

Estratégia dinâmica para determinação de grau de coancestralidade em um banco de dados orientado a grafos*

Eric Dias da Silva Rosso¹, Anderson dos Santos da Rosa², Ana Paula Lüdtke Ferreira¹

¹Programa de Pós-graduação em Computação Aplicada
Universidade Federal do Pampa, Campus Bagé

{ericrosso.aluno, anaferreira}@unipampa.edu.br

²Bacharelado em Ciência da Computação
Universidade Federal do Pampa, Campus Alegrete

andersonrosa.aluno@unipampa.edu.br

Resumo. *Sistemas de melhoramento animal controlam as relações de parentesco do rebanho para evitar os problemas decorrentes de depressão por consanguinidade. Este trabalho apresenta um algoritmo para cálculo da coancestralidade entre animais, implementado sobre um banco de dados de grafos e usando uma estratégia de programação dinâmica. O algoritmo tem complexidade quadrática, que é a complexidade mínima esperada de um algoritmo que computa uma relação binária total.*

Abstract. *Animal breeding systems control the kinship relationships between the animals in a herd, to prevent the problems arising from inbreeding depression. This paper presents an algorithm for calculating coancestry indexes between animals, implemented on a graph database, and using a dynamic programming strategy. The algorithm does not require to recalculate any values when new animals are inserted into the database. The algorithm is $O(n^2)$, which is the minimum expected complexity of an algorithm that computes a full binary relation.*

1. Introdução

Os processos de melhoramento animal visam aprimorar genótipos e fenótipos, eliminando as características indesejáveis dos rebanhos. O melhoramento seleciona os melhores indivíduos para se tornarem pais da próxima geração e aplica estratégias de acasalamento para maximizar o potencial da futura geração [Falconer and Mackay 1996].

A estratégia de fazer com que os melhores reprodutores (machos) acasalem com as melhores matrizes (fêmeas) resulta em animais de qualidade excepcional [Ferreira et al. 2021], mas tende a fazer com que a endogamia aumente, criando problemas como a depressão por consanguinidade, que faz com que as taxas de sobrevivência e fertilidade da prole diminuam, comprometendo a saúde e o bem-estar dos animais, além de gerar perdas financeiras para o produtor [Mi et al. 1965, Charlesworth and Willis 2009]. A solução deste problema requer estratégias para acasalamentos corretivos, para maximizar o ganho genético/econômico, mantendo os níveis de consanguinidade controlados [Cardoso 2009].

*Trabalho concluído, com autores discentes de graduação e de pós-graduação.

A construção de sistemas de acasalamento que maximizem o ganho genético e econômico, favorecendo atributos relacionados ao valor, à saúde e ao bem-estar dos animais, mas mantendo os valores de endogamia baixos, necessita dos índices de parentesco entre os animais do rebanho. O índice de parentesco, ou *coancestralidade*, é uma relação binária que expressa o valor esperado de similaridade genética entre dois animais. Note-se que somente um mapeamento genético completo pode determinar esse valor com exatidão, mas pode-se estimá-lo a partir do conhecimento sobre processos de reprodução.

O cálculo desse índice é computacionalmente custoso, pois rebanhos de dezenas de milhares de animais não são incomuns. Como a relação é binária, não se espera que um algoritmo com complexidade menor que $O(n^2)$ possa resolver o problema para rebanhos com n animais. Nesses casos, ainda que a complexidade seja polinomial, o tempo de execução dos algoritmos é grande e seu impacto é maior se precisarem ser executados várias vezes [Toscani and Veloso 2006].

A relação de parentesco entre dois animais não muda ao longo da vida, precisando ser computada somente uma vez. Grafos são estruturas que representam visualmente relações binárias [Diestel 2005] e recentemente começaram a ser usados para representação de dados relacionais na forma de sistemas gerenciadores de bancos de dados [Robinson et al. 2015]. Nesse tipo de banco de dados, os animais podem ser representados por nodos, e as relações de parentesco (rotuladas com seu valor) são representadas por arestas. Essa abordagem permite armazenar de forma flexível as conexões entre os animais, facilitando consultas e servindo de base para algoritmos que visam controlar a consanguinidade do rebanho.

O objetivo deste trabalho é apresentar um algoritmo de cálculo de valores de coancestralidade entre os indivíduos de um rebanho, que usa uma estratégia de programação dinâmica e é implementado no próprio banco de dados, permitindo que o cálculo somente precise ser realizado para as novas gerações, de forma incremental. O algoritmo recebe uma planilha com os dados dos animais a inserir no banco, que contém os registros de seu pai e de sua mãe, sexo e data de nascimento. Os resultados da inserção podem ser verificados no banco por meio das ferramentas de visualização disponibilizadas.

O restante do texto está estruturado como se segue: a Seção 2 discute os trabalhos relacionados encontrados na literatura, a Seção 3 apresenta as definições necessárias para a implementação do algoritmo, a Seção 4 mostra a construção da solução do problema, a Seção 5 discute os resultados e a Seção 6 finaliza o trabalho, apontando as contribuições e trabalhos futuros.

2. Trabalhos relacionados

Os trabalhos relacionados foram obtidos por meio de um protocolo de revisão de escopo da literatura [Arksey and O'Malley 2005], buscando identificar os principais métodos e algoritmos utilizados no cálculo do coeficiente de parentesco em rebanhos. Os artigos que tratavam de análises genômicas ou outros cálculos que não fossem de relações de parentesco foram excluídos dos achados. Não houve restrição temporal nos critérios de aceitação. A seguir, os achados são discutidos e comparados com a abordagem apresentada neste artigo.

[Boyce 1983] apresenta uma revisão de todos os algoritmos para cálculo de coeficientes de coancestralidade existentes à época. Não existe referência à complexidade

dos algoritmos analisados, mas a maioria usa estratégias de busca e comparação de linhas ancestrais por pares de animais, buscando encontrar o ancestral comum mais próximo, gerando algoritmos exponenciais ou, minimamente, cúbicos por dependerem do número de gerações presentes. Algoritmos chamados “iterativos” usam uma estratégia similar à programação dinâmica, com cálculo incremental dos valores dos animais mais velhos para os mais novos. É interessante notar que algoritmos quadráticos, à época, eram considerados inviáveis por limitações de memória dos computadores.

Algoritmos propostos após o trabalho de [Boyce 1983] seguem as linhas de trabalho lá descritas. [Khang 1989] usa uma pilha para implementar um procedimento recursivo de geração de linhas ancestrais. A complexidade não é analisada, mas parece ser exponencial no número de gerações analisadas, além de exigir que todos os pares sejam calculados sem referência a cálculos já efetuados. [Gholami and Thomas 1994] usa um procedimento que otimiza alguns aspectos do algoritmo de comparação de linhas ancestrais, mas mantém a exponencialidade no número de gerações, não sendo linear nas variáveis do problema como afirmam os autores. [Aguilar and Misztal 2008] traz uma abordagem que estima a endogamia de animais com pais desconhecidos, ainda com uma abordagem exponencial relacionada ao número de gerações da base. [Elliott et al. 2009] usa uma estrutura familiar para facilitar o cálculo de coeficientes de parentesco cuja complexidade é a multiplicação do quadrado do número de animais pela distância ao antepassado mais distante pela quantidade de antecessores dos pais do animais.

[Backus and Gilpin 2002] usa uma abordagem de registro de valores de parentesco em uma matriz, em uma abordagem similar à dinâmica, tirando partido das propriedades dos vetores genéricos da linguagem Java (que permitem retirada de linhas e colunas por primitivas do pacote Array) para tratar do caso dos animais que não pertencem mais ao rebanho atual.

Nosso trabalho difere dos demais em dois pontos importantes. A primeira diferença é a implementação do algoritmo em um banco de dados que não requer o recálculo de toda a matriz a cada nova geração inserida no rebanho. A implementação feita possibilita a retirada de animais mortos ou não mais usados no processo de reprodução da forma correta, pelas próprias primitivas do banco: a eliminação de nodos do grafo automaticamente elimina as arestas incidentes aos nodos eliminados. Note-se que a informação sobre coancestralidade não é perdida, visto que os animais remanescentes já tem seus índices com os demais animais calculados.

O segundo ponto é a garantia de um algoritmo polinomial eficiente para o cálculo dos índices, uma vez que a estratégia dinâmica elimina o recálculo de coeficientes, como é feito na maioria das abordagens listadas, gerando resultados exponenciais em um rebanho com linhas genealógicas com muitos ancestrais e com baixa consanguinidade entre os animais de linhas diferentes.

3. Fundamentos

A abordagem formal para resolução de problemas requer definições precisas dos termos sobre os quais se deseja trabalhar. Nesta seção definimos formalmente os conceitos relacionados aos objetivos do trabalho.

Definição 1 (Rebanho) *Um rebanho é uma tupla $H = (A = S \cup D, s, d : A \rightarrow (A \cup \{\perp\}))$ onde A é um conjunto finito de animais particionado em um conjunto de*

reprodutores (machos) S e um conjunto de matrizes (fêmeas) D , em que cada animal $a \in A$ tem um pai $s(a)$ e uma mãe $d(a)$. Se o progenitor do animal a não pertence ao rebanho, o resultado da função é \perp .

A Definição 1 expressa formalmente o significado do termo “rebanho”: uma coleção de animais e duas funções que indicam, para cada animal, quem são os seus progenitores. Para tornar as funções totais, foi criado o símbolo \perp , que indica que o progenitor do animal não se encontra entre os animais do rebanho. Essa situação pode ocorrer tanto com animais adquiridos de outros produtores ou produzidos por meio de inseminação artificial com sêmen de origem externa.

O primeiro passo para o cálculo do índice de coancestralidade é a atribuição de um valor de *geração* a cada animal. Aqui, a ideia de geração é determinar uma partição no conjunto total de animais, permitindo que eles possam ser estruturados de acordo com o número de cruzamentos existentes dentro do rebanho que foi necessário para o animal ser concebido. A Definição 2 expressa essa ideia formalmente.

Definição 2 (Geração) *Seja $H = (A = S \cup D, s, d : A \rightarrow (A \cup \{\perp\}))$ um rebanho. A geração de cada animal $a \in A$ é dada pela função $g : A \rightarrow \mathbb{N}$ definida como se segue:*

$$g(a) = \begin{cases} 0 & s(a) = \perp \wedge d(a) = \perp \\ g(s(a)) + 1 & s(a) \neq \perp \wedge d(a) = \perp \\ g(d(a)) + 1 & s(a) = \perp \wedge d(a) \neq \perp \\ \max\{g(s(a)), g(d(a))\} + 1 & s(a) \neq \perp \wedge d(a) \neq \perp \end{cases} \quad (1)$$

A geração de um animal é o valor seguinte ao maior valor de geração dos pais do animal. A função g determina uma partição no conjunto dos animais, em que cada classe de equivalência é um valor geracional. A Fig. 1 mostra um exemplo de rebanho com 10 animais, referenciados por a_1, a_2, \dots, a_{10} . Para facilitar a visualização, nodos azuis e rosa representam machos e fêmeas, respectivamente. As relações de parentesco diretas induzem um grafo dirigido acíclico (*directed acyclic graph* ou DAG), visto que o fecho transitivo e reflexivo da relação de ancestralidade é uma relação antissimétrica. O valor entre parênteses denota a geração de cada animal, conforme a Definição 2. As arestas representam a relações diretas de parentesco, dadas pelas funções s e d da Definição 1.

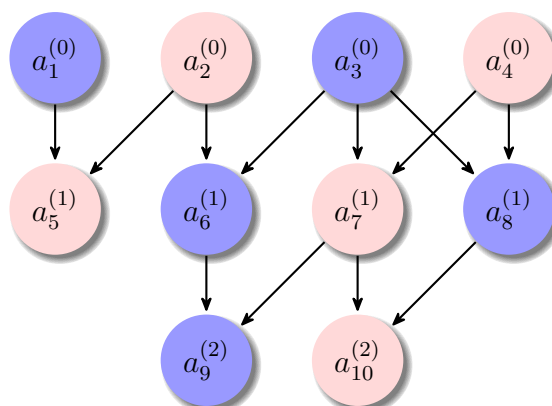


Figura 1. DAG de estrutura do rebanho – exemplo

O índice de coancestralidade indica o grau com que dois animais são aparentados e é uma estimativa do grau de similaridade genética entre dois animais e da quantidades

de cruzamentos consanguíneos em sua linha ancestral. Assim, um animal tem 100% de similaridade genética consigo mesmo e 50% com cada um de seus pais, se eles não forem aparentados; se eles forem, há um acréscimo relacionado ao índice de coancestralidade entre o pai e a mãe do animal. A ideia é mantida para qualquer outro animal do rebanho, com os índices dos ancestrais aparentados somando, mas diminuindo à medida que a distância geracional aumenta. A Definição 3 formaliza essa ideia.

Definição 3 (Coancestralidade) *Seja $H = (A = S \cup D, s, d : A \rightarrow (A \cup \{\perp\}))$ um rebanho. A relação de coancestralidade entre quaisquer dois animais $a, a' \in A$ é dada pela função $c : A \times A \rightarrow \mathbb{R}$ definida como se segue. Sem perda de generalidade, assume-se que $g(a) \geq g(a')$. Se $g(a) < g(a')$, então $c(a', a) = c(a, a')$.*

$$c(a, a') = \begin{cases} 1 & a = a' \\ 1/2 + c(s(a), d(a)) & s(a) = a' \vee d(a) = a' \\ 1/2(c(s(a), a') + c(d(a), a')) & \text{caso contrário} \end{cases} \quad (2)$$

O cálculo do coeficiente de coancestralidade entre os animais do rebanho da Fig. 1 é apresentado na Fig. 2. Note-se que animais fortemente aparentados têm um índice de coancestralidade alto, enquanto que animais sem antepassados comuns têm coancestralidade igual a zero. O cálculo de coancestralidade varia nos trabalhos encontrados na literatura e o algoritmo dinâmico proposto neste trabalho é independente do cálculo específico.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
a_1	1	0	0	0	0,5	0	0	0	0	0
a_2	0	1	0	0	0,5	0,5	0	0	0,25	0
a_3	0	0	1	0	0	0,5	0,5	0,5	0,5	0,5
a_4	0	0	0	1	0	0	0,5	0,5	0,25	0,5
a_5	0,5	0,5	0	0	1	0,25	0	0	0,125	0
a_6	0	0,5	0,5	0	0,25	1	0,25	0,25	0,75	0,25
a_7	0	0	0,5	0,5	0	0,25	1	0,5	0,75	1
a_8	0	0	0,5	0,5	0	0,25	0,5	1	0,375	1
a_9	0	0,25	0,5	0,25	0,125	0,75	0,75	0,375	1	1,125
a_{10}	0	0	0,5	0,5	0	0,25	1	1	1,125	1

Figura 2. Matriz de coancestralidade – exemplo

4. Material e métodos

Os dados da genealogia dos animais foram implementados na versão gratuita do banco de dados Neo4J v. 5.11.0, que é uma base de dados de grafos com todas as propriedades ACID (*atomicidade, consistência, isolamento e durabilidade*). Para manipulação e processamento dos dados do banco de dados Neo4j, utilizamos a linguagem de programação Python na versão 3.11.0.

A estrutura de nodos e as relações (restrita ao problema tratado) do banco são apresentadas na Fig. 3. Os animais e os valores das gerações são os nodos do grafo e as relações de coancestralidade (rotuladas pelo respectivo valor) e de geração dos animais são representadas pelas arestas. O modelo apresentado funciona como um grafo-tipo [Heckel et al. 1996], tipando os nodos e relações dos valores inseridos no banco.

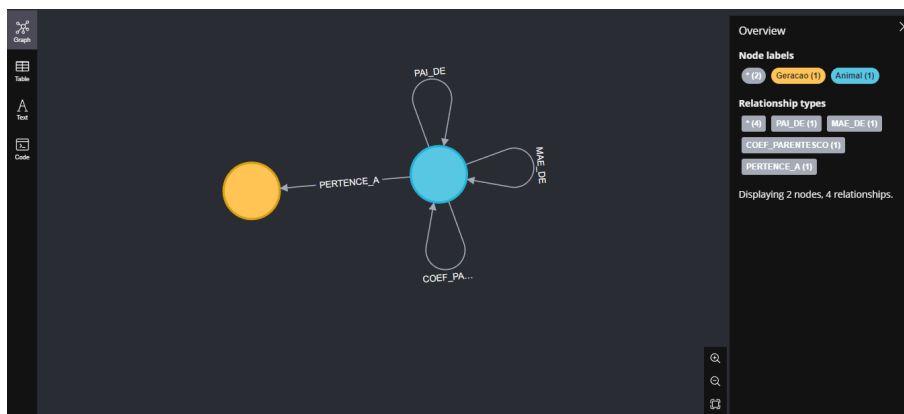


Figura 3. Modelo de dados e relações no banco Neo4J

O cálculo da geração de cada animal usa um procedimento recursivo que leva em conta se a geração dos pais já foi calculada. Os animais da geração 0 têm seus valores inicializados diretamente para garantir a base da recursão, bastando para isso verificar quais são os animais que não possuem os progenitores na base. Para fins de ilustração, a consulta que verifica quais são os animais que não tem progenitores registrados é mostrada a seguir. A consulta é feita na linguagem Cypher, que é a linguagem de consulta do banco Neo4j:

```
MATCH (a:Animal)
WHERE NOT (a)-[:PAI_DE]-() AND NOT (a)-[:MAE_DE]-()
MERGE (g:Geracao {Numero: 0})
CREATE (a)-[:PERTENCE_A]->(g);
```

A consulta busca todos os nodos do tipo animal (MATCH (a:Animal)) em que não existam arcos do tipo PAI_DE ou MAE_DE de outro animal qualquer (denotado pelos parênteses que abrem e fecham sem argumentos) para ele. Nesse caso, é criado um arco do animal para o nodo que representa a geração 0. O comando MERGE é usado em vez do comando CREATE para que não haja duplicação de nodos, visto que a versão gratuita do Neo4j não permite definir elementos que sejam únicos. O arco entre o animal e a geração 0 (zero) é criado (com o comando CREATE) na base, rotulado com o seu tipo PERTENCE_A.

O procedimento recursivo também usa uma estratégia dinâmica, de forma que o valor geracional de cada animal só é calculado uma vez. O algoritmo foi implementado em Python, com chamadas para busca e inserção diretamente no banco. Inicialmente, os animais da geração 0 (zero) são inicializados. A seguir, o algoritmo é chamado para os restantes animais do banco. A ordem não importa, visto que o algoritmo calcula a geração dos ancestrais dos animais, se não estiverem disponíveis. O pseudo-código do trecho do algoritmo chamado para cada $a \in A$ é apresentado a seguir. Primeiro é verificado se o arco do animal para a sua geração já existe. Se não existe, é verificado se os arcos geracionais dos pais existem: se existem o valor é retornado, se não, a função é chamada recursivamente. O teste de existência garante um único cálculo dos valores.

CALCULOGERAÇÃO($a \in A, A, \{s, d\} : A \rightarrow A, g : A \rightarrow \mathbb{N}$) retorna k

- (1) se ($g(a) = null$)
- (2) então

- (3) se $(g(s(a)) = null)$
- (4) então $g_p \leftarrow \text{CalculoGeração}(s(a), A, s, d, g)$
- (5) senão $g_p \leftarrow g(s(a))$
- (6) se $(g(d(a)) = null)$
- (7) então $g_m \leftarrow \text{CalculoGeração}(d(a), A, s, d, g)$
- (8) senão $g_m \leftarrow g(s(a))$
- (9) criar aresta $g(a) = \max\{g_p, g_m\} + 1$
- (10) retornar $g(a)$

O cálculo dos coeficientes de coancestralidade exige que as gerações já estejam calculadas. O texto restante vai exemplificar a solução usando matrizes, mas todas as operações são feitas diretamente no banco, com auxílio de programação em Python para a implementação de ciclos iterativos.

Seja então $H = (A = S \cup D, s, d : A \rightarrow (A \cup \{\perp\}))$ um rebanho, seja $K \in \mathbb{N}$ o número de gerações de animais nascidos no rebanho e seja \bar{A} uma permutação dos elementos de A ordenados por geração. Seja C a matriz de dimensões $n \times n$, em que $n = |A|$ é o número de elementos do conjunto A , e seja $n_i, 0 \leq i \leq K$ o número de animais pertencentes à geração i , ou o tamanho do conjunto $\{a \in A \mid g(a) = i\}$.

Os índices dos elementos da matriz de coancestralidade C são construídos com a ordenação \bar{A} , de forma que para quaisquer dois índices $i, j \in \{1, \dots, n\}$, se $i < j$ então $g(i) \leq g(j)$. Se o animal i é o l -ésimo animal pertencente à geração $k, k = 0, \dots, K$, de acordo com a ordenação \bar{A} , então $i = l + \sum_{s=1}^k n_{s-1}$. Ou seja, os animais da geração 0 começam no índice 1, os animais da geração 1 começam no índice $n_0 + 1$, os animais da geração 2 começam no índice $n_0 + n_1 + 1$, e assim sucessivamente.

O método usado para o cálculo da coancestralidade entre todos os animais do rebanho usa a Eq. (2), em uma estratégia de programação dinâmica [Cormen et al. 2002]. O preenchimento da matriz é feito por geração e, dentro de cada geração, por etapas. As cores usadas na matriz da Fig. 2 ajudam a exemplificar a ordem de preenchimento. Como os animais estão ordenados por gerações, os pais sempre aparecem antes dos filhos na matriz. A ordem entre animais da mesma geração é arbitrária, e o preenchimento é feito de forma a que sempre que um valor de coancestralidade seja necessário, ele já tenha sido calculado.

A coancestralidade entre animais diferentes pertencentes à primeira geração do rebanho é sempre igual a zero, obedecendo à suposição de que eles não são aparentados. Aqui, estamos assumindo que a geração 0 representa os animais que não têm ascendentes (pai e mãe) dentro do rebanho. Assim, a matriz superior esquerda da matriz C é uma matriz identidade, em que todos os seus elementos são iguais a zero, com exceção dos elementos da diagonal principal, que são iguais a 1 (visto que a coancestralidade de um animal consigo mesmo é 1. Essa suposição não apresenta perda de generalidade pois, se os animais forem aparentados, basta inicializar a matriz inicial com os valores corretos correspondentes. Na Fig. 2, os índices de coancestralidade da geração 0 estão apresentados **nesta cor**, os da geração 1 **nesta** e os da geração 2 **nesta**.

Os animais pertencentes à geração 1 são, necessariamente, filhos de dois animais pertencentes à geração 0 (ver Eq. (1)). Os animais da geração 2 têm pelo menos um dos pais na geração 1, com o outro podendo ser da geração 1 ou de gerações anteriores, e

assim sucessivamente. O preenchimento da matriz é descrito no pseudo-código abaixo, assumindo C a matriz em preenchimento, o número total de gerações K e os valores n_0, n_1, \dots, n_K para o número total de animais em cada geração e que os elementos da submatriz $C_{1, \dots, n_0 \times 1, \dots, n_0}$ (i.e., os valores da geração 0) já estão preenchidos:

CALCULO COANCESTRALIDADE($\bar{A}, s, d : \bar{A} \rightarrow \bar{A}, K, C$)

- (1) para cada $k = 1, \dots, K$ fazer
- (2) para cada $i \leftarrow (1 + \sum_{l=1}^{k-1} n_l), \dots, (n_k + \sum_{l=1}^{k-1} n_l)$ fazer
- (3) para cada $j = 1, \dots, i$ fazer
- (4) se $(i = j)$ então $C_{ij} \leftarrow 1$
- (5) senão
- (6) se $(j = s(i)$ ou $j = d(i))$ então $C_{ij} \leftarrow 1/2 + C_{s(i)d(i)}$
- (7) senão $C_{ij} \leftarrow 1/2(C_{s(i),j} + C_{d(i),j})$
- (8) $C_{ji} \leftarrow C_{ij}$

Note que a matriz é simétrica em relação à diagonal principal, visto que coancestralidade é uma relação simétrica.

5. Resultados e discussão

Nesta seção, apresentamos os resultados obtidos com a implementação dos algoritmos descritos no banco de dados Neo4j. A base utilizada foi cedida pela Embrapa Pecuária Sul e contém 1729 animais da raça Brangus, entre machos e fêmeas, com diversas relações de parentesco entre eles, totalizando 6 gerações.

A Fig. 4 apresenta uma amostra das relações existentes no banco Neo4j, demonstrando como as principais relações são organizadas. Os nós do tipo Animal (nodos azuis) estão conectados por meio das relações PAI_DE e MAE_DE, que indicam a relação de filiação entre os animais. A relação PERTENCE_A liga os nodos referentes aos animais às suas respectivas gerações (nodos amarelos). A relação COEF_PARENTESCO é usada para armazenar informações sobre os coeficientes de coancestralidade calculados entre os animais. Essa representação visual ajuda a compreender a estrutura do banco de dados e como as informações genealógicas são armazenadas e recuperadas.

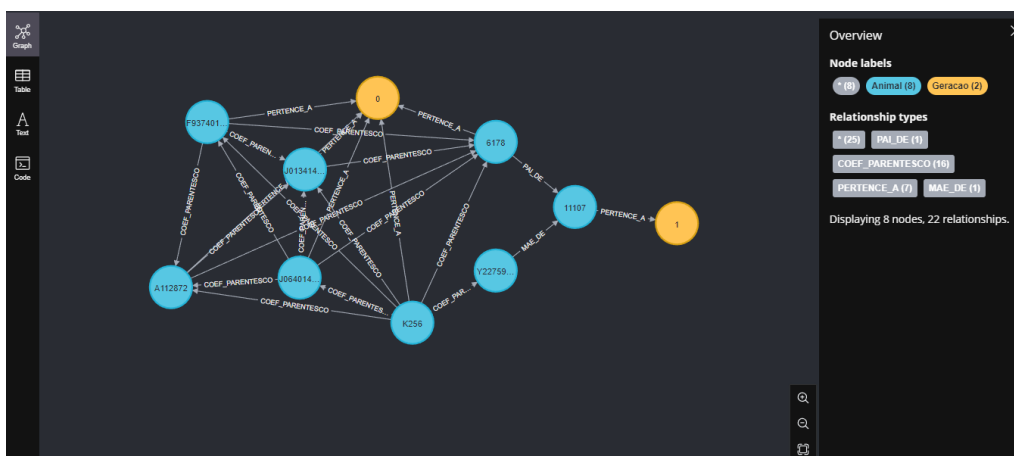


Figura 4. Amostra de relações existentes no banco Neo4J

A Fig. 5 mostra o relacionamento de parentesco calculado entre os animais no banco de dados, em que cada nó do tipo Animal está conectado a todos os outros nós por

meio da relação COEF_PARENTESCO. A figura destaca a complexidade das conexões entre os animais e como as informações genealógicas estão estruturadas para permitir o cálculo dos coeficientes.

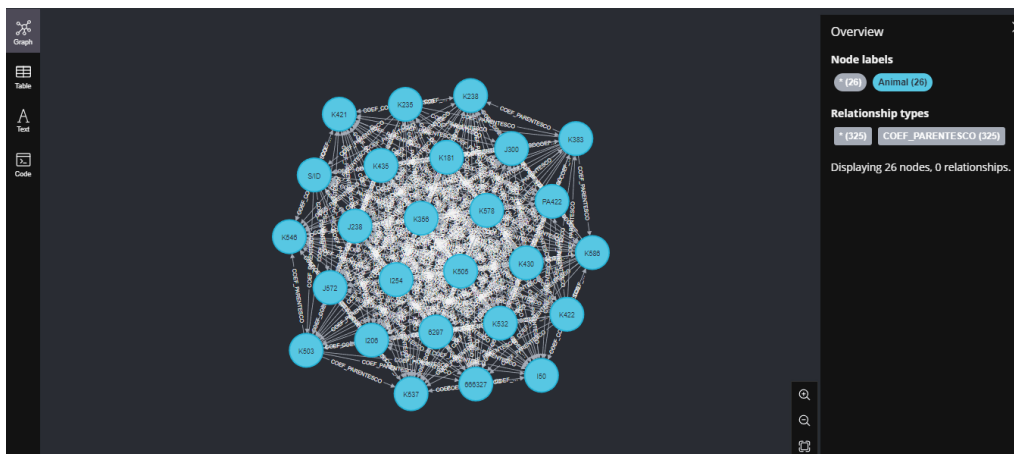


Figura 5. Relacionamento de parentesco entre os animais

A representação das gerações por nodos tem como objetivo otimizar a busca por animais de uma geração, visto que a implementação de bancos de dados baseados em grafos garante uma maior eficiência de busca de arestas do que de atributos de nodos. A flexibilidade de inserção e retirada de quaisquer tipos de nodos ou arestas assegura que novas informações possam ser inseridas ou estruturadas de forma diferente, sem prejuízo dos dados já lançados no banco. Essa é a maior vantagem de um banco de grafos em relação ao um banco relacional. Novos nodos podem ser inseridos no banco à medida que consultas específicas tenham que ser construídas.

6. Conclusão

O controle dos níveis de consanguinidade é essencial para sistemas de melhoramento animal. Este trabalho apresentou um algoritmo dinâmico para o cálculo de valores de coancestralidade entre todos os pares de animais do rebanho, que executa em tempo quadrático no número de animais, mesmo quando somado ao algoritmo de cálculo geracional, que executa em tempo linear.

Bancos de dados orientados a grafos permitem que relações entre entidades do banco sejam implementadas de forma explícita, com acesso otimizado para valores. Adicionalmente, a inserção e retirada de nodos ou aresta do banco não afeta os demais valores, tornando o modelo muito mais flexível para dados pouco estruturados, como é o caso dos sistemas de produção agropecuários. A permanência dos valores calculados não exige seu recálculo e a inserção de novos animais no banco pode ser feita somente com as arestas que ligam esses animais aos demais.

A informação sobre valores de coancestralidade é essencial para a construção de algoritmos que visem construir sistemas de acasalamento que garantam o controle desses índices. A partir do trabalho realizado, pretende-se construir um sistema para acasalamento de animais selecionados dos rebanhos que garanta o maior valor possível de ganho genético e econômico ao mesmo tempo que controle os valores de parentesco, garantindo mais cruzamentos seguros dentro do mesmo rebanho.

Referências

- Aguilar, I. and Misztal, I. (2008). Recursive algorithm for inbreeding coefficients assuming nonzero inbreeding of unknown parents. *Journal of Dairy science*, 91(4):1669–1672.
- Arksey, H. and O'Malley, L. (2005). Scoping studies: Towards a methodological framework. *Int. J. Social Research Methodology*, 8(1):19–32.
- Backus, V. and Gilpin, M. (2002). An efficient algorithm for the additive kinship matrix. *Journal of Heredity*, 93(6):453–456.
- Boyce, A. (1983). Computation of inbreeding and kinship coefficients on extended pedigrees. *Journal of Heredity*, 74(6):400–404.
- Cardoso, F. F. (2009). Ferramentas e estratégias para o melhoramento genético de bovinos de corte. Technical report, Embrapa Pecuária Sul, Bagé.
- Charlesworth, D. and Willis, J. H. (2009). The genetics of inbreeding depression. *Nature Reviews Genetics*, 10:783–796.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2002). *Algoritmos: teoria e prática*. Editora Campus, 2 edition.
- Diestel, R. (2005). *Graph Theory*. Graduate Texts in Mathematics. Springer.
- Elliott, B. et al. (2009). Efficiently calculating inbreeding on large pedigrees databases. *Information systems*, 34(6):469–492.
- Falconer, D. S. and Mackay, T. F. C. (1996). *Introduction to Quantitative Genetics*. Longman Group Ltd., Harlow, 4 edition.
- Ferreira, A. P. L., Yokoo, M. J.-I., and Motta, B. E. T. (2021). On the problem of optimal mating in animal breeding. In *Simposio Latinoamericano de Teoría Computacional, Proceedings of the XLVII Conferencia Latinoamericana de Informática (CLEI 2021)*, pages 1–7, San José, Costa Rica. IEEE. DOI 10.1109/CLEI53233.2021.9640023.
- Gholami, K. and Thomas, A. (1994). A linear time algorithm for calculation of multiple pairwise kinship coefficients and the genetic index of familiarity. *Computers and Biomedical Research*, 27(5):342–350.
- Heckel, R., Corradini, A., Ehrig, H., and Löwe, M. (1996). Horizontal and vertical structuring of typed graph transformation systems. *Mathematical Structures in Computer Science*, 6(6):613–648.
- Khang, J. V. T. (1989). A FORTRAN subroutine to compute inbreeding and kinship coefficients according to the number of ancestral generations. *Bioinformatics*, 5(3):199–204.
- Mi, M. P., Chapman, A. R., and Tyler, W. J. (1965). Effect of mating system on production traits in dairy cattle. *J. Dairy Sci.*, 48:77–84.
- Robinson, I., Webber, J., and Eifrem, E. (2015). *Graph databases: new opportunities for connected data*. "O'Reilly Media, Inc."
- Toscani, L. V. and Veloso, P. A. S. (2006). *Complexidade de Algoritmos: análise, projeto e métodos*. Bookman, Porto Alegre, 2 edition.