DATA MINING EN EVALUACIONES DE BIODIVERSIDAD

Luis López. Departamento de Ingeniería. UNLaM
Pablo Martínez. Departamento de Ingeniería. UNLaM
Ariel Cacho Mendoza. Departamento de Ingeniería. UNLaM
Marcelo Soria. Facultad de Agronomía, Cátedra de Microbiología UBA
Cristóbal R. Santa María. Departamento de Ingeniería. UNLaM
Florencio Varela 1903 San Justo Pcia. de Buenos Aires
54-011-44808952
llopez@ing.unlam.edu.ar
pablowmartinez@yahoo.com.ar
arielcm@gmail.com
soria@agro.uba.ar
csanta_maria@ing.unlam.edu.ar

RESUMEN

Las modernas técnicas de secuenciación de ADN transforman su estructura química en secuencias informáticas de símbolos cada una de las cuales puede ser vista como una instancia de una base de datos. Es posible entonces aplicar técnicas para clasificar casos y predecir patrones de comportamiento de forma similar a como se lo hace sobre otros dominios como las finanzas, el marketing o el texto, aunque la complejidad del dominio microbiológico pueda llevar a una tarea un poco más ardua. En tal sentido la aplicación de data mining en los estudios genómicos es un hecho consolidado en la investigación biológica pues en ella también se trata de clasificar y descubrir patrones sobre grandes bases de datos con el auxilio de técnicas combinadas de aprendizaje automático, estadística y visualización lo que en suma no es más que la definición ontológica de la minería de datos.

El trabajo aquí presentado se refiere a secuencias de ADN correspondientes a distintos microorganismos extraídas de muestras de suelo con el objetivo de evaluar los patrones de riqueza y diversidad de la comunidad microbiológica que lo integra. En particular cada secuencia de ADN correspondiente al gen 16S rRNA que integra la muestra se identificará con un organismo distinto. La tecnología de secuenciación actual es capaz de obtener miles de estas cadenas de símbolos correspondientes a los cuatro componentes básicos del ADN: A-adenina, T-timina, C-citocina y G-guanina. Cada parte de un gen será entonces una secuencia de unos cientos de estos símbolos colocados en algún orden. Tal como se hace por ejemplo en text mining, se puede definir una distancia conveniente entre secuencias y con ella producir un clustering que agrupe las secuencias según su similitud. Así, eligiendo un umbral de disimilaridad adecuado, cada agrupamiento estará integrado por secuencias correspondientes a individuos de la misma especie, Estos clusters se denominan Unidades Taxonómicas Operacionales y a partir de su distribución de abundancia en la muestra, se pretende establecer el patrón de riqueza de la comunidad, lo que significa establecer el número de especies que hay en la misma. Esta tarea se topa con un serio problema estadístico pues en microbiología más del 70% de las especies pueden ser estadísticamente raras a la vez que un 10% es muy abundante.

De tal forma las muestras no contienen individuos de muchas especies presentes y a su vez presentan muchos individuos de las especies dominantes. Es decir; toda muestra resulta pequeña para una inferencia estadística simple de la riqueza poblacional. El algoritmo de recuento de especies ARE, ya presentado en otros trabajos (1) y (2), mejora las estimaciones no paramétricas habituales y las hace compatibles con las apreciaciones ecológicas. En términos más generales el algoritmo resuelve en forma eficiente el problema de inferir desde una muestra de casos el número de clases de casos que hay en una población que contiene una alta proporción de clases raras. Este problema se reconoce también, por ejemplo, en el análisis de texto donde cada palabra distinta es una clase y hay palabras muy poco frecuentes (3). Hay que remarcar que el número inferido para la riqueza como cantidad de especies distintas, o si se quiere palabras distintas, constituye una guía imprescindible para afinar el clustering que se realice sobre nuevas muestras de la población para determinar una clasificación estable y aplicable luego para predicción.

En este trabajo se planteó el objetivo de desarrollar un programa escrito en Lenguaje C o C++ que permitiera reemplazar al programa del algoritmo ARE escrito en lenguaje R con el fin de mejorar los tiempos de ejecución Se estudian además las posibilidades de paralelización en la ejecución de los algoritmos.

CONTEXTO

La línea de trabajo que aquí se presenta se inscribe en el proyecto de investigación de técnicas de minería de datos aplicadas sobre bases de secuencias de ADN correspondientes a una colección de microorganismos. El proyecto tiene la finalidad de buscar instrumentos adecuados para evaluar la biodiversidad. A su vez se intenta desarrollar una programación que mejore los tiempos de procesamiento con respecto a los de ejecución en lenguaje R que fue utilizado para experimentación inicial.

INTRODUCCIÓN

Para abordar el problema planteado se desarrolló un modelo experimental a partir de la riqueza de una muestra. En ese estado inicial se considera la probabilidad de que al elegir un próximo individuo para incorporar a la muestra, éste resulte de una especie hasta ahora no contabilizada. Si tal cosa ocurre, se suma una nueva especie al total contado. Si no, el número de especies queda igual. En cualquier caso ha variado el número de individuos considerados y el sistema se halla en un nuevo estado. Cuando se ha elegido la cantidad suficiente de individuos se obtiene una estimación de la riqueza medida en este caso en número de especies. Para estimar la probabilidad de que el próximo individuo que se incorpore a la muestra sea de una especie nueva se usa, en el paso *i* del algoritmo, el cociente:

$$\hat{T}_i = \frac{n°sgletones}{i-1}$$

Se realiza entonces una simulación por la técnica de Monte Carlo que a partir de la muestra inicial de tamaño *n* genera una comunidad simulada cuya cantidad de especies estima la riqueza de la comunidad real. (1)

El proceso estocástico modelado se formaliza:

Definición 1: Dada una muestra de tamaño n sea, para cada i, con $i = 1,2,...$, la variable aleatoria S_i que toma los valores $S_i = S_{i-1}$ y $S_i = S_{i-1} + 1$ con probabilidades respectivas $1 - p_i$ y p_i siendo además $S_0 = S_n$. La sucesión de variables aleatorias $S_1, S_2, S_3, ...$ se

denomina en adelante Proceso Aleatorio de Cantidad de Especies.
Se prueban dos propiedades de tal proceso :
Propiedad 1: $\lim_{i\to\infty} T_i = 0$
Propiedad 2: $\lim_{i\to\infty} S_i \leq S$ donde S es el número real de especies en la comunidad.
Ambas propiedades, bajo el supuesto de la adecuada estimación de p_i que proporciona $\hat{T}_i$, aseguran la convergencia del procedimiento ARE a un valor menor o igual que el de la real riqueza poblacional. (2)

Los pasos del algoritmo son (4):

1- Dada la muestra elegida, de tamaño n , y su agrupamiento en OTUs, se determina el valor inicial del estimador de Turing $\hat{T}_{i+1} = \frac{n°sgletons}{i}$ siendo $i = n$
2- Se elige un número aleatorio r , tal que $0 \leq r \leq 1$ y se pregunta si está en el intervalo $[0, \hat{T}_{i+1})$. Si es así, se realiza $S_{i+1} = S_i + 1$ y se va al paso 4. Si ocurre lo contrario se realiza $S_{i+1} = S_i$ y se va al paso 3
3- Se utiliza la distribución de abundancia de la muestra para calcular la proporción de individuos que están en OTUs de $1,2,...,n$ individuos y con estas proporciones se determina, por un sorteo de acuerdo a ellas, a que grupo de OTUs ya conocidas pertenece el nuevo individuo. Para establecer a que OTU específica, de entre las de este grupo, corresponde el nuevo individuo se realiza un nuevo sorteo con probabilidad uniforme para cada OTU del grupo.
4- Sea el nuevo individuo de una nueva especie o no, la muestra tiene ahora un elemento más. Se pregunta entonces si el procedimiento debe cortarse porque se cumple el criterio elegido para ello en cuyo caso la simulación ha finalizado. Si el criterio de corte no se cumple, se asigna entonces $i \leftarrow i+1$, se calcula la nueva distribución de abundancia y la nueva estimación de Turing y se repite desde el paso 2.

En cuanto a las muestras del gen 16s rRNA que caracterizan a cada individuo estas fueron extraídas del repositorio internacional NCBI (5). Su procesamiento inicial se realizó por medio del software libre MOTHUR (6) para realizar el clustering con un umbral de disimilaridad del 5% suficiente para reconocer individuos de la misma especie. La salida de tales procesos es un archivo ".list".

LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

El presente trabajo se considera la etapa de desarrollo tecnológico de la línea de investigación que procura establecer estimaciones de patrones de riqueza y diversidad a partir de relevamientos denominados metagenómicos pues reúnen partes de genomas de muchos individuos de comunidades microbianas.

Al reemplazar los programas experimentales escritos en lenguaje R, se plantea entonces como estrategia resolver el problema en las siguientes etapas:

- Proceder a la lectura del archivo ".list" generado por MOTHUR
- Elegir un generador de números al azar

- Generar la simulación de la comunidad evaluando en ella la cantidad de especies.

El objetivo es reducir los tiempos de ejecución del algoritmo preparándolo para su incorporación a una plataforma de procesos estándar de secuencias genómicas.

RESULTADOS Y OBJETIVOS

a) Lectura del archivo ".list".

Se trata de un archivo de texto en formato CSV, en el mismo se utiliza el carácter de tabulación como separador de campo, y dentro de cada campo, si corresponde, los distintos individuos de un cluster u OTU se separan con el carácter ',' (coma). La marca de fin de registro es la habitual (marca de fin de línea de texto 0x0D 0x0A). En el primer campo se indican los niveles de agrupamiento (unique, 0.00, 0.01, etc.), o sea el porcentaje de diferencia en las secuencias de ADN, En el segundo campo se indica el número de clusters (OTUs) de la muestra. A continuación los distintos clusters que pueden estar formados por uno o más individuos. La estrategia elegida, para su uso a futuro, es la de leer como texto el archivo ".list" y generar tantos archivos temporarios como registros tiene el archivo ".list" en el que se reemplazan los caracteres de tabulación por la marca de fin de registro además de reemplazar los caracteres ',' por tabulación. A continuación se leen los distintos archivos temporarios y se genera el informe "info.csv". Se han validado los resultados obtenidos. Se opta por la futura representación de solo los seis primeros dígitos decimales a pesar de que se los calcula con precisión double.

b) Generador de números al azar.

Se analizaron distintos algoritmos de los existentes, y se decidió utilizar el algoritmo denominado ran3 adaptado de Knuth (7). Se ha probado el mismo generado un archivo de texto con 2.000.000.000 de números al azar demorando poco más de una hora, cuyo costo de ejecución en más del 99% se debe a la grabación del archivo.

Se generan 200.000.000 de números al azar en una matriz en que los tres primeros dígitos decimales de cada número al azar direccionan el número de fila y los tres siguientes el número de columna (truncando los restantes dígitos), totalizando en cada posición de la matriz la cantidad de veces que aparece cada uno de los números al azar. Para visualizar los resultados se generó el archivo "azar.csv" a partir del que se generan las planillas de cálculo "azar.ods" y "azar.xls" en las que se hace uso de distintas fórmulas con el objeto de visualizar el resultado obtenido.

Observación: dado que la matriz de 1.000.000 de enteros excede el espacio de memoria que permite Windows, se ha hecho uso de un archivo de paginación programado ad hoc.

Esta prueba ha dado un detalle relacionado con el hecho de que, cuando el individuo generado corresponde a una especie preexistente, se deben generar uno o dos nuevos números al azar. Al ejecutar el programa generando 600.000.000 de números al azar y tomar uno de cada tres de ellos para la matriz descripta, se afecta en gran medida la uniformidad de la distribución.

Esto ha llevado a modificar la función ran3 para poder generar tres secuencias separadas con el mismo algoritmo.

c) Generar la simulación en que se toma como entrada el archivo ".list" y se procesa una de las "filas" del mismo.

Se aprovecha parte del programa del paso a), y para la carga de la información en lugar del uso de un vector se utiliza una lista simplemente enlazada (tras probar con listas doblemente enlazadas cuyo

empleo no se justifica). El tipo de dato lista contiene la información de los totales en tanto que los nodos están ordenados por la cantidad de individuos en el cluster. La salida del programa está hecha por pantalla y se muestra cada 1000 individuos generados la información similar a la generada en el paso a). La salida por pantalla puede hacerse a un archivo de texto con ínfimas modificaciones.

Las modificaciones hechas sobre la rutina de inicialización y generación de números al azar permitirán poder inicializar más de un generador.

Se han contrastado los resultados obtenidos, y coinciden con lo esperable.

Se han utilizado las fuentes de información disponibles. Entre ellas el libro electrónico Numerical Recipes In C (8), los trabajos de Knuth (7) y el de Marsaglia (9)

Actualmente se trabaja en las posibilidades de paralelización del algoritmo que parecen bastante bajas, pero no se descarta que en tanto se actualiza la inserción de un nuevo individuo y el recálculo de frecuencias se pueda realizar otro hilo de ejecución con la generación del los nuevos números al azar para el nuevo individuo. En cambio parece posible paralelizar la corrida de varias simulaciones sobre el archivo de la muestra inicial.

Los programas se han desarrollado en una plataforma Windows XP, utilizado el entorno de desarrollo Code::Blocks con el compilador MinGW/gcc-4.7.1 de distribución gratuita. Los fuentes podrán ser compilados con ínfimas modificaciones, si las hubiera, en otras plataformas.

BIBLIOGRAFÍA

1- Santa María C. y Soria M. (2013) Simulation applied to the estimation of microbial richness Resumen 4CAB2C 14

2- Santa María C. y Soria M. (2013) Inferencia de Parámetros de Biodiversidad por medio de Simulación" MACI Vol 4 1: 5-8

3- Nádas, A. (1985) On Turing's Formula for Word Probabilities. IEEE Transactions on Acoustics, Speech and Signal Processing. Vol ASSP-33 N° 6.

4- Santa María C. y Soria M. (2011) Estimación de Biodiversidad por Data Mining y Simulación" XVII Congreso Argentino de Ciencias de la Computación CACIC2011. 969-978

5- http://www.ncbi.nlm.nih.gov/

6- www.mothur.org

7- Donald Knuth. "The Art of Computer Programming" http://www-cs-faculty.stanford.edu/~uno/taocp.html

8- William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery "Numerical Recipes in c" http://www2.units.it/ipl/students_area/imm2/files/Numerical_Recipes.pdf

9- Marsaglia G. "The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness" http://www.stat.fsu.edu/pub/diehard/

10- Marsaglia G. "Random Number Generator" http://www.cs.pitt.edu/~kirk/cs1501/animations/Random.html

11- Marsaglia G. "The Monty Python method for generating random variables" http://portal.acm.org/citation.cfm?id=292395.292453

12- Marsaglia G. "The Ziggurat Method for Generating Random Variables" http://www.jstatsoft.org/v05/i08/paper.