RNAemo

Especificación de Requerimientos de Software

Franco Gaspar Riberi

7 de junio de 2012

Índice

1. Introducción

En esta sección se describe un panorama completo del SRS.

1.1. Propósito

El propósito de este documento es la especificación de requerimientos de software en el marco de la tesis de grado de la carrera Licenciatura en Ciencias de la Computación de la UNRC, **"Estudio de la relación entre divergencia en el uso de codones sinónimos entre virus y huésped y presencia de microRNA"**. Internamente este proyecto será denominado **"*RNAemo*"**. Los requerimientos del software son provistos por integrantes de **FuDePAN** en su carácter de autores intelectuales de la solución que se pretende implementar y colaboradores de dicha tesis.

Además, este documento establece la primera etapa de dicha tesis y será utilizado como parte de la validación final del proyecto.

1.2. Convenciones del Documento

Las palabras clave **DEBE**, **NO DEBE**, **REQUERIDO**, **DEBERÁ**, **NO DEBERÁ**, **DEBERÍA**, **NO DEBERÍA**, **RECOMENDADO**, **PUEDE** y **OPCIONAL** en este documento son interpretadas como esta descripto en el documento RFC 2119 [8].

1.3. Audiencia Esperada

A continuación se enumeran las personas involucradas en el desarrollo de la tesis y que por lo tanto, representan la principal audiencia del presente documento.

- *Dr. Roberto Daniel Rabinovich*: Miembro del INBIRS, anteriormente CNRS. Profesor Titular de Virología (Departamento Ciencias Biológicas, CAECE). Colaborador de **FuDePAN**.
- *Lic. Lucía Fazzi*: Licenciada en Genética.
- *Dra. María Pilar Adamo*: Colaborador de tesis, **FuDePAN**.
- *Daniel Gutson*: Co-director de tesis, **FuDePAN**.
- *Lic. Guillermo Biset*: Co-director de tesis, **FuDePAN**.
- *Lic. Laura Tardivo*: Directora de tesis, UNRC.

- *Ac. Franco Gaspar Riberi*: Tesista, UNRC.

1.4. Alcance

El producto que se especifica en este documento se denomina **"Estudio de la relación entre divergencia en el uso de codones sinónimos entre virus y huésped y presencia de microRNA"**, y su principal objetivo es contrastar formalmente una idea encomendada y postulada por el Dr. Roberto Daniel Rabinovich que involucra principalmente la molécula de RNA.

Para la comprensión de la hipótesis se deben tener en cuenta algunos hechos referidos al código genético y los microRNA tales como:

- El código genético está organizado en tripletes o codones.
- El código genético es degenerado[1]: existen más tripletes o codones que aminoácidos, de forma que un determinado aminoácido puede estar codificado por más de un triplete. Esos tripletes son conocidos como codones sinónimos.
- En cada especie, se ha seleccionado una proporción de uso de esos codones que guarda relación con la proporción de RNA de transferencia correspondiente de manera de optimizar la síntesis proteica.
- Para algunos patógenos intracelulares como los virus, existe una divergencia entre el uso de codones sinónimos [20] utilizado por el virus y el huésped correspondiente. El origen de esa divergencia no esta suficientemente esclarecido.
- Un microARN [18][19]($_{mi}$ARN o $_{mi}$RNA por sus siglas en inglés) es un RNA monocatenario, de una longitud de entre 21 y 25 nucleótidos, y que tiene la capacidad de regular la expresión de otros genes mediante diversos procesos, utilizando para ello la ruta de ribointerferencia[2]. Se encuentran codificados en el genoma y juegan un papel importante en la regulación de la expresión proteica, en la embriogénesis[3], procesos cancerosos e infecciones virales. La generación de los microRNA puede variar según el órgano o temporalmente.

[1]Implica que al menos uno de los tres nucleótidos (en general el último) puede ser distinto y sin embargo codificar para el mismo aminoácido.

[2]También conocido como RNA$_i$ por el acrónico del inglés RNA interference. Corresponde a un mecanismo de silenciamiento post-transcripcional de genes específicos, ejercido por moléculas de RNA que, siendo complementarias a un RNA mensajero, conducen habitualmente a la degradación de éste.

[3]Proceso que se inicia tras la fertilización de los gametos para dar lugar al embrión.

En este trabajo se estudiará si la divergencia en el uso de codones sinónimos entre virus y huésped contribuye a disminuir la interferencia de los $_{mi}$RNA en la expresión de los RNA$_m$ de origen viral. De esa manera se contribuirá a comprender mejor la relación virus-huésped y la evolución viral.

Dado un conjunto de small-RNA$_s$[4] y una colección de RNA$_m$ de un determinado virus, dado que hay un RNA$_m$ por cada gen, determinar si existen small-RNA$_s$ que se van a hibridar al RNA$_m$. Luego contabilizar la cantidad de small-RNA$_s$ que se hibridarían tanto a la secuencia original como a la secuencia complementaria. De manera similar, realizar el mismo estudio sobre el genoma humanizado[5] contabilizando la cantidad de small-RNA$_s$ que se hibridan.

Que se hibride o no un small-RNA$_s$ a un determinado RNA$_m$ involucra distintas reglas, siendo la más importante la complementariedad de bases. Pero además existen otras, como la presencia de determinadas bases, proporción de uniones GC y motivos específicos.

El proyecto involucrará determinados virus aún no definidos. Para tal fin se aplicarán criterios biológicos.

El sistema a desarrollar comprenderá las siguientes característica:

- Abarcar en su totalidad los requerimientos del problema.
- Construir un sistema que puede ser extendido en otros proyectos, brindando un diseño flexible.
- Que proponga un buen uso de las prácticas de diseño para su mejor desempeño.
- Que posea abundante documentación clara y precisa.
- Lograr un código fuente bien escrito y estructurado respetando las buenas prácticas de programación.

1.5. Descripción general del documento

La estructura de este documento sigue las recomendaciones de la "Guía para la especificación de requerimientos de la IEEE" (IEEE Std 830-1998) [1]. El documento esta organizado en las siguientes secciones generales:

En la *sección 1*, Introducción, se presenta una primera aproximación al proyecto.

[4]Moléculas de RNA muy pequeñas, dentro de las cuales se encuentra la $_{mi}$RNA, $_{si}$RNA, entre otras.

[5]La humanización refiere al reemplazo de nucleótidos en los tripletes que forman la secuencia de RNA. Se acerca a la proporción de uso de codones utilizado en el humano.

En la *sección 2*, Descripción General, se presenta una descripción general de ***RNAemo***, sus principales funcionalidades, interfaces y perfiles de usuarios.

En la *sección 3*, Requerimientos, se detallan los requerimientos funcionales específicos de ***RNAemo*** y los principales atributos que **DEBE** cumplir el software.

Por último, se detalla un pequeño apéndice.

2. Descripción General

Esta sección describe los requisitos del producto de modo general. Los requisitos específicos se describen en la sección 3.

2.1. Perspectiva del Producto

La bioinformática es una disciplina dedicada al análisis de elementos biológicos utilizando a la informática como herramienta principal para generar simulaciones, probar teorías, o realizar cálculos complejos entre otros aspectos. En particular, el producto a desarrollar apunta al cálculo complejo de ciertos datos, los cuales son de gran interés para la biología. El mismo, no será un componente de un sistema de mayor envergadura, sino que por el contrario, será totalmente autónomo e independiente. Además, se espera que sea modular permitiendo:

- Obtener resultados intermedios y numéricos.
- Combinable con otros componentes existentes o a ser desarrollados en el futuro, para obtener nuevos resultados.

El sistema se compondrá de los siguiente módulos:

- **Módulo generador de secuencias humanizadas:** dada una secuencia original, genera una secuencia humanizada. El mismo, corresponde a un componente externo a este desarrollo, y será parte del análisis su obtención.
- **Módulo de matching:** dada una secuencia "larga" de RNA mensajero y una secuencia de small-RNA$_s$, obtener la estructura secundaria del mensajero y calcular el score de matching a lo largo de todo el genoma de la misma. El cálculo se realizará comparando nucleótido a nucleótido de a un paso a la vez, determinando cuales hacen matching por complemento. Además se determinará si el nucleótido del RNA$_m$

está apareado o no. Se mostrará con minúsculas aquellos nucléotidos que no coincidan, en caso contrario, y si el nucleótido del RNA_m no está apareado se utilizarán mayúsculas, en contrapartida, si está apareado, se exhibirá una "M" (Masked), lo cual significa que el nucleótido está enmascarado en la estructura secundaria del RNA blanco, es decir, no esta disponible para aparearse con el $_{mi}$RNA. De manera similar a lo mencionado anteriormente, generará cadenas en las cuales se exhibirá con letras "X","Y" o "Z" aquellos nucleótidos que esten apareados, dependiendo del tipo de unión.

- **Módulo maestro ("Master Of Puppets"):** utilizando los módulos anteriores, el presente contabiliza y genera tablas y gráficos leyendo una base de datos de small-RNA_s.

Para realizar esta tesis, será necesario contar con datos tanto para realizar las pruebas como para obtener los resultados reales y finales. Sin embargo, la obtención de estos datos en volumen está considerado FUERA del ámbito de la Tesis de Licenciatura en Ciencias de la Computación, sí sin embargo, se espera recopilar un mínimo de datos para poder hacer las pruebas. Es por esto que se considerará una etapa de búsqueda de datos mínimos, dejando el grueso de la búsqueda de datos para **FuDePAN**.

El desarrollo de esta tesis seguirá un modelo de cascada, el cual estará compuesto por cuatro iteraciones.

- *Iteración 1:* corresponde al desarrollo de un software para la generación de tablas comparativas y gráficos. Se trabajará sobre la estructura secundaria de las secuencias de RNA_m. Además, se hará la búsqueda de datos mínimos, y se adaptarán formatos de ser necesarios (bibliotecas de **FuDePAN** trabajan con el formato FASTA). Asimismo, se buscará el software existente capaz de humanizar una secuencia dada. Se estudiarán las librerías involucradas.

  Los datos mínimos incluirán:

  1. Un conjunto de al menos 5 RNA mensajeros de virus relevantes.
  2. Un conjunto de al menos 50 small-RNA_s (por ejemplo $_{mi}$RNA).

  Se espera obtener diferentes tablas comparativas y gráficos que se especifican en las sección 3.

- *Iteración 2:* corresponde a un análisis estadístico sobre las tablas generadas. Se **DEBE** determinar qué tipo de análisis aplicar. Luego, se espera obtener scripts los cuales se ejecutarán y darán como resultado diversos gráficos.

- *Iteración 3:* corresponde a un análisis intervirus. Se **DEBERÁN** tratar diversos virus. Se espera generar resultados considerando más de un virus.
- *Iteración 4:* corresponde al cálculo de hibridación sobre la estructura secundaria del RNA, que permitirá una aproximación más exacta a la realidad. Se utilizarán como input de esta iteración los valores y resultados obtenido en las iteraciones previas.

2.1.1. Interfaces del Sistema

El producto será capaz de correr al menos en sistemas GNU/Linux, por lo cual sólo se utilizarán librerías compatibles con el mismo.

2.1.2. Interfaces de Usuario

En primera instancia, el usuario interactuará con el sistema mediante una CLI, no se proveerá de interfaz gráfica de usuario (GUI).

2.1.3. Interfaces de Hardware

El producto de software no requerirá hardware específico alguno para su correcto funcionamiento.

2.1.4. Interfaces de Software

Las librerías requeridas para el funcionamiento del sistema son las siguientes:

1. **MiLi:** Colección de pequeñas bibliotecas C++, compuesta únicamente por headers. Sin necesidad de instalación, sin un makefile, sin complicaciones. Soluciones simples para problemas sencillos.

   mili.googlecode.com.

2. **FuD:** Framework para la implementación de aplicaciones distribuidas.

   fud.googlecode.com.

3. **BioPP:** Librería de C++ para el manejo de estructuras biológicas, código genético, entre otras funciones.

   biopp.googlecode.com.

4. **Biopp-filer:** Librería de persistencia para Biopp.

   biopp-filer.googlecode.com.

5. **Odf-gen:** Librería que permite la generación de archivos OpenOffice.

   odf-gen.googlecode.com.

6. **Fideo:** Provee, entre otras, las funcionalidades necesarias para obtener la energía libre de una secuencia de nucleótidos. Casi la totalidad su código proviene del proyecto **VAC-O**.

   vac-o.googlecode.com.

   fideo.googlecode.com.

2.1.5. Interfaces de Comunicaciones

No hay requerimientos especificados.

2.1.6. Restricciones de Memoria

Este proyecto no presenta restricciones en cuanto a la cantidad de memoria mínima necesaria para operar. El sistema **DEBERÁ** manejar la memoria en forma correcta y sin que ocurran memory leaks. Para realizar depuraciones se utilizará *Valgrind*[6].

Las dependencias externas que serán utilizadas en el producto no deberán ser tenidas en cuenta en el chequeo de memory leaks u otros problemas.

2.1.7. Operaciones

El modo de operación del sistema será:

1. Invocación por consola por parte del usuario. Esta invocación **DEBERÁ** tener asociada los siguientes parámetros:

   - Colección de RNA_m.
   - Base de datos de small-RNA_s.

2. Realización de cálculos internos.

3. En caso de que no se produzcan situaciones erróneas:

   - Generación y exhibición de tablas comparativas de posible hibridación entre microRNA presente en el huésped humano y el RNA presente en la naturaleza y el genoma viral humanizado.

[6]Conjunto de herramientas libres que ayuda en la depuración de problemas de memoria y rendimiento de programas. Permite realizar un seguimiento del uso de la memoria y detectar problemas. http://valgrind.org/

- Generación de gráficos con resumen estadístico a través de los cuales se podrá inferir, según la tendencia de las curvas, si la divergencia en el uso de codones entre virus-huésped y la presencia de microRNA pueden estar relacionados.

De lo contrario, el sistema **DEBERÁ** informar de tal suceso.

2.1.8. Requerimientos de Instalación

No se registran requerimientos de instalación.

2.2. Funciones del Producto

- **DEBERÁ** ser capaz de tomar como entrada una colección de RNA_m y una base de datos de small-RNA_s.
- **DEBERÁ** controlar la validez de los parámetros de entrada.
- **DEBERÁ** calcular la secuencia complementaria de la secuencia original de RNA_m.
- **DEBERÁ** calcular la estructura secundaria de la secuencia original de RNA_m.
- **BEBERÁ** obtener la secuencia humanizada de la secuencia original de RNA_m utilizando un software externo.
- **DEBERÁ** realizar permutaciones sobre la secuencia de RNA_m conservando la cantidad de cada nucleótido y su tamaño.
- **DEBERÁ** generar secuencias enmascaradas tanto para la secuencia original como para la secuencia humanizada.
- **DEBERÁ** generar secuencias según el tipo de uniones entre nucleótidos no disponibles, tanto para la secuencia original como para la secuencia humanizada.
- **DEBERÁ** calcular el score de matching, tanto para la secuencia original, como para la secuencia humanizada.
- **DEBERÁ** mostrar los resultados al usuario mediante tablas comparativas y gráficos.

2.3. Características de Usuario

Se identifica un solo tipo de usuario para el sistema:

1. *Usuario final:* este tipo de usuario refiere a aquellas personas profesionales que utilizarán el producto. Sólo deberán interactuar gráficamente (mediante CLI), cargando los datos de entrada necesarios y ejecutando el programa para luego obtener el resultado.

2.4. Restricciones

El producto **DEBERÁ** ser desarrollado utilizando el lenguaje de programación C++ [3] y bajo la licencia de software GPLv3 [9]. Además, se **DEBERÁ** respetar los lineamientos generales impuestos por **FuDePAN** (Thesis Guideline y Coding Guideline).

2.5. Trabajo Futuro

Inclusión de BLAST como fuente de datos de small-RNA$_s$.

3. Requerimientos

En esta sección se detallan específicamente los requerimientos del producto. Se hará hincapié en los requerimientos funcionales de la iteración 1.

3.1. Funciones del Sistema

3.1.1. Interfaces Externas

No hay requerimientos especificados.

3.1.2. Requerimientos Funcionales

- **Nombre del requerimiento:** Cargar una colección de RNA$_m$. (RF1).
  Propósito: Obtener los RNA$_m$ que codifiquen para un mismo virus. Contra los cuales se determinará si ciertos small-RNA$_s$ se hibridan.
  Input: Colección de RNA$_m$.
  Procesamiento:
  Output:

1. **Sub requerimiento:** Verificar colección de entrada. (RF1.1).
Propósito: asegurar que los datos ingresados están dentro de los parámetros aceptables.
Input: Colección de RNA_m.
Procesamiento:
Output: datos válidos o inválidos.

- **Nombre del requerimiento:** Cargar una base de datos de small-RNA_s. (RF2).
Propósito: Obtener los small-RNA_s los cuales serán utilizados para comparar contra los RNA_m.
Input: Base de datos small-RNA_s.
Procesamiento:
Output:

1. **Sub requerimiento:** Verificar base de datos de entrada. (RF2.1).
Propósito: asegurar que los datos ingresados están dentro de los parámetros aceptables.
Input: Base de datos small-RNA_s.
Procesamiento:
Output: datos válidos o inválidos.

- **Nombre del requerimiento:** Obtener una secuencia humanizada (RF3).
Propósito: Obtener una secuencia humanizada para contabilizar los small-RNA_s que se hibridan a ella. Se utilizará un software externo.
Input: secuencia original de RNA_m.
Procesamiento: Tomar una secuencia de RNA_m, y reemplazar nucleótidos en los tripletes conservando la expresión del aminoácido. Se acerca a la proporción de uso de codones utilizado en el humano.
Output: secuencia humanizada.

- **Nombre del requerimiento:** Realizar control estadístico a través de secuencias random (RF4).
Propósito: Generar N secuencias random y obtener estadísticas comparativas.
Input: secuencias random de RNA_m
Procesamiento: Tomar las secuencias de entrada y aplicar cálculos

estadísticos comparativos, o descriptivos.
Output: Control de secuencia válido o no válido.

1. **Sub requerimiento:** Obtener secuencias random (RF4.1).
Propósito: Obtener secuencias random para establecer los controles estadísticos.
Input: secuencia original de RNA_m.
Procesamiento: Tomar una secuencia de RNA_m, y permutar al azar nucleótidos conservando la misma cantidad de los mismo y el mismo tamaño de secuencia.
Output: secuencias random. Por ejemplo, si la secuencia de entrada es **AAUUCCGG**, permutaciones posibles son: **AUAUGCGC**, **GCGCUAUA**, **GCAUGCUA**, entre otras.

- **Nombre de requerimiento:** Generación de secuencias a partir del RNA_m (RF5).
Propósito: Generar una secuencia o segmento por cada posición (de a nucleótidos) del RNA_m en base al matching con respecto a los small-RNA_s.
Input: RNA_m, small-RNA_s.
Procesamiento: Dado el RNA_m de entrada, obtener su estructura secundaria. Calcular la secuencia complementaria del small-RNA_s. Comparar nucleótido a nucleótido comenzando por la posición 1 del RNA_m, si se corresponden y el nucleótido del RNA_m no está apareado, colocarlo en mayúscula en la secuencia o segmento a generar, de lo contrario ponerlo en minúscula. Al llegar al extremo de la secuencia de small-RNA_s, avanzar un nucleótido en el RNA_m.
Output: lista de secuencias en la que se puede observar en mayúscula los nucleótidos que hacen matching por complemento y no se encuentran apareados en el RNA_m, y en minúscula aquellos que no hacen matching por complemento.

1. **Nombre de requerimiento:** Calcular la estructura secundaria del RNA_m (RF5.1).
Propósito: Obtener la secuencia secundaria del RNA_m.
Input: RNA_m.
Procesamiento: Tomar la secuencia de entrada y utilizar una librería ya implementada para este cálculo.
Output: secuencia secundaria del RNA_m.

2. **Sub requerimiento:** Determinar matching. (RF5.2).
Propósito: identificar si existe matching entre nucleótidos.
Input: un nucleótido del RNA_m y un nucleótido del ${}_{mi}RNA$.
Procesamiento: comparar los nucleótidos, si son iguales retorna true, de lo contrario false.
Output: nucleótidos coinciden o no coinciden.

requerimiento: Generación de secuencias enmascaradas a partir del RNA_m. (RF6).
Propósito: Generar una secuencia por cada posición (de a nucleótidos) del RNA_m que exhiba los nucleótidos no disponibles.
Input: RNA_m, small-RNA_s.
Procesamiento: Dado el RNA_m de entrada, obtener su estructura secundaria. Calcular la secuencia complementaria del small-RNA_s. Comparar nucleótido a nucleótido comenzando por la posición 1 del RNA_m, si se corresponden y el nucleótido del RNA_m está apareado, colocar una "M" (Masked) en la secuencia a generar, de lo contrario, permanece como está. Al llegar al extremo de la secuencia de small-RNA_s, avanzar un nucleótido en el RNA_m. Ver apéndice C.1.
Output: lista de secuencias en la que se puede observar con una "M" a aquellos nucleótidos que quedan enmascarados por el RNA_m al estar apareados, por lo que no están disponibles.

Requerimiento: Generación de secuencias según el tipo de unión en nucleótidos no disponibles. (RF7).
Propósito: Generar una secuencia por cada posición (de a nucleótidos) del RNA_m que exhiba los nucleótidos no disponibles mediante letras X,Y,Z, según el tipo de unión.
Input: RNA_m, small-RNA_s.
Procesamiento: Dado el RNA_m de entrada, obtener su estructura secundaria. Calcular la secuencia complementaria del small-RNA_s. Comparar nucleótido a nucleótido comenzando por la posición 1 del RNA_m, si se corresponden y el nucleótido del RNA_m está apareado, colocar una letra "k" en la secuencia a generar. Donde "k" esta determinada por el tipo de unión entre los nucleótidos comparados. Es decir:

Unión A=U $\rightarrow$ k = "X"

Unión G=C $\rightarrow$ k = "Y"

Unión G=U $\rightarrow$ k = "Z"

Al llegar al extremo de la secuencia de small-RNA$_s$, avanzar un nucleótido en el RNA$_m$.
Output: lista de secuencias en la que se puede observar con una letra "X","Y", o "Z" a aquellos nucleótidos no disponibles según el tipo de unión (A=U, G=C y G=U respectivamente).

- **Nombre del requerimiento:** Generación de secuencias a partir del RNA humanizado (RF8).
  Idem al requerimiento RF5 pero tomando como entrada la secuencia humanizada en lugar de la secuencia original.

- **Nombre del requerimiento:** Generación de secuencias enmascaradas a partir del RNA humanizado (RF9).
  Idem al requerimiento RF6 pero tomando como entrada la secuencia humanizada en lugar de la secuencia original.

- **Nombre del requerimiento:** Calcular el score de matching sobre la secuencia original (RF10).
  Propósito: determinar cuanto matching existe entre el RNA original y el segmento resultante de la hibridación por cada posición del RNA$_m$.
  Input: secuencia de RNA original hibrizada.
  Procesamiento: Cada hibridación, sea A=T o G=C tendrá un valor representado a través de una dos constantes. Se recorrerá la secuencia tomada como entrada y se contará la cantidad de hibridación del tipo A=U (mayúsculas) por un lado, y por otro lado G=C (mayúsculas). Luego se calculará el score de matching teniendo en cuenta las constantes por separado. Ver apéndice C.2.
  Output: score de matching.

- **Nombre del requerimiento:** Calcular el score de matching sobre la secuencia original enmascarada (RF11).
  Propósito: determinar cuanto matching existe entre el RNA original y el segmento resultante de hacer chequeo de apariamiento por cada posición del RNA$_m$.
  Input: secuencia de RNA original enmascarada.
  Procesamiento: Será el mismo que el del requerimiento RF9, pero las "M" serán tomadas como minúsculas.
  Output: score de matching sobre la secuencia enmascarada.

- **Nombre del requerimiento:** Calcular el score de matching sobre la secuencia humanizada (RF12).
  Idem al requerimiento RF10 pero tomando como entrada la secuencia de RNA humanizada hibrizada.

- **Nombre del requerimiento:** Calcular el score de matching de la secuencia humanizada enmascarada (RF13).
  Idem al requerimiento RF11 pero tomando como entrada la secuencia enmascarada.

- **Nombre del requerimiento:** Construir tablas comparativas (RF14).
  Propósito: Exhibir los datos anteriormente mencionados en forma de tabla para una mejor comparación y permitiendo así llegar a ciertas conclusiones. Se construirán tantas tablas como combinación de RNA_m y small-RNA_s existan. Estas tablas serán utilizadas como input de la *iteración 4*.
  Input: posición, RNA_m, RNA humanizado, secuencieas enmascaradas, scores matching original y scores matching humanizado.
  Procesamiento: . Insertar cada dato de entrada en una columna diferente de la tabla.
  Output: Las tablas a generar se pueden observar en la Figura 1.

Posición	Secuencia Original			Secuencia Humanizada			Score secuencia original				Score secuencia humanizada			
	Matching	Masked	XYZ	Matching	Masked	XYZ	% cont=1 (match)	cFolding (match)	%const=1 (masked)	cFolding (Masked)	%const=1 (match)	cFolding (match)	%const=1 (Masked)	cFolding (Masked)
1	aaTTg CacA	aaTTg Maca	AaTTg Xaca	ttAAC GtcT	ttMAC MtcM	ttYAC YtcX	0.44	0.45	0.22	0.24	0.55	0.54	0.11	0.21
...	...	...		...	...		...	...	...	...	...	...	...	...
n														

Donde por ejemplo: (cFolding constAT = 1.25) && (cFolding constAT = 0.95)

Figura 1: Estructura de las tablas a generar.

- **Nombre del requerimiento:** Generar gráficos (RF15).
  Propósito: Exhibir la información a través de gráficos para una comprensión más clara.
  Input: posición, RNA_m, RNA humanizado, scores matching original, scores matching humanizado.

Procesamiento: .
Output: Las gráficos a generar se pueden observar en la Figura 2.

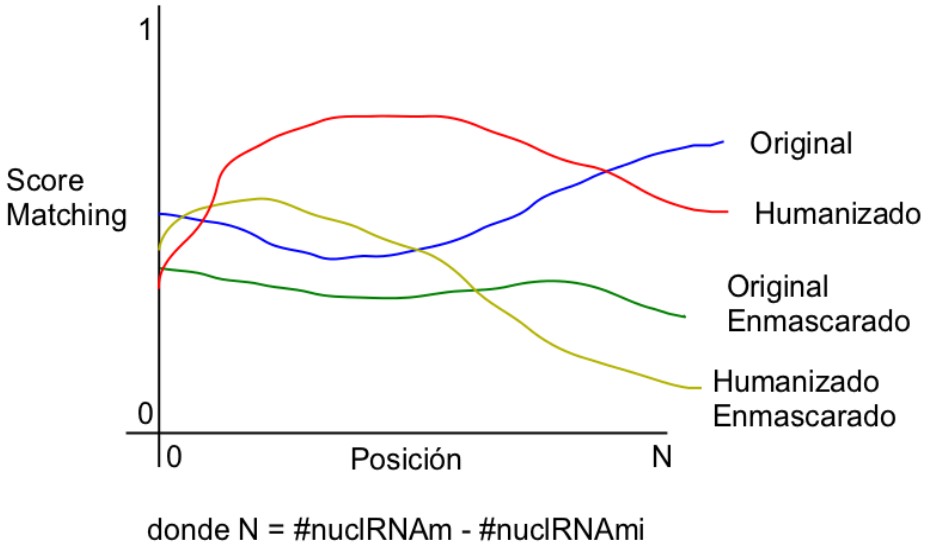

Figura 2: Estructura de los gráficos a generar.

3.2. Restricciones de Rendimiento

No hay requerimientos especificados.

3.3. Herramientas

Se utilizará *fudepan-build* [15] como build system.

3.4. Base de Datos

El sistema requerirá de una base de datos de small-RNA$_s$. En caso alternativo, se permitirá el uso de BLAST para generar secuencias.

3.5. Restricciones de Diseño

El producto **DEBERÁ** cumplir con los siguientes principios de diseño de la programación orientada a objetos. Los 5 primeros, son también

conocidos por el acrónimo **"SOLID"** [2].

- **S**ingle responsibility principle (SRP)
- **O**pen/closed principle (OCP)
- **L**iskov substitution principle (LSP)
- **I**nterface segregation principle (ISP)
- **D**ependency inversion principle (DIP)
- Law of Demeter (LoD)

3.6. Atributos del Software

El código del producto **DEBERÁ**:

- Compilar sin advertencias, o las advertencias aceptadas **DEBERÁN** estar documentadas.
- Cumplir con el estándar ANSI C++ y el "*Coding Guideline*" definido por **FuDePAN**.

El software **DEBERÁ**:

- Funcionar sin memory leaks según *Valgrind*.
- Tener al menos un 85 % de cobertura con pruebas automatizadas.

Apendices

A. Definiciones, Acrónicos y Abreviaturas

- **RNAemo:** Nombre que recibe el presente producto.
- **UNRC:** Universidad Nacional de Río Cuarto.
- **FuDePAN:** Fundación para el Desarrollo de la Programación en Ácidos Nucleicos [17].
- **INBIRS:** Instituto Biomédico en Retrovirus y SIDA.
- **CNRS:** Centro Nacional de Referencia para el SIDA.
- **SIDA:** acrónimo de síndrome de inmunodeficiencia adquirida. También abreviada como VIH-sida o VIH/sida.
- **CAECE:** Centro de Altos Estudios en Ciencias Exactas.
- **IEEE:** Institute of Electrical and Electronics Engineers.
- **SOLID:** acrónimo nemotécnico introducido por Robert C. Martin en la década de 2000, que representa cinco principios básicos de programación y diseño orientado a objetos
- **GPL:** *G*eneral *P*ublic *L*icense.
- **SRS:** Especificación de requerimientos.
- **FuD:** FuDePAN Ubiquitous Distribution [14]. Framework para el desarrollo de aplicaciones distribuídas a través de disposiciones heterogéneas y dinámicas de nodos de procesamientos.
- **CLI:** Interfaz de Línea de Comandos, por su acrónimo en inglés de Command Line Interface. Permite dar instrucciones a algún programa informático por medio de una línea de texto.
- **FASTA:** es un formato de archivos informáticos basado en texto, utilizado para representar secuencias de ácidos nucleicos, y en el que los pares de bases o los aminoácidos se representan usando códigos de una única letra. El formato también permite incluir nombres de secuencias y comentarios que preceden a las secuencias en sí.
- **Nucleótido:** molécula orgánica formada por la unión covalente de un monosacárido de cinco carbonos (pentosa), una base nitrogenada y un grupo fosfato.
- **Tripletes:** conjunto de tres nucleótidos que determinan un aminoácido concreto, también conocido como codón.

- **Aminoácido:** molécula orgánica que conforma la proteína.
- **RNA:** Ácido ribonucléico. Es un tipo de ácido nucleico compuesta por nucléotidos esencial para la vida.
- **DNA:** Ácido desoxirribonucleíco. Es un tipo de ácido nucleico, forma parte de todas las células.
- **RNA$_m$:** RNA mensajero. Se encuentra tanto en el núcleo como en el citoplasma celular. Su función es portar el código genético para las proteínas, es decir, transportan las instrucciones de codificación de las proteínas desde el DNA.
- **$_{si}$RNA:** short interfering RNA. Corresponde a una clase de RNA de cadena doble presente en células eucariotas.
- **$_{mi}$RNA o microRNA:** Corresponde a una clase de RNA de cadena simple presente en células eucariotas.
- **small-RNA$_s$:** Moléculas muy pequeñas de RNA. Dentro de la clasificación de RNA, aparecen como RNA no codificante.
- **Proteína:** macromolécula formada por cadenas lineales de aminoácidos. Se considera proteína a aquellas cadenas de aminoácidos enlazados cuyo peso molecular es superior 6000 Daltons.
- **Virus:** Entidad biológica que para reproducirse necesita de una célula huésped.
- **Δ(G) o energía libre:** Es el potencial químico que se minimiza cuando un sistema alcanza el equilibrio a presión y temperatura constante. [13]
- **RNA no codificante:** es aquel RNA que no genera proteínas. Se encuentra el RNA transcripcional y small-RNA$_s$.
- **Humanización-Deshumanización:** refiere a mutar de forma silente una secuencia. Esto significa, mutar nucleótidos de un triplete conservando la expresión del aminoácido. La diferencia entre humanización y deshumanización radica en que si los tripletes por los que se muta son o no los preferenciales.
- **Mutación silente:** Las mutaciones silentes ocurren cuando se produce un cambio de un sólo nucleótido de DNA dentro de una porción de un gen codificador para una proteína que no afecta la secuencia de aminoácidos que componen la proteína para el gen. Un cambio en un nucleótido, sin embargo, no siempre cambia el significado de un triplete. El triplete mutado puede aún representar el mismo aminoácido. Y cuando los aminoácidos de una

proteína siguen siendo los mismos, esta mantiene su estructura y función.

- **BLAST:** *B*asic *L*ocal *A*lignment *S*earch *T*ool [16]. Es un programa de alineamiento de secuencias, ya se de DNA, RNA o proteínas. Es capaz de comparar una secuencia problema (denominada query) contra una gran cantidad de secuencias almacenadas en una base de datos. Encuentra las secuencias de la base de datos que tienen mayor parecido a la secuencia query. BLAST es desarrollado por los Institutos Nacionales de Salud del gobierno de Estados Unidos.
- **Codones sinónimos**: término más conocido como *"codon usage bias"*. Refiere a la diferencia en la frecuencia de ocurrencias de codones en la codificación del DNA.

B. Manejo de inputs

Para la manipulación de los datos se usaran cadenas de caracteres que representan tanto cadenas de DNA como cadenas de RNA para representar genes como nucleóticos.

- **nuc_arn** → a | u | c | g | _
- **gen_arn** → (**nuc_arn**)$^+$
- nuc_adn → a | t | c | g | _
- gen_adn → (**nuc_adn**)$^+$

Para formar cadenas más complejas, tales como aminoácido y proteínas, se usará:

- **aminoacido** → Ala | Arg | Asn | ...
- **proteina** → **aminoacido**(**aminoacido**)$^+$

C. Sugerencias

C.1 Pseudo-Código para enmascarar nucleótidos ("M")

Para determinar que nucleótidos deben ser reemplazados por una "M" en la generación de secuencias enmascaradas, se propone el siguiente Pseudo-Código.

```
input: nuc_mensajero, nuc_mirna

if (nuc_mensajero == complemento(nuc_mirna)) {
    if (nuc_mensajero.apareado()){
        print "M"
    }else{
        print upper_case(nuc_mirna)
    }
}else{
        print lower_case(nuc_mirna)
}
```

C.2 Scores matching

Para calcular el score de matching sobre las secuencias (secuencia original, secuencia humanizada) se sugiere la fórmula (1).

$$\frac{(\#AT \times constAT + \#GC \times constGC)}{(totalAT \times constAT + totalGC \times constGC} \qquad (1)$$

donde:

- **#AT:** cantidad de Adenina que hace matching con Timina, o viceversa.
- **#GC:** cantidad de Guanina que hace matching con Citosina, o viceversa.

- **constAT:** valor predeterminado para el apareo A=T.
- **constGC:** análogo al anterior, pero con apareo G=C.
- **total AT:** total de adedina y timina (apareadas o no).
- **totalGC:** total de guanina y citosina (apareadas o no).

Esta fórmula permitirá calcular dos score, uno de ellos empleando constantes **constAT** y **constGC** de valor 1 (cuyo resultado corresponderá a un porcentaje), y para el otro se emplearan constantes de folding a determinar en el análisis.

D. Referencias

[1] IEEE Recommended Practice for Software Requirements Specifications. Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Published 1998. Printed in the United States of America. ISBN 0-7381-0332-2.

[2] SOLID: "Design Principles and Design Patterns", Robert C. Martin. http://www.objectmentor.comresources/articles/Principles_and_Patterns.pdf

[3] C++: Lenguaje de programación. http://www.cplusplus.com

[4] G. Biset, D. Gutson, and M. Arroyo, "A framework for small distributed projects and a protein clusterer application", 2009.

[5] G. Biset, D. Gutson, and M. Arroyo, "Fud: Design and implementation of a framework for small distributed applications", 2009.

[6] B. Meyer, "Object-Oriented Software Construction", Second Edition, Santa Barbara: Prentice Hall Professional Technical Reference, 1997.

[7] G. Booch, J. Rumbaugh, and I. Jacobson, "Unified Modeling Language User Guide", Second Edition, 2005.

[8] RFC 2119. http://tools.ietf.org/html/rfc2119

[9] GNU General Public License. http://www.gnu.org/licenses/

[10] H. Curtis, N. Sue Barnes, A. Schnek and G. Flores, "Biología", Editorial Médica Panamericana S.A, 2006, ISBN: 950-06-0423-X.

[11] B. Pierce, "Genética. Un enfoque conceptual", Tercera Edición, Editorial médica panamericana S.A, ISBN: 978-84-9835-216-0.

[12] A. Blanco, "Química Biológica", Séptima Edición, Editorial El Ateneo.

[13] Δ(G): http://en.wikipedia.org/wiki/Gibbs_free_energy

[14] FuD : http://code.google.com/p/fud/

[15] fudepan-build: http://fudepan-build.googlecode.com

[16] BLAST: http://blast.ncbi.nlm.nih.gov/Blast.cgi

[17] FuDePAN: http://www.fudepan.org.ar/

[18] Vinay S. Mahajan, Adam Drake and Jianzhu Chen, "Virus-specific host miRNAs: antiviral defenses or promoters of persistent infection?".

[19] Man Lung YEUNG, Yamina BENNASSER, Shu Yun LE and Kuan Teh JEANG, "siRNA, miRNA and HIV: promises and challenges".

[20] Gareth M. Jenkins and Edward C. Holmes, "The extent of codon usage bias in human RNA viruses and its evolutionary origin", 2003.

[21] Comeron JM and Aguadé M. "An evaluation of measures of synonymous codon usage bias", 1998.

[22] Haruhiko Siomi and Mikiko C. Siomi, "On the road to reading the RNA-interference code".