REVISTA TRIATLS
INNOVACIÓN Y DESARROLLO TECNOLÓGICO Y SOCIAL
EDUCACIÓN PÚBLICA Y GRATUITA
UNIVERSIDAD NACIONAL DE LA PLATA
Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

Herramienta Informática para el Análisis de Progenie

Caso de estudio: producción de leche en tambos de la Asociación de la Región Pampeana de Entidades de Control Lechero (ARPECOL)

Paula López [1,], Waldo Hasperué [1,2], Ramiro Rearte [3,4], Rodolfo Luzbel de la Sota [3,4]*

[1] Instituto de Investigación en Informática LIDI. Facultad de Informática. Universidad Nacional de La Plata

[2] Investigador asociado - Comisión de Investigaciones Científicas (CIC-PBA)

[3] Instituto de Investigaciones en Reproducción Animal (INIRA). Facultad de Ciencias Veterinarias. Universidad Nacional de La Plata

[4] CONICET

[*]pdlopez@ lidi.info.unlp.edu.ar

Resumen. *Una línea de investigación activa en el INIRIA es el estudio de la teoría de programación fetal y en particular en el ganado bovino de los tambos de la provincia de Buenos Aires. Los datos que son analizados en estos estudios están compuestos de registros de producción de leche individuales realizados en entidades de control lechero oficial nucleados por la ARPECOL. La información almacenada en esta base de datos contiene controles lecheros a partir del año 1980 hasta la actualidad, junto con información de la progenie. Una de los interrogantes que se plantean es determinar la magnitud de la mejora en la producción lechera en las nuevas generaciones del ganado bovino y estimar su posible asociación con indicadores reproductivos en las descendencias. Para contestar a estos interrogantes, primero se debe armar el árbol de la progenie de todas las vacas-individuo y luego analizar para cada una sus correspondientes descendencias y así determinar la magnitud de la mejora en la producción lechera. Si bien este análisis es posible realizarlo con los softwares estadísticos y de análisis de datos actuales, éstos no permiten el armado y el tratamiento de una estructura de forma de árbol que permita realizar análisis entre individuos de distintas generaciones y sus relaciones "familiares" de una manera sencilla y amena.*

La herramienta desarrollada para el INIRIA y presentada en este trabajo permite procesar la información de individuos inter-generacionales, determinar relaciones de parentesco para poder facilitar el posterior análisis y así detectar relaciones entre las diferentes variables productivas y reproductivas de la progenie. Esta herramienta fue desarrollada para que pueda ser utilizada en otros ámbitos que involucren grupos de individuos organizados de forma jerárquica, donde éstos cuenten con información de progenie y un historial de producción.

El trabajo presentado consistió en la depuración de la base de datos provista por ARPECOL al INIRA y la implementación de una herramienta en lenguaje Python que permite el establecimiento de las relaciones de ancestros y descendencia entre individuos, lo cual implica la generación de un árbol de progenie. Para luego procesar la información de cada individuo junto a sus respectivos controles periódicos. En particular y como caso de estudio se obtuvieron como resultado del análisis las curvas de la producción lechera de las vacas analizando la relación entre variables relativas a la producción de leche de las madres y la reproducción de sus hijas.

Palabras clave: Producción; Producción lechera; Programación fetal; Minería de datos.

Recibido: 11/06/2020 Aceptado: 04/08/2020

https://doi.org/10.24215/26838559e011

Software tool for pedigree analysis

Case Study: Milk production at dairy farms of the Pampean Region Association of Milk Control Organizations (ARPECOL)

Abstract. *An active research line at INIRIA is the study of the fetal programming theory, in particular in relation to bovine cattle at dairy production facilities in the province of Buenos Aires. The data analyzed in these studies include individual milk production records kept by official milk control organizations nucleated by ARPECOL. The information stored in this database contains milk controls since 1980 and up to the present day, as well as progeny information. One of the questions to be answered is establishing the magnitude of improvement in milk production in*

newer bovine cattle generation and estimating its potential association to reproductive indicators in descendants. To answer these questions, a progeny tree must be put together for every individual cow, and then all descendants should be analyzed for each of them. This will allow establishing the magnitude of improvement in milk production. Even though this analysis can be carried out using current statistical and data analysis software, these do not offer tools to build and analyze tree structures to carry out analyses of individuals from different generations and their "family" relations in a simple and seamless manner.

The tool developed for INIRIA and presented in this paper allows processing information of individuals across generations, establishing blood relations to facilitate the subsequent analysis, and thus detecting relations between the various production and reproduction variables in the progeny. The tool was developed to be used also in other environments involving groups of individuals that are organized in hierarchies, where there is progeny and production history information available.

Our work consisted in purging the database provided by ARPECOL to INIRA and implementing a tool developed in Python that allows establishing ancestry-descendant relations among individuals, which involves creating a progeny tree. Each individual's information was then processed together with that of their corresponding regular controls. In particular, and as a case study, cow milk production curves were obtained by analyzing the relation between variables related to milk production in mothers and their daughter's reproduction.

Keywords: Production; Milk yield; Fetal programming; Data Mining.

Novedad u originalidad local en el conocimiento

En la Argentina se están llevando a cabo algunos proyectos de investigación en bovinos sobre Programación Fetal (PF). La PF involucra los cambios en la expresión génica del feto que son promovidos por causas nutricionales y/o endócrinas que ocurren en la vaca durante la gestación pudiendo afectar aspectos de la vida adulta de la progenie como, por ejemplo, su performance productiva (Chiarle, 2017).

En el Proyecto de Programación Fetal relacionado al ganado bovino cumplen un papel fundamental las entidades de control lechero: organizaciones que brindan el servicio de medición de la producción de leche individual de las vacas a los productores. En el INIRA se realizan estudios sobre los tambos de ARPECOL, una asociación que nuclea entidades de control lechero de la provincia de Buenos Aires que actualmente dispone de información referida a 1.657.914 vacas. El Instituto utiliza diversos softwares estadísticos y de análisis de datos para procesar la información y generar indicadores productivos, pero dichas herramientas no permiten organizar ni procesar los datos en estructuras arbóreas, las cuales resultan esenciales para efectuar un estudio genealógico de los individuos.

La herramienta presentada en este trabajo permite generar, a partir de la base de datos, una estructura en forma de árbol para organizar a los bovinos por su progenie, lo cual facilita estudios posteriores sobre producción lechera asociada a los grupos familiares.

Grado de relevancia

La relación entre la producción lechera y la performance reproductiva es estadísticamente significativa (Rearte et al., 2018), por lo tanto, identificar los determinantes del proceso reproductivo, incide directamente en la producción de leche tanto a nivel individual como a nivel poblacional.

Como se mencionó anteriormente, los softwares estadísticos actuales no permiten un fácil manejo de datos estructurados como árboles, es por ello que el desarrollo de un sistema que cumpla dicha funcionalidad resulta de importancia para el estudio de variables de producción inter-generacional, en particular la producción lechera nacional.

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

La herramienta desarrollada contribuye a facilitar el procesamiento de la información disponible en las entidades de control lechero: logra reconstruir el árbol de progenie con el objetivo de analizar la producción lechera de cada familia, además de clasificar a los individuos por las relaciones entre la producción de madres e hijas y detectar la existencia de relaciones entre diferentes variables productivas y reproductivas de los individuos.

Si bien el sistema fue pensado y desarrollado teniendo como punto central la producción lechera bovina, la herramienta también puede utilizarse en otros ámbitos que involucren grupos de individuos organizados de forma jerárquica, donde se cuente con un historial de producción.

Grado de pertinencia

El producto desarrollado cumplió con los objetivos planteados y resultó de gran utilidad para los integrantes del INIRA, ya que les permitió estructurar de forma arbórea las relaciones de parentesco entre las vacas presentes en la base de datos de ARPECOL, y así facilitar los análisis que se realizaron posteriormente, lo cual concluyó en la obtención de resultados novedosos en el ámbito de la producción lechera bovina de Buenos Aires. Dichos resultados planean ser publicados en revistas internacionales con referato y presentados en congresos relativos a la disciplina de las ciencias médicas veterinarias.

Grado de demanda

El sistema fue desarrollado como parte de una colaboración ante la propuesta planteada por el INIRA para la asistencia del Proyecto de Programación Fetal, investigado en dicho instituto.

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

La herramienta desarrollada permite que en el Proyecto de Programación Fetal se puedan realizar análisis que logran establecer relaciones de mejora en la productividad lechera. El poder detectar casos donde no se cumpla con una mejora en la calidad o cantidad de leche, o incluso en el peor de los casos un decremento de la calidad o cantidad, podría servir como alarma para poder tomar acciones que permitan actuar en consecuencia y así mejorar la calidad de la producción lechera. De esa manera se puede lograr un plan de acción que tiene consecuencias directas en la calidad de los productos lácteos que llegan a los consumidores.

Desarrollo del producto

Previo al desarrollo del producto se realizaron dos etapas que pertenecen al proceso de la minería de datos: limpieza de la base de datos y selección de los atributos más relevantes. Luego de un análisis preliminar de la base de datos y de la problemática planteada y conociendo además las limitaciones de los softwares de análisis convencionales se decidió llevar a cabo la implementación de una herramienta que permita el armado del árbol de progenie para su posterior análisis y así poder brindar respuestas a los interrogantes planteados por el personal del INIRA.

En particular y para este trabajo, la herramienta fue utilizada para el análisis de la relación inter-generacional para la construcción de las curvas de lactancia de Wood (Wood, 1967) y el posterior uso de la información de producción lechera para tareas de regresión y la construcción de modelos predictivos.

Las curvas de lactancia representan la producción de leche a lo largo del ciclo productivo, el cual dura aproximadamente 305 días (Bretschneider et al., 2015), y permiten observar la producción individual para cada lactancia de una vaca y asociarla con las de su progenie y el resto de los individuos. Las mismas se generan a partir de los días en leche de una vaca y de la producción de leche de la misma,

junto con la ecuación de Wood y mínimos cuadrados no lineales (Gauss-Newton) como método de ajuste de curvas. En la **Figura 1** se muestra una curva de lactancia generada por la herramienta.

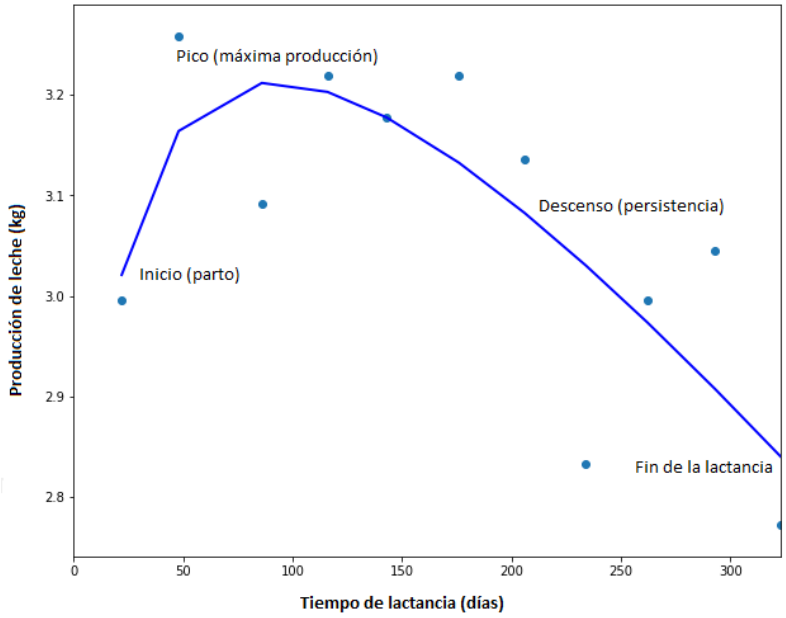

Figura 1. Gráfico de la curva de lactancia de una vaca de la base.

Figure 1. Lactation curve of a dairy cow from the database.

I) Limpieza de la base de datos

En primer lugar se realizó la selección de las tablas y campos de la base de datos provista por ARPECOL que fueran relevantes para los análisis. Fueron seleccionadas tres tablas:

- *vaca*, que contiene un registro por cada vaca en el sistema e información de su nacimiento, de sus progenitores, del historial de su performance productivo y de sus últimas performances productivas. Ver **Tabla 1**.
- *lactancia,* que contiene un registro por cada parto de la vaca (inicio de la lactancia) e información del parto y de la cría. Ver **Tabla 2**.

- *lechero*, que contiene un registro por cada control lechero realizado durante las lactancias de cada vaca. En cada control se mide la producción diaria de leche y se toman muestras de leche para ser analizada en un laboratorio midiendo niveles de grasa, proteínas, sólidos, conteo de células somáticas, entre otras. Ver **Tabla 3**.

Las tres tablas mencionadas se procesaron con Python generando DataFrames que sirven como entrada al proceso del armado del árbol.

Tabla 1. Campos utilizados de la tabla *vaca*.

Table 1. Used attributes from *Cow* table.

Campo	Descripción
ID_ECLO_TAMBO_VACA	Identificación única de la vaca
TAMBO	Número de tambo
RP_VACA	Identificación dada por el tambo
FE_NACIMIENTO	Fecha de nacimiento de la vaca
RP_MADRE	Número de la madre dentro del tambo

Tabla 2. Campos utilizados de la tabla *lactancia*.

Table 2. Used attributes from *Lactation* table.

Campo	Descripción
ID_ECLO_TAMBO_VACA	Identificación única de la vaca
FECHA	Fecha del parto
NROPARTO	Número de parto/hija de la vaca

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

Tabla 3. Campos utilizados de la tabla *lechero*.

Table 3. Used attributes from *Dairy* table.

Campo	Descripción
ID_ECLO_TAMBO_VACA	Identificación única de la vaca
FECHA	Fecha del control lechero
DEL	Días en leche (días que lleva produciendo)
LECHE	Producción de leche en KG
lact	Número de lactancia

II) Descripción y uso de la herramienta

La herramienta desarrollada consiste en un conjunto de funciones implementadas en Python, las cuales permiten armar el árbol de progenie. Además ofrecen dos funcionalidades extras, la primera se utiliza para realizar filtros e ir "podando" el árbol de progenie de acuerdo a las características que deben tener los individuos, por ejemplo: aquellos que tienen más de tres hijas, individuos que sean "hijos únicos", que tenga hermanos, etc. Además de poder realizar los filtros usando los propios datos originales que se utilizaron para armar el árbol.

La segunda funcionalidad es la de poder obtener tablas para posteriores análisis, donde las filas de estas tablas están formadas por datos de individuos con cierta relación de parentesco, como por ejemplo todas las madres y sus hijas, todas las hermanas, etc. En particular para el caso de la producción lechera presentado en este trabajo, se obtuvieron las relaciones madre-hijas para crear las curvas de lactancia de cada individuo y generar modelos predictivos basados en regresión.

III) Preparación de los datos

Para poder armar el árbol de progenie se debe preparar una tabla (DataFrame de Python) con los siguientes atributos obligatorios: ID y ID_MADRE, donde el primero es la identificación única de cada individuo y el segundo es la identificación única de la madre del individuo. Si no se cuenta con estos dos datos, la herramienta es incapaz de armar el árbol de progenie.

Como dato opcional es posible contar con el orden de nacimiento de un individuo. Este debe ser un dato numérico, si vale 1 entonces se interpreta como que el individuo es el primer hijo de una madre, si vale 2 entonces es el segundo hijo, etc. Contar con este dato permite a posterior consultar por la secuencialidad de los individuos que son hermanos entre sí.

En nuestro caso de estudio y luego de la limpieza de la base de datos se armó un DataFrame con las columnas:

- ID. Identificación única del individuo-vaca
- FECHA_NAC. Fecha de nacimiento del individuo-vaca
- VIVE_EN. Identificación del tambo donde nació y vive el individuo-vaca
- ID_MADRE. Identificación individuo-vaca de la madre
- NRO_PROD. Número de orden de nacimiento (número de hijo)

IV) Creación del árbol de progenie

Una vez que se cuenta con el DataFrame con las características descritas en la sección anterior el árbol de progenie se construye de la siguiente manera:

```
arbol = buildTree(dataframe, id="ID", mother="ID_MADRE",
 birth_order="NRO_PROD")
```

donde **buildTree** es la función de la herramienta que construye el árbol de progenie. Esta función debe recibir el DataFrame preparado para esta operación y se deben indicar cuales son los nombres de las columnas donde están los datos necesarios para la identificación del parentesco: el código único del individuo (**id**), el código único de individuo de la madre (**mother**) y, de poseerlo, el número de hijo (**birth_order**).

Esta función devolverá una estructura de datos que permitirá trabajar con el árbol de progenie realizando filtros y solicitando tablas de relaciones para análisis posteriores. En el ejemplo anterior, esa estructura queda almacenada en la variable **arbol**.

V) Limpieza y filtrado del árbol de progenie

La herramienta provee una función de filtro que permite descartar temporalmente aquellos individuos que no cumplen con ciertas condiciones como por ejemplo, un número insuficiente de hijos o que el individuo haya vivido una cierta cantidad de años, pares madre-hija en los cuales no se cumpla una diferencia mínima de edad entre los individuos, etc.

El uso de este filtro es el siguiente:

```
poda1 = arbol.filter(lambda ind:
        (ind["FECHA_NAC"] - ind.mother["FECHA_NAC"]).days > 300)
```

arbol es la variable devuelta por la función **buildTree** y **filter** es la función provista por la herramienta la cual aplica el filtro especificado al árbol. El filtro es pasado en

formato función lambda de Python (Python Software Foundation, 2020). Esta función lambda va a ser aplicada a cada individuo presente en el árbol. En este ejemplo vemos que se chequea que la diferencia entre días entre el nacimiento de un individuo y su madre sea superior a 300. En caso de que el filtro se cumpla, el individuo pasará a formar parte del resultado, en caso contrario no formará parte.

El filtro es usado como filtro de inclusión, es decir, aquellos individuos que cumplan la condición del filtro serán incluidos en el resultado de la operación.

La función de filtro es sumamente versátil y puede utilizarse aplicando más criterios de filtrado que el ejemplo mencionado con anterioridad. El resultado de la función es un nuevo árbol de progenie que está formado por los individuos que cumplieron con los parámetros de filtrado establecidos. A este nuevo árbol puede aplicársele un nuevo filtro y así sucesivamente la cantidad de veces que sea necesaria. Esto permite al usuario combinar los filtros según sus necesidades. De esa manera, el usuario de la herramienta tiene acceso en todo momento tanto al árbol generado en primera instancia como a aquellos creados a partir de distintos filtros. En la **Figura 2** se muestra un ejemplo con el uso de los filtros.

Como se observa en la **Figura 2**, a partir del árbol generado por la función **buildTree**, el usuario puede aplicar diferentes filtros, los cuales devuelven nuevos árboles a los cuales se le puede seguir aplicando filtros. El proceso de filtrado no es lineal, el usuario posee tanto el árbol original como todos los árboles filtrados intermedios para seguir realizando operaciones sobre ellos.

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

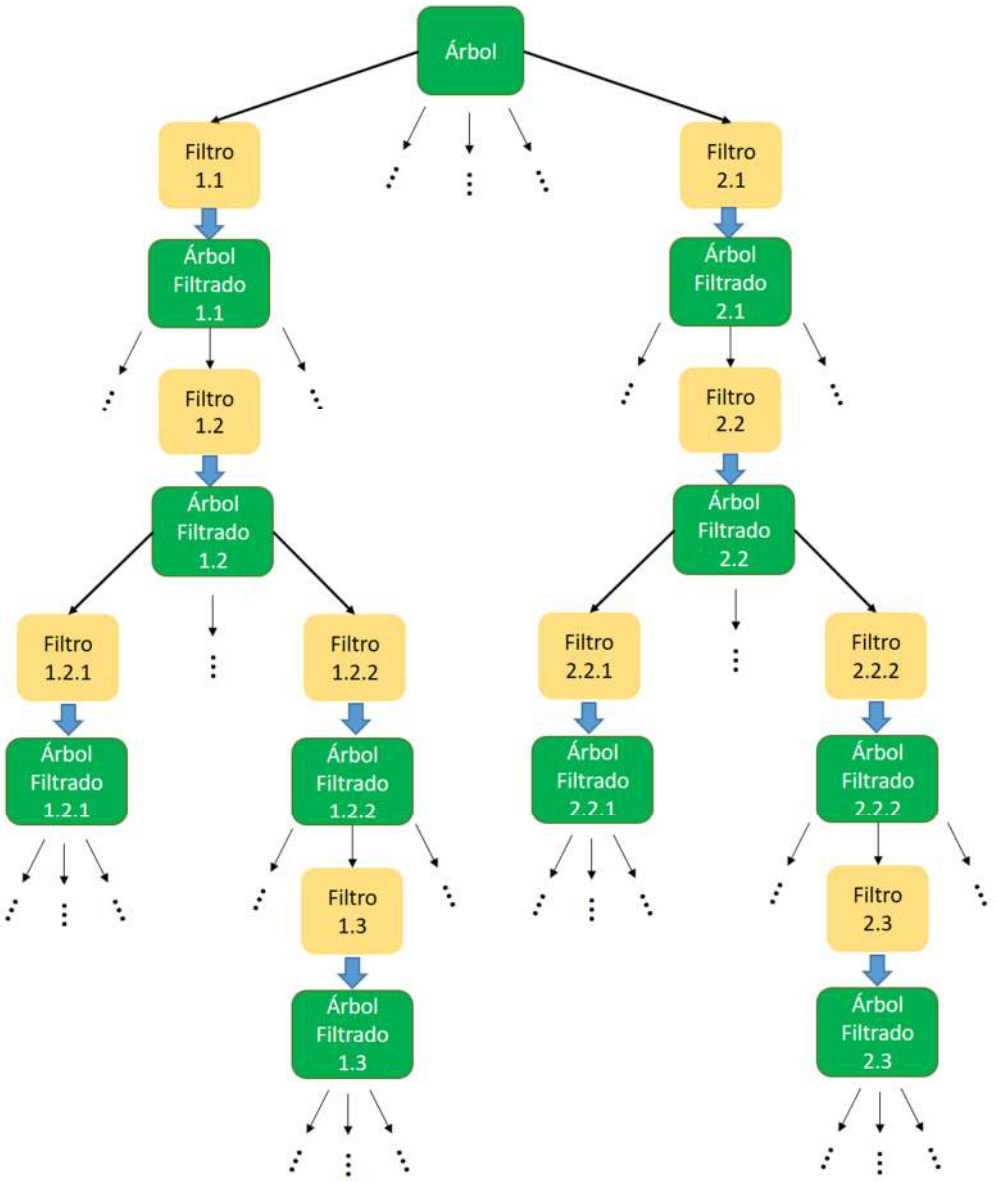

Figura 2. Como resultado de un filtro se obtiene otro árbol el cual puede ser usado para aplicar distintos filtros y así obtener ramificaciones en el filtrado.

Figure 2. As a result of a filter another tree is obtained. This tree can be used to apply other filters obtaining ramifications in the filtering.

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

En el caso de estudio una tarea de limpieza que se realizó fue la de corroborar que una vaca esté en el mismo tambo que la madre. Con este filtro fueron descartados los casos mal cargados en la base de datos original.

```
poda2 = poda1.filter(lambda ind:
(ind["VIVE_EN"] == ind.mother["VIVE_EN"]))
```

En este ejemplo, la función **filter** está siendo aplicada sobre el resultado del filtro mostrado en el ejemplo anterior (**poda1**). Cabe mencionar que el campo **VIVE_EN** que se referencia en el filtro es un campo propio de la base de datos original.

Otras facilidades que incluye la herramienta es poder hacer referencia a individuos según su tipo de parentesco. En los ejemplos anteriores se referencia a la madre del individuo (**ind.mother**), pero también es posible acceder a la abuela (**ind.grandmother**), a la hermana anterior (**ind.prevSister**), a la siguiente (**ind.nextSister**), a una hermana en particular (**ind.sister[3]**) o a un hijo en particular (**ind.children[2]**), donde el número indicado entre corchetes es el índice de listas de Python, es decir, 0 para el primer elemento, 1 para el segundo, etc.

El siguiente ejemplo muestra la eliminación de todas aquellas vacas-individuos que no tuvieran descendencia, ya que no resultaban pertinentes para los análisis posteriores.

```
poda3 = poda2.filter(lambda ind: (ind.childrenCount > 0))
```

Aquí se hace referencia a **ind.childrenCount** que es una propiedad incluida en la herramienta que indica la cantidad de hijos de un individuo. La herramienta

también provee las propiedades **ind.sistersCount** e **ind.granddaughtersCount** para la cantidad de hermanas y nietas de un individuo en particular.

VI) Agregado de datos a los individuos

A cada individuo que forma parte del árbol de progenie se le puede agregar datos extras para posteriores análisis. En el caso de estudio relacionado a este trabajo, de cada vaca-individuo se conocían controles lecheros.

Estos controles pueden contener uno o más registros para cada vaca-individuo, los cuales poseen la identificación única del individuo (ID), la fecha del control realizado (FECHA_CONTROL), la cantidad de días transcurridos desde el inicio de la producción de leche hasta la fecha del control (CANT_DIAS), la cantidad de producción medida en Kg. de leche (CANTIDAD) y el número de producción o número de lactancia al que corresponde el control (NRO_PROD).

Para agregar la información extra a cada individuo debe realizarse sobre el árbol original, el que contiene a todos los individuos de la base de datos y se realiza con la función *addDatum*.

```
arbol.addDatum(controles, id="ID", label="CONTROLES")
```

donde **arbol** es la variable que contiene el árbol devuelto por la función **buildTree**. **controles** es un DataFrame con la información que se desee agregar a cada individuo. Este DataFrame debe tener una columna con el mismo identificador único de individuo que posee el DataFrame utilizado al armar el árbol de progenie. Si bien los identificadores deben ser únicos, el nombre de la columna dentro del DataFrame controles puede ser distinto, por eso hay que indicar en la función **addDatum**, cual es la columna del DataFrame que posee dicho atributo (parámetro

id). Todos estos datos serán agregados al individuo correspondiente dentro de un campo cuyo nombre es pasado en el parámetro **label**, y dentro de este nuevo atributo cada control correspondiente será almacenado usando una lista de python.

Así por ejemplo, la cantidad de controles de un individuo se puede realizar de la siguiente manera:

```
poda4 = poda2.filter(lambda ind: (len(ind["CONTROLES"]) >= 7))
```

en este ejemplo estamos filtrando todos los individuos que tienen siete o más controles. Cabe aclarar que el campo **CONTROLES** es el que se pasó como argumento en el parámetro **label** de la función **addDatum**.

La cantidad de atributos de cada DataFrame adicional como así también la cantidad de conjuntos distintos de datos extra no tiene límite. Cada conjunto de datos extra que se desee agregar a los individuos del árbol se realiza con la misma función **addDatum**, por ende esta función puede ser invocada cuantas veces se necesite. La única condición, por razones obvias, es que cada conjunto de datos extra sea agregado con un **label** distinto.

VII) Obtención de relaciones familiares para análisis posteriores

Una vez establecida la progenie de todos los individuos y de haber aplicado todos los filtros necesarios, se brindan funciones para obtener una lista de n-tuplas por cada relación familiar detectada que cumpla con un determinado criterio de búsqueda como, por ejemplo, encontrar a todas las hermanas de un individuo.

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

Dependiendo del criterio utilizado una n-tupla representa una rama o un nivel del árbol, esto implica que un individuo puede aparecer múltiples veces encabezando un listado dependiendo de la cantidad de descendientes o hermanas que posea.

Las funciones implementadas en esta herramienta son las siguientes:

- descendents(grade). Permite obtener todas las relaciones madre-hija

```
madre_hijas = poda4.descendents(grade=2)
```

esta función devuelve una lista con tuplas de python con toda la información de los individuos que son madres de al menos una hija. La madre puede aparecer como tal en más de una una tupla si es que tiene más de una hija. De la misma manera una hija puede aparecer como madre en otras tuplas si es que también es madre.

Esta misma función puede utilizarse para conseguir las relaciones abuela-madre-hija:

```
abuela_madre_hijas = poda4.descendents(grade=3)
```

donde la diferencia entre obtener madre-hija o abuela-madre-hija está en el parámetro **grade** pasado a la función **descendents**. Esta función también permite obtener las n-tuplas de aquellas descendencias cuyo grado sea n.

- sisters(count) permite obtener todas las duplas (hermana-hermana) consecutivas en nacimientos:

```
hermana_hermana = poda4.sisters(count=2)
```

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

esta función devuelve una lista con tuplas de python con toda la información de los individuos que son hermanas nacidas de manera consecutiva. La hermana que aparece en segundo lugar, podría aparecer en primer lugar si tiene a su vez una hermana más chica. Por ejemplo, entre cuatro hermanas consecutivas se obtendrán las tuplas (hermana1, hermana2), (hermana2, hermana3) y (hermana3, hermana4)

De manera similar a la función descendents se pueden obtener n-tuplas con más hermanas pasando por el parámetro **count** la cantidad de hermanas deseadas. Por ejemplo:

```
hermana_hermana_hermana_hermana = poda4.sisters(count=4)
```

Esta función también permite la obtención de todas las combinaciones posibles entre hermanas, sin importar su orden de nacimiento. Para conseguirlo hay que indicarlo en el parámetro opcional **perm**.

```
hermanas2 = poda4.sisters(2, perm=True)
```

Volviendo al ejemplo anterior de cuatro hermanas, esta función devolvería las tuplas (hermana1, hermana2), (hermana1, hermana3), (hermana1, hermana4), (hermana2, hermana3), (hermana2, hermana4) y (hermana3, hermana4).

Estas dos funciones pueden ser aplicadas tanto al árbol original como a cualquier resultado de algún filtro, en este último caso, todos los individuos que forman parte de las relaciones deben haber cumplido todas las condiciones de filtro que se aplicaron.

Innovación y Desarrollo Tecnológico y Social (2020) 2 (1): 35-54

En el caso de estudio particular presentado en este trabajo se utilizó la función **descendents** con grado 2, ya que necesitábamos dichas tuplas para obtener las curvas de lactancia de cada individuo y usar esa información para alimentar un modelo de regresión que permita establecer la magnitud de la mejoría en la producción de leche de una vaca-hija comparada con la fecha de la inseminación en su madre.

Conclusiones

Pudo aplicarse exitosamente la herramienta desarrollada para el armado del árbol de progenie a partir de los datos del caso de estudio y la base de datos que fuera provista. A partir del árbol inicial se aplicaron todos los filtros sugeridos por los investigadores del INIRIA obteniendo más de 243000 pares madre-hija a los cuales se les pudo calcular la curva de Wood y posteriormente llevar a cabo un análisis de regresión entre las variables de interés.

La herramienta desarrollada fue construida pensando en que pudiera ser utilizada en cualquier entorno que requiera el análisis de individuos inter-generacionales. La funcionalidad provista por esta herramienta en su primera etapa permite:

- Establecer las relaciones familiares entre individuos mediante la generación de un árbol genealógico a través de un identificador único de individuo que se debe poseer a priori.
- Realizar la limpieza del árbol creado a partir de la aplicación de filtros completamente personalizables (desarrollados en código python) según el objetivo buscado.
- Poder hacer referencia a las relaciones parentales de manera fácil e intuitiva en los filtros haciendo uso de atributos predefinidos en la herramienta.
- Obtener una lista de n-tuplas de individuos, cada una de las cuales representa una relación familiar (inter-generacional e intra-generacional).

- Recuperar para cada individuo su producción y controles periódicos de todas sus descendencias.

Es de esperar que al utilizar esta herramienta en otros entornos surjan inconvenientes o nuevas necesidades que permitan completar y mejorar la herramienta presentada.

Referencias bibliográficas

Chiarle, A. (2017). *Programación Fetal en Vacas Lecheras*. Recuperado de: http://sedici.unlp.edu.ar/handle/10915/66335

Bretschneider, G.; Salado, E. E.; Cuatrin, A.; Arias, D. R. (2015). *Lactancia: Pico y Persistencia ¿Por qué cuidarlos?* Recuperado de: https://inta.gob.ar/sites/default/files/ script-tmp-inta_lactancia_pico_y_persistencia_febrero_ 2015.pdf

Python Software Foundation (2020). *Expressions. Python 3.8.3 documentation.* Recuperado de: https://docs.python.org/3/reference/expressions.html

Rearte, R.; LeBlanc, S. J.; Corva, S. G.; de la Sota, R. L.; Lacau-Mengido, I. M.; Giuliodori, M. J. (2018). *Effect of milk production on reproductive performance in dairy herds.* Journal of Dairy Science, vol. 101, no. 8, pp. 7575–7584. doi: 10.3168/jds.2017-13796

Wood, P. (1967). *Algebraic Model of the Lactation Curve in Cattle.* Nature 216, 164–165. doi: 10.1038/216164a0