
**UNIVERSIDAD NACIONAL DE ENTRE RÍOS**

**FACULTAD DE INGENIERÍA**

**Licenciatura en Bioinformática**

**Anteproyecto de tesina**

# “Software para análisis del nivel de expresión en sondas de ARN”

**Salim Taleb, Nasim A.**

**DNI: 42.477.236**

**Correo: nasimtaleb@hotmail.com**

**Teléfono: +54 9 343 405-9487**

**Diamante, Argentina, 2023**

# Contenido

# Introducción

El análisis de datos biológicos es una tarea muy compleja[1], [2], la cual requiere el uso de herramientas bioinformáticas capaces de manejar estos datos[3], en este caso particular, en datos generados por microarrays[4]. La tecnología de Microarray se desarrolló hace unos 30 años[5], desde entonces dominó el área de la secuenciación de ARN durante varios años, hasta ser desplazada recientemente por ARN-seq[6]; sin embargo, se generó un gran volumen de datos durante estos años en bases de datos como Gene Expression Omnibus (GEO)[7].

La hibridación de ácidos nucleicos es el fundamento de la técnica, consiste en colocar miles de secuencias génicas en lugares determinados sobre un portaobjetos de vidrio llamado chip. Una muestra que contiene ARN se pone en contacto con el chip y se produce la hibridación entre las sondas y los fragmentos de ARN presentes en la muestra[5]. Estos microarrays son utilizados para evaluar niveles de expresión de genes entre distintas muestras para determinar diferencias y establecer patrones comunes en enfermedades como el cáncer y sus consecuencias que afectan vías metabólicas[8].

Existen múltiples herramientas que han sido desarrolladas para el análisis de estos datos, algunas de ellas son MAAPster[9], GenePattern[10] y Genome Studio, de Ilumina[11]. Algunos de los criterios principales a la hora de analizarlas son: instalación local, para no tener que enviar la información a servidores externos, que tengan adecuación funcional, es decir que cumplan con las funciones que sean útiles para el usuario, y que la interfaz de la herramienta sea amigable[12], muchos de estos requerimientos siguen vigentes y son muy importantes a la hora de diseñar software. En la actualidad la construcción del software ha tendido al uso de bibliotecas de lenguajes como R[13] y Python[14], como por ejemplo, las bibliotecas Bioconductor de R[15] y Biopython de Python[16], ya que son de código abierto y están en constante mejora por la comunidad. Y para las interfaces de usuario, la opción más popular en Python resulta ser la liberia “PyQt” o “Pyside”[17], y para R, “Shiny”[18], la cual es una interfaz web, también existen librerías derivadas del paquete “tcltk” como “tickle”[19].

La justificación para el desarrollo se basa en la dificultad que existe para el manejo de grandes volúmenes de datos de expresión y su procesamiento, especialmente para usuarios inexpertos en conocimientos informáticos para obtener información

relevante al área que se esté aplicando, y dado que las herramientas que realizan estos procesamientos suelen ser de pago, como Ingenuity[20], o gratis, pero con una capacidad limitada a la hora de obtener resultados o en los formatos permitidos de entrada.
En el contexto de este trabajo se plantea la creación de una herramienta gratuita basada en estas bibliotecas, que permita el análisis de datos de expresión de microarrays de manera simple e intuitiva a usuarios inexpertos en herramientas bioinformáticas.
La realización de la misma se basará en técnicas de la ingeniería de software siguiendo directivas concretas y etapas como son la elicitación, diseño, codificación, testeo, despliegue y verificación de resultados.
Este software podrá automatizar este proceso y permitir un fácil acceso a los resultados para que, por ejemplo, profesionales que poseen conocimientos específicos en genética puedan utilizarlo en sus investigaciones y no deban invertir esfuerzo en tareas informáticas.

## Objetivo general

- Desarrollar un software que permita el análisis de datos de expresión generados por sondas de ARN, a profesionales sin conocimientos específicos en informática del laboratorio IBioGeM del CICYTTP de Diamante.

## Objetivos específicos

- Elicitar los requerimientos del software con los usuarios.
- Desarrollar un software que permita obtener métricas y graficar información relevante al análisis de las muestras.
- Desarrollar documentación y un manual para los usuarios.
- Contrastar resultados con otras herramientas y verificar la adecuación funcional del software[21].

## Alcance

El software final debe funcionar correctamente y proveer resultados útiles y fáciles de interpretar por los usuarios a los cuales está dirigido, esto quiere decir que el

software debe ser intuitivo y sencillo de usar. También se debe documentar adecuadamente el software y proveer un manual simple dónde se describa su utilización al usuario.

## Límites

Algunos de los límites son que los resultados del software están fuertemente influenciados por la calidad de los datos que se usen como entrada, y que no se tiene mucho conocimiento específico acerca de interfaces y experiencia de usuario (UI/UX) al momento de comenzar con el trabajo.

## Limitaciones

No se cuenta con gran potencia de cómputo ni servidores en los laboratorios y los usuarios no poseen profundos conocimientos informáticos.
Otra limitación que existe es que al no haber otras personas involucradas en el proyecto con conocimiento técnico pueden ocurrir sesgos y pueden dificultarse tareas como el testing, ya que no es óptimo que la misma persona codee y testee su código, una forma de contrarrestar esto es realizar un proceso de validación constante con los usuarios.

## Métodos y técnicas

### Elicitación

Consistirá en obtener, analizar y validar los requerimientos funcionales y no funcionales para el software de parte de los usuarios, mediante entrevistas semiestructuradas con los mismos, dónde se charlará sobre el software, previamente, habiendo analizado posibles caminos de solución y preparando algunas preguntas concretas. Estas serán grabadas mediante una app de grabador de voz para luego ser analizadas detenidamente y no omitir información importante, de ser necesario se transcribirán mediante software de IA como Whisper. Luego estos requerimientos serán colocados en un documento llamado especificación de requerimientos de software (ERS), el cual contendrá los requerimientos funcionales, los requerimientos no funcionales (según ISO/IEC 25010[21]) más relevantes y los requisitos de la interfaz externa, principalmente de la interfaz de usuario[22],

también se discutirán otros aspectos como los formatos de entrada permitidos o necesidad de convertirlos.

## Diseño

Se hará uso de diagramas básicos para la ingeniería de software como son el diagrama de flujo de datos (DFD), casos de uso y diagrama de clases, mediante el uso de la herramienta online “diagrams.net” y complementando información de ser necesario en documentos “.doc”. También se analizarán detenidamente las ventajas y desventajas de algunos de los software que ya existen y tenerlas en cuenta a la hora de diseñar.

## Codificación y testing

Para este paso se empleará como entorno de desarrollo (IDE) Microsoft Visual Studio Code y cómo lenguaje principal Python 3.11, también se hará uso de librerías para manejo de datos biológicos como BioPython, y para las interfaces se empleará PyQt5 con su respectivo kit de herramientas. Alternativamente, se puede optar por el uso del lenguaje R con el entorno de R studio y el paquete Tickle para las interfaces, esto dependerá de los requerimientos.

## Documentación

Se realizará la completa documentación del código de manera que este sea entendible, reutilizable y modificable, de ser necesario en un futuro, y se elaborará un pequeño manual de uso de la herramienta. Se utilizarán herramientas como Sphinx o Doxygen.

## Prueba y despliegue

Previo a desplegar el software en las computadoras de los usuarios, se efectuará una prueba en una máquina virtual para verificar que la instalación pueda funcionar sin problemas en una computadora distinta a la que se está usando para el desarrollo.

## Contrastación de resultados

La etapa final consistirá en la verificación de los resultados mediante una comparación con resultados obtenidos en trabajos previos y con otros software.

Algunos software posibles para esta comparación son DAVID[23], WebMeV[24] y Reactome[25], que fueron utilizados en un trabajo de tesina previo[26].
Considerando que se puedan configurar los mismos parámetros entre los software a comparar, al emplear los mismos datos estos deberían devolver los mismos resultados, dos variables interesantes para comparar los resultados son la cantidad de genes sobre expresados y la cantidad de vías afectadas.
Esto será comparable mediante un gráfico de Bland-Altman, teniendo como variable, la cantidad de genes sobre expresados y vías afectadas en cada par de muestra-control que se esté analizando.
A su vez, suponiendo que se cumpla un test de normalidad, se realizará un análisis más analítico mediante un test-t independiente, o de Welch, en caso de que no se pueda asumir igualdad de varianzas, con nivel de confianza del 95%.
Para evaluar la adecuación funcional se verificará que se cumplan los requisitos funcionales anotados en el documento ERS, que el software realice las funciones que están allí descritas y provea los resultados necesarios, todo esto también está sujeto a opinión de los usuarios que son quienes definen estos requerimientos.

## Estructura de la tesis

1. Portada (1 página)
2. Resumen del proyecto (2 páginas)
3. Agradecimientos (1 página)
3. Índice (1 página)
4. Datos generales (1 página)
5. Introducción(7 páginas)
    - 5.1 Marco teórico (3 páginas)
    - 5.2 Antecedentes(2 páginas)
    - 5.3 Propuesta (2 páginas)
6. Objetivos(1 página)
    - 6.1 Generales
    - 6.2 Específicos
7. Metodología(19 páginas)
    - 7.1 Materiales (5 páginas)
        - 7.1.1 Lenguaje (3 páginas)
        - 7.1.2 Software (2 páginas)
    - 7.2 Metodología (14 páginas)
        - 7.2.1 Elicitación (3 páginas)
        - 7.2.2 Diseño (5 páginas)

# Cronograma

| Actividad | Semana | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Elicitación | ■ | ■ | | | | | | | | | | | | | |
| Diagrama de flujo de datos | | | ■ | | | | | | | | | | | | |
| Casos de uso | | | | ■ | ■ | | | | | | | | | | |
| Diagrama de clases | | | | | | ■ | ■ | | | | | | | | |
| Codificación | | | | | | | | ■ | ■ | ■ | | | | | |
| Testing | | | | | | | | | ■ | ■ | ■ | | | | |
| Despliegue | | | | | | | | | | | ■ | ■ | | | |
| Contrastación de resultados | | | | | | | | | | | | | ■ | ■ | ■ |
| Documentación y manual | | | | | | | | ■ | ■ | ■ | ■ | ■ | | | |
| Redacción de tesis | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

# Referencias


[1] V. Marx, «The big challenges of big data», *Nature*, vol. 498, n.º 7453, pp. 255-260, jun. 2013, doi: 10.1038/498255a.

[2] Z. D. Stephens *et al.*, «Big Data: Astronomical or Genomical?», *PLoS Biol.*, vol. 13, n.º 7, p. e1002195, jul. 2015, doi: 10.1371/journal.pbio.1002195.

[3] C. M. Vinay, S. K. Udayamanoharan, N. Prabhu Basrur, B. Paul, y P. S. Rai, «Current analytical technologies and bioinformatic resources for plant metabolomics data», *Plant Biotechnol. Rep.*, vol. 15, n.º 5, pp. 561-572, oct. 2021, doi: 10.1007/s11816-021-00703-3.

[4] L. Koumakis, C. Mizzi, y G. Potamias, «Chapter 19 - Bioinformatics Tools for Data Analysis», en *Molecular Diagnostics (Third Edition)*, G. P. Patrinos, Ed., Third Edition.Academic Press, 2017, pp. 339-351. doi: https://doi.org/10.1016/B978-0-12-802971-8.00019-5.

[5] M. Schena, D. Shalon, R. W. Davis, y P. O. Brown, «Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray», *Science*, vol. 270, n.º 5235, pp. 467-470, 1995, doi: 10.1126/science.270.5235.467.

[6] R. Lowe, N. Shirley, M. Bleackley, S. Dolan, y T. Shafee, «Transcriptomics technologies.», *PLoS Comput. Biol.*, vol. 13, n.º 5, p. e1005457, may 2017, doi: 10.1371/journal.pcbi.1005457.

[7] R. Edgar, M. Domrachev, y A. E. Lash, «Gene Expression Omnibus: NCBI gene expression and hybridization array data repository.», *Nucleic Acids Res.*, vol. 30, n.º 1, pp. 207-210, ene. 2002, doi: 10.1093/nar/30.1.207.

[8] Y. Temate-Tiagueu *et al.*, «Inferring metabolic pathway activity levels from RNA-Seq data», *BMC Genomics*, vol. 17, n.º 5, p. 542, ago. 2016, doi: 10.1186/s12864-016-2823-y.

[9] «MAAPster». CCR Collaborative Bioinformatics Resource, 3 de febrero de 2023. Accedido: 25 de abril de 2023. [En línea]. Disponible en: https://github.com/CCBR/MicroArrayPipeline

[10] «GenePattern». https://www.genepattern.org/#gsc.tab=0 (accedido 25 de abril de 2023).

[11] «GenomeStudio Software». https://www.illumina.com/techniques/microarrays/array-data-analysis-experimental-design/genomestudio.html (accedido 25 de abril de 2023).

[12] A. Koschmieder, K. Zimmermann, S. Trißl, T. Stoltmann, y U. Leser, «Tools for managing and analyzing microarray data», *Brief. Bioinform.*, vol. 13, n.º 1, pp. 46-60, mar. 2011, doi: 10.1093/bib/bbr010.

[13] J. L. Sepulveda, «Using R and Bioconductor in Clinical Genomics and Transcriptomics», *J. Mol. Diagn.*, vol. 22, n.º 1, pp. 3-20, 2020, doi: https://doi.org/10.1016/j.jmoldx.2019.08.006.

[14] G. Chen *et al.*, «mRNA and lncRNA Expression Profiling of Radiation-Induced Gastric Injury Reveals Potential Radiation-Responsive Transcription Factors.», *Dose-Response Publ. Int. Hormesis Soc.*, vol. 17, n.º 4, p. 1559325819886766, dic. 2019, doi: 10.1177/1559325819886766.

[15] «Bioconductor - Home». https://www.bioconductor.org/ (accedido 25 de abril de 2023).

[16] «Biopython · Biopython». https://biopython.org/ (accedido 25 de abril de 2023).

[17] M. F. L. updated FAQ, «PyQt5 vs PySide2: What's the difference between the two Python Qt libraries?», *Python GUIs*, 21 de junio de 2019. https://www.pythonguis.com/faq/pyqt5-vs-pyside2/ (accedido 25 de abril de 2023).

[18] «Shiny». https://shiny.rstudio.com/ (accedido 25 de abril de 2023).

[19] mikefc, «tickle». 1 de marzo de 2023. Accedido: 25 de abril de 2023. [En línea]. Disponible en: https://github.com/coolbutuseless/tickle

[20] «Ingenuity Pathway Analysis», *QIAGEN Digital Insights*. https://digitalinsights.qiagen.com/products-overview/discovery-insights-portfolio/analysis-and-visualization/qiagen-ipa/ (accedido 9 de mayo de 2023).

[21] «ISO 25010». https://iso25000.com/index.php/normas-iso-25000/iso-25010 (accedido 9

de mayo de 2023).
[22] Asana, «Cómo redactar un documento de requisitos de software (incluye una plantilla) • Asana», *Asana*. https://asana.com/es/resources/software-requirement-document-template (accedido 10 de abril de 2023).
[23] «DAVID Functional Annotation Bioinformatics Microarray Analysis». https://david.ncifcrf.gov/ (accedido 18 de mayo de 2023).
[24] «WebMeV». https://mev.tm4.org/#/about (accedido 18 de mayo de 2023).
[25] «What is Reactome ? - Reactome Pathway Database». https://reactome.org/what-is-reactome (accedido 18 de mayo de 2023).
[26] Pini, Gaston, «Incorporación de recursos computacionales para determinación de la reprogramación de vías metabólicas en cáncer colorrectal», FIUNER.

# Director y evaluadores

**Directora**: Veronica, Lucrecia; Martinez, Marignac
DNI: 22.514.341
Correo: veromm99@gmail.com
Teléfono: +54 9 343 509-1710
CV: adjunto en carpeta

**Evaluador:** Walter, Ricardo; Elias
DNI: 23.368.818
Correo: walter.elias@uner.edu.ar
Teléfono: +54 9 343 406-7888

**Evaluador:** Christian Ariel Mista
DNI: 30.339.662
Correo: christian.mista@uner.edu.ar
Teléfono: 0343 154586737

**Evaluadora:** Silvina, Mariel; Richard
DNI: 22.365.331
Correo: silrichard@yahoo.com.ar
Teléfono: +54 221 618-1775
CV: adjunto en carpeta