

RNAemo

Especificación de Diseño de Software

Franco Gaspar Riberi

7 de junio de 2012

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Descripción general del documento	3
2. Consideraciones de diseño	3
2.1. Objetivos	3
2.2. Metodología	4
2.3. Herramientas y convenciones	4
3. Arquitectura del sistema	4
4. Diseño de alto nivel	6
5. Diseño de bajo nivel	7

1. Introducción

1.1. Propósito

El propósito de este documento es la especificación de diseño de software correspondiente a la primera versión del producto ***RNAemo***.

La confección de este documento se contextualiza dentro del desarrollo de la tesis de grado de la carrera Licenciatura en Ciencias de la Computación de la UNRC, ***RNAemo*** a cargo de Franco Gaspar Riberi, con la dirección de la Lic. Laura Tardivo (UNRC) y las colaboraciones de Daniel Gutson, el Lic. Guillermo Biset y el Dr. Roberto Daniel Rabinovich (**FuDePAN**).

El documento está dirigido a las personas involucradas en el desarrollo de la tesis como así también todos los colaboradores de **FuDePAN** que eventualmente podrían participar en las etapas de desarrollo y mantenimiento del software.

1.2. Descripción general del documento

En la sección 2 se mencionan los objetivos, la metodología adoptada y las dependencias del diseño.

En la sección 3 se exhibe la arquitectura general del sistema con sus principales componentes e interacciones.

En la sección 4 se presenta el diseño de alto nivel del sistema, sus interfaces y paquetes principales.

En la sección 5 se observa el diseño de bajo nivel del sistema, esto involucra las clases concretas y sus relaciones para cada paquete.

2. Consideraciones de diseño

2.1. Objetivos

Se pretende lograr un diseño del sistema que cumpla con los principios fundamentales del diseño orientado a objetos, comúnmente conocidos por el acrónimo “**SOLID**” [1].

En particular, se pretenden respetar los principios **SRP** (*Single Responsibility Principle*), **OCP** (*Open-Closed Principle*) y **DIP** (*Dependency Inversion Principle*) debido a su importancia para obtener un sistema fácilmente extensible y configurable con el fin de satisfacer las necesidades de los usuarios.

2.2. Metodología

La metodología empleada para realizar el análisis y descripción del diseño se denomina “*Diseño dirigido por responsabilidades*” [2].

Esta técnica se enfoca en *qué* acciones (responsabilidades) deben ser cubiertas por el sistema y que objetos serán los responsables de llevarlas a cabo. *Cómo* se realizara cada acción, queda en un segunda plano.

2.3. Herramientas y convenciones

Se utiliza UML[3] como lenguaje de modelado, ArgoUML[4] como herramienta para la confección de diagramas, y Dia[5] para la edición de diagramas de propósito general. Además se adopta la convención de nombrar a las interfaces anteponiendo una “*I*” al nombre de la clase concreta que la implementa. Por ejemplo, interface: “*IPersona*” → clase concreta: “*Persona*”).

3. Arquitectura del sistema

La arquitectura del sistema y la interacción entre los diversos componentes que la conforman se exhiben en la figura 1. A continuación se describe brevemente cada uno de los módulos que comprenden el sistema.

- **Main:** representa el componente principal en términos de ejecución del sistema. Comprenderá la inicialización e invocación de los demás componentes.
- **Generador:** representa la generación de secuencias humanizadas. Dada una secuencia original, genera la secuencia humanizada de la misma. Dicho módulo es externo a este desarrollo, y para ello se empleará el software *GeneDesign*¹.
- **Matching:** representa el componente encargado de realizar el matching por complemento entre secuencias de RNA_m y secuencias de small-RNA_s.
- **DataBase:** representa la base de datos de small-RNA_s (particularmente _{mi}RNA).
- **BLAST:** corresponde a un módulo externo el cual permite el alineamiento de secuencias. Básicamente, compara una secuencia con una gran cantidad de secuencias administradas en una base de datos.

¹Descarga de: <http://www.xmarks.com/site/slam.bs.jhmi.edu/gd/>

- **MasterOfPuppets:** representa el módulo encargado de contabilizar y generar las tablas y gráficos que se esperan como salida de este software.

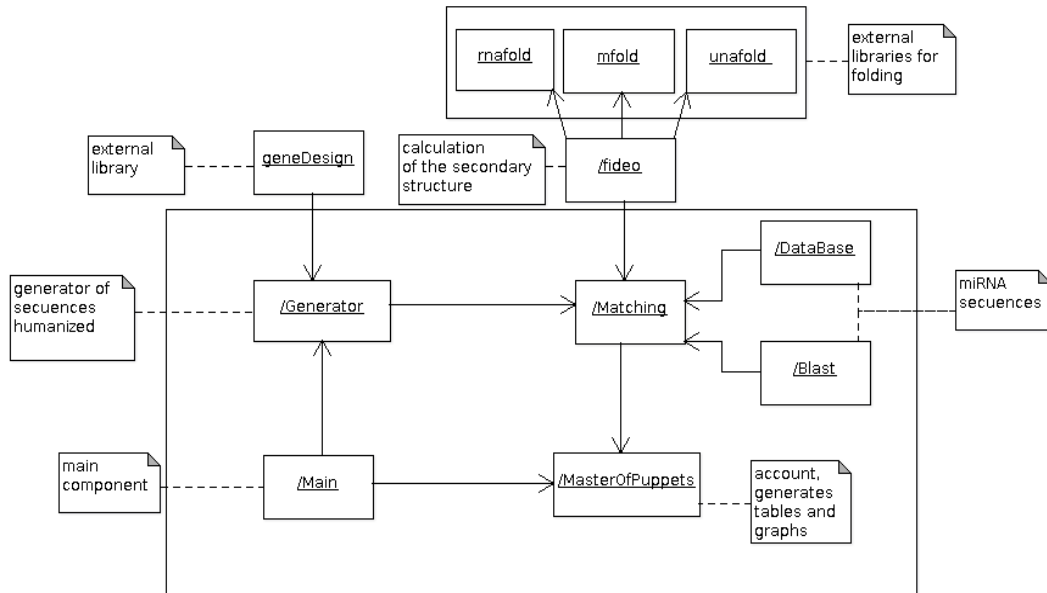


Figura 1: Arquitectura del Sistema

4. Diseño de alto nivel

5. Diseño de bajo nivel

Referencias

- [1] “*Design Principles and Design Patterns.*” ROBERT C. MARTIN, 2000.
<http://www.objectmentor.com>
- [2] “*Object Design: Roles, Responsibilities.*” REBECCA WIRFS-BROCK AND
ALAN MCKEAN AND COLLABORATIONS, Addison-Wesley, 2003.
- [3] “*Unified Modeling Language.*” <http://www.uml.org/>
- [4] “*ArgoUML.*” <http://argouml.tigris.org/>
- [5] “*Dia.*” <http://live.gnome.org/Dia>