Resumen—Los micro-ARN (miARN) son pequeñas moléculas de ARN endógeno que actúan como moléculas reguladoras de la transcripción génica mediante la degradación del ARN mensajero o la inhibición de la traducción, estos puede controlar un gran número de genes diana , cuya caracterización puede ser crucial para identificar el papel que los miARN pueden ejercer como oncogenes o genes supresores de tumores. Encontrar si un miARN se une a la región complementaria de alguna secuencia diana de ARN mensajero por medio de herramientas computacionales puede ser de gran ayuda en el futuro para investigar en la practica si este miARN esta relacionado genes supresores de tumores.

Micro- RNA (miRNA) are small molecules endogenous RNA which act as regulatory molecules of gene transcription by degradation of mRNA or translational inhibition , Micro- RNA can control a large number of target genes, characterization genes can be crucial to identify role miRNAs as oncogenes or tumor suppressor genes . Find miRNA to the complementary region of a target sequence of RNAm using computational tools can be helpful in the future to do test in a laboratory and find if it can be act as miRNA tumor suppressor genes .

Index Terms—RNA, miRNA, ARNm, 3'UTR, Arbol de sufijos

MicroRNA reprimiendo más genes posibles

Janeth De Anda Gil

1. Introducción

MicroARN

Los microARN de regulación tienen una longitud de 19-24 nucleótidos y regulan la expresión de genes mediante la unión directa y preferiblemente a la región no traducida 3 ′ (3'UTR) de codificación de la proteína genes[1].Inicialmente, se creyó que la mayoría de los genes que codifican para los microARN estaban localizados en regiones intergénicas, pero estudios más recientes han demostrado que casi el 70 % de los microARN se localizan en unidades de transcripción[2]. La base de datos mirBase del Instituto sanger [1][3] representa el registro en el que se incluyen las secuencias de los microARN descritos y, actualmente, se han año se han encontrado 695 microARN humanos

Los genes de los miARN son transcritos en el núcleo por la ARN polimerasa II para formar los pri-miARN. Estos pri-miARN son procesados por Drosha y Pasha para formar los pre-miARN, que serán transportados al citoplasma por la exportina 5. Posteriormente, la ARNa tipo III Dicer es capaz de generar un dúplex transitorio de aproximadamente 22-nucleótidos que se asociará al complejo RISC. El miARN maduro se une a la región complementaria de la secuencia diana de ARN mensajero (ARNm) y se inciará el silenciamiento génico de acuerdo con el grado de complementariedad entre ambas hebras. Si la complementariedad es perfecta, se provocará la degradación del ARNm, mientras que si la complementariedad es imperfecta la unión impedirá la síntesis proteica. Sin embargo, si el emparejamiento entre las bases no es perfecto, como ocurre con la mayoría de los mayoría de mamíferos, se producirá la inhibición de la traducción. Un microARN puede tener muchos ARNm diana y cada ARNm puede estar regulado por varios microARN. En estudios bioinformáticos se ha estimado que los microARN pueden regular hasta el 30 % de todos los genes humanos.

El prerequisito básico en metazoos para que un microARN se una a un ARNm es un corto empalme complementado por imperfectos empalmes en las cercanías. Esta región es llamada la secuencia de semilla y se considera que tiene 6-8 nucleótidos de longitud y se encuentra dentro de la primeros 8 nucleotidos en el extremo 5'end de los genes microARN[4]. Y está considerado como la característica más importante para el reconocimiento objetivo de miRNAs en mamíferos[5].

1. (http://microrna.sanger.ac.uk/)

2. Métodos

El objetivo es hallar la existencia de posibles ARNm a los que se puede unir diferentes MicroARN maduros, en este caso se va a hacer uso de la secuencia semilla mencionada en la introducción que tendrá una longitud de 7 a 8 nucleotidos, así mismo se hará uso de la caracteristica que se debe encontrar en las primeras posiciones del extremo 5' de los microARN. Dadas estás caracteristicas, se pueden hacer uso de una herramienta que busque una secuencia semilla que verifique su complementaridad por toda la secuencia del ARNm y que aparezca más veces complementada, así como también hallar datos de termodinámica. En este caso se pueden tener diferentes secuencuas semillas obtenidas del microARN maduro, y verificar cada nucleotido de estas semillas con cada nucleotido de la secuencia del ARNm puede tomar mucho tiempo, es decir, si se tiene AATTTAG y se quiere buscar si esta la semilla AT y cuantas veces está, entonces se busca la primer letra de la semilla, está en la primera posición ,pero la letra T ya no está en la segunda posición, pero no se puede pasar a la tercera posición es decir , se tiene que regresar que regresar a la segunda posición porque puede que exista una 'A', entonces si l es la longitud de la semilla y n es la longitud de la secuencia del ARNm la complejidad será O(l*n) pero si a esto agregamos que se tienen s número de semillas entonces la complejidad aumenta a O(s*l*n), además si se agrega que tenemos g número de genes (ARNm) y m número de microARN , la complejidad sería O(g*m*s*l*n), que sería enorme, para evitar que crezca tanto la complejidad se puede hacer uso de varias herramientas, en este árticulo se proponen dos , una que es muy sencilla de implementar pero requiere mucho espacio en memoria y otra un poco más complicada en implementar , pero la busqueda es muy rápida y no requiere tanto espacio en memória llamada árboles de sufijos.

Herramienta propuesta.

Se sabe que nuestro vocabulario serán solo 4 letra 'C','G','A','T', se propone usar solo una matriz de números que tendrá 4 filas (fila 0 para 'A', fila 1 para 'C' , fila 2 para 'G' y fila 3 para ''T'),y el número de columnas será la longitud del gen en el que se van a buscar las subcadenas de microARN, los valores de los arreglos serán de dos tipos '1' si se encuentra la letra en esa posición de la cadena y '0' en otro caso , por ejemplo suponiendo que mi gen es "TGACT" se tendría el cuadro (1).

Cómo se puede observar la construcción de esta tabla es de tiempo lineal O(g) donde g es la longitud del gen, el

Posiciones	0	1	2	3	4
A	0	0	1	0	0
C	0	0	0	1	0
G	0	1	0	0	0
T	1	0	0	0	1

Cuadro 1
Tabla de posiciones

problema es que si g es muy grande se ocupara demasiado espacio de memoria. En cuanto a la busqueda de subcadenas semillas de microARN, se busca la fila de la primera letra , se localizan los posibles 1, ejemplo, para buscar la subcadena "AC" se sabe que la fila 0 contiene a la letra A , se busca solo en esa fila los posibles '1' y se guarda el índice en un vector , esto nos toma tiempo O(g), ahora la siguiente letra es C, mi indice de la letra A en la tabla anterior es 2 (se encuentra en la columna 2) , la letra 'C' representa la fila 1, entonces busco en la fila 1 y la columna 3 (indice más 1) si contiene un 1 quiere decir que si se encuentra la subcadena AC, esta operación toma tiempo l , donde m es el tamaño de la subcadena semilla , en total la construccion y busqueda nos toma tiempo O(g+l) y si se requieren varios semillas el tiempo será O(g+s*l) donde s es el número de semillas.

Árbol de sufijos

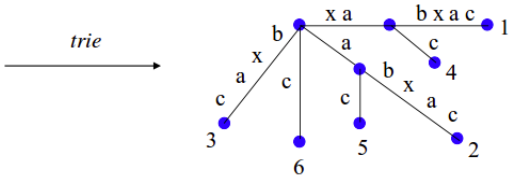
Un árbol de sufijos es una estructura de datos que nos ayuda al reconocimiento de patrones, con esta estructura, es posible determinar si una cadena arbitraria z es subcadena de x en tiempo proporcional al largo de z, independientemente de n.

Definicion: Un árbol de sufijos para una cadena S de longitud m

- Árbol con raíz y con m hojas numeradas de 1 a m
- Cada nodo interno (salvo la raíz) tiene al menos 2 hijos
- Cada arista está etiquetada con una subcadena no vacía de S
- Dos aristas que salen del mismo nodo no pueden tener etiquetas que empiecen por el mismo carácter
- Para cada hoja i, la concatenación de etiquetas del camino desde la raíz reproduce el sufijo de S que empieza en la posición i.

Un árbol de sufijos es como un trie (árbol de prefijos) que almacena todos los sufijos de una cadena, por ejemplo los Sufijos de la cadena S= xabxac.

- c
- ac
- xac
- bxac
- abxac
- xabxac



Para este caso se realizo un algorítmo en java que construye un árbol , se hace uso de la clase Nodo_tree_s que contiendra los nodos del árbol, donde cada nodo contiene los siguientes atributos: etiqueta: que es la letra o letras del nodo tamaño: número de letras; List<Integer> conexion: lista de los nodos con los que se relaciona.

En otra clase Tree_suf_archivo, tendremos un arreglo de nodos, e iremos construyendo el árbol, dada una cadena. Se leeran las letras de la cadena de atrás hacia adelante, y se ira guardando el texto concatenado en otra variable, cada palabra o letra será guardada como etiqueta de un nodo, dependiendo de algunas condiciones, empezando por que los primeros 4 nodos del arreglo corresponderan a las palabras que empiecen con las letras de los nucleotidos (A','C','G','T), o simplemente a las letras de los nucleótidos. Entonces en el arreglo en la primera posicion se tiene la letra 'A' o palabra que empiece con 'A', en la segunda , se tiene la letra 'C' o palabra que empiece con 'C', la siguente será la 'G' y en seguida se tiene la 'T'. Se tomá la primera letra de la cadena concatenada, por ejemplo si es C, se verifica si el arreglo de nodos en la segunda posición está vacio, si es así entonces la etiqueta será la palabra concatenada, en caso contrario,se tienen dos posibilidades:

- Si una parte de la cadena concatenada empalma completamente con la etiqueta del nodo , por ejemplo: si se tiene AGTC (cadena concatenada)y AG (etiqueta del nodo), la parte AG de AGTC empalma con la etiqueta AG, en este caso se agrega a la lista de conexiones del nodo un nuevo nodo que contendra la parte que no empalmo de la cadena concatenada, que sería TC.
- Si una parte de la cadena concatenada empalma con una parte de la etiqueta del nodo , por ejemplo: si se tiene AGTC (cadena concatenada)y AT (etiqueta del nodo), la parte A de AGTC empalma con la parte de la etiqueta A, en este caso se agregan a la lista de conexiones del nodo dos nuevos nodos que contendrán las partes que no empalmaron, en este caso se tendría un nodo con la etiqueta GTC que es parte de la cadena concatenada y otro nodo con la etiqueta T.

La construcción de un árbol de sufijos toma una complejidad de $O(n^2)$, mientras que la busqueda de una subcadena en el arbol toma tiempo lineal, ya que se sabe cual es la primera letra de la subcadena y está letra es uno de los primeros 4 elementos del arreglo de nodos, elemento 0 contiene la 'A' , elemento 1 contiene la 'C', elemento 2 contiene 'G', elemento 3 contiene 'T', entonces se verifica cada letra de la subcadena con cada letra de la etiqueta del nodo, y si la longitud de la etiqueta del nodo es más pequeña que la subcadena, se viaja por cada uno de los elementos de la lista de conexiones del nodo, hasta hallar la cadena o en caso contrario , alguna letra que nos indique que no existe la subcadena en el árbol.

Ambas herramientas reducen el tiempo o complejidad, pero cabe destacar que el árbol de sufijos no consume tanta memoria y debido a que nuestros genes a ocupar pueden contener un gran número de nucleotidos se decidio usar los arboles de sufijos Una vez teniendo el árbol de sufijos se parte un microARN en semillas, se busca en el árbol cuales empalman y cuantas veces, y se elige la que empalme más veces, este procedimiento lo repetimos para todos los microARN que desemos y buscamos las semillas de los microARN que más se hallan empalmado y su termodinámica, la cual se obtiene sumando 4 a cada empalme C-G y sumando 2 a cada empalme A-T.

GEN	MicroARN (miRBase)	Repeticiones (miRBase)	MicroARN Programa árticulo)	Repeticiones (Programa árticulo)	Termodinámica (miRBase-árticulo)
YWHAZ	Hsa-miR-1302	TGTCCCA: 3	Hsa-miR-1302 Hsa-miR-4298	TGTCCCA: 3 TGTCCCA: 3	22-22 22-22
ZC3H12C	hsa-miR-200b-5p	GTAAGAT: 3	hsa-miR-200b-5p hsa-let-7a-3p	GTAAGAT: 3 ATTGTAT: 3	18-18 18-16
FGFRL1	hsa-miR-210-3p	ACGCACA: 7	hsa-miR-210-3p hsa-miR-210-3p hsa-miR-4298	ACGCACA: 8 CACGCAC: 8 CTGTCCC: 8	22-22 24-24 24-24
RIMKLB	hsa-let-7a-3p	TTGTATA: 2	hsa-let-7a-3p hsa-miR-429 hsa-miR-4684-5p	TTGTATA: 2 CAGTATT: 2 TAGAGAG : 2	16-16 16-18 16-20
CIPC	hsa-miR-5691	CAGAGCA : 4	hsa-miR-5691	CAGAGCA : 4	22-22
ULK2	hsa-miR-3159	TAATCCT: 2	hsa-let-7a-3p hsa-miR-4298 hsa-miR-4298 hsa-miR-3159 hsa-miR-4456 hsa-miR-146a-5p hsa-miR-429	TTGTATA : 2 CTGTCCC: 2 CCTGTCCC: 2 TAATCCT: 2 GCCACCA : 2 CAGTTCT: 2 CAGTATT: 2	18-16 18-24 18-28 18-18 18-24 18-20 18-18
HMGXB4	hsa-miR-4456	CCACCAG: 3	hsa-miR-4456	CCACCAG: 3	24-24
NOVA1	hsa-miR-146a-5p	AGTTCTC: 2	hsa-miR-146a-5p hsa-miR-429	AGTTCTC: 2 CAGTATT: 2	20-20 20-18
HIPK3	hsa-miR-429	CAGTATT:4	hsa-miR-429	CAGTATT:4 ATTGTAT:4	18-18 18-16
ZEB1	hsa-miR-429	CAGTATT: 6	hsa-miR-429	CAGTATT: 6	18-18
ZEB2	hsa-miR-429	CAGTATT: 5	hsa-miR-429	CAGTATT: 5 AGTATTA: 5	18-18 16-18
TRIM33	hsa-miR-429	CAGTATT: 4	hsa-miR-429	CAGTATT: 4	18-18
FAM76A	hsa-miR-34a-5p	CACTGCC: 2	hsa-miR-34a-5p	ACTGCCA:3	24-22
CELF3	hsa-miR-34a-5p	CACTGCC:2	hsa-miR-34a-5p	CACTGCC:2	24-24
FLOT2	hsa-miR-34a-5p	CACTGCC:3	hsa-miR-34a-5p	CACTGCC:3	24-24
MYCN	hsa-miR-34a-5p	CACTGCC:2	hsa-miR-34a-5p hsa-let-7a-3p	CACTGCC:2 TTGTATA: 2	24-24 24-16
VAMP2	hsa-miR-34a-5p	CACTGCC: 3	hsa-miR-34a-5p	CACTGCC: 3	24-24
LIMA1	hsa-miR-3675-3p	TAGAGAT: 3	hsa-miR-3675-3p hsa-miR-3675-3p	TAGAGAT: 3 TTAGAGA: 3	18-18 18-18
MEF2C	hsa-let-7a-3p	TGTATAG: 3	hsa-let-7a-3p hsa-let-7a-3p	TGTATAG: 3 ATTGTAT: 3	18-18 18-16
JADE2	hsa-miR-4684-5p	GTAGAGA: 2	hsa-miR-4298	GTCCCAG: 4	20-24

Figura 3.1. Tabla de resultados.

3. Resultados

Haciendo uso de la base de datos miRBase, se obtuvieron microARN que empalmaban con 3'UTR de ciertos ARNm, con esto se tiene una idea de las caracteristicas de los microARN necesarios, con esto uno se puede preguntar si estos microARN pueden empalmar con otros genes, haciendo uso de la herramienta desarrollada y explicada anteriormente se buscaron las posibles semillas que empalman con mayor número de repeticiones, en la siguente tabla se muestra en la primera columna los genes, en la segunda los microARN con los que empalman según la base de datos miRBase , en la tercera columna el número de repeticiones (miRBase), en la cuarta columna el mejor microARN con el que empalma el gen según el programa realizado y explicado en este árticulo y en la quinta columna el número de repeticiones (Programa árticulo). Se puede observar en los resultados del programa del árticulo que hay algunas repeticiones de distintos microARN que son mayores o iguales que las repeticiones del microARN escogido en la página de mirBAse, sería interesante probar en la práctica si estos microARN propuestos son realmente microARN objetivo del respectivo gen que aparece en la tabla. También existen casos donde se repiten más veces los microARN en los resultados del programa del árticulo que en la página miRBase, por ejemplo el gen FGFRL1, esto es debido a que en el caso del programa del árticulo se toman en cuenta los sobrelapamientos y en miRBase no. También sería de gran interes probar estos microARN que se proponen para averiguar si son realmente microARN objetivo del respectivo gen que aparece en la tabla.

Referencias

[1] Alexiou P., et al. Lost in translation: an assessment and perspective for computational microrna target identification. Bioinformatics. 2009;25:3049–3055.

[2] Rodriguez A, Griffiths-Jones s, Ashurst JL, Bradley A Identification of mammalian microrNA host genes and transcription units. Genome res. 2004;14:1902-10.

[3] Griffiths-Jones s, Grocock rJ, Van Dongen s, Bateman A, Enright AJ. mirBase: microrNA sequences, targets and gene nomenclature. Nucleic Acids res. 2006;34:D140-4.

[4] Lewis et al., 2003.

[5] Bartel , 2009 ; Nielsen et al., 2007.