cpg_levels);
            Tcomg<-cptable(~comg|mpg:mmg,values=cond.prob.cmg,levels=cmg_levels);
            Tcgt<-cptable(~cgt|capg:camg, values=cond.prob.cgt,levels=cgt_levels);

            Tcapg<-cptable(~capg|copg,values=qpf[[j]],levels=mut_levels.pf)
            Tcamg<-cptable(~camg|comg,values=qm[[j]],levels=mut_levels.m)

            plist<-compileCPT(list(Tpfgt,Ttfpf,Tpfpg,Tpfmg,Tmpg,Tmmg,Tcomg,Tmgt,Ttfpg,
                  Ttfmg,Tcopg,Tcgt,Tcapg,Tcamg));


      ##Criacao da RB do caso e consequente calculo do valor de LR

      net<-grain(plist);

      # este passo e necessario para definir informacao discordante observada como "x"

      s<-as.character(c(paste(x[i,2],x[i+1,2]),paste(x[i,4],x[i+1,4]),paste(x[i,6],x[i+1,6])));
      allel<-intersect(c(x[i,4],x[i+1,4]),c(x[i,6],x[i+1,6]));
      eq<-setequal(c(x[i,4],x[i+1,4]),c(x[i,6],x[i+1,6]));
      if(sum(s[3]==cgt_levels)<1)
            ifelse(eq<1,s[3]<-paste(allel,"x"),s[3]<-paste("x","x"));

      #propagacao da evidencia observada na rede
      bnetE<-setEvidence(net,nodes=c("pfgt","mgt","cgt"),states=s)

      #calculo do valor de LR
      lr[j+1]<-querygrain(bnetE)$tfpf[1]/querygrain(bnetE)$tfpf[2]

} #fim do ciclo

#calculo do valor de LR global como o produto dos LR de cada um dos marcadores analisados

lr<-as.numeric(lr);
lhr<-prod(lr);
output<-list(lr,lhr);

#devolve como output os LR de cada marcador e o LR global
return(output);

}

#executa a funcao LR_trio.
```

```
LR_trio(x,0.5)
```

Código de R para o Caso II

```
#Executar a biblioteca gRain

library(gRain)

##criar ficheiro com as frequencias alelicas para execucoes futuras

#lista de marcadores geneticos analisados
marker<-c("CSF1PO","CSF1PO","D2S1338","D2S1338","D3S1358","D3S1358","D5S818","D5S818",
"D7S820","D7S820","D8S1179","D8S1179","D13S317","D13S317","D16S539","D16S539",
"D18S51","D18S51","D19S433","D19S433","D21S11","D21S11","FGA","FGA","PentaD","PentaD",
"PentaE","PentaE","TH01","TH01","TPOX","TPOX","vWA","vWA")

#criacao da matrix de marcadores alelicos e respetivas frequencias
y<-matrix(0,nrow=28,ncol=34);
colnames(y)<-marker;

#insercao dos alelos de cada um dos marcadores
#CSF1PO
y[,1]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D2S1338
y[,3]<-c(13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D3S1358
y[,5]<-c(9,10,11,12,13,14,15,16,17,18,19,20,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D5S818
y[,7]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D7S820
y[,9]<-c(6,7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D8S1179
y[,11]<-c(8,9,10,11,12,13,14,15,16,17,18,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D13S317
y[,13]<-c(8,9,10,11,12,13,14,15,16,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D16S539
y[,15]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D18S51
y[,17]<-c(10,10.2,11,12,13,13.2,14,14.2,15,15.2,16,17,18,19,20,21,22,23,24,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D19S433
y[,19]<-c(10,11,11.2,12,12.2,13,13.2,14,14.2,15,15.2,16,16.2,17,17.2,18,18.2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D21S11
y[,21]<-c(21,24,24.2,25,25.2,26,26.2,27,28,28.2,29,29.2,30,30.2,31,31.2,32,32.2,33,33.2,34,34.1,34.2,35,35.2,36,37,-2)
#FGA
y[,23]<-c(16,17,18,18.2,19,20,20.2,21,21.2,22,22.2,23,23.2,24,24.2,25,25.2,26,27,27.2,28,29,30.2,31.2,43.2,-1,-1,-2)
#PentaD
y[,25]<-c(2.2,3.2,5,6,7,8,9,10,11,12,13,14,15,16,17,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#PentaE
y[,27]<-c(5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,-1,-1,-1,-1,-1,-1,-2)
#TH01
y[,29]<-c(3,4,5,6,7,8,9,9.3,10,11,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#TPOX
y[,31]<-c(6,7,8,9,10,11,12,13,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#vWa
y[,33]<-c(11,13,14,15,16,17,18,19,20,21,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)

#insercao das frequencias de cada um dos alelos de
#cada um dos marcadores e respetivas frequencias minimas
#CSF1PO
y[,2]<-c(0.0064,0.0114,0.0185,0.2657,0.3155,0.3215,0.05235,0.0079,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0010)
#D2S1338
y[,4]<-c(0.0004,0.0007,0.0015,0.0501,0.2415,0.0723,0.1045,0.1418,0.0468,0.0409,
0.1034,0.1041,0.0779,0.0127,0.0011,0.0004,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D3S1358
y[,6]<-c(0.0004,0.0002,0.0006,0.0039,0.0057,0.1036,0.2589,0.2629,0.2007,0.1491,
0.0134,0.0007,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D5S818
y[,8]<-c(0.0004,0.0145,0.0376,0.0617,0.3349,0.3676,0.1724,0.0101,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D7S820
y[,10]<-c(0.0002,0.0204,0.1478,0.1183,0.2896,0.2266,0.1561,0.0353,0.0053,0.0006,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D8S1179
y[,12]<-c(0.0095,0.0101,0.0762,0.0846,0.1236,0.2901,0.2525,0.1230,0.0261,0.0035,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D13S317
y[,14]<-c(0.1340,0.0560,0.0485,0.3256,0.2793,0.1144,0.0406,0.0015,0.0002,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D16S539
y[,16]<-c(0.0002,0.0211,0.1355,0.0602,0.3010,0.2745,0.1776,0.0292,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D18S51
y[,18]<-c(0.0145,0.0006,0.0083,0.1328,0.1269,0.0009,0.1381,0.0002,0.1428,0.0002,0.1553,
0.1272,0.0709,0.0426,0.0220,0.0112,0.0024,0.0017,0.0006,-1,-1,-1,-1,-1,-1,-1,-1,0.0012)
#D19S433
y[,20]<-c(0.0029,0.0112,0.0002,0.1104,0.0050,0.2534,0.0132,0.3127,0.0307,0.1460,0.0455,
0.0466,0.0152,0.0031,0.0017,0.0006,0.0017,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D21S11
y[,22]<-c(0.0002,0.0002,0.0024,0.0004,0.0009,0.0013,0.0004,0.0257,0.1577,0.0002,0.2264,0.0007,
0.2402,0.0354,0.0610,0.1087,0.0088,0.0883,0.0015,0.0305,0.0006,0.0002,0.0050,0.0024,0.0007,0.0002,0.0002,0.0011)
#FGA
y[,24]<-c(0.0004,0.0026,0.0156,0.0011,0.0676,0.1267,0.0002,0.1759,0.0018,0.1864,0.0059,0.1484,
```

```
0.0022,0.1357,0.0004,0.0817,0.0004,0.0347,0.0073,0.0002,0.0028,0.0013,0.0002,0.0006,0.0002,-1,-1,0.0011)
#PentaD
y[,26]<-c(0.0187,0.0015,0.0033,0.0007,0.0095,0.0206,0.1906,0.1144,0.1616,0.1855,0.1913,
0.0747,0.0202,0.0062,0.0011,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#PentaE
y[,28]<-c(0.0648,0.0013,0.1399,0.0288,0.0134,0.0854,0.1254,0.1968,0.1160,0.0628,0.0406,0.0380,
0.0382,0.0189,0.0134,0.0083,0.0037,0.0022,0.0015,0.0004,0.0002,-1,-1,-1,-1,-1,-1,0.001)
#TH01
y[,30]<-c(0.0002,0.0002,0.0006,0.2000,0.1871,0.1449,0.1906,0.2642,0.0121,0.0002,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#TPOX
y[,32]<-c(0.0101,0.0042,0.4927,0.1065,0.0656,0.2870,0.0327,0.0013,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0009)
#vWa
y[,34]<-c(0.0011,0.0020,0.1096,0.1328,0.2255,0.2611,0.1752,0.0760,0.0141,0.0026,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)

y<-data.frame(y) #criacao do dataframe com as frequencias alelicas da populacao

#obter a dimensao do ficheiro com as frequencias alelicas
n1<-dim(y)[1];n2<-dim(y)[2];

## marcadores observados num caso tipo (pai,mae e filho(a))

x<-read.csv("dados.csv", header= T ,sep = ";")
x[,2]<-as.character(x[,2])

#cria o vector com os alelos observados no pai do pressuposto pai
gf<-rep(0,34);

j<-0;
for(j in 0:17){
      p<-1;j<-j+1;
      for(p in 1:17){
            l<-p*2-1;
            if(x[j,2]==marker[l]){
                  gf[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,gf[l+1]<-x[j,3],gf[l+1]<-x[j,4]);
            }
      }
}

#cria o vector com os alelos observados na mae do pressuposto pai
gm<-rep(0,34);

j<-0;
for(j in 18:34){
      p<-1;j<-j+1;
      for(p in 18:34){
            l<-(p-17)*2-1;
            if(x[j,2]==marker[l]){
                  gm[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,gm[l+1]<-x[j,3],gm[l+1]<-x[j,4]);
            }
      }
}

#cria o vector com os alelos observados na mae
m<-rep(0,34);

j<-0;
for(j in 35:51){
      p<-1;j<-j+1;
      for(p in 35:51){
            l<-(p-34)*2-1;
            if(x[j,2]==marker[l]){
                  m[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,m[l+1]<-x[j,3],m[l+1]<-x[j,4]);
            }
      }
}

#cria o vector com os alelos observados na crianca
c<-rep(0,34);

j<-0;
for(j in 51:67){
      p<-1;j<-j+1;
      for(p in 52:68){
            l<-(p-51)*2-1;
            if(x[j,2]==marker[l]){
                  c[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,c[l+1]<-x[j,3],c[l+1]<-x[j,4]);
            }
      }
}

## este ciclo corresponde as frequencias alelicas aos respetivos alelos observados
## no caso de um trio familiar


fr_gf<-rep(0,n1);fr_gm<-rep(0,n1);fr_m<-rep(0,n1);fr_c<-rep(0,n1);

i<-0;
for(i in 0:n2){
      k<-1;i<-i+1;
      for(k in 1:n1){
```

```
            ifelse(gf[i]==y[k,i],fr_gf[i]<-y[k,i+1],2)
            ifelse(gf[i+1]==y[k,i],fr_gf[i+1]<-y[k,i+1],2)
            ifelse(gm[i]==y[k,i],fr_gm[i]<-y[k,i+1],2)
            ifelse(gm[i+1]==y[k,i],fr_gm[i+1]<-y[k,i+1],2)
            ifelse(m[i]==y[k,i],fr_m[i]<-y[k,i+1],2)
            ifelse(m[i+1]==y[k,i],fr_m[i+1]<-y[k,i+1],2)
            ifelse(c[i]==y[k,i],fr_c[i]<-y[k,i+1],2)
            ifelse(c[i+1]==y[k,i],fr_c[i+1]<-y[k,i+1],2)
            ifelse(fr_gf[i]==0,fr_gf[i]<-y[28,i+1]n,2)
            ifelse(fr_gm[i]==0,fr_gm[i]<-y[28,i+1],2)
      }
}

## Dataframe final com os dados relevantes para o calculo do Likelyhood Ratio (LR)

x<-data.frame(marker,gf,fr_gf,gm,fr_gm,m,fr_m,c,fr_c);

## Criacao do modelo de mutacao

# Inserir os mutation rates conforme a ordem estipulada dos marcadores

m_rates<-c(0.0037,0.0025,0.0010,0.0013,0.0013,0.0010,0.0013,0.0020,
0.0010,0.0025,0.0029,0.0022,0.0014,0.0016,0.0004,0.0037,0.0022)


# Cria a matriz s

y1<-y[,seq(2, ncol(y), by = 2)]
y2<-y[,seq(1, ncol(y), by = 2)]

N<-length(y1)

s<-list();

i<-j<-k<-1
for(k in 1:N){
      k<-k*2-1;
      dim<-sum(as.numeric(y[,k]!=-1));
      s[[k]]<-matrix(0,nrow=dim,ncol=dim);
      for(i in 1:dim){
            sum<-0;
            for(j in i+1:dim-i){
                  s[[k]][i,j]<-0.5^abs(y[i,k]-y[j,k]);
                  s[[k]][j,i]<-s[[k]][i,j];
                  sum<-sum+s[[k]][i,j];
            }
            s[[k]][i,i]<--sum;
      }
}

s[sapply(s, is.null)] <- NULL #remove os NULL da lista

# Definicao do parametro lambda

lambda<-rep(0,N);i<-j<-0
for(i in 1:N){
      sum<-0;
      for(j in 1:dim(s[[i]])[1]){
            sum<-sum+s[[i]][j,j];
      }
      lambda[i]<-m_rates[i]/-sum;
}

## Criacao da matriz Q

Q<-list();

i<-j<-1
for(k in 1:N){
      dim<-sum(as.numeric(y1[,k]!=-1));
      names<-y2[,k][sapply(y2[,k], function(x) x!=-1)]
      Q[[k]]<-matrix(0,nrow=dim,ncol=dim,
            dimnames = list(as.character(names),
                        as.character(names)));

      for(i in 1:dim){
            j<-i+1;
            while(j<=dim){
                  Q[[k]][i,j]<-lambda[k]*s[[k]][i,j]/y1[i,k];
                  Q[[k]][j,i]<-y1[i,k]*Q[[k]][i,j]/y1[j,k];
                  j<-j+1;
            }
            Q[[k]][i,i]<-1+lambda[k]*s[[k]][i,i]/y1[i,k];
      }
}

# Criacao das matrizes de mutacao para os alelos provenientes do pai
# e para os provenientes da mae

qpf<-list();
qm<-list();

i<-K1<-K2<-J1<-J2<-k<-j<-p<-1;
for(k in 1:N){
      i<-k*2-1;K1<-K2<-1;

      gfH<-0;gmH<-0;mH<-0;cH<-0;dif_gf<-dim_gm<-0;allel_gf<-allel_gm<-0;equal<-0;
```

```
      if(identical(x[i,2],x[i+1,2])==TRUE)gfH<-1;
      if(identical(x[i,4],x[i+1,4])==TRUE)gmH<-1;
      if(identical(x[i,6],x[i+1,6])==TRUE)mH<-1;
      if(identical(x[i,8],x[i+1,8])==TRUE)cH<-1;

      mut.allel.pf<-unique(c(x[i,8],x[i+1,8])); #guarda os alelos do pai
      mut.allel.m<-unique(c(x[i,6],x[i+1,6])); #guarda os alelos da mae


      allel_gf<-intersect(c(x[i,2],x[i+1,2]),c(x[i,8],x[i+1,8]));
      allel_gm<-intersect(c(x[i,4],x[i+1,4]),c(x[i,8],x[i+1,8]));
      if(length(allel_gf)==0)allel_gf<-0;
      if(length(allel_gm)==0)allel_gm<-0;

            if(length(allel_gf)==1 && length(allel_gm)==1 && allel_gf!=0 && allel_gm!=0){
                  ifelse(allel_gm==allel_gf,mut.allel.pf<-c(allel_gf),
                        ifelse(allel_gm<=allel_gf,mut.allel.pf<-c(allel_gm,allel_gf),
                              mut.allel.pf<-c(allel_gf,allel_gm)));
            }
            if((length(allel_gf)==2 || allel_gf==0) && (length(allel_gm)==1 && allel_gm!=0)){
                  ifelse(allel_gf==0,mut.allel.pf<-c(allel_gm),
                        ifelse(allel_gm<=allel_gf[1],mut.allel.pf<-c(allel_gm,allel_gf[2]),
                           mut.allel.pf<-c(allel_gf[1],allel_gm)));

            }
            if((length(allel_gm)==2 || allel_gm==0) && (length(allel_gf)==1 && allel_gf!=0)){
                  ifelse(allel_gm==0,mut.allel.pf<-c(allel_gf),
                        ifelse(allel_gm[1]<=allel_gf,mut.allel.pf<-c(allel_gm[1],allel_gf),
                           mut.allel.pf<-c(allel_gf,allel_gm[1])));
            }

      mut.allel.pf<-sort(mut.allel.pf)
      mut.allel.m<-sort(mut.allel.m)

      allel.pf<-vector();
      allel.m<-vector();
      for(j in 1:(n1-1)){
            if(sum(ifelse(y[j,i]==mut.allel.pf,s<-1,s<-0))==1){
                  allel.pf[K1]<-j; #guarda a posicao do alelo no dataframe das frequencias
                  K1<-K1+1;
            }
            if(sum(ifelse(y[j,i]==mut.allel.m,s<-1,s<-0))==1){
                  allel.m[K2]<-j; #guarda a posicao do alelo no dataframe das frequencias
                  K2<-K2+1;
            }

      }

      #define-se a dimensao das matrizes de mutacao
      dim1<-length(allel.pf)+1;dim2<-length(allel.m)+1;

      #contingencia para o caso de se veririficar um alelo ausente da populacao
      if(length(allel.pf)<length(mut.allel.pf)){
            dim1<-length(allel.pf)+2;
            allel.pf<-c(allel.pf,28);
      }

      qpf[[k]]<-matrix(0,nrow=dim1,ncol=dim1,
            dimnames = list(as.character(c(mut.allel.pf,"x")),
                        as.character(c(mut.allel.pf,"x"))));
      qm[[k]]<-matrix(0,nrow=dim2,ncol=dim2,
            dimnames = list(as.character(c(mut.allel.m,"x")),
                        as.character(c(mut.allel.m,"x"))));

      #este ciclo preenche as matrizes de mutacao com as probabilidades de
      # transicao dos alelos para os seus vizinhos

      for(p1 in 1:dim1-1){
            qpf[[k]][p1,p1]<-Q[[k]][allel.pf[p1],allel.pf[p1]]
            J1<-p1+1;
            while(J1<dim1){
                  qpf[[k]][p1,J1]<-Q[[k]][allel.pf[p1],allel.pf[J1]];
                  qpf[[k]][J1,p1]<-Q[[k]][allel.pf[J1],allel.pf[p1]];
                  J1<-J1+1;
            }
            qpf[[k]][p1,dim1]<-1-sum(qpf[[k]][p1,]);
            qpf[[k]][dim1,p1]<-sum(Q[[k]][,allel.pf[p1]])-sum(qpf[[k]][,p1]);
      }
      for(p2 in 1:dim2-1){
            qm[[k]][p2,p2]<-Q[[k]][allel.m[p2],allel.m[p2]]
            J2<-p2+1;
            while(J2<dim2){
                  qm[[k]][p2,J2]<-Q[[k]][allel.m[p2],allel.m[J2]];
                  qm[[k]][J2,p2]<-Q[[k]][allel.m[J2],allel.m[p2]];
                  J2<-J2+1;
            }
            qm[[k]][p2,dim2]<-1-sum(qm[[k]][p2,]);
            qm[[k]][dim2,p2]<-sum(Q[[k]][,allel.m[p2]])-sum(qm[[k]][,p2]);
      }
      qpf[[k]][dim1,dim1]<-1-sum(qpf[[k]][dim1,]);
      qm[[k]][dim2,dim2]<-1-sum(qm[[k]][dim2,]);
}

## funcao que cria as redes para calcular o LR

LR_avos<-function(x,alpha){
```

```
lr<-list();
lr[1]<-1;
i<-1;j<-0;

#este ciclo define toda a estrutura da rede para um trio familiar
#executa-se para criar uma RB para cada marcador e calcular o seu LR

for(i in 1:(n2/2)){ #inicio do ciclo
      i<-i*2-1;j<-j+1;
      gfH<-0;gmH<-0;mH<-0;cH<-0;dif_gf<-dim_gm<-0;allel_gf<-allel_gm<-0;equal<-0;

      #verifica se existem alelos iguais e quantos sao
      equal<-length(unique(c(x[i,6],x[i+1,6],x[i,8],x[i+1,8])));

      #determina se existe homozigotia

      if(identical(x[i,2],x[i+1,2])==TRUE)gfH<-1;
      if(identical(x[i,4],x[i+1,4])==TRUE)gmH<-1;
      if(identical(x[i,6],x[i+1,6])==TRUE)mH<-1;
      if(identical(x[i,8],x[i+1,8])==TRUE)cH<-1;

      #definicao dos niveis de cada nodo da rede

      dif_gf<-length(unique(c(x[i,2],x[i+1,2],x[i,8],x[i+1,8])));
      allel_gf<-intersect(c(x[i,2],x[i+1,2]),c(x[i,8],x[i+1,8]));
      if(length(allel_gf)==0)allel_gf<-0;
      eq_gf<-setequal(c(x[i,2],x[i+1,2]),c(x[i,8],x[i+1,8]));
      gf_values<-unique(c(x[i,9],x[i+1,9]));
      ifelse(allel_gf==0,fr_allel_gf<-gf_values,fr_allel_gf<-intersect(c(x[i,3],x[i+1,3]),c(x[i,9],x[i+1,9])));

      dif_gm<-length(unique(c(x[i,4],x[i+1,4],x[i,8],x[i+1,8])));
      allel_gm<-intersect(c(x[i,4],x[i+1,4]),c(x[i,8],x[i+1,8]));
      if(length(allel_gm)==0)allel_gm<-0;
      eq_gm<-setequal(c(x[i,4],x[i+1,4]),c(x[i,8],x[i+1,8]));
      gm_values<-unique(c(x[i,9],x[i+1,9]));
      ifelse(allel_gm==0,fr_allel_gm<-gm_values,fr_allel_gm<-intersect(c(x[i,5],x[i+1,5]),c(x[i,9],x[i+1,9])));

      yn<-c("yes","no");
      allel.frq_gf<-unique(c(x[i,3],x[i+1,3],1-sum(unique(x[i:(i+1),3]))));
      allel.frq_gm<-unique(c(x[i,5],x[i+1,5],1-sum(unique(x[i:(i+1),5]))));
      allel.frq_m<-unique(c(x[i,7],x[i+1,7],1-sum(unique(x[i:(i+1),7]))));
      allel.frq_c<-unique(c(x[i,9],x[i+1,9],1-sum(unique(x[i:(i+1),9]))));

      frq_gf<-intersect(c(x[i,3],x[i+1,3]),c(x[i,9],x[i+1,9]));
      frq_gm<-intersect(c(x[i,5],x[i+1,5]),c(x[i,9],x[i+1,9]));


      ifelse(length(frq_gf)!=0,allel.frq_gf<-c(frq_gf,1-sum(frq_gf)),allel.frq_gf<-allel.frq_c);
      ifelse(length(frq_gm)!=0,allel.frq_gm<-c(frq_gm,1-sum(frq_gm)),allel.frq_gm<-allel.frq_c);

      m_levels<-as.character(unique(c(x[i,6],x[i+1,6],"x")));
      mgt_levels<-unique(c(paste(x[i,6],x[i,6]),paste(x[i,6],x[i+1,6]),
            paste(x[i,6],"x"),paste(x[i+1,6],x[i+1,6]),paste(x[i+1,6],"x"),paste("x","x")));

      gf_levels<-sort(unique(c(x[i,2],x[i+1,2],x[i,8],x[i+1,8],"x")));
      gm_levels<-sort(unique(c(x[i,4],x[i+1,4],x[i,8],x[i+1,8],"x")));


      copg_levels<-unique(c(x[i,8],x[i+1,8]));
      comg_levels<-unique(c(x[i,6],x[i+1,6]));

      cpg<-0;
      cond.prob.cpg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0.5,0,0,1,0,0,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);

      if(length(allel_gf)==1 && length(allel_gm)==1 && allel_gf!=0 && allel_gm!=0){
            ifelse(allel_gm==allel_gf,copg_levels<-c(allel_gf),
                  ifelse(allel_gm<=allel_gf,copg_levels<-c(allel_gm,allel_gf),
                     copg_levels<-c(allel_gf,allel_gm)));
            cpg<-1; #partilham os dois 1 alelo
      }
      if((length(allel_gf)==2 || allel_gf==0) && (length(allel_gm)==1 && allel_gm!=0)){
            ifelse(allel_gf==0,copg_levels<-c(allel_gm),
                  ifelse(allel_gm<=allel_gf[1],copg_levels<-c(allel_gm,allel_gf[2]),
                     copg_levels<-c(allel_gf[1],allel_gm)));
            cpg<-2; #avo partilha 1 alelo
      }
      if((length(allel_gm)==2 || allel_gm==0) && (length(allel_gf)==1 && allel_gf!=0)){
            ifelse(allel_gm==0,copg_levels<-c(allel_gf),
                  ifelse(allel_gm[1]<=allel_gf,copg_levels<-c(allel_gm[1],allel_gf),
                     copg_levels<-c(allel_gf,allel_gm[1])));
            cpg<-3; #avo partilha 1 alelo
      }

      if(cpg==3){
            if(cH==0){
                  if(allel_gm[1]==0){
                        cond.prob.cpg<-c(0.5,0,0,0.5,1,0,0.5,0.5,0.5,0.5,0,1);
                  }
                  else { ifelse(allel_gm<=allel_gf,cond.prob.cpg<-c(1,0,0,0.5,0,0.5,0.5,0.5,0,
                        0,0.5,0.5,0.5,0,0.5,0,0,1),cond.prob.cpg<-c(0.5,0.5,0,0.5,0,0.5,0,1,0,
                        0,0.5,0.5,0.5,0,0.5,0,0,1));
                  }
            }
            if(cH==1){
                  cond.prob.cpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            }
```

```
}
if(cpg==2){
      if(cH==0){
            if(allel_gf[1]==0){
                  cond.prob.cpg<-c(0.5,0,1,0,0.5,0.5,0,0.5,0.5,0.5,0,1);
            }
            else { ifelse(allel_gm<=allel_gf,cond.prob.cpg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
                  0.5,0,0.5,0,0.5,0.5,0,0,1),cond.prob.cpg<-c(0.5,0.5,0,0,1,0,0,0.5,0.5,
                  0.5,0,0.5,0,0.5,0.5,0,0,1));
            }
      }
      if(cH==1){
            cond.prob.cpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
      }
}
if(cpg==1)
      ifelse(allel_gm==allel_gf,cond.prob.cpg<-c(1,0,0.5,0.5,0.5,0.5,0,1),
            ifelse(allel_gm<=allel_gf,cond.prob.cpg<-c(0.5,0.5,0,0.5,0,0.5,0,0.5,0.5,0,0,1),
                  cond.prob.cpg<-c(0.5,0.5,0,0,0.5,0.5,0,0.5,0.5,0,0,1)));

c_levels<-as.character(unique(c(x[i,8],x[i+1,8],"x")));

remove_gf<-0;
if(gfH==0){
      ifelse(dif_gf!=4,gf_levels<-c(allel_gf,"x"),remove_gf<-unique(c(x[i,2],x[i+1,2])));
} else ifelse(dif_gf!=3,gf_levels<-paste("x","x"),remove_gf<-unique(c(x[i,2],x[i+1,2])));

gf_levels<-gf_levels[! gf_levels %in% remove_gf]


if(dif_gf==2 && cH<=gfH) gf_levels<-c_levels;

ifelse(sum(allel_gf)==0 || eq_gf==TRUE,gf_levels<-c_levels,gf_levels<-c(allel_gf,"x"));

gfgt_levels<-vector();
k1<-k2<-p<-1;k3<-1
for(k1 in 1:(length(gf_levels)-1)){
      for(k2 in k3:(length(gf_levels))){
            gfgt_levels[p]<-paste(gf_levels[k1],gf_levels[k2]);
            p<-p+1;
      }
      k3<-k3+1;
}

gfgt_levels<-c(gfgt_levels,paste("x","x"));

remove_gm<-0;
if(gmH==0){
      ifelse(dif_gm!=4,gm_levels<-c(allel_gm,"x"),remove_gm<-unique(c(x[i,4],x[i+1,4])));
} else ifelse(dif_gm!=3,gm_levels<-paste("x","x"),remove_gm<-unique(c(x[i,4],x[i+1,4])));

gm_levels<-gm_levels[! gm_levels %in% remove_gm]

if(dif_gm==2 && cH<=gmH) gm_levels<-c_levels;

ifelse(sum(allel_gm)==0 || eq_gm==TRUE,gm_levels<-c_levels,gm_levels<-c(allel_gm,"x"));


gmgt_levels<-vector();
k1<-k2<-p<-1;k3<-1;
for(k1 in 1:(length(gm_levels)-1)){
      for(k2 in k3:(length(gm_levels))){
            gmgt_levels[p]<-paste(gm_levels[k1],gm_levels[k2]);
            p<-p+1;
      }
      k3<-k3+1;
}

gmgt_levels<-c(gmgt_levels,paste("x","x"));
comg_levels<-unique(c(x[i,6],x[i+1,6]));

k<-p<-1;remove_copg<-vector();
for(k in 1:(length(copg_levels)-1)){
            remove_copg[p]<-paste(copg_levels[k],copg_levels[k+1]);
            p<-p+1;
}

k<-p<-1;remove_comg<-vector();
for(k in 1:(length(comg_levels)-1)){

            remove_comg[p]<-paste(comg_levels[k],comg_levels[k+1]);
            p<-p+1;

}

dim_comg<-length(comg_levels);dim_copg<-length(copg_levels);
eq_cog<-length(unique(c(comg_levels,copg_levels)));

ifelse(dim_comg+dim_copg==2 && eq_cog==2 || eq_cog>2,rem_cgt<-c(remove_comg,remove_copg),rem_cgt<-0);
if(dim_comg+dim_copg==4 && eq_cog>2) rem_cgt<-0;

c_lvls<-c(copg_levels,comg_levels)
c_lvls<-sort(c_lvls);c_lvls<-c(as.character(c_lvls),"x")
cgt_levels<-vector();

k1<-k2<-p<-1;
for(k1 in 1:(length(c_lvls)-1)){
```

```
      for(k2 in (k1+1):length(c_lvls)){
            cgt_levels[p]<-paste(c_lvls[k1],c_lvls[k2]);
            p<-p+1;
      }
}
cgt_levels<-unique(cgt_levels);
cgt_levels<-c(cgt_levels,paste("x","x"));


cgt_levels<-cgt_levels[! cgt_levels %in% rem_cgt]

copg_levels<-c(copg_levels,"x");
comg_levels<-c(comg_levels,"x");

equal<-length(unique(c(comg_levels,copg_levels)));
if(length(copg_levels)==2) cH<-1;

#mediante as possiveis combinacoes alelicas dos intervenientes sao criados
#os estados possiveis para cada tabela de probabilidade de cada nodo

if(cH+mH==2){
      cond.prob.pfgt<-c(1,0,0.5,0.5,0.5,0.5,0,1);
      cond.prob.mgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,1,
            unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])),
            unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])));
      cond.prob.cmg<-c(1,0,0.5,0.5,0.5,0.5,0,1);

      if(x[i,2]>x[i,4]){
                  cond.prob.cgt<-c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1);
      } else ifelse(equal==1,cond.prob.cgt<-c(1,0,0,0,1,0,0,1,0,0,0,1),
                  cond.prob.cgt<-c(1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1));
}
if(cH==0 && mH==1){
      cond.prob.pfgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
            0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
            0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
      cond.prob.mgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,0,1,0,0,0,1,
            x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),
            x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),
            x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]));
      cond.prob.gmgt<-cond.prob.gfgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,
            0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,
            0,0,1,0,0,0,0,0,0,1);
      cond.prob.cmg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
      cond.prob.pfpg<-cond.prob.pfmg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0.5,0,0,
                  1,0,0,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);

      if(length(allel_gf)==1 && allel_gf != 0){
            cond.prob.pfpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfpg<-c(1,0,0,1,fr_allel_gf,1-fr_allel_gf,fr_allel_gf, 1-fr_allel_gf);
            cond.prob.gfgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      }

      if(length(allel_gm)==1 && allel_gm != 0){
            cond.prob.pfmg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfmg<-c(1,0,0,1,fr_allel_gm,1-fr_allel_gm,fr_allel_gm, 1-fr_allel_gm);
            cond.prob.gmgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      }

      if(equal==3){
            if(x[i+1,6]==x[i,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,
                        0,0,0,0,0,1,0,0,0,0,0,0,1);
            if(x[i+1,6]>x[i,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,
                        0,1,0,0,0,0,0,0,1,0,0,0,0,0,1);
      }
      if(equal==4){
            if(x[i,2]<x[i,4] && x[i+1,2]<x[i,4])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                        0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
            if(x[i,2]<x[i,4] && x[i+1,2]>x[i,4])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                        0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
            if(x[i,2]>x[i,4] && x[i+1,2]>x[i,4])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
                        1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
      }
}
if(cH==1 && mH==0){
      cond.prob.pfgt<-c(1,0,0.5,0.5,0.5,0.5,0,1);
      cond.prob.mgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
            0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
            0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
      cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,0,1,0,0,0,1,
            unique(x[i:(i+1),9]),1-sum(unique(x[i:(i+1),9])),
            unique(x[i:(i+1),9]),1-sum(unique(x[i:(i+1),9])),
            unique(x[i:(i+1),9]),1-sum(unique(x[i:(i+1),9])));
      cond.prob.gmgt<-cond.prob.gfgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,
            0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,
            0,0,1,0,0,0,0,0,0,1);
      cond.prob.cmg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
            0.5,0.5,0,0,1,0,0,0.5,0.5,
            0.5,0,0.5,0,0.5,0.5,0,0,1);
      cond.prob.pfpg<-cond.prob.pfmg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0.5,0,0,
```

```
                  1,0,0,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);
      if(length(allel_gf)==1 && dif_gf<4){
            cond.prob.pfpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfpg<-c(1,0,0,1,fr_allel_gf,1-fr_allel_gf,fr_allel_gf, 1-fr_allel_gf);
            cond.prob.gfgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      }
      if(length(allel_gm)==1 && dif_gm<4){
            cond.prob.pfmg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfmg<-c(1,0,0,1,fr_allel_gm,1-fr_allel_gm,fr_allel_gm, 1-fr_allel_gm);
            cond.prob.gmgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      }

      if(equal==3){
            if(x[i,6]<x[i,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,1,0,0,0,0,0,1,0,
                        0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,1);
            if(x[i,6]==x[i,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,
                        0,0,0,1,0,0,0,1,0,0,0,0,0,0,1);
            if(x[i,6]>x[i,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,
                        0,0,1,0,0,0,0,1,0,0,0,0,0,1);
      }
      if(equal==4){
            if(x[i,6]>x[i,8] && x[i,6]>x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                        0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
            if(x[i,6]>x[i,8] && x[i,6]<x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,
                        0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
            if(x[i,6]<x[i,8] && x[i,6]<x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,
                        1,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
            if(x[i,6]==x[i,8] && x[i,6]<x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,
                        1,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
      }
}
if(cH+mH==0){
      cond.prob.tfpg<-cond.prob.tfmg<-c(1,0,0,0,1,0,0,0,1,
            x[i,9],x[i+1,9],1-(x[i,9]+x[i+1,9]),
            x[i,9],x[i+1,9],1-(x[i,9]+x[i+1,9]),
            x[i,9],x[i+1,9],1-(x[i,9]+x[i+1,9])));
      cond.prob.gmgt<-cond.prob.gfgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,
            0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,
            0,0,1,0,0,0,0,0,0,0,1);
      cond.prob.mgt<-c(1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,
            0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,
            0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);
      cond.prob.cmg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
            0.5,0.5,0,0,1,0,0,0.5,0.5,
            0.5,0,0.5,0,0.5,0.5,0,0,1);
      cond.prob.pfgt<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0.5,0,0,1,0,0,0.5,0.5,
            0.5,0,0.5,0,0.5,0.5,0,0,1);
      cond.prob.pfpg<-cond.prob.pfmg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0.5,0,0,
            1,0,0,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);

      if(length(allel_gf)==1 && allel_gf != 0){
            cond.prob.pfpg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfpg<-c(1,0,0,1,fr_allel_gf,1-fr_allel_gf,fr_allel_gf, 1-fr_allel_gf);
            cond.prob.gfgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      }
      if(length(allel_gm)==1 && allel_gm != 0){
            cond.prob.pfmg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfmg<-c(1,0,0,1,fr_allel_gm,1-fr_allel_gm,fr_allel_gm, 1-fr_allel_gm);
            cond.prob.gmgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
      }
      if(equal==3){
            cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,
                  0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,
                  0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
      }
      if(equal==4){
            if(x[i,6]<x[i,8] && x[i+1,6]==x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                        0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,
                        0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
            if(x[i,6]==x[i,8] && x[i+1,6]<x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                        1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,
                        0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
            if(x[i,6]<x[i,8] && x[i+1,6]>x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                        1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,
                        0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
            if(x[i,6]>x[i,8] && x[i+1,6]<x[i+1,8])
                  cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                        0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,
                        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
            if(x[i,6]<x[i,8] && x[i+1,6]<x[i+1,8])
                  ifelse(x[i+1,6]>x[i,8],cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,
                     0,0,0,0,1, 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,
                     0,0,0,1,0,0,0,0,0,0,0,0,0,1),
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,
                        0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                        0,0,0,0,0,0,1));
            if(x[i,6]>x[i,8] && x[i+1,6]>x[i+1,8])
```

```
                        ifelse(x[i,6]<x[i+1,8],cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,
                           0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
                           0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1),
                              cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,
                              0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,
                              0,0,0,0,0,0,1));
            }
            if(equal==5){
                  if(x[i,6]<x[i,8] && x[i+1,6]<x[i+1,8])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,
                              0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
                  if(x[i,6]==x[i,8] && x[i+1,6]>x[i+1,8])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
                  if(x[i,6]<x[i,8] && x[i+1,6]==x[i+1,8])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1);
                  if(x[i,6]==x[i,8] && x[i+1,6]<x[i+1,8])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
                  if(x[i,6]>x[i,8] && x[i+1,6]>x[i+1,8])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
                              0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
                  if(x[i,6]>x[i,8] && x[i+1,6]==x[i+1,8])
                        cond.prob.cgt<-c(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
                              0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,
                              1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1);
            }
      }

      ##criacao das tabelas de probabilidade de cada nodo
      #parametro alpha define a probabilidade a priori a favor da paternidade


            Ttfpf<-cptable(~"tf=pf?",values=c(alpha,1-alpha),levels=yn);
            Tpfpg<-cptable(~pfpg|gfpg:gfmg, values=cond.prob.pfpg, levels=gf_levels);
            Tpfmg<-cptable(~pfmg|gmpg:gmmg, values=cond.prob.pfmg, levels=gm_levels);
            Tmpg<-cptable(~mpg, values=allel.frq_m, levels=m_levels);
            Tmmg<-cptable(~mmg, values=allel.frq_m, levels=m_levels);
            Tmgt<-cptable(~mgt|mpg:mmg, values=cond.prob.mgt,levels=mgt_levels);
            Ttfpg<-cptable(~tfpg|pfpg:"tf=pf?", values=cond.prob.tfpg,levels=gf_levels);
            Ttfmg<-cptable(~tfmg|pfmg:"tf=pf?", values=cond.prob.tfmg,levels=gm_levels);
            Tcopg<-cptable(~copg|tfpg:tfmg,values=cond.prob.cpg,levels=copg_levels);
            Tcomg<-cptable(~comg|mpg:mmg,values=cond.prob.cmg,levels=comg_levels);
            Tcgt<-cptable(~cgt|capg:camg, values=cond.prob.cgt,levels=cgt_levels);

            Tcapg<-cptable(~capg|copg,values=qpf[[j]],levels=copg_levels)
            Tcamg<-cptable(~camg|comg,values=qm[[j]],levels=comg_levels)

            Tgfpg<-cptable(~gfpg, values=allel.frq_gf, levels=gf_levels);
            Tgfmg<-cptable(~gfmg, values=allel.frq_gf, levels=gf_levels);
            Tgmpg<-cptable(~gmpg, values=allel.frq_gm, levels=gm_levels);
            Tgmmg<-cptable(~gmmg, values=allel.frq_gm, levels=gm_levels);
            Tgfgt<-cptable(~gfgt|gfpg:gfmg, values=cond.prob.gfgt,levels=gfgt_levels);
            Tgmgt<-cptable(~gmgt|gmpg:gmmg, values=cond.prob.gmgt,levels=gmgt_levels);


            plist<-compileCPT(list(Ttfpf,Tpfpg,Tpfmg,Tmpg,Tmmg,Tcomg,Tmgt,Ttfpg,
                  Ttfmg,Tcopg,Tcgt,Tcapg,Tcamg,Tgfpg,Tgfmg,Tgmpg,Tgmmg,Tgfgt,Tgmgt));


      ##Criacao da RB do caso e consequente calculo do valor de LR

      net<-grain(plist);

      # este passo e necessario para definir informacao discordante observada como "x"

      s<-as.character(c(paste(x[i,2],x[i+1,2]),paste(x[i,4],x[i+1,4]),paste(x[i,6],x[i+1,6]),
         paste(x[i,8],x[i+1,8])));

      allelgf<-intersect(c(x[i,2],x[i+1,2]),c(x[i,8],x[i+1,8]));
      if(sum(s[1]==cgt_levels)==1 && length(allelgf)==1)
            s[1]<-paste(allelgf,"x");
      if(sum(s[1]==cgt_levels)<1)
            ifelse(length(allelgf)>=1,s[1]<-paste(allelgf,"x"),s[1]<-paste("x","x"));
      allelgm<-intersect(c(x[i,4],x[i+1,4]),c(x[i,8],x[i+1,8]));
      if(sum(s[2]==cgt_levels)==1 && length(allelgm)==1)
            s[2]<-paste(allelgm,"x");
      if(sum(s[2]==cgt_levels)<1)
            ifelse(length(allelgm)>=1,s[2]<-paste(allelgm,"x"),s[2]<-paste("x","x"));

      #propagacao da evidencia observada na rede
      bnetE<-setEvidence(net,nodes=c("gfgt","gmgt","mgt","cgt"),states=s)

      #calculo do valor de LR
      lr[j+1]<-querygrain(bnetE)$`"tf=pf?"`[1]/querygrain(bnetE)$`"tf=pf?"`[2];

} #fim do ciclo


#calculo do valor de LR global como o produto dos LR de cada um dos marcadores analisados
```

```
      lr<-as.numeric(lr);
      lhr<-prod(lr);
      output<-list(lr,lhr);

      #devolve como output os LR de cada marcador e o LR global
      return(output);

}

#executa a funcao LR_avos.
LR_avos(x,0.5)
```

Código de R para o Caso III

```
#Executar a biblioteca gRain

library(gRain)

##criar ficheiro com as frequencias alelicas para execucoes futuras

#lista de marcadores geneticos analisados
marker<-c("CSF1PO","CSF1PO","D2S1338","D2S1338","D3S1358","D3S1358","D5S818","D5S818",
"D7S820","D7S820","D8S1179","D8S1179","D13S317","D13S317","D16S539","D16S539",
"D18S51","D18S51","D19S433","D19S433","D21S11","D21S11","FGA","FGA","PentaD","PentaD",
"PentaE","PentaE","TH01","TH01","TPOX","TPOX","vWA","vWA")

#criacao da matrix de marcadores alelicos e respetivas frequencias
y<-matrix(0,nrow=28,ncol=34);
colnames(y)<-marker;

#insercao dos alelos de cada um dos marcadores
#CSF1PO
y[,1]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D2S1338
y[,3]<-c(13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D3S1358
y[,5]<-c(9,10,11,12,13,14,15,16,17,18,19,20,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D5S818
y[,7]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D7S820
y[,9]<-c(6,7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D8S1179
y[,11]<-c(8,9,10,11,12,13,14,15,16,17,18,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D13S317
y[,13]<-c(8,9,10,11,12,13,14,15,16,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D16S539
y[,15]<-c(7,8,9,10,11,12,13,14,15,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D18S51
y[,17]<-c(10,10.2,11,12,13,13.2,14,14.2,15,15.2,16,17,18,19,20,21,22,23,24,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D19S433
y[,19]<-c(10,11,11.2,12,12.2,13,13.2,14,14.2,15,15.2,16,16.2,17,17.2,18,18.2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#D21S11
y[,21]<-c(21,24,24.2,25,25.2,26,26.2,27,28,28.2,29,29.2,30,30.2,31,31.2,32,32.2,33,33.2,34,34.1,34.2,35,35.2,36,37,-2)
#FGA
y[,23]<-c(16,17,18,18.2,19,20,20.2,21,21.2,22,22.2,23,23.2,24,24.2,25,25.2,26,27,27.2,28,29,30.2,31.2,43.2,-1,-1,-2)
#PentaD
y[,25]<-c(2.2,3.2,5,6,7,8,9,10,11,12,13,14,15,16,17,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#PentaE
y[,27]<-c(5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,-1,-1,-1,-1,-1,-1,-2)
#TH01
y[,29]<-c(3,4,5,6,7,8,9,9.3,10,11,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#TPOX
y[,31]<-c(6,7,8,9,10,11,12,13,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)
#vWa
y[,33]<-c(11,13,14,15,16,17,18,19,20,21,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-2)

#insercao das frequencias de cada um dos alelos de
#cada um dos marcadores e respetivas frequencias minimas
#CSF1PO
y[,2]<-c(0.0064,0.0114,0.0185,0.2657,0.3155,0.3215,0.05235,0.0079,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0010)
#D2S1338
y[,4]<-c(0.0004,0.0007,0.0015,0.0501,0.2415,0.0723,0.1045,0.1418,0.0468,0.0409,
0.1034,0.1041,0.0779,0.0127,0.0011,0.0004,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D3S1358
y[,6]<-c(0.0004,0.0002,0.0006,0.0039,0.0057,0.1036,0.2589,0.2629,0.2007,0.1491,
0.0134,0.0007,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D5S818
y[,8]<-c(0.0004,0.0145,0.0376,0.0617,0.3349,0.3676,0.1724,0.0101,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D7S820
y[,10]<-c(0.0002,0.0204,0.1478,0.1183,0.2896,0.2266,0.1561,0.0353,0.0053,0.0006,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D8S1179
y[,12]<-c(0.0095,0.0101,0.0762,0.0846,0.1236,0.2901,0.2525,0.1230,0.0261,0.0035,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D13S317
y[,14]<-c(0.1340,0.0560,0.0485,0.3256,0.2793,0.1144,0.0406,0.0015,0.0002,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#D16S539
y[,16]<-c(0.0002,0.0211,0.1355,0.0602,0.3010,0.2745,0.1776,0.0292,0.0007,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
```

```
#D18S51
y[,18]<-c(0.0145,0.0006,0.0083,0.1328,0.1269,0.0009,0.1381,0.0002,0.1428,0.0002,0.1553,
0.1272,0.0709,0.0426,0.0220,0.0112,0.0024,0.0017,0.0006,-1,-1,-1,-1,-1,-1,-1,-1,0.0012)
#D19S433
y[,20]<-c(0.0029,0.0112,0.0002,0.1104,0.0050,0.2534,0.0132,0.3127,0.0307,0.1460,0.0455,
0.0466,0.0152,0.0031,0.0017,0.0006,0.0017,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#D21S11
y[,22]<-c(0.0002,0.0002,0.0024,0.0004,0.0009,0.0013,0.0004,0.0257,0.1577,0.0002,0.2264,0.0007,
0.2402,0.0354,0.0610,0.1087,0.0088,0.0883,0.0015,0.0305,0.0006,0.0002,0.0050,0.0024,0.0007,0.0002,0.0002,0.0011)
#FGA
y[,24]<-c(0.0004,0.0026,0.0156,0.0011,0.0676,0.1267,0.0002,0.1759,0.0018,0.1864,0.0059,0.1484,
0.0022,0.1357,0.0004,0.0817,0.0004,0.0347,0.0073,0.0002,0.0028,0.0013,0.0002,0.0006,0.0002,-1,-1,0.0011)
#PentaD
y[,26]<-c(0.0187,0.0015,0.0033,0.0007,0.0095,0.0206,0.1906,0.1144,0.1616,0.1855,0.1913,
0.0747,0.0202,0.0062,0.0011,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)
#PentaE
y[,28]<-c(0.0648,0.0013,0.1399,0.0288,0.0134,0.0854,0.1254,0.1968,0.1160,0.0628,0.0406,0.0380,
0.0382,0.0189,0.0134,0.0083,0.0037,0.0022,0.0015,0.0004,0.0002,-1,-1,-1,-1,-1,0.001)
#TH01
y[,30]<-c(0.0002,0.0002,0.0006,0.2000,0.1871,0.1449,0.1906,0.2642,0.0121,0.0002,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.001)
#TPOX
y[,32]<-c(0.0101,0.0042,0.4927,0.1065,0.0656,0.2870,0.0327,0.0013,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0009)
#vWa
y[,34]<-c(0.0011,0.0020,0.1096,0.1328,0.2255,0.2611,0.1752,0.0760,0.0141,0.0026,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.0011)

y<-data.frame(y) #criacao do dataframe com as frequencias alelicas da populacao

#obter a dimensao do ficheiro com as frequencias alelicas
n1<-dim(y)[1];n2<-dim(y)[2];

## marcadores observados num caso tipo (pai,mae e filho(a))

x<-read.csv("dados.csv", header= T ,sep = ";")
x[,2]<-as.character(x[,2])

#cria o vector com os alelos observados na mae do pressuposto pai
gm<-rep(0,34);

j<-0;
for(j in 0:17){
      p<-1;j<-j+1;
      for(p in 1:17){
            l<-p*2-1;
            if(x[j,2]==marker[l]){
                  gm[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,gm[l+1]<-x[j,3],gm[l+1]<-x[j,4]);
            }
      }
}

#cria o vector com os alelos observados na crianca
c<-rep(0,34);

j<-0;
for(j in 17:33){
      p<-1;j<-j+1;
      for(p in 18:34){
            l<-(p-17)*2-1;
            if(x[j,2]==marker[l]){
                  c[l]<-x[j,3];
                  ifelse(is.na(x[j,4])==TRUE,c[l+1]<-x[j,3],c[l+1]<-x[j,4]);
            }
      }
}

## este ciclo obtem as frequencias respectivas aos alelos observados
## no caso em estudo

fr_gm<-rep(0,n1);fr_c<-rep(0,n1);


i<-0;
for(i in 0:n2){
      k<-1;i<-i+1;
      for(k in 1:n1){
            ifelse(gm[i]==y[k,i],fr_gm[i]<-y[k,i+1],2)
            ifelse(gm[i+1]==y[k,i],fr_gm[i+1]<-y[k,i+1],2)
            ifelse(c[i]==y[k,i],fr_c[i]<-y[k,i+1],2)
            ifelse(c[i+1]==y[k,i],fr_c[i+1]<-y[k,i+1],2)
            ifelse(fr_gm[i]==0,fr_gm[i]<-y[28,i+1],2)
      }
}

## Dataframe final com os dados relevantes para o calculo do Likelyhood Ratio (LR)


x<-data.frame(marker,gm,fr_gm,c,fr_c);

## Criacao do modelo de mutacao

# Inserir os mutation rates conforme a ordem estipulada dos marcadores

m_rates<-c(0.0037,0.0025,0.0010,0.0013,0.0013,0.0010,0.0013,0.0020,0.0010,
0.0025,0.0029,0.0022,0.0014,0.0016,0.0004,0.0037,0.0022)
```

```
# Cria a matriz s

y1<-y[,seq(2, ncol(y), by = 2)]
y2<-y[,seq(1, ncol(y), by = 2)]

N<-length(y1)

s<-list();

i<-j<-k<-1
for(k in 1:N){
      k<-k*2-1;
      dim<-sum(as.numeric(y[,k]!=-1));
      s[[k]]<-matrix(0,nrow=dim,ncol=dim);
      for(i in 1:dim){
            sum<-0;
            for(j in i+1:dim-i){
                  s[[k]][i,j]<-0.5^abs(y[i,k]-y[j,k]);
                  s[[k]][j,i]<-s[[k]][i,j];
                  sum<-sum+s[[k]][i,j];
            }
            s[[k]][i,i]<--sum;
      }
}

s[sapply(s, is.null)] <- NULL #remove os NULL da lista

# Definicao do parametro lambda

lambda<-rep(0,N);i<-j<-0
for(i in 1:N){
      sum<-0;
      for(j in 1:dim(s[[i]])[1]){
            sum<-sum+s[[i]][j,j];
      }
      lambda[i]<-m_rates[i]/-sum;
}

## Criacao da matriz Q

Q<-list();

i<-j<-1
for(k in 1:N){
      dim<-sum(as.numeric(y1[,k]!=-1));
      names<-y2[,k][sapply(y2[,k], function(x) x!=-1)]
      Q[[k]]<-matrix(0,nrow=dim,ncol=dim,
            dimnames = list(as.character(names),
                        as.character(names)));

      for(i in 1:dim){
            j<-i+1;
            while(j<=dim){
                  Q[[k]][i,j]<-lambda[k]*s[[k]][i,j]/y1[i,k];
                  Q[[k]][j,i]<-y1[i,k]*Q[[k]][i,j]/y1[j,k];
                  j<-j+1;
            }
            Q[[k]][i,i]<-1+lambda[k]*s[[k]][i,i]/y1[i,k];
      }
}

# Criacao das matrizes de mutacao para os alelos provenientes do pai

q<-list();

i<-K<-J<-k<-j<-p<-1;
for(k in 1:N){
      i<-k*2-1;K<-1;
      mut.allel<-unique(c(x[i,4],x[i+1,4])); #guarda os alelos do pai
      mut.allel<-sort(mut.allel);
      allel<-vector();
      for(j in 1:(n1-1)){
            if(sum(ifelse(y[j,i]==mut.allel,s<-1,s<-0))==1){
                  allel[K]<-j; #guarda a posicao do alelo no dataframe das frequencias
                  K<-K+1;
            }
      }
      #define-se a dimensao das matrizes de mutacao

      dim<-length(allel)+1;

      #contingencia para o caso de se veririficar um alelo ausente da populacao
      if(length(allel)<length(mut.allel)){
            dim<-length(allel)+2;
            allel<-c(allel,28);
      }

      q[[k]]<-matrix(0,nrow=dim,ncol=dim,
            dimnames = list(as.character(c(mut.allel,"x")),
                        as.character(c(mut.allel,"x"))));
      #este ciclo preenche as matrizes de mutacao com as probabilidades de
      # transicao dos alelos para os seus vizinhos

      for(p in 1:dim-1){
            q[[k]][p,p]<-Q[[k]][allel[p],allel[p]]
            J<-p+1;
```

```
            while(J<dim){
                  q[[k]][p,J]<-Q[[k]][allel[p],allel[J]];
                  q[[k]][J,p]<-Q[[k]][allel[J],allel[p]];
                  J<-J+1;
            }
            q[[k]][p,dim]<-1-sum(q[[k]][p,]);
            q[[k]][dim,p]<-sum(Q[[k]][,allel[p]])-sum(q[[k]][,p]);
      }
      q[[k]][dim,dim]<-1-sum(q[[k]][dim,]);
}


## funcao que cria as redes para calcular o LR

LR_avo<-function(x,alpha){

      lr<-list();
      lr[1]<-1;
      i<-1;j<-0;

      #este ciclo define toda a estrutura da rede para um trio familiar
      #executa-se para criar uma RB para cada marcador e calcular o seu LR

      for(i in 1:(n2/2)){ #inicio do ciclo
            i<-i*2-1;j<-j+1;
            gmH<-0;cH<-0;dif<-0;allel<-0

            #determina se existe homozigotia
            if(identical(x[i,2],x[i+1,2])==TRUE)gmH<-1;
            if(identical(x[i,4],x[i+1,4])==TRUE)cH<-1;

            #verifica se existem alelos iguais e quantos sao
            dif<-length(unique(c(x[i,2],x[i+1,2],x[i,4],x[i+1,4])));
            allel<-intersect(c(x[i,2],x[i+1,2]),c(x[i,4],x[i+1,4]));
            eq<-setequal(c(x[i,2],x[i+1,2]),c(x[i,4],x[i+1,4]));
            fr_allel<-intersect(c(x[i,3],x[i+1,3]),c(x[i,5],x[i+1,5]));

            #definicao dos niveis de cada nodo da rede

            mut_levels<-unique(c(x[i,4],x[i+1,4]));
            mut_levels<-sort(mut_levels);
            mut_levels<-c(mut_levels,"x");

            yn<-c("yes","no");
            gm_values<-unique(c(x[i,3],x[i+1,3],1-sum(unique(x[i:(i+1),3]))));

            allel.frq_c<-unique(c(x[i,5],x[i+1,5],1-sum(unique(x[i:(i+1),5]))));

            gm_levels<-sort(unique(c(x[i,2],x[i+1,2],x[i,4],x[i+1,4],"x")));
            c_levels<-as.character(unique(c(x[i,4],x[i+1,4],"x")));

            if(gmH==0){
                  ifelse(dif!=4,gm_levels<-c(allel,"x"),remove<-unique(c(x[i,2],x[i+1,2])));
            } else ifelse(dif!=3,gm_levels<-paste("x","x"),remove<-unique(c(x[i,2],x[i+1,2])));

            gm_levels<-gm_levels[! gm_levels %in% remove]

            if(dif==2 && cH<=gmH) gm_levels<-c_levels;

            ifelse(sum(allel)==0 || eq==TRUE,gm_levels<-c_levels,gm_levels<-c(allel,"x"));


            copg_levels<-unique(c(x[i,4],x[i+1,4]));

            c_lvls<-c(copg_levels,copg_levels)
            c_lvls<-sort(c_lvls);c_lvls<-c(as.character(c_lvls),"x")
            cgt_levels<-vector();

            k1<-k2<-p<-1;
            for(k1 in 1:(length(c_lvls)-1)){
                  for(k2 in (k1+1):length(c_lvls)){
                        cgt_levels[p]<-paste(c_lvls[k1],c_lvls[k2]);
                        p<-p+1;
                  }
            }
            cgt_levels<-unique(cgt_levels);cgt_levels<-c(cgt_levels,paste("x","x"));
            copg_levels<-c(copg_levels,"x")

            ifelse(dif>=2 && gmH+cH>=1,gmgt_levels<-cgt_levels,
                  ifelse(dif==3 && gmH+cH==0,gmgt_levels<-c(paste(allel,allel),paste(allel,"x"),paste("x","x")),
                              gmgt_levels<-cgt_levels));
            if(dif==2 && gmH==1 && cH==0) gmgt_levels<-c(paste(allel,allel),paste(allel,"x"),paste("x","x"));

            #mediante as possiveis combinacoes alelicas dos intervenientes sao criados
            #os estados possiveis para cada tabela de probabilidade de cada nodo

            if(gmH+cH==2){
                  cond.prob.gmgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
                  cond.prob.tfpg<-c(1,0,0,1,unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])),
                        unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])));
                  cond.prob.tfmg<-c(1,0,0,1,unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])),
                        unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])));

                  cond.prob.gm<-c(1,0,0.5,0.5,0.5,0.5,0,1);
                  cond.prob.cgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);

                  ifelse(dif==1,cond.prob.copg<-c(1,0,0.5,0.5,0.5,0.5,0,1),
```

```
         cond.prob.copg<-c(0.5,0,0,0.5,0.5,0.5,0,1));
}
if(gmH==0 && cH==1){
      cond.prob.gmgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
            0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,
            0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1);

      cond.prob.tfpg<-c(1,0,0,1,unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])),
            unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])));

      cond.prob.tfmg<-c(1,0,0,0,0,1,0,0,0,1,unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])),
            unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])),
            unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])));

      cond.prob.gm<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
            0.5,0.5,0,0,1,0,0,0.5,0.5,
            0.5,0,0.5,0,0.5,0.5,0,0,1);

      cond.prob.cgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);

      if(dif==2){
            cond.prob.copg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.gmgt<-c(1,0,0,0,0,1,0,0,1,0,0,0,0,1);
            gm_values<-c(fr_allel,1-sum(fr_allel));
            cond.prob.gm<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfmg<-c(1,0,0,1,fr_allel,1-sum(fr_allel),fr_allel,1-sum(fr_allel));
      }
      if(dif==3)
            cond.prob.copg<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.gmgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
            gm_values<-allel.frq_c;
            cond.prob.gm<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfmg<-cond.prob.tfpg
}
if(gmH==1 && cH==0){
      cond.prob.gmgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);

      cond.prob.tfpg<-c(1,0,0,0,1,0,0,0,1,unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])),
            unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])),unique(x[i:(i+1),5]),
               1-sum(unique(x[i:(i+1),5])));

      cond.prob.tfmg<-c(1,0,0,1,unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])),
            unique(x[i:(i+1),3]),1-sum(unique(x[i:(i+1),3])));

      cond.prob.gm<-c(1,0,0.5,0.5,0.5,0.5,0,1);

      cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1,
            0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1);

      if(dif==2){
            if(x[i,2]==x[i,4])
                  cond.prob.copg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);
            if(x[i,2]==x[i+1,4])
                  cond.prob.copg<-c(0.5,0.5,0,0,1,0,0,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);
      }
      if(dif==3){
            cond.prob.copg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0.5,0,0,1,0,
               0,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);
            cond.prob.gm<-cond.prob.copg;
            cond.prob.gmgt<-cond.prob.cgt;
            cond.prob.tfmg<-cond.prob.tfpg;
            gm_values<-allel.frq_c;

      }

}
if(gmH+cH==0){
      cond.prob.tfpg<-c(1,0,0,0,1,0,0,0,1,unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])),
            unique(x[i:(i+1),5]),1-sum(unique(x[i:(i+1),5])),unique(x[i:(i+1),5]),
               1-sum(unique(x[i:(i+1),5])));

      cond.prob.tfmg<-c(1,0,0,0,1,0,0,0,1,x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),
            x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]),x[i,3],x[i+1,3],1-(x[i,3]+x[i+1,3]));
      cond.prob.gmgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
            0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,
            0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1);

      cond.prob.gm<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,
            0.5,0.5,0,0,1,0,0,0.5,0.5,
            0.5,0,0.5,0,0.5,0.5,0,0,1);
      cond.prob.cgt<-c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
            0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,
            0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1);
      if(dif==2){
            cond.prob.copg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0.5,0,0,1,0,
                  0,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);
      }
      if(dif==3){
            gm_values<-c(fr_allel,1-sum(fr_allel));
            cond.prob.gmgt<-c(1,0,0,0,1,0,0,1,0,0,0,1);
            cond.prob.gm<-c(1,0,0.5,0.5,0.5,0.5,0,1);
            cond.prob.tfmg<-c(1,0,0,1,fr_allel,1-sum(fr_allel),fr_allel,1-sum(fr_allel));

            if(x[i,2]==x[i,4])
                  cond.prob.copg<-c(1,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0,0.5,0,
                        0.5,0.5,0,0,1);
            if(x[i,2]==x[i+1,4])
```

```
                        cond.prob.copg<-c(0.5,0.5,0,0,1,0,0,0.5,0.5,0.5,0,0.5,0,
                              0.5,0.5,0,0,1);
                  if(x[i+1,2]==x[i,4])
                        cond.prob.copg<-c(0.5,0,0,0,0.5,0,0,0,0.5,1,0,0,0.5,0.5,0,0.5,
                              0,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);
                  if(x[i+1,2]==x[i+1,4])
                              cond.prob.copg<-c(0.5,0.5,0,0,1,0,0,0.5,0.5,0.5,0,0.5,0,
                              0.5,0.5,0,0,1);
            }
            if(dif==4)
                  cond.prob.copg<-c(0.5,0,0,0,0.5,0,0,0,0.5,0.5,0,0,
                        0,0.5,0,0,0,0.5,0.5,0,0.5,0,0.5,0.5,0,0,1);


      }

      ##criacao das tabelas de probabilidade de cada nodo
      #parametro alpha define a probabilidade a priori a favor da paternidade

            Ttfpf<-cptable(~tfpf,values=c(alpha,1-alpha),levels=yn);
            Tgmpg<-cptable(~gmpg, values=gm_values, levels=gm_levels);
            Tgmmg<-cptable(~gmmg, values=gm_values, levels=gm_levels);
            Tgmgt<-cptable(~gmgt|gmpg:gmmg, values=cond.prob.gmgt,levels=gmgt_levels);
            Tpfpg<-cptable(~pfpg, values=allel.frq_c, levels=c_levels);
            Tpfmg<-cptable(~pfmg|gmpg:gmmg, values=cond.prob.gm,levels=gm_levels);

            Ttfpg<-cptable(~tfpg|pfpg:tfpf, values=cond.prob.tfpg,levels=c_levels);
            Ttfmg<-cptable(~tfmg|pfmg:tfpf, values=cond.prob.tfmg,levels=gm_levels);
            Tcopg<-cptable(~copg|tfpg:tfmg,values=cond.prob.copg,levels=copg_levels);
            Tcgt<-cptable(~cgt|capg:camg, values=cond.prob.cgt,levels=cgt_levels);

            Tcapg<-cptable(~capg|copg,values=q[[j]],levels=mut_levels)
            Tcamg<-cptable(~camg,allel.frq_c,levels=c_levels)

            plist<-compileCPT(list(Tgmgt,Ttfpf,Tgmpg,Tgmmg,Tpfpg,Tpfmg,Ttfpg,
                  Ttfmg,Tcopg,Tcgt,Tcapg,Tcamg));

      ##Criacao da RB do caso e consequente calculo do valor de LR

      net<-grain(plist);

      # este passo e necessario para definir informacao discordante observada como "x"

      s<-as.character(c(paste(x[i,2],x[i+1,2]),paste(x[i,4],x[i+1,4])));
      if(sum(s[1]==cgt_levels)<1)
            ifelse(sum(allel)>=1,s[1]<-paste(allel,"x"),s[1]<-paste("x","x"));

      #propagacao da evidencia observada na rede
      bnetE<-setEvidence(net,nodes=c("gmgt","cgt"),states=s)

      #calculo do valor de LR
      lr[j+1]<-querygrain(bnetE)$tfpf[1]/querygrain(bnetE)$tfpf[2]

} #fim do ciclo

#calculo do valor de LR global como o produto dos LR de cada um dos marcadores analisados

lr<-as.numeric(lr);
lhr<-prod(lr);
output<-list(lr,lhr);

#devolve como output os LR de cada marcador e o LR global
return(output);

}

#executa a funcao LR_avo.
LR_avo(x,0.5)
```