UNA INTRODUCCIÓN A LA PREDICCIÓN DE LA ESTRUCTURA SECUNDARIA DEL RNA MEDIANTE MÉTODOS ESTOCÁSTICOS

Steven Van Vaerenbergh
Dpt. of Electronics and Information Systems
Ghent University, Belgium
e-mail: steven.vanvaerenbergh@rug.ac.be

Luis Vielva
Grupo de Tratamiento Avanzado de Señal, DICOM
Universidad de Cantabria
e-mail: luis@gtas.dicom.unican.es

***Abstract*— This article comments on an ongoing investigation, the prediction of RNA secondary structure using stochastic methods, in particular stochastic context-free grammars. While the investigation in this field has already made a lot of progress and is currently refining and improving its methods, this article is meant to provide an introduction to this subject for researchers in the digital signal processing area. After situating the problem in its biological context, we explain the basics of transformational grammars, which are used to model the RNA secondary structure. Then we present the three basic problems for these structures, and explain the three main algorithms to solve them, relating these to the analogous algorithms for hidden Markov models.**

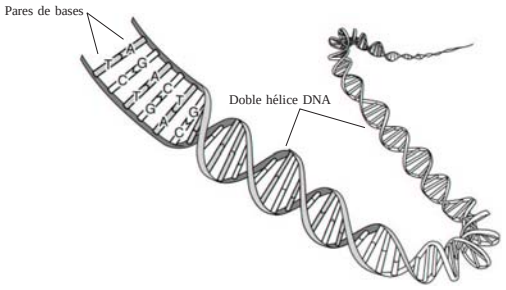

Fig. 1. La doble hélice del DNA.

I. INTRODUCCIÓN: BIOLOGÍA MOLECULAR

Las células son las unidades funcionales básicas de todo ser vivo. Todas las instrucciones necesarias para dirigir sus actividades se encuentran codificadas dentro de su DNA (ácido desoxirribonuleico). El DNA de todos los organismos está formado por los mismos componentes y se organiza como una macromolécula con dos cadenas entrelazadas formando la famosa doble hélice de la Figura 1. Cada una de las cadenas es una secuencia lineal de las cuatro unidades básicas (nucleótidos) posibles; cada una de ellas caracterizada por una base distinta: adenina (A), timina (T), citosina (C) y guanina (G).

Las cadenas se mantienen unidas mediante enlaces débiles entre las bases, de forma que la A en una cadena se enlaza con la T en la opuesta y la C con la G. Este emparejamiento complementario hace que ambas cadenas contengan la misma información, y explica el mecanismo básico de la duplicación de células (ambas cadenas se separan y se crean dos réplicas mediante atracción de las bases complementarias) y de la expresión de genes (una sección —un gen— de una cadena atrae a las bases complementarias y se crea una molécula lineal de cadena sencilla que codifica la proteína a sintetizar). Esta molécula de cadena sencilla se denomina RNA (ácido ribonucleico) y está compuesta de las mismas cuatro bases que el RNA, salvo que el uracilo (U) sustituye a la timina. La molécula de RNA así creada (denominada RNA mensajero) viaja hasta los ribosomas (estructuras formadas a partir de RNA estructural) que se encargan de la transcripcción mediante la ayuda del RNA de transferencia, que selecciona uno de los veinte aminoácidos distintos por cada tres bases (un codón), y los ensambla en moléculas conocidas como proteínas. Esta asociación entre tripletas de bases y aminoácidos es universal para todos los organismos y recibe el nombre de código genético.

Esta descripción pone de manifiesto la importancia fundamental del RNA en los procesos básicos de la vida. Además de los descritos, el RNA desempeña también un papel crítico en la infección por retrovirus (como el virus de la inmunodeficiencia humana), y es capaz de catalizar y regular muchas funciones celulares básicas.

Mientras que las cadenas de DNA se estructuran como una doble hélice, la molécula de RNA se pliega sobre sí misma creando enlaces entre bases complementarias de la propia cadena, dando lugar a lo que se conoce como estructura secundaria del RNA, que desempeña un importante papel en los procesos reguladores, catalíticos o estructurales de la célula. Además, las moléculas de RNA que desempeñan funciones similares en distintos organismos tienden más a conservar su estructura secundaria que su composición lineal específica, que ha podido ir cambiando a lo largo de la evolución a través del árbol filogenético. Por lo tanto, el ser capaces de predecir la estructura secundaria de una molécula de RNA a partir de la secuencia de bases que lo componen despierta un enorme interés en el campo de la bioinformática [1]. En la Figura 2 se muestra una representación de dicha estructura secundaria

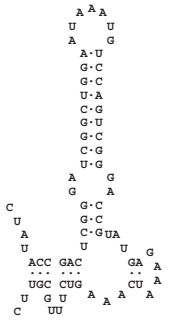

Fig. 2. Ejemplo de estructura secundaria del RNA: parte de la subunidad ribosómica de la molécula de Tetrahymena Bergeri. Los puntos marcan bases enlazadas.

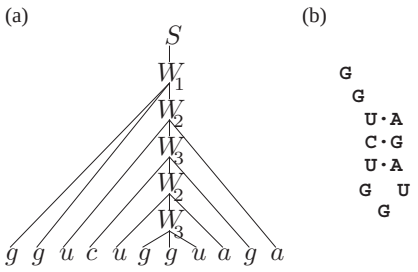

Fig. 3. (a) Árbol de análisis para $ggucugguaga$. (b) Estructura secundaria del RNA correspondiente, dónde se observan los enlaces generados por las reglas de transformación para W_2 y W_3.

para una subsección de una molécula de RNA estructural.

El análisis de secuencias biológicas, como el DNA, el RNA o las proteinas, puede realizarse mediante métodos estocásticos como los modelos ocultos de Markov (HMM) [2], [3] o las gramáticas estocásticas independientes del contexto (SCFG).

II. Gramáticas transformacionales

Existen tres problemas básicos en la utilización de los HMMs [3], existiendo para cada uno de ellos algoritmos eficientes: 1) Calcular la probabilidad de una secuencia de observaciones (algoritmos *forward* y *backward*). 2) Encontrar la secuencia de estados más probable (algoritmo de Viterbi). 3) Entrenar el modelo a partir de las observaciones (algoritmo de Baum-Welch, que es un caso especial del EM, *expectation-maximization*).

Los HMM tienen múltiples aplicaciones en bioinformática [4], [5], tales como el aliniemiento de secuencias, la creación de perfiles para familias de secuencias, o la identificación de genes. Sin embargo, no son muy apropiados para el modelado de la estructura secundaria del RNA, ya que no son capaces de modelar la interacción de largo alcance entre los nucleótidos, como los enlaces conservados entre posiciones no contíguas del RNA. Lo que se necesita es herramientas más versatiles que los HMM, capaces de incorporar en su estructura interna las restricciones propias de la estructura secundaria del RNA. Dichas herramientas se encuentran dentro del ámbito de las *gramáticas transformacionales*, de las que los HMM forman parte.

Alrededor de 1950, el linguista Noam Chomsky comenzo a formalizar los mecanismos generación de los lenguajes naturales [6], [7], [8]. Según la definición de Chomsky, una gramática transformacional consiste en un conjunto de *símbolos* y un conjunto de *reglas de transformación*. Los símbolos pueden ser tanto *terminales* (los que aparecen realmente en los mensajes) como *no terminales* (los que únicamente se necesitan como pasos intermedios durante la transformación de las cadenas). Las reglas de transformación tienen el aspecto $\alpha \rightarrow \beta$, donde α y β son cadenas de símbolos.

Consideremos a modo de ejemplo una gramática transformacional con los siguientes elementos: 1) Los terminales $\{a, g, c, u\}$. 2) Los no terminales $\{S, W_1, W_2, W_3\}$. 3) Las de reglas de transformación $\{S \rightarrow W_1, W_1 \rightarrow ggW_2, W_2 \rightarrow uW_3a, W_3 \rightarrow cW_2g, W_3 \rightarrow ggu\}$. Observemos que existen dos reglas de transformación posibles para W_3, de modo que podemos escoger cualquiera de ellas. La secuencia de nucleótidos $ggucugguaga$ podría generarse mediante la aplicación de las siguientes reglas:

$$S(\text{comienzo, no terminal}) \rightarrow W_1\,(\text{regla } 1) \rightarrow ggW_2\,(\text{regla } 2)$$
$$\rightarrow gguW_3a\,(\text{regla } 3) \rightarrow ggucW_2ga\,(\text{regla } 4)$$
$$\rightarrow ggucuW_3aga\,(\text{regla } 3) \rightarrow ggucugguaga\,(\text{regla } 5).$$

Dada una gramática podemos preguntarnos si una secuencia dada pudo generarse mediante sus reglas de transformación. El análisis gramatical de una secuencia consiste en encontrar una derivación válida. El análisis de una secuencia puede llevarse a cabo mediante un *alineamiento* de la secuencia con la gramática, y su resultado puede representarse mediante un árbol de análisis, en el que la raíz es el no terminal S, los nodos internos son otros símbolos no terminales y las hojas son los símbolos terminales. En la Figura 3 se representa el árbol para la secuencia del ejemplo anterior.

A. Jerarquía de Chomsky de gramáticas transformacionales

Chomsky [6] describió cuatro tipos de gramáticas, en base a los tipos de restricciones sobre sus reglas de producción. Desde la más restrictiva a la más genérica son:

1) Gramáticas regulares: sólo permite reglas de la forma $W_1 \rightarrow aW_2$ y $W_1 \rightarrow a$, donde W_1 y W_2 representan cualquier no terminal, y a cualquier terminal.
2) Gramáticas independientes del contexto (CFG): se admite $W \rightarrow \alpha$, donde α puede ser cualquier cadena, conteniendo tanto terminales como no terminales.
3) Gramáticas sensibles al contexto: se admite $\alpha_1 W \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$. La transformación del no terminal W depende de su contexto, las subcadenas α_1 y α_2.
4) Gramáticas sin restricciones: admiten $\alpha_1 W \alpha_2 \rightarrow \gamma$.

Las gramáticas regulares son básicamente gramáticas *de izquierda a derecha*. No son apropiadas para modelar interacciones de largo alcance entre elementos de la cadena. Las gramáticas independientes del contexto, por el contrario,

pueden incorporar reglas para generar secuencias con interacciones de largo alcance, como por ejemplo secuencias palindrómicas. La característica interesante de este tipo de gramáticas para el análisis de la estructura secundaria del RNA es que las uniones entre bases complementarias obedecen a una estructura pseudo palindrómica (en la que la cadena inversa tiene las bases complementarias de la directa).

Todas las gramáticas de la jerarquía de Chomsky admiten una versión estocástica que permite su utilización en el modelado probabilistíco de secuencias. A cada regla de producción se le asigna una probabilidad. Un hecho relevante es que los HMM son completamente equivalentes a las gramáticas regulares estocásticas. La versión estocástica de las CFG, denominada SCFG, es por tanto una generalización de los HMM.

III. Modelado de secuencias mediante SCFG

Al igual que con los HMM, existen tres problemas básicos en las SCFG y tres algoritmos que los resuelven: 1) Calcular el alineamiento óptimo entre una secuencia y una SCFG parametrizada (algoritmo CYK) 2) Calcular la probabilidad de una secuencia dada una gramática (algoritmos *inside* y *outside*). 3) Estimar los parámetros óptimos de una SCFG a partir de un conjunto de secuencias (algoritmo *inside-outside*).

Para analizar estos algoritmos, es útil suponer que las gramáticas están expresadas en la *forma normal* de Chomsky, que requiere que todas las reglas de producción sean de la forma $W_v \rightarrow W_y W_z$ o $W_v \rightarrow a$, donde a es cualquier terminal. Siempre es posible expresar una SCFG mediante su forma normal [6], [9].

A. El algoritmo inside

Supongamos una SCFG en forma normal de Chomsky con M no terminales $W_1, \ldots, W_M$ con la que pretendemos analizar la secuencia x, que está compuesta por L símbolos terminales $x_1, \ldots, x_L$.

Definimos $\alpha(i, j, v)$ como la probabilidad de generar la subsecuencia $x_i, \ldots, x_j$ mediante un árbol cuya raíz es el no terminal W_v. Calcularemos dicha probabilidad para todos los valores de i, j, v, mediante un algoritmo de programación dinámica que utiliza una matriz tridimensional $L \times L \times M$.

El cálculo es recursivo, comenzando con secuencias de longitud 1 ($i = j$). En el paso de inicialización, calculamos todas las probabilidades de que una secuencia de un terminal en la posición i sea generada a partir de un no terminal W_v según la segunda regla de la forma normal de Chomsky. La probabilidad para la regla de producción $W_v \rightarrow a$ se denomina *probabilidad de emisión*, $e_v(a) = P(W_v \rightarrow a)$.

En el paso recursivo, calculamos la probabilidad de generar la subsecuencia $x_i, \ldots, x_j$ a partir de W_v según la primera regla de la forma normal de Chomsky. Es decir, W_y y W_z (los dos "hijos" de W_v) generan a su vez respectivamente las secuencias $x_i, \ldots, x_k$ y $x_{k+1}, \ldots, x_j$, tal y como se muestra en la Figura 4. Durante este paso la longitud de la secuencia $j - i + 1$ se incrementa en cada iteración hasta alcanzar L. La probabilidad de la regla de producción $W_v \rightarrow W_y W_z$ se denomina *probabilidad de transición*, $t_v(y, z) = P(W_v \rightarrow W_y W_z)$.

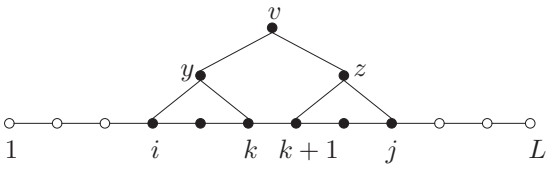

Fig. 4. Un paso de la iteración del cálculo *inside* de $\alpha(i, j, v)$. Los círculos blancos no se consideran en este paso.

Algoritmo 1: inside

for $i = 1$ **to** L, $v = 1$ **to** M

$\alpha(i, i, v) = e_v(x_i)$

for $i = L - 1$ **downto** 1, $j = i + 1$ **to** L, $v = 1$ **to** M

$$\alpha(i, j, v) = \sum_{y=1}^{M} \sum_{z=1}^{M} \sum_{k=i}^{j-1} \alpha(i, k, y)\alpha(k + 1, j, z)t_v(y, z)$$

$$P(x|\Theta) = \alpha(1, L, 1)$$

En el último paso se obtiene $\alpha(1, L, 1)$. Ésta es la probabilidad de que se haya generado la secuencia completa a partir del primer no terminal.

B. El algoritmo outside

Si el algoritmo *inside* calcula primero la probabilidad para símbolos sencillos y después expande *desde dentro* para *incluir* subárboles de análisis, el algoritmo *outside* comienza *desde fuera* con la secuencia completa x y *excluye* subárboles de análisis, como se muestra la Figura 5.

Definimos como $\beta(i, j, v)$ la probabilidad del árbol completo con raíz en el no terminal inicial para la secuencia completa x, excluyendo todos los subárboles posibles para las subsecuencias $x_i, \ldots, x_j$ con raíz en el no terminal W_v.

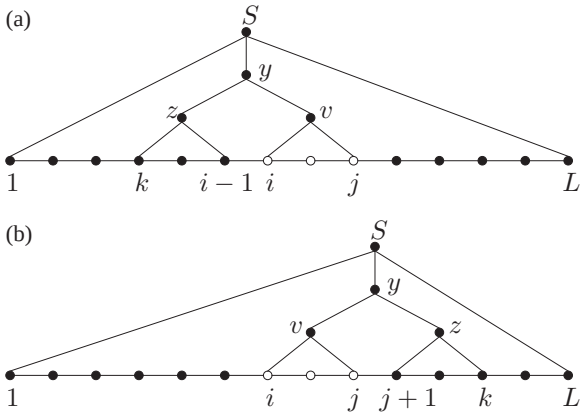

Fig. 5. Algoritmo *outside*: los puntos blancos representan la subsecuencia excluida. (a) En la primera parte de las contribuciones a $\beta(i, j, v)$ sólo se consideran los puntos negros a la izquierda de la subsecuencia de puntos blancos $x_i, ..., x_j$. (b) Para la segunda parte de las contribuciones se utilizan los símbolos a la derecha de los puntos blancos.

Algoritmo 2: outside

$\beta(1, L, 1) = 1$
for $v = 2$ **to** M
  $\beta(1, L, v) = 0$

for $s = L - 1$ **downto** 1, $j = L - s$ **to** L
  $i = j - s + 1$
  for $v = 1$ **to** M

$$\beta(i,j,v) = \sum_{y=1}^{M}\sum_{z=1}^{M}\sum_{k=1}^{i-1}\alpha(k,i-1,z)\beta(k,j,y)t_y(z,v) + \sum_{y=1}^{M}\sum_{z=1}^{M}\sum_{k=j+1}^{L}\alpha(j+1,k,z)\beta(i,k,y)t_y(v,z)$$

$P(x|\Theta) = \sum_{v=1}^{M}\beta(i,i,v)e_v(x_i)$, $\forall i$

Algoritmo 3: CYK

for $i = 1$ **to** L, $v = 1$ **to** M
  $\gamma(i, i, v) = \log(e_v(x_i))$
  $\tau(i, i, v) = (0, 0, 0)$

for $i = L - 1$ **downto** 1, $j = i + 1$ **to** L, $v = 1$ **to** M
  $\psi(i, j, k, v, y, z) = \tau(i, k, y) + \tau(k+1, j, z) + \log(t_v(y, z))$
  $\gamma(i, j, v) = \max_{(y,z,k),k=i\ldots j-1} \psi(i, j, k, v, y, z)$
  $\tau(i, j, v) = \operatorname{argmax}_{(y,z,k),k=i\ldots j-1} \psi(i, j, k, v, y, z)$

$\log(P(x|\Theta)) = \gamma(1, L, 1)$

C. Reestimación de parámetros

Combinando las α y β pueden reestimarse los probabilidades de una SCFG. Partimos de una elección inicial para las matrices estocásticas de emisión y transición, que puede ser aleatoria. Calculamos las α y β correspondientes y definimos la función $c(v)$ como el valor esperado de veces que el no terminal W_v se utiliza en una producción:

$$c(v) = \frac{1}{P(x|\Theta)}\sum_{i=1}^{L}\sum_{j=1}^{L}\alpha(i,j,v)\beta(i,j,v).$$

El número esperado de veces que se usa la regla de producción $W_v \to W_y W_z$ es

$$c(v \to yz) = \frac{1}{P(x|\Theta)} \times \sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\sum_{k=1}^{j-1}\beta(i,j,v)\alpha(i,k,y)\alpha(k+1,j,z)t_v(y,z).$$

La ecuación EM de reestimación de las reglas de producción es por tanto

$$\hat{t}_v(y,z) = \frac{c(v \to yz)}{c(v)}.$$

Análogamente podemos hacer para las reglas de producción del segundo tipo, $W_v \to a$,

$$\hat{e}_v(a) = \frac{c(v \to a)}{c(v)} = \frac{\sum_{i|x_i=a}\beta(i,i,v)e_v(a)}{\sum_{i=1}^{L}\sum_{j=1}^{L}\alpha(i,j,v)\beta(i,j,v)}.$$

Una vez estimadas las probabilidades, se recalculan α y β y se repite el proceso hasta la convergencia.

D. El algoritmo CYK

El problema del alineamiento para las SCFG se resuelve mediante el algoritmo de Cocke-Younger-Kasami (CYK). Calcula la matriz $\gamma(i, j, v)$ que permite obtener $\log P(x, \hat{\pi}|\Theta)$, donde $\hat{\pi}$ es el árbol más probable. Para recorrer hacia atrás la matriz de programación dinámica tridimensional y obtener el árbol más probable, se utiliza la variable $\tau(i, j, v)$.

En la iteración, la maximización según k puede interpretarse como la *partición óptima* (la más probable) de la subsecuencia en dos partes, ya que k determina qué subsecuencias acabarán generando los no terminales W_y y W_z, como se muestra en la Figura 4. La maximización según y (z) devolverá el no terminal W_y (W_z) con la mayor probabilidad de generar la subsecuencia $x_i, \ldots, x_k$ ($x_{k+1}, \ldots, x_j$). Por lo tanto, la tripleta (y, z, k) contiene los índices óptimos para obtener la subsecuencia $x_i, \ldots, x_j$ a partir del no terminal W_v.

IV. Conclusiones

Las gramáticas estocásticas independientes del contexto son una generalización de los modelos ocultos de Markov que permiten incorporar las restricciones asociadas a la estructura secundaria del RNA. Los algoritmos presentados no garantizan siempre la predicción correcta de dicha estructura, pero forman la base de algoritmos más sofisticados, que incorporar características tales como leyes biofísicas que modulan el plegamiento del RNA. Se trata de un campo de investigación en bioinformática que se encuentra en sus comienzos, apareciendo constantemente mejoras sobre los métodos existentes, por lo que esta comunicación puede considerarse como una introducción al tema para los investigadores en el campo del procesado de señal.

References

[1] S. R. Eddy, R. Durbin *RNA Sequence Analysis Using Covariance Models*, Nucleic Acids Res., 22, 2079-2088 (1994).

[2] R. Dugad, U. B. Desai, "A Tutorial on Hidden Markov Models", Signal Processing and Artificial Neural Networks Laboratory, Department of Electrical Engineering, Indian Institute of Technology, Bombay, Technical Report No.: SPANN 96.1., May 1996.

[3] L.R. Rabiner, B. H. Juang, "An Introduction to Hidden Markov Models", IEEE ASP Mag., pp 4-16, June 1986.

[4] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, *Biological Sequence Analysis*, Cambridge University Press (1998).

[5] M.S. Waterman, *Introduction to Computational Biology*, CRC Press (1995).

[6] N. Chomsky, "Three models for the description of language", IRE Transactions on Information Theory, 2, 113–124.

[7] D. B. Searls, "The Linguistics of DNA", American Scientist, 80:579-591 (1992).

[8] D. B. Searls, "The Language of Genes", Nature, vol. 420, november 2002.

[9] Y. Sakakibira, M. Brown, R. Underwood, I. S. Mian, and D. Haussler, *Stochastic Context-Free Grammars for Modeling RNA*, Nucleic Acids Res., 22, 5112-5120 (1999).