HMMTEACHER: APOYO COMPUTACIONAL PARA LA ENSEÑANZA DE LAS CADENAS DE MARKOV OCULTAS EN BIOINFORMÁTICA.

Matias Fuentes Pérez, Escuela de Ingeniería Civil en Bioinformática, Universidad de Talca, mfuentesperez11@gmail.com
Camila Rojas Villalobos, Escuela de Ingeniería Civil en Bioinformática, Universidad de Talca, c.rojas.villalobos@hotmail.com
Gonzalo Riadi Mahias, Escuela de Ingeniería Civil en Bioinformática, Universidad de Talca, griadi@utalca.com

RESUMEN

En bioinformática, una de las tareas más comunes a realizar, es el análisis de secuencias biológicas. El fin es encontrar características propias de las moléculas que las secuencias de ácidos nucleicos y aminoácidos representan. Una de las técnicas más sofisticadas en el análisis de secuencias biológicas son las Cadenas de Markov Ocultas (HMM). Éstas se basan en la utilización de 3 algoritmos matemáticos, Forward, Backward y Viterbi, que responden respectivamente a las siguientes preguntas: ¿Cuál es la probabilidad de la secuencia?; En una posición específica de la secuencia, ¿cuál es la probabilidad de que esa observación sea emitido por un cierto estado oculto?, y ¿Cuál es la secuencia de estados ocultos más probables de la secuencia?. A pesar de ser una de las técnicas más sofisticadas, su utilización y entendimiento se ve limitado debido a la complejidad matemática que representan sus algoritmos. Para resolver esta limitación, se diseñó un software que recibe como input el modelo de un problema a través de HMMs, lo resuelve y entrega un informe detallado paso a paso de su resolución matemática. Este software tiene por objetivo acortar la brecha en el entendimiento de las HMM, por biólogos y estudiantes de bioinformática, para avanzar en el conocimiento científico.
Palabras Clave: HMM, Bioinformática, software, análisis de secuencias, Cadenas de markov ocultas.

LA BIOINFORMÁTICA

Actualmente, los desarrollos Bioinformáticos avanzan a grandes pasos junto con la vanguardia de las nuevas tecnologías, siendo capaces de utilizar el desarrollo de software de mejor manera debido a que el hardware mejora cada año tanto su calidad y capacidad. Existe un gran número de aplicaciones desarrolladas para el análisis de datos relacionados a la Bioinformática. Entre las más remarcables, están las bases de datos biológicas como; PDB[6, 8, 7], Uniprot[2], Kegg[28] y Genbank [4, 5]. Además de las bases de datos biológicas, existen herramientas como BLAST (Basic Local Alignment Search Tool)[30], que es utilizado para buscar patrones, o secuencias problemas, en una base de datos de secuencias de nucleótidos o aminoácidos. BLAST es una de las herramientas Bioinformáticas más utilizadas para descubrir la procedencia de secuencias desconocidas o para asociar una secuencia de la cual no se tiene ninguna información con aquellas que se conocen de antemano[30].

Una gran variedad de software se han desarrollado hasta ahora, tales como visualizadores (Artemis, IGV), de alineamiento (Bowtie, Bowtie2, TopHat), ensambladores (Trinity, Velvet assembler), que miden calidad (FastQ, PrintSec) y otros predictivos basados principalmente en los Modelos de Markov oculto (HMMER, HMMConverter, Hammock, SoDA2). En los últimos casi 30 años, la utilización de los Modelos de Markov Oculto (HMM) ha tenido una

excelente llegada por parte de los científicos, desarrollando herramientas que pueden identificar motivos de una secuencia, predecir estructuras de genes, realizar alineamientos múltiples, buscar en bases de datos de proteínas, e incluso el modelamiento de familias de proteínas y genes.

LAS HMM

Uno de los métodos más utilizados en la bioinformática para el análisis de secuencias biológicas son las HMM. Para entender qué son las HMM, primero se debe comprender el concepto de las Cadenas de Markov (Markov Chain). Las Cadenas de Markov, son una serie de sucesos estocásticos, que se consideran ”independientes” de los sucesos anteriores y solo dependen del que está justo después, con probabilidades propias de ocurrencia. Son los modelos más simples dentro de las probabilidades, y se pueden representar como la siguiente ecuación:

$$P(X_{n+1}) = x_{n+1}|X_n = x_n\ , X_{n-1} = x_{n-1}\ , X_{n-2} = x_{n-2}\ ..., X_2 = x_2\ ,\ X_1 = x_1 == P(X_{n+1}) = x_{n+1}|X_n = x_{\,n}.$$

Según como lo describe Eddy et. al., ”la idea clave es que una HMM es un modelo finito que describe una distribución de probabilidades sobre un infinito número de posibles secuencias”. Éstas se componen de un número de estados, los cuales pueden corresponder a las posiciones de una estructura tridimensional, o columnas en un alineamiento múltiple, y que cada estado emite un símbolo de acuerdo a sus probabilidades de emisión, creando una secuencia observable de símbolos, y que éstas probabilidades están conectadas por unas probabilidades de estados transitorios[9]. Otra definición es que, las HMM son similares a las cadenas de Markov, pero son más generales, y por este motivo más flexibles, permitiendo así modelar fenómenos que no son posibles modelar lo suficientemente bien con las Cadenas de Markov [14]. Además se menciona también que “las HMM son modelos estadísticos, en el cual el sistema a ser modelado se asume que es o se comporta como un proceso de Markov con estados inobservables (estados ocultos)” [10]. ”Las HMM son modelos de Markov de tiempo discreto con algunas características extras. La adición principal es que cuando un estado es ’visitado’ por la cadena de Markov, el estado ’emite’ una letra perteneciente a un alfabeto independiente de tiempo” [14]. ”Cuando una HMM está en funcionamiento hay, primero, una secuencia de estados observados, y segundo, una secuencia de símbolos emitidos” [14]. Por lo que el proceso se puede visualizar como se muestra en la figura 1.

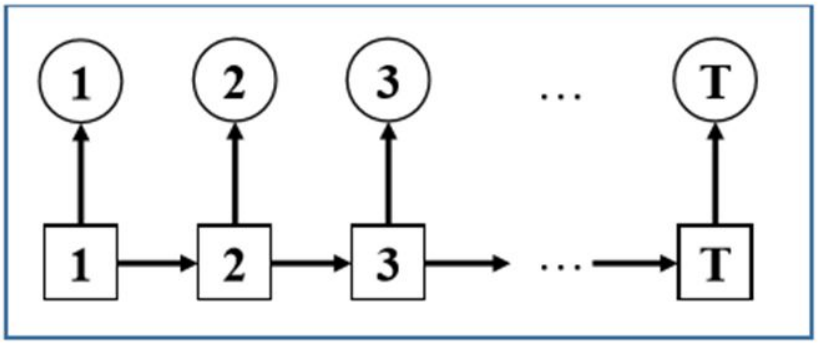


Figura 1: Representación de una HMM. Los cuadrados representan estados ocultos y los círculos representan estados observados. Las flechas verticales indican las emisiones y las horizontales las transiciones.

Parte del trabajo de un Bioinformático, y lo que es algo muy común, es el análisis de secuencias biológicas, ya sea de aminoácidos, DNA u otras, con el fin de predecir su función, estructuras, sitios de unión, características. Las HMM son una de las técnicas más sofisticadas para el análisis de secuencias biológicas, con las cuales, a través de sus tres algoritmos, se pueden responder tres preguntas específicas; ¿Cuál es la probabilidad de que la secuencia observada ocurra en la realidad?, conocido como algoritmo Forward; ¿Cuál es la probabilidad de encontrar cierto estado oculto en una posición específica de la secuencia?, llamado algoritmo Backward; y el último algoritmo, llamado Viterbi que responde a ¿cuál es la secuencia de estados ocultos más probable de ocurrir?[22].

Algoritmo Forward:

Inicilización (1.1)

$$\alpha(0,0) = 1$$

$$\alpha(t,0) = 0$$

$$\alpha(t=1,i) = \pi_i * e_s(O_{t=1}) \qquad \forall i$$

Recursión $(t = 2...L)$ (1.2)

$$\alpha(t,i) = e_{q_t=S_i}(O_t) * \sum_{j=1}^{N} \alpha(t-1,j) * \alpha_{ji} \qquad \forall i$$

Terminación (1.3)

$$P(O) = \sum_{j=1}^{N} \alpha(L,i)$$

Algoritmo Backward:

$$P(q_1 = S_k|O) = \frac{\alpha(t,k)*\beta(t,k)}{P(O)} \quad (2.1)$$

$$\beta(t-1,i) = \sum_{j=1}^{N} \alpha_{ij} * e_{q_t=S_j}(O_t) * \beta(t,j) \quad (2.2)$$

Algoritmo Viterbi:

Inicialización (t=0) *(3.1)*

$$\delta_0(S_0) = 1$$

$$\delta_0(S_i) = 0 \qquad \forall i > 0$$

$$\delta_{t=1}(S_i) = \pi_i * e_{q_1=S_i}(O_1) \qquad t = 1,\ 1 \le i \le N$$

Recursión (t=2...L) *(3.2)*

$$\delta_{t=1}(S_i) = e_{q_t=S_j}(O_t) * max\left[\delta_{t-1}(S_i) * a_{ij}\right] \qquad 2 \le t \le L \qquad 1 \le j \le N$$

$$ptr_t(S_j) = arg\ max\left[\delta_{t-1}(S_i) * a_{ij}\right]$$

Terminación *(3.3)*

$$P(O\ y\ Q) = max\left[\delta_L(S_i) * a_{i0}\right]$$

$$q^*_L = arg\ max\left[\delta_L(S_i) * a_{i0}\right]$$

Backtracking (t= L...1) *(3.4)*

$$q^*_{t-1} = ptr_t(S_j)$$

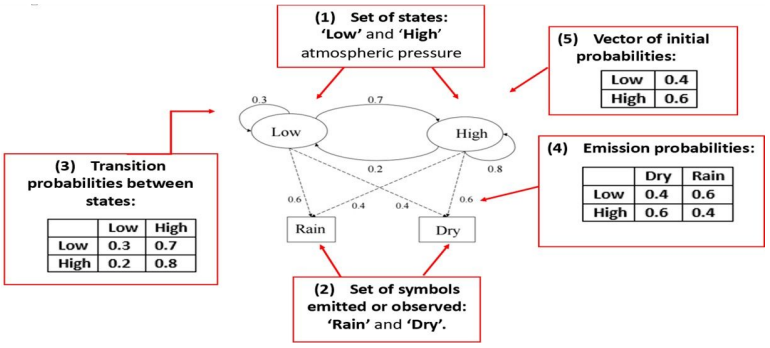


Figura 2: Ejemplo de un problema modelado. Cada elemento de una HMM se encuentra en una casilla numerada.

En la Figura 2 se observa un ejemplo de un problema modelado listo para ser resuelto con los algoritmos de las HMM, además se presentan los parámetros necesarios que debe tener un modelo para su posterior resolución, que son: 1. Estados Ocultos; 2. Estados Observados; 3. Matriz de probabilidades de transición; 4. Matriz de probabilidades de emisión y; 5. Vector de probabilidades iniciales.

La primera publicación relacionada a las HMM aplicada a la tecnología, específicamente para la detección de palabras, fue realizada por el ingeniero eléctrico Lawrence Rabiner[22] en 1989, donde explica, que son, la teoría detrás de las HMM, reconociendo las 3 preguntas principales que buscan responder los diferentes algoritmos de las HMM, Forward, Backward y Viterbi. No obstante, ya existían aproximaciones de lo que son las HMM y sus posibles usos en los años 60s y 70s. Andrei Markov dio su nombre para la teoría matemática de los procesos de Markov a principios de siglo veinte[21], pero fue Baum y sus colegas los que desarrollaron la teoría de las HMM en los 60s[3]. Por otra parte, también en el año 1989, se publica la primera utilización de las HMM en la Bioinformática por el investigador Churchill G. A. et al.[10], en el área del estudio de las secuencias de DNA. Las HMM, son una de las técnicas más sofisticadas para el análisis de secuencias biológicas aplicado a la Bioinformática. Actualmente, las HMM están siendo cada vez más utilizadas en diferentes áreas de la ciencia, desde el reconocimiento de voz[15, 22, 23], análisis de imágenes[1], en aplicaciones Bioinformáticas como el modelado de proteínas[16, 18, 27], alineamiento y análisis de secuencias biológicas[11, 13, 17], análisis filogenéticos [20, 25], identificación de regiones codificantes en genes[19, 29], hasta incluso la predicción del clima[24].

EL DESAFÍO DE LAS HMM

Las HMM son cada día más y más utilizadas, especialmente en el área de la Bioinformática. Pero a pesar de éste incremento de su uso en los últimos 20 años, no es trivial dominarlas. La dificultad de las HMM radica en la resolución de los algoritmos matemáticos, que requieren un nivel avanzado de aplicación matemática, que los biólogos no desarrollan en su formación. Actualmente los softwares que existen que tienen relación con las HMM se encargan solo de resolver el problema sin explicar al usuario cómo se llega a esa solución. Cabe destacar que existe interés de la comunidad científica por usar y aprender las HMM pero se ven limitados por la complejidad que presentan. Para los biólogos, aprender a modelar un problema es relativamente sencillo, e interpretar los resultados tiene una dificultad similar. El problema principal al que se ven enfrentados radica en la teoría

matemática necesaria detrás de cada algoritmo. Por este motivo, existe la necesidad de generar un método de enseñanza de las HMM de una manera menos compleja que la actual.

LA PROPUESTA

Con el fin de afrontar la principal dificultad que presenta la utilización de las HMM, se desarrolló un software que recibe como input, los parámetros necesarios obtenidos desde un problema previamente modelado con HMMs, para luego utilizarlos en cada uno de los algoritmos anteriormente mencionados, resolviendo así la problemática matemáticamente, entregando finalmente, un informe detallado de cada paso realizado hasta llegar a la solución final de uno o más algoritmos.

METODOLOGÍA

El núcleo para el desarrollo de éste software es la implementación de las ecuaciones matemáticas pertenecientes a cada algoritmo que responde una pregunta en las HMM. Estas ecuaciones fueron obtenidas desde: [14], [22], [25]; para luego transformarlas a código e implementarlas en el software. Este proceso adoptó la metodología iterativa incremental propia del desarrollo de software, con el fin de que la implementación realizada sea lo más optimizada posible.

El lenguaje de programación utilizado es el Java, debido a que es un lenguaje multiplataforma, además de poseer la capacidad de generar interfaces gráficas. Este lenguaje fue utilizado mediante la IDE llamada Eclipse en su versión Juno (4.2), con el fin de facilitar la programación y optimizar los tiempos de la misma.
El cuerpo del software se programó en base al modelado del flujo de información requerida para la resolución de las HMM, identificado en la figura 3.

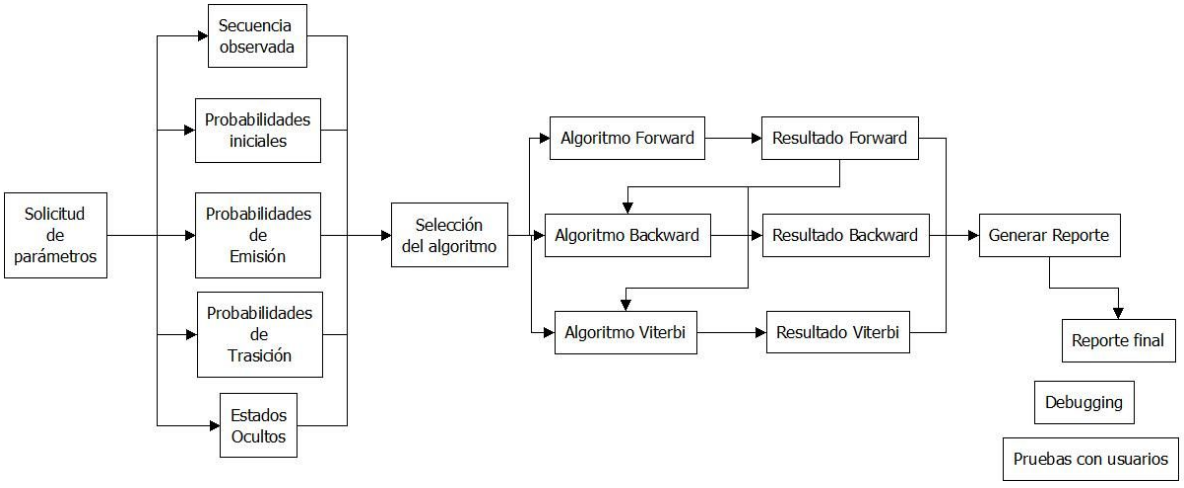


Figura 3. Diagrama de información para la resolución de las HMM.

RESULTADOS

Se generó un software capaz de resolver matemáticamente las HMM, entregando un informe final con el detalle de ésta resolución. Se generaron 5 ventanas particulares para así guiar al usuario en el proceso de la utilización del software. Al iniciar el software se muestra la ventana de presentación (figura 4), donde se indica el objetivo del software,

además de opciones como la de “Beginner mode”, que permite la observación de tips desplegables tipo tutorial en cada sección donde el usuario debe ingresar datos. La siguiente ventana(Figura 5) corresponde a aquella donde el usuario debe ingresar la secuencia de estados observados y estados ocultos, además tiene la opción de elegir una secuencia predeterminada o ingresar una nueva secuencia. La opción “Clear Data” eliminará toda la información que se haya ingresado previamente.

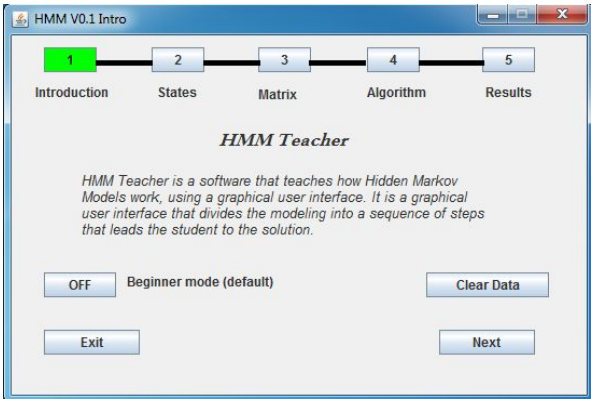


Figura 4. Ventana de presentación y objetivo del software.

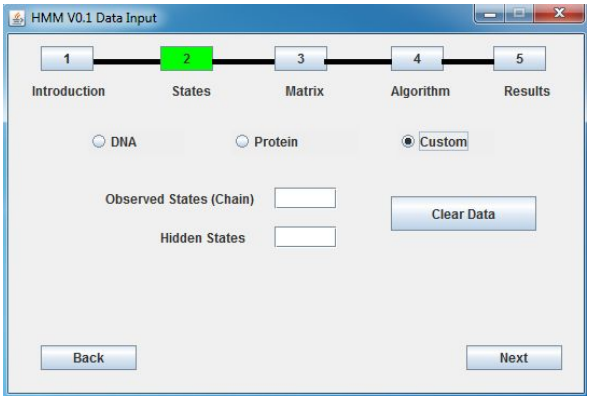


Figura 5. Ventana de inserción de estados observados y ocultos.

La tercera parte corresponde a la inserción de las matrices de probabilidades (Figura 6) del modelo del problema. Aquí, el usuario debe rellenar cada casilla que se observe en la pantalla. Al ser probabilidades, la suma de las casillas de manera vertical deberá ser máximo de 1, mientras que horizontalmente no habrá restricción. Se añadió la opción de que el usuario no tenga que ingresar los valores, con el botón “Random”, en donde el programa agrega números de manera aleatoria en las casillas, siguiendo las reglas de las probabilidades.

Luego de ingresar todos los datos provenientes desde el modelado del problema, se debe hacer la elección del algoritmo/pregunta que se desea responder. Para ello, la ventana número 4 corresponde a la elección del(los) algoritmo(s) a resolver(Figura 7). El programa da la opción de la selección de 0, 1, 2 o 3 algoritmos a resolver. En el caso de que se 0, automáticamente el programa ejecutará todos los algoritmos. Para la elección del algoritmo Backward, se requiere de más información, por lo que el software habilitará las opciones de más abajo.

Por último, la ventana 5 del software corresponde a la confirmación de los datos ingresados y la petición de resolver el problema y generar el informe final, todo eso en el único botón llamado “Generate PDF”(Figura 8).

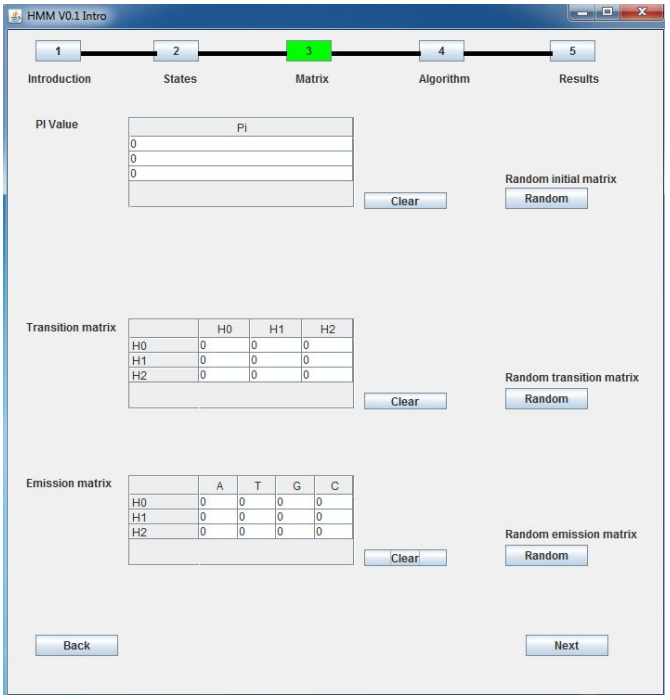


Figura 6. Ventana de solicitud de matrices de probabilidades.

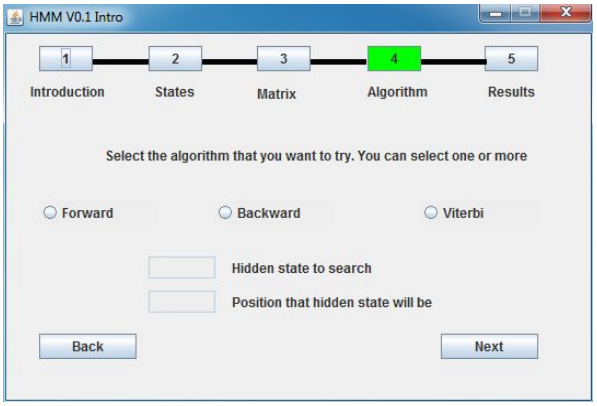


Figura 7. Ventana de selección de algoritmo a resolver.

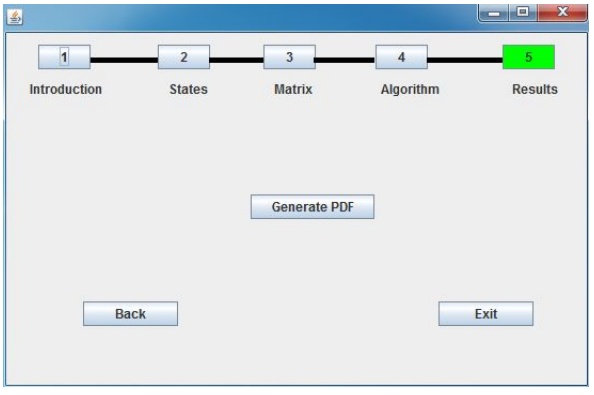
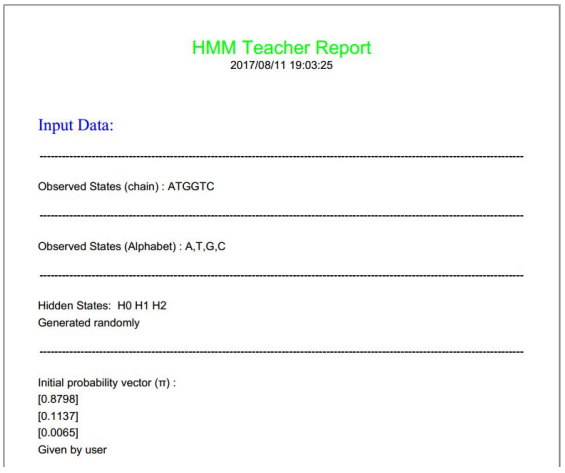


Figura 8. Generación de reporte.

El último paso del software generará un informe en formato pdf (Figura 9), el cual muestra datos básicos, como la fecha y hora del reporte, además de todos los datos ingresados en el software, indicando si fueron ingresados por el usuario o generados de manera aleatoria. Luego de mostrar los datos ingresados, procede a la sección de mostrar las ecuaciones utilizadas, así como también cada paso de la resolución de los algoritmos seleccionados previamente. Los pasos intermedios están claramente identificados mediante números, para que así, cada vez que se utilice uno de estos pasos, sea llamado mediante ese número característico (Figura 10).


HMM Teacher Report
2017/08/11 19:03:25

Input Data:

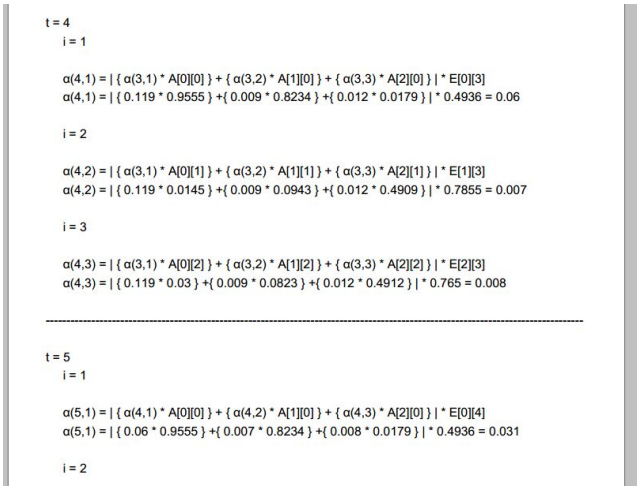
Observed States (chain) : ATGGTC

Observed States (Alphabet) : A,T,G,C

Hidden States: H0 H1 H2
Generated randomly

Initial probability vector (π) :
[0.8798]
[0.1137]
[0.0065]
Given by user

Figura 9. Reporte final.


t = 4
i = 1

α(4,1) = | { α(3,1) * A[0][0] } + { α(3,2) * A[1][0] } + { α(3,3) * A[2][0] } | * E[0][3]
α(4,1) = | { 0.119 * 0.9555 } +{ 0.009 * 0.8234 } +{ 0.012 * 0.0179 } | * 0.4936 = 0.06

i = 2

α(4,2) = | { α(3,1) * A[0][1] } + { α(3,2) * A[1][1] } + { α(3,3) * A[2][1] } | * E[1][3]
α(4,2) = | { 0.119 * 0.0145 } +{ 0.009 * 0.0943 } +{ 0.012 * 0.4909 } | * 0.7855 = 0.007

i = 3

α(4,3) = | { α(3,1) * A[0][2] } + { α(3,2) * A[1][2] } + { α(3,3) * A[2][2] } | * E[2][3]
α(4,3) = | { 0.119 * 0.03 } +{ 0.009 * 0.0823 } +{ 0.012 * 0.4912 } | * 0.765 = 0.008

---

t = 5
i = 1

α(5,1) = | { α(4,1) * A[0][0] } + { α(4,2) * A[1][0] } + { α(4,3) * A[2][0] } | * E[0][4]
α(5,1) = | { 0.06 * 0.9555 } +{ 0.007 * 0.8234 } +{ 0.008 * 0.0179 } | * 0.4936 = 0.031

i = 2

Figura 10. Pasos intermedios en el reporte final.

CONCLUSIONES

Se ha desarrollado un software con interfaz gráfica amistosa que permite a un estudiante de esta técnica de modelamiento de secuencias, las HMMs, resolver problemas sencillos y ver el desarrollo de los algoritmos hasta la respuesta final.

Para los alumnos de la carrera de Ingeniería Civil en Bioinformática de la Universidad de Talca, esta herramienta permitirá en el curso de “Modelos matemáticos aplicados a sistemas biológicos”, dedicar una mayor cantidad de tiempo al aprendizaje del modelado del problema biológico con HMMs que a la resolución del mismo, dejando que el software cumpla la tarea de interiorizar al alumnado en la resolución matemática del problema.

De esta manera se pretende acelerar el aprendizaje de esta técnica, por parte de estudiantes y profesionales de las ciencias, promoviendo el uso de las HMMs para modelar y resolver problemas biológicos.

FINANCIAMIENTO
Proyecto Fondecyt 11140869.

BIBLIOGRAFÍA

[1] Jia Li, Gray, Robert M, Image Segmentation and Compression Using Hidden Markov Models, 2000.

[2] Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, Maria J Martin, Darren A Natale, Claire O'Donovan, Nicole Redaschi, and Lai-Su L Yeh. UniProt: the Universal Protein knowledgebase. *Nucleic acids research*, 32(Database issue):D115–9, 2004.

[3] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[4] Dennis A. Benson, Karen Clark, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and Eric W. Sayers. GenBank. *Nucleic Acids Research*, 43(D1):D30–D35, 2015.

[5] Dennis A. Benson, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and David L. Wheeler. GenBank. *Nucleic Acids Research*, 33(DATABASE ISS.):34–38, 2005.

[6] H. M. Berman. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[7] Helen M Berman, Tammy Battistuz, T N Bhat, Wolfgang F Bluhm, E Philip, Kyle Burkhardt, Zukang Feng, Gary L Gilliland, Lisa Iype, Shri Jain, Phoebe Fagan, Jessica Marvin, David Padilla, Veerasamy Ravichandran, Narmada Thanki, Helge Weissig, and John D Westbrook. The Protein Data Bank. *Biological Crystallography*, 58:899–907, 2002.

[8] Frances C. Bernstein, Thomas F. Koetzle, Graheme J B Williams, Edgar F. Meyer, Michael D. Brice, John R. Rodgers, Olga Kennard, Takehiko Shimanouchi, and Mitsuo Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *Archives of Biochemistry and Biophysics*, 185(2):584–591, 1978.

[9] Phil Blunsom. Hidden Markov Models. *Lecture notes, August*, pages 1–7, 2004.

[10] G A Churchill. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology*, 51(1):79–94, 1989.

[11] Sean R. Eddy. Multiple alignment using hidden Markov models. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 3:114–120, 1995.

[12] Sean R. Eddy. Hidden Markov models. *Current Opinion in Structural Biology*, 6(3):361–365, 1996.

[13] Sr Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

[14] W. Ewens and Grant G. Statistical Methods in Bioinformatics: An Introduction. *Springer*, 2005.

[15] Mark Gales and Steve Young. The application of hidden Markov models in speech recognition. *Found. Trends Signal Process.*, 1(3):195–304, 2007.

[16] K Karplus, C Barrett, M Cline, M Diekhans, L Grate, and R Hughey. Predicting protein structure using only sequence information. *Proteins*, Suppl 3(May):121–125, 1999.

[17] Kevin Karplus, Christian Barrett, and Melissa Cline. Predicting protein structure using only sequence information. *Proteins: Structure,*, 125(May):121–125, 1999.
[18] Anders Krogh, Michael Brown, I.Saira Mian, Kimmen Sjolander, and David Haussler. Hidden Markov Models in Computational Biology: Applications to Protein Modeling, 1994.
[19] Alexander V. Lukashin and Mark Borodovsky. GeneMark.hmm: New solutions for gene finding. *Nucleic Acids Research*, 26(4):1107–1115, 1998.
[20] T. Mailund, J.Y. Dutheil, A. Hobolth, G. Lunter, and M.H Schierup. Estimating divergence time and ancestral effective population size of bornean and Sumatran orangutan subspecies using a coal escent hidden Markov model. *PLoS Computational Biology*, 7(e1001319), 2011.
[21] A. A. Markov. An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains. *Science in Context*, 19(04):591, 2006.
[22] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
[23] Lokesh Selvaraj and Balakrishnan Ganesan. Enhancing Speech Recognition Using Improved Particle Swarm Optimization Based Hidden Markov Model. *The Scientific World Journal*, 2014(i):1–10, 2014.
[24] Kriti Shrivastava, Ratna Wakle, and Mayur Nakade. Weather Prediction Using Hidden Markov Model. 2(3):61–63, 2015.
[25] A. Siepel and D Haussler. Phylogenetic Hidden Markov Models. In Statistical Methods in Molecular Evolution. *Springer*, pages 325–351, 2005.
[26] Linda B. Smith and Esther Thelen. Development as a dynamic system, 2003.
[27] E L Sonnhammer, G von Heijne, and A Krogh. A hidden Markov model for predicting transmembrane helices in protein sequences. *Proceedings*, 6:175–182, 1998.
[28] Mao Tanabe and Minoru Kanehisa. Using the KEGG database resource. *Current Protocols in Bioinformatics*, (SUPPL.38), 2012.
[29] Paula Tataru, Andreas Sand, Asger Hobolth, Thomas Mailund, and Christian N S Pedersen. Algorithms for hidden markov models restricted to occurrences of regular expressions. *Biology*, 2(4):1282–95, 2013.
[30] Tatiana A. Tatusova and Thomas L. Madden. BLAST 2 SEQUENCES, a new tool for comparing protein and nucleotide sequences. *FEMS Microbiology Letters*, 174(2):247–250, 1999.