Predicción funcional de microRNAs en base a los genes que regulan

Trabajo de fin de máster de Jorge Rdo. Alonso Fernández

Tabla de Contenidos

Extracto....3
Introducción....3
MicroRNAs....3
Bases de datos....3
Resumen y objetivos....4
Desarrollo....5
Materiales....5
Fuentes de datos sobre microRNAs....5
miRTarBase....5
Fuentes de datos sobre funciones de genes....6
Bases de datos....6
Tecnologías Informáticas....7
Entorno....7
Lenguage....7
API REST....8
Métodos....9
Carga de datos de miRTarBase a MongoDB....9
Estructura original de la base de datos....9
Errores y aspectos a mejorar....10
Estructura nueva para MongoDB....11
Recuperación de los datos....12
Integración con gProfiler2....14
Trabajo futuro....15
Carga de la bibliopedia....15
Mejora de la API....15
Resultados....16
Descriptivo de la carga de miRTarBase....16
Casos de validación funcional....16
Discusión....17
Conclusiones....18
Referencias....19
Trabajo propio....19
Abreviaciones....19
Bibliografía....20

Extracto

Introducción

MicroRNAs

En 1993 se descubrió por primera vez un microRNA en *C. Elegans*, llamado *lin-4*, junto con algunas mutaciones que provocaban pérdidas de función. Y no fue hasta el 2000 que se descubrió otro más, también en *C. Elegans* y de función parecida. Desde entonces se han descubierto más funciones de estas moléculas y más mutaciones involucradas en patologías. También se han desarrollado técnicas que los emplean diagnóstico e investigación.

Estos microRNA son moléculas de RNA no codificantes y de longitud reducida (de 18 a 26 nucleótidos) que hibridan con uno o varios mRNA para regular la expresión de los genes que los transcriben, produciendo cambios significativos en varios procesos fisiológicos y patológicos. A medida que se han ido descubriendo más de éstos, han surgido diversas bases de datos que recopilan deferentes aspectos de estas moléculas.

Una de esas bases de datos es miRTarBase, que recopila información verificada manualmente sobre la interacción entre estos microRNA y los genes a los que regulan (MTI). A midida que esta base de datos ha ido creciendo, ha obtenido datos suficientes como para poder describir o predecir las funciones biológicas que regulan.

Sin embargo, miRTarBase no posee funciones ni interfaces que permitan su uso adecuado con herramientas bioinformáticas. Además, al descargarla su formato tampoco permite su uso directo por dichas herramientas. Para poder hacerlo, habría que importar los datos en un sistema que permitiera su explotación computerizada.

Bases de datos

A medida que el estudio de la biología en las últimas décadas ha ido generando más información también han crecido en número, tamaño y tipos las bases de datos que la recogen, lo que a su vez ha generado más retos.

El primer reto es el de gestionar eficientemente el guardado de toda esa información. A mediada que las tecnologías informáticas han avanzado, han salido muchas soluciones optimizadas para diferentes usos. La que vamos a usar en este trabajo, MongoDB, es una base de datos que permite guardar la información de forma descentralizada y copiarla en diferentes nodos en función de las necesidades de uso, todo de forma automática.

El segundo reto es el de poder acceder de manera sistemática al contenido de las bases de datos. Puesto que may muchas bases de datos que se manejan de maneras diferentes, y también hay muchos programas escritos en diferentes lenguajes para analizar esa información, es necesario tener un método de acceso común que puedan usar todos ellos. Las APIs abordan ese problema, permitiendo a los programas comunicarse entre sí mediante diferentes métodos. En este trabajo se emplea una API REST, que es una manera estándar hoy en día de intercambiar información a través de internet, y es compatible con casi todos los lenguages de programación.

Resumen y objetivos

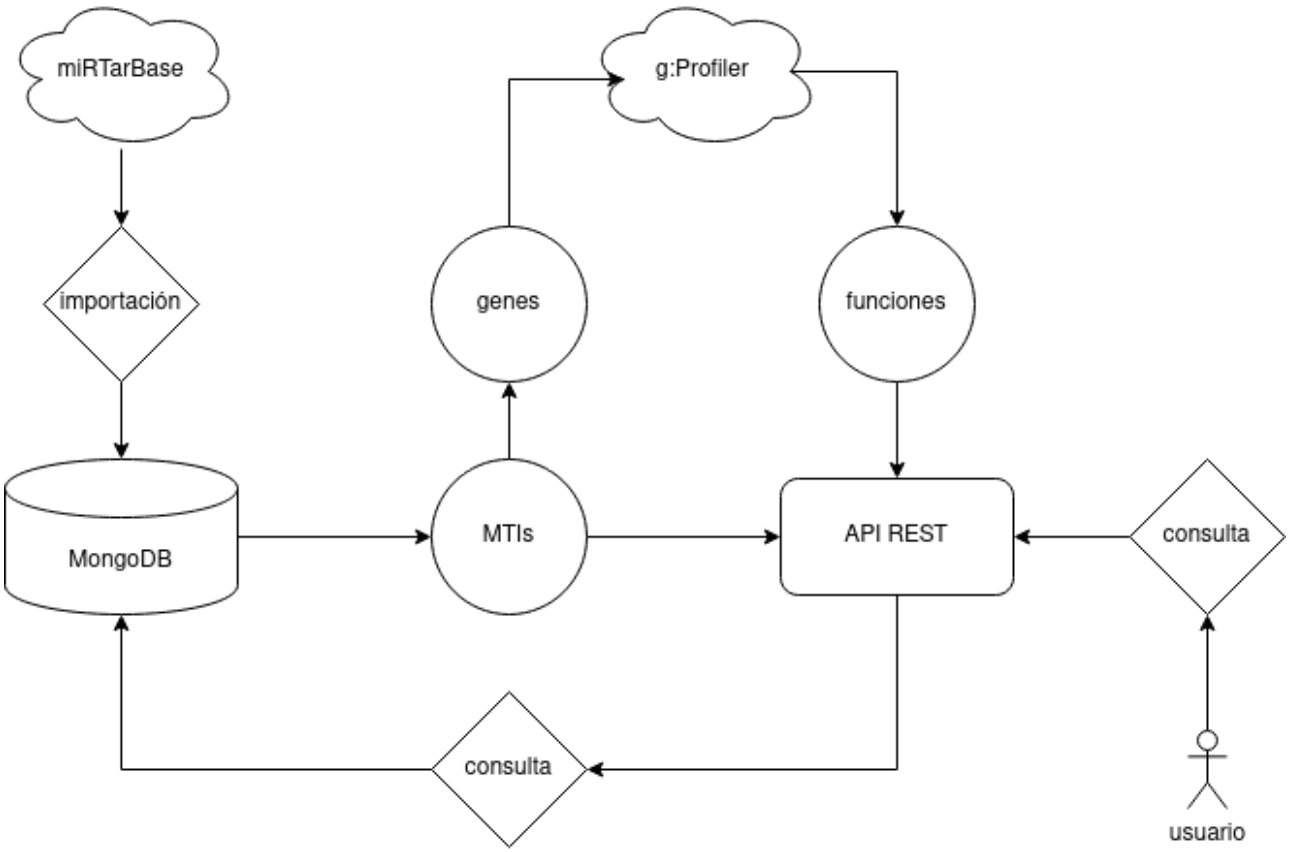
El objetivo principal de este trabajo es concer la función biológica de un determinado microRNA en base a los genes que regula, usando los MTI de miRTarBase como mecanismo para ello. Además, se pretende que el conjunto de scripts desarrollados puedan ser reutilizados para poder importar futuras versiones de miRTarBase con el mínimo de modificaciones posibles.

El portal de miRTarBase no posee ninguna API. Sólo se puede consultar mediante una GUI web o desgargar en formato .xlsx de Microsoft Excel. Para poder explotarla correctamente, se realizará un volcado de miRTarBase a MongoDB, una base de datos no relacional.

Una vez se tenga el contenido en la base de datos local, se desarrollará un servicio REST que permita consultar la base de datos y, además, conecte con otras bases de datos para ofrecer la predicción funcional.

Para este último objetivo se empleará g:Profiler, que devuelve un análisis de enrequecimiento en varios sistemas en función de una lista de genes que recibe como entrada. Se filtrarán los términos GO (de Gene Ontology, un estándar para describir funciones, localizaciones y más de genes) y HP (de Human Phenotype Ontology, un estándar para hablar de fenotipos humanos sanos y patológicos).

Finalmente, para analizar la validez de los resultados del servicio, se realizarán consultas funcionales de microRNAs de los cuáles se conozcan sus funciones y, por lo tanto, los términos HPO y GO esperables.



Desarrollo

Materiales

Fuentes de datos sobre microRNAs

Existen muchas bases de datos que recogen información relacionada com microRNAs. Algunas de ellas son:

- miRBase recoge secuencias de RNA y localizaciones cromosómicas de dichas moléculas. Además, recoge anotaciones y referencias a otras herramientas y bases de datos. No posee una API para consumo computerizado. Las secuencias se pueden descargar en formato fasta, las localizaciones cromosómicas en un archivo .gff3 y el resto de datos en un archivo con formato EMBL.
- TarDB recoge predicciones de MTI de plantas e información relacionada de sus funciones. Tampoco posee una API, pero se pueden descargar los datos en .cons, una tabla con alineaciones.
- miRmap recoge predicciones computerizadas sobre los MTI. Posee una API, interfaz web con muchas opciones, y pueden descargarse los datos en formato libre .csv.
- miR2Disease recoge los MTI implicados en diversas enfermedades humanas. No posee una API, se puede descargar una tabla en .txt.
- miRTarBase recoge evidencias experimentales de los MTI de muchas especies. Tampoco posee una API, se puede descargar como una tabla .xlsx.

Base de datos	Entradas y tipo	Actualización	Especies	Descargas	Interfaz
miRBase	38589 microRNAs, con evid. experimental	Irregular, últ. ver. en 2018	Múltiples	`.dat` con microRNAs y `.gff3` con localizaciones	Web
TarDB	62888 MTIs, de prediciones comput.	Irregular, últ. ver. en 2021	Sólo plantas, 43 en total	`.txt` con alineaciones	Web y API
miRmap	{ #TODO: download and check }	Irregular, últ. ver. en 2022	7 mamíferos y 1 pez	`.csv` con MTIs y secuencias	Web
miR2Disease	3273 MTIs, con evid. experimental	Desconocido	Sólo humanos	`.txt` con mutaciones y alineaciones	Web
miRTarBase	> 2000000 MTIs, con evid. experimental	Cada 2 años, últ. ver. en 2021	Múltiples, 23 en total	`.xlsx` con MTIs	Web

Tabla: comparación de las características de las bases de datos.

miRTarBase

Puesto que los objetivos de este trabajo requieren partir de evidencias experimentales, y se pretende aplicar al máximo de entradas posibles, se ha escogido miRTarBase para obtener información de los MTIs. Esta base de datos surgió en 2011, y durante la última década ha ido recogiendo manualmente imformación sobre MTIs creciendo su tamaño exponencialmente.

Desde su inicio y hasta ahora, la única manera de consultarla ha sido mediante su interfaz web, que han mejorado con cada actualización de la base de datos. Devuelve una tabla con varios campos, que incluyen:

- Un identificador asignado para cada entrada, que corresponde a un MTI
- Los microRNA implicados, junto con la especie del mismo
- Los genes a los que regulan, junto con la especie del mismo
- Los experimentos que demuestran cada MTI, desglosados según la evidencia
    - Asignan evidencia “fuerte” al *reporter assay*, al *western blot* y al *qPCR*, y muestran uno en cada columna.
    - También muestran por separado el *microarray*, la *NGS*, el *pSILAC* y el *CLIP-Seq*.
    - El resto de experimentos apareden como *Other*.
- El conteo de los experimentos y de los artículos publicados donde aparecen

Por desgracia, no han implementado una API que permita el análisis y la consulta computerizadas, y la única manera de descargarlo es en formato .xlsx, que es propietario y no orientado a la programación. Para explotar la información que contiene habrá que exportarlo a la base de datos que hemos escogido.

Fuentes de datos sobre funciones de genes

Para los objetivos de este trabajo también se necesitan fuentes de información sobre funciones celulares de los genes. Para ello se usará g:Profiler, un servidor web para análisis de enriquecimiento funcional. Su uso para este trabajo será recuperar una lista de funciones en términos HP y términos GO en función de una lista de genes que se obtendrán de miRTarBase.

Human Phenotype Ontology es un estándar para la descripción de anormalidades fenotípicas encontradas en enfermedades humanas. Proveen, además del vocabulario controlado, herramientas informáticas para la integración de datos y la investicagión.

Gene Ontology es el recurso más completo y ampliamente utilizado que provee información sobre las funciones de los genes y sobre sus productos. Además de estar diseñado para proveer la información de manera computerizable, usa una ontología formal y bien definida para su estructura. Se basa en términos llamados GO, que tienen un significado concreto y pueden estar definidos usando otros GO. Hay tres tipos de términos: de componente celular, de proceso biológico y de función molecular.

Bases de datos

Debido a la gran cantidad de conocimiento que se genera en relación con la biología, es necesario poder organizar, indexar y recuperar toda esa información. Las bases de datos, que son memorias informáticas en las que pueden integrarse datos dispuestos de modo que sean accesibles individualmente, cumplen exactamente esa función y han proliferado muchas diferentes para cada campo o tema de la biología.

Las primeras bases de datos que surgieron eran bases de datos relacionales. Empezaron a ganar popularidad después de 1970, fueron usadas en muchos ámbitos y usaban principalmente SQL para su manejo y consulta. De entre sus propiedades destacan:

- Tienen una estructura de datos rígida. Eso implica que hay que diseñar y definir su estructura antes de implementarla, pero fuerza a tener consistencia y permite otras características de este tipo de bases de datos.
- Sus implementaciones están diseñadas para tener un nodo centralizado en una máquina potente. Esto encaja con las necesidades conputacionales de su época.

Pero a partir del 2010 el crecimiento exponencial del tráfico de datos promovió el auge de la búsqueda de otros tipos de bases de datos que cubrieran las nuevas necesidades computacionales haciendo uso de el desarrollo de las tecnologías actuales. Surgieron así varias bases de datos llamadas NoSQL. Aunque hay muchas diferentes, suelen coincidir en que:

- Descartan la estructura rígida, puesto que la mayoría de usos no requieren esta complejidad. Se reemplazan con varios modelos de datos (de grafos, colecciones, documentos, llaves, columnas, *etc.*) que suelen permitir escalar las bases de datos horizontalmente, es decir, cambiar o añadir campos.
- Sus implementaciones permiten el servicio descentralizado, permitiendo escalar el hardware fácilmente y aumentar el rendimiento en conjunto. Además, reduce los costes del mismo.

La base de datos escogida para este proyecto es MongoDB, una base de datos con modelo de documentos. Las características que la hacen adecuada para este trabajo son:

- Como es documental, se basa en llaves que pueden tener valores u otras llaves anidadas. Esto permite definir estructuras con campos arbitrariamente relacionados. Además, esta estructura puede representarse bien en como JSON, que es el estándar para transacciones en internet, o como diccionarios de Python, que es el lenguaje más extendido en el campo de la biología.
- Permite el *sharding*, que es un método efectivo para distribuir dato en múltiples máquinas.
    - Su funcionamiento se basa en dividir la base de datos en varios *shards*, que contienen una porción de la base de datos y pueden replicarse y distribuirse en función de las necesidades de uso de cada una a lo largo del tiempo.
    - Por ejemplo, si se dividen los datos de miRTarBase por especies, las especies más consultadas tendrán copias de su *shard*, por lo que las consultas podrán ser respondidas por esos varios ordenadores que las contengan.
    - Esto ayuda a acomodar grandes cantidades de datos o a trabajar con recursos de harware limitados. También permite soportar grandes cargas de consultas a través de internet.

Tecnologías Informáticas

Entorno

Este trabajo se ha desarrollado con Ubuntu 22.04 LTS, una distribución de Linux, como sistema operativo. Como editor se ha empleado Visual Studio Code Community Edition, que permite visualizar código de una manera más clara y provee de funciones que ayudan a escribirlo, formatearlo y corregirlo.

Lenguage

Como lenguage de programación se ha usado Python 3.10, un lenguage interpretado de alto nivel. A diferencia de los lenguages compilados, permite programar sin necesidad de tener que interaccionar con el hardware y a diferencia de los lenguages de bajo nivel, permite una sintaxis más legible y sencilla, lo que lo ha convertido en el lenguage más empleado en el campo de la biología.

Para desarrollar este trabajo, además de las librerías nativas de Python, se han usado otras librerías externas junto con sus dependencias.

La librería requests simplifica, con respecto a la librería nativa, el envío de peticiones HTTP y el procesamiento de sus respuestas. En este trabajo se ha usado para la descarga automatizada de ficheros y para pruebas de conexión.

La librería pandas es una herramienta rápida y flexible de análisis y manipulación de datos. Permite importar, modificar y exportar bases de datos de y a diferentes formatos. Se ha usado para trabajar con los ficheros descargados de miRTarBase.

La librería pymongo es la herramienta oficial recomendada por MongoDB para trabajar desde Python. Permite administrar instalaciones locales o remotas de MongoDB y guardar o recuperar datos de ellas. En este trabajo se ha usado tanto para guardar y servir los datos de miRTarBase.

Flask es una infraestructura ligera para desarrollo de aplicaciones web en Python. Aunque para la implementación final se usarán otros recursos, en este trabajo se ha usado para montar un prototipo de la API REST final.

Finalmente la librería gprofiler-official es interfaz oficial de g:Profiler para consultar desde Python. En este trabajo se usa para consultar g:Profiler sin tener que escribir una gran cantidad de código con requests.

API REST

Para poder enviar a través de internet las funciones de los microRNAs que se soliciten, se ha usado una API REST. Las APIs son interfaces para los programas que permiten comunicarse con otros programas y automatizar procesos que requieran varios de ellos, tanto en local como a través de internet.

Con respecto a REST, es una arquitectura cuyo diseño de intercambio de datos permite a cualquier aplicación desarrollada con cualquier lenguaje información de una manera estándar sin necesidad de implementaciones complejas o específicas. Permite una variedad de ventajas con respecto a la principal alternativa SOAP.

	REST	SOAP
acceso	recursos	operaciones
API pública	única	múltiple
interfaz	consistente	variable
formatos	varios	sólo XML
caché	permitida	absente

Tabla: comparación entre REST y SOAP.

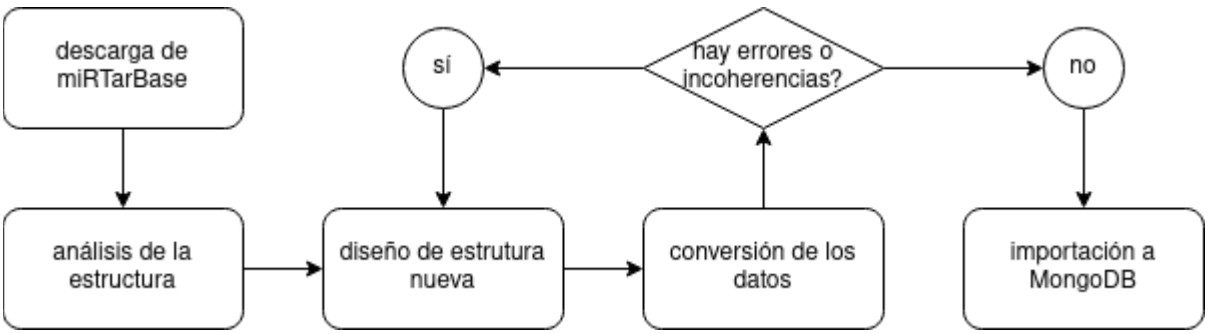
Así, las ventajas de tener una API REST comprenden desde el rendimiento hasta la consistencia en el uso. Además, suele ser implementada con HTTP, que permite conexiones a internet, y JSON, que también puede utilizarse con la API de MongoDB.

Métodos

Carga de datos de miRTarBase a MongoDB

Para cargar los datos a MongoDB se han desarrollado scripts de python que realizan los siguientes pasos:

- Descarga de los ficheros .xlsx de miRTarBase a partir de una lista con sus URL.
  - Se ha realizado con un script simple usando la librería requests.
- Conversión de .xlsx a .json, que es el formato más adecuado para importar con MongoDB. Para esta conversión, es necesario:
  - Estudiar la estructura de la base de datos original.
  - Diseñar una nueva estructura que acomode los datos y permita su uso posterior.
  - Corregir las inconsistencias o los errores de ambas estrucuras, si los hubieran.
- Instalación o configuración de MongoDB y subida de los ficheros .json generados.
  - En este trabajo se ha usado la instalación por defecto de MongoDB que ofrece apt en Ubuntu.
  - Se ha realizado la subida con un script simple usando la librería pymongo.



Estructura original de la base de datos

Para convertir la base de datos y optimizarla para nuestras herramientas y necesidades, es necesario analizar la estructura original de la base de datos original, que es una tabla con las siguientes cabeceras:

- miRTarBase ID es el identificador que asigna miRTarBase a cada MTI.
- miRNA es el identificador del microRNA, que hace referencia a otras bases de datos.
- Species (miRNA) es la especie en la cuál se ha identificado el microRNA.
- Target Gene es el gen a cuyo mRNA se adhiere el microRNA.
- Target Gene (Entrez ID) identificador Entrez para el gen anterior.
- Species (Target Gene) especie para el gen anterior.
- Experiments lista de experimentos que usan en un artículo para investigar el MTI.
- Support Type evaluación de los autores de miRTarBase sobre los experimentos.
- References (PMID) identificador de PubMed para un artículo que describe un MTI.

miRTar Base ID	miRNA	Species (miRNA)	Target Gene	Target Gene (Entrez ID)	Species (Target Gene)	Experiments	Support Type	References (PMID)
MIRT00 0515	dre-miR-125b-5p	Danio rerio	tp53	30590	Danio rerio	Luciferase reporter assay// Western blot	Functional MTI	20216554
MIRT00 0515	dre-miR-125b-5p	Danio rerio	tp53	30590	Danio rerio	In situ hybridization// Luciferase reporter assay//*(etc.)*	Functional MTI	19293287
MIRT00 0515	dre-miR-125b-5p	Danio rerio	tp53	30590	Danio rerio	Luciferase reporter assay// qRT-PCR//Western blot	Functional MTI	21935352

Tabla: ejemplo de una entrada MTI en el fichero .xlsx de miRTarBase.

Puesto que hay varios artículos que hacen referencia al mismo MTI, y varios experimentos que hacen referencia a los mismos artículos, los autores han usado las dos soluciones más comunes para forzar ese tipo de datos en una tabla:

- Para los campos de los artículos (Support Type y References (PMID)) han copiado las filas a las que hacen referencia y han variado esas columnas.
- Para el campo Experiments han usado una secuencia de caracteres como separados entre los ítems, que han juntado en la misma celda.

Errores y aspectos a mejorar

Puesto que las columans de los artículos repiten cada MTI al que hacen referencia, la base de datos original tiene la información de los MTI muy repetida y la hace innecesariamente pesada.

El campo Experiments tiene varios problemas. Los nombres de las técnicas son inconsistentes, usando o no abreviaciones y sinónimos. Además, hay muchos errores de otrografía arbitrarios y diferentes en diferentes entradas de los nombres de cada técnica. Finalmente, los separadores cambian en algunas entradas. En la mayoría son dos barras // o punto y coma ;.

El campo Target Gene recoge genes cuyos nombres suelen estar compuestos por pocas letras y números. Puesto que la base de datos se ha guardado en .xlsx, y que Excel puede reemplazar conjuntos de letras y números por fechas, cambiando no sólo la representación sino el valor (se reemplaza por una estampa de tiempo), estas entradas pierden la información irreversiblemente. Además, hay unps pocos Target Gene que tienen diferentes Target Gene (Entrez ID) asociados.

Finalmente, el campo miRTarBase ID debería ser un identificador único para cada MTI, pero en un análisis de hsa_MTI.xlsx he identificado más de 16 000 conflictos (distintos pares de MTI a los que les han asignado el mismo ID). Me pondré en contacto con el grupo que confecciona la base de datos para informarles una vez haya acabado este trabajo.

Estructura nueva para MongoDB

El mejor diseño que he considerado para los objetivos de este trabajo es la siguiente:

```
entry = {
   "_id": "dre-miR-125b-5p_tp53", # miRNA + _ + Target Gene
   "mirtarbase_id": "MIRT000515", # miRTarBase ID
   "mirna_symbol": "dre-miR-125b-5p", # miRNA
   "mirna_specie": "Danio rerio", # Species (miRNA)
   "gene_symbol": "tp53", # Target Gene
   "gene_entrez": 30590, # Target Gene (Entrez ID)
   "gene_specie": "Danio rerio", # Species (Target Gene)
   "experiments": [
      "Luciferase reporter assay",
      "Western blot",
      "In situ hybridization",
      "qRT-PCR",
      "(etc.)",
   ], # Experiments
   "support_type": 4, # Support Type
   "pubmed_ids": [
      20216554,
      19293287,
      21935352,
   ], # References (PMID)
}
```

Bloque: ejemplo de entrada formateada para ser importada a MongoDB, basada en la tabla del ejemplo anterior.

Para permitir una integración más sencilla con otros programas y APIs, he renombrado los campos para que no contengan espacios ni caracteres especiales.

El campo _id es un identificador que necesita MongoDB para cada entrada. Aunque lo mejor sería usar miRTarBase ID como _id, para evitar los conflictos sin modificar los datos he tenido que usar un ínidce compuesto que debería caracterizar el MTI inequívocamente. Aún así, he descartado un conjunto pequeño de entradas que provocaban conflictos con este _id.

El campo pubmed_ids resulta de juntar en una lista todos los artículos que hacen referencia al mismo MTI, y el campo support_type es el valor más relevante de entre todos los correspondientes a esos artículos, simplificado a un número.

MTI	Strong	Weak	None
Functional	4	3	
Non-Functional	2	1	0

Tabla: asignación de variables numéricas para support_type.

Finalmente, el campo experiments corresponde a la lista de todos los experimentos de todos los artículos. Puesto que no es necesario para los objetivos de este trabajo, esta estructura descarta la relación entre cada conjunto de experimentos y su artículo, juntándolos todos para cada MTI.

Para poder lidiar con los otros problemas descritos anteriormente sobre este campo, he usado varios pasos:

- Para la separación de los elementos en la base de datos original, lo he implementado de manera que se intentan usar los dos separadores principales, // y ;. El resto de entradas se dejan sin resolver en este paso.
- Para estandarizar los elementos de manera que se puedan usar programáticamente, además de para resolver algunos casos aislados de otros separadores como / o ,, he desarrollado un diccionario de Python que relaciona los elementos con sus correcciones y lo he rellenado manualmente.
    - El script pregunta cómo corregir cada entrada al detectarla si no está ya presente en dicho diccionario, y actualiza el diccionario al recibir esas correcciones. Esto será útil para las futuras versiones de las bases de datos que se tengan que importar.
    - Debido a la similitud de algunos nombres de técnicas y a la diversidad y cantidad de errores de otrografía, no he podido desarrollar ningún algoritmo para ayudar a corregir las entradas automáticamente.

Recuperación de los datos

Para la consulta de los datos he desarrollado una sencilla API REST local, que luego se podrá desplegar en el servidor del grupo de investigación para el que hago este trabajo. Tiene dos rutas y la raíz /, que devuelve una pequeña ayuda sobre como usar el servicio.

En /search se recibe uno o varios argumentos (campos de la base de datos con valores para buscar) y se devuelven todos los MTI que lo contienen sus correspondientes parámetros. Esta ruta equivaldría a una API para consultar miRTarBase.

Ejemplos de consulta:

- http://127.0.0.1:5000/search?mirna_symbol=hsa-miR-222-3p devuelve los MTI que se conocen para el microRNA hsa-miR-222-3p, y consecuentemente los genes que regula.
- http://127.0.0.1:5000/search?gene_symbol=RAN&experiments=Western%20blot devuelve los MTI en los que participa el gen RAN, y consecuentemente los microRNA que lo regulan. Además, limita los resultados a los MTI cuyas evidancias experimentales incluyan Western blot.
- http://127.0.0.1:5000/search?pubmed_ids=19438724 devuelve los MTI para los que el artículo 19438724 proporciona evidencia.

Ejemplo de respuesta:


```
[
  {
    "_id": "hsa-miR-222-3p_BCL2L11",
    "experiments": [ "Luciferase reporter assay", "PAR CLIP", "Western blot" ],
    "gene_entrez": 10018, "gene_specie": "Homo sapiens", "gene_symbol": "BCL2L11",
        "mirna_specie": "Homo sapiens", "mirna_symbol": "hsa-miR-222-3p", "mirtarbase_id":
"MIRT000134",
    "pubmed_ids": [ 33942856, 19438724, 23446348, 20371350 ], "support_type": 3
  },
  {
    "_id": "hsa-miR-221-3p_BCL2L11",
    "experiments": [
      "Real Time PCR (qPCR)", "Luciferase reporter assay", "PAR CLIP",
      "Western blot", "Cross-linking, Ligation, and Sequencing of Hybrids (CLASH)"
    ],
    "gene_entrez": 10018, "gene_specie": "Homo sapiens", "gene_symbol": "BCL2L11",
        "mirna_specie": "Homo sapiens", "mirna_symbol": "hsa-miR-221-3p", "mirtarbase_id":
"MIRT000140",
    "pubmed_ids": [ 19438724, 23622248, 26503209, 23446348, 20371350 ], "support_type": 4
  }, "...",
  {
    "_id": "rno-miR-222-3p_Bcl2l11",
      "experiments": [ "Immunoblot", "Real Time PCR (qPCR)", "Reporter assay", "Luciferase
reporter assay" ],
    "gene_entrez": 64547, "gene_specie": "Rattus norvegicus", "gene_symbol": "Bcl2l11",
      "mirna_specie": "Rattus norvegicus", "mirna_symbol": "rno-miR-222-3p", "mirtarbase_id":
"MIRT004034",
    "pubmed_ids": [ 19438724 ], "support_type": 4
  }
]
```


Bloque: ejemplo de respuesta para los MTI descritos en el artículo 19438724.

En /detail se recibe el identificador de un microRNA bajo su clave en la base de datos mirna_symbol. Además, también se acepta como patámetro opcional el support_type para limitar los resultados. Esta ruta devuelve la caracterización funcional del microRNA, basándose en los genes de sus MTI con evidencia suficiente para deducirla. El concepto se explica en detalle en el siguiente apartado.

Un ejemplo de consulta sería http://127.0.0.1:5000/detail?mirna_symbol=hsa-miR-664b-5p, que devolvería la caracterización funcional del microRNA hsa-miR-664b-5p.


```
{
  "Homo sapiens": [
    {
      "description": "\"Any process that modulates the frequency, rate or extent of the chemical
reactions and pathways resulting in the formation of substances, carried out by individual
cells.\" [GOC:mah]",
      "name": "regulation of cellular biosynthetic process",
      "native": "GO:0031326"
    }, "...",
    {
        "description": "\"The chemical reactions and pathways resulting in the formation of
aromatic compounds, any substance containing an aromatic carbon ring.\" [GOC:ai]",
      "name": "aromatic compound biosynthetic process",
      "native": "GO:0019438"
    },
    {
      "description": "\"Any process that modulates the frequency, rate or extent of the chemical
reactions and pathways by which individual cells transform chemical substances.\" [GOC:mah]",
      "name": "regulation of cellular metabolic process",
      "native": "GO:0031323"
    }
  ]
}
```


Bloque: ejemplo de respuesta para la caracterización funcional de hsa-miR-664b-5p.

Integración con gProfiler2

El objetivo principal de este trabajo es poder ofrecer una predicción de caracterización funcional de un microRNA en base a a los genes que regula, que se ofrecerá a través de una API REST. En la implementación actual corresponde a la ruta /detail y recibe el nombre del microRNA como argumento mirna_symbol.

Al recibir la petición se recuperan las entradas correspondientes a ese mirna_symbol en la base de datos importada, de manera equivalente a lo que sucedería en /search. Pero en vez de devolver los resultados, se agrupan los genes de esas entradas en función de la especie del gen.

Después, para cada grupo de genes se hace una consulta en tiempo real a g:Profiler para recibir la predicción funcional de éstos, que correspondería a la predicción de la función del microRNA. Esta consulta se hace a través del paquete gprofiler-official, la librería de Python que g:Profiler provee.

Finalmente se devuelve una lista simplificada de las predicciones funcionales para cada especie de esos genes, tal y como se aprecia en el apartado anterior.

Trabajo futuro

Puedo quitar esta seccion si no hay tiempo de que me diga lo de la carga en la bibliopedia.

Carga de la bibliopedia

{ #TODO: hablar de cómo haremos lo de la bibliopedia, aunque lo haga después de entregar la memoria }

Mejora de la API

{ #TODO: más métodos de búsqueda, generador de búsqueda, interfaz web }

Resultados

Descriptivo de la carga de miRTarBase

En total se han importado 455087 entradas, la mayoría siendo MTIs de microRNA humano. Esto contrasta con el número de entradas que dice tener miRTarBase, más de 2000000. Esto se debe a que aunque se han corregido muchas entradas para no descartarlas, hay un problema en los ficheros que miRTarBase provee que impiden leer más entradas correctamente.

En total hay 23 especies, y las que tienen más entradas son:

mirna_specie	Count
Homo sapiens	394973
Mus musculus	55012
Caenorhabditis elegans	3236
Rattus norvegicus	796
Bos taurus	298

A pesar de la gran cantidad de MTIs registrados en la base de datos, los microRNAs que los componen son muy limitados, siendo 4993 en total. De entre éstos, los que tienen más entradas son:

mirna_symbol	Count
hsa-miR-335-5p	2705
hsa-miR-26b-5p	1935
hsa-miR-16-5p	1606
hsa-miR-124-3p	1527

Finalmente los genes son algo más variados, llegando a 24429 distintos. Los que tienen más entradas son:

gene_symbol	Count
ZNF460	359
CDKN1A	335
NUFIP2	331
AGO2	308

Casos de validación funcional

Para comprobar la validez del enriquecimiento funcional, se buscarán microRNAs de los cuales se conozcan las funciones y se contrastará con la respuesta de la API REST implementada.

Validación funcional con GO

Para comprobar la adecuación de los términos GO, que describen funciones, reacciones y localizaciones, podemos hacer una consulta con un microRNA cuyas funciones sean conocidas y esten bien descritas.

Como ejemplo se va a utilizar *lin-4* de *Caenorhabditis elegans*, el primer microRNA descrito en la literatura científica. Las mutaciones en éste provocan diversos problemas en el desarrollo e impiden la reproducción.

Su identificador en miRTarBase corresponde a cel-lin-4-5p, por lo que la URL de consulta es http://127.0.0.1:5000/detail?mirna_symbol=cel-lin-4-5p y su respuesta contiene los siguientes términos:

Término ("native")	Descripción corta ("name")
GO:0040034	regulation of development, heterochronic
GO:0004019	adenylosuccinate synthase activity

Tabla: términos en la respuesta de la consulta para lin-4.

Como es de esperar, aparece un término de función biológica que hace referencia a la regulación del desarrollo. Además, aparece un término de reaccion molecular sobre la actividad adenylosuccinato.

He buscado información sobre ésta, y resulta que se ha demostrado que la acrividad

Validación funcional con HPO

Para comprobar el correcto funcionamiento con términos HPO, que describen fenotipos en patologías humanas, podemos hacer una consulta con algún microRNA que se sepa que está involucrado en alguna patología y deberíamos obtener varios términos HP que describan esas anormalidades denotípicas, además de los términos GO.

Como ejemplo se va a utilizar uno de los primeros microRNAs descubierto en cáncer humano, el miR-15. Primero se observó en leucemia linfocítica crónica, y luego se encontró también en varios cánceres que afectan al cerebro.

Su identificador en miRTarBase corresponde a hsa-miR-15a-5p, por lo que la URL de consulta es http://127.0.0.1:5000/detail?mirna_symbol=hsa-miR-15a-5p y su respuesta contiene, además de múltiples términos GO, 5 términos HP:

Término ("native")	Descripción corta ("name")
HP:0002011	Morphological central nervous system abnormality
HP:0012639	Abnormal nervous system morphology
HP:0002683	Abnormal calvaria morphology
HP:0012443	Abnormality of brain morphology
HP:0002152	Hyperproteinemia

Tabla: términos HP incluidos en la respuesta de la consulta para miR-15.

Como se puede apreciar, un término indica la presencia de grandes cantidades de proteína en sangre y el resto aluden a anormalidades morfológicas en el cerebro. Así, podemos apreciar que la consulta ha relacionado efectivamente este microRNA con términos HP sobre fenotipos esperables para la leucemia y el cáncer en cerebro.

Discusión

Tal y como se puede apreciar al consultar la base de datos miRTarBase, contiene varios errores y no está optimizada para el uso computerizado. Es por ello que ha sido necesario rediseñar, corregir y reimplementar dicha base de datos para poder hacer consultas complejas o automatizadas.

Y es que a pesar de que hoy en día las tecnologías informáticas y sus correspondientes bases de datos han llegado a un punto de su desarrollo que están integradas en una infinidad de servicios que usamos a diario, muchas de las bases de datos bioinformáticas están estancadas. Algunos de los motivos más probables son que no hay interés por parte de los investigadores de escribir sus artículos y publicar su información de manera que pueda ser aprovechada de manera automática, o bien porque los investigadores del campo de la biología no tienen conocimiento informático suficiente como para hacerlo.

{ #TODO: discutir más, no estoy seguro de como enfocarlo }

Conclusiones

{ #TODO: enfocarlo mejor }

Referencias

Trabajo propio

Todo el código escrito para este trabajo está disponible en el siguiente repositorio de github:

https://github.com/jricardo-um/mirtarbase-importing

Abreviaciones

Abreviación	Inglés	Español
RNA	ribonucleic acid	ácido ribonucléico
miRNAμRNA	micro-RNA	micro RNA
mRNA	messenger RNA	RNA mensajero
MTI	miRNA-target interaction	interacción de miRNA con su diana
API	application program interface	interfaz de programación de aplicaciones
GUI	graphical user interface	interfaz gráfica de usuario
URL	unique resource locator	localizador de recurso único
SOAP	simple object access protocol	protocolo de acceso a objeto simple
REST	representation state transfer	transferencia de estado representado
SQL	structured query language	lenguaje de consulta estructurada
NoSQL	not only SQL	no sólo SQL

Bibliografía

intro > mirnas

Maria I. Almeida, Rui M. Reis, George A. Calin, MicroRNA history: Discovery, recent applications, and next frontiers, Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis, Volume 717, Issues 1–2, 2011, Pages 1-8, ISSN 0027-5107, https://doi.org/10.1016/j.mrfmmm.2011.03.009

Huang HY, Lin YC, Cui S, *et al*. miRTarBase update 2022: an informative resource for experimentally validated miRNA-target interactions. Nucleic Acids Res. 2022 Jan 7;50(D1):D222-D230. doi: https://doi.org/10.1093/nar/gkab1079

intro > bases de datos

Stephan Philippi, Light-weight integration of molecular biological databases, *Bioinformatics*, Volume 20, Issue 1, January 2004, Pages 51–57, https://doi.org/10.1093/bioinformatics/btg372

Dan M. Bolser *et al*. MetaBase - the wiki-database of biological databases, *Nucleic Acids Research*, Volume 40, Issue D1, 1 January 2012, Pages D1250–D1254, https://doi.org/10.1093/nar/gkr1099

materiales > fuentes micrornas

Kozomara A., Birgaoanu M., Griffiths-Jones S. miRBase: from microRNA sequences to function. Nucleic Acids Res 2019 47:D155-D162. https://doi.org/10.1093/nar/gky1141

Liu J, *et al*. TarDB: an online database for plant miRNA targets and miRNA-triggered phased siRNAs. BMC Genomics. 2021; 22(1):348. https://doi.org/10.1186/s12864-021-07680-5

Charles E. Vejnar, Evgeny M. Zdobnov. miRmap: Comprehensive prediction of microRNA target repression strength Nucleic Acids Research 2012 Dec 1;40(22):11673-83. https://doi.org/10.1093/nar/gks901

Jiang Q., Wang Y., Hao Y., *et al.* (2009) miR2Disease: a manually curated database for microRNA deregulation in human disease. *Nucleic Acids Res*. https://doi.org/10.1093/nar/gkn714

materiales > fuentes funciones

Sebastian Köhler *et al*. The Human Phenotype Ontology in 2021, *Nucleic Acids Research*, Volume 49, Issue D1, 8 January 2021, Pages D1207–D1217, https://doi.org/10.1093/nar/gkaa1043

The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong, *Nucleic Acids Research*, Volume 47, Issue D1, 08 January 2019, Pages D330–D338, https://doi.org/10.1093/nar/gky1055

materiales > bases de datos

J. E. Pagán, J. S. Cuadrado, J. G. Molina. A repository for scalable model management. *Softw Syst Model* **14**, 219–239 (2015). https://doi.org/10.1007/s10270-013-0326-8

M. W. Khan, E. Abbasi (2015). Differentiating Parameters for Selecting Simple Object Access Protocol (SOAP) vs. Representational State Transfer (REST) Based Architecture. *Journal of Advances in Computer Networks*, *3*(1), 63-6. http://dx.doi.org/10.7763/JACN.2015.V3.143

materiales > tecnologias

Requests documentation https://requests.readthedocs.io/en/stable/

W. McKinney, *et al*. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56). https://doi.org/10.5281/zenodo.3509134

PyMongo documentation https://pymongo.readthedocs.io/en/stable/

GProfiler on Python Package Index https://pypi.org/project/gprofiler-official/

M. Grinberg (2018). *Flask web development: developing web applications with python*. O'Reilly Media, Inc. https://dl.acm.org/doi/book/10.5555/2621997

Y. Punia, R. R. Aggarwal (2014). below: Implementing Information System Using MongoDB and Redis. https://www.warse.org/IJATCSE/static/pdf/file/icace2014sp05.pdf

MongoDB documentation https://www.mongodb.com/docs/manual/sharding/

materiales > api rest

Andrew Yates *et al*, The Ensembl REST API: Ensembl Data for Any Language, *Bioinformatics*, Volume 31, Issue 1, January 2015, Pages 143–145, https://doi.org/10.1093/bioinformatics/btu613

recultados > validación funcional

R. Marsac, B. Pinson, C. Saint-Marc *et al* (2019). Purine Homeostasis Is Necessary for Developmental Timing, Germline Maintenance and Muscle Integrity in *Caenorhabditis elegans*. *Genetics*, 211(4), 1297–1313. https://doi.org/10.1534/genetics.118.301062

Ivo D'Urso Pietro, Fernando D'Urso Oscar *et al*. miR-15b and miR-21 as Circulating Biomarkers for Diagnosis of Glioma, Current Genomics 2015; 16(5) . https://dx.doi.org/10.2174/1389202916666150707155610

INFORMACIÓN SOBRE LA ELABORACIÓN DE LA MEMORIA (guía docente)

El trabajo se concluirá con la elaboración de una memoria, que deberá tener la estructura de un trabajo científico (resumen, introducción, materiales y métodos, resultados, discusión, conclusiones, referencias) y una extensión máxima de 25 páginas en formato de 1 columna. Las 25 páginas incluyen desde el resumen hasta las referencias, excluyendo las páginas en blanco intermedias que se pudieran usar por cuestiones de formato. Se recomienda una orientación vertical y espaciado vertical sencillo, con unos márgenes mínimos de 2 cm (superior e inferior) y de 2.5 cm (izquierdo y derecho), letra Arial-10 o Arial Narrow-10 (texto) y Arial Narrow-10 (bibliografía).

La memoria se podrá escribir en inglés o en español. Si el estudiante opta por escribir la memoria en inglés, ésta deberá incluir una traducción al español del apartado "Resumen".

Si la investigación realizada ha dado lugar a resultados recogidos en algún Trabajo de Investigación (publicado o no) o Comunicación a Congreso, se hará constar en la Memoria.

Cuando el trabajo haya sido realizado en el seno de una entidad externa, el estudiante deberá asegurarse de que no incumple ningún contrato de confidencialidad ni viola ningún derecho de propiedad intelectual. En este caso se deberá incluir en la memoria la autorización expresa por parte de la empresa para realizar y presentar el trabajo y la citada memoria. La Universidad de Murcia se exime de cualquier responsabilidad derivada del no cumplimiento de este reglamento por parte del estudiante.

La memoria se entregará en formato PDF a través de la plataforma TF en las fechas establecidas a tal efecto y cumpliendo las limitaciones de tamaño vigentes en la plataforma.