- Tesis Doctoral -

VNiVERSiDAD
D SALAMANCA

Bioinformática aplicada a datos genómicos para la caracterización de subtipos de cáncer: estudios integrativos en hemopatías malignas

Bioinformatic strategies to analyze multiple cancer subtypes and build associated gene networks using genomic profiling: integrative studies on hematological malignancies

Sara Aibar Santos

Director

Dr. Javier De Las Rivas

Centro de Investigación del Cáncer (CiC-IBMCC)

Universidad de Salamanca

2015

Esta tesis doctoral corresponde a un compendio de trabajos previamente publicados o aceptados para publicación:

Artículo 1: **Analyse multiple disease subtypes and build associated gene networks using genome-wide expression profiles**

Autores: Sara Aibar, Celia Fontanillo, Conrad Droste, Beatriz Rosón, Francisco J. Campos-Laborie, Jesus M. Hernández-Rivas and Javier De Las Rivas

Afiliación de los autores: Centro de Investigación del Cáncer (CSIC/USAL/IBSAL), Salamanca, Spain.

Aceptado en *BMC Genomics* (2015) Vol 16 Suppl 4

Artículo 2: **Functional Gene Networks: R/Bioc package to generate and analyse gene networks derived from functional enrichment and clustering**

Autores: Sara Aibar, Celia Fontanillo, Conrad Droste and Javier De Las Rivas

Afiliación de los autores: Centro de Investigación del Cáncer (CSIC/USAL/IBSAL), Salamanca, Spain.

Publicado en *Bioinformatics* (2015)

doi: 10.1093/bioinformatics/btu864 (PMID: 25600944)

Carta de aceptación del artículo 1:

From: Marcelo Brandão

Date: November 14, 2014 1:45:55 PM GMT+01:00

To: Javier De Las Rivas

Subject: BMC Genomics Decision

Dear author,

I am glad to inform that your manuscript entitled "*Analyze multiple disease subtypes and build associated gene networks using genome-wide expression profiles*" was accepted for publication on the BMC Genomics special issue to be published on behalf of the third International Society for Computational Biology Latin America X-Meeting on Bioinformatics with BSB and SoiBio (ISCB-Latin America). The paper has been considered by members of the Editorial Board and peer reviewers who are expert in the field.

Please wait the contact from the BMC central office with the procedures for submission a final version and fee payment.

With my best regards

Marcelo Brandão

Autorización del director de Tesis Doctoral

El Dr. D. Javier DE LAS RIVAS SANZ, con D.N.I. nº 15949000H, Investigador Científico del Consejo Superior de Investigaciones Científicas (CSIC) director del grupo de Bioinformática y Genómica Funcional y profesor del Programa de Doctorado y Máster del Centro de Investigación del Cáncer (CiC-IBMCC) de la Universidad de Salamanca (USAL)

CERTIFICA

que ha dirigido la Tesis Doctoral titulada "Bioinformática aplicada a datos genómicos para la caracterización de subtipos de cáncer: estudios integrativos en hemopatías malignas", presentada en la modalidad de compendio de publicaciones por Dña. Sara AIBAR SANTOS, alumna del programa de *Doctorado en Biociencias: Biología y Clínica del Cáncer y Medicina Traslacional* de la Universidad de Salamanca; y autoriza la presentación de la misma, considerando que reúne las condiciones de originalidad requeridas para optar al grado de Doctor por la Universidad de Salamanca.

En Salamanca, a 9 de abril de 2015

El Director de la Tesis Doctoral,

Dr. Javier De Las Rivas Sanz

Investigador Científico del CSIC

Centro de Investigación del Cáncer (CiC-IBMCC)

Esta Tesis Doctoral ha sido realizada siendo Sara Aibar beneficiaria de una ayuda de la Junta de Castilla y León destinada a financiar la contratación de personal investigador de reciente titulación universitaria, en el marco de la Estrategia Regional de Investigación Científica, Desarrollo Tecnológico e Innovación 2007-2013, cofinanciada por el Fondo Social Europeo.

Índice

Introducción

1. Bioinformática aplicada al estudio de datos ómicos en biomedicina

En los últimos años, el uso de datos "ómicos" (datos a escala global) en biomedicina está creciendo muy rápidamente. Estos datos permiten estudiar las enfermedades desde un punto de vista biomolecular que anteriormente no estaba disponible. Con ello, ofrecen grandes oportunidades para mejorar tanto el entendimiento de la enfermedad, como el desarrollo de nuevos métodos de base molecular para el diagnóstico, pronóstico y tratamiento de los pacientes. Sin embargo, para explorar en detalle la cantidad ingente de datos producidos por las tecnologías ómicas, es indispensable la aplicación de técnicas avanzadas de análisis y cálculo computacional que permitan extraer la información biológica disponible en ellos. En este sentido, estamos hablando de un ámbito de conocimiento e investigación ampliamente interdisciplinar. El objetivo central es estudiar problemas médicos a través de la biología molecular, aplicando métodos estadísticos y computacionales. Concretamente, esta Tesis Doctoral se ubica en el área de "bioinformática aplicada a biomedicina", desarrollando y aplicando técnicas de inteligencia artificial y minería de datos para analizar datos genómicos y transcriptómicos de muestras humanas, con el objetivo es caracterizar subtipos de enfermedades o patologías (*Figura 1*).

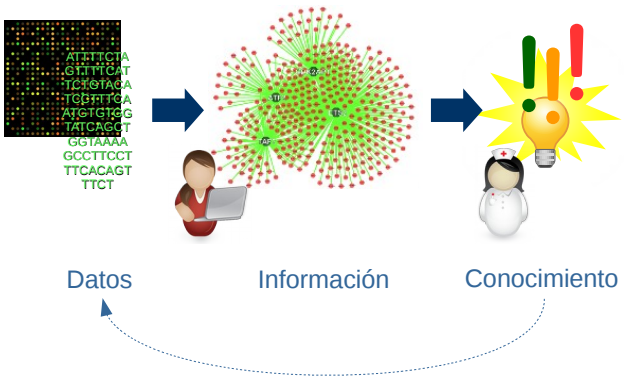


Figura 1. El objetivo general de la bioinformática es manejar, integrar y analizar las cantidades ingentes de datos biológicos y biomoleculares para transformarlos en información estructurada que pueda ser más fácilmente comprensible

En la caracterización molecular de enfermedades complejas, es especialmente interesante identificar genes que estén alterados específicamente en los distintos subtipos o clases patológicas definidos por los médicos para cada enfermedad según sus parámetros clínicos. Estos genes podrían utilizarse como marcadores para el diagnóstico o para entender los procesos subyacentes a la enfermedad.

Los genes en los sistemas biológicos no suelen trabajar como elementos independientes, sino que interactúan unos con otros formando redes de genes que trabajan conjuntamente en funciones biológicas específicas. Actualmente, estas relaciones pueden obtenerse del análisis de datos moleculares experimentales que sean *genome-wide* –es decir, que exploren todo el genoma, como por ejemplo las redes de coexpresión–. También se pueden obtener a partir del análisis integrado del conocimiento previo almacenado en bases de datos biológicas. Para este segundo caso, existen las llamadas herramientas de análisis de enriquecimiento funcional. Habitualmente estos pasos se realizan de manera independientemente. Por ello, en este trabajo de investigación decidimos desarrollar métodos que integren dichos tipos de análisis en un único proceso automatizado, explorando datos derivados de estudios biomoleculares sobre series de pacientes, y con ello, obtener información biológicamente relevante sobre las enfermedades.

Por otro lado, para el desarrollo de nuevos métodos bioinformáticos de análisis de datos derivados de estudios biomédicos, es importante estar en contacto con grupos clínicos que posean datos complejos asociados a problemas y cuestiones concretas que tengan relevancia en el campo. Por ejemplo, una de las principales dificultades en estudios biomédicos es la identificación clara y el análisis diferencial de los distintos subtipos de ciertas enfermedades, sobre todo cuando desde el diagnóstico clínico la separación entre ellos no está clara. Así, a menudo no existen todavía identificados marcadores moleculares propios de cada clase o subclase patológica y, sin embargo, la evolución clínica de los pacientes –en diagnóstico y pronóstico– se conoce que es distinta. Este tipo de problemas plantea marcos analíticos verdaderamente relevantes, sobretodo si se quiere llegar a realizar una verdadera "medicina molecular" capaz de encontrar la etiología y causalidad en las enfermedades.

Bajo estos presupuestos y principios, esta Tesis Doctoral se ha estructurado en cuatro capítulos. Todos ellos con un objetivo común: el desarrollo de metodologías para el análisis de datos ómicos de subtipos de enfermedades, entre ellas el cáncer. Los dos primeros capítulos se presentan a través de dos publicaciones asociadas a dos herramientas bioinformáticas desarrolladas; y los dos últimos capítulos presentan la aplicación y desarrollos realizados para dos estudios concretos sobre muestras de síndromes mielodisplásicos (MDS). En ambos casos, la publicación o informe original está disponible en la sección en inglés, y su resumen/traducción en la sección en castellano:

Capítulo 1 – Herramienta para el análisis de subtipos de enfermedades y construcción de redes usando perfiles de expresión genómica.

Artículo: *Analyse multiple disease subtypes and build associated gene networks using genome-wide expression profiles. BMC Genomics* (2015).

Paquete de Bioconductor asociado: *geNetClassifier: classify diseases and build associated gene networks using gene expression profiles*

Capítulo 2 – Functional Gene Networks (Redes funcionales de genes). Paquete de R/Bioconductor para generar y analizar redes de genes derivadas de análisis y clustering de enriquecimiento funcional

Artículo: *Functional Gene Networks: R/Bioc package to generate and analyse gene networks derived from functional enrichment and clustering. Bioinformatics* (2015)

Paquete de Bioconductor asociado: *FGNet: Functional Gene Networks derived from biological enrichment analyses.*

Capítulo 3 – Análisis combinado de perfiles de metilación y expresión de síndromes mielodisplásicos de bajo riesgo

Capítulo 4 – Integración de perfiles de expresión de distintas plataformas genómicas e identificación de patrones de expresión en la progresión de los síndromes mielodisplásicos hacia la leucemia

Para partir de un contexto común, en las próximas secciones de esta introducción se explican los fundamentos de las distintas disciplinas involucradas: empezando por qué estudiamos y cómo lo medimos (*Secciones 2: Fundamentos de transcriptómica* y *3: Tecnologías y plataformas ómicas*), los métodos que se utilizan para analizar los datos obtenidos (*Sección 4: Fundamentos de análisis bioinformático*) y, finalmente, una breve introducción a las enfermedades específicas que se estudian (*Sección 5: Enfermedades estudiadas: leucemia y síndromes mielodisplásicos*).

2. Fundamentos de transcriptómica

Todos los estudios realizados durante esta Tesis giran en torno al *dogma central de la biología molecular*: el DNA es el mecanismo de almacenamiento y transmisión de la información necesaria para el funcionamiento de los seres vivos. Esta información, codificada en secuencias formadas por cuatro bases, se puede traducir a proteínas a través de la transcripción a RNA (*Figura 2*).

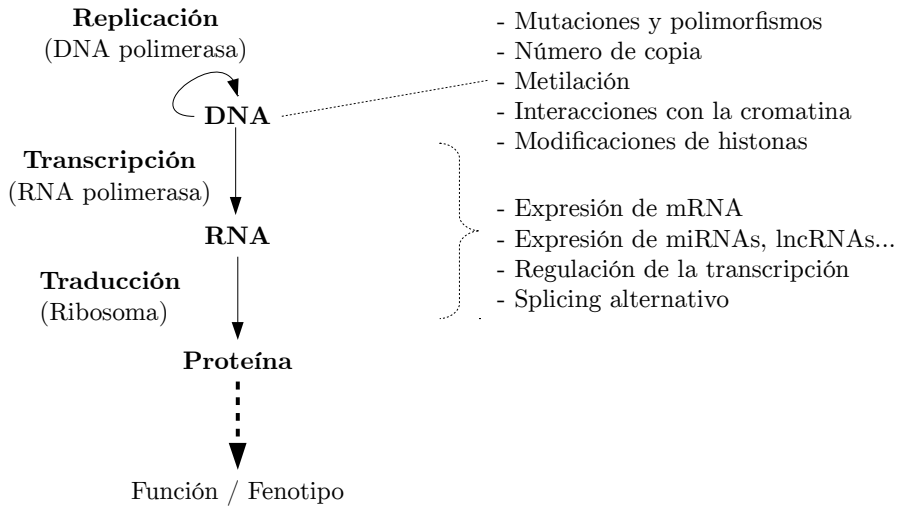


Figura 2. Esquema del dogma central de la biología molecular (*izquierda*) y principales estudios ómicos que se pueden hacer a nivel de DNA y RNA (*derecha*)

Hasta hace relativamente poco, el gen era considerado la unidad básica de herencia a través de la codificación de una proteína. Sin embargo, los avances en el conocimiento del genoma y transcriptoma han hecho que el concepto de gen quede más difuso (*Djebali et al., 2012*). Actualmente, una aceptación más extendida es que el gen es una región genómica que se transcribe a RNA. De esta forma, aunque hay 19.881 genes codificantes de proteína en humano, al incluir otras formas de RNA, el número total asciende a los 60.155 (estadísticas de Gencode de Junio de 2014, *Harrow et al., 2012*).

Puesto que el DNA está en la base molecular de todo sistema vivo, pequeños fallos o variaciones en su secuencia (mutaciones) pueden provocar errores vitales. De hecho, estos errores son el principal origen del cáncer y muchas enfermedades. En las enfermedades hereditarias, las mutaciones dañinas se producen en los gametos o ya están incluidas en el genoma del individuo, y por tanto pueden pasar a sus descendientes. También existen mutaciones heredadas no dañinas que cuando se extienden a un grupo suficientemente grande de la población se consideran polimorfismos. Sin embargo, el origen del cáncer está en las mutaciones que se producen en las células somáticas (las células que forman el organismo, excluyendo las células germinales y los gametos). La mayoría de las variaciones dañinas matan a la célula, pero a veces estas modificaciones pueden suponer un pequeño cambio adaptativo beneficioso para la proliferación de la célula. Aunque estas mutaciones no se transmitirán a los descendientes del individuo, sí que las heredarán las células descendientes de la célula modificada. A pesar de que cada variación individual no afecte mucho, las mutaciones se van acumulando generación tras generación. Por ello, tras muchas generaciones de células, se pueden haber ido acumulando características que acaben dando a esa población de células comportamientos nuevos. Por ejemplo, que se reproduzcan más de lo normal, que no mueran cuando les corresponda, o que ignoren las señales del resto del organismo. Cuando se juntan todos estos comportamientos en una misma población de células, el organismo acaba perdiendo el control sobre ellas dando lugar

a un cáncer (*Hanahan and Weinberg, 2000*). Para evitarlo, los organismos tienen muchos mecanismos de control de calidad del DNA y complejos sistemas de regulación interna.

Uno de los principales mecanismos de protección del DNA en las células eucariotas es tenerlo plegado adecuadamente. Para ello, el DNA suele estar asociado a histonas, unas pequeñas moléculas alrededor de las cuales se enrolla, formando una cadena que recuerda a un collar de perlas. Esta estructura, llamada cromatina, es el estado en el que se encuentra el DNA habitualmente en la célula; evitando que el DNA se enrede o dañe, pero lo suficientemente accesible para poder usarlo. Dependiendo del estado de la célula y de si una determinada zona del DNA se va a utilizar, la compactación de la cromatina puede ir variando.

A pesar de que todas las células de nuestro cuerpo parten de la misma secuencia del genoma (constituido por el compendio de moléculas de DNA propio de cada especie), no todas las células son iguales. Una célula de la piel es muy distinta a una de un hueso. Esto es debido a que no todas las células utilizan todo el DNA que tienen disponible. Normalmente se considera que sólo están activos aquellos genes que se están transcribiendo a mRNA para producir proteínas o RNAs funcionales (expresión). La cantidad de mRNA no tiene por qué correlacionar exactamente con la cantidad de proteína. Sin embargo, los mRNA presentes en una célula o población celular pueden ser un buen indicativo de los procesos que se están llevando a cabo en ellas. Por ello, el estudio de los niveles de expresión permite analizar las diferencias entre los distintos tipos celulares, la reacción de las células ante distintas circunstancias o distintas señales externas o cómo están afectadas en una determinada enfermedad.

Está estimado que sólo una parte de los genes codificantes de proteína se expresan en todas las lineas celulares (*Djebali et al., 2012*), y los que lo hacen suelen tener entre 1 y 10 copias de su mRNA por célula (*Marguerat et al., 2012*). Por ello, los mecanismos de *regulación transcripcional* son importantes ya que se encargan de controlar qué zonas del DNA se transcriben y cuándo lo hacen. Los principales mecanismos de regulación transcripcional son los factores de transcripción y los mecanismos epigenéticos.

Los factores de transcripción (TFs) son proteínas que tienen afinidad por una secuencia específica de DNA a la que tienden a unirse. Estas secuencias, llamadas en inglés *transcription factor binding sites* (TBFS) suelen estar en los promotores de los genes (zonas reguladoras normalmente ubicadas en las regiones previas al punto de inicio de la transcripción). A grandes rasgos, cuando un TF activador se une a ellas, favorece que la RNA polimerasa inicie la transcripción del gen. Por otro lado, si es un TF inhibidor, evitará que el gen se pueda transcribir. Esto puede hacerlo a través de múltiples mecanismos, por ejemplo evitando que el complejo de la RNA polimerasa pueda actuar, actuando sobre la estructura de la cromatina o evitando que un TF activador se una a esa secuencia.

A día de hoy, la mayoría de los TFs humanos están identificados (1391 según *Vaquerizas et al., 2009*). Sin embargo, sus puntos de unión al DNA todavía son objeto de estudio. Se estima que la mayoría de los factores de transcripción tienen varios miles de TFBS a lo largo del genoma, pero sólo una parte de ellos está localizado en zonas asociadas a genes (*Cawley et al., 2004*). Además, su accesibilidad depende del tipo y estado celular. Por ello, aunque hay métodos bioinformáticos que identifican TFBS en los promotores de los genes, para demostrar los puntos de unión de un TF al DNA genómico en un tipo celular y en determinadas condiciones suele ser necesario el uso de tecnologías ómicas.

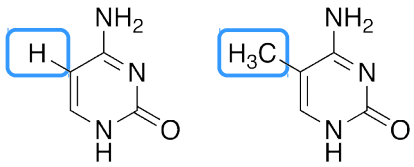
Se consideran mecanismos de regulación epigenética aquellos cambios en el DNA que, sin cambiar la secuencia, son heredados en la división por mitosis. Estas modificaciones incluyen principalmente la metilación del DNA y las modificaciones en las histonas (acetilación, fosforilación y metilacion de ciertos aminoácidos), y suelen ser necesarias para mantener la identidad de cada tipo celular(*Lokk et al., 2014*).

La metilación del DNA consiste en añadir un grupo metilo a una base citosina, convirtiéndola en 5-metilcitosina (*Figura 3*). Esta reacción sólo se produce en citosinas ubicadas justo antes de una guanina (CpG). La distribución de las CpG en el genoma no es aleatoria. En los mamíferos hay zonas con gran densidad de CpG (llamadas islas CpG) ubicadas principalmente cerca de los promotores de los genes. Las citosinas en las islas CpG no suelen

estar metiladas. Sin embargo, cuando están metiladas, normalmente van asociadas con la inhibición del gen cercano. Por el contrario, las CpG aisladas sí suelen estar metiladas, y se asocian con el mantenimiento de la estabilidad cromosómica.

Una de las principales características de la metilación del DNA es que a pesar de ser una marca heredable, hay muchos factores que la pueden modificar. Desde la nutrición, hasta la edad. De hecho, es posible estimar la edad de un individuo a través de los patrones de metilación en sus células (*Weidner et al., 2014*).

En cáncer la metilación también está alterada. Se ha detectado que las islas CpG en los promotores de muchos genes suelen estar más metiladas que en las células normales. Esto puede estar asociado con la pérdida de función de genes supresores tumorales y genes asociados con funciones clave como la reparación del DNA o apoptosis. Por otro lado, los niveles de metilación a nivel global del DNA son menores de lo normal, especialmente en zonas con muchos elementos repetitivos, retrotransposones e introns (*Sandoval and Esteller, 2012*). Puesto que la metilación en estas zonas normalmente se asocia con el mantenimiento de estabilidad del genoma, su reducción (combinado con otras alteraciones en la cromatina) parece ser la causa de que en las células cancerígenas sea muy habitual encontrar cariotipos aberrantes. Entre ellas, las translocaciones (cambio de posición de un trozo de cromosoma) y duplicaciones o delecciones tanto de fragmentos como de cromosomas enteros.



Citosina *5-metilcitosina*

Figura 3. La metilación de la citosina se produce añadiendo un grupo metilo (CH_3) al carbono en 5ª posición

Además de los mecanismos de regulación transcripcional, las células cuentan con otros muchos sistemas de regulación. Por ejemplo, las vías de señalización o los microRNA (miRNA), que se unen al RNA mensajero para evitar que se traduzca a proteína. La mayoría de los genes frecuentemente mutados en cáncer (oncogenes y supresores tumorales) son precisamente genes que, en condiciones normales, codifican proteínas con funciones de regulación.

3. Tecnologías y plataformas ómicas

El término *ómico* se refiere al estudio global de los sistemas celulares en un nivel concreto. De esta forma salen los términos *genómica* (que estudia los datos relativos al DNA), *transcriptómica* (RNA), y *proteómica* (proteínas). En este mismo ámbito también se usan las denominaciones *genoma* (que incluye todo el material genético del organismo), *transcriptoma* (el conjunto de todas las moléculas de RNA de una célula o grupo de células), *proteoma* y las denominaciones equivalentes a otros niveles (*metiloma*, *metaboloma*...).

En las últimas décadas se han mejorado enormemente las tecnologías que permiten recopilar información a gran escala en cualquiera de estos niveles. Actualmente, las tecnologías ómicas permiten leer secuencias y medir la cantidad de moléculas de DNA o mRNA de miles de genes simultáneamente. Permitiendo estudiar, por ejemplo, el estado del transcriptoma completo de una población celular en tiempos y costes muy asequibles.

Hay principalmente dos grandes familias de tecnologías ómicas: las basadas en *hibridación* por complementariedad con secuencias de nucleótidos de referencia (microarrays) y las basadas en *secuenciación* (identificación del orden de las bases en fragmentos de DNA, RNA u otro ácido nucléico). Aunque el primer genoma completo se secuenció en 1977 (*Sanger et al., 1977*), hasta

hace relativamente poco, las tecnologías de secuenciación eran extremadamente lentas y caras como para aplicarlas habitualmente a escala global. Las tecnologías basadas en hibridación también existían desde hace tiempo, pero no fue hasta 1995 cuando se publicó el primer estudio utilizando microarrays (*Schena et al., 1995*). Sin embargo, a partir de entonces, los microarrays se han perfeccionado rápidamente, permitiendo su uso generalizado y creando una auténtica revolución en la investigación en biología molecular.

El objetivo de los microarrays es cuantificar la cantidad de mRNA o DNA de cada gen o "característica" en la muestra. Inicialmente se aplicaban principalmente para detectar diferencias en las cantidades de mRNA entre distintos estados biológicos. Sin embargo, actualmente hay plataformas explícitamente diseñadas para analizar polimorfismos, metilación, splicing alternativo, etc. Su principal ventaja es que son rápidos, sencillos y relativamente baratos. Además, al ser plataformas reproducibles y muy usadas, hay muchísimas herramientas disponibles para el análisis de datos de microarrays y no requieren tanta potencia de computación como las plataformas de secuenciación. Por otro lado, su principal desventaja es que son plataformas cerradas. No es posible tomar medidas de las secuencias que no estén incluidas en el array, y por lo tanto, no se pueden utilizar para organismos con genomas no conocidos.

Por el contrario, las tecnologías basadas en secuenciación no requieren conocimiento previo del genoma, y de hecho se utilizan frecuentemente para secuenciar *de novo* el genoma de nuevos organismos. Esto también es una ventaja en estudios transcriptómicos, ya que permiten detectar mutaciones y estudiar tránscritos o isoformas desconocidos. Además, las medidas que proveen son valores absolutos que estiman la concentración de nucleótidos, puesto que se basan en secuenciar cada uno de los fragmentos de DNA o RNA y anotar el número de veces que aparece cada secuencia. Sin embargo, el preprocesamiento de los datos de secuenciación todavía es complejo y requiere gran potencia de cálculo. Esto, sumado a su precio más elevado, hace que los microarrays sean una alternativa a considerar, especialmente en estudios con muchas muestras.

En la sección 3.2 se da una visión global de cómo se aplican estas tecnologías en estudios ómicos específicos, pero primero es necesario entender en qué consisten los microarrays (sección 3.1).

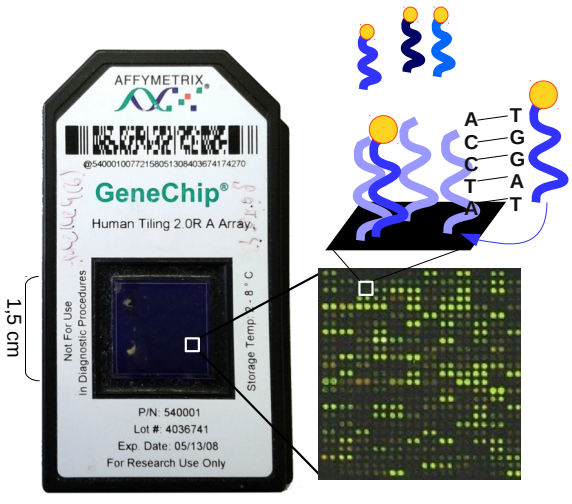


Figura 4. El microarray está formado por miles de celdillas con cadenas de DNA específicas (sondas). Al depositar en él los fragmentos de la muestra a analizar, estos hibridan con las sondas de secuencia complementaria. Como los fragmentos de la muestra están marcados con moléculas fluorescentes, al escanear el microarray la intensidad de la fluorescencia representa el valor de hibridación de cada sonda.

3.1. Fundamentos de los microarrays: hibridación

Muchas de las técnicas de estudio del DNA a nivel molecular se basan en aprovechar su principal propiedad: que dos cadenas complementarias tienden a unirse (hibridación). La unión de este tipo de técnicas con los avances tecnológicos que permitieron miniaturizarlas es lo que dio lugar a los microarrays.

Cada microarray o *chip* es una placa sobre la que hay unidos millones de fragmentos de DNA de unas pocas decenas de bases (oligonucleótidos). Su estructura está organizada en forma tabular, de modo que se puede saber las coordenadas de cada una de las casillas. En cada casilla hay pegados oligonucleótidos con una secuencia específica (sondas). Puesto que estas secuencias se diseñan complementarias a las que se quieren estudiar, al exponerlos a fragmentos del DNA o RNA de la muestra a analizar, las cadenas complementarias hibridarán. Para poder medir qué sondas han hibridado, el cDNA o RNA de la muestra se marca con una molécula fluorescente que es detectada por un escáner. De esta forma, para cada secuencia se obtiene un valor de fluorescencia que representa la cantidad de DNA/RNA con esa secuencia que hay en la muestra estudiada (*Figura 4*).

Tipos de microarrays

Dependiendo de las características que se tengan en cuenta, los microarrays se pueden clasificar de muchas formas. Aquí veremos principalmente dos clasificaciones: la basada en el número de muestras por array (ya que afecta a la interpretación de los resultados) y la basada en el método de fabricación.

Dependiendo de cómo se hibridan las muestras, hay dos tipos de microarray: microarrays de un canal o un color (en los que se pone una única muestra por array) y microarrays de dos colores o dos canales (en los que se ponen dos muestras etiquetadas con distintas moléculas fluorescentes).

En los microarrays de dos canales el objetivo es comparar la cantidad relativa en las dos muestras estudiadas. Por ejemplo, una muestra antes del tratamiento y otra después, una muestra normal y una de tumor, etc. El valor resultante para cada sonda suele ser un valor relativo –normalmente llamado valor relativo de cambio o *fold change*– que representa en cuál de las dos muestras hay más cantidad de la secuencia dada. Su principal ventaja es que dan un resultado claro sin necesidad de muchas muestras. Sin embargo, resulta difícil incluir más controles o hacer otras comparaciones posteriores, por lo que sólo son recomendables en estudios en los que haya un claro punto de referencia (p.ej. muestra de tejido sana y una tumoral de un mismo paciente).

Por otro lado, los microarrays de un canal siguen un enfoque distinto: medir o estimar la cantidad absoluta total en cada muestra para tener más libertad para realizar distintas comparaciones entre ellas. De este modo, el valor resultante se aproxima a una medida absoluta de la concentración de señal en las muestras. Sin embargo, para realizar la comparación entre múltiples arrays son necesarias normalizaciones robustas y por ello normalmente también preprocesamientos más complejos (*Sección 4.1*).

En esta Tesis se utilizan ambos tipos de microarrays. Los arrays de expresión –utilizados en casi todos los capítulos– son de un canal, mientras que los de metilación, utilizados en el Capítulo 3, son de dos canales.

En cuanto al tipo de fabricación de los microarrays, también hay dos tipos principales: los *spotted microarrays* y los *sintetizados in situ*.

Los *spotted microarrays* se producen con un robot que deposita las secuencias previamente sintetizadas sobre una placa de cristal. Suelen tener menos reproducibilidad y ser de dos canales. Esta tecnología es una de las primeras en aparecer y ha ido cayendo en desuso. Sin embargo, tiene la ventaja de que es más flexible (se pueden hacer arrays personalizados más fácilmente) y pueden tener secuencias de nucleótidos más largas.

Los microarrays *sintetizados in situ* son arrays producidos sintetizando los oligonucleótidos directamente sobre el soporte. Aunque hoy en día hay distintas tecnologías para producirlos, la inicial desarrollada por la compañía *Affymetrix* se basa en fotolitografía, el sistema utilizado para crear circuitos integrados. Esta técnica consiste en ir añadiendo los nucleótidos uno a uno, utilizando máscaras que tapan/destapan las ubicaciones de cada sonda para que el nucleótido sólo se deposite en aquellas que corresponde. Estos arrays son mucho más precisos y permiten mayor densidad de sondas. Sin embargo la longitud de la sonda está más limitada (25-100 bases, según la tecnología), y en el caso de querer un array personalizado habrá que contactar con la compañía productora.

Una curiosidad sobre la visualización de resultados de expresión en estudios ómicos, es que normalmente se suele asignar verde a los genes infraexpresados y rojo a los sobreexpresados. Esto suele resultar muy antiintuitivo al principio (normalmente estamos acostumbrados a asignar rojo a los números negativos). Sin embargo, tiene sentido si tenemos en cuenta que los primeros microarrays eran de dos colores, y normalmente se etiqueta con Cy3 (fluorescencia verde) la muestra de referencia (sana/antes del tratamiento...) y con Cy5 (fluorescencia roja) la objeto del estudio (p. ej. muestra tumoral/después del tratamiento). Este etiquetado produce que cuando una sonda está más expresada en la muestra de referencia, la fluorescencia sea principalmente verde, y cuando está más expresada en la del tratamiento, roja. Aunque hoy en día se utilizan también otras tecnologías ómicas, se sigue utilizando este código de color.

Protocolo en estudios con microarrays

Aunque el procedimiento específico utilizado en un estudio de microarrays depende de muchos factores, en general todos los estudios con microarrays siguen unos pasos genéricos comunes:

1. Extraer y purificar el RNA o DNA
2. Fraccionar y etiquetar del DNA o RNA para poder detectarlo con el escáner. Este paso varía según la tecnología y ácido nucleico. En general consiste en sintetizar y amplificar en cDNA (DNA copia) o aRNA (antisense RNA). Durante este proceso se aprovecha para añadir nucleótidos etiquetados con las moléculas fluorescentes (en arrays de dos colores) o con biotina (en microarrays de expresión de Affymetrix). Las cadenas de DNA o RNA también son fragmentadas para coincidir con la longitud de los oligos.
3. Depositar el cDNA o aRNA de la muestra sobre el microarray –manteniendo las condiciones adecuadas– para que hibride con las sondas complementarias. En caso de los microarrays de dos colores, primero se mezclan las dos muestras.
4. Lavar el microarray para quitar los restos de muestra que no haya hibridado con las sondas del microarray.
5. Escanear el microarray. Para ello se utiliza un láser que excite las moléculas utilizadas para etiquetar y se mide la señal de fluorescencia emitida por cada sonda. En el caso de los microarrays de dos colores se escanean por separado (utilizando el láser con longitud de onda adecuada para cada molécula de etiquetado) y se fusionan posteriormente.
6. Preprocesamiento y análisis de los datos (*Sección 4*)

3.2. Aplicaciones de las tecnologías ómicas

Los estudios ómicos a casi todos los niveles se pueden realizar tanto con métodos basados en hibridación, como con métodos basados en secuenciación. Las principales excepciones son la secuenciación del genoma completo y detección de mutaciones, en las que obviamente sólo se puede usar la secuenciación.

- **Transcriptómica**: La medida de la cantidad de mRNA en la célula es una de las principales aplicaciones tanto de los microarrays como de la secuenciación (RNA-Seq). El RNA-Seq provee sin duda más flexibilidad y cobertura que los microarrays. No sólo

no está limitado por el conocimiento del genoma del organismo, sino que permite detectar mutaciones, polimorfismos, splicing alternativo y todo tipo de tránscritos. Además es muy reproducible y resulta más fácil comparar muestras entre distintos estudios. También tiene un mayor rango dinámico de detección que los microarrays, por lo que requiere menor cantidad de RNA. Puesto que puede detectar desde unas pocas moléculas de mRNA, es posible incluso hacer RNA-Seq con una única célula (*Patel et al., 2014*).

A pesar de todas las ventajas del RNA-Seq, los microarrays siguen siendo la tecnología más utilizada. Probablemente porque la información extra no siempre es necesaria, por su precio y por mayor su facilidad de uso y análisis. Además, siguen apareciendo nuevas versiones de microarrays que proveen más resolución que las versiones anteriores, centrándose las últimas versiones en la detección de más tipos de RNA y en información detallada de *splicing* e isoformas.

- **Genotipado (detección de polimorfismos en poblaciones) y detección de número de copia:** Aunque el método más natural para detectar polimorfismos del DNA parecería ser la secuenciación, también hay microarrays para detectar *single nucleotide polimorfisms* (SNP). Estos arrays utilizan sondas con las distintas secuencias alternativas para detectar cuáles de ellas están presentes en la muestra. Este tipo de microarrays son una alternativa más asequible que el RNA-Seq para los estudios de asociación del genoma completo (GWAS). En estos estudios se buscan SNPs asociados con determinados fenotipos, por lo que suelen hacer falta cientos de muestras para poder llegar a resultados significativos.

  Puesto que los arrays de SNPs son plataformas que detectan DNA, se vió que también se podían usar para cuantificar variaciones en número de copia. Esto provocó que actualmente haya microarrays comerciales preparados para detectar tanto polimorfismos como variaciones en el número de copia.

- **Interacción de moléculas con la cromatina (factores de transcripción, histonas, etc):** Las interacciones con el DNA y sus modificaciones a nivel genómico también se pueden estudiar tanto con microarrays como con secuenciación. Las interacciones más estudiadas suelen ser con factores de transcripción o proteínas relacionadas con la replicación (ORC), pero también se pueden utilizar estas técnicas para estudiar otros aspectos como las modificaciones de histonas. Para ello, una de las técnicas más utilizadas es la inmunoprecipitación de la cromatina (ChIP), una técnica que utiliza anticuerpos para aislar las secuencias de DNA unidas a determinadas proteínas. Los fragmentos de DNA obtenidos por esta técnica se pueden hibridar en microarrays (ChIP-on-Chip) o secuenciar (ChIP-Seq) para localizar su ubicación en el genoma. De una forma similar, las técnicas de análisis de la conformación cromosómica y sus derivados (3C, 4C, Hi-C) permiten estudiar la organización espacial de los cromosomas en el núcleo midiendo la interacción entre distintas zonas de la cromatina.

- **Metilación:** Para estudiar la metilación de DNA se utiliza un enfoque parecido al de interacciones de moléculas con el DNA: primero se aísla el DNA con alguna técnica que permita distinguir entre las citosinas metiladas y no metiladas, y posteriormente se secuencia el DNA aislado o se hibrida en un microarray (*Laird, 2010*). El método más utilizado para capturar la metilación es el tratamiento con bisulfito de sodio. Este tratamiento convierte las citosinas no metiladas en timinas, manteniendo las metiladas sin convertir. Otros enfoques se basan en inmunoprecipitación o en digestión del DNA con encimas sensibles a la metilación. Esta última técnica es la utilizada en el Capítulo 3 y se explica allí más detalladamente.

Durante la última década, estas tecnologías ómicas han producido cantidades enormes de datos. Ahora el reto principal está en la integración de los datos de distintos niveles para poder llegar a entender la interrelación entre ellos. Entender cómo las alteraciones en un nivel afectarán a los otros, y cómo van cambiando estas relaciones en función de las circunstancias o a lo largo de la vida celular (*Jones et al., 2013*). Los estudios integrativos también tienen un gran

potencial en la investigación del cáncer, ayudando a entender cómo y qué funciones están desreguladas (*Kristensen et al., 2014*).

4. Fundamentos de análisis bioinformático

4.1. Preprocesamiento de datos

En cualquier estudio con tecnologías ómicas, la obtención de la medida cruda es sólo uno de los pasos iniciales. Los datos obtenidos directamente del escáner o de la máquina de secuenciación tienen que ser preprocesados, normalizados y convertidos en valores representativos de los parámetros biomoleculares que se están midiendo. Por ejemplo, un valor que represente la concentración de un tránscrito en una muestra. Una vez los datos hayan sido preprocesados, estarán listos para analizar y responder las preguntas del estudio.

La elección de los métodos de preprocesamiento y análisis dependen tanto de la plataforma como del tipo de datos. Por ejemplo, el método de normalización por cuantiles –ampliamente utilizado para datos de microarrays de expresión– no se debe utilizar para datos de metilación porque la cantidad global de metilación puede variar entre muestras (*Siegmund, 2012*). Sin embargo, sí que hay algunos pasos genéricos por los que deberán pasar los datos procedentes de un mismo tipo de tecnología.

En las plataformas basadas en **secuenciación**, los datos obtenidos corresponden al número de fragmentos de secuencia (*reads*). Estos fragmentos tienen que alinearse con el genoma de referencia para localizar su ubicación. Puesto que en el genoma hay muchas zonas con repeticiones (genes duplicados, elementos repetitivos, etc), es posible que un fragmento coincida con más de una ubicación, o también que no se logre alinear. Por otro lado, en el caso de que todavía no haya un genoma de referencia para el organismo del estudio, entonces habría que ensamblar el genoma/transcriptoma *de novo*. Para ello, suele ser recomendable tener fragmentos largos y solapantes (mayor *sequencing depth*).

Si la secuenciación se esta utilizando para algún estudio de cuantificación (p. ej. RNA-Seq o ChIP-Seq), el número de fragmentos anotados a cada ubicación determinará la intensidad de la medida. Para obtener el valor definitivo, habrá que utilizar el enfoque adecuado según la tecnología. Por ejemplo, en ChIP-Seq se utilizan programas de *peak calling,* para determinar el valor de los picos, y en RNA-Seq de sumarización, para agrupar los fragmentos que corresponden a un mismo gen. Además, en el caso de RNA-Seq, para poder comparar distintas muestras, también es necesario normalizar los posibles sesgos producidos por la variación entre muestras (p. ej. si han sido obtenidas con distinta profundidad de secuenciación) y por las diferencias en la longitud de los tránscritos (sesgo intra-muestra). Aunque la medida final depende del programa utilizado, una de las más estándar es la propuesta por *cufflinks (Trapnell et al., 2011*): los RPKM o FPKM (lecturas o fragmentos por kilobase de exón por millón de lecturas mapeadas) que normalizan los valores por el número total de lecturas en la muestra y por la longitud del gen.

En el caso de los **microarrays**, los datos crudos son los valores de fluorescencia en cada celda, que a su vez representan el valor de hibridación de una sonda. Sin embargo, aparte de la señal de fluorescencia propia de la hibridación de las sondas, también suele haber una ligera fluorescencia remanente en todo el array. Esta fluorescencia puede ser producida por las sondas que han hibridado a pesar de no ser perfectamente complementarias, y por lo tanto hay que corregirla. Este paso se denomina corrección de la señal de fondo (*background correction*).

Posteriormente, hay que calcular la señal correspondiente a cada sonda. En el caso de los arrays de dos colores, esto se hace calculando la señal relativa entre los colores (por ejemplo el *fold change*). En el caso de los microarrays de un canal, hay que calcular un valor de señal para cada conjunto de sondas que represente un gen o unidad de medida (sumarización). La agrupación de las sondas se suele definir en el fichero de descripción del chip (CDF, *chip definition file*) que indica qué conjunto de sondas (*probe-set*) corresponden a un mismo gen. Sin embargo, el valor obtenido para cada sonda no se puede utilizar directamente. Hay muchos factores que pueden haber alterado la medida: desde la cantidad de muestra con la que se ha

hibridado el array, manipulación de la muestra, efectos del escaneado, ubicación de la sonda en el array, proporción de nucleótidos GC en la sonda, etc. Puesto que muchos de estos efectos son sistemáticos, se pueden compensar con un proceso de normalización dentro de cada array y entre arrays para pasar todos los datos a escalas comparables. La mayoría de las plataformas de microarrays proveen métodos para realizar estos pasos. Sin embargo, *Robust Multi-array Average* (RMA) (*Irizarry et al., 2003b*) es un método no paramétrico muy robusto de gran eficacia y reproducibilidad (*Millenaar et al., 2006*), cuyo uso está tan generalizado que se ha convertido prácticamente en un estándar *de facto* para los microarrays de expresión de *Affymetrix*.

Una de las novedades que introdujo *RMA*, fue el no utilizar las sondas de *mismatch* (MM) para la corrección de la señal de fondo. Los arrays de *Affymetrix* incluyen tanto las sondas para medir las secuencias de interés (*perfect match* PM), como sondas *mismatch* (diseñadas como controles) en las que se ha cambiado la base central. El objetivo de las sondas *mismatch* es obtener información sobre la capacidad de hibridación inespecífica. Los métodos existentes antes de que apareciese *RMA* calculaban la señal "real" usando la diferencia (PM-MM) o la proporción entre ellas (PM/MM). Sin embargo *Irizarry et al.* demostraron que realmente las sondas de *mismatch* también capturan señal, no sólo hibridación inespecífica. Por lo tanto, al tenerlas en cuenta, atenúan la señal real. Por ello, *RMA* ignora estas sondas y calcula la señal de fondo como una distribución a nivel global del array. Una vez se ha descontado la señal de fondo de la señal observada, se tiene el valor de cada sonda. *RMA* usa entonces la normalización por cuantiles para hacer que la distribución de la intensidad de todos los arrays sea la misma (*Figura 5*), y el logaritmo para facilitar el uso de métodos paramétricos. Finalmente, sumariza los valores normalizados utilizando un modelo lineal que asume que el *probe effect* –el hecho de que debido a su secuencia algunas sondas sean más afines a la hibridación que otras– es constante para cada sonda entre todos los arrays. Tras pasar por este proceso, se obtiene un valor de expresión para cada gen o probeset en cada microarray. Este valor permite comparar la señal de los probesets o genes entre los microarrays que han sido normalizado juntos. Habitualmente no deben compararse arrays normalizados independientemente. Juntar arrays procedentes de distinto laboratorio, estudio o tratados con distinto protocolo tampoco está recomendado: los efectos sistemáticos introducidos sobre los arrays serán distintos y la normalización no será capaz de compensarlos completamente, dando lugar al *efecto batch*. Sin embargo, en los últimos años se han desarrollado interesantes métodos bioinformáticos destinados a normalizar de modo conjunto múltiples estudios de distinto origen para lograr realizar meta-análisis integrativos.

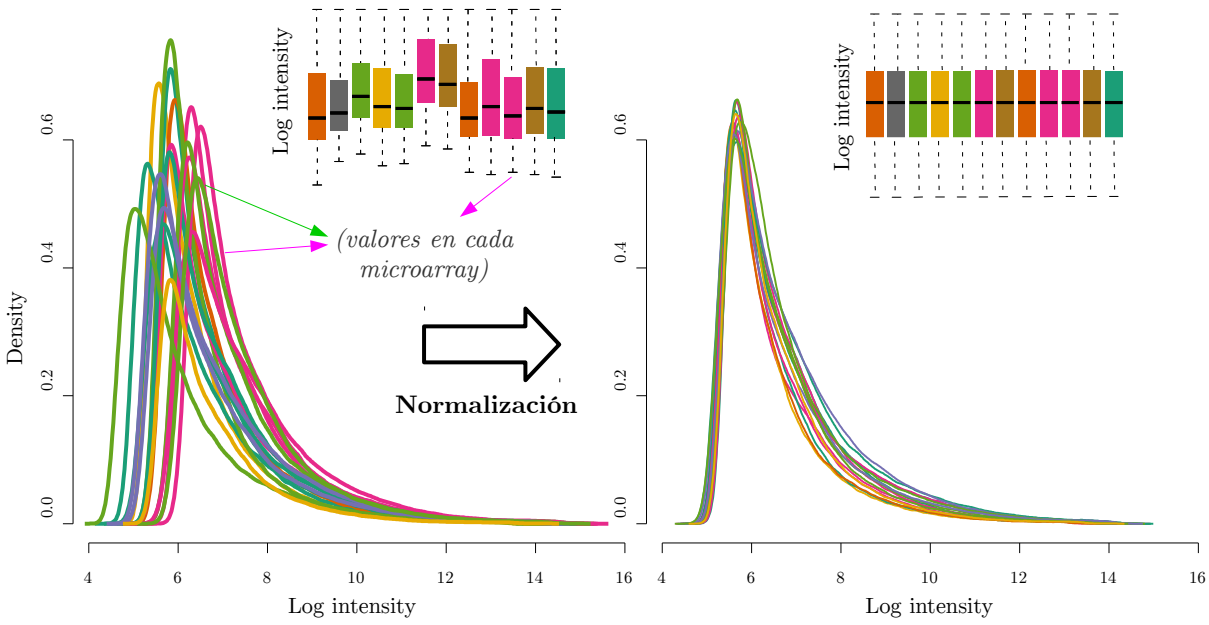


Figura 5. Efecto de la normalización. Cada linea/boxplot representa la distribución de valores de intensidad en cada microarray: antes (*izquierda*) y después de normalizar (*derecha*). Los boxplots ilustran el efecto de la normalización con otro tipo de visualización.

4.2. Clasificadores (predicción de clases) y clustering (descubrimiento de clases)

Para el análisis de datos ómicos se utilizan muchos métodos de minería de datos y de búsqueda de patrones en grandes volúmenes de datos. Muchos de estos métodos están basados en técnicas de aprendizaje automático, como los clasificadores y las técnicas de clustering y reducción de dimensionalidad.

El aprendizaje automático es la rama de la inteligencia artificial orientada a diseñar algoritmos que permitan a los ordenadores "aprender" de los datos. Dependiendo del conocimiento previamente existente sobre los datos de partida, el aprendizaje automático se divide en dos grandes ramas: el aprendizaje supervisado y el no supervisado.

Las técnicas de aprendizaje **supervisado** se basan en proporcionar al programa ejemplos de datos ya conocidos para que aprenda a identificarlos. El ejemplo de aprendizaje supervisado más conocido es el de los clasificadores. Un clasificador es un programa que se utiliza con el objetivo de identificar a qué tipo pertenece una muestra de tipo desconocido. Por ejemplo, determinar si la muestra es tumoral o sana. Para ello, primero es necesario pasar por un proceso de entrenamiento. El entrenamiento consiste en proporcionarle unas cuantas muestras de ejemplo de cada una de las clases a estudiar con sus correspondientes etiquetas. El clasificador las analizará para determinar las características comunes dentro de cada tipo y las que las diferencian del resto. Una vez el clasificador está entrenado, se le pueden proporcionar nuevas muestras para que determine de que tipo son.

Un clasificador se podría utilizar tanto para diagnóstico (identificación de enfermedad o subtipo de enfermedad) como para pronóstico (predisposición a una enfermedad, previsión de evolución). Simplemente habría que proporcionarle las muestras de entrenamiento etiquetadas adecuadamente (por ejemplo, divididas en función del desarrollo que obtuvieron).

Los clasificadores en un principio se pueden aplicar a muchos tipos de datos. Entre ellos datos ómicos, por ejemplo de expresión. Un ejemplo muy sencillo de clasificador basado en expresión sería un caso con dos categorías (estados, subtipos, ...) en el que un gen siempre esté expresado en una de ellas pero nunca en la otra. De esta forma, simplemente con saber el valor de expresión del gen, se sabría el tipo de muestra que es. Obviamente, en este caso no sería necesario un clasificador, pero este tampoco es un caso muy habitual. Normalmente es necesario usar una combinación de genes que se utilizarán como variables para ajustar un modelo estadístico. El principal problema de aplicar los clasificadores a datos genómicos es que los enfoques estadísticos tradicionales normalmente están pensados para tener los valores de muchas muestras en unas pocas variables. Puesto que los datos genómicos son el caso contrario (muchas variables pero pocas muestras), no se pueden usar los datos completos, sino que hay que hacer algún proceso de preselección o filtrado previo.

Por otro lado, el aprendizaje **no supervisado** es capaz de estudiar los datos sin necesidad de disponer de un conocimiento previo. Las técnicas no supervisadas incluyen enfoques de reducción de dimensionalidad, búsqueda de patrones (p.ej. *hidden-markov-models*), o clustering. Concretamente, la técnica más utilizada en esta Tesis es el clustering, que consiste en el agrupamiento de los datos disponibles. Las técnicas de clustering se utilizan habitualmente tanto para agrupar datos, como para buscar patrones o descubrir subtipos entre datos aparentemente homogéneos o demasiado complejos. En biomedicina es una técnica muy utilizada, que incluso ha servido para descubrir nuevos subtipos de cáncer (*Golub et al., 1999*).

4.3. Análisis de enriquecimiento funcional

El resultado de muchos estudios ómicos son listas de genes asociadas con una determinada condición (p. ej. genes mutados, diferencialmente expresados, metilados, o reguladas por un determinado factor de transcripción). Para identificar las posibles implicaciones asociadas a estos genes, hay que realizar algún tipo de análisis funcional, es decir, identificar los procesos biológicos en los que los genes están involucrados. De esta forma, no sólo se mejora la comprensión de la lista de genes obtenida, sino que además, –si muchos de ellos están colaborando en un mismo proceso– se podría reducir la complejidad de los resultados (pasando de decenas, cientos o incluso miles de genes individuales, a sólo unas pocas decenas de funciones o procesos biológicos).

Con el aumento del conocimiento disponible sobre los distintos genes, es indispensable recopilarlo y ordenarlo en bases de datos. Para ello, existen múltiples recursos bioinformáticos que ponen a nuestra disposición prácticamente todo tipo de información. Desde la conocida sobre las vías celulares o metabólicas (KEGG, Biocarta, Reactome, ...), hasta la posible asociación de genes con ciertas enfermedades (OMIM), pasando por todo el conocimiento relativo a las proteínas y su clasificación: dominios y estructura (Interpro), secuencia y función (UniProt) e interacciones entre proteínas (BIND, MINT). Con el objetivo de estandarizar la terminología utilizada en distintas bases de datos y organismos, se creó la ontología de genes (*Gene Ontology*: GO). Esta base de datos contiene información sobre los genes en forma de anotación a términos biológicos. La peculiaridad es que estos términos están organizados en una jerarquía que indica cuáles están incluidos dentro de otros. De este modo, sabemos que si un gen esta anotado a "*positive regulation of apoptosis*", también está involucrado en el concepto de regulación de apoptosis más global. GO está formado por tres jerarquías independientes: una para los procesos biológicos en los que el gen o producto génico está involucrado (*biological process:* BP), otra para sus funciones moleculares (*molecular function*: MF) y otra para la ubicación del producto génico en la célula (*cellular component*: CC). Hay que tener en cuenta que, en muchas de estas bases de datos, todavía hay muchas anotaciones "predichas" o "inferidas de anotación electrónica", que son mucho menos fiables que las comprobadas experimentalmente. Además, que una anotación en particular sea aplicable a un caso específico puede depender de muchos factores. Por ejemplo, del tipo celular y de variaciones en los genes, proteínas o tránscritos (splicing alternativo, SNPs, modificaciones post-translacionales, etc).

Aunque se puede acceder a la información en estas bases de datos directamente, también hay herramientas bioinformáticas que facilitan el análisis de listas de genes. Esto suele involucrar buscar los genes en múltiples bases de datos, e identificar cuáles son los términos y funciones biológicas más recurrentes o relevantes. Uno de los enfoques más extendidos en este tipo de herramientas es la búsqueda de términos sobrerrepresentados en la lista de genes del estudio. Para ello, se realizan tests estadísticos que comprueban si, para cada término, el número de genes asociados en la lista es mayor que lo que se podría esperar por azar (teniendo en cuenta el número total de genes anotados al término en la base de datos). Por ejemplo, si en una lista de genes el 20% de ellos está involucrado en una determinada vía de señalización, pero sólo el 3% de los genes del total del genoma lo están, se podría considerar que esa vía está "enriquecida". Por ello, este tipo de herramientas se denominan herramientas de enriquecimiento funcional (*Functional Enrichment Analysis*: *FEA*). Estas herramientas se basan en la premisa de que un proceso biológico normalmente se lleva a cabo a través de la colaboración de muchos genes. Si una función está alterada en una determinada condición, seguramente lo estén múltiples genes involucrados en ella.

Para calcular el enriquecimiento y realizar la anotación hay múltiples enfoques. A grandes rasgos, se suelen agrupar en tres grandes familias (*Huang et al., 2009a*): **(1)** Herramientas que calculan el enriquecimiento individualmente para cada término. Serán considerados términos enriquecidos aquellos que pasen el test estadístico correspondiente (habitualmente un test de *Fisher*, hipergeométrico o binomial). Es un enfoque sencillo, fácil de aplicar a cualquier lista de genes, pero muchas veces devuelve demasiados términos, por lo que los importantes acaban pasando desapercibidos. **(2)** Herramientas "modulares", que en vez de considerar los términos

independientemente, también tienen en cuenta las relaciones entre ellos (por ejemplo, su posición relativa dentro de la jerarquía). Este enfoque, utilizado por herramientas como *Genecodis* (*Tabas-Madrid et al., 2012*) y *TopGO* (*Alexa and Rahnenfuhrer, 2010*), permite descubrir información a través de combinaciones de términos que habrían pasado desapercibidos en los enfoques de enriquecimiento individual. **(3)** Finalmente, las herramientas de *Gene Set Enrichment Analysis* (*GSEA*) o Gene Set Analysis (*GSA*), llamadas así debido a la primera herramienta de este tipo: *GSEA* (*Subramanian et al., 2005*). El objetivo de estas herramientas es evitar tener que definir un umbral para crear la lista de genes de partida (por ejemplo, la extraída del análisis con tecnologías ómicas). Para ello evalúan todos los genes disponibles, ordenados en un ranking, en vez de analizar una lista de genes predefinida. Los métodos específicos varían según la herramienta, pero un enfoque habitual es el llamado *random walk*: testar el enriquecimiento de cada *gene set* (grupo de genes asociados a un término) añadiendo iterativamente genes del ranking. Si el término está enriquecido, cabría esperar que tenga muchos genes en el inicio del ranking. Por ello, el resultado del test va aumentando según se van añadiendo genes del inicio del ranking hasta llegar a un punto máximo en el que empiece a disminuir. Este punto máximo determina la puntuación de enriquecimiento del término. Los métodos GSA tienen la ventaja de evitar tener que elegir un umbral. Por el contrario, sólo son aplicables a casos en los que se pueda construir un ranking. Por supuesto, la elección de cómo se construye el ranking también afecta los resultados.

A pesar de la enorme ayuda de este tipo de herramientas a la hora de interpretar listas de genes, hay que tener en cuenta algunas limitaciones (*Khatri et al., 2012*). La limitación más obvia es que los resultados pueden estar muy sesgados en función de la información disponible sobre los genes en las bases de datos. Este tipo de análisis únicamente tiene en cuenta si los genes están en la lista o no. No tienen en cuenta la intensidad de la señal (expresión, metilación...) ni ningún otro valor del gen. Además, tanto los genes como las vías de señalización se tratan como variables independientes, cuando en realidad no lo son. Tanto los genes pueden regularse entre sí, como la activación de una vía puede a su vez tener efectos en otra. Incluso un único gen podría alterar una vía completamente (y sin embargo no estar "sobrerepresentado"). Para intentar solucionar estos problemas, existen herramientas que intentan tener en cuenta la topología de la red que representa la vía. Sin embargo, estas herramientas todavía no resultan eficaces. En gran medida porque la topología real de la red depende del tipo de célula y esta información todavía no está disponible o está fragmentada en muchas bases de datos. Además, las vías de señalización contienen muchos bucles de retro-regulación –que requerirían un modelado dinámico– e interacciones con otro tipo de moléculas que no se tienen en cuenta.

4.4. Aplicaciones de las redes en biología

Las redes son un método muy efectivo de presentar y visualizar relaciones con el que estamos muy familiarizados. Diariamente estamos expuestos a muchas visualizaciones basadas en redes sin ni siquiera ser conscientes de ello. Desde mapas de carreteras hasta organigramas de empresas y muchos otros tipos de esquemas. En el ámbito científico también se utilizan ampliamente tanto para visualización (p. ej. dibujos de moléculas), como para análisis (p. ej. diagramas de Feynman, análisis de grafos y topología de redes).

La biología no es una excepción. Se llevan utilizando redes desde sus inicios. De hecho, Darwin ya utilizaba árboles para representar la evolución (*Figura 7*). Actualmente también son la forma más natural para representar vías de señalización, de regulación, rutas metabólicas, etc. Sin embargo, también son un método muy útil para representar cualquier otro tipo de interacción entre moléculas: interacciones entre proteínas, unión de factores de transcripción, interacciones genéticas o coexpresión.

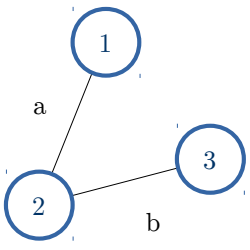
Las redes de coexpresión están formadas por genes unidos en función de la similitud de sus patrones de expresión. Normalmente la similitud se mide basada en correlación, pero también se pueden utilizar otras medidas como información mutua. Estas redes se basan en la idea de que los genes que tienen que trabajar juntos necesitan expresarse simultáneamente. En este sentido, pueden servir para identificar proteínas que son miembros de un mismo complejo (*Teichmann and Babu, 2002*), que están involucradas en las mismas funciones (especialmente si

existe conservación evolutiva, *Stuart et al., 2003*) o genes cuya expresión está regulada por un mismo mecanismo, por ejemplo factores de transcripción (*Allocco et al., 2004*). La topología de las redes de coexpresión es de gran ayuda para identificar estos grupos de genes. Su estructura modular resalta los genes que actúan como *hubs* (concentradores, nodos con muchas relaciones), ya que la mayoría de los genes se asocian a pocos genes coexpresados (*Barabási and Oltvai, 2004*).

Aunque hasta ahora casi todos los estudios de coexpresión se han hecho con datos de microarrays, últimamente también se están aplicando a datos de RNA-Seq. El RNA-Seq provee una mayor cobertura, incluyendo el transcriptoma completo en vez de limitarse a los tránscritos detectados por el array. Sin embargo, estudios que comparan los resultados obtenidos por ambas tecnologías, parecen indicar que los basados en microarray tienen una mayor similitud con las redes biológicas, y que las redes obtenidas con microarrays y RNA-Seq no son muy solapantes. Los datos de RNA-Seq preprocesados con *Variance-Stabilizing-Transformed (VST)* son los más parecidos a los de microarrays en cuanto a distribución del coeficiente de correlación y topología *(Giorgi et al., 2013*). En cualquier caso, ya hay algunas bases de datos que están empezando a incluir datos de coexpresión basados en RNA-Seq (*Obayashi et al., 2013*).

Las redes también son una forma muy natural de mostrar los resultados de relaciones funcionales. Por ejemplo, *Genemania* (*Warde-Farley et al., 2010*) y *STRING* (*Franceschini et al., 2013*) son herramientas bioinformáticas que muestran sus resultados como redes de genes o proteínas con enlaces entre aquellos que tienen algún tipo de relación funcional. También *Enrichment Map* (*Merico et al., 2010*) y *ClueGO* (*Bindea et al., 2009*) son aplicaciones de *Cytoscape* que crean redes de términos basadas en análisis de enriquecimiento funcional. Sin embargo, crear una red basada en los resultados de enriquecimiento funcional no es algo fácil y directo, y en la mayoría de las aplicaciones de enriquecimiento no se hace. Por ello, nosotros nos planteamos trabajar en esta tarea y desarrollamos la herramienta presentada en el Capítulo 2. Para entender mejor los fundamentos de esta herramienta, es importante entender cómo se representan los grafos internamente para que el ordenador los entienda (*implementación*).

Uno de los principales métodos para implementar grafos (el nombre matemático de las redes), son las matrices de adyacencia. Estas matrices son una tabla en la que las filas y columnas representan los vértices (nodos). Los elementos de la matriz representan la relación entre ellos a modo de coordenadas, poniendo un valor 1 en aquellas casillas en las que deba haber una arista (enlace) y 0 en las que no (*Figura 6*). De esta forma, se asocia cada vértice del grafo con todos aquellos vértices adyacentes (nodos enlazados). Si sólo se usan valores 1 y 0, se dice que la matriz es *booleana*. Si por el contrario se utilizan más valores, estos se pueden interpretar como *pesos*. Las matrices de incidencia son una variación sobre las matrices de adyacencia, en las que en vez de incluir los nodos en filas y columnas, usan nodos y aristas. También se pueden crear grafos en los que haya dos tipos de nodos (por ejemplo genes y fármacos, proteínas y drogas). A estos grafos se les denomina grafos *bipartito*.



	1	2	3
1	0	1	0
2	1	0	1
3	0	1	0

Matriz de adyacencia

	a	b
1	1	0
2	1	1
3	0	1

Matriz de incidencia

Figura 6. Matrices de adyacencia e incidencia.
Ambas matrices representan la red dibujada a la izquierda.

Una vez se tienen las estructuras de datos que representan la red, estas se pueden utilizar para analizar el grafo más allá de ser una mera visualización. Teniendo en cuenta que los grafos pueden llegar a ser muy grandes y complejos, es importante aprender a sacar información de ellos. ¿Qué carreteras habría que cortar para colapsar el tráfico de la zona? ¿Se pueden construir caminos desde tres casas a tres pozos sin que los caminos se crucen? Para resolver este tipo de preguntas está la teoría de grafos, el área de las matemáticas que estudia las redes. Esto lo hace a través del estudio y clasificación de su topología (grafos centralizados, distribuidos, etc), operaciones entre ellos (unión, intersección, producto, complementariedad, etc) y por supuesto el estudio de sus propiedades. Desde las más básicas, como el número de enlaces que tiene un nodo (grado, *degree*) hasta la detección de los nodos importantes o *hubs* (a través de medidas de la centralidad como el *betweeness*). Este tipo de análisis aplicados a la biología ha permitido caracterizar muchos tipos de redes biológicas (*Zhu et al., 2007*). Por ejemplo, descubriendo que las redes asociadas a procesos esenciales de la célula, como el ciclo celular, tienen muchas conexiones y bucles (seguramente para garantizar la retroalimentación y estabilidad), mientras que las redes que regulan procesos asociados a respuestas de estrés tienen un diámetro mucho menor (menor distancia de inicio a fin) y por lo tanto proporcionan una repuesta más rápida (*Luscombe et al., 2004*).

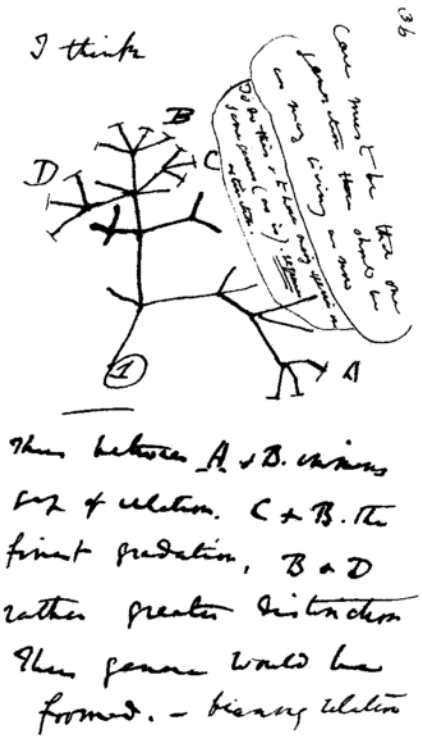


Figura 7. Las redes se han utilizado como medio de representación desde los inicios de la biología. Árbol evolutivo realizando por Darwin en 1837.
Fuente: Wikimedia commons. Original expuesto en el Museo de Historia Natural de Manhattan.

5. Enfermedades estudiadas: leucemia y síndromes mielodisplásicos

La leucemia es un tipo de cáncer hematológico que se inicia en las células de la médula ósea que generan los glóbulos blancos (del griego *leuco*, *λευκό*, blanco). En el desarrollo de las células de la sangre (hematopoyésis) a partir de células madre hematopoyéticas se dan dos líneas o linajes celulares principales: mieloide y linfoide. Si las células hematopoyéticas transformadas en malignas son de linaje mieloide, el cáncer se denomina leucemia mielocítica (o mieloide). Si por el contrario, las células hematopoyéticas transformadas son de linaje linfoide, se denomina leucemia linfocítica (o linfoblástica). En cualquiera de los dos casos, los glóbulos blancos no se producen adecuadamente y se liberan defectuosos a la sangre. Estas células defectuosas, también conocidas como células leucémicas o *blastos*, no pueden cumplir su función y proliferan de modo anómalo. Además, al producirse en cantidades mucho mayores a lo habitual, se acaban acumulando en la médula y la sangre dificultando la producción de otras células sanguíneas. Por ello, si la leucemia no se controla, suele ahogar la hematopoyésis normal produciendo frecuentemente anemias (por la falta de glóbulos rojos), hemorragias y hematomas (por la falta de plaquetas) y por supuesto, mayor debilidad ante infecciones (por falta de linfocitos sanos).

Dependiendo del las características de las células anormales, las leucemias se denominan agudas o crónicas. En las leucemias agudas, las células tumorales malignas son normalmente más inmaduras y proliferan más, provocando estados patológicos más agresivos. Las leucemias agudas se suelen extender rápidamente y pueden causar problemas graves a los pacientes en cuestión de meses. En las leucemias crónicas, las células transformadas son aparentemente más parecidas a los glóbulos blancos maduros normales; aunque, igualmente, son defectuosos y no realizan sus funciones adecuadamente. Además también sobreviven más de lo normal y acaban acumulándose, pero normalmente a lo largo de años. En general las leucemias agudas son las más habituales en niños y las crónicas en personas adultas de edad avanzada.

Con estas dos divisiones principales, basadas en el linaje y agresividad de las células afectadas, se han definido tradicionalmente los cuatro principales tipos de leucemia (*Tabla 1*).

	Linfocítica	**Mielocítica**
Aguda	ALL *Acute lymphoblastic leukemia*	AML *Acute myeloid leukemia*
Crónica	CLL *Chronic lymphocytic leukemia*	CML *Chronic myeloid leukemia*

Tabla 1. Principales tipos de leucemia.
Siglas y correspondencia en inglés por consistencia con los artículos

El hecho de que las leucemias sean enfermedades neoplásicas con relativamente alta incidencia, subtipos bien definidos y que afectan a la sangre, hace que resulte accesible conseguir muestras de los pacientes y se hayan convertido en una familia de enfermedades muy estudiada. Esto ha favorecido no sólo que se conozcan muchas características a nivel molecular para mejorar el diagnóstico, pronóstico y tratamiento de los pacientes, sino que también ha permitido el desarrollo de nuevos métodos basados en tecnologías ómicas que luego se han extendido a otras enfermedades (*Ebert and Golub, 2004*). De hecho, la leucemia fue pionera en la aplicación de métodos de clasificación de enfermedades (*Golub et al., 1999*), y hoy en día es habitual usarla para probar y comparar nuevos métodos de análisis, como por ejemplo validación de métodos de clasificación.

Los síndromes mielodisplásicos (MDS), antes llamados pre-leucemias, son un complejo conjunto de enfermedades de la médula ósea que afectan a la correcta producción de células sanguíneas, principalmente en personas mayores de 60 años. La hematopoyésis defectuosa puede afectar a uno o más linajes de células sanguíneas de la línea mieloide, produciendo citopenias (reducción del número de células) o displasias (anomalías). Dependiendo de la severidad de la enfermedad y de las células sanguíneas a las que afecta, los MDS se dividen en varios subtipos. Sin embargo, a diferencia de los grandes tipos de leucemia, las divisiones en subtipos dentro de la familia de síndromes mielodisplásicos son menos claras. Los pacientes en los estados denominados de "bajo riesgo" (*Figura 9*), pueden permanecer años en ese estado, y no resultan fáciles de identificar, no siendo inusual que sean diagnosticados con otras afecciones hematológicas no malignas ni tumorales. Los MDS de "alto riesgo" son mucho más agresivos y casi siempre acaban derivando en leucemia en un margen de apenas unos meses si no son tratados.

La división y diagnóstico entre los subtipos de enfermedad se establece principalmente por el porcentaje de blastos (células inmaduras) presentes en la muestra de sangre o médula ósea [1] (*Vardiman et al., 2009*). También hay ciertas anomalías citogenéticas –principalmente en los cromosomas 5, 7, 8 y 20– que se detectan recurrentemente y se utilizan en el sistema de pronóstico (*Figura 8*). A pesar de ello, no está claro cuáles son las características genéticas comunes o diferenciales entre todos los subtipos, ni a nivel global de todos los MDS. En general, es una familia de enfermedades muy heterogénea, lo que dificulta su estudio y diagnóstico incluso a nivel clínico.

Con los estudios ómicos, se han encontrado mutaciones somáticas asociadas a los MDS en más de 40 genes (*Papaemmanuil et al., 2013*). Se estima que el 78% de los MDS tienen alguna mutación: se han detectado mutaciones recurrentes en genes relacionados con *splicing* del RNA, transducción de la señal, complejo de cohesión, modificaciones de histonas, metilación de DNA y regulación de la transcripción (*Zhang et al., 2015*). Sin embargo, no se ha localizado ninguna vía de señalización específica común a múltiples subtipos en la que se concentren estas mutaciones (*Raza and Galili, 2012*). Esto también se refleja a nivel transcriptómico. En un gran estudio a nivel internacional, se construyó un clasificador basado en microarrays de expresión para estudiar y clasificar los distintos tipos de leucemia y hemopatías malignas. En este estudio, los MDS fueron los que peor resultados de clasificación obtuvieron, incluso manteniéndolos como un único grupo sin subtipos (*Haferlach et al., 2010*).

Los estudios moleculares y celulares sobre los MDS siguen siendo difusos o contradictorios en muchos aspectos. En general, la alteración de las células madres hematopoyéticas y del microambiente de la médula ósea (*Bulycheva et al., 2015*) hace que se produzcan mayor cantidad de células y que se acumulen en la médula ósea. Sin embargo, estas células no siguen un desarrollo hematopoyético normal y en sangre provocan frecuentemente deficiencia de células (citopenias). Una de las principales funciones alteradas que se detectó en los MDS fue el aumento de apoptosis (muerte celular programada, *Raza et al., 1995*) (que se pierde al convertirse en AML; *Parker et al., 1998*). Este aumento de apoptosis –que sigue siendo la principal característica común que se ha encontrado en MDS– podría ser la causa de las citopenias: aunque las células hematopoyéticas tumorales transformadas se reproducen más rápido de lo normal, parece que en una parte importante de la población celular maligna se induce la muerte programada antes de llegar a la sangre (*Galili et al., 2009; Raza and Galili, 2012*).

[1] El porcentaje de blastos está recomendado que se obtenga por inspección y contaje al microscopio, utilizando una cuenta diferencial de 200 leucocitos de un frotis de sangre, ó 500 células mononucleadas de médula ósea.

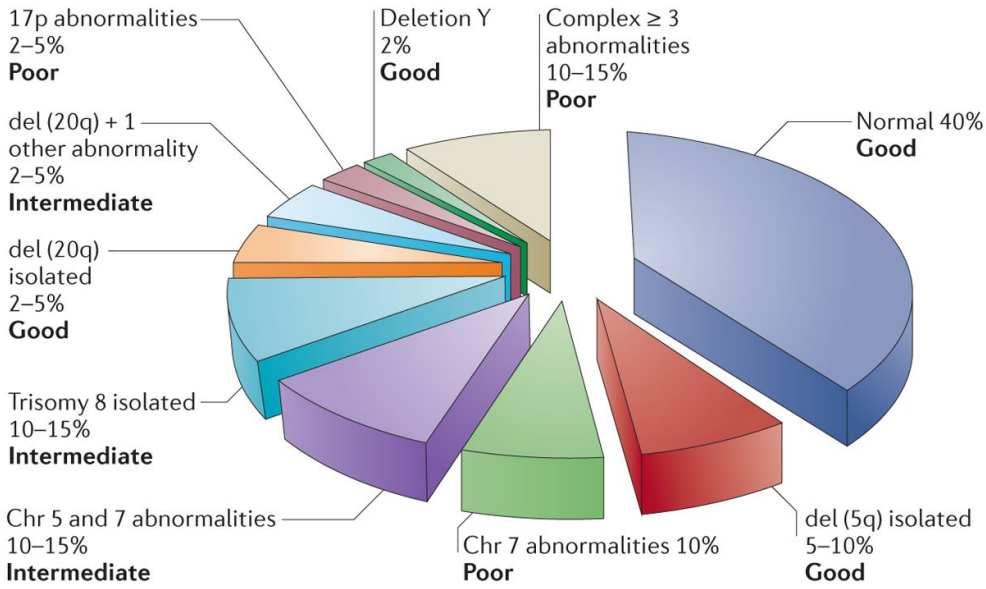


Figura 8. Principales cariotipos en los síndromes mielodisplásicos.
Figura tomada de *Raza & Galili 2012*

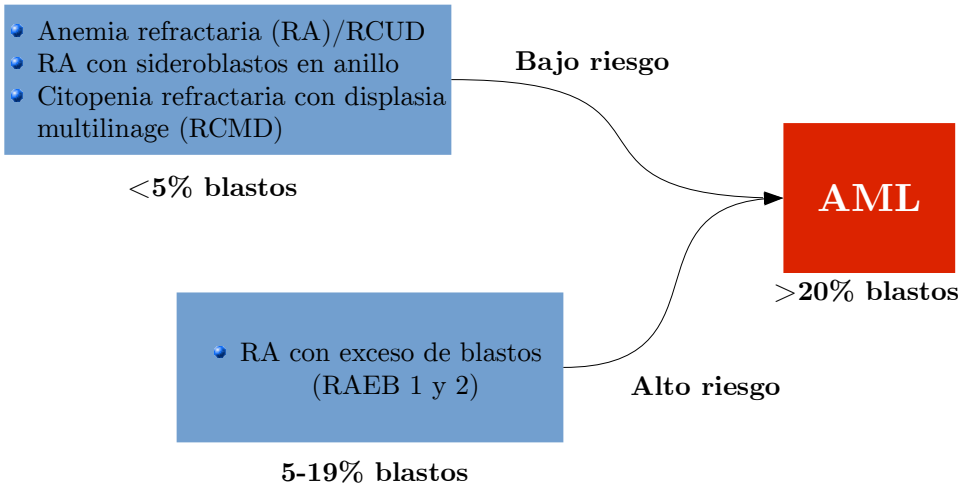


Figura 9. Principales subtipos de MDS con los que se trabajará en los estudios de esta Tesis.
La anemia refractaria (RA) actualmente está incluida dentro de las citopenias refractarias con displasia unilinaje (RCUD)

Hipótesis y objetivos

Hipótesis generales

- Las enfermedades somáticas adquiridas afectan y alteran distintos subconjuntos de genes que en muchos casos no son conocidos por no existir todavía análisis genómicos detallados de las mismas.
- Las enfermedades complejas suelen presentar múltiples subtipos patológicos que a veces son difíciles de diferenciar biomédicamente y para los cuales interesa encontrar genes marcadores o perfiles génicos característicos.
- Frente a la búsqueda de genes singulares para una enfermedad, el análisis de conjuntos de genes y su integración en redes relacionales pueden facilitar la identificación de los procesos biológicos afectados en las patologías. La identificación de estos procesos se puede lograr mediante análisis de enriquecimiento funcional.
- El análisis robusto por métodos bioinformáticos e integración de datos ómicos de distintos tipos obtenidos con plataformas complementarias –p.e. expresión y metilación– es muy necesario para obtener una visión molecular nueva de las enfermedades.
- El análisis de datos ómicos de cohortes clínicamente bien controladas de enfermedades malignas complejas (como MDS) y de sus subtipos patológicos debe permitir identificar nuevos genes directores relacionados con el diagnóstico y pronóstico de los enfermos estudiados.

Objetivo general

Desarrollar métodos y estrategias computacionales para análisis integrado y robusto de datos derivados de tecnologías ómicas –principalmente genómica y transcritómica– obtenidos de muestras de pacientes con origen clínico, para obtener perfiles moleculares génicos característicos de distintos subtipos de cáncer o estadios de enfermedades malignas y de su progresión.

Objetivos específicos

1. Desarrollar una herramienta para el análisis automático de perfiles de expresión de subtipos de enfermedades –o estados patológicos relacionados– estudiados conjuntamente, que permita identificar los grupos de genes alterados o asociados únicamente en cada uno de los estados y provea información de posibles relaciones entre los genes identificados para cada enfermedad. Sería deseable que también informe de la capacidad de los genes como biomarcadores para discriminar los estados patológicos.

2. Desarrollar un método bioinformático para facilitar el análisis, visualización e interpretación de listas de genes etiquetados con múltiples anotaciones biológicas funcionales. El método de visualización debe ser compatible con herramientas de análisis de enriquecimiento funcional que realicen un agrupamiento (*clustering*) de los resultados y debe permitir estudiar las asociaciones entre genes y términos. Posterior implementación como herramienta para su uso público.

3. Realizar el análisis bioinformático combinado de datos genómicos de expresión y de metilación en síndromes mielodisplásicos de bajo riesgo para identificar los genes cuya expresión puede estar alterada debido a cambios en los perfiles de metilación del DNA.

4. Realizar la integración y análisis bioinformático de perfiles transcriptómicos de síndromes mielodisplásicos y de leucemia mieloide aguda procedentes de varios estudios clínicos complementarios. Para ello se estudiaran las diferencias a nivel transcriptómico entre las distintos subtipos de enfermedad y se desarrollará una metodología alternativa para analizar la evolución del perfil de expresión de los distintos estados de la mielodisplasia en su progresión hacia la leucemia.

Resúmenes

Capítulo 1

Resumen del artículo 1: Análisis de subtipos de enfermedades y construcción de redes usando perfiles de expresión genómicos

Título original: Analyse multiple disease subtypes and build associated gene networks using genome-wide expression profiles

Autores: Sara Aibar, Celia Fontanillo, Conrad Droste, Beatriz Rosón, Francisco J. Campos-Laborie, Jesus M. Hernández-Rivas and Javier De Las Rivas

Aceptado en *BMC Genomics* (2015) Vol 16 Suppl 4

Paquete de Bioconductor asociado: *geNetClassifier*
URL: http://bioconductor.org/packages/release/bioc/html/geNetClassifier.html

La versión original del artículo, incluyendo figuras y ficheros adicionales, está disponible en la sección en inglés: página 57

Introducción y objetivos

La identificación de firmas génicas suele ser un objetivo muy habitual en los estudios de estados patológicos. Es especialmente interesante identificar genes marcadores que permitan diferenciar una enfermedad de otras relacionadas o subtipos de la enfermad entre sí. Sin embargo, para estudiar una patología a fondo, es importante no sólo identificar los genes alterados en ella, sino también entender su papel en su desarrollo. Para ello, es necesario disponer de información de cómo pueden estar relacionados o interactuando los genes entre sí. Además, a pesar del gran aumento de estudios transcriptómicos para estudiar enfermedades, todavía son necesarios enfoques integrativos que permitan caracterizarlas de forma sencilla.

Con el objetivo de abordar estos problemas hemos desarrollado una metodología bioinformática para la caracterización molecular de enfermedades basada en sus perfiles transcriptómicos. Esta metodología integra métodos estadísticos y de aprendizaje automático en una única herramienta que provee:

1. Reconocimiento de genes asociados a cada uno de los subtipos de enfermedad o estados patológicos estudiados: estos genes que están alterados se identifican comparando los perfiles de expresión de los estados analizados en el estudio.
2. Selección de un grupo reducido de genes que permite diferenciar los subtipos patológicos entre sí.
3. Medida del *poder discriminante* de los genes: el poder discriminante representa la relevancia de cada gen para poder diferenciar los estados patológicos.
4. Construcción de redes de genes asociadas a cada uno de los subtipos de enfermedad.

Esta herramienta está implementada en un paquete de R/Bioconductor llamado *geNetClassifier*. *geNetClassifier* está disponible libremente para facilitar estudios como los ilustrados en el artículo. Su manual se incluye como *Additional File 4*.

Métodos (Algoritmo)

1. Genes asociados a cada enfermedad: Ranking de genes

Teniendo en cuenta que las enfermedades pueden afectar distintos conjuntos de genes, en patologías relacionadas probablemente haya genes afectados por todas ellas y genes afectados únicamente en alguna específica (Figura 1A). El objetivo de *geNetClassifier* es identificar aquellos genes que tienen un perfil de expresión diferente en cada una de las enfermedades estudiadas. Para ello, se exploran sus perfiles de expresión y se construye un ranking de genes.

El ranking se construye basado en la expresión diferencial de cada gen en cada enfermedad representada en su valor de probabilidad posterior. La probabilidad posterior para cada gen en cada clase (enfermedad) se calcula comparando los patrones de expresión diferencial entre las distintas enfermedades con un contraste *uno-contra-resto* (OvR): se comparan las muestras de una clase con las muestras de las otras clases (el resto de las muestras). Para ello, se utiliza un método de esperanza-maximización (EM) para modelos de mixturas de expresión génica (implementado en *emfit* del paquete *Ebarrays, Yuan et al., 2007*). Como la probabilidad posterior cuantifica la diferencia entre la expresión del gen en una clase y en el resto (1 es el mejor valor, 0 el peor), se puede utilizar para construir el ranking.

La versión previa del ranking se construye ordenando los genes según su probabilidad posterior (decreciente) después de haber filtrado los genes que no muestran ninguna diferencia entre las clases. En caso de empates, el algoritmo utiliza la diferencia de la media de la expresión en la clase y en la clase más cercana. La versión final del ranking se construye manteniendo cada gen únicamente en la clase en la que tiene el mejor ranking.

La probabilidad posterior también se puede utilizar para cuantificar el número de genes asignados a cada clase a un mismo nivel de significación. Los genes considerados significativos

para cada clase son aquellos cuya probabilidad posterior en la clase asignada está por encima del umbral (por defecto 0,95). El número de genes significativos varía en función de la enfermedad y de las enfermedades que se compare. Por ello, no es el valor total de los genes afectados por la enfermedad, sino los afectados únicamente en cada una de las enfermedades estudiadas.

2. Selección de genes marcadores y construcción de un clasificador para enfermedades

Para identificar un subconjunto de los genes asociados a cada enfermedad que sea capaz de identificarla, se utiliza un clasificador. Entrenando el clasificador con distintas combinaciones de genes, se puede comprobar si esos genes realmente son capaces de discriminar las clases.

Concretamente, el clasificador utilizado es una máquina de soporte vectorial (*SVM*). Este *SVM* se integra en enfoque *wrapper forward selection* basado en validación cruzada. Cada iteración de validación cruzada empieza entrenando un clasificador con el primer gen del ranking de cada clase. Se entrena un clasificador con estos genes y se evalúa y guarda su precisión. En cada paso de la iteración, se añade un gen más del ranking de las clases en las que no se consiguió una "predicción perfecta" en la iteración anterior (es decir, aquellas clases en las que no se identificaron correctamente todas las muestras). Los genes se van tomando del ranking de genes de cada clase, hasta que en alguna iteración se llegue al máximo número de genes o hasta que se consiga identificar correctamente las muestras de todas las clases, dependiendo de los parámetros elegidos para la ejecución. La precisión de cada uno de los clasificadores entrenados en estas iteraciones se guardan para seleccionar el número óptimo de genes: el número de genes que dio lugar al clasificador con menor error (en caso de empate se elige el menor valor).

Para conseguir una buena estabilidad en el número de genes seleccionados, la validación cruzada se repite con nuevos muestreos (por defecto, seis veces). En cada una de estas iteraciones, se selecciona un número de genes que provee el menor error. La selección final se realiza basada en el número de genes seleccionado en todas las iteraciones.

Los genes seleccionados con este método son los que se consideran el mínimo subgrupo para identificar las clases (marcadores). Además, con estos genes se entrena un clasificador que puede ser utilizado para clasificar nuevas muestras. Este clasificador se analiza para obtener el *poder discriminante* de los genes.

Clasificación de nuevas muestras: *geNetclassifier* se construyó siguiendo un enfoque que simula las decisiones tomadas por expertos humanos. Por ello, el clasificador entrenado para clasificar nuevas muestras considera la posibilidad de que al hacer una clasificación algunas muestras queden como "no asignadas" si no está seguro de a qué clase pertenecen. Para ello, al realizar la asignación se requiere que la probabilidad de asignar una muestra a una clase sea mayor que la "probabilidad aleatoria" de pertenecer a esa clase. Además, la diferencia con la siguiente clase, debe ser al menos 0.8 veces la probabilidad aleatoria. Si no se cumplen ambas condiciones, se considera que la asignación no es segura, y se mantiene la muestra como "no asignada" (NA). Estos valores por defecto se pueden cambiar por el usuario para hacerlos más o menos restrictivos, o incluso bajarlos a cero para que las muestras siempre se asignen a la clase en la que tienen mayor probabilidad.

3. Poder discriminante de los genes

El *poder discriminante* es un parámetro que representa la importancia de ese gen para diferenciar las clases. Un valor de *poder discriminante* alto (en valor absoluto) indica que el gen es útil para marcar e identificar muestras de la clase asignada.

El *poder discriminante* se calcula basado en los coeficientes de Lagrange de los vectores soporte de cada gen seleccionado para clasificación. Debido a que el *SVM* multi-clase que utilizamos es una implementación *uno-contra-uno* (OvO), utiliza un grupo de vectores soporte para cada comparación binaria entre clases. Para cada gen, se suman los coeficientes de Lagrange de los vectores soporte de cada clase, y dan un valor para cada clase (barras apiladas en la figura). El *poder discriminante* se calcula como la diferencia entre el valor de la clase mayor menos el de la siguiente (distancia marcada por dos líneas rojas en las figuras).

4. Red de genes asociada a las enfermedades

Para estudiar las posibles asociaciones entre los genes asignados a cada enfermedad, *geNetClassifier* calcula la correlación e información mutua entre los perfiles de expresión de los genes. Esto permite identificar posibles relaciones de coexpresión entre los genes o detectar redundancia de cara a elegir marcadores. Con las relaciones detectadas se construye una red para cada clase. La red integra también los parámetros derivados de los análisis de expresión diferencial (sobre-expresión/infra-expresión) y selección de genes (poder discriminante...). De este modo, se obtiene una visión integrativa de los genes asociados a cada enfermedad y las relaciones entre ellos.

Resultados (Validación)

Para ilustrar el funcionamiento de la herramienta, aplicamos *geNetClassifier* a dos series de datos de leucemia independientes: una serie de datos de microarrays de expresión con 250 muestras de pacientes de los cuatro tipos principales de leucemia (ALL, AML, CLL, CML) y pacientes de control (NoL); y una serie de datos de RNA-Seq con 45 muestras de dos subtipos de AML, entre ellos el subtipo "cariotipo normal" (nk-AML) incluido en el set de datos de microarrays. Con ambas series de datos se realizaron múltiples ejecuciones dejando muestras fuera del entrenamiento para poder ser usadas como validación externa.

Los resultados obtenidos muestran buena concordancia entre ambos sets de datos, a pesar de las diferencias en la plataforma y los subtipos incluidos en cada uno de ellos. En ambos casos entre los primeros genes del ranking de cada enfermedad están genes ampliamente conocidos y reportados en la literatura como alterados en la enfermedad. Por ejemplo, en el caso de AML con cariotipo normal (la enfermedad incluida en ambos sets de datos) se seleccionan genes como ANGPT1, MEIS1 y múltiples genes del clúster HOXA. Las redes de genes obtenidas para cada subtipo se pueden ver en las *Figuras 3* y *5* del artículo original. Las *Figuras 1* y *4* ilustran otros de los resultados y gráficos producidos por *geNetClassifier*.

La validación externa de los clasificadores entrenados con los dos sets de datos da resultados con precisión por encima del 90% en todos los casos. Siendo el valor mínimo siempre para nkAML, y obteniendo resultados del 100% de sensibilidad y especificidad para ALL y CLL. Estos resultados confirman que los genes elegidos para cada enfermedad son buenos marcadores, permitiendo diferenciarlas del resto. El paquete también incluye la opción de simular una validación externa a través de validación cruzada con los datos suministrados.

Conclusiones

Los resultados de aplicar la herramienta a los dos sets de datos muestran que *geNetClassifier* ofrece una selección robusta de genes para marcar e identificar subtipos de enfermedades. Además, permite crear redes de genes asociados a cada enfermedad integrando información como el nivel de expresión y poder discriminante del gen, y la especificidad del gen a la enfermedad. Las anotaciones biológicas de estos genes y las redes construidas confirman su posible papel en la enfermedad. De este modo, también se comprueba la importancia de las redes para ubicarlos en un contexto que ayude a estudiar las enfermedades.

En general, se confirma que la construcción de redes de genes integrando información relevante pueden ser muy útiles para el estudio de enfermedades. Tanto para crear mapas de genes y enfermedades, como para desvelar las características moleculares de los estados patológicos. Aplicado a enfermedades con múltiples subtipos, *geNetClassifier* permite obtener esta caracterización molecular a partir de sus perfiles de expresión genómicos.

Capítulo 2

Resumen del artículo 2: Functional Gene Networks (Redes funcionales de genes). Paquete de R/Bioconductor para generar y analizar redes de genes derivadas de análisis y clustering de enriquecimiento funcional

Título original: Functional Gene Networks: R/Bioc package to generate and analyse gene networks derived from functional enrichment and clustering

Authors: Sara Aibar, Celia Fontanillo, Conrad Droste and Javier De Las Rivas

Publication status: *Bioinformatics* (2015) doi: 10.1093/bioinformatics/btu864 PMID: 25600944

Associated Bioconductor package: *FGNet*
URL: http://bioconductor.org/packages/release/bioc/html/FGNet.html

La versión original del artículo, incluyendo figuras y ficheros adicionales, está disponible en la sección en inglés: página 111

Introducción y objetivos

El incremento de estudios ómicos hace necesario disponer de métodos de análisis funcional de listas de genes o proteínas para entender los procesos biológicos en los que están involucrados. Los análisis de enriquecimiento funcional (*Functional Enrichment Analysis*: FEA) son el enfoque bioinformático más popular para la obtención de información funcional a partir de grupos de genes. Los métodos de FEA buscan en bases de datos con información biológica (Gene Ontology, Kegg pathways entre otros) y usan tests estadísticos para encontrar términos biológicos y anotaciones funcionales enriquecidas en la lista de genes. Sin embargo, en la mayoría de los casos, los resultados de estos análisis son listas muy largas de genes y términos asociados que son difíciles de digerir e interpretar. Algunas herramientas como *DAVID-FAC* (*Huang et al., 2009b*) y *GeneTerm Linker* (*Fontanillo et al., 2011*) agrupan los resultados del FEA, pero su resultado siguen siendo grandes tablas y no hay muchas herramientas para integrar y visualizar estos resultados.

En este artículo presentamos Functional Gene Networks (FGNet), un paquete de R/Bioconductor que genera genes derivadas de los resultados de análisis de enriquecimiento funcional con el objetivo de facilitar su análisis y visualización. La red funcional provee una visualización de los resultados del FEA, revelando información importante como distancia y solapamiento entre clústers, módulos de genes con funciones similares, y genes que están involucrados en múltiples funciones (conectores/*hubs*).

Métodos

1. Análisis de enriquecimiento funcional y clustering

Las herramientas de enriquecimiento funcional proveen sus resultados como grupos de genes y términos asociados llamados *gene-term sets* (*gtsets*). *FGNet* crea las redes funcionales basadas en las agrupaciones obtenidas en un clustering de *gene-term sets*. Este clustering puede venir integrado en algunas herramientas de enriquecimiento funcional, como *DAVID-FAC* y *GeneTerm Linker*, o realizarse independientemente, pero no es imprescindible. En caso de que sólo se provean los gtsets, *FGNet* considerará que cada clúster contiene un único *gtset*.

El paquete incluye una interfaz para realizar el análisis funcional de los genes a través de cuatro herramientas de FEA: *DAVID* con *clustering de anotaciones funcionales* (que devuelve clusters de *gtsets, Cl*); *GAGE* (que también provee clusters) (*Luo et al., 2009*); *Genecodis* con *GeneTerm Linker* (que devuelve metagrupos, Mg) y *TopGO* (que sólo devuelve gtsets) (*Alexa and Rahnenfuhrer, 2010*). *FGNet* también se puede utilizar con los resultados de otras herramientas de FEA si los resultados se transforman en tablas de genes y términos asociados.

2. Construcción de la red funcional

La red funcional se construye basada en el análisis de los *gtsets* que devuelve la herramienta de FEA. Estas agrupaciones de genes permiten construir una matriz booleana M de *genes* por *gtsets*, en la que cada elemento $m_{g,s}=1$ si el gen g está en el gtset s. Esta matriz de pertenencia es transformada en una matriz de adyacencia A $n \times n$; siendo n el número total de genes y $a_{i,j}$ el número de *gtsets* s en el que una pareja de genes está incluida: $\sum (m_{is} \times m_{js})(1-\delta_{ij})$, siendo δ una delta de Kronecker ($\delta_{ij}=1$ si i=j, $\delta_{ij}=0$ si i≠j). Esta matriz de adyacencia se utiliza para generar la red funcional estableciendo un enlace con peso entre los pares de genes (g_i, g_j) en los que $a_{i,j} \neq 0$. Finalmente, el clustering de gtsets se utiliza para generar una segunda matriz de adyacencia con el número de clusters o metagrupos comunes. Esta matriz se utiliza para definir los grupos de genes y determinar el layout de los genes (*Figura 1A* en el artículo). La red producida se provee como objeto *igraph* para su análisis, y puede ser exportado a otras herramientas de redes como *Cytoscape*.

3. Visualización de la red funcional

La principal visualización de la red representa los genes con asociaciones funcionales (*Figura 1B* en el artículo). Los enlaces unen genes que están en los mismos *gtsets*. Los nodos en el mismo clúster/metagrupo se ubican próximos entre sí utilizando el layout *Fruchterman-Reingold* y se colocan en una zona con un color de fondo común. Genes en un único cluster/metagrupo se dibujan con el color del metagrupo, mientras que los genes incluidos en varios clústers o metagrupos se dejan en blanco.

4. Análisis de módulos en la red

Además de construir la red, *FGNet* analiza la similitud entre los grupos de genes y provee una matriz de distancias (heatmap en la *Figura 1C* del artículo) y una red bipartita de genes con funciones solapantes (*Figura 1D*). Estos análisis permiten la cuantificación del solapamiento entre grupos y la identificación de módulos de genes.

La matriz de distancias se calcula basada en las distancias binarias en la matriz de adyacencia de clústers/metagrupos comunes. Estas distancias se agrupan con un enfoque de clustering jerárquico que se dibuja como heatmap para revelar la proximidad y similitud entre los clústers o metagrupos de genes. (Para más detalles ver la documentación del paquete).

Ejemplo de uso y resultados

Hemos aplicado el método a distintos sets de datos y confirmado que la red funcional realmente facilita el análisis de los resultados de enriquecimiento. La figura 1 del artículo muestra los resultados de *FGNet* para una lista de 175 genes diferencialmente expresados en neuronas humanas de corteza entorrinal de pacientes con Alzheimer (GEO dataset: GSE4757).

Realizando el análisis de enriquecimiento funcional a través de GeneTerm Linker, se obtuvieron seis metagrupos que etiquetamos de acuerdo a sus anotaciones principales:

Mg1: Adhesión celular

Mg2: Canales de voltaje sodio/potasio (*voltage-gated ion/potassium channels*)

Mg3: Axón y proyección celular

Mg4: Dendrita y cuerpo neuronal

Mg5: Interacción entre ligando y receptor en sinapsis neuronal

Mg6: Señalización de MAPK y Alzheimer

La red de estos seis metagrupos provee una visión general del solapamiento funcional de los genes y permite identificar genes *hubs* de grupos. Por ejemplo, CNTNAP1 y NLGN4X aparecen como *hubs* en el metagrupo 1. CNTNAP1 (que regula la distribuciones de canales de potasio) enlaza el metagrupo 1 con el 2, y NLGN4X (que facilita la transmisión sináptica) enlaza el metagrupo 1 con el 4 y el 5. NLGN4X es el gen con mayor centralidad (*betweenness*) en esta red. Otro *hub* importante es APOE, recientemente asociado con Alzheimer.

La matriz de distancia permite cuantificar la similitud entre los grupos de genes, mostrando que los metagrupos más comunes son el 3, el 4 y el 6, compartiendo ocho nodos. Esto también se muestra en la red de intersección.

Finalmente, la red funcional puede revelar más información sobre los metagrupos. Por ejemplo, si un metagrupo comparte muchos genes con otros metagrupos, puede ser indicación de que muchos de los genes de la red están involucrados en esas funciones. Este es el caso para el metagrupo 6, que está anotado a Alzheimer.

Conclusión

Las redes funcionales producidas por *FGNet* ofrecen un método eficaz de visualización de resultados de FEA y clustering de anotaciones funcionales. Estas redes permiten identificar rápidamente qué genes tienen anotaciones funcionales en común (genes enlazados y módulos) y genes anotados a muchas funciones (hubs entre clústers). Además los análisis complementarios como el heatmap de distancias entre clústers, la red bipartita y las estadísticas sobre los genes, permiten explorar más a fondo los resultados, cuantificar las intersecciones de metagrupos y detectar módulos de genes con múltiples anotaciones comunes. De este modo, *FGNet* facilita enormemente la interpretación de los procesos biológicos significativamente enriquecidos en la lista de genes inicial.

Capítulo 3

Resumen del estudio 1:
Análisis combinado de perfiles de expresión y metilación genómicos en síndromes mielodisplásicos de bajo riesgo

El informe completo sobre el estudio está disponible en la sección en inglés: página 151

Introducción y objetivos

En la época en la que empezó este estudio, se sabía que la metilación del DNA está alterada en los síndromes mielodisplásicos (MDS). Sin embargo, todavía quedaban muchas cuestiones por responder. Entre ellas, hasta qué nivel estaba alterada la metilación del DNA a escala global del genoma y cómo podían afectar estos cambios a la expresión. Por ello, se empezó un estudio colaborativo para caracterizar las alteraciones en los perfiles de expresión y metilación del DNA en pacientes de síndromes mielodisplásicos de bajo riesgo. El estudio se llevó a cabo a través del análisis integrativo de los perfiles de expresión y metilación del DNA obtenidos de la misma cohorte de pacientes de MDS de bajo riesgo.

En este capítulo se presenta el trabajo bioinformático realizado para analizar e integrar los datos de expresión y de metilación genómica. El objetivo de este análisis era por un lado identificar los patrones de expresión y metilación alterados en los MDS de bajo riesgo, y posteriormente integrarlos para determinar los genes que pudiesen estar silenciados debido a la hipermetilación del DNA. De esta forma, se pueden identificar procesos biológicos potencialmente alterados por la metilación aberrante en los síndromes mielodisplásicos de bajo riesgo.

Material y métodos

Pacientes y muestras: Los análisis de expresión y metilación del DNA se realizaron sobre la misma cohorte de pacientes de MDS de bajo riesgo. Concretamente 6 pacientes con anemia refractaria (RA) y 12 pacientes con anemia refractaria con sideroblastos en anillo (RARS). Además, también se incluyeron muestras de 7 pacientes con otras afecciones, distintas a los MDS y leucemia, que se utilizarían como control (NoL). Para cada paciente se hibridaron los microarrays arrays de expresión y de metilación con RNA/DNA extraído de células mononucleares de su médula ósea.

Análisis de metilación: El análisis de metilación se llevó a cabo utilizando una plataforma denominada *microarray de amplificación de islas CpG metiladas* (MCAM). Esta plataforma se basa en hibridar un microarray de islas CpG con los amplicones de DNA obtenidos al seleccionar y ampliar fragmentos de DNA metilados. Los fragmentos metilados se obtienen utilizando encimas de restricción que corta el DNA en las secuencias CCCGGG. En el primer paso se utiliza la encima que corta la secuencia si no está metilada (provocando que estas zonas se eliminen) y posteriormente se utiliza la encima complementaria, que corta los CCCGGG metilados, pero dejando extremos cohesivos (*overhang*) que permite amplificarlos por PCR.

El microarray de islas CpG utilizado es una plataforma de dos colores que contiene 12.192k islas CpG (University Health Network, Toronto, Canada). Este array se hibrida con los amplicones obtenidos para cada muestra de paciente etiquetada con Cy5 (rojo) utilizando un DNA comercial (Human Cot-1) etiquetado con Cy3 (verde) como control. Las muestras de NoL también se hibridan en los arrays de forma equivalente, pero dividiendo las muestras disponibles en dos grupos y mezclando las de cada grupo en un *pool*. De este modo se utilizan únicamente dos microarrays.

Una vez se obtuvieron los datos de los microarrays, se procesaron y se realizaron los análisis para identificar las islas CpG diferencialmente metiladas entre las muestras control (NoL) y los MDS de bajo riesgo (AR y ARS).

Análisis de expresión: Los perfiles de expresión se obtuvieron a través de microarrays de alta densidad de Affymetrix *GeneChip® Human Genome U133 Plus 2.0*. Tras el preprocesamiento adecuado, se utilizó SAM (*Tusher et al., 2001*) para identificar los genes diferencialmente expresados en las muestras control (NoL) y los MDS de bajo riesgo (AR y ARS).

Integración de metilación y expresión: Puesto que los datos de metilación están basados en islas CpG, hubo que relacionarlos con los genes en sus proximidades para saber a qué genes podían estar afectando las variaciones en su metilación. Para ello se utilizaron las anotaciones del microarray, en las que se indican qué genes hay ubicados en las proximidades de cada isla.

Tras identificar los genes diferencialmente expresados y las islas CpG diferencialmente metiladas entre los pacientes control y los MDS de bajo riesgo, se identificaron los genes que tenían perfiles de expresión y metilación "consistentes". Puesto que la metilación se conoce como un mecanismo de inhibición de la expresión, se consideraron genes con perfiles de expresión y metilación "consistentes" los genes infraexpresados ubicados en la proximidad de islas hipermetiladas y, por completitud, también los genes sobreexpresados ubicados cerca de islas hipometiladas. Para calcular los genes con perfiles de expresión y metilación consistentes, se buscaron las intersecciones correspondientes entre los genes diferencialmente metilados (p-valor$<$0.15) o expresados (FDR$<$0.15), resultando en un p-valor combinado máximo para estos genes de 0.0225.

Resultados

Los análisis identificaron 476 genes asociados a las 246 islas CpG diferencialmente metiladas y 334 genes diferencialmente expresados con p-valor o FDR $<$ 0.05. Entre los 1198 genes diferencialmente metilados y los 1975 genes diferencialmente expresados a p-valor o FDR $<$ 0.15, se identificaron 122 genes en ambas listas. De estos 122 genes, 64 genes tenían patrones de expresión y metilación consistentes (*Figuras 6 y 7 en la versión original*).

Los 64 genes con perfiles de expresión y metilación consistentes se exploraron más a fondo. El análisis funcional de estos genes, reveló que había muchos de ellos involucrados en la regulación de la expresión génica, apoptosis, ciclo celular y proliferación. Entre ellos se identificaron algunos genes clave. Por ejemplo, ETS1 (un factor de transcripción hipermetilado e infraexpresado) con muchos genes diana también infraexpresados.

Conclusiones

El análisis combinado de perfiles de expresión génica y metilación del DNA realizado en este estudio permitió identificar un conjunto de genes cuya activación puede estar alterada en MDS de bajo riesgo debido a la alteración de sus perfiles de metilación. Las funciones asociadas a estos genes desvelaron que la metilación aberrante podría ser la causa de algunas de las funciones frecuentemente alteradas en MDS (apoptosis, ciclo celular, diferenciación...). Además, se identificaron algunos mecanismos concretos por los que los genes hipermetilados e infraexpresados podrían estar contribuyendo a la patogénesis de la enfermedad (ETS1 a la infraexpresión de muchos genes, BCL2 al aumento de apoptosis, IR27RA a la alteración de la respuesta inmune, DICER al procesamiento de miRNAs, ...). Con todos estos resultados, este estudio es un buen ejemplo de cómo los métodos bioinformáticos robustos aplicados al análisis e integración de diferentes capas de datos ómicos pueden ayudar a mejorar el conocimiento sobre los mecanismos moleculares subyacentes o causantes de las enfermedades estudiadas.

Capítulo 4

Resumen del estudio 2:
Integración de perfiles de expresión procedentes de distintas plataformas genómicas para identificar patrones de expresión en la progresión de los síndromes mielodisplásicos hacia leucemia

El informe completo sobre el estudio está disponible en la sección en inglés: página 177

Introducción y objetivos

Los síndromes mielodisplásicos (MDS) son un grupo muy heterogéneo de hemopatías malignas. Aunque están divididos en varios subtipos, el principal criterio utilizado para su división y para determinar el diagnóstico de un paciente es el porcentaje de células blásticas en la sangre y médula ósea (*Vardiman et al., 2009*). Además, la gran heterogeneidad de los MDS, incluso entre pacientes de un mismo subtipo, también se refleja a nivel genético. Por ello, resulta difícil encontrar características unificadoras para todos los MDS de un mismo tipo que las diferencien de los demás.

Con el objetivo de estudiar la subdivisión y patogénesis de los síndromes mielodisplásicos a nivel transcriptómico, se realizó un estudio integrativo de tres series de datos de MDS que incluyen pacientes de bajo riesgo (RCUD y RCMD) y de alto riesgo (RAEB 1 y 2). Basándose en la hipótesis de que estos tipos de MDS son estadíos consecutivos en la progresión hacia leucemia, se buscaron patrones en la expresión de los genes que pudiesen estar asociados con la enfermedad. Para ello, se probaron varios enfoques. Finalmente el más eficaz e interesante resultó ser la búsqueda de genes cuya expresión siga una tendencia creciente o decreciente en las distintas MDS, estando estas ordenados de menor a mayor riesgo/malignidad: NoL → Bajo riesgo (RCUD, RCMD) → Alto riesgo (RAEB 1, RAEB 2) → AML.
La expresión de estos genes estaría asociada con el nivel de riesgo o malignidad de la enfermedad, y por tanto, permitirían estudiar qué procesos biológicos relacionados con la progresión a leucemia se alteran en los MDS.

Material y métodos

Datos y preprocesamiento: El estudio se realiza con tres series de datos en microarrays de expresión de *Affymetrix*, dos de ellas con Human Genome U133 Plus 2.0 Array (*hgu133plus2*) y otra con Human Exon 1.0 ST Array (*HuEx1.0*). La serie en *HuEx1.0* es la serie principal; las dos series en *hgu133plus2* se utilizaron como validación tras fusionarlas en un único set de datos utilizando *frozen RMA* (*McCall et al., 2010*) e *inSilicoMerging* (*Taminau et al., 2012*).

Las tres series de datos se realizaron con muestras de médula ósea de pacientes de MDS de distintos tipos, leucemia (AML) y pacientes control con otro tipo de afecciones (NoL). De ellos se seleccionaron las muestras que incluían sólo las células mononucleadas de médula ósea y de pacientes con cariotipo normal. De los MDS disponibles se utilizaron únicamente las muestras de pacientes con RCUD, RCMD, RAEB 1 y RAEB 2 (sin sideroblastos).

Búsqueda de patrones de expresión: La búsqueda de genes con tendencias crecientes/decrecientes en la progresión de la enfermedad se realizó utilizando un enfoque basado en la correlación gamma (*Goodman and Kruskal, 1954*): seleccionar aquellos genes cuya expresión correlaciona con una variable categórica que representa el estadío de la muestra en la evolución de la enfermedad.

Los perfiles de expresión de los genes seleccionados por este método se agruparon utilizando SOM (*Kohonen, 1982*), un método de clustering no supervisado, que dio lugar a cuatro patrones (dos crecientes y dos decrecientes). En dos de los patrones la mayor variación de la expresión se observaba principalmente en AML. En los otros dos, la variación de la expresión era más constante durante la progresión de la enfermedad, siendo un gran porcentaje del cambio total (desde NoL a AML) durante las MDS.

Análisis de los genes y patrones: Para explorar y validar los genes obtenidos por estos métodos se realizaron varios análisis: (i) Análisis funcional de las listas de genes, (ii) asociación de los patrones obtenidos con el porcentaje de células blásticas en la muestra, (iii) confirmación de la existencia de los perfiles crecientes/decrecientes en otros sets de datos de

MDS y (iv) estudio de la regulación de los genes de las listas a través de factores de transcripción.

Resultados

La integración de las dos series de *hgu133plus2* se realizó con éxito, pudiendo utilizarla como una única serie de datos con una gran cohorte de pacientes (Serie 1+2).

Los análisis de los perfiles de expresión identificaron 1163 genes cuya expresión en el set de datos principal correlacionaba con el nivel/estadío de la enfermedad (*Full List*). De estos 1163 genes, 266 genes se confirmaron también con los datos de la Serie 1+2 (*Core List*). En las validaciones, se confirmó que estas tendencias crecientes/decrecientes también se encuentran en otros sets de datos.

Los análisis de los cuatro patrones identificados determinaron que la expresión de los genes en los patrones con mayor cambio en AML (FLT3, MEIS1, HOXA...) también está más asociada al porcentaje de blastos presente en la muestra. Por otro lado, entre las funciones claramente alteradas en los patrones de cambio en los MDS destacan los ribosomas. Entre los 1163 genes con expresión creciente/decreciente, también se identificaron varios factores de transcripción que a su vez regulan otros genes en la lista. En las redes de regulación se observó que las redes correspondientes a los patrones de leucemia son mucho más complejas e interconectadas que las de los genes con patrones asociados a MDS.

Conclusiones

La metodología utilizada en este estudio permitió identificar genes asociados con la malignidad en los MDS: genes cuya expresión empieza a alterarse en los síndromes mielodisplásicos y continúa esa tendencia en la progresión hacia leucemia. En vez de utilizar el enfoque tradicional de buscar genes marcadores, se estudió la patogénesis de los MDS centrándose en las similitudes y evolución de los perfiles mielodisplásicos y la leucemia. Para ello, la disponibilidad de las muestras de leucemia como referencia del estado maligno de "destino" fue un punto clave. En este sentido, la metodología utilizada en este estudio ofrece un enfoque diferente para el análisis de enfermedades con un contexto común, pudiendo ser generalizada para el estudio de otros tipos de enfermedades que se desarrollen en etapas, como por ejemplo el cáncer.

Conclusiones

Conclusiones

1. La aplicación de métodos y técnicas bioinformáticas a datos genómicos o transcriptómicos de muestras de pacientes con cáncer se ha demostrado eficaz para lograr una mejor determinación de los perfiles moleculares de cada enfermedad estudiada, y en particular, para la identificación de distintos subtipos patológicos a veces poco definidos previamente. Esto se ha demostrado en este trabajo usando técnicas computacionales automáticas de inteligencia artificial que exploran la información transcriptómica de modo preciso.

2. Dos de los métodos bioinformáticos presentados en esta Tesis Doctotral se han implementado en herramientas bioinformáticas disponibles para su uso publico en *Bioconductor* (www.bioconductor.org): *geNetClassifier* y *FGNet*.

3. El estudio de subtipos de enfermedades se puede centrar tanto en identificar los genes y características que las diferencian, como las que tienen en común. Los genes con un perfil único en cada estado son interesantes como posibles biomarcadores de la enfermedad y ayudan a conocer los procesos subyacentes en los distintos estadíos de la enfermedad. Por otro lado, las características en común dan la posibilidad de obtener una visión global del desarrollo de la enfermedad y de su posible progresión en distintas fases.

4. La integración y análisis bioinformático de distintas capas de datos ómicos permite, no solo una mejor comprensión de las enfermedades, sino que también puede ayudar a descubrir posibles causas.

5. Los análisis de enriquecimiento funcional ayudan a encontrar y comprender mejor los procesos biológicos en los que están involucrados los genes alterados derivados de estudios ómicos. Además, la identificación de funciones alteradas suele ser mucho más estable entre estudios que la identificación de los conjuntos de genes cambiados en cada paciente.

6. La aplicación de las técnicas bioinformáticas presentadas en esta Tesis Doctoral a enfermedades onco-hematológicas ha permitido mejorar su caracterización molecular basada en datos ómicos. De modo concreto, se ha obtenido un perfil génico de la evolución y progreso de los síndromes mielodiplásicos (MDS) desde sus formas más benignas de bajo riesgo hasta los estadíos de más riesgo de tipo leucémico; también, se ha logrado la integración de datos ómicos complementarios de expresión y metilación de MDS de bajo riesgo para identificar reguladores génicos antes no descritos.

English section

Introduction

The usage of omic data for biomedical sciences has been growing quickly during the last decade. These data allow to study diseases from a biomolecular point of view that was not available before. This offers huge opportunities to improve the understanding of the disease, as well as the development of new molecular diagnostic, prognostic and treatment tools. However, in order to explore in detail the overwhelming amount of data produced by omic technologies, it is necessary to develop advanced analysis techniques that allow to extract the biological information available in them. In this way, this is a very interdisciplinary research field, where the central aim is to study biomedical questions through molecular biology, applying statistical and computational methods. In particular, this Doctoral Thesis is within the field of "Bioinformatics applied to medicine", and its aim is to develop and apply new artificial intelligence and data mining methodologies to analyze genomic and transcriptomic data from human samples to characterize diseases or pathologies.

In the molecular characterization of complex diseases, it is specially important to identify altered genes in specific subtypes or defined pathologies. These genes could be used as biomarkers for diagnosis or to understand the processes underlying the disease. However, in biological systems, genes do not act as independent elements but rather interact with each other, forming gene networks working together in specific biological functions. Currently, these associations can be obtained from the analysis of genome-wide experimental molecular data, but they can also be obtained from the previous knowledge integrated into databases (i.e. functional enrichment tools). In this research project, we develop methods to integrate these analyses into automated processes, exploring data derived from biomolecular studies on clinical series, to obtain biologically relevant information about the diseases.

For the development of new bioinformatic methods to analyze data from biomedical studies, it is also important to stay in contact with clinical groups that have complex data and need to resolve issues for which there are still no available methods. For example, one of the main difficulties in biomedical studies is the clear identification of subtypes in certain diseases, specially when the separation between the states is not clear at clinical stage. This kind of issues require new analytical frameworks to study the diseases.

Within this landscape, this Doctoral Thesis has been structured into four chapters. All of them with a common objective: the development of new methodologies for analysis of omics data of disease subtypes, specially cancer. The first two chapters present two bioinformatic tools through the corresponding scientific publications; the last two chapters present the bioinformatic strategies and methodologies used to carry on two specific studies on myelodysplastic syndromes (MDS).

Chapter 1 – Article 1: Analyse multiple disease subtypes and build associated gene networks using genome-wide expression profiles. *BMC Genomics* (2015).

Associated Bioconductor package: *geNetClassifier: classify diseases and build associated gene networks using gene expression profiles*

Chapter 2 – Article 2: Functional Gene Networks: R/Bioc package to generate and analyse gene networks derived from functional enrichment and clustering. *Bioinformatics* (2015)

Associated Bioconductor package: *FGNet: Functional Gene Networks derived from biological enrichment analyses.*

Chapter 3 – Combined analysis of genome-wide methylation and expression profiles from low-risk Myelodysplastic Syndromes

Chapter 4 – Integration of multi-platform gene expression profiles and identification of expression patterns of Myelodysplastic Syndromes in their progression to Leukemia

Hypotheses and objectives

General hypotheses

- Acquired somatic diseases affect and alter different subsets of genes that in many cases are not known since their genomic profiles have still not been analyzed in detail.
- Complex diseases usually present multiple pathological subtypes that are sometimes difficult to differentiate. It would be of interest to identify gene markers or characteristic genomic profiles.
- The analysis of sets of genes –instead of independent genes– and their integration in functional networks can ease the analysis and identification of the biological processes affected in the pathology. The identification of these processes can be achieved through functional enrichment analysis.
- The robust analysis and integration of different types of omic data obtained through complementary platforms –e.g. expression and methylation– is necessary to obtain a new molecular view of diseases.
- The analysis of omic data from well controlled cohorts of patient samples with complex malignant diseases (i.e. MDS) and their pathologic subtypes should allow to identify new driver genes related to the diagnostic and prognosis of the studied patients.

General objective

To develop computational methods and bioinformatic strategies for integrated and robust analysis of omic data –mainly genomics and transcriptomics– obtained from patient samples, in order to obtain genomic profiles characteristic from different cancer subtypes or malignant diseases and their progression.

Specific objectives

1. To develop a tool for the automatic analysis of expression profiles of disease subtypes –or related pathological states– studied together. This tool should identify the groups of genes that are altered or associated only to each of the states and provide information about possible associations between the genes identified for each disease. It would also be desirable that it provides information regarding the potential of each gene as biomarker to discriminate between the pathological states.
2. To develop a bioinformatic method to ease the analysis, visualization and interpretation of gene lists associated with multiple biological annotations. The visualization method should be compatible with functional enrichment analysis tools that perform a clustering of the results. It should also allow studying the associations between genes and terms based on the functional enrichment analysis results. Implement this method in a tool.
3. To perform the combined analysis of expression and methylation genome-wide profiles in low risk myelodysplastic syndromes, to identify genes whose expression might be altered due to changes in the DNA methylation patterns.
4. To perform the integration and bioinformatic analysis of transcriptomic profiles from myelodysplastic syndromes and acute myeloid leukemia from several complementary clinical studies. Study the differences between the different subtypes at transcriptomic level, and develop an alternative methodology to analyze the evolution of the expression profiles of the stages of myelodysplasia in its progression towards leukemia.

Chapter 1

Article 1: Analyse multiple disease subtypes and build associated gene networks using genome-wide expression profiles

Authors: Sara Aibar, Celia Fontanillo, Conrad Droste, Beatriz Rosón, Francisco J. Campos-Laborie, Jesus M. Hernández-Rivas and Javier De Las Rivas

Accepted for publication in *BMC Genomics* (2015) Vol 16 Suppl 4

Associated Bioconductor package: *geNetClassifier*: *classify diseases and build associated gene networks using gene expression profiles.* Published on April 2013
URL: http://bioconductor.org/packages/release/bioc/html/geNetClassifier.html

RESEARCH

Analyse multiple disease subtypes and build associated gene networks using genome-wide expression profiles

Sara Aibar[1], Celia Fontanillo[1,3], Conrad Droste[1], Beatriz Rosón[1], Francisco J. Campos-Laborie[1], Jesus M. Hernández-Rivas[2] and Javier De Las Rivas[1*]

Abstract

Background: Despite the large increase of transcriptomic studies that look for gene signatures on diseases, there is still a need for integrative approaches that obtain separation of multiple pathological states providing robust selection of gene markers for each disease subtype and information about the possible links or relations between those genes.

Results: We present a network-oriented and data-driven bioinformatic approach that searches for association of genes and diseases based on the analysis of genome-wide expression data derived from microarrays or RNA-Seq studies. The approach aims to **(i)** identify gene sets associated to different pathological states analysed together; **(ii)** identify a minimum subset within these genes that unequivocally differentiates and classifies the compared disease subtypes; **(iii)** provide a measurement of the *discriminant power* of these genes and **(iv)** identify links between the genes that characterise each of the disease subtypes. This bioinformatic approach is implemented in an R package, named *geNetClassifier*, available as an open access tool in Bioconductor. To illustrate the performance of the tool, we applied it to two independent datasets: 250 samples from patients with four major leukemia subtypes analysed using expression arrays; another leukemia dataset analysed with RNA-Seq that includes a subtype also present in the previous set. The results show the selection of key deregulated genes recently reported in the literature and assigned to the leukemia subtypes studied. We also show, using these independent datasets, the selection of similar genes in a network built for the same disease subtype.

Conclusions: The construction of gene networks related to specific disease subtypes that include parameters such as gene-to-gene association, gene disease specificity and gene discriminant power can be very useful to draw gene-disease maps and to unravel the molecular features that characterize specific pathological states. The application of the bioinformatic tool here presented shows a neat way to achieve such molecular characterization of the diseases using genome-wide expression data.

Keywords: gene; expression; expression profile; gene networks; microarray; RNA-Seq; disease; disease classification; cancer; leukemia; acute leukemia

Background

Last decade of experimental work using genomic technologies has provided many data on gene expression profiling of different biological and pathological states [1]. This great effort in biomedical research has lead to a large need for tools and strategies that allow clinicians to translate the genome-wide expression data into useful information, such as transparent and robust signatures to characterize and distinguish multiple pathological subtypes [2]. There are many machine learning and computational procedures that can be applied to build classification systems that allow identifying the type or category of query samples whose class is not-known *a priori* [3, 4, 5]. However, a common problem of these methods is that they often do not reveal any information about the genes that are selected as variables for the classification process [4]. Although obtaining an efficient classifier might seem enough in some cases, there is a clear loss of biological information if the value or power of the chosen genes is not translated into parameters that allow to characterize and rank the genes.

Many clinical and biomedical studies look for the separation between multiple disease subtypes as distinct pathological states, but they are also very interested in finding the specific genes that are altered in each disease subtype. To identify and quantify the power of such 'marking genes' is the only way by which machine learning techniques can bring back biological meaning to this kind of biomedical studies. Moreover, gene products do not work in isolation as 'independent features', but rather interact with others in biomolecular networks to perform specific biological functions [6]. Therefore, together with the identification of the genes that mark a disease, genome-wide studies of related biological states should also provide information about the associations between the affected genes [7].

Following these questions we have developed a bioinformatic approach to provide gene-based analysis and characterization of different diseases and construction of associated gene networks using expression profiles derived from experimental transcriptomic data. The approach integrates established statistical and ma-

*Correspondence: jrivas@usal.es
[1]Cancer Research Center (IMBCC, CSIC/USAL/IBSAL), Campus Miguel de Unamuno s/n, Salamanca, 37007, Spain
Full list of author information is available at the end of the article

chine learning methods into a single tool that allows to **(i)** identify the set of genes that are specifically altered in a disease when a collection of several diseases -or disease subtypes- are studied and compared together using genome-wide expression profiling; **(ii)** obtain a minimum subset of these genes that enable to differentiate each disease subtype from the other; **(iii)** provide information about how relevant each of these genes is for discriminating each studied class; and **(iv)** find associations between the genes based on the analysis of the experimental expression profiles. This tool has been implemented in an R/Bioconductor package named *geNetClassifier* (available at http://www.bioconductor.org/). In order to validate the tool as a whole and prove whether the results it provides have biological and functional meaning, here we present its application to two independent genome-wide expression datasets of human samples isolated from individuals with different subtypes of leukemia: one using high-density oligonucleotide microarrays and another using deep RNA-sequencing.

Results and discussion

Finding genes associated to specific disease subtypes

The human gene landscape can be structured in functionally associated groups of genes which are specific to biological processes or states. Since a disease will normally affect and alter one or several biological processes, we could depict a theoretical multidimensional "gene space" divided in regions that include genes associated to specific pathological states (**Figure 1A**). The identification of these groups of genes is a great scientific endeavour for biomedical research, and some biological databases (e.g. OMIM [8]) have been built following the idea of a "gene-to-disease mapping", as it is known to happen in Mendelian inherited diseases. In this theoretical scenario, the genes that are affected by a given disease can be overlapping with the ones affected by a similar pathological state. This will define genes that can be altered in multiple pathologies, but it will also expect to define genes that are only affected by a specific malignancy when compared with other diseases.

Considering the recognition of such theoretical gene-disease space (**Figure 1A**), we apply expression profiling to find the genes that are altered in one specific disease subtype using differential expression analysis. To do so, we compare each disease category versus all the others using package EBarrays [9], that implements an empirical Bayes method [10]. This provides a *posterior probability* for each gene to be differentially expressed in one of the classes (see Methods). Sorting the genes by their probability allows to build a ranking of the genes ordered by their statistical significance (**Figure 1B**). Since each gene has a probability of differential expession per class, it is assigned to the class in which it has the best ranking. This allows to build non-overlapping gene lists that optimize the specificity and separation between classes. The posterior probability also allows to quantify the association of a gene with a class and identify how many genes are related to each class at a certain significance level.

Constructing gene-based classifiers for multiple diseases

Once the gene rankings have been established, the tool selects from the top of the list the minimum subset of genes required to identify each class. To achieve this, it uses a multiclass implementation [11] of Support Vector Machine (SVM), as a method that has been proven very efficient for classification of gene expression microarray datasets [12, 13, 14]. The SVM is integrated into a wrapper forward selection scheme to test whether a selected subset of genes is actually enough to discriminate the classes [15]. Several **SVM** classifiers are iteratively trained with an increasing number of genes taken from the ranked lists and evaluated through double nested cross-validation. The smallest subset of genes that provides the best performance is selected as feature set (**Figure 1C**) and used to train and build a final classifier that will include all the available samples of the training set.

The classifier built for a given set of compared diseases can be used to query and identify new unlabeled samples. In addition, the classifier is analysed in order to obtain the *discriminant power* of the selected genes (**Figure 1D**). Each gene's *discriminant power* is a quantitative parameter that resembles the value of such gene in class differentiation. Therefore, a high *discriminant power* (either positive or negative, in absolute value) indicates that the gene is useful to mark and identify samples from its assigned class. Full description of this parameter is provided in Methods section.

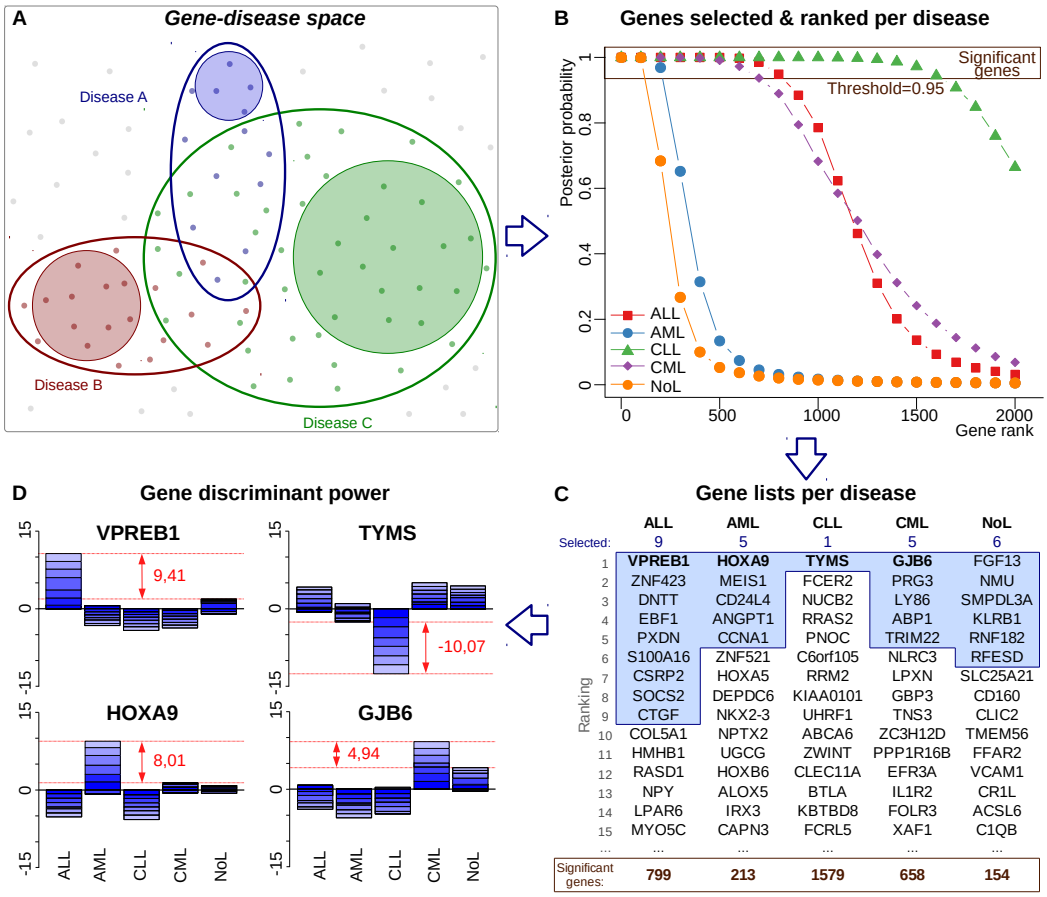
Building networks of genes associated to diseases

To infer possible associations between the genes assigned to each disease, *geNetClassifier* calculates gene-to-gene correlation and mutual information [16] in the expression dataset. This allows to identify possible relations of co-expression between the genes and possible relations of mutual redundancy. The detected associations are integrated in a network that also includes parameters derived from the differential expression analysis and from the classification analysis. Since networks are built for each class, they provide an integrative view of the gene sets associated to each disease in a relational characterized context. Examples of these networks are presented in the case studies in the following sections.

Using *geNetClassifier*: analysis of a leukemia dataset

We have applied *geNetClassifier* to a dataset of genome-wide expression microarrays of samples from leukemia, as a well known disease that allows to test the tool in a real case study and confirm the biological relevance of the results. This dataset includes 50 microarray samples from bone marrow of patients of four major leukemia subtypes (ALL, AML, CLL and CML; described in Methods) plus non-leukemia controls (NoL), making a total of 5 distinct classes.

The first result that *geNetClassifier* provides is the set of rank-ordered lists of genes selected for each class, being the top genes the ones most significantly associated with each disease (as indicated in **Figure 1C**). The resulting lists of genes-per-disease do not overlap, in this way the method is optimized to find specific markers of each compared disease. The number of genes associated to each disease for a common threshold of significance is quite different from one class to another (e.g. 1027 genes for ALL but only 273 genes for AML). This observation seems to indicate that some diseases can affect more genes than others according



	ALL	AML	CLL	CML	NoL
Selected:	9	5	1	5	6
1	VPREB1	HOXA9	TYMS	GJB6	FGF13
2	ZNF423	MEIS1	FCER2	PRG3	NMU
3	DNTT	CD24L4	NUCB2	LY86	SMPDL3A
4	EBF1	ANGPT1	RRAS2	ABP1	KLRB1
5	PXDN	CCNA1	PNOC	TRIM22	RNF182
6	S100A16	ZNF521	C6orf105	NLRC3	RFESD
7	CSRP2	HOXA5	RRM2	LPXN	SLC25A21
8	SOCS2	DEPDC6	KIAA0101	GBP3	CD160
9	CTGF	NKX2-3	UHRF1	TNS3	CLIC2
10	COL5A1	NPTX2	ABCA6	ZC3H12D	TMEM56
11	HMHB1	UGCG	ZWINT	PPP1R16B	FFAR2
12	RASD1	HOXB6	CLEC11A	EFR3A	VCAM1
13	NPY	ALOX5	BTLA	IL1R2	CR1L
14	LPAR6	IRX3	KBTBD8	FOLR3	ACSL6
15	MYO5C	CAPN3	FCRL5	XAF1	C1QB
...	...	...	...	...	...
Significant genes:	799	213	1579	658	154

Figure 1 Description of geNetClassifier main features and results. (A) Scheme representing a gene-disease space for three hypothetical diseases. The color-line ovals would enclose the genes (dots) affected by a given disease. The coloured-background circles would mark the altered genes that are specific for a disease, which are the ones we aim to identify: i.e. genes affected by only one of the compared diseases. **(B)** Graph presenting the posterior probability of the top 2000 genes included in the gene ranking for each of the 4 leukemia subtypes and the non-leukemia samples. The genes are selected and ranked based on their posterior probability for each class. Genes with posterior probability over the threshold (> 0.95) can be considered *significant* candidates to mark each disease subtype. **(C)** Lists presenting the top-15 genes in the ranking of each class. The bottom row shows the total number of significant genes in the whole ranking. The shaded area contains the genes selected as the minimum subset to separate the classes. The number of genes selected per class is shown on top. **(D)** Discriminant power plot of the first ranked gene for each leukemia subtype: *VPREB1* (pre-B lymphocyte 1) for ALL; *HOXA9* (homeobox 9) for AML; *TYMS* (thymidylate synthetase) for CLL; and *GJB6* (gap junction beta-6 30kDa protein) for CML. The red numbers indicate the *discriminat power* values assigned to each gene.

to their comparative changes in the global expression profiles. These sizes do not represent the absolute number of genes each disease affects, but rather the genes that are only affected by each disease in the specific contrast. In any case, this phenomenological consideration supports the proposed hypothesis of a gene-disease space, where different diseases affect different number of genes.

After the classification process the minimun subset of genes that allow the best class separation were selected: 9 genes for ALL, 5 for AML, 1 for CLL, and 5 for CML (blue-shaded boxes in **Figure 1C**; detailed information about these genes is included in **Additional File 1**).

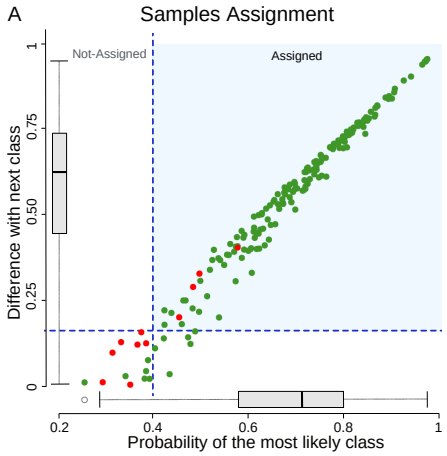
External validation and performance of *geNetClassifier*

Once the classifier for leukemias was built, an external validation was conducted to evaluate the accuracy and performance of the algorithm and to confirm the robustness of the genes selected as markers of the corresponding classes [17].

An external validation consists on querying the classification system with an independent set of samples whose class is *a priori* known. We used a different set of 200 samples of the same five classes (**Figure 2**). Sensitivity, specificity, MCC, global accuracy and global call rate were calculated to evaluate the performance. These statistical parameters were estimated in 10 runs of external validation randomly splitting the available samples.

The external validation could be performed following two different approaches: **(i)** assigning all the samples to their most likely class or (**ii**) leaving doubtful samples as *not-assigned*. (See Methods).

When the *not-assigned* option was selected, the external validation done with 200 leukemia samples provided an average of 4 misclassifications per run (shaded region in **Figure 2A**). All other samples were either correctly assigned or left unclassified (*not-assigned*), resulting in an average global accuracy of 98% and average call rate of 92% (assignment percentage). By contrast, since most samples that would have been incorrectly assigned had a probability under the thresh-



B External Validation

Confusion matrices (Average percentage of 10 runs)

With Not-Assigned

	NotA	ALL	AML	CLL	CML	NoL
ALL	1	99	0	0	0	0
AML	23	0	70	0	1.25	5.75
CLL	0.50	0	0	99.50	0	0
CML	7.75	0	0.50	0	91.75	0
NoL	7.75	0	0	0	1.25	91
Sensitivity		100	90.9	100	99.5	98.6
Specificity		100	99.9	100	99.3	98.4
MCC		1.00	0.94	1.00	0.98	0.95
Call Rate		99	77	99.5	92.3	92.3

Global: Accuracy **98.10**, Call Rate 92.00

All Assigned

	NotA	ALL	AML	CLL	CML	NoL
ALL	0	99.50	0	0	0	0.50
AML	0	0.50	80.75	0.75	6.25	11.75
CLL	0	0	0	100	0	0
CML	0	0	2.25	0.25	96.75	0.75
NoL	0	0	0	0	2.75	97.25
Sensitivity		99.5	80.8	100	96.8	97.3
Specificity		99.9	99.4	99.8	97.8	96.8
MCC		0.99	0.86	0.99	0.93	0.91
Call Rate		100	100	100	100	100

Global: Accuracy **94.85**, Call Rate 100

Figure 2 External validation of geNetClassifier using a large dataset of leukemia samples. (A) Assignment of 200 leukemia samples in one run of external validation. Green dots correspond to samples in which the most likely class is correct and red dots to samples in which it is incorrect. Samples in the blue background area are assigned to the most likely class; samples under any of the two thresholds (dashed blue lines) are considered doubtful samples and can be left as "Not-Assigned". **(B)** Summary of the results of 10 independent external validations. Average confusion matrices (in percentage values) and statistical parameters: sensitivity, specificity, MCC, call rate (calculated per class), global accuracy and global call rate (calculated globally for all the classes). On the right (labelled *All Assigned*) the statistics are calculated assigning all the samples to their most likely class. On the left (*With Not-Assigned*) the statistics are calculated leaving doubtful samples unassigned.

olds (red dots in **Figure 2A**), the accuracy when all samples were forced to be assigned to their most likely class was 94.85%.

In overall, the external validation for the leukemias showed that the best performance –allowing not assignment– was obtained for ALL and CLL (100% sensitivity and specificity, MCC=1.0), while nk-AML presented the lowest values (90.9% sensitivity, 0.94 MCC and 77% call rate). Difficulties in the identification and classification of nk-AMLs were already described in a large-scale international leukemia study where the rate of misclassification for this specific subtype was 11.4% [18]. In conclusion, the classification accuracy rates provided by *geNetClassifier* confirms that the genes sets selected for each class can be good markers of the analysed disease subtypes.

Genes and networks associated to each leukemia subtype

The gene networks produced for each leukemia subtype are presented in **Figure 3**. The plots include the top-30 genes selected for each class as characteristic markers of each leukemia subtype.

Several of these genes have been already reported as functionally associated to these diseases. For example, in the case of ALL, the gene *VPREB1* -that is the first gene in ALL ranking- encodes a protein that belongs to the immunoglobulin superfamily and is expressed selectively at the early stages of B lymphocytes development (i.e. on the surface of pro-B and early pre-B cells). This gene has already been proposed as a useful marker for the detection of normal and malignant human pre-B lymphocytes [19]. Since all ALL samples included in this study correspond to pre-B-ALL without t(9;22), the selection of *VPREB1* seems quite adequate. Another gene selected to mark ALL is *DNTT*. The protein encoded by *DNTT* is expressed in a restricted population of normal and malignant pre-B and pre-T lymphocytes during early differentiation.

In the case of the genes selected for nk-AML, the network shows a cluster of homeobox genes (*HOXA4*, *HOXA5*, *HOXA7*, *HOXA9*, *HOXA10*). The co-expression of these genes detected in the dataset reveals that they are coregulated. *MEIS1* is a transcriptional regulator also included in the homeobox co-expression cluster and selected as one of the genes with best discriminant power for the nk-AML class. Two recent publications have reported that downregulation of *MEIS1* and *HOXA* genes impair proliferation and expansion of acute myeloid leukemia cells [20, 21]. Moreover, *HOXA* has a specific translocation event that has been associated with myeloid leukemogenesis, and overexpression of *HOXA9* has been shown as representitative of nk-AML patients during first diagnosis and if they suffer relapse [22]. These and other reports support the selection of *MEIS1* and *HOXA9* in the gene network that characterizes AML with normal karyotype [23]. Another gene related to AML is *ANGPT1*, that encodes protein angiopoietin 1. Angiopoietins are proteins with important roles in vascular development and angiogenesis which have also been identified as over expressed in bone marrow of AML patients [24].

Finally, the gene network produced for CML includes characteristic genes such as *PRG3*, that encodes for eosinophil major basic protein 2 (MBP2) which is specific of eosinophil granulocytes, a myeloid cell type. Moreover, it has been shown that many molecules essential for tumor cell growth (like polyamines) enter cells via a proteoglycan-dependent pathway that involves PRG3 [25]. All these published reports do not prove that the genes included in the networks for each leukemia subtype are essential for the development of such diseases. However, they give important support to the results and underline the value of the method for creating significant gene sets and gene networks associated to specific disease subtypes.

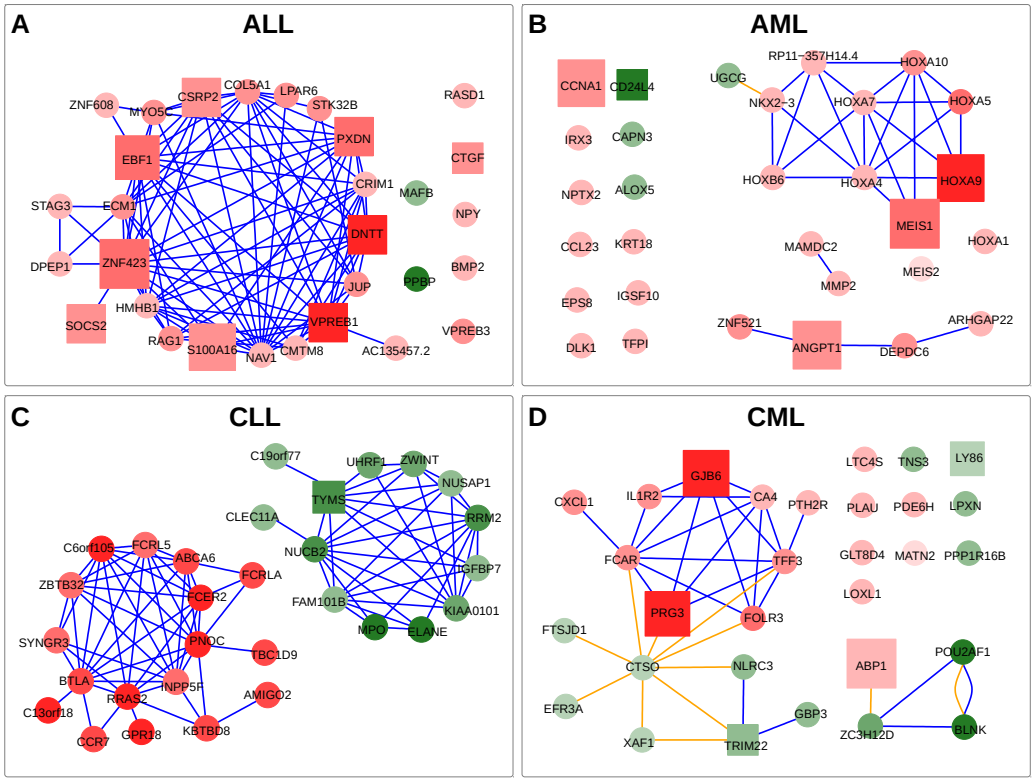


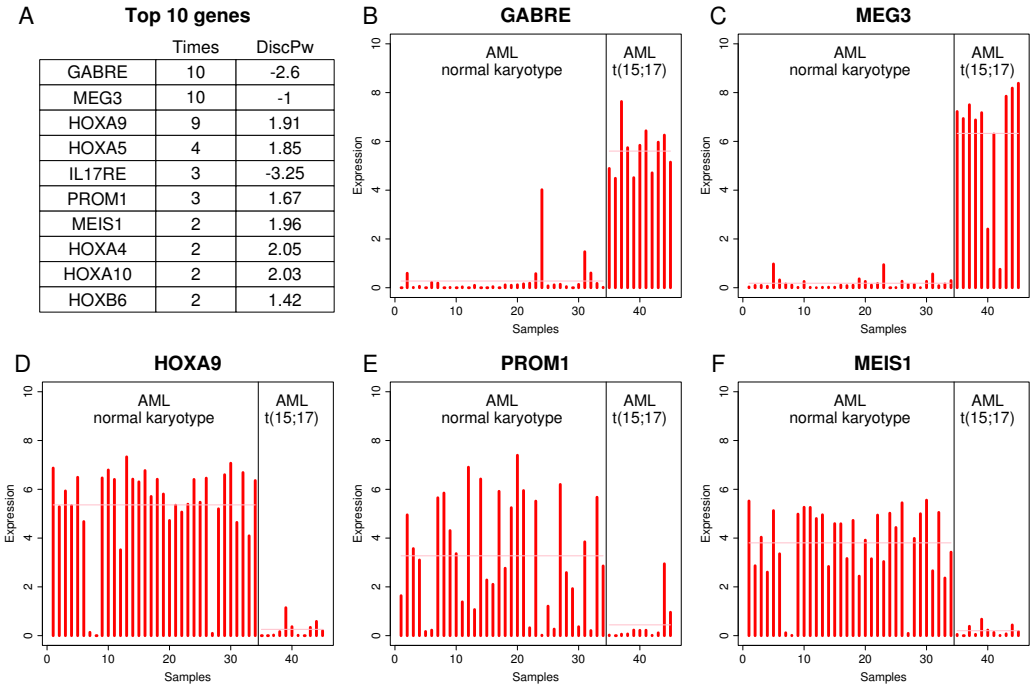
Figure 3 Gene networks produced for each studied leukemia subtype. The networks contain the top-30 ranked genes selected for each class. Genes are coloured in red if they are overexpressed and in green if they are repressed in the given subtype. The intensity of these colours indicate the magnitude of the observed expression difference. Squares correspond to genes that were selected as *minimum subset* to differentiate the classes. The size of these squares is proportional to the *discriminant power* of the gene. The color of the edges indicates the type of relations between the genes: blue meaning co-expression (correlation) and orange mutual information.

Application of *geNetClassifier* to an RNA-Seq dataset

geNetClassifier can be applied to different types of genomic data produced with different platforms. We have also applied it to an RNA-Seq dataset of acute leukemia samples [26] from which we selected 45 samples from patients with two AML subtypes: **(i)** 11 samples of patients with t(15;17) chromosomal translocation characteristic of acute promyelocytic leukemia (APL), and **(ii)** 34 samples of AML patients with normal karyotype and no detected FISH abnormalities (nk-AML). APL is an AML subtype that has good clinical prognosis. Its sensitivity to all-trans retinoic acid (ATRA) allows an efficient treatment unique among leukemias. By contrast, nk-AML is one of the most frequent subtypes of AML (approx. 50%) and usually has a poor clinical prognosis due to the lack of an efficient treatment [26]. Out of these two AML subtypes, nk-AML was also present in the previous microarray dataset analysed. This allows us to investigate the performance of the algorithm studying a common disease subtype in a different context and using a different type of expression data.

geNetClassifier was applied to the RNA-Seq dataset of APLs and nk-AMLs using 8 samples from each class as training samples and then validated with the rest of the samples. We repeated this process 10 times randomly selecting the training samples. The global accuracy obtained in this analysis was 100% with a call rate of 91.38%. The list of genes most frequently selected for classification (**Figure 4A**) included several homeobox genes (HOXA and HOXB) and MEIS1, showing agreement with the results obtained for nk-AML in the microarray analysis. In this way, the expression profiles from these genes in the RNA-Seq dataset are consistent with the results obtained from microarrays, e.g.: genes *HOXA9* and *MEIS1* were down regulated in APL in comparison to nk-AML (**Figure 4D and 4F**). In addition, the network generated for nk-AML selected a set of homeobox genes that form a highly connected co-expression cluster (**Figure 5**). Other genes detected in this analysis, for example MEG3 (**Figure 4C**), showed over-expression in APL versus nk-AML. In fact, it has been reported that MEG3 expression is lost in multiple cancer cell lines of various tissue origins, and it inhibits tumor cell proliferation in vitro. The identification of MEG3 as marker over-expressed in the AML subtype with better prognosis (**Figure 4C**) provides support to the selection of this gene as a discriminant feature between APL and nk-AML.

Finally, to have a better estimation of the global agreement provided by the algorithm in the analysis of the genes assigned to a given disease subtype, we analysed the total overlapping of the genes selected for nk-AML in the arrays dataset and the RNA-Seq dataset. Both platforms included a common set of 16,611 human protein-coding genes. Within this set, the number of significant genes selected for nk-AML were 202 (using posterior probability > 0.95). The RNA-Seq results included 95 of these genes (considering the 10



	Times	DiscPw
GABRE	10	-2.6
MEG3	10	-1
HOXA9	9	1.91
HOXA5	4	1.85
IL17RE	3	-3.25
PROM1	3	1.67
MEIS1	2	1.96
HOXA4	2	2.05
HOXA10	2	2.03
HOXB6	2	1.42

Figure 4 Genes selected for the RNA-seq dataset of acute leukemia. (A) The table shows the genes most frequently selected to distinguish between normal karyoptype AML (nk-AML) and AML with t(15;17) (APL). *geNetClassifier* was run ten times with different combinations of samples. The table includes the mean discriminant power within these runs and the number of times that each gene is included in the *minimum subset*. **(B-F)** Expression profiles corresponding to some of the top selected genes: *GABRE*, *MEG3*, *HOXA9*, *PROM1* and *MEIS1*. Each red bar corresponds to the expression signal ($log_2(RPKM+1)$) in one sample.

runs indicated above), and 76 of them were selected in more than three runs. An overlap of 95 genes corresponds to an odds ratio of 2.17 and to an enrichment p-value < 0.000001 (using hypergeometric test). Therefore, it can be said that the consistency of the method to select genes that mark a specific disease subtype is high.

Comparison of *geNetClassifier* with other methods

Finally, we have evaluated the performance of *geNetClassifier* relative to other gene selection and classification methodologies. We compared *geNetClassifier* with four machine learning methods for feature selection using CMA package [27], which provides a comprehensive collection of various microarray-based classification algorithms (see **Additional File 2**). We have also evaluated the classification procedure of *geNetClassifier* using svb-IMPROVER contest platform [28], which includes a Diagnostic Signature Challenge with several datasets to assess and verify computational approaches that classify clinical samples based on transcriptomics data (see **Additional File 3**). In both cases, the performance of *geNetClassifier* algorithm is within the best methods. However, it should be noted that we could only compare the classification and gene selection procedures. The other features included in our package could not be found integrated in other methods.

Conclusions

Biological annotation of the genes selected and the networks built to mark and separate different pathological states confirm the value of using *geNetClassifier* to analyse multiple disease subtypes based on genome-wide expression profiles. The tool is provided open access in Bioconductor to facilitate the type of studies illustrated in this report.

As a general conclusion, the results using *geNetClassifier* showed a robust selection of gene markers for characterizing disease subtypes and allowed the construction of specific and weighted gene networks associated to each disease subtype. The method can be applied to data derived from different types of technologies (such as microarrays or RNA-Seq) and it is designed to analyse datasets with multiple categories of samples.

Methods

Implementation and availability

geNetClassifier has been developed as an R package following Bioconductor (BioC) standards and technical requisites (www.bioconductor.org). It has attained BioC package submission process and package guidelines to be included in BioC software release. It is freely available, open source and open access. The package includes help pages with usage examples for each specific function. Together with the package, we have written a *vignette* including a detailed tutorial to use the algorithm (**Additional File 4**).

Microarray dataset

The microarray leukemia dataset is a subset of 250 samples collected from the Microarray Innovations in Leukemia (MILE) study [18] available at Gene Expression Omnibus database (www.ncbi.nlm.nih.gov/geo/)

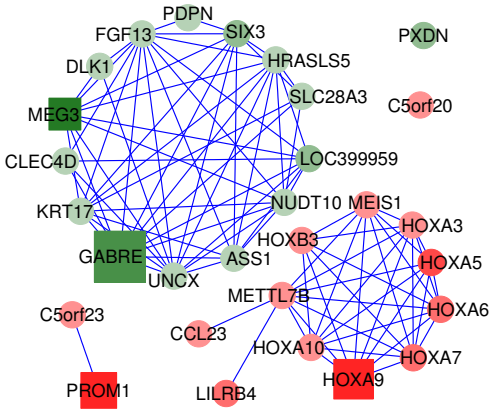


Figure 5 Gene network obtained for AML with the RNA-seq dataset. The network contains the top-30 ranked genes selected after running *geNetClasifier* to analyse the RNA-Seq expression data of normal karyoptype AML (nk-AML) versus AML with t(15;17) (APL) samples. The network shows two clear clusters: one including genes that are up-regulated in nk-AML and another with down-regulated genes. The red cluster includes many homeobox (HOX) genes highly correlated. These genes are characteristic of nk-AMLs and show good agreement with the results obtained with microarrays in spite of being two totally independent datasets.

under series accession number GSE13159. The genome-wide expression signal corresponding to these samples was measured using *Affymetrix* Human Genome U133 Plus 2.0 microarrays. The samples correspond to mononuclear cells isolated by Ficoll density centrifugation from bone marrow of untreated patients with: **(1)** *Acute Lymphoblastic Leukemia* (ALL) subtype *childhood* or *precursor B-cell* (c-ALL/pre-B-ALL) without translocation t(9;22); **(2)** *Acute Myeloid Leukemia* (AML) subtype *normal karyotype* (nk); **(3)** *Chronic Lymphocytic Leukemia* (CLL) subtype *B-cell*; **(4)** *Chronic Myeloid Leukemia* (CML); **(5)** *Non-leukemia* and healthy bone marrow (NoL).

The microarrays were normalized using the algorithm Robust Multi-Array Average (RMA) [29] and applying a gene-centric redefinition of the probes from the *Affymetrix* arrays to Ensembl genes (Ensembl IDs ENSG). This alternative Chip Definition File (CDF) with complete unambiguous mapping of microarray probes to genes is available at GATExplorer (http://bioinfow.dep.usal.es/xgate/) [30].

RNA-Seq dataset

The leukemia dataset analysed with RNA-sequencing corresponds to a subset of samples collected by the Cancer Genome Atlas (TCGA) [26] available at the TCGA data portal (https://tcga-data.nci.nih.gov/). These RNA-Seq data correspond to samples obtained from bone marrow aspirate of patients with AMLs of *de novo* diagnosis. Out of the available samples in TCGA, we selected 45 samples of the following subtypes: **(1)** AML patients with translocation t(15;17) (also called *Acute Promyelocytic Leukemia*, APL) (11 samples); and **(2)** AML patients with normal karyotype and no detected FISH abnormalities (nk-AML) (34 samples). The preprocessed RNA-Seq expression data matrices containing the *reads per kilobase per million mapped reads* (RPKM) were downloaded from the TCGA data portal and were log2 transformed ($log_2(RPKM+1)$) prior to be analysed with *geNetClassifier*.

Statistical methods and algorithm procedures

Gene ranking: To create the gene ranking, *geNetClassifier* uses the function *emfit*, a Parametric Empirical Bayes method, included in package *EBarrays* [9]. This method implements an expectation-maximization (EM) algorithm for gene expression mixture models, which compares the patterns of differential expression across multiple conditions and provides a *posterior probability*. The posterior probability is calculated for each gene-class pair with a *One-versus-Rest* contrast: comparing the samples of one class *versus* all the other samples. In this way, the posterior probability represents how much each gene differentiates a class from the other classes (being 1 the best value, and 0 the worst). The ranking is built, in a first step, by ordering the genes decreasingly by their posterior probability for each class. To resolve ties, the algorithm uses the value of the difference between the signal expression mean for each gene in the given class and the mean in the closest class. In a second step, the ranking procedure assigns each gene to the class in which it has the best ranking. As a result of this process, even if a gene is found associated to several classes during the expression analysis, it will only be on the ranking its best class. In addition, genes that do not show any significant difference between classes are filtered out before building the ranking. Finally, the set of genes considered *significant* in the ranking of each class is determined by a threshold of the posterior probability, which by default is set up to be greater than 0.95 .

Classifier: The classifier included in the algorithm is a multi-class *Support Vector Machine* (SVM) available in R package *e1071* [11]. This package provides a linear kernel implementation that allows the classification of multiple classes by using a *One-versus-One* (OvO) approach, in which all the binary classifications are fitted and the correct class is found based on a voting system.

Gene selection: The gene selection is done through a *wrapper forward* selection scheme based on *8-fold cross-validation*. Each cross-validation iteration starts with the first ranked gene of each class: it trains a temporary internal classifier with these genes, and evaluates its performance. One more gene is added in each step to those classes for which a 'perfect prediction' is not achieved (i.e. in case not all samples are correctly identified). The genes are taken in order from the *gene ranking* of each class until reaching zero error or the maximum number of genes allowed (determined by the arguments *maxGenesTrain* and *continueZeroError*). The error for each of the classifiers and the number of genes used to construct them are saved. Once the cross-validation loop is finished, it selects the minimum number of genes per class which produced the classifier with minimum error. To achieve the best stability in the number of selected genes, the cross-validation is repeated with new samplings as many times as indicated by the user (6 times by default). In each of these iterations, the minor number of genes that provided the smallest error is selected. The final selection is done based on the genes selected in each of the iterations. For each class, the top ranked genes

are selected by taking the 'highest number' of genes selected in the cross-validaton iterations, but excluding possible 'outlier numbers' (i.e. selecting trimmed values).

Discriminant power: The *discriminant power* is a parameter calculated based on the *Lagrange coefficients* (alpha) of the *support vectors* for all the genes selected for the classification. Since the multi-class SVM algorithm is a *One-versus-One* implementation, it produces a set of *support vectors* for each binary comparison between classes. For each gene, the *Lagrange coefficients* of all the *support vectors* for each class are added up to give a value per class (represented as piled up bars in **Figure 1D**). The *discriminant power* is then calculated as the difference between the largest value and the closest one (i.e. the distance marked by two red lines in the plots in **Figure 1D**).

Assignment conditions: The whole tool *geNetClassifier* is built considering an *expert decision system* approach, because once the classifier is build it keeps open the possibility of '*do not assign*' when it is not sure about the class of a query sample. To make the assignment decision the probability to assign a sample to a given class should be at least double than the *random probability*, and the difference with the second most likely class should be higher than 0.8 times the *random probability*. If these conditions are not met, the sample is left as *Not-Assigned* (NA). These probability thresholds for assignment conditions are set up by default, but they can be changed by the user.

List of abbreviations used
MCC: Matthews Correlation Coefficient
t(9;22): translocation between chromosomes 9 and 22

Competing interests
The authors declare that they have no competing interests.

Author's contributions
SA carried out the development of the tool, performed its validation with several expression datasets and its statistical analysis, carried out the implementation of the R package and drafted the manuscript. CF participated in the design of the study, carried out the development of the algorithm and the trials of the methods included. CD participated in the application of the package. BR participated in the validation of the methods and helped in writing the manuscript. FJCL participated in the validation and application of the algorithm with different datasets. JMHR participated in the selection and analyses of the diseases and provided the clinical patient samples. JDLR conceived the study, directed the design and development of the algorithm and wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements
We acknowledge the funding and grants provided to Dr. J. De Las Rivas group by the Local Government, "Junta de Castilla y Leon" (JCyL, Valladolid, Spain, grants number CSI07A09 and BIO/SA68/13) and the Spanish Government, "Ministerio de Economia y Competitividad" (MINECO, ISCiii, Madrid, Spain, grants number PI09/00843 and PI12/00624). We also acknowledge PhD research grants given to SA, CR and BR by JCyL ("Ayudas a la Contratación de Personal Investigador") provided with the support of the European Social Fund (ESF). We also want to acknowledge Celgene-CITRE (citre@celgene.com) for the support provided to this research (Project Celegene-FICUS 2013). Since performing the work described, CF has become an employee of Celgene Research S.L., part of the Celgene Corporation.

Author details
[1]Cancer Research Center (IMBCC, CSIC/USAL/IBSAL), Campus Miguel de Unamuno s/n, Salamanca, 37007, Spain. [2] Hematology Department, Hospital Universitario de Salamanca (HUS/IBSAL/USAL), Paseo San Vicente 58-182, Salamanca, 37007, Spain. [3] (present address) Celgene Institute for Translational Research Europe (CITRE), Parque Científico y Tecnológico Cartuja 93, c/ Isaac Newton 4, Sevilla, 41092, Spain.

References

1. Culhane, A.C., Schröder, M.S., Sultana, R., Picard, S.C., Martinelli, E.N., Kelly, C., Haibe-Kains, B., Kapushesky, M., St Pierre, A.-A., Flahive, W., Picard, K.C., Gusenleitner, D., Papenhausen, G., O'Connor, N., Correll, M., Quackenbush, J.: GeneSigDB: a manually curated database and resource for analysis of gene expression signatures. Nucleic Acids Res **40**(Database issue), 1060–1066 (2012)
2. Venet, D., Dumont, J.E., Detours, V.: Most Random Gene Expression Signatures Are Significantly Associated with Breast Cancer Outcome. PLoS Comput Biol. **7**, 1002240 (2011)
3. De Ridder, D., De Ridder, J., Reinders, M.J.T.: Pattern recognition in bioinformatics. Brief Bioinform **14**, 633–647 (2013)
4. Larranaga, P.: Machine learning in bioinformatics. Brief Bioinform **7**, 86–112 (2006)
5. Cruz, J., Wishart, D.: Applications of machine learning in cancer prediction and prognosis. Cancer Inform **2**, 59–77 (2006)
6. De Las Rivas, J., Fontanillo, C.: Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. PLoS Comput Biol **6**, 1000807 (2010)
7. Zhang, K., Pirooznia, M., Arabnia, H.R., Yang, J.Y., Wang, L., Luo, Z., Deng, Y.: Genomic signatures and gene networking: challenges and promises. BMC Genomics **12**(Suppl 5), 1 (2011)
8. in Man. OMIM, O.M.I.: http://omim.org
9. Yuan, M., Newton, M., Sarkar, D., Kendziorski, C.: EBarrays: Unified Approach for Simultaneous Gene Clustering and Differential Expression Identification. (2007). http://www.bioconductor.org/packages/release/bioc/html/EBarrays.html
10. Kendziorski, C.M., Newton, M.A., Lan, H., Gould, M.N.: On parametric empirical bayes methods for comparing multiple groups using replicated gene expression profiles. Stat Med **22**, 3899–3914 (2003)
11. Meyer, D.: Support Vector Machines. The Interface to Libsvm in Package E1071. (2001). http://cran.r-project.org/web/packages/e1071/
12. Statnikov, A., Aliferis, C., Tsamardinos, I., Hardin, D., Levy, S.: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics **21**, 631–643 (2005)
13. Statnikov, A., Wang, L., Aliferis, C.F.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics **9** (2008)
14. Pirooznia, M., Yang, J.Y., Yang, M.Q., Deng, Y.: A comparative study of different machine learning methods on microarray gene expression data. BMC Genomics **9** (2008)
15. Kohavi, A., Johnb, G.: Wrappers for feature subset selection. Artificial intelligence **97**, 273–324 (1997)
16. Meyer, P.E., Lafitte, F., Bontempi, G.: minet: A r/bioconductor package for inferring large transcriptional networks using mutual information. BMC Bioinformatics **9**, 461 (2008)
17. Ambroise, C., McLachlan, G.: Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Natl Acad Sci U S A **99**, 6562–6566 (2002)
18. Haferlach, T., Kohlmann, A., Wieczorek, L., Basso, G., Kronnie, G.T., Béné, M.-C., De Vos, J., Hernández, J.M., Hofmann, W.-K., Mills, K.I., Gilkes, A., Chiaretti, S., Shurtleff, S.a., Kipps, T.J., Rassenti, L.Z., Yeoh, A.E., Papenhausen, P.R., Liu, W.-M., Williams, P.M., Foà, R.: Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the International Microarray Innovations in Leukemia Study Group. J Clin Oncol **28**, 2529–2537 (2010)
19. Bauer, S.R., Kudo, A., Melchers, F.: Structure and pre-B lymphocyte restricted expression of the VpreB gene in humans and conservation of its structure in other mammalian species. EMBO J **7**, 111–116 (1988)
20. Orlovsky, K., Kalinkovich, A., Rozovskaia, T., Shezen, E., Itkin, T., Alder, H., Ozer, H.G., Carramusa, L., Avigdor, A., Volinia, S., Buchberg, A., Mazo, A., Kollet, O., Largman, C., Croce, C.M., Nakamura, T., Lapidot, T., Canaani, E.: Down-regulation of homeobox genes MEIS1 and HOXA in MLL-rearranged acute leukemia impairs engraftment and reduces proliferation. Proc Natl Acad Sci USA **108**, 7956–7961 (2011)
21. Wolthuis, C., Han, L., Verkaik-Schakel, R., van Gosliga, D., Kluin, P., Vellenga, E., Schuringa, J., Huls, G.: Downregulation of MEIS1 impairs long-term expansion of CD34+ NPM1-mutated acute myeloid leukemia cells. Leukemia **26**, 848–853 (2012)
22. Grubach, L., Juhl-Christensen, C., Rethmeier, A., Olesen, L.H., Aggerholm, A., Hokland, P., Østergaard, M.: Gene expression profiling of polycomb, hox and meis genes in patients with acute myeloid leukaemia. Eur J Haematol **81**, 112–122 (2008)
23. Bullinger, L., Döhner, K., Bair, E., Fröhling, S., Schlenk, R.F., Tibshirani, R., Ph, D., Döhner, H., Pollack, J.R.: Use of Gene-Expression Profiling to Identify Prognostic Subclasses in Adult Acute Myeloid Leukemia. N Engl J Med **350**, 1605–1616 (2004)
24. Schliemann, C., Bieker, R., Padro, T., Kessler, T., Hintelmann, H., Buchner, T., Berdel, W., Mesters, R.: Expression of angiopoietins and their receptor tie2 in the bone marrow of patients with acute myeloid leukemia. Haematologica **91**, 1203–1211 (2006)
25. Mani, K., Sandgren, S., Lilja, J., Cheng, F., Svensson, K., Persson, L., Belting, M.: HIV-Tat protein transduction domain specifically attenuates growth of polyamine deprived tumor cells. Mol Cancer Ther **6**, 782–788 (2007)
26. Ley, T.J., Miller, C., Ding, L., Raphael, B.J., Mungall, A.J., *et al.*:

Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. N Engl J Med **368**, 2059–2074 (2013)
27. Slawski, M., Daumer, M., Boulesteix, A.: CMA: a comprehensive Bioconductor package for supervised classification with high dimensional data. BMC Bioinformatics **9**, 439 (2008)
28. Rhrissorrakrai, K., Rice, J.J., Boue, S., Talikka, M., Bilal, E., Martin, F., Meyer, P., Norel, R., Xiang, Y., Stolovitzky, G., Hoeng, J., Peitsch, M.C.: sbv IMPROVER Diagnostic Signature Challenge. Systems Biomedicine **1**, 196–207 (2013)
29. Irizarry, R.a., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U., Speed, T.P.: Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics **4**, 249–264 (2003)
30. Risueño, A., Fontanillo, C., Dinger, M., De Las Rivas, J.: GATExplorer: genomic and transcriptomic explorer; mapping expression probes to gene loci, transcripts, exons and ncRNAs. BMC Bioinformatics **11**, 221 (2010)

Additional Files

Additional file 1: Table S1. Table with data and information about the genes selected by *geNetClassifier* in the analyses of the leukemia microarrays dataset (classes: four leukemia subtypes and control class NoL): **Class**: The category a gene has been assigned to. **Rank**: Position of the gene within the list of genes ranked by significance assigned to a disease. **Posterior probability**: Probability value given by the expectation-maximization algorithm to each gene. This value is used to establish the ranking. In this result all values were very close to 1 (with more than 10 significant digits). Ties are further ranked based on the differential expression. **Expression**: Difference between the mean expression of the gene within its class and the mean expression in the other classes. UP or DOWN indicates whether the gene is overexpressed or repressed in its class compared to the other classes. **Discriminant Power**: Parameter calculated based on the *Lagrange* coefficients of the support vectors of the classifier. Represents the weight that the classifier gives to each gene to differentiate a given class. **Redundancy**: If TRUE, the gene has a high correlation or mutual information with other genes in the list. The threshold to consider a gene redundant can be set through the arguments (by default: correlationsThreshold=0.8 and interactionsThreshold=0.5). **Chosen for classification**: Number of times the gene was chosen for classification (as part of the minimum required subset) in the 5 internal cross-validation loops. Rank mean and rank standard deviation (SD) of the gene in these classifiers. **Cross-validation**: Mean and standard deviation of the rank that the gene has obtained in *geNetClassifier*'s internal cross-validation, including the times it was not selected for classification.

Additional file 2: Table S2. Comparison of *geNetClassifier* gene selection procedure with four other machine learning methods for gene selection (i.e. feature selection): Limma, F-test, Boosting and Random Forest.
The comparison has been done on the dataset of 250 leukemia samples, using R/Bioc package CMA that provides a comprehensive collection of various microarray-based classification algorithms [27].

Additional file 3: File S3. Evaluation of the performance of *geNetClassifier* classification procedure in the sbv-IMPROVER contest platform (https://sbvimprover.com/), which includes a Diagnostic Signature Challenge to assess and verify computational approaches that classify clinical samples based on transcriptomics data [28].
The performance has been evaluated using the dataset from IMPROVER that includes four classes corresponding to lung cancer subtypes.

Additional file 4: File S4. *geNetClassifier vignette*
Vignette including a tutorial with executable examples and description of all the methods. This *vignette* is available in Bioconductor:
http://www.bioconductor.org/packages/release/bioc/vignettes/geNetClassifier/inst/doc/geNetClassifier-vignette.pdf

Additional file 1: File S1.

Class	Gene	Rank	Posterior Probability	Expression		Discriminant Power		Redundant	Chosen for classification			Cross-Validation	
				Mean difference	UP/DW	Value	Class		Times	Rank mean	Rank SD	Rank mean	Rank SD
ALL	VPREB1	1	1	6.33	UP	9.41	ALL	0	5	1	0	1	0
	ZNF423	2	1	5.09	UP	13.24	ALL	1	4	2.75	1.5	3	1.41
	DNTT	3	1	6.89	UP	8.97	ALL	1	5	2.8	0.45	2.8	0.45
	EBF1	4	1	5.41	UP	10.51	ALL	1	2	3	1.41	3.8	1.48
	PXDN	5	1	5.03	UP	8.65	ALL	1	2	5	1.41	5.2	0.84
	S100A16	6	1	4.34	UP	12.38	ALL	1	2	5.5	0.71	5.4	1.14
	CSRP2	7	1	4.04	UP	8.78	ALL	1	1	7	0	7.8	1.3
	SOCS2	8	1	4.53	UP	8.69	ALL	0	1	8	0	10.8	3.27
	CTGF	9	1	3.61	UP	5.55	ALL	0	0	NA	0	14.8	10.03
AML	HOXA9	1	1	4.43	UP	8.01	AML	0	5	1.2	0.45	1.2	0.45
	MEIS1	2	1	3.27	UP	10.31	AML	1	4	2.5	1	3	1.41
	CD24L4	3	1	-4.49	DOWN	-5.73	AML	0	2	3.5	3.54	3.8	2.17
	ANGPT1	4	1	2.74	UP	9.21	AML	0	2	4.5	0.71	4.8	1.3
	CCNA1	5	1	2.55	UP	8.24	AML	0	4	4	1.83	5.4	3.51
CLL	TYMS	1	1	-5.51	DOWN	-10.07	CLL	0	4	1.25	0.5	1.8	1.3
CML	GJB6	1	1	5.25	UP	4.94	CML	0	5	2.2	1.79	2.2	1.79
	PRG3	2	1	4.97	UP	4.09	CML	1	3	3	1	92.4	166.45
	LY86	3	1	-2.2	DOWN	-5.56	CML	0	1	2	0	39.6	21.9
	ABP1	4	1	2.51	UP	8.47	CML	0	4	3	2.16	5	4.85
	TRIM22	5	1	-2.67	DOWN	-9.05	CML	0	1	5	0	35.8	18.27
NOL	FGF13	1	1	2.69	UP	3.78	NoL	0	5	1.2	0.45	1.2	0.45
	NMU	2	1	1.96	UP	4.1	NoL	0	2	2.5	0.71	9	6.44
	SMPDL3A	3	1	1.95	UP	5.07	NoL	0	3	7	3.46	13.8	10.83
	KLRB1	4	1	2.23	UP	3.39	NoL	0	2	6.5	4.95	22.2	16.51
	RNF182	5	1	1.84	UP	1.06	NoL	0	3	2.33	1.53	5.6	4.72
	RFESD	6	1	2.36	UP	2.94	NoL	0	4	4.25	1.5	5.8	3.7

Additional file 2: File S2.

Comparison of five multi-class feature selection (i.e. "gene selection") methods. (Done using svmCMA function from CMA R package)

Data: expression microarrays from 250 leukemia samples. 50 samples are used for training (10 per class); 200 for testing as external validation (40 per class).

The classification is done always using SVM. To be comparable with the other methods, geNetClassifier is forced to assign all samples to a class.

10 runs

Accuracy	CallRate
94.0	100
92.0	100
92.5	100
89.5	100
97.5	100
91.5	100
89.5	100
95.5	100
94.0	100
89.5	100

SUM of the 10 confusion matrices for 10 runs: METHOD Boosting

Always Assign	ALL	AML	CLL	CML	NoL	Sensitivity	Specificity	MCC	CallRate
ALL	394	12	0	2	0	96.651	99.627	96.94	100
AML	4	332	0	4	4	96.744	95.948	87.179	100
CLL	0	4	400	0	0	99.024	100	99.384	100
CML	1	13	0	358	29	89.343	97.384	86.752	100
NoL	1	39	0	36	367	83.285	97.895	84.003	100

Global Accuracy	Global CallRate
92.55	100

10 runs

Accuracy	CallRate
92.0	100
95.5	100
99.5	100
96.0	100
95.5	100
92.5	100
95.0	100
93.5	100
91.5	100
97.0	100

SUM of the 10 confusion matrices for 10 runs: METHOD Limma

Always Assign	ALL	AML	CLL	CML	NoL	Sensitivity	Specificity	MCC	CallRate
ALL	395	0	0	0	0	100	99.692	99.215	100
AML	0	376	0	2	0	99.443	98.538	95.883	100
CLL	0	0	400	0	0	100	100	100	100
CML	1	4	0	358	33	91.087	97.420	87.726	100
NoL	4	20	0	40	367	85.708	97.940	85.585	100

Global Accuracy	Global CallRate
94.8	100

10 runs

Accuracy	CallRate
98.5	100
95.5	100
98.0	100
96.0	100
92.5	100
97.5	100
97.0	100
98.0	100
97.5	100
96.0	100

SUM of the 10 confusion matrices for 10 runs: METHOD Random Forest

Always Assign	ALL	AML	CLL	CML	NoL	Sensitivity	Specificity	MCC	CallRate
ALL	398	0	0	0	0	100	99.876	99.686	100
AML	0	376	0	2	1	99.232	98.528	95.752	100
CLL	0	2	400	0	0	99.512	100	99.692	100
CML	0	3	0	380	20	94.531	98.769	93.389	100
NoL	2	19	0	18	379	91.132	98.688	91.012	100

Global Accuracy	Global CallRate
96.65	100

10 runs

Accuracy	CallRate
96.0	100
96.5	100
96.0	100
96.0	100
94.5	100
92.5	100
94.5	100
92.5	100
93.5	100
97.0	100

SUM of the 10 confusion matrices for 10 runs: METHOD F-test

Always Assign	ALL	AML	CLL	CML	NoL	Sensitivity	Specificity	MCC	CallRate
ALL	395	1	0	0	0	99.756	99.691	99.061	100
AML	0	381	1	0	7	98.122	98.836	95.816	100
CLL	0	0	399	0	0	100	99.938	99.843	100
CML	0	4	0	351	21	93.528	97.006	88.291	100
NoL	5	14	0	49	372	84.880	98.215	85.842	100

Global Accuracy	Global CallRate
94.9	100

10 runs

Accuracy	CallRate
95.5	100
93.0	100
96.0	100
96.0	100
94.5	100
96.0	100
97.0	100
94.5	100
90.5	100
95.5	100

SUM of the 10 confusion matrices for 10 runs: METHOD geNetClassifier (for this comparison forced to "always assign")

Always Assign	ALL	AML	CLL	CML	NoL	Sensitivity	Specificity	MCC	CallRate
ALL	398	0	0	0	2	99.500	99.880	99.380	100
AML	2	323	3	25	47	80.750	99.440	86.200	100
CLL	0	0	400	0	0	100	99.750	99.380	100
CML	0	9	1	387	3	96.750	97.750	92.560	100
NoL	0	0	0	11	389	97.250	96.750	90.690	100

Global Accuracy	Global CallRate
94.85	100

Additional file 3: File S3.

Evaluation of the performance of ***geNetClassifier*** classification procedure in the **sbv-IMPROVER** contest platform (https://sbvimprover.com/challenge-1), which includes a **Diagnostic Signature Challenge** to assess and verify computational approaches that classify clinical samples based on transcriptomics data.

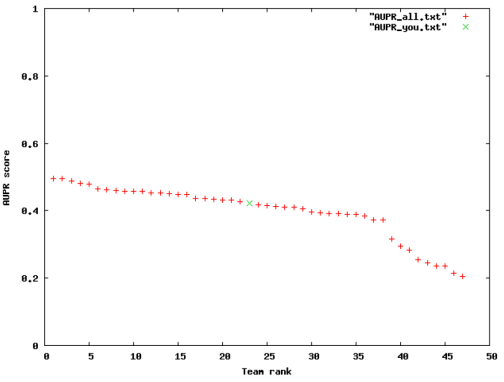
The performance of the algorithm ***geNetClassifier*** has been evaluated using a data-set that has multiple classes (a data-set of **lung cancer** included in **IMPROVER**). We show below the results corresponding to the performance measured with three parameters: **(i) AUPR**, that computes the precision-recall curve for each class, from which the Area Under the Precision-Recall curve is extracted (Precision is a measure of specificity whereas Recall is a measure of completeness); **(ii) BCM**, Belief Confusion Matrix, that is a matrix whose element {i,j} is the average confidence that a sample belonging to class i is in class j (Each prediction has its own belief confusion matrix. The perfect belief confusion matrix is the identity matrix); **(iii) CCEM**, Correct Class Enrichment Metric, that is computed adding the confidence of the samples whose classes were correctly predicted and subtract the confidence of the subjects whose classes were incorrectly predicted (In other words, this is a measure of enrichment of the correctly classified samples. The final value is normalized to be between 0 and 1).

These parameters indicate, as shown in the tables below, that ***geNetClassifier*** is within the best methods, performing as the **6th best** out of 47 different methods submitted to the **Diagnostic Signature Challenge** when it is applied using the option of "**not-assignment**"; and as the **7th best** in the rank of 47 methods when it is used forced to assign always a query sample to a class ("**all assigned**").

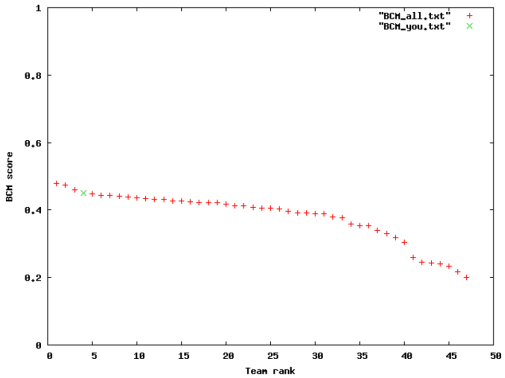
Plots that present the results of AUPR, BCM and CCEM corresponding to the performance of ***geNetClassifier*** using the option of "**not-assignment**".

The RESULTS TABLE placed after these plots presents the values of these parameters and the rank for the top-15 methods.

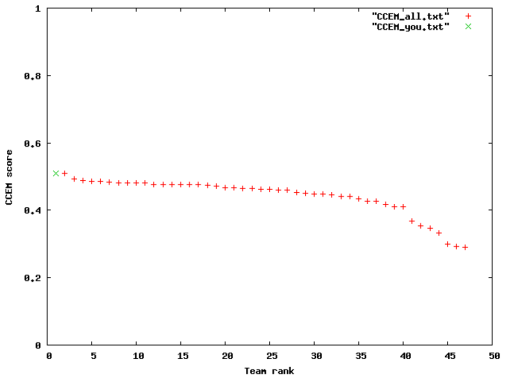
AUPR



BCM



CCEM



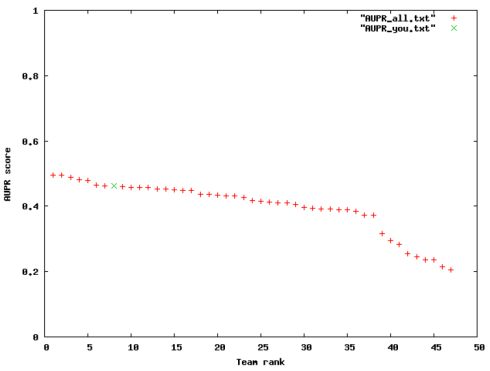
RESULTS TABLE

Team	AUPR	BCM	CCEM	Rank-sum	Rank
Team036	0.458	0.479	0.509	13	1
Team221	0.454	0.459	0.492	18	2
Team114	0.464	0.443	0.483	20	3
Team063	0.489	0.427	0.489	21	4
Team161	0.496	0.431	0.481	23	5
you	0.421	0.452	0.510	28	6
Team273	0.462	0.431	0.480	29	7
Team227	0.428	0.474	0.480	35	8
Team080	0.447	0.423	0.482	41	9
Team187	0.461	0.440	0.459	43	10
Team245	0.480	0.403	0.477	43	10
Team122	0.481	0.413	0.468	45	12
Team290	0.458	0.408	0.476	45	12
Team132	0.448	0.392	0.487	48	14
Team297	0.496	0.378	0.476	49	15

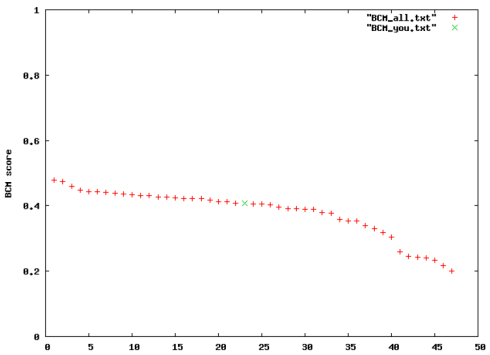
Plots that present the results of AUPR, BCM and CCEM corresponding to the performance of ***geNetClassifier*** using the option of "**all assigned**".

The RESULTS TABLE placed after these plots presents the values of these parameters and the rank for the top-15 methods.

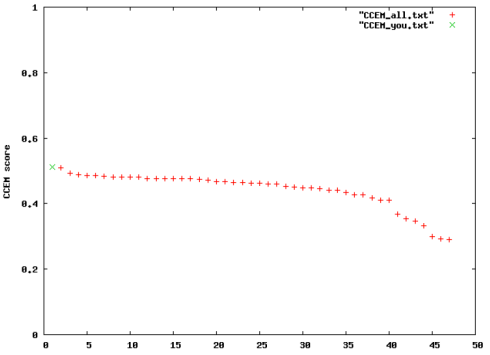
AUPR



BCM



CCEM



RESULTS TABLE

Team	AUPR	BCM	CCEM	Rank-sum	Rank
Team036	0.458	0.479	0.509	14	1
Team221	0.454	0.459	0.492	19	2
Team114	0.464	0.443	0.483	19	2
Team063	0.489	0.427	0.489	20	4
Team161	0.496	0.431	0.481	22	5
Team273	0.462	0.431	0.480	28	6
you	0.462	0.408	0.511	32	7
Team227	0.428	0.474	0.480	36	8
Team080	0.447	0.423	0.482	41	9
Team187	0.461	0.440	0.459	43	10
Team245	0.480	0.403	0.477	43	10
Team122	0.481	0.413	0.468	44	12
Team290	0.458	0.408	0.476	45	13
Team297	0.496	0.378	0.476	49	14
Team132	0.448	0.392	0.487	49	14

geNetClassifier
classify multiple diseases and build associated gene networks using gene expression profiles

Sara Aibar, Celia Fontanillo, Conrad Droste, and Javier De Las Rivas

Bioinformatics and Functional Genomics Group
Centro de Investigacion del Cancer (CiC-IBMCC, CSIC/USAL)
Salamanca - Spain

October 24, 2014

Version: 1.6

Contents

1 Introduction to *geNetClassifier*

geNetClassifier is an algorithm designed to build transparent classifiers and the associated gene networks based on genome-wide expression data.

geNetClassifier() is also the name of the main function in the package. This function takes as input the *expressionSet* or expression matrix of the studied samples and the classes the samples belong to (i.e. the diseases or disease subtypes). Once the data are analyzed, geNetClassifier() provides: **(i)** ranked gene sets (or gene signatures) that identify each class; **(ii)** a multiple-class classifier; and **(iii)** gene networks associated to each class.

- Gene ranking: The genes, probesets, or any other variables that are input in the *expressionSet* are considered *features* for the classification. These features are analyzed by *geNetClassifier*, and ranked according to the class they best identify, in order to select the optimum set for training the classifier. This ranking is returned by *geNetClassifier()* as well as the parameters calculated for gene selection.

- Classifier: *geNetClassifier()* also returns a multi-class SVM-based classifier, which can be queried later on; the genes (features) chosen for classification; their discriminant power (a parameter that measures the importance that the classifier internally gives to each gene); and, optionally, the classifier's generalization error and statistics about the selected genes.

- Network: The mutual-information (interactions) and the co-expression (correlations) between the genes are also calculated and analyzed by the algorithm. These allow to estimate the degree of association between the variables and they are used to generate a gene network for each class. These networks can be plotted, providing a integrated overview of the genes that characterized each disease (i.e. each class).

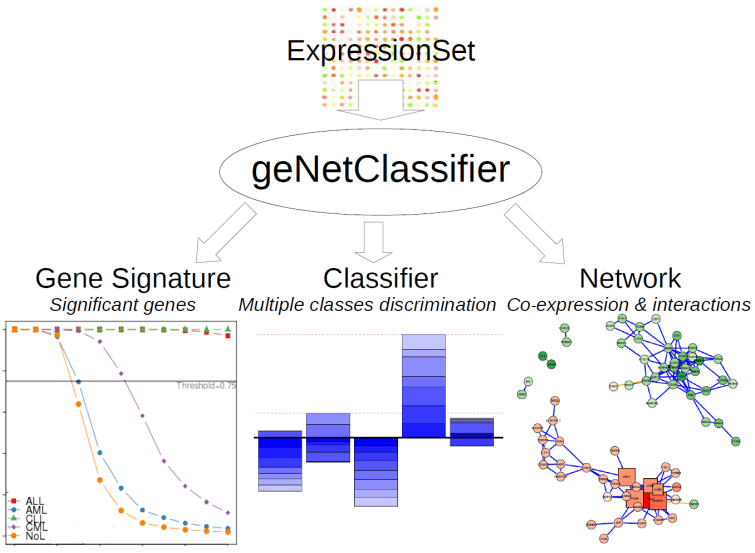


Figure 1. Taking an *expressionSet* as input, *geNetClassifier()* returns a gene signature for each class, a classifier to discriminate the classes, and gene networks associated to each class. The package also includes several analytic and visualizing tools to explore these results.

Examples of use

The algorithm shows a robust performance applied to patient-based gene expression datasets that study disease subtypes or disease classes. In this vignette, we show its performance for a leukemia dataset that includes 60 microarray samples from bone marrow of patients with four major leukemia subtypes (ALL, AML, CLL, CML) and no-leukemia controls (NoL). The results outperform a previously published classification analysis of these data [1].

The method is designed to be applied to the analysis and classification of different disease subtypes. Therefore, in the R package and this vignette, all the explanations and examples are disease-oriented. However, *geNetClassifier* can be applied to the classification of any other type of biological states, pathological or not.

Methods

The algorithm *geNetClassifier()* integrates several existing machine learning and statistical methods. The *feature ranking* is achieved based on a Parametric Empirical Bayes method (PEB). Double-nested internal cross-validation (CV) [2] is used for the *feature selection* process and to estimate the *generalization error* of the classifier. The machine learning method implemented in the classifier is a multi-class Support Vector Machine (SVM) [3]. The gene *networks* are built calculating the relations derived from gene to gene co-expression analysis (by default, *Pearson correlation*) and the interactions derived from gene mutual information analysis (using *minet* package) [4]. More details about these methods are available in the appropriate sections.

Queries

geNetClassifier includes a *query* function that allows either validation of the classifiers using external independent samples of known class (section 4) or classification of new samples whose class is unknown (section 5). This function facilitates the application of the classification algorithm as a predictor for new samples, and it is designed to resemble expert behavior by allowing *NotAssigned* (NA) instances when it is not sure about the class labelling. In order to assign a sample to a class, the algorithm requires a minimum certainty (i.e. probability), leaving it unassigned in case it does not achieve a clear call to a single class. These probability thresholds can be tuned to achieve a more or less stringent assignment. By following this procedure, the algorithm emulates human experts in the decision-making.

2 Install the package and example data

To install *geNetClassifier* from *Bioconductor*:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("geNetClassifier")
```

To follow the examples presented in this *Vignette*, we also need to install a sample dataset called *leukemiasEset*:

```
> biocLite("leukemiasEset")
```

This dataset contains an *expresssionSet* built with 60 gene expression microarrays (HG-U133 plus 2.0 from *Affymetrix*) hybridized with mRNA extracted from bone marrow biopsies of patients of the 4 major types of leukemia (ALL, AML, CLL and CML) and from non-leukemia controls (NoL). These data was produced by the Microarray Innovations in LEukemia (MILE) research project [1] and are available at GEO, under accession number GSE13159. The selected samples are labeled keeping their source GEO IDs.

To have an overview of this *ExpressionSet* and its available info:

```
> library(leukemiasEset)
> data(leukemiasEset)
> leukemiasEset

ExpressionSet (storageMode: lockedEnvironment)
assayData: 20172 features, 60 samples
  element names: exprs, se.exprs
protocolData
  sampleNames: GSM330151.CEL GSM330153.CEL ... GSM331677.CEL (60 total)
  varLabels: ScanDate
  varMetadata: labelDescription
phenoData
  sampleNames: GSM330151.CEL GSM330153.CEL ... GSM331677.CEL (60 total)
  varLabels: Project Tissue ... Subtype (5 total)
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: genemapperhgu133plus2

> summary(leukemiasEset$LeukemiaType)

ALL AML CLL CML NoL
 12  12  12  12  12

> pData(leukemiasEset)
```

For further information/help about this *ExpressionSet*:

```
> ?leukemiasEset
```

CEL files were preprocessed using an alternative Chip Description File (CDF), which allows mapping the expression directly to genes (Ensembl IDs ENSG) instead of *Affymetrix* probesets. This alternative CDF, with redefines gene-based annotation files for the *Affymetrix* expression microarrays, can be found in *GATExplorer* (a bioinformatic web platform that integrates a gene loci browser with nucleotide level re-mapping of the oligo *probes* from *Affymetrix* expression microarrays to genes: mRNAs and ncRNAs)[5].

To translate these Ensembl gene IDs into Gene Symbols for easier reading, the optional argument *geneLabels* from *geNetClassifier* can be used. This option allows to extend the annotation and labelling of the genes by providing a table that contains the gene symbol and other characteristics of the genes in the *expresssionSet*. This option can be used with any annotation (i.e. Bioconductor's *org.Hs.eg.db* package) as long as it is provided in the correct format. However, for increased consistency between versions, when using *GATExplorer* CDF, we recomend to also use *GATExplorer* annotation files. Annotation files with the Gene Symbol corresponding to each Ensembl gene ID can be found at: `http://bioinfow.dep.usal.es/xgate/mapping/mapping.php?content=annotationfiles`. The one used in this example is the *Human Genes* R annotation file. A subset of this file was saved into the object *geneSymbols* for easier use in the examples:

```
> data(geneSymbols)
> head(geneSymbols)
```

This annotation file provides further information which can be used to filter the genes. i.e. To consider only protein-coding genes for the construction of *geNetClassifier*, use the following filter:

```
> load("genes-human-annotation.R")
> leukEset_protCoding <- leukemiasEset[featureNames(leukemiasEset)
+ %in% rownames(genes.human.Annotation[genes.human.Annotation$biotype
+ %in% "protein_coding",]),]
> dim(leukemiasEset)
> dim(leukEset_protCoding)
```

Please note that *geNetClassifier* is designed to work with genes. In case the expression data is not summarized into genes (i.e. it uses the default *probesets*) *geNetClassifier* can still be used but those probesets/features will still be called *genes*.

3 Main function of the package: *geNetClassifier()*

geNetClassifier() is the main function of the package. It builds the classifier and the gene network associated to each class, and also returns the genes ranking and further information about the selected genes.

The workflow internally followed by *geNetClassifier()* includes the following steps:

1.- Filtering data and calculating the genes ranking.
2.- Calculating correlations between genes.
3.- Calculating interactions between genes.
Optional - Filter of redundant genes from the ranking (see arguments *removeCorrelations* and *removeInteractions*).

4.- Construction of the classifier: Selects of a subset of genes to train the classifier through 8-fold cross-validation. The selected genes are used to train the classifier with the complete set of samples.
5.- Estimation of performance: calculates the *generalization error* of the classifier and the statistics about the genes adding an 5-fold *cross-validation* around the construction of the classifier (*nested cross-validation*).
6.- Construction of the gene networks: a gene network is built for each one of the classes using the pairwise gene-to-gene correlations and interactions.
7.- Writing and saving the results including a series of plots for visualization.

The following sections show: how to load the package and the data (sec. 3.1); how to run the algorithm (sec. 3.2); an overview of the results and returned data (sec. 3.3): the genes ranking (sec. 3.4), the classifier (sec. 3.5) and the gene networks (sec. 3.6).

3.1 Loading the package and data

In order to have *geNetClassifier* functions available, the first step is to load the package:

```
> library(geNetClassifier)
```

To list all available tutorials for this package, or to open this *Vignette* you can use:

```
> # List available vignettes for package geNetClassifier:
> vignette(package="geNetClassifier")
> # Open vignette named "geNetClassifier-vignette":
> vignette("geNetClassifier-vignette")
```

To list all the available functions and objects included in *geNetClassifier* use the function *objects()*. Typing its name with a question mark (?) before any function, will show its help file. Through this tutorial, we will see how to use the main ones:

```
> objects("package:geNetClassifier")
```

```
 [1] "calculateGenesRanking"          "externalValidation.probMatrix"
 [3] "externalValidation.stats"       "gClasses"
 [5] "genesDetails"                   "geNetClassifier"
 [7] "getEdges"                       "getNodes"
 [9] "getNumEdges"                    "getNumNodes"
[11] "getRanking"                     "getSubNetwork"
[13] "getTopRanking"                  "initialize"
[15] "network2txt"                    "numGenes"
[17] "numSignificantGenes"            "overview"
[19] "plotAssignments"                "plotDiscriminantPower"
[21] "plotExpressionProfiles"         "plot.GenesNetwork"
[23] "plot.GenesRanking"              "plot.GeNetClassifierReturn"
[25] "plotNetwork"                    "queryGeNetClassifier"
[27] "querySummary"                   "setProperties"
[29] "show"
```

```
> ?geNetClassifier
```

After the package is loaded, you can proceed to analyze your data. In this vignette we use *leukemiasEset*: 60 microarrays from bone marrow from patients of the 4 major types of leukemia (ALL, AML, CLL, CML) and from healthy non-leukemia controls (NoL).(For installation and further information regarding *leukemiasEset* data package see Section 2).

```
> library(leukemiasEset)
> data(leukemiasEset)
```

In *leukemiasEset* there are 60 samples: 12 of each class (ALL, AML, CLL, CML and NoL). We will select 10 samples from each class to execute *geNetClassifier()*, and leave 2 for external validation of the resulting classifier. In this way, it makes a total of 50 samples for the *training* and 10 samples for the *validation*.

```
> trainSamples <- c(1:10, 13:22, 25:34, 37:46, 49:58)
> summary(leukemiasEset$LeukemiaType[trainSamples])
```

```
ALL AML CLL CML NoL
 10  10  10  10  10
```


3.2 Run *geNetClassifier()*

The essential input elements that *geNetClassifier* needs are:

1.- An *expressionSet*: R object defined in Bioconductor that contains a genome-wide expression matrix with data for multiple samples; see *?ExpressionSet*. Note that since the ranking is built though package *EBarrays*, the data in the expression set should be normalized intensity values (positive and on raw scale, not on a logarithmic scale).

2.- The *sampleLabels*: a vector with the class name of each sample or the *ExpressionSet phenoData* object containing this information. Note that to run *geNetClassifier* it is highly recommended to have the **same number of samples in each class**. A balanced number of samples allows an even exploration of each class and provides better classification.

The algorithm input also includes many other arguments that allow to personalize the execution or modify some of the parameters internally used. All of them have a default value and there is no need to modify them. In the following step we will see examples on how to use the main ones. Information about them can be found using the help options (i.e. *?geNetClassifier*). This is the full list of arguments with their default values:

```
geNetClassifier(eset, sampleLabels, plotsName=NULL, buildClassifier=TRUE,
estimateGError=FALSE, calculateNetwork=TRUE, labelsOrder=NULL,
geneLabels=NULL, numGenesNetworkPlot=100, minGenesTrain=1,
maxGenesTrain=100, continueZeroError=FALSE, numIters=6, lpThreshold=0.95,
numDecimals=3, removeCorrelations=FALSE, correlationsThreshold=0.8,
correlationMethod="pearson", removeInteractions=FALSE, interactionsThreshold=0.5,
skipInteractions=FALSE, minProbAssignCoeff=1, minDiffAssignCoeff=0.8,
IQRfilterPercentage=0, precalcGenesNetwork=NULL, precalcGenesRanking=NULL,
returnAllGenesRanking=TRUE, verbose=TRUE)
```

The execution time will depend on the computer and the size of the dataset. To avoid waiting now for the construction of a new classifier to continue this tutorial, a pre-executed example is included in the package:

```
> data(leukemiasClassifier)
```

This classifier was built running the following code:

```
> leukemiasClassifier <- geNetClassifier(leukEset_protCoding[,trainSamples],
+ sampleLabels="LeukemiaType", plotsName="leukemiasClassifier",
+ estimateGError=TRUE, geneLabels=geneSymbols)
```

These are some examples of standard use:

- The fastest execution would be training the classifier exploring a reduced number of genes (by default *maxGenesTrain=100*). In order ot skip calculating the network within the genes, set *calculateNetwork=FALSE*. However, since the correlations are relatively fast to calculate, we recommend keeping *calculateNetwork=TRUE*, and set *skipInteractions=TRUE* instead.

  ```
  > leukemiasClassifier <- geNetClassifier(eset=leukemiasEset[,trainSamples],
  + sampleLabels="LeukemiaType", plotsName="leukemiasClassifier",
  + skipInteractions=TRUE, maxGenesTrain=20, geneLabels=geneSymbols)
  ```

- The default execution (*buildClassifier=TRUE*, *calculateNetwork=TRUE*) only requires the *expressionSet* and the *sampleLabels*. Providing *plotsName* is also recommended in order to produce the plots:

  ```
  > leukemiasClassifier <- geNetClassifier(eset=leukemiasEset[,trainSamples],
  + sampleLabels="LeukemiaType", plotsName="leukemiasClassifier")
  ```

- In order to also estimate the classifier's performance, set *estimateGError=TRUE*. This option will take longer to execute

  ```
  > leukemiasClassifier <- geNetClassifier(eset=leukemiasEset[,trainSamples],
  + sampleLabels="LeukemiaType", plotsName="leukemiasClassifier",
  + estimateGError=TRUE)
  ```

Some of the parameters allow to provide extra information for an easier reading of the results:
- *labelsOrder* allows to show and plot the classes in a specific order (i.e. *labelsOrder=c('ALL', 'CLL', 'AML', 'CML', 'NoL')*)
- *geneLabels* can be used to add a label to the genes to show in the outputs instead of the *featureNames* from the *ExpressionSet*.
In the example, the genes were labeled with the gene symbols provided by *GATExplorer* gene-based probe mapping (*geneLabels=geneSymbols*), as it was indicated in section 3.1.

After running *geNetClassifier()*, we recommend to save the output:

```
> getwd()
> save(leukemiasClassifier, file="leukemiasClassifier.RData")
```

3.3 Overview of the data returned by *geNetClassifier()*

The main results that *leukemiasClassifier()* provides are: the **genes ranking** (sec. 3.4), the **classifier** (sec.3.5) and the **gene networks** (sec. 3.6). All this information is returned by *geNetClassifier()* in an object of class *GeNetClassifierReturn*. This objec tontains several slots which can be seen with the function names():

```
> names(leukemiasClassifier)
```

```
[1] "call"                "classifier"
[3] "classificationGenes" "generalizationError"
[5] "genesRanking"        "genesRankingType"
[7] "genesNetwork"        "genesNetworkType"
```

The slot **@call** contains the R sentence that was used to execute *geNetClassifier()*. It is the only slot that will always be returned by *geNetClassifier()*, the presence and contents of the other components returned by the algorithm will depend on the arguments used to run it.

```
> leukemiasClassifier@call
```

```
geNetClassifier(eset = leukEset_protCoding[, trainSamples], sampleLabels = "LeukemiaT
    plotsName = "leukemiasClassifier", buildClassifier = TRUE,
    estimateGError = TRUE, calculateNetwork = TRUE, geneLabels = geneSymbols)
```

All the outputs and returned components are explained in detail in the following sections:

- @**genesRanking** in section 3.4
- @**classifier** and @**classificationGenes** in section 3.5
- @**generalizationError** in section 3.5.2
- @**genesNetwork** in section 3.6
- The **plots** are explained in section 6

A general view of the output can be seen by just typing the assigned name:

```
> leukemiasClassifier
```

```
R object summary:
Classifier trained with 50 samples.
Total number of genes included in the classifier: 26.
Number of genes per class:
ALL AML CLL CML NoL
  9   5   1   5   6
For classificationGenes details: genesDetails(EXAMPLE@classificationGenes)

Generalization error and gene stats calculated through 5-fold cross-validation:
[1] "accuracy"                  "sensitivitySpecificity"
[3] "confMatrix"                "probMatrix"
[5] "querySummary"              "classificationGenes.stats"
```

```
[7] "classificationGenes.num"

The ranking of all genes contains (genes per class):
 ALL  AML  CLL  CML  NoL
2342 3023 2824 2539 3049

The networks calculated for the topGenes genes of each class contain:
                      ALL AML   CLL  CML  NoL
Number of genes      1027 400  1916  949  400
Number of relations 1942 296 18506 6540 1993

Available slots in this R object:
[1] "call"                "classifier"          "classificationGenes"
[4] "generalizationError" "genesRanking"        "genesRankingType"
[7] "genesNetwork"        "genesNetworkType"
To see an overview of all available slots type "overview(EXAMPLE)"
```

3.4 Return I: Genes ranking

The first step of *geNetClassifier* algorithm is to determine a ranking of genes for each class based in the analysis of the expression signal. To create this ranking, it uses the function *emfit*, a Parametric Empirical Bayes method [6], included in package *EBarrays* [7]. This method implements an expectation-maximization (EM) algorithm for gene expression mixture models, which compares the patterns of differential expression across multiple conditions and provides a *posterior probability*.

The posterior probability is calculated for each gene-class pair, and represents how much each gene differentiates a class from the other classes; being 1 the best value, and 0 the worst. In this way, the posterior probability allows to find the genes that show significant differential expression when comparing the samples of one class *versus* all the other samples (One-versus-Rest comparison).

A first version of the ranking is built by ordering the genes decreasingly by their posterior probability for each class. To resolve the ties, *geNetClassifier* uses the expression difference between the mean for each gene in the given class and the mean in the closest class. In addition, the genes with a posterior probability greater or equal to 0.95 for the 'no difference' -the genes that do not show any difference between classes- are filtered out before proceeding into further steps.

The final version of the ranking is built assigning each gene to the class in which it has the best ranking. In this way the separation between classes is optimized, and the method will choose first the genes that best differentiate any of the classes. As a result of this process, even if a gene is found associated to several classes during the expression analysis, **each gene can only be on the ranking of one class**.

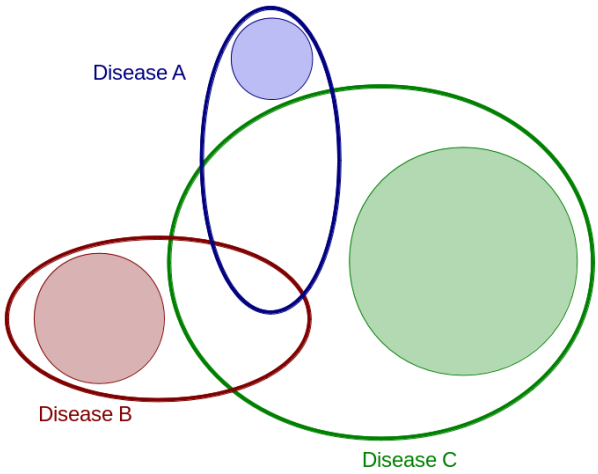


Figure 2. Scheme representing the overlap between the sets of genes that each disease may affect. *geNetClassifier* explores all the genes that affect each disease (**ovals**) and selects as significant, the genes that are unique (differentially expressed) to each disease (**coloured circles**).

The genes ranking obtained for each class is used for the gene selection in the classification procedure and it is also provided as an output of *geNetClassifier()* in the slot: ...@genesRanking.

```
> leukemiasClassifier@genesRanking

Top ranked genes for the classes:  ALL AML CLL CML NoL
      ALL        AML       CLL         CML        NoL
 [1,] "VPREB1"   "HOXA9"   "TYMS"      "GJB6"     "FGF13"
 [2,] "ZNF423"   "MEIS1"   "FCER2"     "PRG3"     "NMU"
 [3,] "DNTT"     "CD24L4"  "NUCB2"     "LY86"     "SMPDL3A"
 [4,] "EBF1"     "ANGPT1"  "RRAS2"     "ABP1"     "KLRB1"
 [5,] "PXDN"     "CCNA1"   "PNOC"      "TRIM22"   "RNF182"
 [6,] "S100A16"  "ZNF521"  "C6orf105"  "NLRC3"    "RFESD"
 [7,] "CSRP2"    "HOXA5"   "RRM2"      "LPXN"     "SLC25A21"
 [8,] "SOCS2"    "DEPDC6"  "KIAA0101"  "GBP3"     "CD160"
 [9,] "CTGF"     "NKX2-3"  "UHRF1"     "TNS3"     "CLIC2"
[10,] "COL5A1"   "NPTX2"   "ABCA6"     "ZC3H12D"  "TMEM56"
...

Number of ranked significant genes (posterior probability over threshold):
         ALL AML CLL CML NoL
         1027 273 1916 949 191
To see the whole ranking (3049 rows) use: getRanking(...)
Details of the top X ranked genes of each class: genesDetails(..., nGenes=X)
```

This ranking an object of class *GenesRanking*. This class provides some utility functions which will help working with the information contained in the object. The total number

of genes in the ranking for each class can be queried using the function *numGenes()*. These numbers include all the genes that have some ability to distinguish between classes, although only the top ones are really significant.

```
> numGenes(leukemiasClassifier@genesRanking)

 ALL  AML  CLL  CML  NoL
2342 3023 2824 2539 3049
```

With *getTopRanking()* a subset of the ranking containing only the given number of top genes can be obtained. Since the returned object is also a *GenesRanking* object, no information is lost and other functions (i.e. *genesDetails()*) can be used afterwards.

```
> subRanking <- getTopRanking(leukemiasClassifier@genesRanking, 10)
```

In order to retrieve the whole ranking in the form of a matrix (i.e. to print the full version or get a subset of it), the function *getRanking()* can be used. This function provides the option to show the ranking with the gene IDs or the gene Labels.

```
> getRanking(subRanking)

$geneLabels
      ALL       AML      CLL         CML       NoL
 [1,] "VPREB1"  "HOXA9"  "TYMS"      "GJB6"    "FGF13"
 [2,] "ZNF423"  "MEIS1"  "FCER2"     "PRG3"    "NMU"
 [3,] "DNTT"    "CD24L4" "NUCB2"     "LY86"    "SMPDL3A"
 [4,] "EBF1"    "ANGPT1" "RRAS2"     "ABP1"    "KLRB1"
 [5,] "PXDN"    "CCNA1"  "PNOC"      "TRIM22"  "RNF182"
 [6,] "S100A16" "ZNF521" "C6orf105" "NLRC3"   "RFESD"
 [7,] "CSRP2"   "HOXA5"  "RRM2"      "LPXN"    "SLC25A21"
 [8,] "SOCS2"   "DEPDC6" "KIAA0101" "GBP3"    "CD160"
 [9,] "CTGF"    "NKX2-3" "UHRF1"     "TNS3"    "CLIC2"
[10,] "COL5A1"  "NPTX2"  "ABCA6"     "ZC3H12D" "TMEM56"

> getRanking(subRanking, showGeneID=TRUE)$geneID[,1:4]

      ALL                AML                CLL                CML
 [1,] "ENSG00000169575" "ENSG00000078399" "ENSG00000176890" "ENSG00000121742"
 [2,] "ENSG00000102935" "ENSG00000143995" "ENSG00000104921" "ENSG00000156575"
 [3,] "ENSG00000107447" "ENSG00000185275" "ENSG00000070081" "ENSG00000112799"
 [4,] "ENSG00000164330" "ENSG00000154188" "ENSG00000133818" "ENSG00000002726"
 [5,] "ENSG00000130508" "ENSG00000133101" "ENSG00000168081" "ENSG00000132274"
 [6,] "ENSG00000188643" "ENSG00000198795" "ENSG00000111863" "ENSG00000167984"
 [7,] "ENSG00000175183" "ENSG00000106004" "ENSG00000171848" "ENSG00000110031"
 [8,] "ENSG00000120833" "ENSG00000155792" "ENSG00000166803" "ENSG00000117226"
 [9,] "ENSG00000118523" "ENSG00000119919" "ENSG00000034063" "ENSG00000136205"
[10,] "ENSG00000130635" "ENSG00000106236" "ENSG00000154262" "ENSG00000178199"
```

The function *genesDetails()* allows to show all the available info of the genes in the ranking.

```
> genesDetails(subRanking)$AML
```

```
                GeneName ranking class postProb exprsMeanDiff exprsUpDw
ENSG00000078399    HOXA9       1   AML        1        4.4362        UP
ENSG00000143995    MEIS1       2   AML        1        3.2785        UP
ENSG00000185275   CD24L4       3   AML        1       -4.4926      DOWN
ENSG00000154188   ANGPT1       4   AML        1        2.7427        UP
ENSG00000133101    CCNA1       5   AML        1        2.5558        UP
ENSG00000198795   ZNF521       6   AML        1        2.5697        UP
ENSG00000106004    HOXA5       7   AML        1        3.1729        UP
ENSG00000155792   DEPDC6       8   AML        1        2.4803        UP
ENSG00000119919   NKX2-3       9   AML        1        2.1962        UP
ENSG00000106236    NPTX2      10   AML        1        2.0582        UP
                isRedundant
ENSG00000078399       FALSE
ENSG00000143995        TRUE
ENSG00000185275       FALSE
ENSG00000154188       FALSE
ENSG00000133101       FALSE
ENSG00000198795        TRUE
ENSG00000106004        TRUE
ENSG00000155792        TRUE
ENSG00000119919       FALSE
ENSG00000106236       FALSE
```

NOTE: If the console splits the table into several lines, try:

```
> options(width=200)
```

By default, the *rownames* are the ID included in the *expressionSet*: in our case the ENSEMBL IDs. The *GeneName* column has been added by setting the argument *geneLabels=geneSymbols* (see sec. 3.2).

To see the description of the content of this table write: *?genesDetails*.

More details about *GenesRanking* class is available at: *?GenesRanking*.

3.4.1 Significant genes

The set of genes considered *significant* for each of the classes is determined by a common threshold for the posterior probability (by default *lpThreshold=0.95*). This common threshold provides a way to quantify the size of the *gene signature* assigned to each disease (as always: compared to the other diseases in the study). In this way, the algorithm provides a framework to compare biological states, i.e. the biological or pathological conditions represented in the samples.

plotSignificantGenes() provides a plot of the distribution of the posterior probabilities of the genes within the rankings for each class:

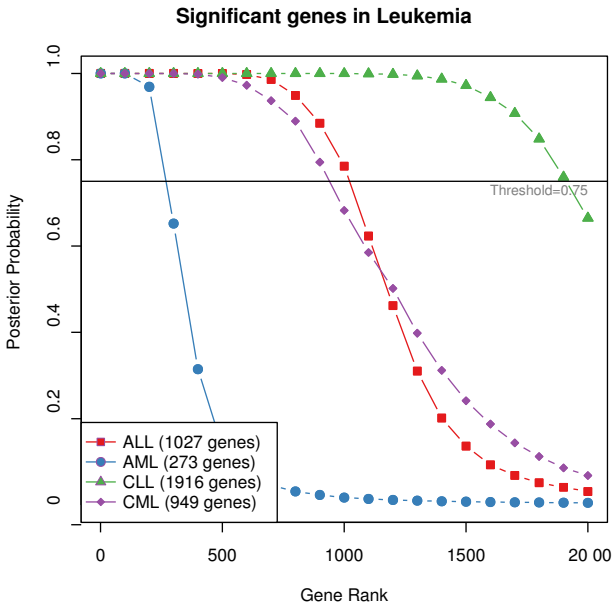


Figure 3. Plot of the posterior probabilities of the genes of 4 leukemia classes, ordering the genes according to their rank.

This example shows the big differences in size of the gene sets assigned to a disease: at lpThreshold 0.95 CLL has been assigned 2028 genes, while AML only 308 genes. The biological interpretation of this observation will depend on the specific study. Larger gene signatures may be an indication of more *systemic* diseases (i.e. a disease affect more genes than another), but it may also be an indication of the relative differences between the diseases in the study (i.e. one of the diseases affects different genes than the others). In any case, the results provided by *geNetClassifier* may help to unravel disease sub-types differences based on the gene signatures.

numSignificantGenes() provides the number of significant genes, the number of genes with posterior probability over the threshold:

```
> numSignificantGenes(leukemiasClassifier@genesRanking)

 ALL  AML  CLL  CML  NoL
1027  273 1916  949  191
```

The plot of the posterior probability (*plotSignificantGenes()*) is the default plot for objects of class *GenesRanking*. (More details in section 6.1).

```
> plot(leukemiasClassifier@genesRanking)
```

In both functions, the threshold can be modified through *lpThreshold*:

```
> plot(leukemiasClassifier@genesRanking,
+ numGenesPlot=3000, lpThreshold=0.80)
```

3.5 Return II: Classifier

The information regarding the classifier is saved into the slots: @classifier, @classifcation-Genes and @generalizationError.

The ***@classifier*** slot contains the SVM classifier that can later be used to make queries. The SVM method included in the algorithm is a linear kernel implementation from R package *e1071*. This implementation allows multi-class classification by using a One-versus-One (OvO) approach, in which all the binary classifications are fitted and the correct class is found based on a voting system.

```
> leukemiasClassifier@classifier

$SVMclassifier

Call:
svm.default(x = t(esetFilteredDataFrame[buildGenesVector, trainSamples]),
    y = sampleLabels[trainSamples], kernel = "linear", probability = T,
    C = 1)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.03846154

Number of Support Vectors:  29
```

@classificationGenes contains the final genes selected to build the classifier. Since *@classificationGenes* is an object of class *GenesRanking*, functions such as *numGenes()* or *genesDetails()* can be used to explore it.

```
> leukemiasClassifier@classificationGenes

Top ranked genes for the classes:  ALL AML CLL CML NoL
      ALL       AML      CLL    CML      NoL
 [1,] "VPREB1"  "HOXA9"  "TYMS" "GJB6"   "FGF13"
 [2,] "ZNF423"  "MEIS1"  NA     "PRG3"   "NMU"
 [3,] "DNTT"    "CD24L4" NA     "LY86"   "SMPDL3A"
 [4,] "EBF1"    "ANGPT1" NA     "ABP1"   "KLRB1"
```

```
 [5,] "PXDN"    "CCNA1"  NA     "TRIM22" "RNF182"
 [6,] "S100A16" NA       NA     NA       "RFESD"
 [7,] "CSRP2"   NA       NA     NA       NA
 [8,] "SOCS2"   NA       NA     NA       NA
 [9,] "CTGF"    NA       NA     NA       NA

Details of the top X ranked genes of each class: genesDetails(..., nGenes=X)

> numGenes(leukemiasClassifier@classificationGenes)

ALL AML CLL CML NoL
  9   5   1   5   6

> genesDetails(leukemiasClassifier@classificationGenes)$ALL

                GeneName ranking gERankMean class postProb exprsMeanDiff
ENSG00000169575   VPREB1       1        1.0   ALL        1        6.3307
ENSG00000102935   ZNF423       2        3.0   ALL        1        5.0980
ENSG00000107447     DNTT       3        2.8   ALL        1        6.8948
ENSG00000164330     EBF1       4        3.8   ALL        1        5.4171
ENSG00000130508     PXDN       5        5.2   ALL        1        5.0387
ENSG00000188643  S100A16       6        5.4   ALL        1        4.3434
ENSG00000175183    CSRP2       7        7.8   ALL        1        4.0479
ENSG00000120833    SOCS2       8       10.8   ALL        1        4.5383
ENSG00000118523     CTGF       9       14.8   ALL        1        3.6167
                exprsUpDw discriminantPower discrPwClass isRedundant
ENSG00000169575        UP          9.416945          ALL       FALSE
ENSG00000102935        UP         13.240579          ALL        TRUE
ENSG00000107447        UP          8.978735          ALL        TRUE
ENSG00000164330        UP         10.515557          ALL        TRUE
ENSG00000130508        UP          8.657167          ALL        TRUE
ENSG00000188643        UP         12.385161          ALL        TRUE
ENSG00000175183        UP          8.782649          ALL        TRUE
ENSG00000120833        UP          8.697958          ALL       FALSE
ENSG00000118523        UP          5.551344          ALL       FALSE
```

Note that besides the common information about the genes provided by the genes ranking (sec. 3.4), the classification genes also have information about the **discriminant power** of the genes (sec. 6.3).

For details on the *gene selection procedure* (sec. 3.5.1) and the *estimation of performance and generalization error procedure* (slot ***@generalization***) (sec. 3.5.2), see the next two sections.

3.5.1 Gene selection procedure

The optimum number of genes to train the classifier is selected by evaluating the classifiers trained with increasing number of genes. This is done using several iterations of 8-fold cross-validation. Each cross-validation iteration starts with the first ranked gene of each class: it trains an internal classifier with these genes, and evaluates its performance.

One more gene is added in each step to those classes for which a 'perfect prediction' is not achieved (i.e. not all samples correctly identified). The genes are taken in order from the *genes ranking* of each class until any of the classes reaches gets to the maximum number of genes (*maxGenesTrain=100*) or until zero error is reached (*continueZeroError=FALSE*). The error for each of the classifiers and the number of genes used to construct them are saved. Once the cross-validation loop is finished, it saves the minimum number of genes per class which produced the classifier with minimum error.

To achieve the best stability in the number of selected genes, the cross-validation is not run just once, but it is repeated several times with new samplings. This process is repeated as many times as indicated by the optional parameter *numIters* (6 by default). In each of these iterations, the minor number of genes that provided the smallest error is selected.

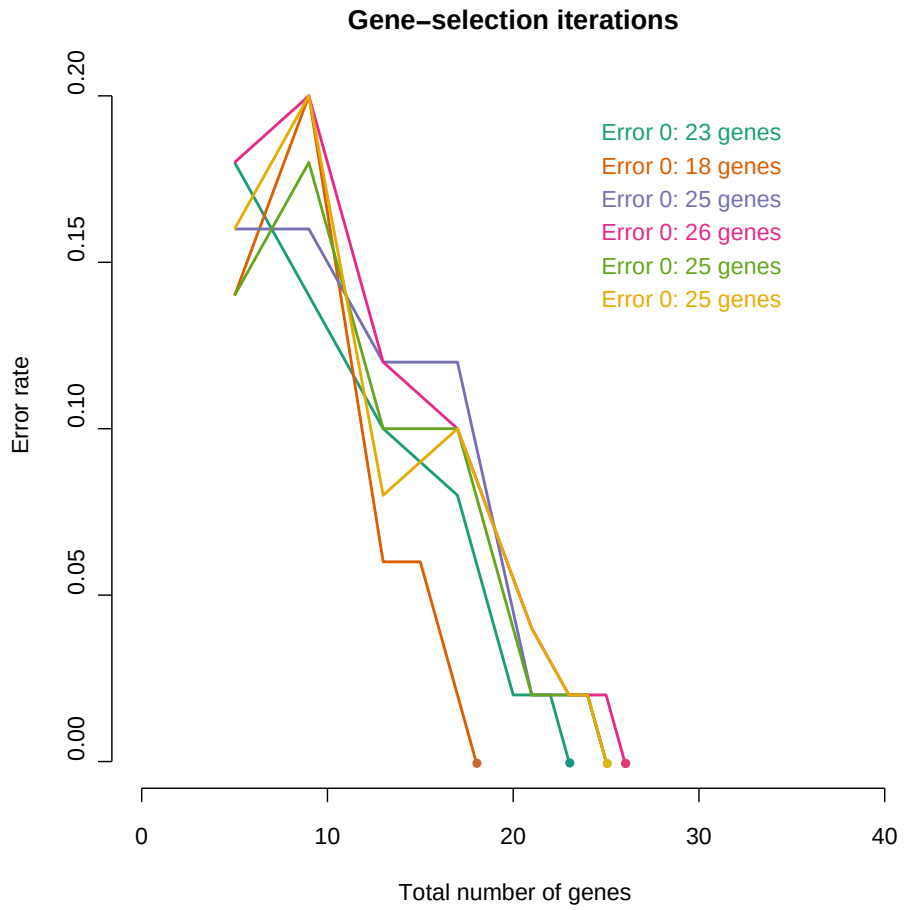


Figure 4. Plot of the gene-selection iterations. Each line represents an iteration and the error rates observed for each number of genes (starting at 5, one per class). The algorithm runs until exploring a maximum number of genes in any class (*maxGeneTrain=100*) or until zero error is reached (*continueZeroError=FALSE*). In each iteration the minimum number of genes with minimum error is selected.

The final selection is done based on the genes selected in each of the iterations. For each class, the top ranked genes are selected by taking the highest number of genes –excluding outliers– selected in the cross-validaton iterations. This allows to identify a stable number of genes, while accounting for the diffences in sampling.

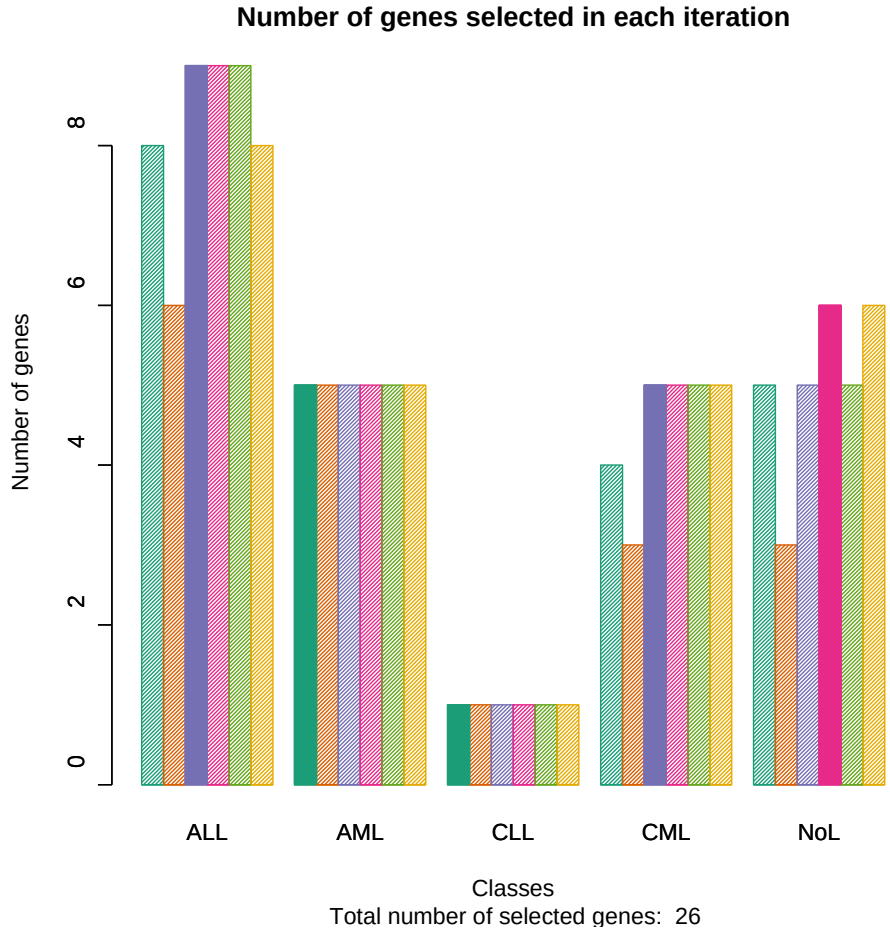


Figure 5. Plot of the number of genes selected in each iteration. The bars represent the number of genes with minimum error rates in each iteration. Each color represents an iteration. The filled bar is the final number of genes of each class selected to train the classifier.

Figures 4 and 5 show the gene selection for the leukemia's example.

3.5.2 Estimation of performance and generalization error procedure

The estimation of the *generalization error* (GE) of the classification algorithm is an option that can be included using the parameter *estimateGError=TRUE*. When this option is chosen, an independent validation is simulated by adding a second loop of cross-validation (CV) around the construction of the classifier. In each iteration of this loop, a few sam-

ples are left out of the *training* and used as *test* samples. This step allows to estimate and provide statistics and metrics regarding the quality of the classifier and the genes selected for classification. The parameters measured for the classifier are the following:

- **Sensitivity**: Proportion of samples from a given class which were correctly identified. In statistical terms it is the rate of true positives (TP). *Sensitivity* relates to the ability of the test to identify positive results.

$$Sensitivity = \frac{TP}{TP + FN} = TruePositiveRate$$

- **Specificity**: Proportion of samples assigned to a given class which really belonged to the class. In statistical terms it is the rate of true negatives (TN). *Specificity* relates to the ability of the test to identify negative results.

$$Specificity = \frac{TN}{TN + FP} = TrueNegativeRate$$

Note: In order to truly evaluate the classification, both sensitivity and specificity need to be taken into account. For example, 100% sensitivity for AML will be achieved by assigning all AML samples to AML. In the same way, 100% specificity will be achieved by not assigning any sample from other class to AML. Therefore, the classification will only be reliable if both -sensitivity and specificity- are optimized, by identifying all samples from one class while not having samples from another classes miss-classified.

- **Matthews Correlation Coefficient** (MCC): It is a measure which takes into account both true and false positives and negatives. It is generally regarded as a balanced measure of performance. In machine learning it is used as a measure of the quality of binary classifications.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- **Global Accuracy**: Proportion of true results within the assigned samples.

- **Call Rate per class and Global Call Rate**: Proportion of *assigned* samples within a class or in the whole prediction.

$$CallRate = \frac{Assigned}{Assigned + NotAssigned}$$

The results about the estimation of performance and the generalization error are saved in the slot: *@generalizationError*

```
> leukemiasClassifier@generalizationError

Estimated accuracy, sensitivity and specificity for the classifier:
       Accuracy CallRate
Global      100       90
    Sensitivity Specificity MCC CallRate
ALL         100         100 100       90
```

```
AML         100         100 100       70
CLL         100         100 100      100
CML         100         100 100      100
NoL         100         100 100       90

To see all available statistics type "overview(EXAMPLE@generalizationError)"
```

To see all the available info gathered during estimation of performance use the *overview()* function:

```
> overview(leukemiasClassifier@generalizationError)
```

This object contains all the information regarding estimation of performance in different slots: *@accuracy*, *@sensitivitySpecificity*, *@confMatrix*, *@probMatrix*, *@querySummary*.

The slot *...@confMatrix* contains the confusion matrix. A confusion matrix is a table used to quickly visualize and evaluate the performance of a classification algorithm. The rows represent the real class of the samples, while the columns represent the class to which the samples were assigned. Therefore, the correctly assigned samples are in the diagonal.

```
> leukemiasClassifier@generalizationError@confMatrix
```

```
          prediction
testLabels ALL AML CLL CML NoL NotAssigned
       ALL   9   0   0   0   0           1
       AML   0   7   0   0   0           3
       CLL   0   0  10   0   0           0
       CML   0   0   0  10   0           0
       NoL   0   0   0   0   9           1
```

The slot *...@probMatrix* presents the probabilities of assignment to each class that are calculated during the 5-fold cross-validation. This *probability matrix* provides a good estimation of how easy or difficult is to assign each sample to its class. It also provides an indication about the likelihood to confuse one class with others:

```
> leukemiasClassifier@generalizationError@probMatrix
```

```
      ALL   AML   CLL   CML   NoL
ALL 0.697 0.060 0.073 0.067 0.102
AML 0.058 0.770 0.083 0.044 0.045
CLL 0.088 0.094 0.673 0.064 0.080
CML 0.055 0.107 0.064 0.633 0.141
NoL 0.073 0.072 0.055 0.145 0.654
```

The slot *...@classificationGenes.stats* includes calculations about the number of times that each gene was selected for classification in the 5-fold cross-validation executions:
- *timesChosen*, number of times that each gene is chosen for classification in the 5 CV.
- *chosenRankMean*, average rank of the gene only within the CV loops in which the gene was chosen for classification.
- *chosenRankSD*, standard deviation of the gene rank only within the CV loops in which the gene was chosen for classification.

- *geRankMean*, average rank of the gene in the 5 CV loops performed during the generalization error estimation.
- *geRankSD*, standard deviation of the rank of the gene in the 5 CV loops performed during the generalization error estimation.

```
> leukemiasClassifier@generalizationError@classificationGenes.stats$CLL
```

```
                timesChosen chosenRankMean chosenRankSD gERankMean gERankSD
ENSG00000176890           4           1.25         0.50        1.8     1.30
ENSG00000070081           2           1.50         0.71        2.4     0.89
ENSG00000104921           1           1.00         0.00        2.8     1.48
```

The slot *...@classificationGenes.num* includes calculations about the number of genes selected for each class in the 5 runs of the 5-fold cross-validation applied for the estimation of performance. These numbers allow to explore the number of genes that are used per class. However, the proper calculation of the final *number of genes* selected for each class in the classifier is done with the other 8-fold cross-validation which includes all the available samples (as indicated in section 3.5.1).

```
> leukemiasClassifier@generalizationError@classificationGenes.num
```

```
      ALL AML CLL CML NoL
CV 1:   6   7   1   3  10
CV 2:   3   2   1   8   6
CV 3:   9   2   2   6   5
CV 4:   2  16   2   9  16
CV 5:   3   5   1   8  10
```

3.6 Return III: Gene networks

Together to the classifier and the genes ranking, the third major result that the algorithm *geNetClassifier* produces are the gene networks associated to each class.

The gene networks for each class are built based on association parameters between genes. These association parameters are gene to gene co-expression calculated using a correlation coefficient (*Pearson* by default) and gene to gene interactions derived from *mutual information* (MI) analysis (*mi.empirical* entropy estimator from the R package minet [4]); both calculated along all the samples of each class of the studied dataset.

The *correlations* and *interactions* also allow to find possible redundancy between the genes as features in the classification procedure. Such redundancy can be tested by producing comparative classifiers that include or not the associated genes. Usually, classifiers without redundant genes need less features for classification.

The *...@genesNetwork* slot contains the list of networks.

```
> leukemiasClassifier@genesNetwork
```

```
$ALL
Attribute summary of the GenesNetwork:
```

```
Number of nodes (genes): [1] 1027
Number of edges (relationships): [1] 1942

$AML
Attribute summary of the GenesNetwork:
Number of nodes (genes): [1] 400
Number of edges (relationships): [1] 296

$CLL
Attribute summary of the GenesNetwork:
Number of nodes (genes): [1] 1916
Number of edges (relationships): [1] 18506

$CML
Attribute summary of the GenesNetwork:
Number of nodes (genes): [1] 949
Number of edges (relationships): [1] 6540

$NoL
Attribute summary of the GenesNetwork:
Number of nodes (genes): [1] 400
Number of edges (relationships): [1] 1993
```

```
> overview(leukemiasClassifier@genesNetwork$AML)
```

```
getNodes(...)[1:10]:
 [1] "ENSG00000078399" "ENSG00000143995" "ENSG00000185275" "ENSG00000154188"
 [5] "ENSG00000133101" "ENSG00000198795" "ENSG00000106004" "ENSG00000155792"
 [9] "ENSG00000119919" "ENSG00000106236"
... (400 nodes)

getEdges(...)[1:5,]:
     gene1             class1 gene2             class2 relation
[1,] "ENSG00000078399" "AML"  "ENSG00000143995" "AML"  "Correlation - pearson"
[2,] "ENSG00000154188" "AML"  "ENSG00000198795" "AML"  "Correlation - pearson"
[3,] "ENSG00000078399" "AML"  "ENSG00000106004" "AML"  "Correlation - pearson"
[4,] "ENSG00000154188" "AML"  "ENSG00000155792" "AML"  "Correlation - pearson"
[5,] "ENSG00000119919" "AML"  "ENSG00000108511" "AML"  "Correlation - pearson"
     value
[1,] "0.922460476283629"
[2,] "0.804443836092871"
[3,] "0.836149615702043"
[4,] "0.815177435058601"
[5,] "0.940367679337551"
... (296 edges)
```

Each of the networks in this list is an object of the class *GenesNetwork*. This class offers some functions to retrieve and count the edges and nodes, and also to subset the network (*getSubNetwork()*). Note that *getNodes()* includes all possible nodes even if they are no linked by edges.

```
> getNumEdges(leukemiasClassifier@genesNetwork$AML)

[1] 296

> getNumNodes(leukemiasClassifier@genesNetwork$AML)

[1] 400

> getEdges(leukemiasClassifier@genesNetwork$AML)[1:5,]

     gene1             class1 gene2             class2 relation
[1,] "ENSG00000078399" "AML"  "ENSG00000143995" "AML"  "Correlation - pearson"
[2,] "ENSG00000154188" "AML"  "ENSG00000198795" "AML"  "Correlation - pearson"
[3,] "ENSG00000078399" "AML"  "ENSG00000106004" "AML"  "Correlation - pearson"
[4,] "ENSG00000154188" "AML"  "ENSG00000155792" "AML"  "Correlation - pearson"
[5,] "ENSG00000119919" "AML"  "ENSG00000108511" "AML"  "Correlation - pearson"
     value
[1,] "0.922460476283629"
[2,] "0.804443836092871"
[3,] "0.836149615702043"
[4,] "0.815177435058601"
[5,] "0.940367679337551"

> getNodes(leukemiasClassifier@genesNetwork$AML)[1:12]

 [1] "ENSG00000078399" "ENSG00000143995" "ENSG00000185275" "ENSG00000154188"
 [5] "ENSG00000133101" "ENSG00000198795" "ENSG00000106004" "ENSG00000155792"
 [9] "ENSG00000119919" "ENSG00000106236" "ENSG00000148154" "ENSG00000108511"
```

The function *network2txt()* allows to save or export the networks as text files. This function produces two text files: one with the information about the *nodes* and another with the information about the *edges*. They are flat text files (.txt). In the case of the *edges* file, it includes the nodes that interact (gene1 – gene2), the type of link (correlation or interaction) and the value of such relation.

```
> network2txt(leukemiasClassifier@genesNetwork, filePrefix="leukemiasNetwork")
```

To produce just the files with the information about the *edges*:

```
> geneNtwsInfo <- lapply(leukemiasClassifier@genesNetwork,
+     function(x) write.table(getEdges(x),
+     file=paste("leukemiaNtw_",getEdges(x)[1,"class1"],".txt",sep="")))
```

These flat text files allow to export the networks to external software (e.g. *Cytoscape*, http://www.cytoscape.org).

The networks can also be exported using direct R connectors (e.g. RCytoscape) with the igraph objects returned by the function *plotNetwork* (sec. 6.4).

For more information see the class help *?GenesNetwork*.

4 External validation: query with new samples of known class

Once a classifier is built for a group of diseases or disease subtypes, it can be queried with new samples to know their class. However, before proceeding with samples whose class is unknown, an external validation is normally performed. An external validation consists on querying the classifier with several samples whose class is *a priori* known, in order to see if the classification is done correctly. As indicated in section 3.5.2, if the number of known samples is limited (as it is usually the case) to avoid leaving a sub-set of known samples out of the training, *geNetClassifier()* provides the *generalization error* option, which will simulate an external validation by using cross-validation. Despite this possibility, it is clear that using external samples (totally independent to the classifier built) is the best option to validate its performance.

In this section, we will proceed with an example of external validation with the leukemia's classifier. In *leukemiasEset*, the class of all the available samples is known *a priori*. Since we had 60 samples in the initial leukemia dataset and only 50 were used to train the classifier, the 10 remaining can be used for external validation.

The first step is to select the 10 samples that were not used for training:

```
> testSamples <- c(1:60)[-trainSamples]
> testSamples

 [1] 11 12 23 24 35 36 47 48 59 60
```

The classifier is then be asked about the class of these 10 samples using *queryGeNetClassifier()*:

```
> queryResult <- queryGeNetClassifier(leukemiasClassifier,
+ leukemiasEset[,testSamples])
```

This query will return the class that each sample has been assigned to, which will be saved into $class. It also returns the probabilities of assignment of each sample to each class in $probabilities.

```
> queryResult$class

GSM330195.CEL GSM330201.CEL GSM330611.CEL GSM330612.CEL GSM331037.CEL
          ALL           ALL           AML           AML           CLL
GSM331048.CEL GSM331392.CEL GSM331393.CEL GSM331675.CEL GSM331677.CEL
          CLL           CML           CML           NoL           NoL
Levels: ALL AML CLL CML NoL

> queryResult$probabilities

    GSM330195.CEL GSM330201.CEL GSM330611.CEL GSM330612.CEL GSM331037.CEL
ALL    0.82480476    0.72132949    0.04584317    0.03853380    0.04233982
AML    0.04145204    0.05669690    0.68053161    0.84706650    0.09093176
CLL    0.02591494    0.03200663    0.09622283    0.02028114    0.75041107
CML    0.04409894    0.08325862    0.08198307    0.07359096    0.04732196
```

```
NoL    0.06372931    0.10670835    0.09541932    0.02052760    0.06899539
    GSM331048.CEL GSM331392.CEL GSM331393.CEL GSM331675.CEL GSM331677.CEL
ALL    0.04115917    0.04569346    0.02645151    0.05492039    0.02213885
AML    0.09742257    0.17443163    0.02549073    0.05510842    0.04907441
CLL    0.71914364    0.13181179    0.03923288    0.09748276    0.01016039
CML    0.07715866    0.56354463    0.87901701    0.04714128    0.03225649
NoL    0.06511596    0.08451848    0.02980787    0.74534715    0.88636986
```

Since the real class of the samples is known, we can create a confusion matrix. Note: For using this matrix as input in upcoming functions the real classes should be placed as row names (*rownames*) and the predicted classes (assigned by the classifier) as column names (*colnames*).

```
> confusionMatrix <- table(leukemiasEset[,testSamples]$LeukemiaType,
+ queryResult$class)
```

Once we have executed the query, *externalValidation.stats()* can be used to calculate the parameters to evaluate the classifier (Section 3.5.2).

```
> externalValidation.stats(confusionMatrix)
```

```
$byClass
    Sensitivity Specificity MCC CallRate
ALL         100         100 100      100
AML         100         100 100      100
CLL         100         100 100      100
CML         100         100 100      100
NoL         100         100 100      100

$global
       Accuracy CallRate
Global      100      100

$confMatrix
    ALL AML CLL CML NoL NotAssigned
ALL   2   0   0   0   0           0
AML   0   2   0   0   0           0
CLL   0   0   2   0   0           0
CML   0   0   0   2   0           0
NoL   0   0   0   0   2           0
```

The class to class assignment probability matrix, that gives support to the confusion matrix, can be also created for the external validation analysis:

```
> externalValidation.probMatrix(queryResult,
+ leukemiasEset[,testSamples]$LeukemiaType, numDecimals=3)
```

```
      ALL   AML   CLL   CML   NoL
ALL 0.773 0.049 0.029 0.064 0.085
AML 0.042 0.764 0.058 0.078 0.058
CLL 0.042 0.094 0.735 0.062 0.067
CML 0.036 0.100 0.086 0.721 0.057
NoL 0.039 0.052 0.054 0.040 0.816
```

4.1 Assignment conditions

queryGeNetClassifier() includes an expert-like approach to decide if a sample is assigned to a class: instead of directly assigning a sample to the class with the highest probability, it takes into account the probability of belonging to the class and the probability of the closest class before taking the final decision.

By default, the probability to assign a sample to a given class should be at least double than the *random probability*, and the difference with the next likely class should also be higher than 0.8 times the *random probability*. For example, to assign a sample in a 5 class classifier, the highest probability should be at least 40% (2 x 0.20 = 0.40) and the probability of belonging to the closest class should be at least 16% lower than the highest (0.8 x 0.20 = 0.16). This implies that if a sample's probability to belong to one class is 55% and to belong to another class is 40%, since the the difference is lower than 16%, it is not clear enough, and it will be left as a *NotAssigned* (NA). This feature allows modulation of the assignment to resembles expert decision-making.

To allow adapting these conditions, *queryGeNetClassifier()* includes two coefficients that determine the *minimum probability for assignment* (minProbAssignCoeff), and the *minimum difference between the of the first and the second classes* (minDiffAssignCoeff). If these two coefficients are set up to 0 all samples will be assigned to the most likely class and therefore no samples will be left as *NotAssigned*.

```
> queryResult_AssignAll <- queryGeNetClassifier(leukemiasClassifier,
+     leukemiasEset[,testSamples], minProbAssignCoeff=0, minDiffAssignCoeff=0)
> which(queryResult_AssignAll$class=="NotAssigned")
```

```
integer(0)
```

On the contrary, the thresholds can be raised to increase the the certainty of the assignments: i.e. by setting the coefficients to 1.5 and 1, the minimum probability to be assigned is 0.6 (**1.5** x 2 x 0.20) and the minimum difference between first and second class probabilities is 0.2 (**1** x 0.20).

```
> queryResult_AssignLess <- queryGeNetClassifier(leukemiasClassifier,
+     leukemiasEset[,testSamples], minProbAssignCoeff=1.5, minDiffAssignCoeff=1)
> queryResult_AssignLess$class
```

```
GSM330195.CEL GSM330201.CEL GSM330611.CEL GSM330612.CEL GSM331037.CEL
          ALL           ALL           AML           AML           CLL
GSM331048.CEL GSM331392.CEL GSM331393.CEL GSM331675.CEL GSM331677.CEL
          CLL   NotAssigned           CML           NoL           NoL
Levels: ALL AML CLL CML NoL NotAssigned
```

In this case, these samples were left as *NotAssigned*:

```
> t(queryResult_AssignLess$probabilities[,
+     queryResult_AssignLess$class=="NotAssigned", drop=FALSE])
```

```
                     ALL       AML       CLL       CML        NoL
GSM331392.CEL 0.04569346 0.1744316 0.1318118 0.5635446 0.08451848
```

To help understanding how these thresholds behave for a specific dataset, if *geNetClassifier()* is executed with *estimateGError=TRUE*, it generates a plot presenting the assignment probabilities for each sample. This plot shows the probability of the most likely class *versus* the probability difference with next likely class for each sample. Therefore, it allows to view the effects of the 2 coefficients (*minProbAssignCoeff* and *minDiffAssignCoeff*) in the assignment.

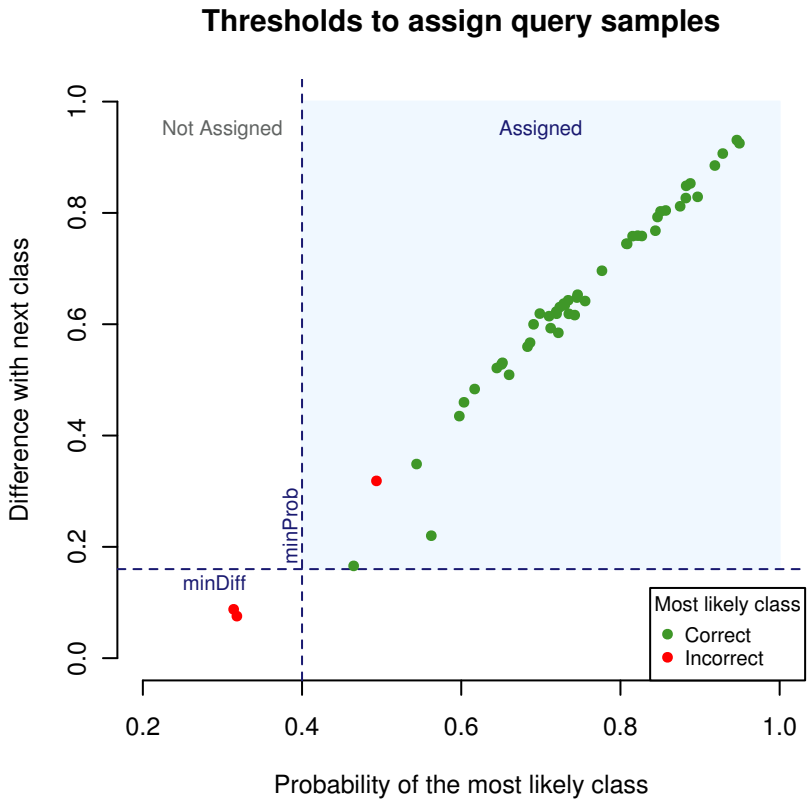


Figure 6. Assignment probabilities plot: It shows for each sample the probability of its most likely class *versus* the difference in probability with the next likely class. **Green** dots indicate that the probability of the most likely class is the correct class. **Red** dots indicate that the probability of the most likely class is not the correct class and, if assigned, such sample would have been missclassified. Dotted lines represent the chosen thresholds. The green area between them shows the samples that are actually assigned, those out of the green area are left as *NotAssigned*.

The plot in Figure 6 was obtained through the execution of *geNetClassifier()* with the leukemia's dataset. It shows that there are several samples under the assignment thresholds: these samples are left as *NotAssigned*. Out of these not assigned samples, the highest probability of some of them was to the real class (green), but some others was to an incorrect class (red). If the classifier had assigned the samples in red, it would have been an incorrect assignment.

5 Sample classification: query with new samples of unknown class

Once a classifier is built for a group of diseases or biological states, we can take external samples from new patients or new studies to query the classifier and know their class type.

Since we had 60 samples in the initial leukemia dataset and only 50 were used in the classifier, the 10 not used for training can be used as new samples to query the classifier and find out their class. In this case we will consider that the class of these samples is unknown.

```
> testSamples <- c(1:60)[-trainSamples]
```

queryGeNetClassifier() can then be used to ask the classifier about the class of the new samples.

```
> queryResult_AsUnkown <- queryGeNetClassifier(leukemiasClassifier,
+ leukemiasEset[,testSamples])
```

In the field *$class* of the return, we can see the class that each sample has been assigned to.

```
> names(queryResult_AsUnkown)

[1] "call"          "class"         "probabilities"

> queryResult_AsUnkown$class

GSM330195.CEL GSM330201.CEL GSM330611.CEL GSM330612.CEL GSM331037.CEL
          ALL           ALL           AML           AML           CLL
GSM331048.CEL GSM331392.CEL GSM331393.CEL GSM331675.CEL GSM331677.CEL
          CLL           CML           CML           NoL           NoL
Levels: ALL AML CLL CML NoL
```

If there were samples that had not been assigned to any class, they would be marked as*NotAssigned*. In the field *$probabilities*, we could see the probability of each sample to belong to each class. All these steps are very similar to the ones describes in section 4.1.

```
> t(queryResult_AsUnkown$probabilities[ ,
+ queryResult$class=="NotAssigned"])

     ALL AML CLL CML NoL
```

The function *querySummary()* provides a summary of the results by counting the number of samples that were assigned to each class and with which probabilities. It is a good way to have an overview of the classification results. In this case, the 100% *call rate* indicates that all samples have been assigned.

```
> querySummary(queryResult_AsUnkown, numDecimals=3)
```

```
$callRate
[1] 100

$assigned
    Count MinProb MaxProb  Mean    SD
ALL     2   0.721   0.825 0.773 0.073
AML     2   0.681   0.847 0.764 0.118
CLL     2   0.719   0.750 0.735 0.022
CML     2   0.564   0.879 0.721 0.223
NoL     2   0.745   0.886 0.816 0.100

$notAssigned
[1] "All samples have been assigned."
```

6 Functions to plot the results

6.1 Plot Ranked Significant Genes: *plot(...@genesRaking)*

As indicated in section 3.4.1, the default plot of a *genesRanking* can be obtained through the plot() function. This plot represents the gene rank obtained for each class *versus* the posterior probability of the genes.

```
> plot(leukemiasClassifier@genesRanking)
```

Some of the parameters to personalize this plot are:

- *lpThreshold* to set the value of the posterior probability threshold (marked as an horizontal line in the plot)
- *numGenesPlot* to determine the maximum number of genes that will be plot

```
> plot(leukemiasClassifier@genesRanking, numGenesPlot=3000,
+ plotTitle="5 classes: ALL, AML, CLL, CML, NoL", lpThreshold=0.80)
```

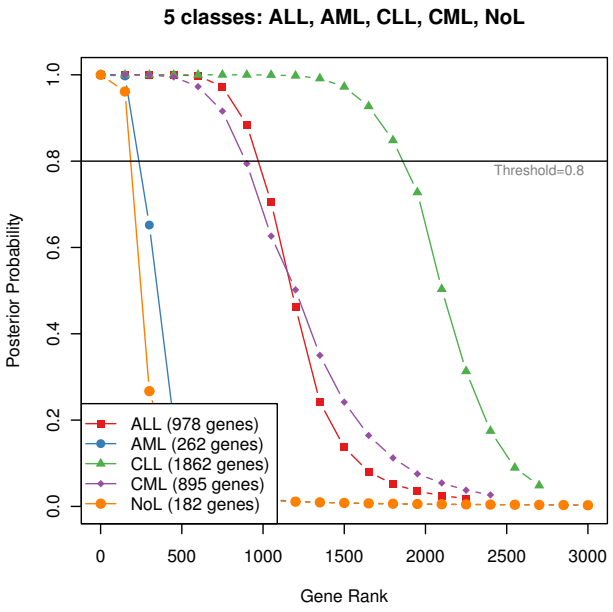


Figure 7. Plot of the posterior probabilities of the genes of 4 leukemia classes and the non-leukemia controls, ordering the genes according to their rank and setting the *lpThreshold* at 0.80.

calculateGenesRanking() allows to calculate (and plot) the ranking for a given data set without building the classifier:

```
> ranking <- calculateGenesRanking(leukemiasEset[,trainSamples],
+ "LeukemiaType")
```

6.2 Plot Gene Expression Profiles: *plotExpressionProfiles()*

The function *plotExpressionProfiles()* generates an overview of the expression profile of each gene along all the samples contained in the studied dataset. The plot will be saved as a PDF if *fileName* is indicated. The parameter *geneLabels* can be used to show a different name to the one included in the expression matrix (i.e. gene symbol instead of ENSEMBL ID or *Affymetrix* ID).

To plot the expression of 4 specific genes across the samples included in the leukemia's set:

```
> data(geneSymbols)
> topGenes <- getRanking(
+ getTopRanking(leukemiasClassifier@classificationGenes,numGenesClass=1),
+ showGeneID=TRUE)$geneID
> plotExpressionProfiles(leukemiasEset, topGenes[,c("ALL","AML"), drop=FALSE],
+     sampleLabels="LeukemiaType", geneLabels=geneSymbols)
```

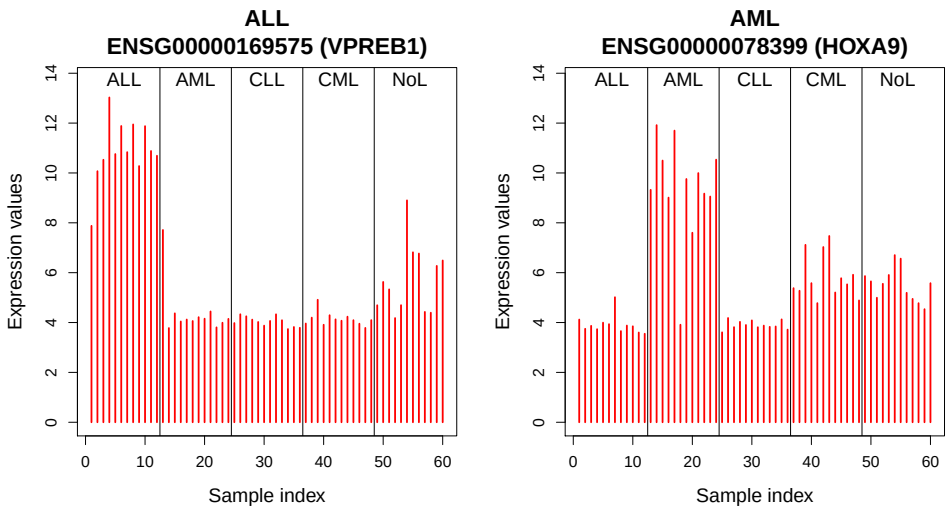


Figure 8. Plot of the expression profiles across 60 samples of 2 genes.

If a *geNetClassifierReturn* object is provided instead of a list of genes, it will plot the expression of all the genes used for training the classifier:

```
> plotExpressionProfiles(leukemiasEset[,trainSamples], leukemiasClassifier,
+     sampleLabels="LeukemiaType", fileName="leukExprs_trainSamples.pdf")
```

To plot the expression of all the genes chosen for classification for a specific class, for example AML:

```
> classGenes <- getRanking(leukemiasClassifier@classificationGenes,
+     showGeneID=TRUE)$geneID[,"AML"]
> plotExpressionProfiles(leukemiasEset, genes=classGenes,
+     sampleLabels="LeukemiaType", geneLabels=geneSymbols, fileName="AML_genes.pdf")
```

These plots can be modified in several ways, for example coloring specific samples or classes, or plotting the expression as boxplot

- Coloring specific samples or classes:

```
> plotExpressionProfiles(leukemiasEset, genes=topGenes[,3, drop=FALSE],
+                        sampleLabels="LeukemiaType",
+                        showMean=TRUE, identify=FALSE,
+                        sampleColors=c("grey","red")
+                        [(sampleNames(leukemiasEset)%in% c("GSM331386.CEL","GSM3313

> plotExpressionProfiles(leukemiasEset, genes=topGenes[,3, drop=FALSE],
+                        sampleLabels="LeukemiaType",
+                        showMean=TRUE, identify=FALSE,
+                        classColors=c("red","red", "blue","red","red"))
```

- Plotting the expression as boxplot (grouped by classes):

```
> plotExpressionProfiles(leukemiasEset, genes=topGenes[,3, drop=FALSE],
+                        sampleLabels="LeukemiaType",
+                        type="boxplot", geneLabels=geneSymbols, sameScale=FALSE)
```

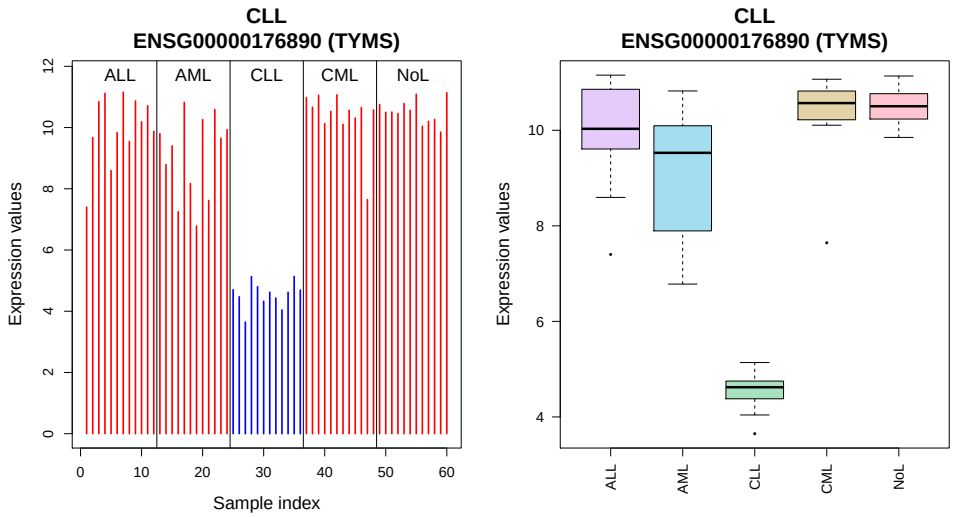


Figure 9. Two different versions of expression plot.

See *?plotExpressionProfiles* for more details.

6.3 Plot Genes Discriminant Power: *plotDiscriminantPower()*

The *discriminant power* is a parameter derived from the classifier's *support vectors* which resembles the power of each gene to mark the difference between classes.

The multi-class SVM algorithm (One-versus-One, OvO) produces a set of *support vectors* for each binary comparison between classes. Such *support vectors* include the *Lagrange coefficients* (alpha) for all the genes selected for the classification. Therefore, we can assign to each gene the sum of the *Lagrange coefficients* of all the *support vectors* of each class (represented as piled up bars in the plot). The *discriminant power* is then calculated as the difference between the value of the largest class and the closest (the distance marked by two red lines in the plot). In conclusion, the *discriminant power* is a parameter that allows the characterization of the genes based in their capacity to separate different classes (i.e. different diseases or diseases subtypes compared).

The *discriminant power* is calculated for each gene included in the classifier (the *@classificationGenes*) when it is built *geNetClassifier()*). The *plotDiscriminantPower()* function is included in the package to generate a graphic representation of the *discriminant power*.

```
> plotDiscriminantPower(leukemiasClassifier,
+ classificationGenes="ENSG00000169575")
```

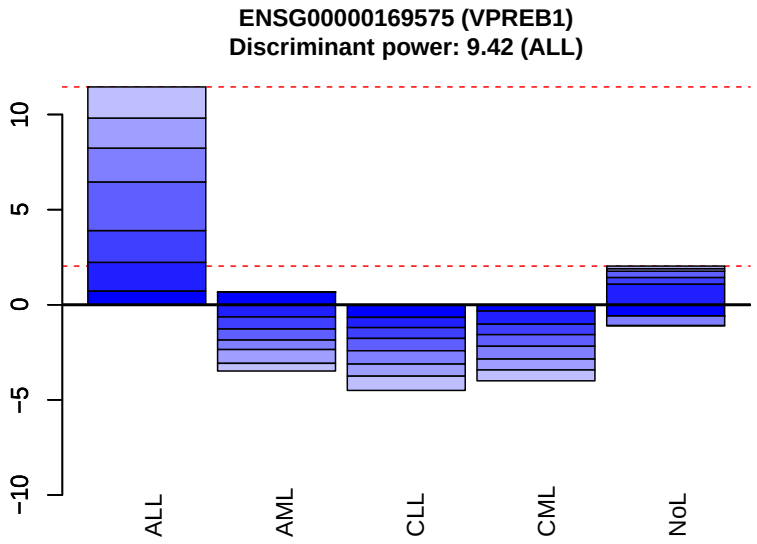


Figure 10. Plot of the discriminant power of gene VPREB1 (ENSG00000169575). The plot shows that this gene identifies class ALL and the closest class is NoL.

The next example shows the discriminant power of the top genes of a class. In order to plot more than 20 genes, or to save the plots as PDF, provide a *fileName*.

```
> discPowerTable <- plotDiscriminantPower(leukemiasClassifier,
+ classificationGenes=getRanking(leukemiasClassifier@classificationGenes,
+ showGeneID=TRUE)$geneID[1:4,"AML",drop=FALSE], returnTable=TRUE)
```

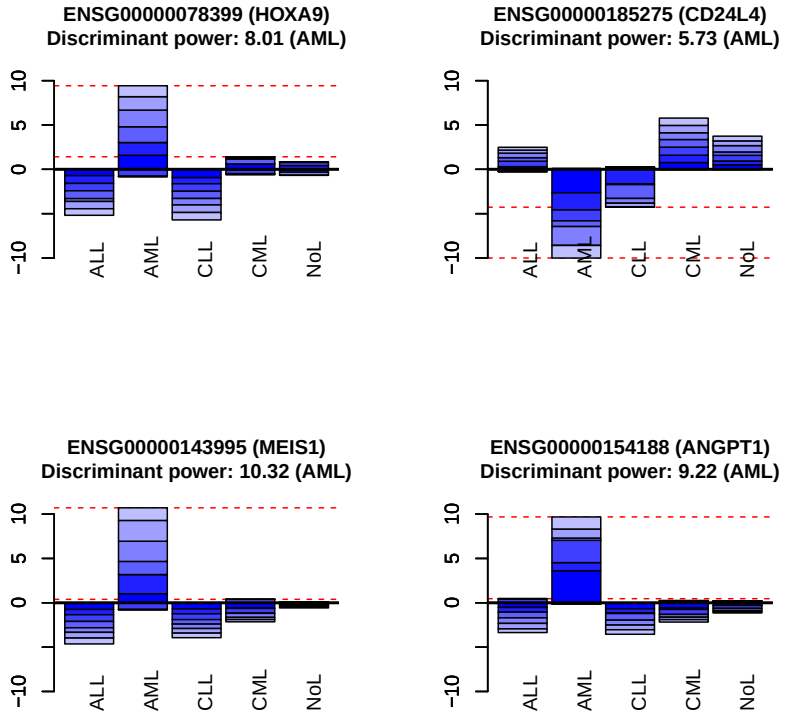


Figure 11. Plot of the discriminant power of the 4 genes that best discriminate AML class from the other classes. The figures indicate that MEIS1 (ENSG00000143995) presents the highest discriminant power. This gene encodes a homeobox protein that has been involved in myeloid leukemia. A high discriminant power can help to identify gene markers.

Some of the options to personalize the plot are *classNames* to provide a different name for the classes and textitgeneLabels to provide a alias for the genes. As usual, more details about the function are available at *?plotDiscriminantPower*.

6.4 Plot Gene Networks: *plotNetwork()*

The package also includes some functions to manipulate the networks produced by *geNetClassifier()* (i.e. select part of a network and personalize the plots).

Step 1: Select a network or sub-network.
getSubNetwork() allows to select sub-networks. i.e. the sub-network containing only the classification genes:

```
> clGenesSubNet <- getSubNetwork(leukemiasClassifier@genesNetwork,
+ leukemiasClassifier@classificationGenes)
```

Step 2: Get the info of the genes to plot.
genesDetails() provides the available information about the genes. This information can be shown in the network: The gene name will be the node label. The expression of the gene will be shown with the node color, and the discriminant power will determine its size. In case the network includes genes selected for classification and genes which were not selected, the genes selected for classification will be plot as squares and the not selected as circles (only available for PDF plot, not on the dynamic view). For more details see the network legend in figure 14.

```
> clGenesInfo <- genesDetails(leukemiasClassifier@classificationGenes)
```

Step 3: Plot the network.
The network plots can be produced either using R interactive view (*tkplot* from *igraph*) or plotted as saved PDF files. Use *plotType="pdf"* to save the network as a static PDF file. This option is recommended to produce an overview of several networks. To produce interactive networks skip this argument. Iteractive plotscan be exported as a *postscript* files (.eps).

Some plot examples:
Network of ALL classification genes:

```
> plotNetwork(genesNetwork=clGenesSubNet$ALL, genesInfo=clGenesInfo)
```

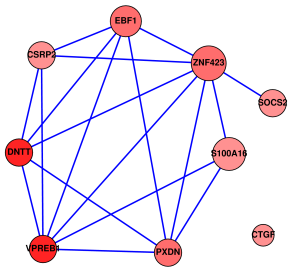


Figure 12. Gene network obtained for class ALL including the 9 classification genes selected for this disease.

Only connected nodes from ALL classification genes network:

```
> plotNetwork(genesNetwork=clGenesSubNet$ALL, genesInfo=clGenesInfo,
+ plotAllNodesNetwork=FALSE, plotOnlyConnectedNodesNetwork=TRUE)
```

AML network of the top 30 genes from the ranking (calculated as co-expression and mutual information):

```
> top30g <- getRanking(leukemiasClassifier@genesRanking,
+ showGeneID=TRUE)$geneID[1:30,]
> top30gSubNet <- getSubNetwork(leukemiasClassifier@genesNetwork, top30g)
> top30gInfo <- lapply(genesDetails(leukemiasClassifier@genesRanking),
+ function(x) x[1:30,])
> plotNetwork(genesNetwork=top30gSubNet$AML, genesInfo=top30gInfo$AML)
```

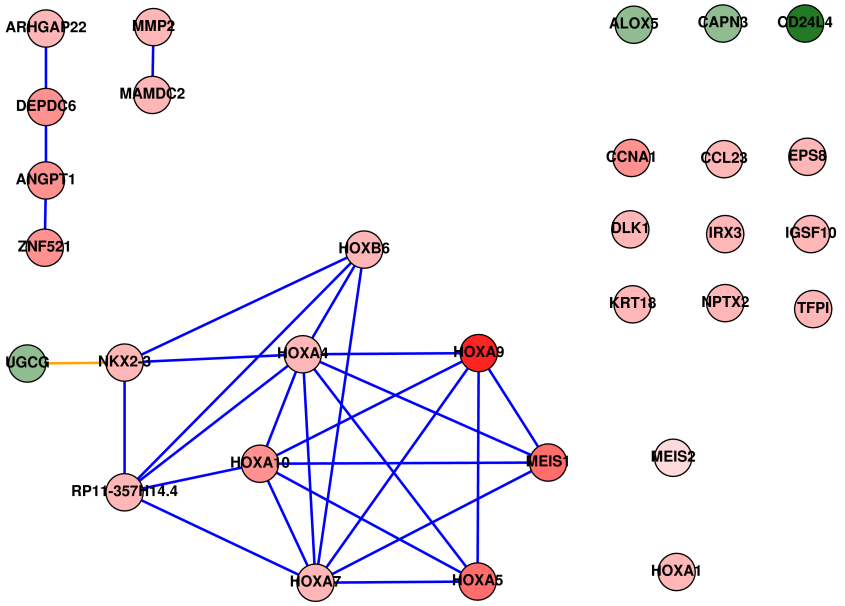


Figure 13. Gene network obtained for class AML including the top 30 genes selected from the gene ranking of this disease.

Network of the top 100 genes from AML ranking.
A preview of this network is automatically plotted for every class by *geNetClassifier()* if *plotsName* is provided.

```
> top100gRanking <- getTopRanking(leukemiasClassifier@genesRanking,
+ numGenes=100)
> top100gSubNet <- getSubNetwork(leukemiasClassifier@genesNetwork,
+ getRanking(top100gRanking, showGeneID=TRUE)$geneID)
> plotNetwork(genesNetwork=top100gSubNet,
+ classificationGenes=leukemiasClassifier@classificationGenes,
+ genesRanking=top100gRanking, plotAllNodesNetwork=TRUE,
+ plotOnlyConnectedNodesNetwork=TRUE, labelSize=0.4,
+ plotType="pdf", fileName="leukemiasNetwork")
```

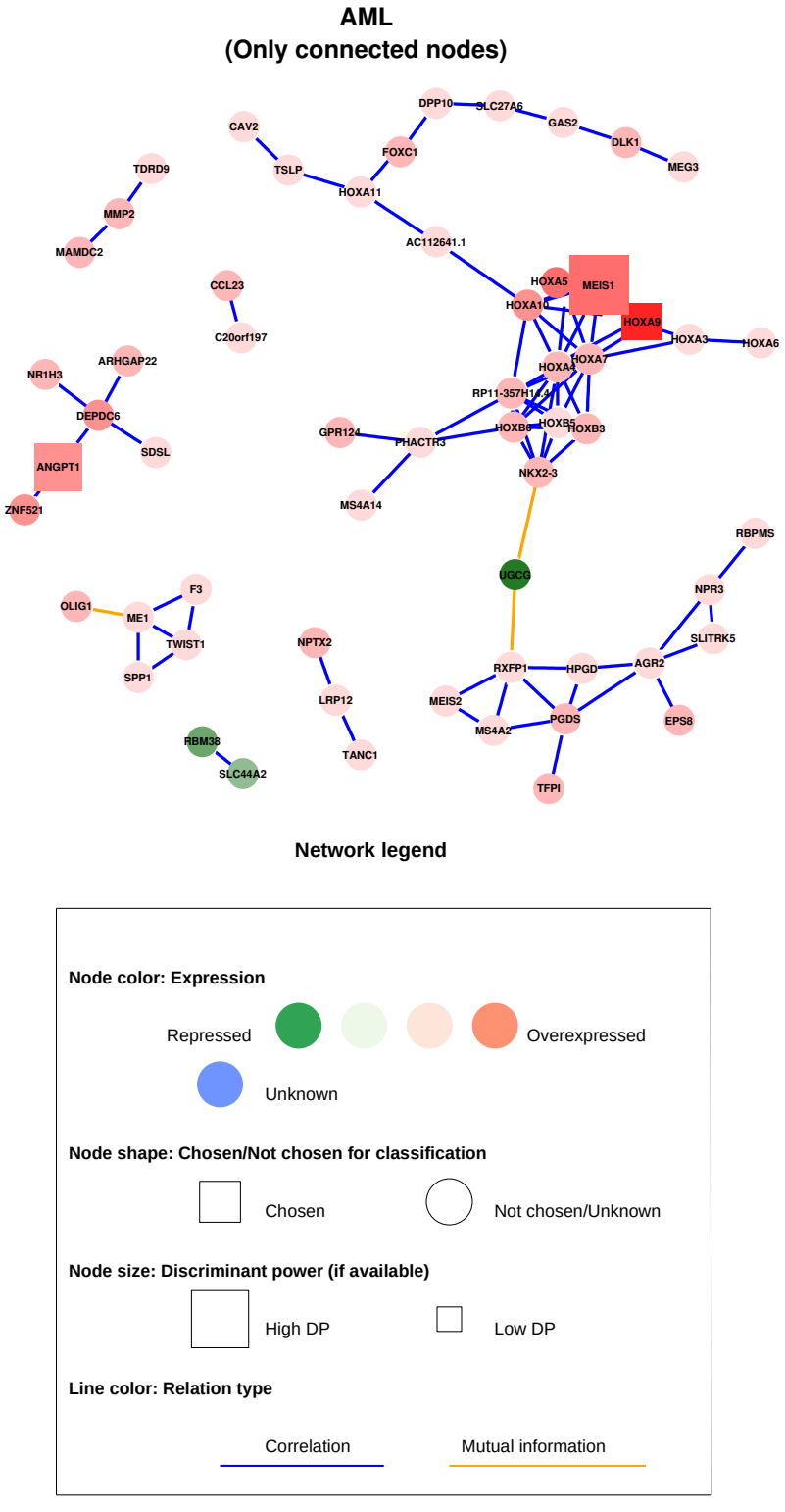


Figure 14. Gene network obtained for class AML selecting the 100 top genes from the gene ranking of this disease, but presenting only the connected nodes. The figure also includes the network legend indicating the meaning of the shapes and colors given to the nodes and edges.

Acknowledgements

This work was supported by Instituto de Salud Carlos III and by a grant from the Junta de Castilla y Leon and the European Social Fund to S.A and C.D.

References

[1] Haferlach T, Kohlmann A, Wieczorek L, Basso G, Kronnie GT, Bene MC, De Vos J, Hernandez JM, Hofmann WK, Mills KI, Gilkes A, Chiaretti S, Shurtleff SA, Kipps TJ, Rassenti LZ, Yeoh AE, Papenhausen PR, Liu WM, Williams PM, Foa R (2010). *Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the International Microarray Innovations in Leukemia Study Group.* J Clin Oncol. 28: 2529-2537.

[2] Barrier A, Lemoine A, Boelle PY, Tse C, Brault D, Chiappini F, Breittschneider J, Lacaine F, Houry S, Huguier M, Van der Laan MJ, Speed T, Debuire B, Flahault A, Dudoit S (2005) *Colon cancer prognosis prediction by gene expression profiling.* Oncogene. 24: 6155-6164.

[3] Meyer D, Leischa F, Hornik K (2005). *The supportvector machine under test.* Neurocomputing. 55: 169-186

[4] Meyer PE, Lafitte F, Bontempi G (2008). *minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information.* BMC Bioinformatics. 9: 461.

[5] Risueno A, Fontanillo C, Dinger ME, De Las Rivas J (2010). *GATExplorer: genomic and transcriptomic explorer; mapping expression probes to gene loci, transcripts, exons and ncRNAs.* BMC Bioinformatics. 11: 221.

[6] Morris C (1983). *Parametric empirical Bayes inference: theory and applications.* JASA. 78: 47-65.

[7] Kendziorski CM, Newton MA, Lan H, Gould MN (2003). *On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles.* Statistics in Medicine 22: 3899-3914.

[8] Benjamini Y, Hochberg Y (1995). *Controlling the false discovery rate: A practical and powerful approach to multiple testing.* J. Roy. Statist. Soc. Ser. B. 57: 289-300.

Chapter 2

Article 2: Functional Gene Networks: R/Bioc package to generate and analyse gene networks derived from functional enrichment and clustering

Authors: Sara Aibar, Celia Fontanillo, Conrad Droste and Javier De Las Rivas

Publication status: *Bioinformatics* (2015) doi: 10.1093/bioinformatics/btu864 PMID: 25600944

Associated Bioconductor package: *FGNet: Functional Gene Networks derived from biological enrichment analyses*. Published on October 2013.
URL: http://bioconductor.org/packages/release/bioc/html/FGNet.html

Bioinformatics Advance Access published February 4, 2015

Bioinformatics, 2015, 1–3
doi: 10.1093/bioinformatics/btu864
Advance Access Publication Date: 18 January 2015
Applications Note

Systems biology

Functional Gene Networks: R/Bioc package to generate and analyse gene networks derived from functional enrichment and clustering

Sara Aibar, Celia Fontanillo†, Conrad Droste and Javier De Las Rivas*

Bioinformatics and Functional Genomics Research Group, Cancer Research Center (Consejo Superior de Investigaciones Científicas, Universidad de Salamanca and Instituto de Investigación Biomédica de Salamanca, CSIC/USAL/IBSAL), Salamanca, Spain

*To whom correspondence should be addressed.
†Present address: Celgene Institute for Translational Research Europe (CITRE), Sevilla, Spain
Associate Editor: Jonathan Wren

Received on June 20, 2014; revised on December 16, 2014; accepted on December 29, 2014

Abstract

Summary: Functional Gene Networks (*FGNet*) is an R/Bioconductor package that generates gene networks derived from the results of functional enrichment analysis (FEA) and annotation clustering. The sets of genes enriched with specific biological terms (obtained from a FEA platform) are transformed into a network by establishing links between genes based on common functional annotations and common clusters. The network provides a new view of FEA results revealing gene modules with similar functions and genes that are related to multiple functions. In addition to building the functional network, *FGNet* analyses the similarity between the groups of genes and provides a distance heatmap and a bipartite network of functionally overlapping genes. The application includes an interface to directly perform FEA queries using different external tools: *DAVID*, *GeneTerm Linker*, *TopGO* or *GAGE*; and a graphical interface to facilitate the use.
Availability and implementation: *FGNet* is available in Bioconductor, including a tutorial. URL: http://bioconductor.org/packages/release/bioc/html/FGNet.html
Contact: jrivas@usal.es
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Due to the increasing number of omic studies, efficient functional analysis of large lists of genes or proteins is essential to understand the biological processes in which they are involved. Functional enrichment analysis (FEA) is the most popular bioinformatic methodology to obtain significant functional information from sets of cooperating genes. FEA methods search in biological databases (such as Gene Ontology and KEGG pathways, among others) and use statistical testing to find biological terms and functional annotations that are significantly enriched in a list of genes. However, in most cases the results of these analyses are very long lists of biological terms associated to genes that are difficult to digest and interpret. Some tools cluster the FEA results, like *DAVID-FAC* (Huang *et al.*, 2009) and *GeneTerm Linker* (Fontanillo *et al.*, 2011), but their output is provided as large tables and there are not many tools to integrate and visualize these results. Here we present Functional Gene Networks (*FGNet*), an R/Bioconductor package that uses FEA results to perform network-based analyses and visualization. The main network is built by establishing links between genes annotated to similar functional terms. In this way, *FGNet* generates and provides a network representing the links and associations between the clusters of genes and enriched terms. The network summarizes and facilitates the interpretation of the biological processes significantly enriched in the initial list of genes, revealing important information such as: distance and overlap

between clusters, identification of modules and hubs. The tool can also help to disclose new associations among genes cooperating in hidden biological processes not annotated yet, which can be revealed by the topology of the functional network.

2 Methods

2.1 Input: functional enrichment and clustering

FGNet builds functional networks based on the groups obtained from clustering *gene-term sets* (*gtsets*, genes and terms associated by an enrichment p-value) returned by a FEA. The package includes an interface to do queries with gene lists using four FEA tools: *DAVID* with *Functional Annotation Clustering* (that returns clustered *gtsets*, Cl); *GAGE* (that also provides clusters) (Luo *et al.*, 2009); *GeneCodis* with *GeneTerm Linker* (that returns metagroups, Mg) and *TopGO* (that only returns *gtsets*) (Alexa *et al.*, 2010). The package can be also applied to the results from other EA tools, as long as the input results are transformed into tables of genes and associated terms.

2.2 Construction of the functional network

The functional network is built based on the analysis of all the *gtsets* provided by the FEA tool. These sets allow to generate a boolean matrix M of *genes* by *gtsets*, in which each element $m_{g,s} = 1$ if gene g is in set s. This membership matrix is then transformed into an adjacency matrix A $n \times n$; being n the total number of genes and $a_{i,j}$ the number of *gtsets* s in which a gene-pair is included: $a_{i,j} = \sum_s (m_{i,s} \times m_{j,s})(1 - \delta_{i,j})$, where δ is a *Kronecker* delta ($\delta_{i,j} = 1$ if $i = j$, $\delta_{i,j} = 0$ if $i \neq j$). This adjacency matrix is used to generate the functional network by establishing a weighted link between each pair of genes (g_i, g_j) in which $a_{i,j} \neq 0$. Finally, the clustering of *gtsets* provided by the FEA tool is used to generate a second genes' adjacency matrix with the number of common clusters/metagroups (Fig. 1A), that is used to define and allocate gene groups. The network produced is provided as an *igraph* object for further analysis, and can be exported to other network-based tools like *Cytoscape*.

2.3 Visualization and plots of the functional network

The main plot of the network presents the functionally associated genes (Fig. 1B). Edges link the genes that are in the same *gtsets*. Nodes within the same Cl/Mg are placed together using a force-directed *Fruchterman–Reingold* layout, within a common background colour. Genes in only one Cl/Mg are plotted with the colour of such group, while genes that are included in more than one Cl/Mg are left white.

2.4 Analysis of functional modules in the network

To facilitate the analysis and quantification of the modules and the overlap between groups, *FGNet* also provides a distance matrix and a heatmap (Fig. 1C), plus an intersection network (Fig. 1D). The distance matrix is calculated based on the pairwise binary distance in the adjacency matrix of common Cls/Mgs. These distances are analysed by hierarchical average linkage and plotted as a heatmap that reveals the proximity and similarity between the groups of genes (Cls/Mgs). The intersection network is a bipartite network which includes only the genes associated to several Cls/Mgs (white nodes in Fig. 1B,D), showing their connectivity to such Cls/Mgs. This intersection network facilitates the identification of *multifunctional* genes. (For more details see *FGNet* documentation in Bioconductor).

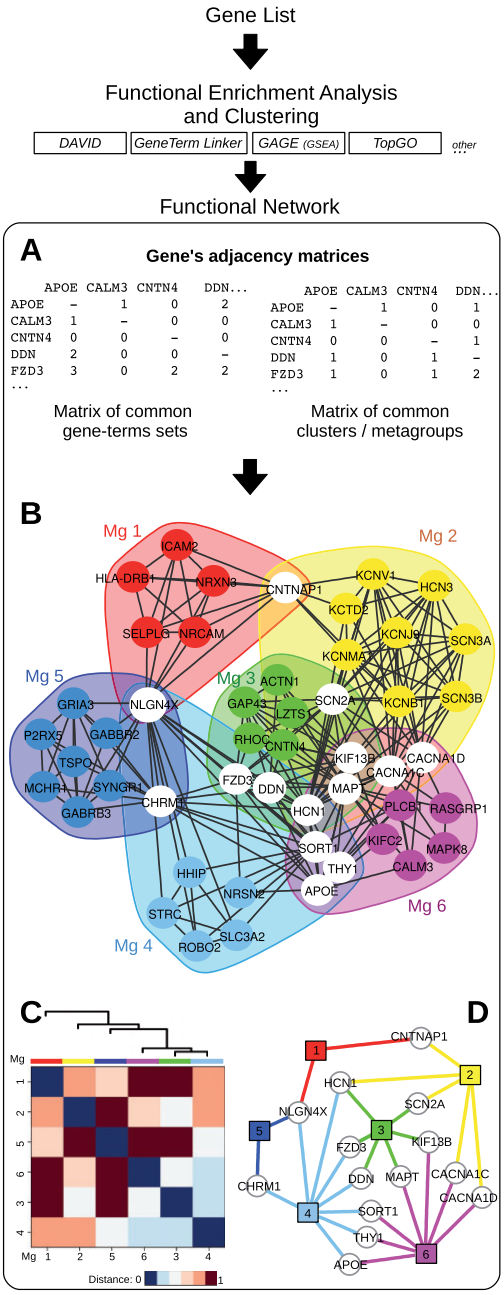


Fig. 1. Schematic workflow. A query gene list is analysed through a FEA tool and the generated *gene-term sets* are used to build: (**A**) gene's adjacency matrices; (**B**) a functional network (general view); (**C**) a distance heatmap and (**D**) an intersection network (to highlight multifunctional genes)

3 Example of use

We applied the method to several datasets and confirmed that the functional network greatly facilitates the analysis of enrichment results. Figure 1 shows the results of *FGNet* for a list of 175 genes

differentially expressed in human samples of entorhinal cortex neurons from Alzheimer's disease (AD) patients (obtained from Gene Expression Omnibus database, GEO: dataset GSE4757). Performing a FEA through *GeneTerm Linker*, we obtained six meta-groups that we labelled according to their main annotations: (Mg1) cell adhesion; (Mg2) voltage-gated ion/potassium channels; (Mg3) axon and cell projection; (Mg4) dendrite and neuronal cell body; (Mg5) synaptic neuroactive ligand-receptor interaction and (Mg6) MAPK signaling and Alzheimer. The network of these six Mgs (Fig. 1B) provides a global overview of the functionally overlapping genes and allows to identify hub genes that interconnect groups. For example, CNTNAP1 and NLGN4X appear as hubs in Mg1. CNTNAP1 (that regulates distribution of K+ channels) links Mg1 and 2; and NLGN4X (that facilitates synaptic neurotransmission) links Mg1 with 4 and 5. NLGN4X is the gene with highest betweenness centrality in this network. Another important hub is APOE, recently associated to Alzheimer. The distance matrix (Fig. 1C) allows to quantify the similarity between gene groups, showing that the closest Mgs are 3, 4 and 6, sharing eight nodes. This is also presented in the intersection network (Fig. 1D). Finally, the functional network can reveal further information about some Mgs. For example, if a Mg shares many genes with several other Mgs, it will indicate that such Mg brings the most common features that define the studied biological state. This is the case for Mg6, which, in fact, is annotated to Alzheimer's Disease.

Funding

This work was supported by the "Accion Estrategica en Salud" (AES) of the "Instituto de Salud Carlos III" (ISCiii) from the Spanish Government (projects granted to J.D.L.R.: PS09/00843 and PI12/00624); and by the "Consejeria de Educación" of the "Junta Castilla y Leon" (JCyL) and the European Social Fund (ESF) with grants given to S.A. and C.D.

Conflict of Interest: none declared.

References

Alexa,A. and Rahnenfuhrer,J. (2010). topGO: Enrichment analysis for Gene Ontology. R package version 2.18.0.

Fontanillo,C. *et al.* (2011). Functional analysis beyond enrichment: non-redundant reciprocal linkage of genes and biological terms. *PLoS One,* **6**, e24289.

Huang,D. *et al.* (2009). Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources. *Nat. Protoc.*, **4**, 44–57.

Luo,W. *et al.* (2009). GAGE: generally applicable gene set enrichment for pathway analysis. *BMC Bioinformatics,* **10**, 161.

Aditional files

Aditional file 1: FGNet package vignette

Also available at *Bioinformatics* online and *Bioconductor.*

URLs:

http://bioinformatics.oxfordjournals.org/lookup/suppl/doi:10.1093/bioinformatics/btu864/-/DC1

http://bioconductor.org/packages/release/bioc/vignettes/FGNet/inst/doc/FGNet.html

FGNet
Functional Gene Networks derived from biological enrichment analyses

Sara Aibar, Celia Fontanillo, Conrad Droste, and Javier De Las Rivas

Bioinformatics and Functional Genomics Group
Centro de Investigacion del Cancer (CiC-IBMCC, CSIC/USAL)
Salamanca - Spain

December 10, 2014

Version: 3.0

Contents

1 Introduction to FGNet

FGNet allows to perform a Functional Enrichment Analysis (FEA) on a list of genes or expression set, and transform the results into networks. The resulting functional networks provide an overview of the biological functions of the genes/terms, and allows to easily see links between genes, overlap between clusters, finding key genes, etc.

FGNet takes as input a query list of genes selected by the user, and builds and displays networks of genes based in the existence of common functional terms that are enriched in certain subsets of genes of the list. By doing this, the tool allows to disclose groups/clusters of genes that have similar annotations and so they may have similar biological function in the cell. The discovery of molecular machines or functional modules within the cell (i.e. genes or proteins that work together to perform a biological process in the cells) is essential in modern molecular medicine and systems biology, because many times we do not know which are the gene partners playing in the same roles in a pathological state. FGNet is a tool that helps to create functional connections between different genes/proteins based on annotations. By grouping similar, redundant and homogeneous annotation content from the same or different biological resources into gene-term groups, the biological interpretation of large gene lists moves from a gene centric approach (where each gene is independent) to a functional-module centric approach (where the genes are interconnected). In this way, FGNet can provide a better representation of complex biological processes and reveal associations between genes.

Biological functional analysis

After obtaining a list of genes or proteins from an experiment or omic studies (microarrays, RNAseq, mass spectrometry, etc), the next step is usually to perform a functional analysis of the genes to search for the biological functions or processes in which they are involved. In order to facilitate the analysis of large lists of genes, multiple functional enrichment tools have been developed. These tools search for the genes in biological databases (i.e. GO, Kegg, Interpro), and test whether any biological annotations are over-represented in the query gene list compared to what would be expected in the whole population. However, the raw output from a functional enrichment analysis often provides dozens or hundreds of terms, and it still requires a lot of time and attention to go through the whole list of genes and annotations. A way to simplify this task is grouping genes and terms which often appear together and create associated networks: the Functional Networks.

FGNet builds the functional networks, based on data from a previous functional enrichment analysis (FEA). The package provides the functions to perform the FEA through four specific tools:

- **DAVID** with Functional Annotation Clustering (DAVID-FAC), which measures relationships among annotation terms based on their co-association with subsets of genes within the query gene list (Huang et al.). This type of clustering mostly results in groups of highly related terms, such as synonymous annotations from different annotation spaces (i.e. term “glycolysis” in KEGG and GO-BP), which also share most of their genes. This tool provides great coverage but does not avoid redundant terms and very general terms (like “signal transduction” or “regulation of transcription” that correspond to specific terms in Gene Ontology, GO).

- **GeneTerm Linker**, a post-enrichment tool, which focuses on clearing and sorting the results from a previous modular enrichment analysis. This is achieved by filtering general terms with low information content (i.e. *cellular process* or *protein binding*) and redundant annotations (i.e. *metabolic process* and *primary metabolic process*). The remaining gene-term sets are grouped into `metagroups` based on their shared genes and terms (using a reciprocal linkage approach) (Fontanillo et al.).
- **TopGO** (Alexa et al.), an enrichment analysis tool based on Gene Ontology (GO) that tests GO terms while accounting for the topology of the GO graph to eliminate local similarities and dependencies between GO terms. TopGO does not provide clusters, and therefore the functional network is built using only the gene-term sets. TopGO can be applied off-line.
- **GAGE** (Luo et al.), a gene set enrichment analysis (GSEA) tool. It searches for functional enrichment in gene sets (i.e. KEGG pathways, Reactome, GO) and allows including a signal value -like expression changes- to rank the genes and then to identify the enrichment in functional terms that are altered (i.e. changed in genes UP and DOWN) or altered consistently in one direction (UP or DOWN). GAGE also clusters the resulting enriched gene-term sets and can be applied off-line.

To build the network based on other *other tools*, the raw output should be saved into a text file which contains the enriched terms and their genes. (For more details see function `format_results()`).

Functional network

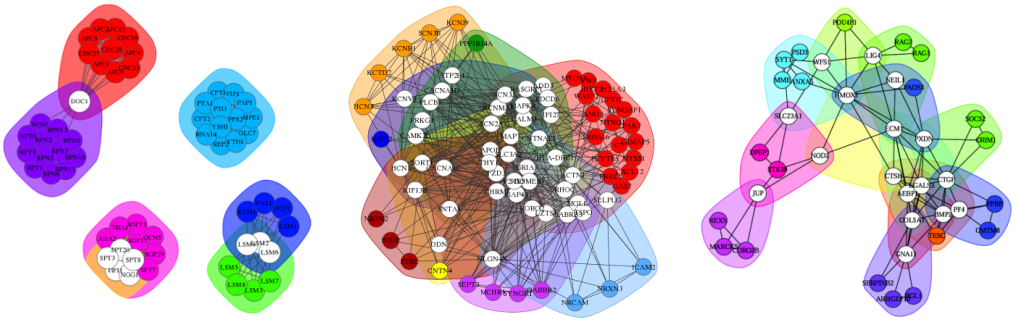
The **functional network** is the representation of the results from a functional enrichment analysis.

In the **default** network, all the nodes of the network are of the same type, i.e. genes OR terms, which are linked to each other if they are in the same gene-term set. In the plot, the genes/terms in the same groups (metagroups or clusters) are surrounded by a common background color.

In the **bipartite** network, the nodes are of two types, allowing to link the genes or terms, with the clusters they belong to. This network, can be built as an *intersection network* , a simplified functional network where all the genes/terms that belong to only one metagroup are clustered into a single node. This simplified network contains only the nodes in several groups.

In addition to the networks, FGNet also provides a few functions for further analysis. These functions allow to get a **distance matrix** , which represents the similarity between the groups based on the genes they share with each other (binary distance), and the distribution of **degree and betweenness** within the network and subnetworks, in order to find the most important genes (hubs).

All these functionalities can be accessed directly through the appropriate functions or the graphical user interface (GUI). In addition, FGNet also allows to generate an **HTML report** with an overview of these plots and analyses for a specific gene list.



Examples of funtional network for different analyses.

2 Installation

To install *FGNet* from *Bioconductor*, type in your R console:

```
source("http://bioconductor.org/biocLite.R")
biocLite("FGNet")
```

To reduce system requirements, only the minimum packages are required to execute FGNet. However, there are several functionalities that require further packages. i.e. the Graphical User Interface (GUI) requires "RGtk2", the FEA analyses might require "RDAVIDWebService", "gage", "topGO" or some annotation packages... etc.

To make sure all FGNet functionalities are available, install the following packages:

```
biocLite(c("RGtk2", "RCurl",
    "RDAVIDWebService", "gage", "topGO", "KEGGprofile",
    "GO.db", "KEGG.db", "reactome.db", "org.Sc.sgd.db"))
```

3 Creating a network from a list of genes/proteins

To generate a functional network with FGNet:

1. Perform a Functional Enrichment Analysis (FEA) on a list of genes or expression set.

FEA tool	Online?	Input	Annotations
DAVID	Yes	Gene list	Many
Gene-Term Linker	Yes	Gene list	GO, KEGG, Interpro
TopGO	No	Gene list	GO
GAGE (GSEA)	No	Expression set	Any gene set

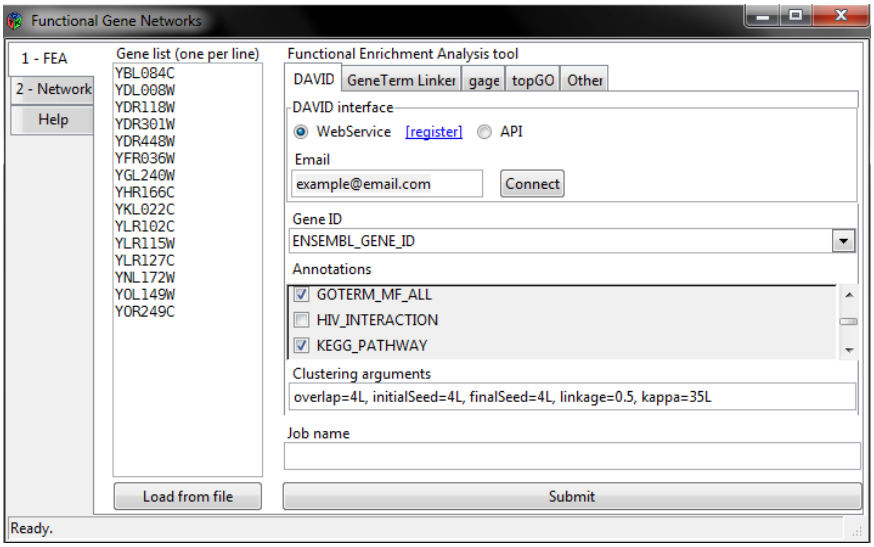
2. Create an HTML report with multiple views of the networks and analyses.
3. Personalize or analyze an specific network.

These steps are integrated into the Graphical User Interface (GUI), which provides access to the main functionalities of FGNet.

3.1 Graphical User Interface (GUI)

The Graphical User Interface (GUI) provides access to most FGNet functionalities in Windows and Linux (The current version of the GUI is not available for Mac OS X Snow Leopard).
To launch the GUI, type in the R console:

```
library(FGNet)
FGNet_GUI()
```



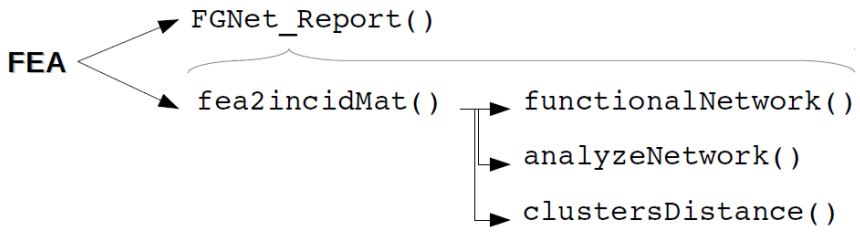
In case you already have a gene list or gene expression from a previous analysis, it is possible to load it directly into the GUI genes field by passing it as argument:

```
geneExpr <- c("YBL084C", "YDL008W", "YDR118W", "YDR301W", "YDR448W",
    "YFR036W", "YGL240W", "YHR166C", "YKL022C", "YLR102C", "YLR115W",
    "YLR127C", "YNL172W", "YOL149W", "YOR249C")
geneExpr <- setNames(c(rep(1,10),rep(-1,5)), geneExpr)

FGNet_GUI(geneExpr)
```

3.2 In R code...

The first step in the workflow is always is to perform a Functional Enrichment Analysis (FEA) on a list of genes or expression set.
Once the FEA is ready, you can proceed to generate the HTML report or the individual network/analyses:

FGNet 7



For help or more details on any functions or their arguments, just set a ? before its name.

```
?FGNet_report
```

3.2.1 Functional Enrichment Analysis (FEA)

Since the arguments required to perform the FEA depends on the tool, there are several FEA functions:

FEA tool	Function	Output group type
DAVID	`fea_david()`	Clusters
TopGO	`fea_topGO()`	No grouping
Gene-Term Linker	`fea_gtLinker()` & `fea_gtLinker_getResults()`	Metagroups
GAGE	`fea_gage()`	Clusters
Other	`format_feaResults()`	

All the FEA functions and `FGNet_report()` save the results in the current working directory.

```
getwd()
```

Here is an example analyzing a gene list with the different tools:

```
genesYeast <- c("ADA2", "APC1", "APC11", "APC2", "APC4", "APC5", "APC9",
    "CDC16",  "CDC23", "CDC26", "CDC27", "CFT1", "CFT2", "DCP1", "DOC1",
    "FIP1", "GCN5", "GLC7", "HFI1", "KEM1", "LSM1", "LSM2", "LSM3",
    "LSM4", "LSM5", "LSM6", "LSM7", "LSM8", "MPE1", "NGG1", "PAP1",
    "PAT1", "PFS2", "PTA1", "PTI1", "REF2", "RNA14", "RPN1", "RPN10",
    "RPN11", "RPN13", "RPN2", "RPN3", "RPN5", "RPN6", "RPN8", "RPT1",
    "RPT3", "RPT6", "SGF11", "SGF29", "SGF73", "SPT20", "SPT3", "SPT7",
    "SPT8", "TRA1", "YSH1", "YTH1")

library(org.Sc.sgd.db)
geneLabels <- unlist(as.list(org.Sc.sgdGENENAME))
genesYeast <- sort(geneLabels[which(geneLabels %in% genesYeast)])

# Optional: Gene expression (1=UP, -1=DW)
genesYeastExpr <- setNames(c(rep(1,28), rep(-1,30)),genesYeast)
```

DAVID
Using DAVID requires internet connection. In addition, we recommend to register at `http://david.abcc.ncifcrf.gov/webservice/register.htm` to perform the queries through its Web Service.

By default, `geneIdType="ENSEMBL_GENE_ID"`. To replace the gene IDs by readable names in the plots and HTML report, use the argument `geneLabels`. To see the gene IDs supported by DAVID's Web Service, use: `getIdTypes(DAVIDWebService$new(email=...))`.

```
feaResults_David <- fea_david(names(genesYeast), geneLabels=genesYeast,
                              email="example@email.com")
?fea_david
```

TopGO
Since TopGO uses local databases, it does not require internet connection.

The results from topGO are provided as individual gene-term sets not grouped into clusters. FGNet treats each *gene-term set* as a single cluster.

```
feaResults_topGO <- fea_topGO(genesYeast,
    geneIdType="GENENAME", organism="Sc")
?fea_topGO
```

Gene-Term Linker
Since the analysis with Gene-Term Linker usually takes several minutes to be ready, the workflow is divided in two steps: (1) sending the analysis request, and (2) retrieving the results:

```
jobID <- fea_gtLinker(geneList=genesYeast, organism="Sc")
?fea_gtLinker
```

once the analysis is ready...

```
jobID <- 3907019
feaResults_gtLinker <- fea_gtLinker_getResults(jobID=jobID, organism="Sc")
```

GAGE
As a GSEA approach, instead of performing the functional enrichment over a gene list, gage requires a raw expression set and the samples to compare:

```
library(gage)
data(gse16873)

# Set gene labels? (they need to have unique identifiers)
library(org.Hs.eg.db)
geneSymbols <- select(org.Hs.eg.db,columns="SYMBOL",keytype="ENTREZID",
    keys=rownames(gse16873))
```

```
geneLabels <- geneSymbols$SYMBOL
names(geneLabels) <- geneSymbols$ENTREZID
head(geneLabels)

# GAGE:
feaResults_gage <- fea_gage(eset=gse16873,
                            refSamples=grep('HN',colnames(gse16873)),
                            compSamples=grep('DCIS',colnames(gse16873)),
                            geneLabels=geneLabels, annotations="REACTOME",
                            geneIdType="ENTREZID", organism="Hs")
?fea_gage
```

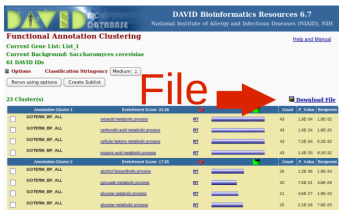
Other tools
To import the results from a functional enrichment analysis performed with other tools, see:

```
?format_results()
```

Web analysis
FGNet can also be applied to an analysis performed at DAVID and GeneTerm Linker web site:

- DAVID: `http://david.abcc.ncifcrf.gov` (Functional Annotation Clustering Tool)
- GeneTerm Linker: `http://gtlinker.cnb.csic.es`

To import these results into FGNet, use DAVID's `download file` or GeneTerm linker's `job ID`, and the functions `format_david()` or `fea_gtLinker_getResults()`:




```
feaResults_David <- format_david(
    "http://david.abcc.ncifcrf.gov/data/download/90128.txt")
feaResults_gtLinker <- fea_gtLinker_getResults(jobID=3907019)
```

3.2.2 HTML report

The HTML report function allows to create a comprehensive report including different views of the Functional Network, the cluster/metagroup legend, and some further statistics directly directly from a gene list.

Here is the code to use `FGNet_report()` with each of the previous examples:

```
FGNet_report(feaResults_David, geneExpr=genesYeastExpr, plotKeggPw=FALSE)
FGNet_report(feaResults_topGO, geneExpr=genesYeastExpr)
FGNet_report(feaResults_gtLinker, geneExpr=genesYeastExpr)
FGNet_report(feaResults_gage)
```

By default, the clusters included in these reports are filtered out to get cleaner results. The default values depend on the tool, and can be modified through FGNet_report arguments:

```
data(FEA_tools)
FEA_tools[,4:6]
```

```
FGNet_report(feaResults_gtLinker, filterThreshold=0.3)
```

```
?FGNet_report
```

3.2.3 Individual networks

After the FEA is ready, it is also possible to generate specific networks rather than the full report. Here is a simple example on how to use `fea2incidMat()` to generate the incidence matrices that represent the networks and plot them. There are more detailed examples on how to edit and explore the networks in sections *editing and creating new networks* (sec. 4) and *filtering and selecting clusters* (sec. 5).

```
feaResults <- feaResults_gtLinker
incidMat <- fea2incidMat(feaResults)
incidMat$metagroupsMatrix[1:5, 1:5]
```

```
##       1 2 3 4 5
## ADA2  0 1 0 0 0
## APC1  0 0 1 0 0
## APC11 0 0 1 0 0
## APC2  0 0 1 0 1
## APC4  0 0 1 0 0
```

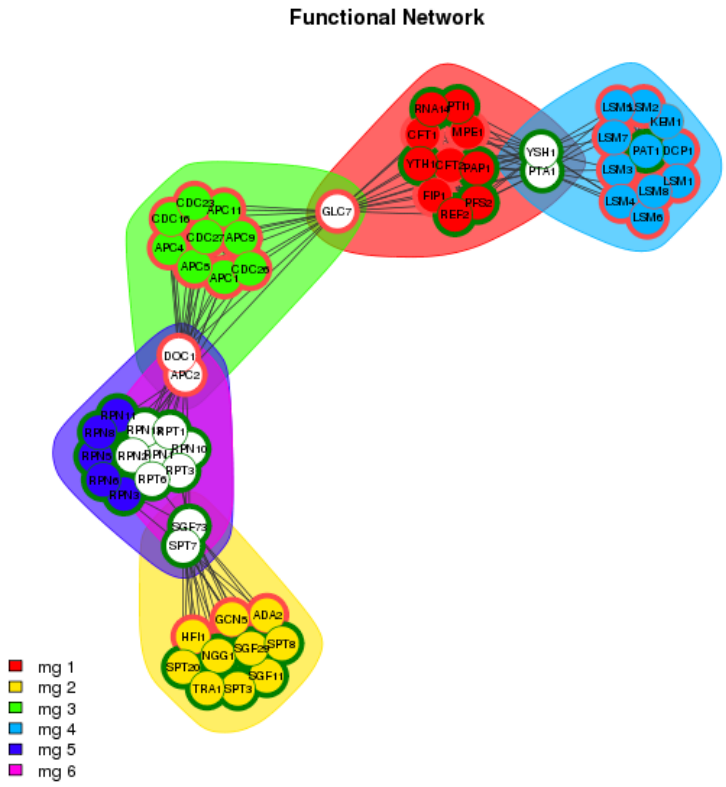
```
incidMat_terms <- fea2incidMat(feaResults, key="Terms")
incidMat_terms$metagroupsMatrix[5:10, 1:5]
```

```
##                                                 1 2 3 4 5
## Chromatin assembly (BP) (GO:0031497)            0 0 1 0 0
## Chromatin modification (BP) (GO:0016568)        0 1 0 0 0
## Cytoplasmic mRNA processing body (CC) (GO:0000932) 0 0 0 1 0
## Enzyme regulator activity (MF) (GO:0030234)     0 0 0 0 1
## Histone acetylation (BP) (GO:0016573)           0 1 0 0 0
## Histone acetyltransferase activity (MF) (GO:0004402) 0 1 0 0 0
```

These incidence matrices can be plotted and analyzed in different ways:

FGNet 11

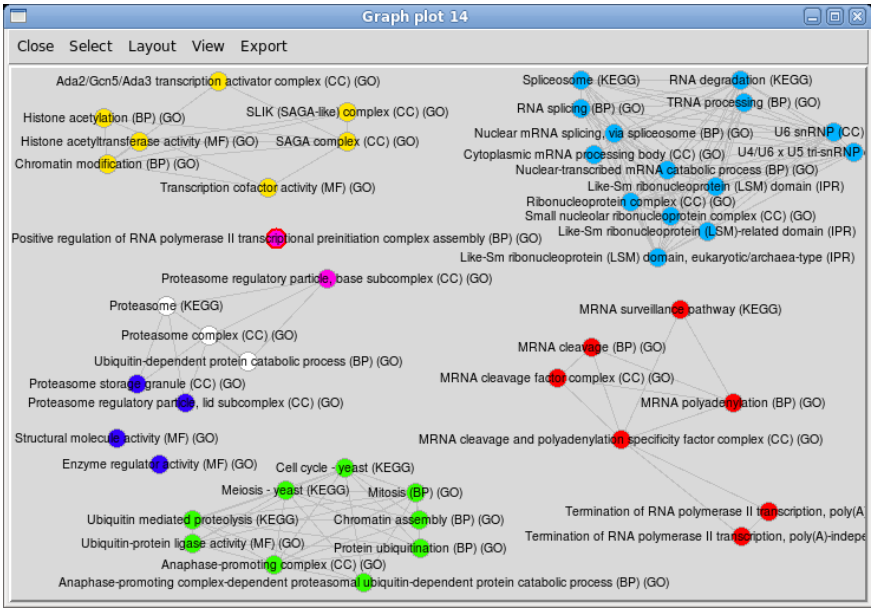
```
functionalNetwork(incidMat, geneExpr=genesYeastExpr,
    plotTitleSub="Default gene view")
```



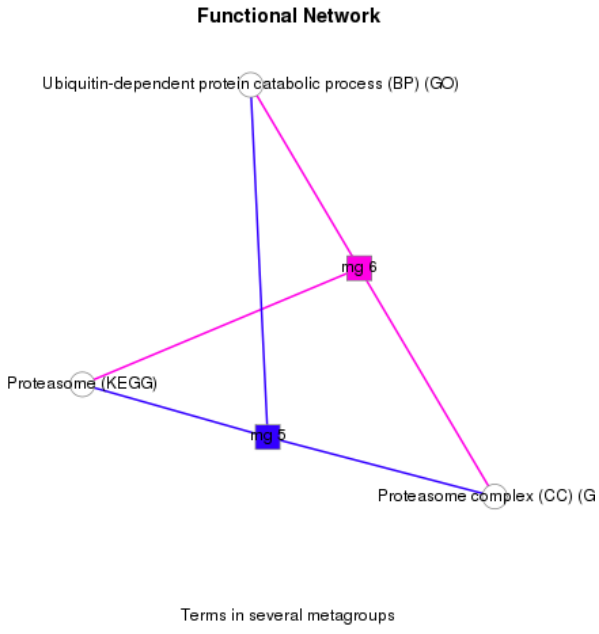

```
getTerms(feaResults)[1]
```

```
## $`Metagroup 1`
##      Terms
## [1,] "MRNA cleavage and polyadenylation specificity factor complex (CC)"
## [2,] "MRNA cleavage (BP)"
## [3,] "MRNA cleavage factor complex (CC)"
## [4,] "MRNA polyadenylation (BP)"
## [5,] "MRNA surveillance pathway"
## [6,] "Termination of RNA polymerase II transcription, poly(A)-coupled (BP)"
## [7,] "Termination of RNA polymerase II transcription, poly(A)-independent (BP)"
```

```
functionalNetwork(incidMat_terms, plotOutput="dynamic")
```




```
functionalNetwork(incidMat_terms, plotType="bipartite",
    plotTitleSub="Terms in several metagroups")
```



4 Editing and creating new networks

In this section we will use the functional analysis of an Alzheimer dataset (GSE4757):

```
jobID <- 1639610
feaAlzheimer <- fea_gtLinker_getResults(jobID=jobID, organism="Hs")
```

The variable `feaAlzheimer` contains the raw results from the functional analysis. The slot `metagroups` could also be `clusters` or missing depending on the FEA tool:

```
names(feaAlzheimer)
```

```
## [1] "queryArgs"    "metagroups"   "geneTermSets" "fileName"
```

```
head(feaAlzheimer$metagroups)
```

To see the terms in each cluster/metagroup use `getTerms()`:

```
getTerms(feaAlzheimer)[3:4]
```

```
## $`Metagroup 3`
##       Terms
##  [1,] "Alzheimer's disease"
##  [2,] "Calcium ion transport (BP)"
##  [3,] "Calcium signaling pathway"
##  [4,] "Calmodulin binding (MF)"
##  [5,] "GnRH signaling pathway"
##  [6,] "Induction of apoptosis by extracellular signals (BP)"
##  [7,] "Long-term potentiation"
##  [8,] "Melanogenesis"
##  [9,] "Neurotrophin signaling pathway"
## [10,] "Salivary secretion"
## [11,] "Tuberculosis"
## [12,] "Vascular smooth muscle contraction"
## [13,] "Wnt signaling pathway"
##
## $`Metagroup 4`
##      Terms
## [1,] "Glutamatergic synapse"
## [2,] "Postsynaptic density (CC)"
## [3,] "Postsynaptic membrane (CC)"
## [4,] "Synapse (CC)"
```

4.1 Incidence matrices

The FEA results should be transformed into incidence matrices to create the network. These matrices are the internal representation of the network: they contain which genes are in each metagroup or cluster and in each gene-term set. Therefore, it is in this step where the main shape of the network is determined.

The function to create the incidence matrices is `fea2incidMat()`. It allows to filter out clusters, decide whether the networks should be gene-based or term-based, establish the groups to link the genes/terms, etc...
We will start the example creating a simple gene-based network:

```
incidMat <- fea2incidMat(feaAlzheimer)
```

```
head(incidMat$metagroupsMatrix)
```

```
##        1 2 3 4 5 6 7 8 9
## ACTN1  1 0 0 0 1 0 0 0 0
## ADD3   1 0 1 0 0 0 0 0 0
## ANO3   1 0 0 0 0 0 0 0 0
## APOE   1 0 0 0 0 1 0 0 1
## ATP2B1 0 0 1 0 0 0 0 0 0
## C7     1 0 0 0 0 0 0 0 0
```

```
incidMat$gtSetsMatrix[1:5, 14:18]
```

```
##        1.14 1.15 1.16 1.17 1.18
## ACTN1     0    1    0    1    1
## ADD3      0    0    0    0    0
## ANO3      0    0    0    0    0
## APOE      0    0    1    0    0
## ATP2B1    0    0    0    0    0
```

To filter or select with metagroups to show, use the arguments `filterAttribute`, `filterOperator` and `filterThreshold`. `filterAttribute` should be a column from the `feaAlzheimer$clusters` or `feaAlzheimer$metagroups` data frames. The recommended filters for each tool can be seen in the object `FEA_tools`, which contains the default filters when generating the HTML report:

```
data(FEA_tools)
FEA_tools[,4:6]
```

```
incidMatFiltered <- fea2incidMat(feaAlzheimer,
    filterAttribute="Silhouette Width", filterOperator="<", filterThreshold=0.2)
```

To see which metagroups/clusters have been filtered out and will not be shown in the networks:

```
incidMatFiltered$filteredOut
```

For more on selecting and filtering groups see section 5. To build the networks based on terms, use the argument `key="Terms"`.

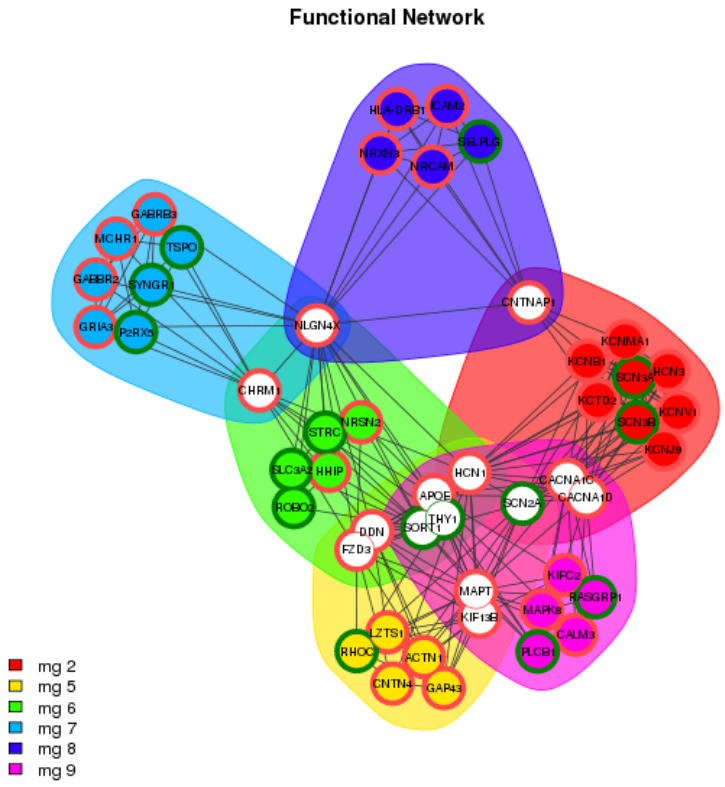
4.2 Functional network

The function `functionalNetwork()` generates and plots the networks. In case there is available expression data, it can be used for representation in this step:

```
# (Fake expression data)
genesAlz <- rownames(incidMat$metagroupsMatrix)
genesAlzExpr <- setNames(c(rep(1,50), rep(-1,27)), genesAlz)
```

The `default` plot will plot all the genes/terms in the network, and will return the networks as igraph objects and matrices in an invisible list. The argument `keepColors` determine whether the colors should be consistent, taking into account the filtered groups, or restarted:

```
fNw <- functionalNetwork(incidMatFiltered, geneExpr=genesAlzExpr, keepColors=FALSE)
```



By setting the parameter `plotOutput="dynamic"` instead of an static plot, it will create an interactive one. By setting `plotOutput="none"`, it is possible to produce only the network without plotting.

```
functionalNetwork(incidMatFiltered, geneExpr=genesAlzExpr, plotOutput="dynamic")
fNw <- functionalNetwork(incidMatFiltered, plotOutput="none")
```

Since the returned networks are `iGraph` objects, they can be used or analyzed as such:

```
names(fNw)
```

```
## [1] "iGraph" "adjMat"
```

```
names(fNw$iGraph)
```

```
## [1] "commonClusters" "commonGtSets"
```

```
library(igraph)
clNw <- fNw$iGraph$commonClusters
clNw
```

```
## IGRAPH UN-- 49 334 --
## + attr: name (v/c)
```

```
vcount(clNw)
ecount(clNw)
sort(betweenness(clNw), decreasing=TRUE)[1:10]
igraph.to.graphNEL(clNw)
```

In dynamic plots (`tkplot`) it is not possible to draw the metagroup background. However, you can save the layout of a dynamic network, and plot it as static using the argument `vLayout`:

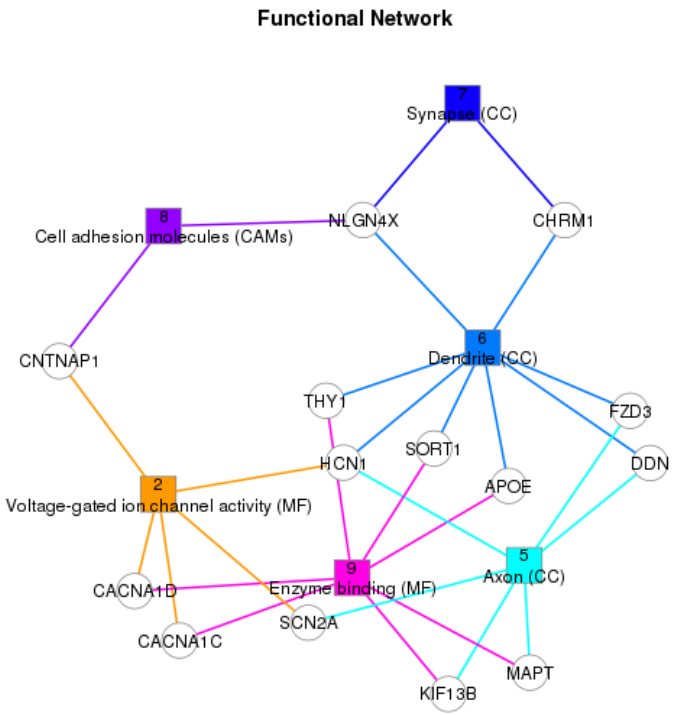
```
functionalNetwork(incidMatFiltered, plotOutput="dynamic")
# Modify the layout...
saveLayout <- tkplot.getcoords(1)    # tkp.id (ID of the tkplot window)
functionalNetwork(incidMatFiltered, vLayout=saveLayout)
```

FGNet 17

4.3 Bipartite and intersection network

The default `bipartite` version of the functional network plots the *intersection network*: a simplified functional network, containing only the nodes in several metagroups and the metagroups they belong to. In this network, metagroup nodes (the coloured nodes) can be seen as a cluster of all the genes/proteins that belong only to that metagroup:

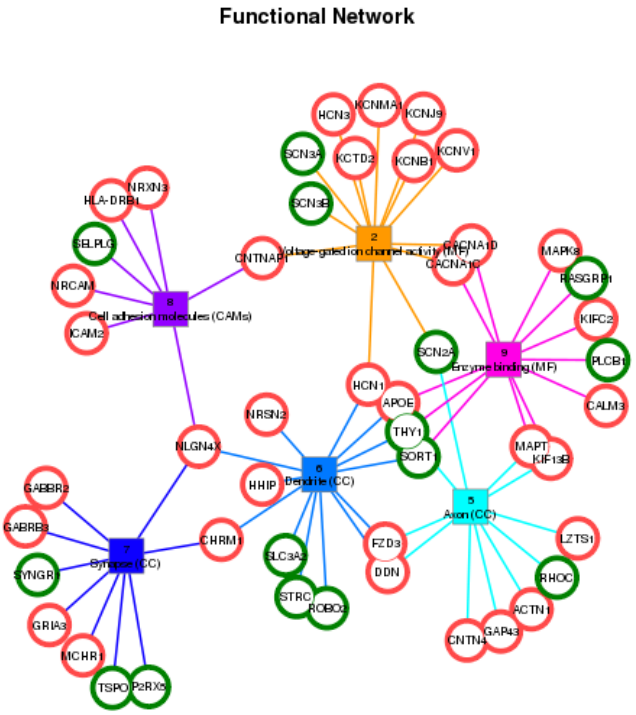
```
mgKeyTerm <- keywordsTerm(getTerms(feaAlzheimer),
    nChar=100)[-c(as.numeric(incidMatFiltered$filteredOut))]
functionalNetwork(incidMatFiltered, plotType="bipartite", legendText=mgKeyTerm)
```



FGNet 18

To plot a full bipartite network including all the nodes, just set `keepAllNodes=TRUE`:

```
functionalNetwork(incidMatFiltered, geneExpr=genesAlzExpr, plotType="bipartite",
    keepAllNodes=TRUE, plotTitleSub="Bipartite network will all nodes",
    legendText=mgKeyTerm)
```



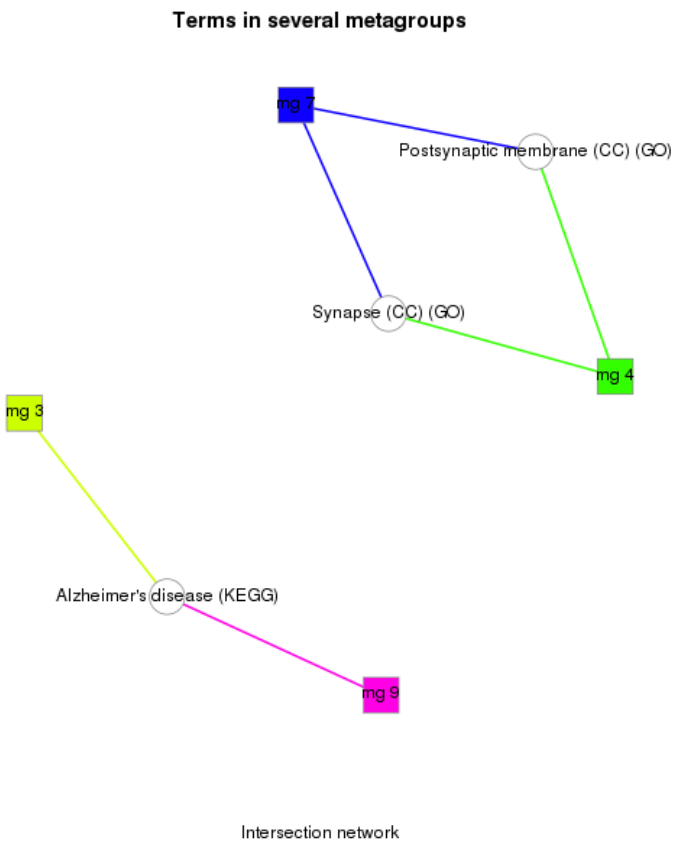
Bipartite network will all nodes

4.4 Terms networks

In the same way we have built networks to explore the relationship between genes, the same approach can be used to explore the relationship between the biological terms in the enrichment analysis. i.e. to see which biological terms are usually associated, or locate which terms are in several groups. To do so, build the incidence matrices based on terms instead of genes using the argument `key="Terms"`.

```
incidMatTerms <- fea2incidMat(feaAlzheimer, key="Terms")
```

```
functionalNetwork(incidMatTerms, plotType="bipartite",
    plotTitle="Terms in several metagroups")
```



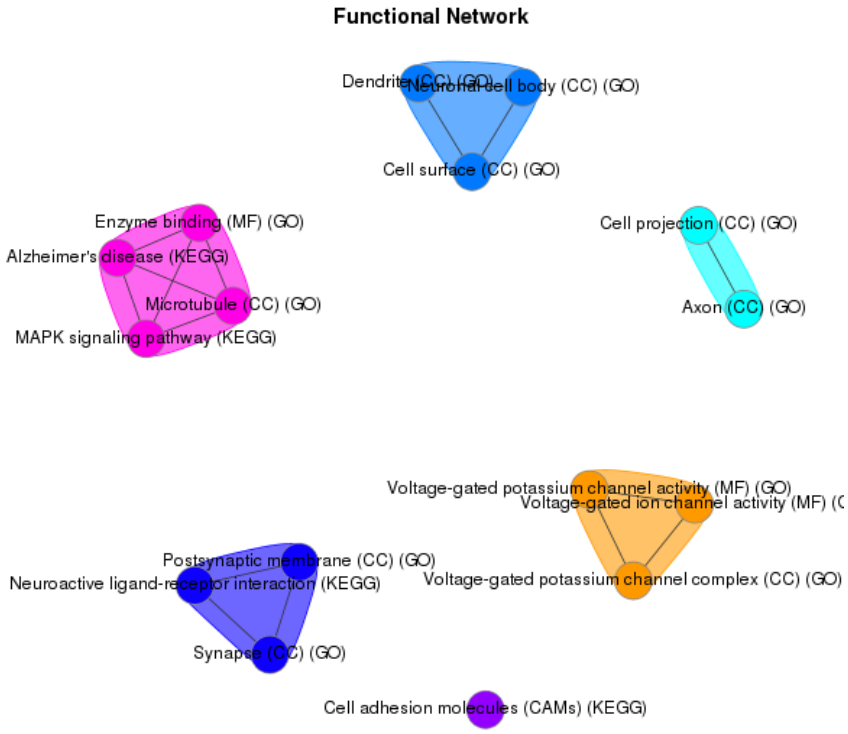
By default, the functional network is built establishing links between nodes (genes or terms) in the same gene-term sets. Depending on the tool, this network might have few or no edges:

```
functionalNetwork(incidMatTerms,  weighted=TRUE, plotOutput="dynamic")
```

To plot a network with links between all the terms in the same cluster or metagroups, use `fea2incidMat()` with the `$cluster` or `$metagroup` slots from the FEA, in order to consider the whole cluster/metagroup as a gene-term set:

```
incidMatTerms <- fea2incidMat(feaAlzheimer$metagroups, clusterColumn="Metagroup",
    key="Terms",
    filterAttribute="Silhouette.Width", filterThreshold=0.2)
functionalNetwork(incidMatTerms, legendText=FALSE, plotOutput="dynamic")
```

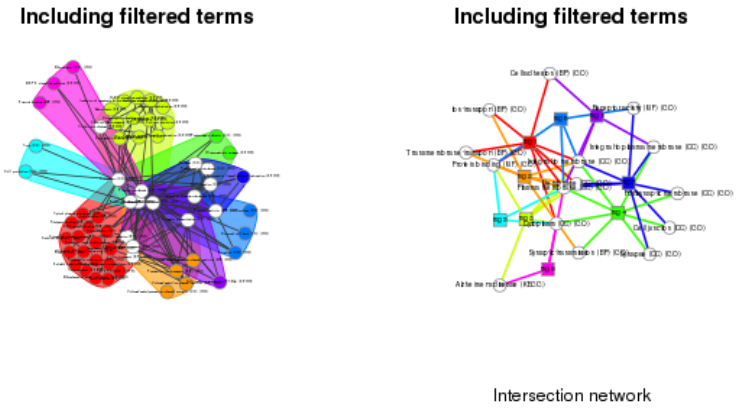
```
functionalNetwork(incidMatTerms, legendText=FALSE)
```



Since GeneTerm Linker filters out generic and redundant terms from the final metagroups, by default these terms are not plotted. To include them in the graph, set the argument `removeFiltered=FALSE` (only available for GeneTerm Linker).

```
incidMatTerms <- fea2incidMat(feaAlzheimer, key="Terms", removeFilteredGtl=FALSE)
par(mfrow=c(1,2))
functionalNetwork(incidMatTerms, vLabelCex=0.2,
    plotTitle="Including filtered terms", legendText=FALSE)
functionalNetwork(incidMatTerms, plotType="bipartite", vLabelCex=0.4,
    plotTitle="Including filtered terms")
```

For more information on the filtered terms see (Fontanillo et al) or `http://gtlinker.cnb.csic.es/gtset/help` .



4.5 Genes - Terms networks

To build a genes-terms network, we can use the bipartite plot with the appropriate formatting of the input matrices.
For many FEA tools it will be enough with applying the `fea2incidMat()` directly to the `$geneTermSets` matrix selecting the gene-term sets we want to plot. i.e. gene-term sets in a specific cluster, filter generic terms (terms annotated to more than X genes), etc...
Note that this approach might not be appropriate for GeneTerm Linker, since it groups several terms into each gene-term set.

```
txtFile <- paste(file.path(system.file('examples', package='FGNet')),
    "DAVID_Yeast_raw.txt", sep=.Platform$file.sep)
feaResults_David <- format_david(txtFile, jobName="David_example",
    geneLabels=genesYeast)
```

```
feaResults_David <- fea_david(names(genesYeast), email="...",
    geneLabels=genesYeast)
```

```
gtSets <- feaResults_David$geneTermSets
gtSets <- gtSets[gtSets$Cluster %in% c(9),]
gtSets <- gtSets[gtSets$Pop.Hits<500,]
```

Then, create a terms-genes incidence matrix with `fea2incidMat()`, and plot the network...

```
termsGenes <- t(fea2incidMat(gtSets, clusterColumn="Terms")$clustersMatrix)
library(R.utils)
rownames(termsGenes) <- sapply(strsplit(rownames(termsGenes), ":"),
    function(x) capitalize(x[length(x)]))
termsGenes[1:5,1:5]
```

```
##                                                CDC16 DOC1 GLC7
## Anatomical structure morphogenesis                 1    1    1
## Cell differentiation                               1    1    1
```

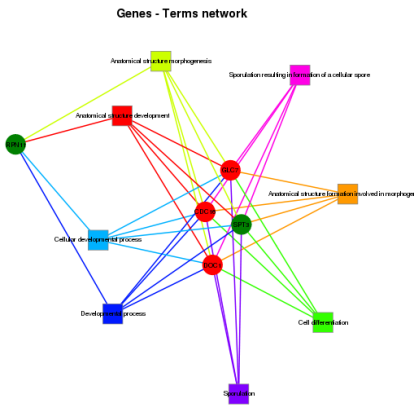
```
## Sporulation resulting in formation of a cellular spore     1    1    1
## Developmental process                                      1    1    1
## Sporulation                                                1    1    1
##                                                         RPN11 SPT3
## Anatomical structure morphogenesis                          1    1
## Cell differentiation                                        0    1
## Sporulation resulting in formation of a cellular spore      0    1
## Developmental process                                       1    1
## Sporulation                                                 0    1
```

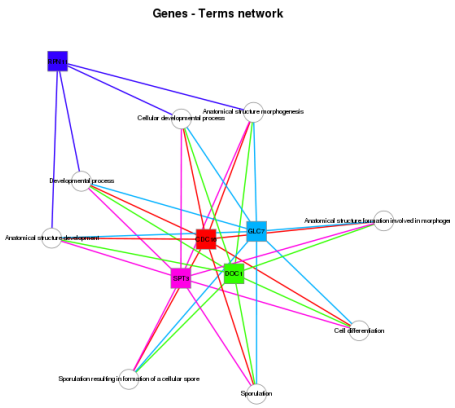
Network with genes colored based on their expression and terms on alphabetical order:

```
functionalNetwork(t(termsGenes), plotType="bipartite", keepAllNodes=TRUE,
    legendPrefix="", plotTitle="Genes - Terms network", plotTitleSub="",
    geneExpr=genesYeastExpr, plotExpression="Fill")
```

Network with genes colored by alphabetical order (from red to pink), terms white:

```
functionalNetwork(termsGenes, plotType="bipartite", keepAllNodes=TRUE,
    legendPrefix="", plotTitle="Genes - Terms network", plotTitleSub="")
```



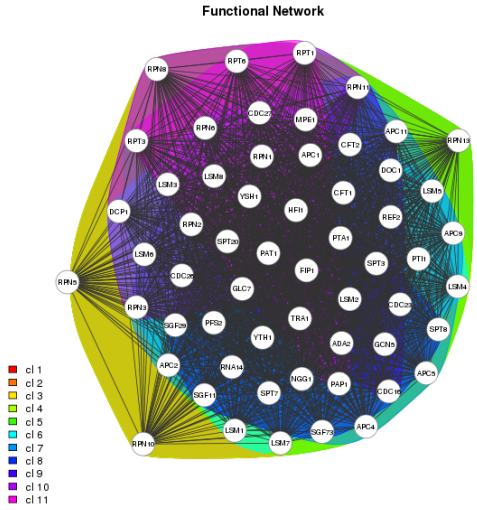


5 Filtering and selecting clusters

As an example of analysis of a network with very overlapping clusters, we will use the yeast gene list analyzed with DAVID:

```
incidMat <- fea2incidMat(feaResults_David)
functionalNetwork(incidMat)
```

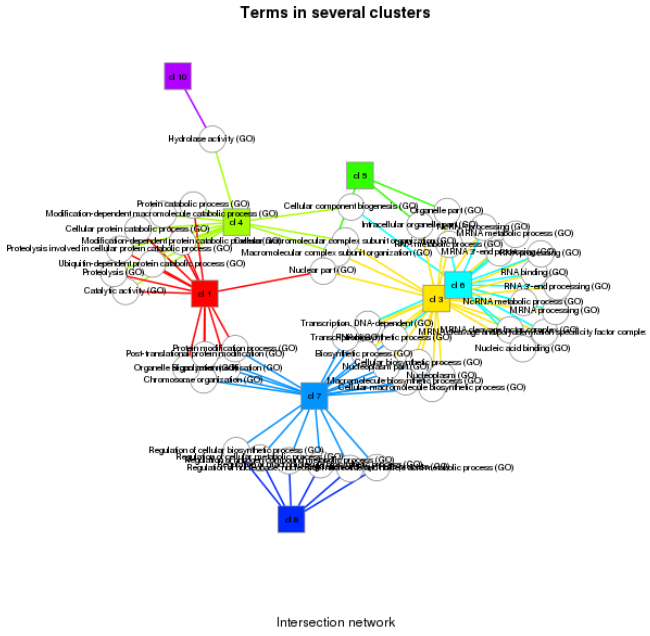
FGNet 23




```
incidMatTerms <- fea2incidMat(feaResults_David, key="Terms")
```

```
functionalNetwork(incidMatTerms$clustersMatrix, plotOutput="dynamic",
  weighted=TRUE, eColor="grey")
```

```
functionalNetwork(incidMatTerms$clustersMatrix, plotType="bipartite",
 plotTitle="Terms in several clusters")
```



5.1 Filtering based on a *cluster* property

The clusters to plot can be selected/filtered based on any property that is available in the clusters matrix:

```
colnames(feaResults_David$clusters)
```

```
## [1] "Cluster"                "nGenes"
## [3] "ClusterEnrichmentScore" "Genes"
## [5] "Terms"                  "keyWordsTerm"
```

i.e. Selecting the clusters with highest Enrichment Score or least genes (setting `eColor=NA`, plots the networks without edges):

```
par(mfrow=c(1,2))

# Highest enrichment score
filterProp <- as.numeric(as.character(feaResults_David$
    clusters$ClusterEnrichmentScore))
quantile(filterProp, c(0.10, 0.25, 0.5, 0.75, 0.9))
```

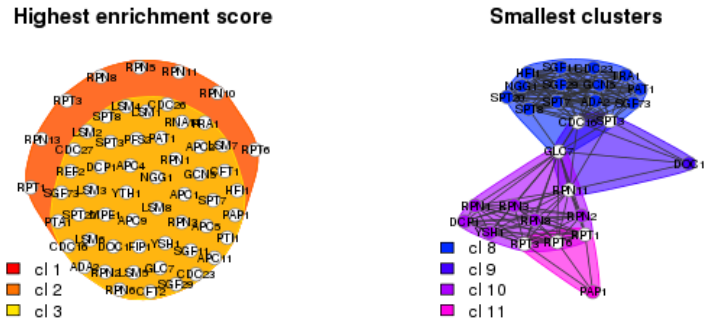
```
##        10%        25%        50%        75%        90%
## 0.08585003 0.33812100 5.90148600 7.65222050 7.85874500
```

```
incidMatFiltered <- fea2incidMat(feaResults_David,
    filterAttribute="ClusterEnrichmentScore",
    filterOperator="<", filterThreshold=7.65)
functionalNetwork(incidMatFiltered, eColor=NA,
    plotTitle="Highest enrichment score")

# Lowest genes
quantile(as.numeric(as.character(feaResults_David$clusters$nGenes)),
    c(0.10, 0.25, 0.5, 0.75, 0.9))
```

```
##  10%  25%  50%  75%  90%
##  5.0 13.5 44.0 55.0 58.0
```

```
incidMatFiltered <- fea2incidMat(feaResults_David,
 filterAttribute="nGenes", filterOperator=">", filterThreshold=20)
functionalNetwork(incidMatFiltered, plotTitle="Smallest clusters")
```

To use any property that is not available in the `$clusters` data frame, just add it as column to the dataframe.



5.2 Selecting clusters with specific keywords

```
keywordsTerm(getTerms(feaResults_David), nChar=100)
```

```
##                              Cluster 1                              Cluster 2
## "Cellular protein catabolic process"                  "Metabolic process"
##                              Cluster 3                              Cluster 4
##                         "Transcription" "Cellular protein catabolic process"
##                              Cluster 5                              Cluster 6
##                            "Organelle"                    "MRNA processing"
##                              Cluster 7                              Cluster 8
##                         "Transcription" "Regulation of biosynthetic process"
##                              Cluster 9                             Cluster 10
##   "Anatomical structure development"                 "Hydrolase activity"
##                             Cluster 11
##                           "ATP binding"
```

```
keywords <- c("hydrolase")
selectedClusters <- sapply(getTerms(feaResults_David),
    function(x)
    any(grep(paste("(", paste(keywords, collapse="|") ,")",sep=""), tolower(x))))
```

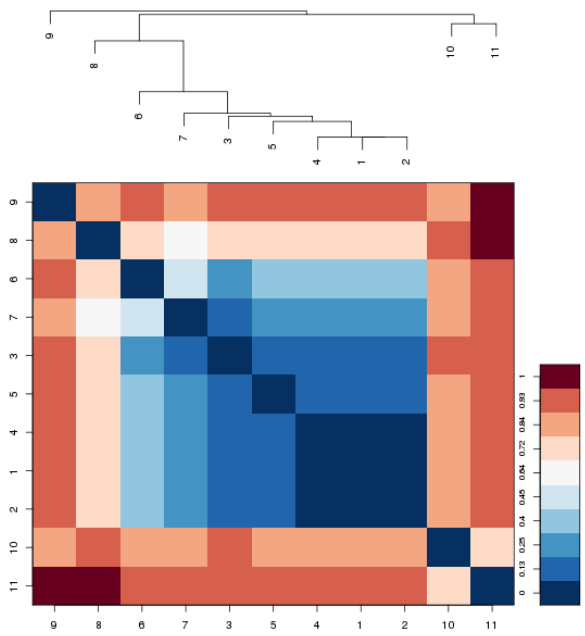
```
getTerms(feaResults_David)[selectedClusters]
```

```
tmpFea <- feaResults_David
tmpFea$clusters <- cbind(tmpFea$clusters, keywords=selectedClusters)
incidMatSelection <- fea2incidMat(tmpFea,
 filterAttribute="keywords", filterOperator="!=",filterThreshold="TRUE")
functionalNetwork(incidMatSelection, plotType="bipartite")
```

5.3 Selecting specific clusters

`clustersDistance()` allows to explore the overlap between clusters:

```
distMat <- clustersDistance(incidMat)
```



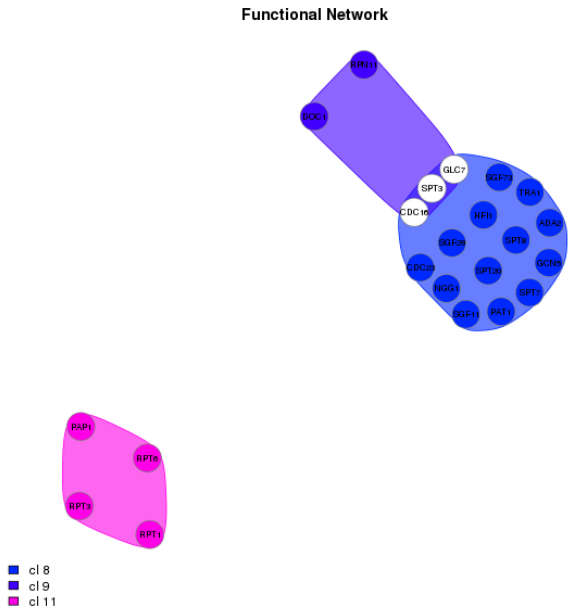
Clusters 4, 1 and 2 seem to be completely overlapping (distance 0). While cluster 11 does not have any intersection with clusters 8 and 9. Let's see:

```
selectedClusters <- rep(FALSE, nrow(feaResults_David$clusters))
selectedClusters[c(8,9,11)] <- TRUE

tmpFea <- feaResults_David
tmpFea$clusters <- cbind(tmpFea$clusters, select=selectedClusters)
incidMatSelection <- fea2incidMat(tmpFea,
  filterAttribute="select", filterOperator="!=",filterThreshold="TRUE")
functionalNetwork(incidMatSelection, eColor=NA)
```

5.4 Filtering based on a *gene-term set* property

In some ocasions, it might also be useful to filter out gene-term sets within a cluster. i.e. The terms in the top of the GO ontologies are annotated to many genes and make most clusters overalp.
To filter out terms, (1) filter or select the terms in the the `feaReults$geneTermSets` data frame, (2) save it as text file, and (3) import it with `readGeneTermSets()`
In this case, we will use DAVID's example, and keep the terms thar are annotated to less than 100 genes in yeast:

FGNet 27




```
# Same analysis, setting overlap to 6:
feaResults_David_ov6 <- fea_david(names(genesYeast), geneLabels=genesYeast,
    email="example@email.com",
    argsWS=c(overlap=6, initialSeed=3, finalSeed=3, linkage=0.5, kappa=50))
```

```
# Filter/select
sum(feaResults_David_ov6$geneTermSets$Pop.Hits < 100)
```

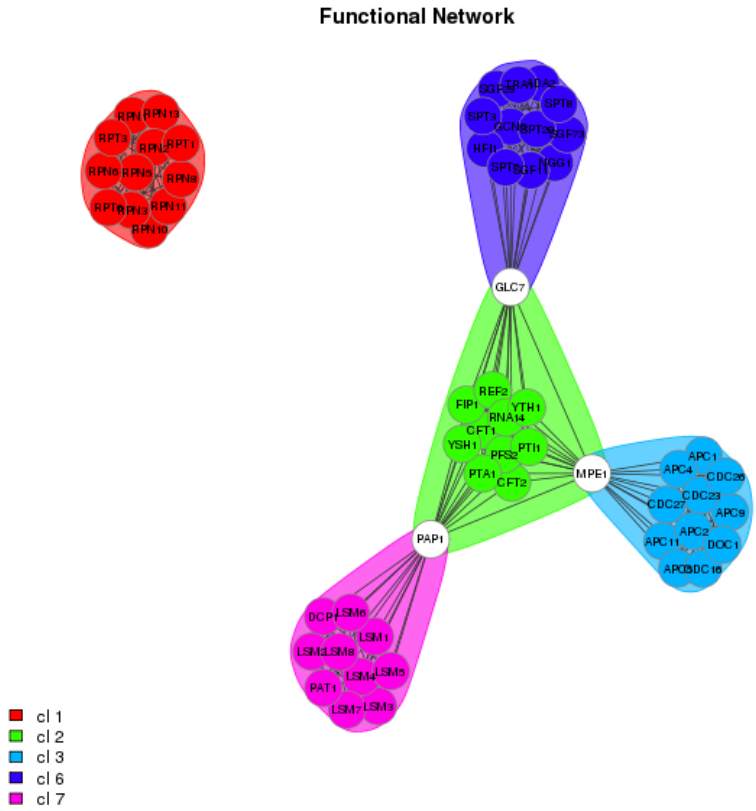
```
## [1] 64
```

```
gtSets <- feaResults_David_ov6$geneTermSets[
    feaResults_David_ov6$geneTermSets$Pop.Hits < 100,]
# Save
write.table(gtSets, file="david_filteredGtsets.txt", sep="\t",
    col.names = TRUE, quote=FALSE)
# Load with "readGeneTermSets"
feaResults_filteredGtsets <- readGeneTermSets("david_filteredGtsets.txt",
    tool="DAVID")
# ...
functionalNetwork(fea2incidMat(feaResults_filteredGtsets))
```

To explore the distribution of genes-terms in a specific organism:

```
# Yeast
library(org.Sc.sgd.db)
goGenesCountSc <- table(sapply(as.list(org.Sc.sgdGO2ORF), length))
barplot(goGenesCountSc, main="Number of genes annotated to GO term (Sc) ",
```

FGNet 28



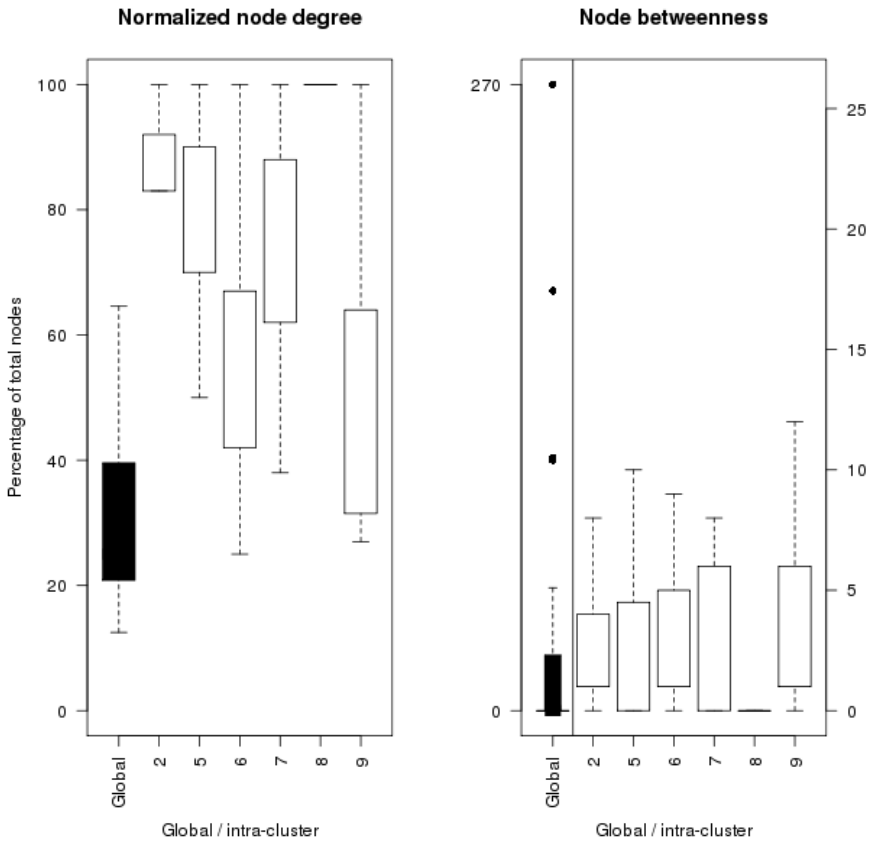

```
    xlab="Number of genes", ylab="Number of GO terms")

# Human
library(org.Hs.eg.db)
goGenesCountHs <- table(sapply(as.list(org.Hs.egGO2EG), length))
barplot(goGenesCountHs, main="Number of genes annotated to GO term (Human)",
    xlab="Number of genes", ylab="Number of GO terms")
```

6 Other auxiliary functions

6.1 analyzeNetwork()

`analyzeNetwork()` can be used to explore the structure of the network. It also returns statistics about the nodes betweenness within each cluster, etc...
The example with GeneTerm Linker (Alzheimer):

```
incidMatFiltered <- fea2incidMat(feaAlzheimer,
    filterAttribute="Silhouette Width", filterOperator="<", filterThreshold=0.2)
stats <- analyzeNetwork(incidMatFiltered)
```



```
names(stats)
```

```
## [1] "degree"           "betweeness"       "transitivity"
## [4] "betweenessMatrix" "hubsList"         "intraHubsCount"
```

```
stats$transitivity
```

```
## commonClustersNw   commonGtSetsNw
##        0.6947699        0.5943638
```

`$degree` and `$betweenness` are the values used for the plots. They contain the values for each of the nodes in the global network (commonClusters) and within each cluster/metagroup (subsets of commonGtSets network). The degree is given as percentage, normalized based on the total number of nodes of the network. i.e. a value of 90 in a network of 10 nodes, would mean the actual degree of the node is 9: it is connected to 9 nodes (90% of 10)).
The betweenness of each node in each cluster as matrix:

```
head(stats$betweennessMatrix)
```

Inter-modular hubs: Nodes with betweenness within the top 75% in the global network

```
stats$hubsList$Global
```

```
## [1] "NLGN4X"  "HCN1"    "CHRM1"   "CNTNAP1" "SCN2A"
```

Intra-modular hubs: Nodes with betweenness within the top 75% in each cluster sub-network

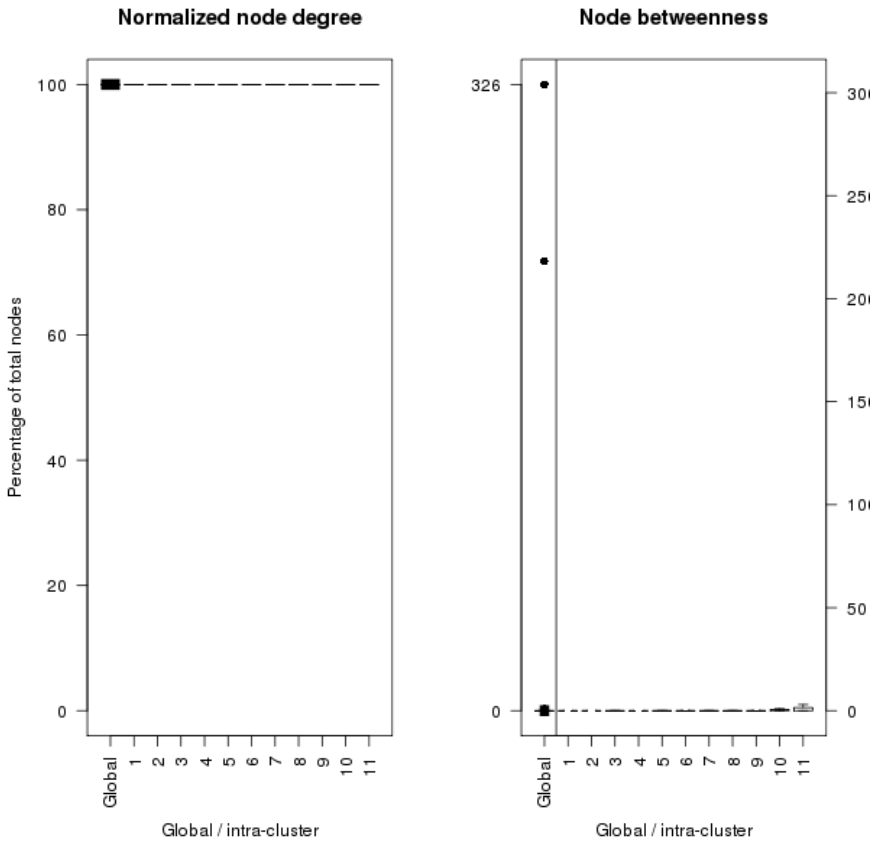
```
stats$hubsList$"9"
```

```
## [1] "MAPT"  "CALM3" "APOE"
```

FGNet 31

DAVID's example:

```
incidMat_metab <- fea2incidMat(feaResults_David)
analyzeNetwork(incidMat_metab)
```

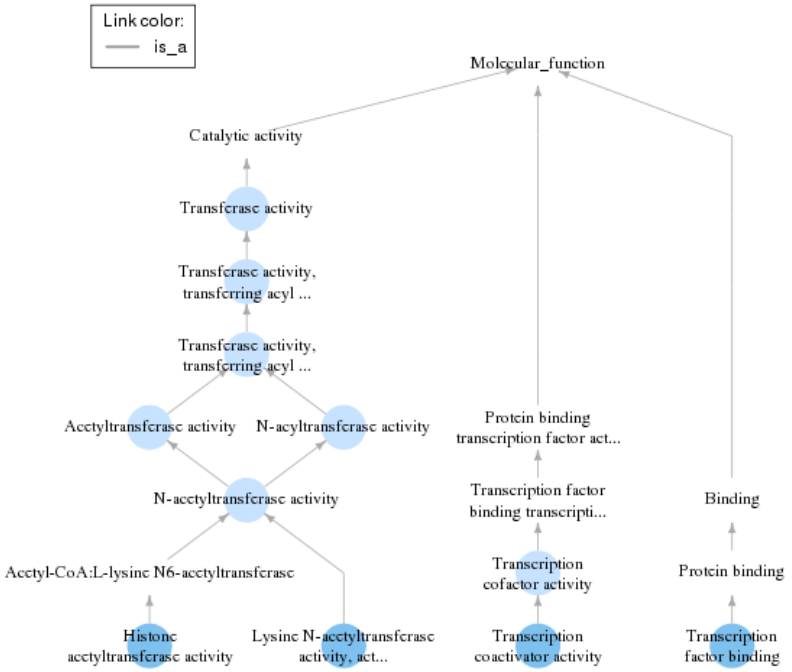


Note the structure of the network varies not only depending on the dataset, but also on the tool. Since tools like DAVID link all the nodes/terms within each cluster, their internal normalized degree is always 100%.

6.2 plotGoAncestors()

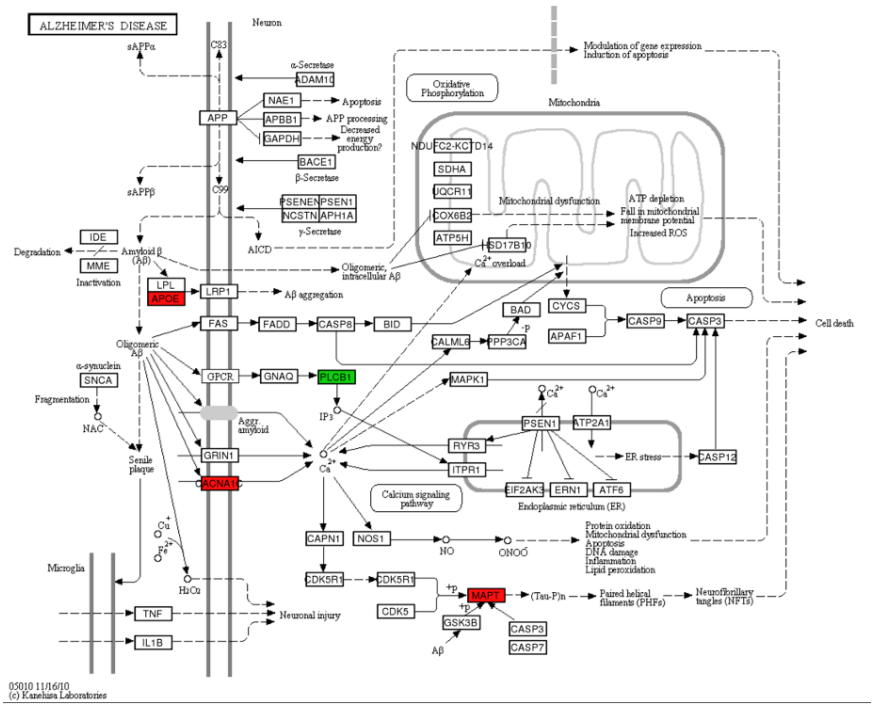
`plotGoAncestors()` and `plotKegg()` also allow to explore the significant gene term sets:

```
goIds <- getTerms(feaResults_David, returnValue="GO")[[7]]
plotGoAncestors(goIds, ontology="MF", nCharTerm=40)
```



6.3 plotKegg()

```
keggIds <- getTerms(feaAlzheimer, returnValue="KEGG")[[3]]
plotKegg("hsa05010", geneExpr=genesAlzExpr, geneIDtype="GENENAME")
# Saved as .png in current directory
```



Acknowledgements

This work was supported by Instituto de Salud Carlos III [Research Projects PS09/00843 and PI12/00624] and by a grant from the Junta de Castilla y Leon and the European Social Fund to S.A and C.D.

References

[1] Huang DW, Sherman BT, Lempicki RA. *Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources.* Nature Protoc. 2009;4(1):44-57.

[2] Huang DW, Sherman BT, Lempicki RA. *Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists.* Nucleic Acids Res. 2009;37(1):1-13.

[3] Fontanillo C, Nogales-Cadenas R, Pascual-Montano A, De Las Rivas J (2011) *Functional Analysis beyond Enrichment: Non-Redundant Reciprocal Linkage of Genes and Biological Terms.* PLoS ONE 6(9): e24289. doi:10.1371/journal.pone.0024289

[4] Alexa A, and Rahnenfuhrer J (2010) topGO: Enrichment analysis for Gene Ontology. R package version 2.16.0. URL: `http://www.bioconductor.org/packages/release/bioc/html/topGO.html`

[5] Luo W, Friedman MS, Shedden K, Hankenson KD, Woolf PJ (2009) GAGE: generally applicable gene set enrichment for pathway analysis. BMC Bioinformatics. 10:161. URL: `http://www.bioconductor.org/packages/release/bioc/html/gage.html`

Chapter 3

Study 1: Combined analysis of genome-wide DNA methylation and expression profiles from low-risk myelodysplastic syndromes

Chapter index

1. Introduction

The methylation profile of the DNA has long been known to be altered in multiple hematological malignancies (Galm et al., 2006), and it has even been identified as a potential marker in the early stages of several diseases (*Laird, 2003*). In myelodysplastic syndromes (MDS) it was observed that patients responded specially well to inhibitors of DNA methyltransferase (e.g. azacitidine and decitabine). This led to many studies on the role of methylation in MDS: from studies on the global state of the DNA methylation in MDS (*Jiang et al., 2009*), to more specific studies, such as the effects of the demethylating agents on the genome-wide methylation patterns (*Figueroa et al., 2009*) or the potential of DNA methylation to predict the progression of the disease (*Shen et al., 2010*). However, at the time this study started, there were still many questions to be answered. It was known that MDS patients had methylation patterns altered, but the specific profile was not well defined. In addition, it was not known to what extent these alterations in DNA methylation were the cause of the observed changes in expression patterns or which biological processes it affected (*Raza and Galili, 2012*).

In order to solve some of these questions, a collaborative study on the alteration of expression and DNA methylation patterns of low-risk myelodysplastic syndromes was started. The aim of this project was to study the underlaying mechanisms in the deregulation of the expression patterns that had been identified in previous studies. This would be carried on through an integrative analysis of the methylation and gene expression profiles obtained for the same cohort of low-risk MDS patients. These included patients with refractory anemia (RA), patients with refractory anemia with ringed sideroblasts (RARS) and other donors having non-MDS/non-leukemia disease that would be used as controls (i.e. non-leukemia cases: NoL).

The results and major findings of this study were published in the following article:

> Del Rey M, O'Hagan K, Dellett M, Aibar S, Colyer HA, Alonso ME, Díez-Campelo M, Armstrong RN, Sharpe DJ, Gutiérrez NC, García JL, De Las Rivas J, Mills KI, Hernández-Rivas JM (2013) **Genome-wide profiling of methylation identifies novel targets with aberrant hypermethylation and reduced expression in low-risk myelodysplastic syndromes.** *Leukemia* 27, 610-618. doi: 10.1038/leu.2012.253 PMID: 22936014
>
> The raw data are available at the Gene Expression Omnibus (GEO) database, under the series accesion number GSE41216, which contains both, the expression dataset (GSE41130) and the methylation dataset (GSE41215).

This chapter presents the bioinformatic work done to analyze and integrate the genomic data in this study. The objective of these analyses was to combine the expression and methylation data to identify the set of genes which might be silenced by hyper-methylation. These genes would allow identifying the biological processes potentially altered by the aberrant DNA methylation in low-risk myelodysplastic syndromes. A brief explanation of other key steps of the study is also provided in order to understand the data and give cohesion to the work.

2. Methods

2.1. Patient samples information

The low-risk MDS samples used in this study were obtained from 6 patients with refractory anemia (RA), and 12 patients with refractory anemia with ringed sideroblasts (RARS) according to the 2008 World Health Organization criteria (*Vardiman et al., 2009*). In addition, there are 7 control samples, labeled as *no leukemia* (NoL), from patients with other conditions not related to MDS:

1. Bicytopenia
2. Idiopathic thrombocytopenic purpura
3. Neutropenia
4. Neutropenia and anemia
5. Non-hematologic disease
6. Pancytopenia
7. Trombocitopenia

From each of the patients, samples of RNA and DNA extracted from bone marrow mononuclear cells and isolated by density gradient (Ficoll) were hybridized on high-density genome-wide expression and methylation microarrays.

The mean age of the patients was 77 years old (ranging from 29 to 95), and the proportion of male-female was 42.1% to 57.9%. Most of the samples had normal karyotype (71.9%), although there were also a 21.1% that had some karyotypic alteration and 7% 'no mitosis'.

2.2. Methylation analysis

The DNA methylation profiles were measured using *methylated CpG island amplification microarray* (MCAM, *Estécio et al., 2007*. This is a genomic platform which consists on a human CpG island microarray that is hybridized with the amplicons obtained by *methylated CpG island amplification* (MCA, *Toyota et al., 1999*) (*Figure 1*).

Methylated CpG island amplification (MCA) is a molecular biology method based on using restriction enzymes to identify and cut methylated CpG islands from a given query genome (DNA), and amplifying these sequences by PCR. This process consists on three main steps:

1. Digesting the DNA with *Sma*I, a restriction enzyme with recognition sequence: 'CCCGGG'. Due to the fact that this enzyme is methylation sensitive, it eliminates un-methylated sites by cutting between the C and G.
2. Second digestion of the resulting DNA fragments with *Xma*I[2], another restriction enzyme whose recognition site is also 'CCCGGG', but this enzyme cuts methylated sites leaving a 'CCGG' overhang.
3. The 'CCGG' overhang is used to ligate adaptors to perform PCR and amplify the methylated sequences producing the amplicons. These amplified sequences can be used to hybridize the CpG microarray.

The specific microarray platform used in our study is the human 12K CpG microarray from the University Health Network (UHN, Toronto, Canada). This platform is a two-color array including 12,192 CpG Island clones (obtained from the Sanger Centre, UK), which quantifies methylation by co-hybridizing *case* DNA and *control* DNA to the array. An important restriction of this platform is the limited coverage of genes. According to the annotation files

[2] *Xma*I is a *Sma*I isoschizomer/neoschizomer. *Isoschizomers* are restriction enzymes that recognize the same sequence. *Neoschizomer*s are a subtype of *isoschizomers:* they recognize the same sequence, but cut the DNA in a different way.

provided by the manufacturer[3], the 12,192 probesets are located nearby to only 4,382 Entrez gene IDs or 5,375 unigene gene IDs. Moreover, 2,428 probesets are not mapped to any genomic coordinates. Another limitation of the MCA method is that it only detects and amplifies CpG islands containing at least two 'CCCGGG' sites within less than 1kb. These are estimated to be about 70-80% of the total CpG islands.

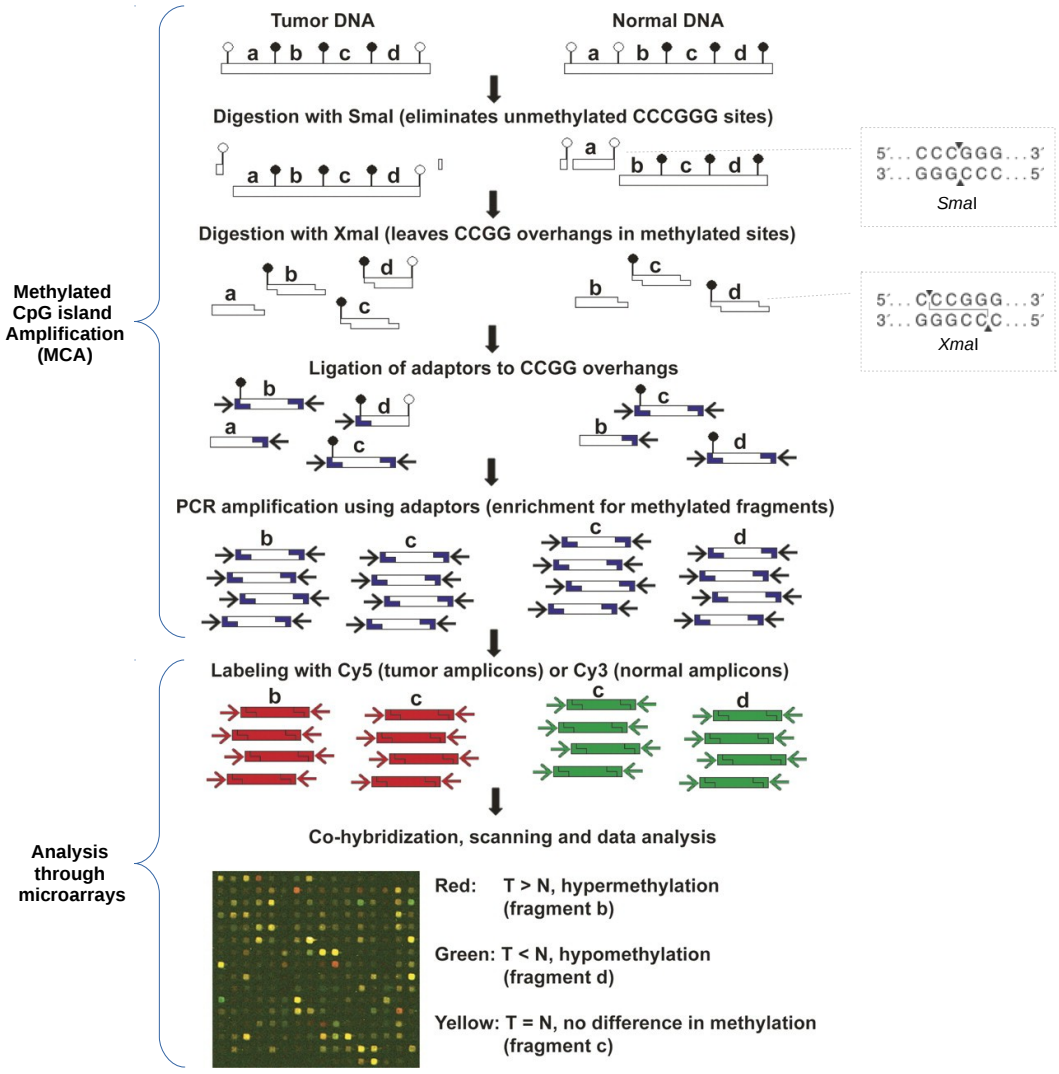


Figure 1. MCAM protocol overview
(Source: Based on the figure by Estecio et al.)

In our study, the whole MCAM procedure was done mostly as described by *Estécio et al. (2007)*, only with some minor variations (for details see *del Rey et al., 2013*). All MDS samples were labeled with Cy5 (red) and hybridized into the UHN 12k CpG microarrays using Human Cot-1 DNA –a commercial DNA– labeled with Cy3 (green) as control. The same process was also followed for the no-leukemia control samples, but instead of hybridizing an array for each sample, the available samples were merged into two pools which were used to hybridize two microarrays: *Pool 1* containing the *Pancytopenia*, *Bicytopenia*, and *Non-hematologic disease* samples; and *Pool 2* containing the *Neutropenia*, *Neutropenia and anemia*, *Thrombocytopenia* and *Idiopathic thrombocytopenic purpura* samples.

[3] Annotation file for 12.1k array downloaded from http://data.microarrays.ca/cpg/download.htm (version 2010-05-05)

After scanning and performing the quality control on the microarrays, they were preprocessed to get the signal intensities. Out of the 12,193 probesets, 10,426 had missing data in one or more samples (i.e. correspond to spots in the arrays flagged as 'no-good-signal' by the scanner). If all these probesets had been removed, only 1,767 would have been left and this was very restrictive. Therefore, only the 6,019 probesets with missing values in more than three samples were removed, and the missing values in the remaining probesets were calculated using K-nearest neighbors. After this filter, the probesets were mapped to a chromosome location finding 4,900 which were kept and annotated to the genes located nearby. These genes were labeled either 'up-stream' or 'down-stream' depending on their position from the CpG island location, or 'within' if the probeset/CpG island was within the gene locus. Then, *Limma* package (*Ritchie et al., 2015*) was used to perform three contrasts: (1) NOL *versus* AR, (2) NOL *versus* ARS, and (3) NOL *versus* AR+ARS.

The output results of these differential methylation analyses –done by the collaborators in Belfast (UK)– were integrated with the differential gene expression results as indicated in the upcoming steps.

2.3. Expression analysis

The gene expression profiling was done with the high-density oligonucleotide platform *GeneChip® Human Genome U133 Plus 2.0 Arrays* following the protocols from *Affymetrix*. The bioinformatic analysis started by normalization and calculation of the expression signal for each gene in each sample using the algorithm *Robust Microarray Analysis* (*RMA*, *Irizarry et al., 2003a*) with a mapping protocol (CDF) of the oligonucleotide probesets to 17,582 genes (*Risueño et al., 2010*). During quality control of the arrays, one of the RARS samples presenting biased signal was excluded from the analysis.

Since for the methylation analysis the control samples had been grouped into pools, at first it was considered to simulate an artificial 'pool of control NoL samples' (calculating average expression values). This option was discarded because it produces a decrease in statistical power.

The differential expression analysis was performed using the algorithm *Significant Analysis of Microarrays* (*SAM*, *Tusher et al., 2001*) implemented in *siggenes* R/Bioconductor package. The contrasts were done in the same way as in the methylation analysis: (1) NoL *versus* RA (7 *versus* 6 samples), (2) NoL *versus* RARS (7 *versus* 12 samples) and (3) NoL *versus* low-risk MDS (including the samples of RA and RARS: 7 *versus* 18 samples). The three comparisons were executed using the same arguments in *SAM*: 100 permutations, two-class unpaired cases, and assuming non-equal variances. In order to use the same samples in the low-risk contrast as in the individual RA and RARS contrast, one sample with dubious RARS diagnosis was excluded from the differential expression analysis. Each of the contrasts was done over the full expression set (including all available genes) and also over a filtered expression set without the 25% of genes with lowest inter-quartile range (IQR).

2.4. Combined analysis of expression and methylation data

As DNA methylation is a mechanism to suppress gene expression, the objective of the integrative analysis was to find which genes are down-regulated and hyper-methylated. These genes, and the ones with over-expression and hypo-methylation, will be referred to as genes with *consistent* expression and methylation profiles.

The DNA methylation data used for the combined analysis with expression was the output of the differential methylation analysis. This output was formatted as tables containing: CpG island ID and locus, the result of the differential methylation test and the annotated nearby genes. For the combined analysis of expression and methylation, we look for the intersection between the differentially expressed and differentially methylated genes. This analysis started by identifying the subset of genes from the genome that would be possible to study: i.e. the gene loci that had coverage provided by both microarray platforms. Afterwards, we explored the combined distributions and intersections of the methylation and expression data considering in either case the up-regulation and the down-regulation.

For the initial visualization of the data, we used the starburst plot proposed by *Noushmehr et al., 2010*. This starburst plot combines the volcano plots[4] of two datasets into a single one by representing the $\log_{10}$ of the p-values from one volcano plot (e.g. expression) *versus* the same from the other (e.g. methylation). In order to represent the direction of the effect, the $-\log_{10}$(p-value) is multiplied by 1 or -1 depending on the sign of the expression or methylation change (i.e. up or down). This representation allows to easily see the proportion of samples in each of the up/down quadrant combinations (e.g. up-regulated and hyper-methylated). This initial screening already showed that there were very few genes clearly differentially methylated and expressed (*Figure 5* in results Section 3.3). This was confirmed by the intersection of the lists of differentially methylated and expressed gene symbols at the standard significance level, that was: p-value < 0.05 for the methylation data and FDR < 0.05 for the expression data. Although this intersection seems small, its empirical p-value was 0.003 (calculated with 100k random samplings) and the maximum combined probability for the individual genes was 0.0025 (assuming independence between the variables). Since these requirements were too restrictive and the resulting list of gene loci was too small to provide a clear view of related biological processes, the individual cut-off thresholds were increased. Setting 0.05 as maximum combined probability would have allowed to increase the individual list thresholds to 0.22 p-value or FDR. However, in order to avoid too many false positives, we finally decided to use a cut-off of p-value$<$0.15. This list provided enough potential candidates to study the related biological processed in depth, while still being statistically significant because it corresponded to a maximum combined p-value$<$0.0225 and an empirical p-value$<$0.006.

The relative positions of the CpG islands and the genes with consistent expression and methylation patterns were explored using the annotation provided in the methylation data. These annotations indicate the genes that are up-stream, down-stream or within each CpG island (*Figure 2A*). However, since this annotation is relative to the genomic position of the CpG island, it does not take into account the transcription direction of the gene. In this way, even if the gene is labeled as 'down-stream', it is not possible to know whether the CpG island is actually on the promoter of the gene or not, because that would depend on whether the gene is on the forward (+) or reverse (−) strand. Although the relative position of the gene and the island information was finally not used for the study, we consider that it would be more correct to take the transcription direction into account, and decided to redo the results in this way (as sown in *Figure 11* compared with *Figure 10*). The new annotation (*Figure 2B*), takes the gene as reference, and labels the CpG island as up-stream or down-stream from the gene. The 'strand' information available in the annotation package *TxDb.Hsapiens.UCSC.hg18.knownGene* from Bioconductor was used to identify the gene transcription direction. The maximum distance at which a gene was annotated to an island was unkown. Therefore, we also used the gene genomic location from this package to check the distances between the genes and the islands. Using human genome version 18, the median distance between the transcription start site of the genes and the up-stream or down-stream CpG islands would be 61,100bp, being 90% of them within 1,005 and 375,184bp.

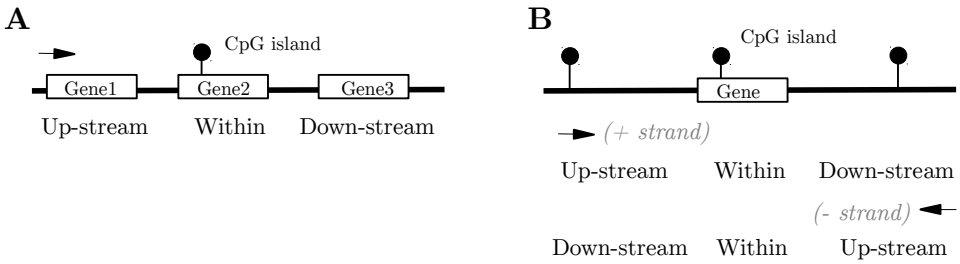


Figure 2. Labeling of the relative position of the genes and the CpG islands. (A) Taking the island as reference, and (B) taking the gene and its transcription direction as reference.

[4] A volcano plot is a common representation used in microarray studies: –log10 of the p-value *versus* the measured change (e.g. log ratio or fold-change)

3. Results

3.1. Methylation data

The output of the methylation analysis was provided as tabular files containing the results of the three differential methylation contrasts between RA, RARS and NoL (*Section 2.2*). These tabular files contained the probesets IDs and genomic coordinates, and the p-value and fold change of the given contrast (*Table 2*). The fold change in the original analysis was done taking MDS as 'reference'. Therefore a positive value means normal bone marrow is hyper-methylated compared to MDS, or that MDS is hypo-methylated. For consistency with the expression analysis and to simplify the interpretation, in the rest of this Thesis, the value was inverted so positive values represent hyper-methylation in MDS. In addition, each probeset was also annotated to the gene symbols located near to the probeset/CpG Island locus. These genes are labeled 'up-stream' or 'down-stream' depending on their relative position to the probeset loci, or 'within' if the probeset loci overlaps the gene location.

```
> Metilacion_NoLvsLowRisk[1:10,-c(7,9)]
```

Probeset.ID	within	up-stream	down-stream	GenomicCoordinates	pValue	FoldChange
UHNhscpg0011649		OR4F5	FAM87B	chr1:557999-558507	0.000027700	4.73285
UHNhscpg0003583		C1GALT1	FLJ20323	chr7:7573515-7573606	0.000064900	-3.69854
UHNhscpg0001373	DTNA	DTNA	MAPRE2	**chr18:30686547-30687058**	0.000510271	2.27822
UHNhscpg0001280	DTNA	DTNA	MAPRE2	**chr18:30686549-30687058**	0.000524326	2.27408
UHNhscpg0001101		AY026352	AK131313	chr10:41680123-41693656	0.000809729	2.59005
UHNhscpg0001595		HOXA13	EVX1	chr7:27210763-27211582	0.000940833	-1.55545
UHNhscpg0000906		BC029861	FLJ43980	**chr16:44964780-44965472**	0.001022260	1.91831
UHNhscpg0001000		BC029861	FLJ43980	**chr16:44964780-44965472**	0.001022260	1.91831
UHNhscpg0001821		BC029861	FLJ43980	**chr16:44964780-44965472**	0.001022260	1.91831
UHNhscpg0008378		PFN4	FLJ30851	chr2:24212910-24213052	0.001069740	1.5104

Table 2. Header of one of the data matrices

As indicated in Methods, although the UHN 12K CpG microarray contained 12,192 probesets, only 4,900 probesets were considered after the preprocessing steps. Several of these probesets might map to the CpG loci, and each CpG loci can also be associated to several genes. Taking this into account, *Table 4* shows the number of probesets that were found significant in each contrast, while *Table 3* shows these same statistics based on unique genomic coordinates and nearby genes (i.e. unique gene symbols annotated to the CpG islands). Note that since the overlap between the probesets' genomic coordinates is not checked, some CpG islands in *Table 3* might still be counted more than once. In this way, the 302 differentially methylated probesets in low-risk at p-value$<$0.05 correspond to 246 unique genomic coordinates (132 hyper-methylated and 114 hypo-methylated), which were annotated to 476 genes (268 genes associated to hyper-methylated loci and 218 genes to hypo-methylated).

Since the aim of these analyses was to study low-risk MDS, it focused on the contrast that compares RA and RARS together *versus* NoL. This contrasts provided 246 differentially methylated islands (including the 93 islands in the intersection between the individual contrasts for RA and RARS). According to these contrasts, it seems that low-risk MDS tend to have higher methylation than the normal samples. However, it should be noticed that while RARS also follows this trend, when taking into account only RA samples, this proportion seems to reverse. In general, we can say that there is a clear change in the methylation profile of the MDS patients affecting about 450 gene loci with clear significant signal.

For the integration of the results of the methylation and expression analyses, different p-value cut-offs of the methylation data were considered (*Sections 2.4 and 3.3*). To avoid loss of

biological signal, the one finally used was the cut-off at p < 0.15, which selects 650 differentially methylated genomic coordinates (*Figure 3*), annotated to 1198 genes.

Contrast	p-value cut-off	Number of CpG islands & genes		
		Hypo-methylated in MDS	Hyper-methylated in MDS	Total
NoL *vs* RA	p<0.05	121 (260)	96 (195)	217 (450)
NoL *vs* RARS	p<0.05	95 (167	131 (264	226 (426
NoL *vs* Low-risk (RA&RARS)	**p<0.05**	114 (217)	132 (267)	**246 (476)**
	p<0.10	215 (398)	242 (474)	457 (853)
	p<0.15	317 (591)	333 (637)	650 (1197)
	p<0.25	511 (921)	525 (991)	1,036 (1,839)

Table 3. Number of CpG island locus differentially methylated in each of the contrasts. In brackets, the number of nearby genes.

Contrast	p-value cut-off	Number of probesets		
		Hypo-methylated in MDS	Hyper-methylated in MDS	Total
NoL *vs* RA	p<0.05	145	109	254
NoL *vs* RARS	p<0.05	124	153	277
NoL *vs* Low-risk (RA&RARS)	**p<0.05**	152	150	**302**
	p<0.10	272	280	552
	p<0.15	397	382	779
	p<0.25	621	604	1,225

Table 4. Number of CpG Island probesets differentially methylated in each of the contrasts. Note that some CpG loci might have several annotated probesets.

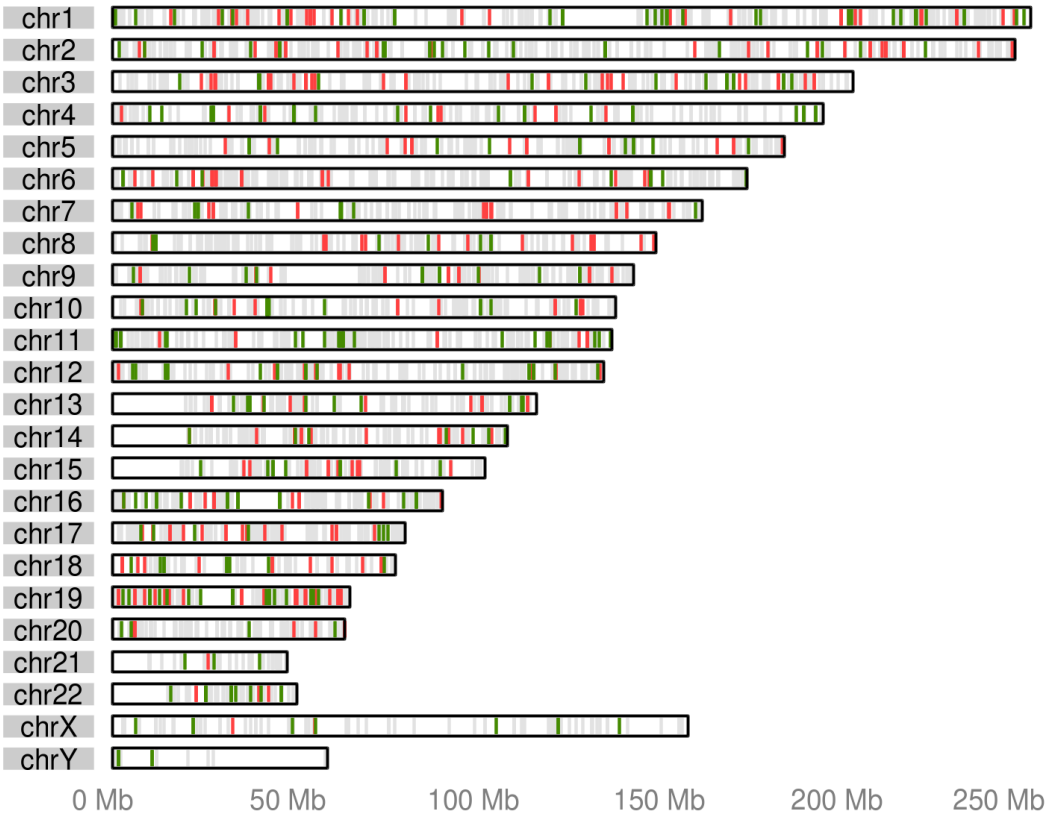


Figure 3. Location of the CpG island probesets in the chromosomes.
Red shows hyper-methylation and green shows hypo-methylation in low-risk MDS (at p-value < 0.15)

3.2. Differential expression

In order to have the same contrasts as in the methylation analysis, we performed the same three differential expression comparisons: individual RA and RARS *versus* NoL, and both together (as low-risk MDS: RA and RARS) *versus* NoL. We also did the contrasts using the full expression set –including all genes– and filtering out the 25% of genes with lowest IQR (*Table 5*).

	No filtering			25% IQR filter			Intersection
	<0.05	<0.10	<0.15	<0.05	<0.10	<0.15	<0.15
NoL *vs* RA	180	544	950	227	668	1,077	723
NoL *vs* RARS	192	946	1,974	593	1,807	3,078	1,811
NoL *vs* Low-risk	172	738	1,358	334	1,019	1,975	1,272

Table 5. Number of genes differentially expressed for each contrast at different FDR cutt-offs, using the full expression set or filtering out 25% of the genes with lowest IQR. The intersection column shows the size of the intersection of the contrasts at 0.15 FDR using the IQR filter and no-filtering (data columns 3 and 6)

In an initial screening, the different contrasts were compared to the corresponding methylation ones. This screening allowed to select the low-risk MDS contrast using the 25% IQR filter as main contrast for the rest of the study.

334 genes were identified as differentially expressed at 0.05 FDR between low-risk MDS and NoL using *SAM*. At 0.15 FDR (delta=0.767, *Figure 4*), the threshold for the combined analysis with methylation, there were 1975 differentially expressed genes; 764 of them up-regulated and 1211 down-regulated (*Table 6*), with an R-fold ranging from 0.21 to 8.08 (mean 0.97). The 1005 genes referred in the paper as differentially expressed, are the subset of these 1975 genes with q-value < 0.10.

NoL *vs* Low-risk	FDR < 0.05	FDR < 0.10	FDR < 0.15
Up-regulated	70	262	764
Down-regulated	262	757	1,211
Total	334	1,019	1,975

Table 6. Number of genes differentially expressed between non-leukemia and low-risk MDS samples at different FDR cut-offs.

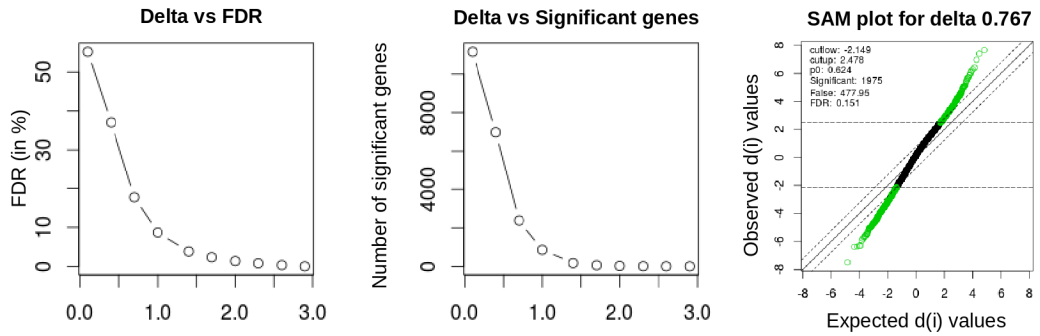


Figure 4. SAM plots for the NoL *vs* low-risk MDS contrast at FDR < 0.15

3.3. Combined analysis of expression and methylation data

While the expression microarray platform maps to 17,503 human genes, only 5,964 genes are annotated to a CpG Island in the methylation platform. Therefore, the intersection of both microarray platforms corresponds only to 4,280 gene loci (recognized with human gene symbols) and these are the only genes that the integrative analysis can study together.

In a first graphical exploration of the data (*Figure 5*), there are very few of these genes that are clearly differentially expressed and methylated. This is confirmed by the intersections of the differentially methylated genes and the differentially expressed genes. The intersection of the significantly differentially and methylated genes is only 7 genes using the standard cut-off<0.05 of FDR or p-value for each of the lists. When using a maximum combined p-value<0.025 (corresponding to 0.15 individual thresholds) the intersection increases to 122 genes (empirical p-value < 0.003, *Table 7* and *Figure 6*). Out of these 122 genes, 64 have a 'consistent' pattern of methylation and expression: hyper-methylated and repressed, or hypo-methylated and over-expressed (*Figure 7*). Although the chi-square test of these contingency tables does not confirm that there is dependency between the variables (expression and methylation), in all these data the most frequent association was hyper-methylation with under-expression.

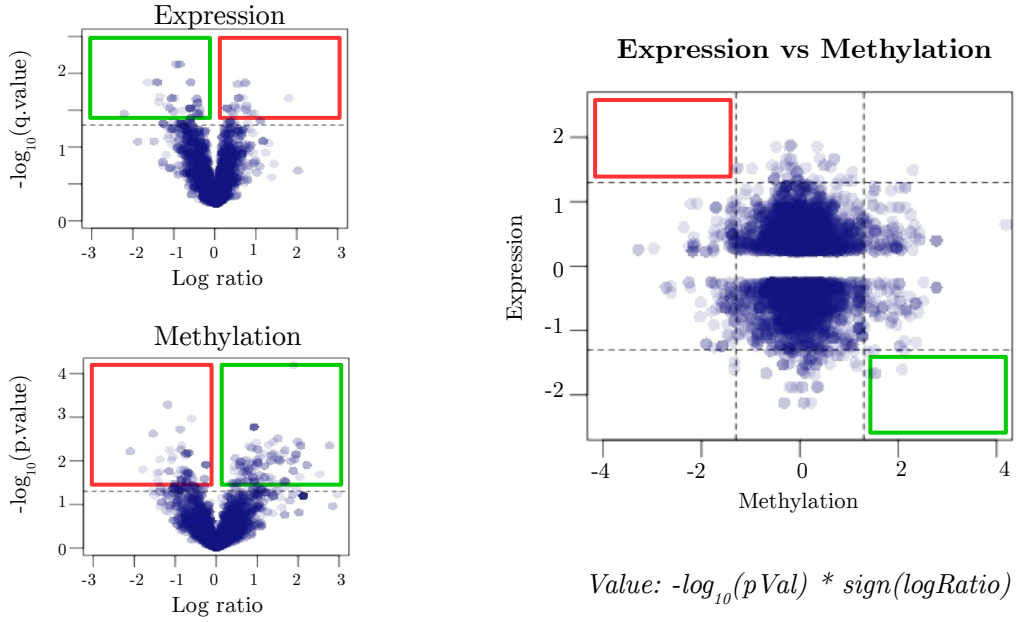


Figure 5. Volcano plots of the differential expression and methylation analyses (left) and their integration into a starburst plot (right). The dashed line at 1.3 indicates the p-value<0.05 threshold

		Methylation (477 genes)	
		218 hypo	268 hyper
Expression (334 genes)	70 up	0	3
	262 dw	3	7
	Total:	13 (Consistent: 7)	

		Methylation (1,198 genes)	
		592 hypo	638 hyper
Expression (1975 genes)	764 up	19	*22*
	1211 dw	*37*	45
	Total:	122 (Consistent:64)	

Table 7. Size of the intersections of differentially expressed and methylated genes at different FDR/p-value cuttoffs: 0.05 (left) and 0.15 (right)

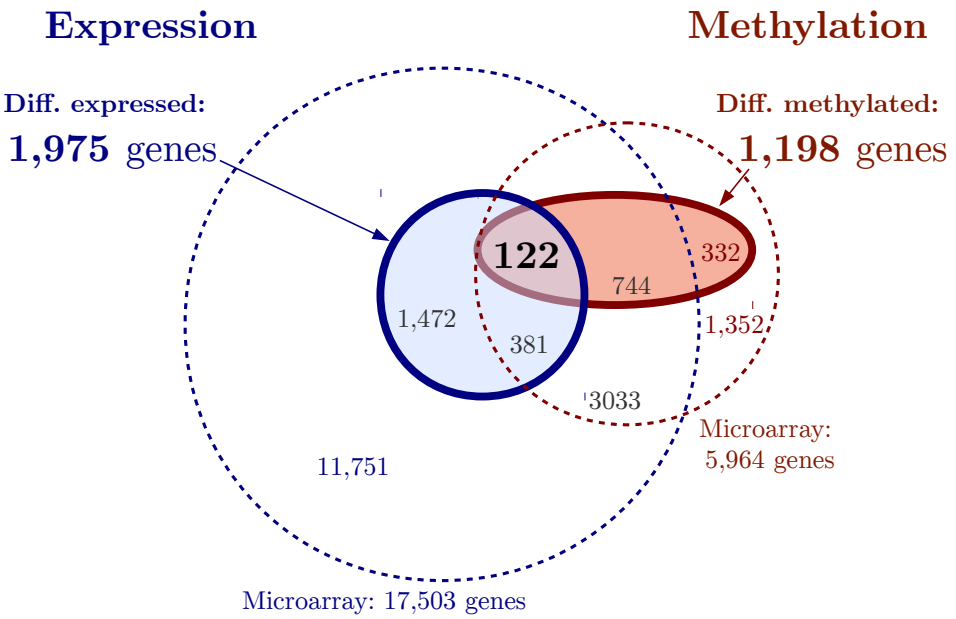


Figure 6. Intersections of the genes that are available in the expression and methylation microarrays, and the differentially methylated and expressed genes using a common cut-off p-value<0.15

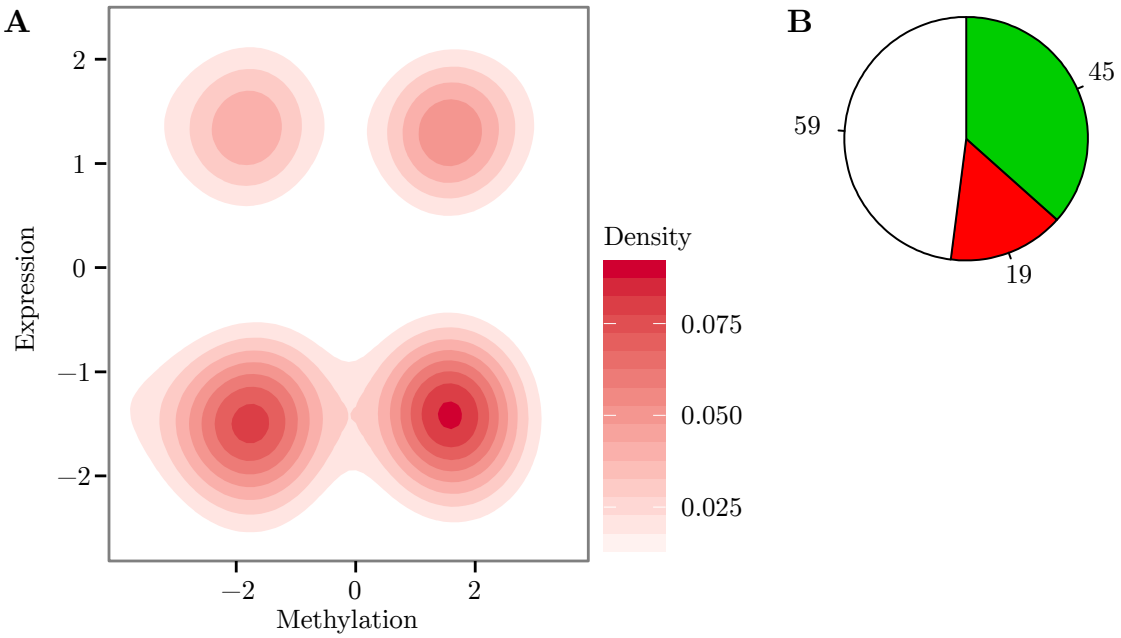


Figure 7. (**A**) Density plot of the plot change in expression *versus* methylation for the differentially expressed and methylated genes (using a common cut-off <0.15) (**B**) Pie plot of the 122g differentially expressed and methylated (**green**: hyper-methylated & down-regultated, **red**: hypo-methylated & over-expressed, **white**: inconsistent).

Next step was to consider the relative positions of the CpG islands and the genes to see whether they had any relevant implications. Also, by ignoring the island-gene relative positions, some genes might be associated to several nearby islands; for example, within the list of 122, gene PTPRC (repressed) is annotated to two CpG islands with different methylation status (one hypo-methylated and another hyper-methylated). *Figure 8* shows a similar situation for ETS1, although in this case, one of the CpG islands is not differentially methylated.

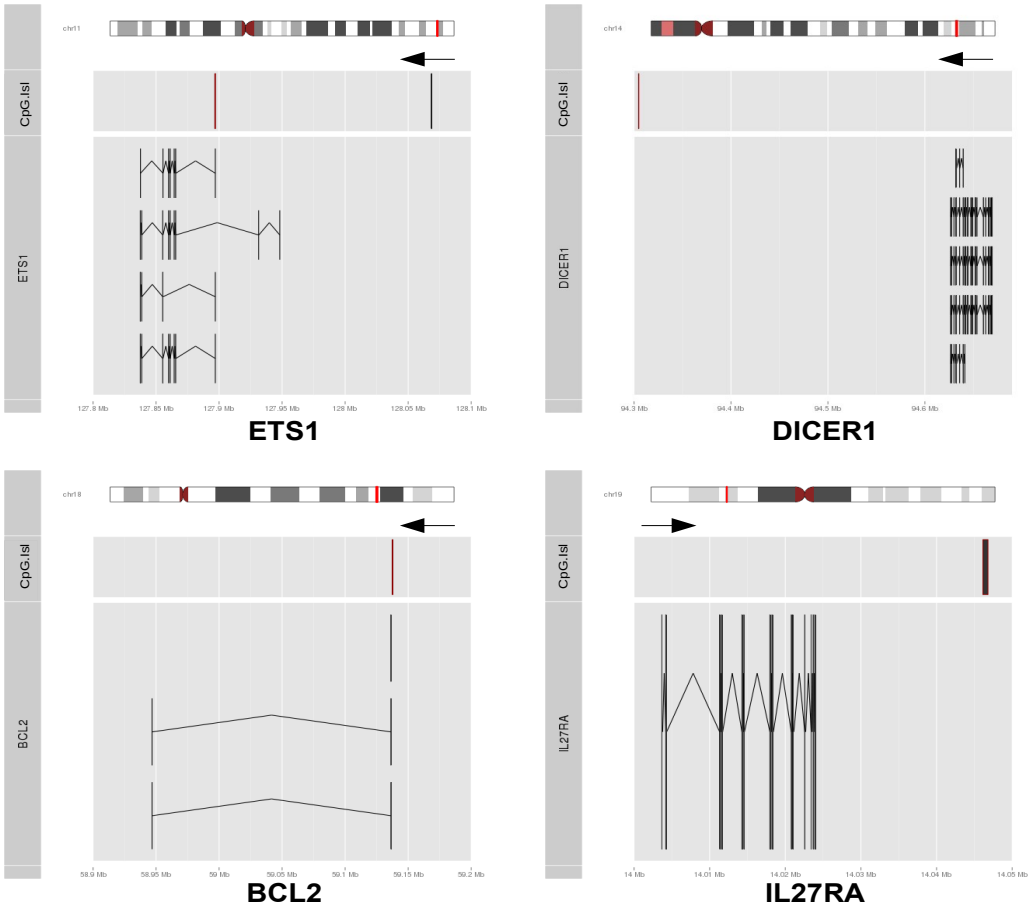


Figure 8. Locations of four genes and the closest CpG islands

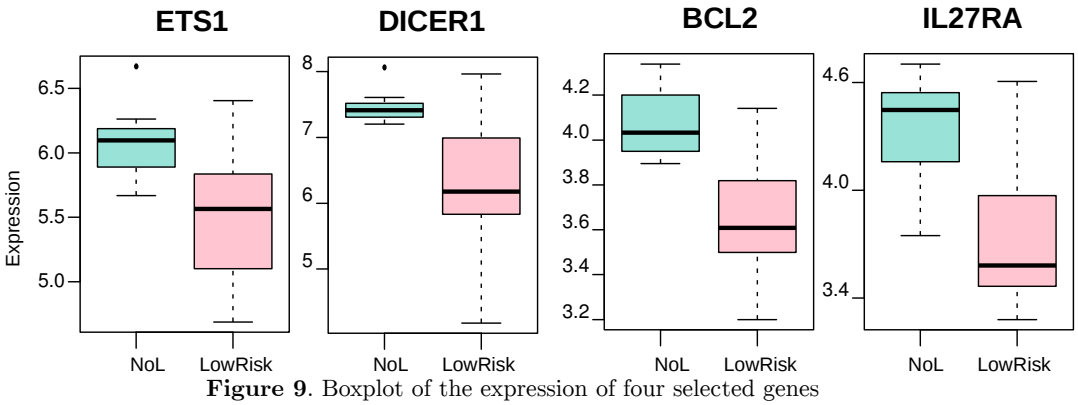


Figure 9. Boxplot of the expression of four selected genes in the NoL (green plots) and the low-risk MDS (red plots) samples

Figures 10 and *11* show the proportion of genes with consistent expression and methylation (green or red) depending on the relative positions of the gene and the island. *Figure 10* shows the data according to the annotation on the methylation files, and *Figure 11* shows the data recalculated taking into account the gene's transcription direction (Methods *Section 2.4*). *Table 8* contains the list and info about the 64 consistent genes.

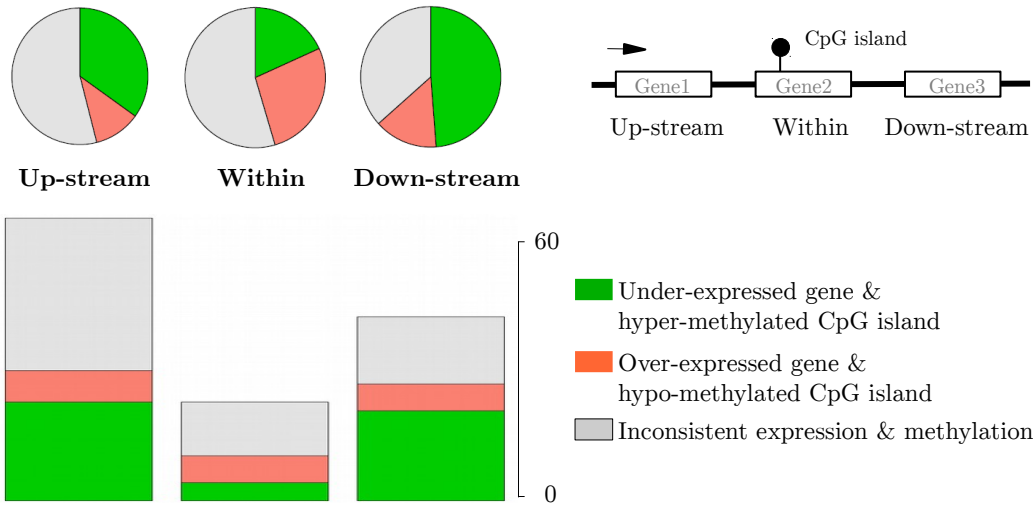


Figure 10. Consistency between expression and methylation taking into account the relative position of the CpG island and the gene (methylation data original annotation)

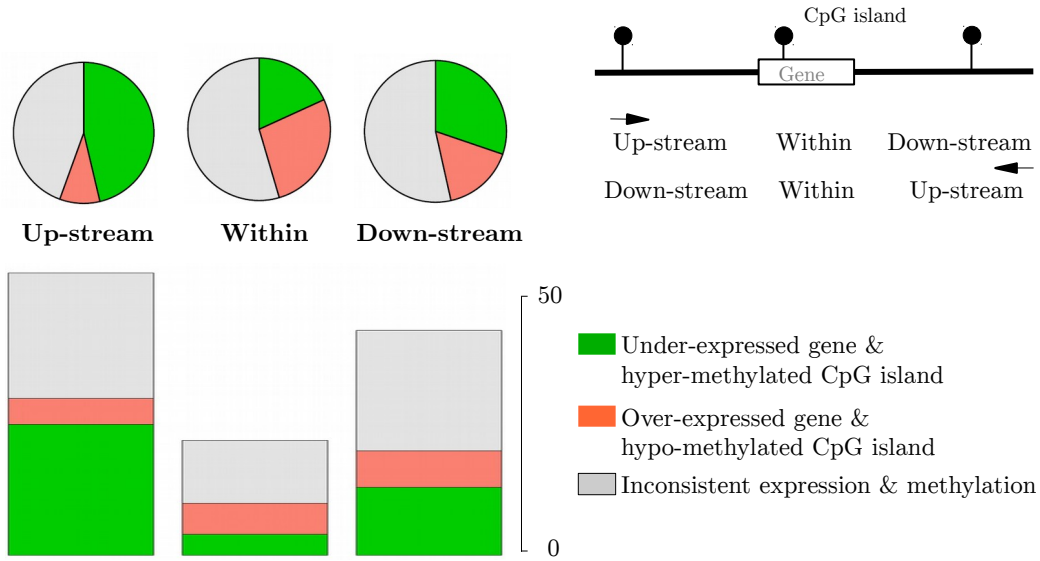


Figure 11. Consistency between expression and methylation taking into account the relative position of the CpG island and the gene (recalculated taking into account the gene strand)

	strand	Gene	CpG island position			Met. FC	Expr. FC
			up-stream	within	down-stream		
Hyper-methylated & down-regulated	+	AIG1	✓			1.81	-1.33
		ALCAM	✓			1.98	-1.75
		C10orf11	✓			1.50	-1.45
		CD28	✓			2.02	-1.15
		GIMAP2	✓			1.29	-2.22*
		HMGN4			✓	1.23	-1.49*
		IL27RA			✓	2.16*	-1.54*
		IPO9	✓			1.56	-1.18
		MAP2K1	✓			1.60*	-1.45
		NFE2L3	✓			1.62	-1.28*
		POP4	✓			1.19	-1.14
		PTPRC			✓	2.43*	-1.45
		RHOQ	✓		✓	3.21*	-1.39
		RHOU			✓	1.58	-1.69
		SCP2			✓	1.42	-1.61
		ZNF33A			✓	2.22	-1.56
		ZNF37A	✓			2.22	-1.25
	-	AK2	✓			1.50*	-1.37
		BCL2		✓		1.37*	-1.35*
		BNIP2	✓			1.36	-1.43
		CHIT1	✓			1.50	-2.44
		CHML			✓	2.55*	-1.59
		CNOT6L	✓			2.78*	-1.45
		CTSC			✓	1.85*	-1.59
		DICER1			✓	1.46	-2.22*
		ENC1	✓			2.31	-1.43
		ETS1		✓		2.29	-1.45
		FBXL17			✓	1.85	-1.18
		GNS	✓			1.65	-1.49
		IER3IP1	✓			3.25*	-1.49*
		KLHL8	✓			1.75*	-1.47
		NELL2	✓			2.12*	-2.56*
		NPHP3	✓			1.94	-1.47
		NSMCE1	✓			4.37	-1.25
		OPN3		✓		2.55*	-1.72*
		PLAGL1	✓			1.91*	-2.08*
		RFX2		✓		1.33	-1.61
		RPL36AL			✓	7.82	-1.16
		RPS6KA5			✓	1.66*	-1.72*
		SLC4A7	✓			1.44	-1.19
		ZC3HAV1	✓			1.45	-1.43
	NA	CENTD1				2.09	-1.59
		FVT1				1.37*	-1.25
		KIAA1128				1.51*	-1.28
		PH-4				1.49	-1.18

	strand	Gene	CpG island position			Met. FC	Expr. FC
			up-stream	within	down-stream		
Hypo-methilated & up-regulated	+	CDH4			✓	-1.95	1.22*
		CYB5D1			✓	-1.77*	1.22
		FADS2		✓		-2.09*	1.26
		H2AFJ		✓		-1.88*	1.4
		HCN3	✓			-4.29*	1.18
		ING1		✓		-1.52	1.37
		MLF1		✓		-1.24	1.58
		RAB8B	✓			-2.58	1.70*
		TBPL1	✓			-2.96*	1.49
	-	AAAS	✓			-1.44*	1.29
		AP3S2	✓			-1.47	1.31
		CLK1	✓			-1.24	1.32
		FXYD2			✓	-1.45	1.19
		PTPRN2	✓			-1.24	1.32
		RRAS2			✓	-2.73*	1.43
		SYN3		✓		-1.62*	1.22
		UBE2D3		✓		-1.43	1.18
		ZCCHC6			✓	-1.35	1.28
	NA	HSPA9B				-1.31	1.29

Table 8. Genes with consistent differential expression and methylation and their location relative to the CpG island (taking into account the gene transcription direction: reversed the annotation for genes in strand '-'). The last two columns contain the fold changes in expression and methylation of the most significant island, both values are relative to the signal of the NoL samples. An asterisk indicates p-value or q-value$<$ 0.05. There are 7 genes which are significant in both lists (underlined).

3.4. Further bioinformatic analyses of the genes

Functional enrichment analysis

The functional enrichment analysis of the hyper-methylated and down-regulated genes for the original publication was done through DAVID, Ingenuity Pathway Analysis 9.0 and Metacore Analytical Suite (*Figure 12*).

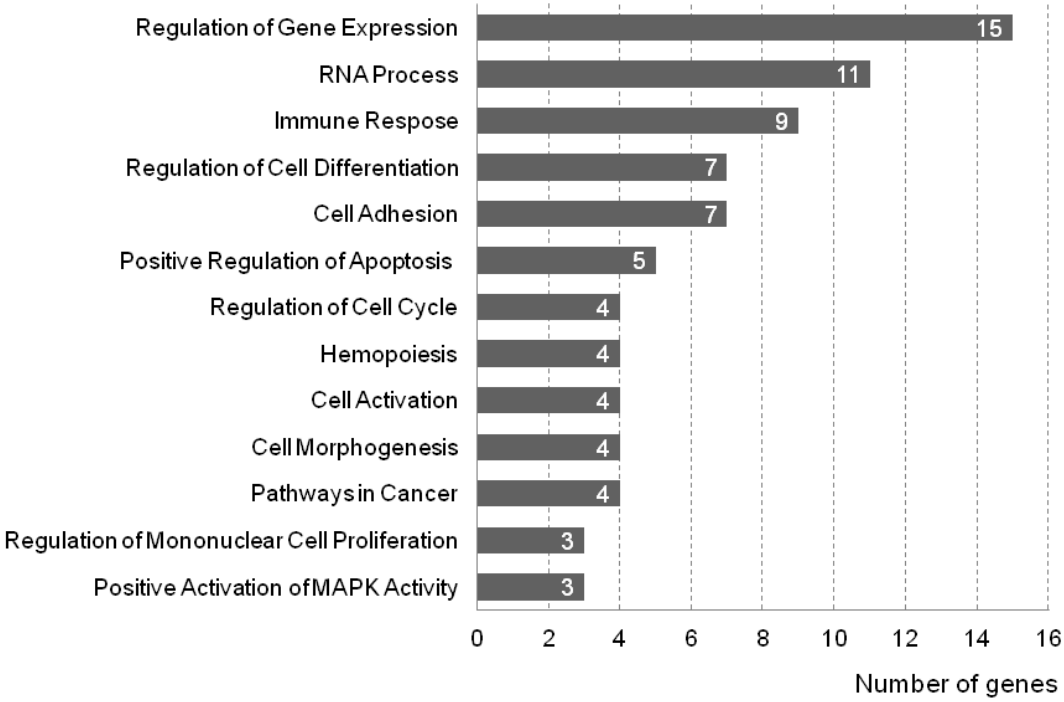
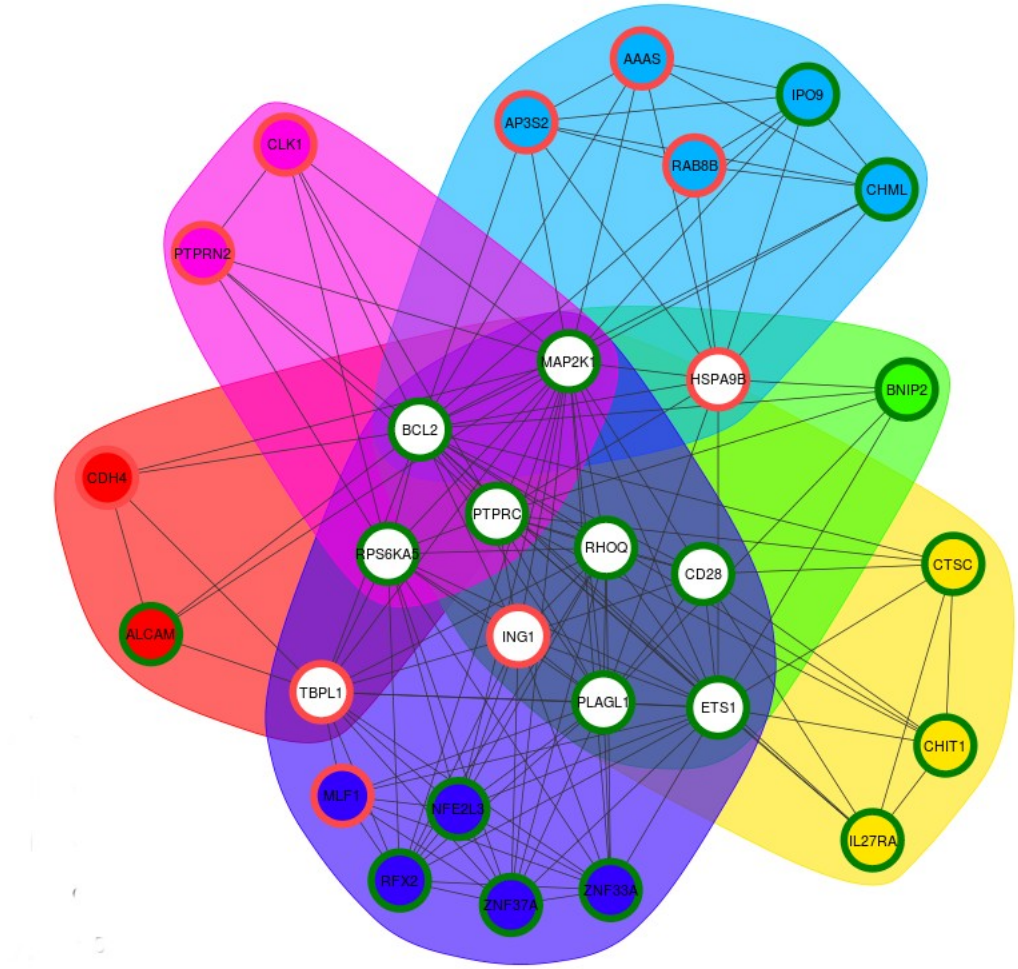


Figure 12. Functional analysis of the hyper-methylated and down-regulated genes published in Del Rey et al. (Figure 3)

The functional analysis of the 64 genes with consistent differential expression and methylation has been also done using *FGNet* (the algorithm presented in *Chapter 2*) with DAVID-FAC. *Figure 13* shows the functional network corresponding to the enriched genes and terms found in GO-BP database (Gene Ontology Biological Process, FAT) using the default arguments of FGNet. *Figure 15* shows the functional network corresponding to the genes and pathways enriched with a p-value < 0.1 (in this case excluding GO-BP terms). In addition, *Figure 14* shows the ontology-tree plot for the apoptosis-related terms in cluster 3, showing that these terms are consistently down-regulated. Although this is not taken into account by the enrichment analysis tools, is easily revealed by FGNet.

In the first overview of the results, the main functions are already highlighted (i.e. regulation of apoptosis, regulation of cell proliferation, regulation of gene expression; *del Rey et al., 2013*). In addition, the four genes selected as key genes in the study (IL27RA, DICER1, ETS1 and BCL2) are well identified with the new tool *FGNet*. Specially noting that ETS1 and BCL2 are selected as potential inter-modular hubs in the network, and highlighting the potential interest of MAP2K1, RHOQ, CD28, PLAGL1, that appear together in the green group with ETS1 and BCL2.



Cluster	Terms (GO-BP)
1	Neuron projection morphogenesis
2	Immune response Positive regulation of macromolecule metabolic process * Regulation of cell cycle Regulation of cell proliferation
3	Positive regulation of apoptosis Positive regulation of macromolecule metabolic process *
4	Intracellular protein transport
5	Positive regulation of cellular biosynthetic process Positive regulation of gene expression Positive regulation of macromolecule biosynthetic process Positive regulation of RNA metabolic process Regulation of transcription from RNA polymerase II promoter
6	Cell proliferation Protein amino acid phosphorylation

Figure 13. Functional network for the genes with consistent methylation and expression patterns
Annotation: GO Biological process (default clustering args).
The legend shows only terminal terms (leaves)

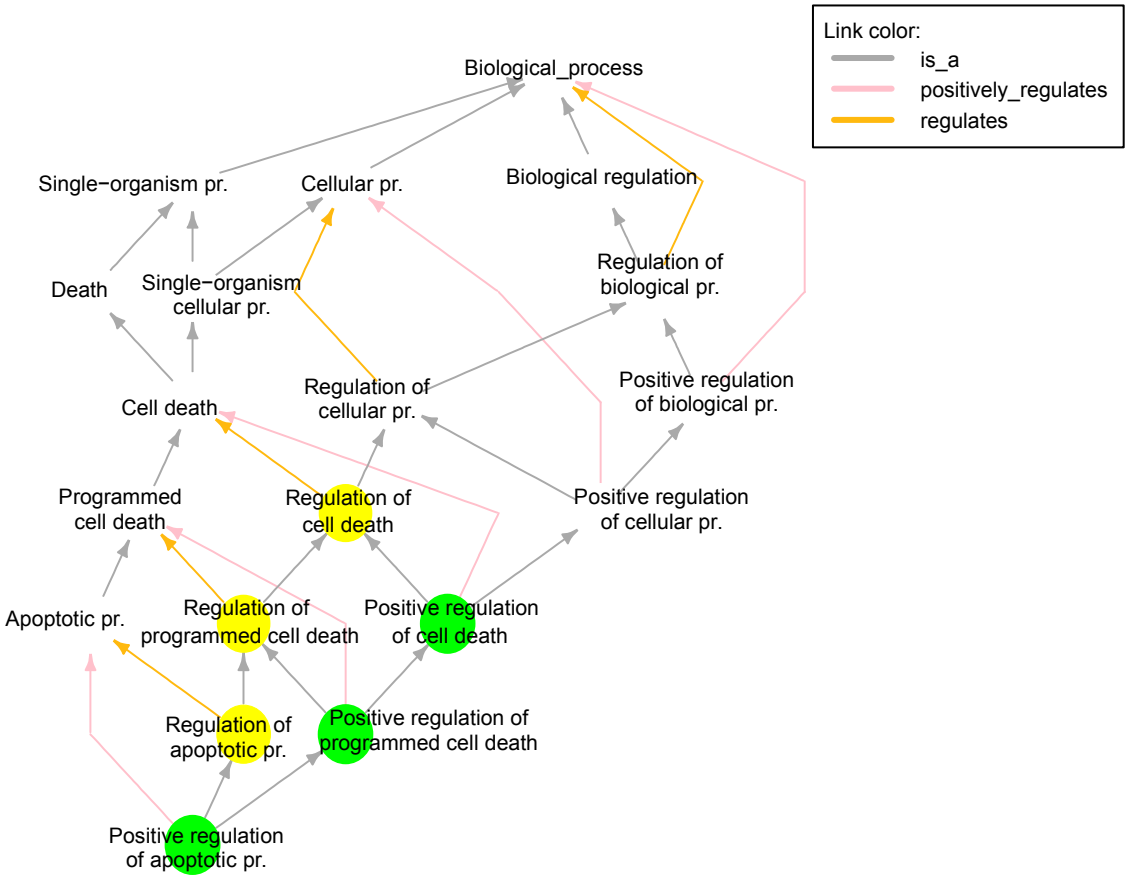
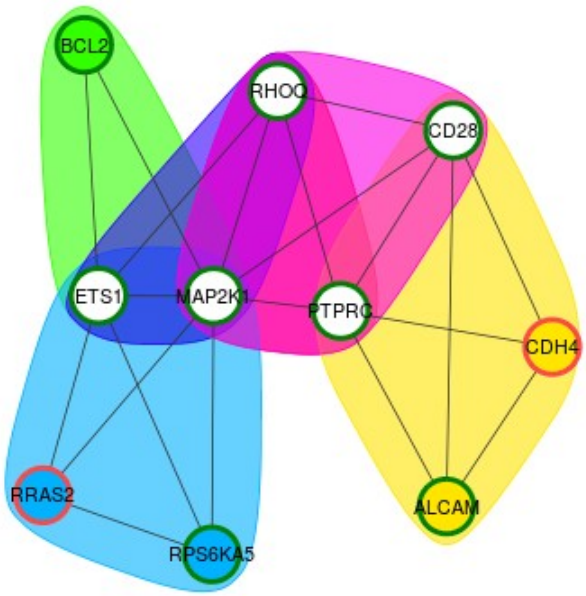


Figure 14. GO ontology tree plot for the apoptosis-related terms in cluster 3



Pathway	Database	Terms	P-value
1	PANTHER	B cell activation (ID P00010)	0.07
2	KEGG Pw	Cell adhesion molecules (CAMs)	0.02
3	BIOCARTA	Keratinocyte Differentiation	0.01
4	PANTHER	PDGF signaling pathway (ID P00047)	0.06
5	PANTHER	Ras Pathway (ID P04393)	0.07
6	PANTHER	T cell activation (ID P00053)	0.02

Figure 15. Functional network for the genes with consistent methylation and expression patterns Annotation: Pathways (excluding GO-BP terms).

Transcription factors

Since there were several known transcription factors (TFs) hyper-methylated and down-regulated (i.e. ETS1, PLAGL1, NFE2L3, ZNF37A), the promoters of the other down-regulated genes were explored in search for common transcription factor binding sites (TFBSs). We used several bioinformatic tools to search for enrichment in specific TFBSs and TFs: oPOSSUM[5] (*Ho Sui et al., 2005*), TransFind[6] (*Kiełbasa et al., 2010*), Pscan[7] (*Zambelli et al., 2009*) and TFM-Explorer[8] (*Tonon et al., 2010*). These analyses found that ETS-family binding motifs were within the most consistently over-represented TFBSs (about 80-100 out of 561 genes contain ETS binding sites). This is a relevant result for the MDS study as ETS transcription factors had been previously reported as related to cancer. In particular, ETS1 was known to control the expression of genes implicated in biological processes like apoptosis, differentiation, hematopoiesis, and cellular proliferation, which had been identified as altered processes in MDS in previous analyses (*Seth and Watson, 2005*). Since the hyper-methylation of ETS1 could be leading to the down-regulation of many of the target genes, ETS1 was selected for further validation as one of the key regulators in the study.

miRNAs and DICER

Another of the clearly hyper-methylated and down-regulated genes was DICER1 –known to be essential in miRNA processing– (*Rhyasen and Starczynowski, 2012*; *Reid et al., 2009*; *Erdogan et al., 2011*; *Hussein et al., 2010*; *Merkerova et al., 2011*). Since miRNAs had been described as de-regulated in MDS by several studies (*Merkerova et al., 2011*; *Merkerova et al., 2014*), RNA processing genes were further investigated. This search found that some other genes involved in miRNA processing were also altered in the low-risk MDS samples: NFE2L3 and POP4 are hyper-methylated and under-expressed, and ATXN1 is under-expressed.

For further validation of a possible down-regulation in the expression of miRNAs in MDS patients we look for genomic public data that could answer this question. In this way we found some samples available on *Affymetrix GeneChip Human Exon Arrays* from a parallel study on similar non-leukemia *versus* low-risk MDS patients. These arrays were used to compare the miRNA levels of the 183 miRNAs available in the chip. Although no significant differences were found for some specific requested miRNAs (i.e. miRNA-145 and miRNA-196), there was a general down-regulation of the miRNAs in low-risk MDS patients compared to the control (Wilcoxon p-value: 0.039, *Figure 16*).

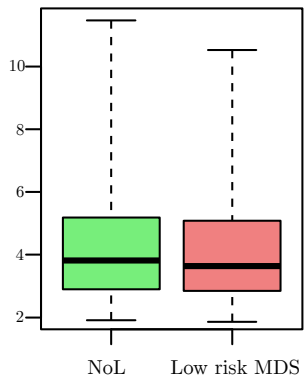


Figure 16. Box plot of the expression of 183 miRNAs

[5] http://www.cisreg.ca/cgi-bin/oPOSSUM/opossum [6] http://transfind.sys-bio.net

[7] http://159.149.160.51/pscan/ [8] http://bioinfo.lifl.fr/TFM/TFME

4. Discussion

This study allowed to identify a set of genes whose activation pattern and methylation profiles are altered in low-risk MDS. In this way, we found a correspondence between the down-regulation in the expression of certain genes and the hyper-methylation of their regulatory CpG regions. Since methylation is a mechanism of gene expression regulation, the hyper-methylation of these genes could be the underlying reason for their down-regulation.

In order to identify the altered genes, the first step was to analyze the genome-wide expression and methylation datasets separately. This allowed to identify the differentially expressed genes, and the differentially methylated CpG islands in low-risk MDS *versus* the controls. To integrate these two lists, we used the annotation file from the methylation microarray. Since this annotation links the island loci to nearby genes, we only needed to find the intersection between the methylation and expression lists. However, this approach probably lost some gene coverage. In fact, 1,685 of the total 5,964 gene symbols and IDs associated to the CpG islands were not available in the expression microarray. Most of these missing genes might indeed not be available on the expression platform (i.e. many of them are *putative uncharacterized proteins* like FLJ42280), but some of the 331 differentially methylated genes might have been lost due to the annotation. For example, by manual mapping we found that gene SELT corresponds to ID AC069236.27 included in the MCA arrays, and this gene was also available in the expression platform. In addition, in this annotation file some islands seem to be very far away from the genes they have been annotated to.

An alternative to address both issues could have been to link the genes and CpG islands from both datasets through their genomic coordinates, instead of using the annotation provided by the methylation microarray (which only indicates the genes in the surroundings of the CpG islands). In this way, we could have identified the CpG islands within the defined range of the differentially expressed genes. This may provide better coverage and integration to detect the associations between methylation and expression, than simply looking for the intersection at the level of annotated genes. Nowadays there are already some bioinformatic tools that take the genome-based approach to do this kind of analyses (e.g. MethylMix, *Gevaert, 2015* and COHCAP, *Warden et al., 2013*).

Despite these chances to improve, the results of the integrative analysis of expression and methylation were useful and led to discover new molecular features characterizing the cancer disease studied: low-risk myelodysplastic syndromes. Not only it identified the set of genes with concordant methylation patterns –which are the genes whose expression is most likely to be altered by the aberrant methylation patterns–, but it allowed to discover some specific mechanisms by which the hyper-methylation of these genes could be the underlying cause for other aspects of the disease. In fact, the functional enrichment analysis identified some of the biological effects that the hyper-methylated and repressed genes could be provoking: **(1)** the increased apoptosis detected in low-risk MDS that could be led by the repression of BCL2; **(2)** an observed inhibition of leukocytes differentiation –i.e. lymphocytes activation– marked by the down-regulation of several signaling genes like CD28, MAP2K1, PTPRC and RHOQ; **(3)** the immune response alteration through the down regulation of genes such as IL27RA; and **(4)** the alteration of miRNA processing in low-risk MDS patients associated to the down-regulation of DICER1 and de-regulation of many of its interacting genes. In addition, the analysis of transcription factor binding sites also revealed that ETS1 had many potential targets within the down-regulated genes, and therefore **(5)** the under-expression of ETS1 could be the cause of the non-activation (i.e. not expression) of many of its targets (i.e. BCL2).

In conclusion, this study illustrates how the application of robust bioinformatic analyses, and the integration of different layers of omics data, provide a way to achieve new knowledge and better molecular understanding of diseases, and can also lead to discover potential underlying causes or drivers of the pathological states studied.

Chapter 4

Study 2: Integration of multi-platform gene expression profiles and identification of expression patterns in the progression of myelodysplastic syndromes to leukemia

Chapter index

1. INTRODUCTION

Myelodysplastic syndromes (also known as MDS or myelodysplasia) are a heterogeneous group of hematological malignancies with ineffective production and maduration –i.e. dysplasia– of blood cells that many times includes anemia symptoms. In spite of the efforts to identify and categorize MDS, accurate pathology diagnosis and classification of these bone marrow disorders still remains a challenge.

The main diagnostic criteria to identify MDS as a hematological cancer is the detection of 'blasts', malignant cells in peripheral blood and bone marrow. Other diagnostic criteria include the detection of abnormal morphologic characteristics in the blood cells, such as the degree and lineage of the cytopenia and dysplasia or the presence of ring sideroblasts (abnormal erythroblasts with iron granules forming a ring). The presence of cytogenetic abnormalities (i.e. alterations in chromosome profile and karyotype) was added to the diagnostic criteria in the last WHO classification (*Vardiman et al., 2009*). However, MDS are still difficult to identify and diagnose, specially in its early stages when the percentage of blast cells is very low.

Being MDS a difficult disorder to identify, it is many times harder to provide a clear medical sub-classification in its different stages or subtypes. For example, when there are environmental factors provoking a secondary dysplasia the identification of MDS can be diffused (*Vardiman et al., 2009*). Even within the subgroups currently defined by medical consortiums, there is still considerable clinical heterogeneity (*Theilgaard-Mönch et al., 2011*). This heterogeneity is also reflected at genetic level, where there are several features known to be associated with MDS (i.e. the chromosomal alterations used for diagnosis, mutations in RUNX1, TP53 and ETV6, ...), but each of them is not necessarily present in every patient (*Raza and Galili, 2012*).

In practical terms, the current sub-divisions within MDS are mostly based on the patient prognosis and outcome. It was observed that the separation between the MDS patients with a good and bad outcome was maximized by establishing the division at certain threshold values (blasts $\leq 5\%$, $\leq 10\%$ or $\leq 20\%$). In this way, the percentage of blast cells in the blood and bone marrow is used as the most clear diagnostic tool (having bad survival prognostic when the percentage of blasts is high and good when it is low or not detectable). However, it is not clear whether there is actually much pathological difference between a patient with 4% of blast cells and another with 6%. Therefore, it would indeed be very interesting to investigate the biomolecular underlying reasons that could be provoking the difference in survival and drive the progression in the percentage of blasts.

In order to investigate on these questions, we started a new collaborative study on MDS drived by onco-hematologists from the University Hospital of Salamanca (HUS) and the Cancer Research Center (CiC-IBMCC). This team had several cohorts of MDS patients that had been studied using genome-wide expression platforms. The collection of these analyses provided three different expression microarray datasets from clinical series including many samples from myelodysplastic syndromes. The objective was to integrate them into a single study with a big number of MDS patients at different stages of the disease. All the microarrays had been hybridized with samples taken from bone marrow from MDS subtypes, from AML (acute myeloid leukemia) and from non-leukemia controls. However, the three datasets corresponded to different types of expression array platforms that had been done at different times with different hybridization protocols (i.e. three *batches*). Therefore, the bioinformatic integration and analysis of these three *batches* to build a unified study was a challenge. Once the datasets were integrated, the specific approach to follow for their analysis was open. The main objective was to study the pathogenesis of MDS –i.e. the origin and development of the disease– finding out the genes and biological functions that are altered in the different stages of MDS (from the early low-risk stages to the late high-risk MDS that is closer to leukemia). In this way, the main hypothesis underlying the study would be that MDS present different progressive stages that range from good to bad prognosis and in many cases can evolve stage-by-stage to acute myeloid leukemia (*Figure 1*).

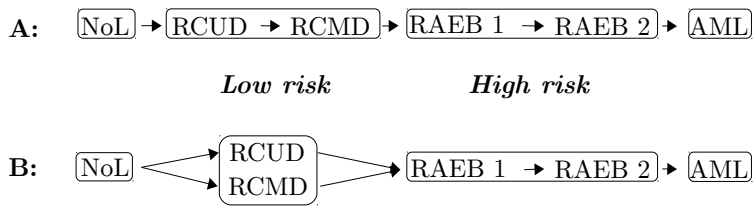


Figure 1. Hypothesis of progression of MDS from low-risk to high-risk and to acute leukemia (AML). In the low-risk MDS two ways or possibilities can be considered: that subtypes RCUD and RCMD are consecutive steps (A) or alternative paths (B) during the evolution of the disease.

In this methodological context and defined objective, we considered that one of the best ways to study the evolution or progression of a disease was to take such subtypes as progressive pathological stages of the disease. This could be done by studying the increasing or decreasing gene expression trends to explore the expected evolution from early MDS (i.e. low-risk MDS) to late MDS (i.e. high-risk MDS) and to AML. Not all the MDS cases will evolve to AML, and the suggested evolution of the disease may be a continuous rather than a discrete event. In any case, the analyses of the transcriptomic profiles along different stages will be the way for finding the genes whose expression might be associated with higher risk or malignancy, and also the genes that might be associated to early states of the disease. Finding and analyzing these genes became the core of our study.

Since this project is still in progress, its **results have not been published** yet. They are part of **a collaborative study with the group of onco-hematology** at the CiC-IBMCC, leaded by Dr. Jesus María Hernández-Rivas, and is part of the research work of Maria Abaigar in his group. In this collaborative work we developed the methodology and bioinformatic analyses as main parts presented in this Doctoral Thesis. Therefore, the main sections explained in this chapter include: **(i)** the pre-processing of the transcriptomic data and integration of the multi-platform datasets, **(ii)** the co-expression analyses and identification of genes with increasing or decreasing trends, **(iii)** the analyses of biological enrichment in the found gene-sets, and **(iv)** the methods to identify transcription factors that may regulate the transcriptomic signatures analysed. The results section includes the outcome of the analyses for validation of the methodology, but it is not within our aims to study the biomedical interpretation and implications of the gene lists and patterns. In this way, the part of the results referring to the transcriptomic profiles and patterns of **MDS** and the biomedical conclusions about this specific **disease** will be prepared for a new **scientific publication** led by this group, that is not the main scope of this chapter.

2. Methods

2.1. Samples, patients information and data pre-processing steps

Samples distributed in data series and platforms

The samples analyzed in this study come from three different series of expression microarrays that studied myelodysplastic syndromes and acute myeloblastic leukemia (*Table 9*). The first two series, including the largest number of samples, use *Affymetrix* Human Genome U133 Plus 2.0 Arrays (*hgu133plus2*). However, these two series were produced at different times and hybridized following different molecular biology protocols. Therefore, they can not be mixed using the standard pre-processing tools. The third series, the most recent, uses *Affymetrix* GeneChip Human Exon ST 1.0 Array (*HuEx1.0*).

Series	Microarray platform	# samples
Series 1	*Affymetrix* GeneChip Human Genome U133 Plus 2.0 Array (*hgu133plus2*)	113
Series 2		101
Series 3	*Affymetrix* GeneChip Human Exon 1.0 ST Array (*HuEx1.0*)	87

Table 9. Data series included in this study

In order to do a unified integrative study –having as many samples per sub-type of MDS as possible– it would have been desirable to merge all three series. However, this option was discarded because the microarray platform used in Series 3, *HuEx1.0*, has much more gene coverage than *hgu133plus2* and includes probes in all exons of each known human gene locus. If all three series had been merged, all this extra information would have been lost –or we would have need to find a way to integrate it afterwards–. In addition, the biomedical classification and diagnosis of the samples in Series 3 was considered more confident than in the previous series, since it had been done more recently and the patients had been followed up to know their evolution. In fact, about half of the AMLs in Series 3 were known to be 'secondary AML' (i.e. acute leukemias in patients that previously had MDS). For all these reasons, it was finally decided to keep Series 3 separate, and use it as the most complete data series for the study. On the other hand, Series 1 and Series 2 were done on the same microarray platform and included similar cohorts of patients. Moreover, each one individually had too few samples in some of the subtypes of MDS that we wanted to study. Therefore, it would be good to find a way to integrate them into a single series.

Since Series 3 was going to be used independently, it was preprocessed in the standard way: performing background correction, normalization, and summarization with *RMA* (*Irizarry et al., 2003a*) using the *gene mapper* CDF from *GATExplorer* (*Risueño et al., 2010*) (Ensembl Version 57, March 2010). Three outlier samples were detected with *arrayQualityMetrics* (*Kauffmann et al., 2009*). They were not from MDS classes in the focus of the study, so they were removed and the remaining data re-preprocessed, producing for Series 3 a normalized data matrix with 84 samples.

Integration of two independent series: Series 1 and Series 2

In order to integrate Series 1 and 2, each dataset was normalized individually using the algorithm *Frozen RMA* (*fRMA*) (*McCall et al., 2010*) and then merged both of them into a single dataset with the tool *InSilicoMerging,* a Biconductor package that implements different methods to merge batches (*Taminau et al., 2012*).

Frozen RMA is a variation of *RMA* in which the statistical parameters required for the normalization –probe-specific effects and variances– are not calculated based on the specific dataset. Instead, it uses parameters previously calculated based on a large collection of multiple datasets from the same platform. If these parameters are calculated on a big enough number of datasets, containing samples with enough biological and technical variability, the parameters should represent the potential variability on any sample. Therefore, they could be applicable to any other microarray from the same platform. In this way, this set of common parameters allows to pre-process either a single array or arrays from different batches. According to the authors, "*fRMA* is comparable to *RMA* when the data are analyzed as a single batch and outperforms *RMA* when analyzing multiple batches" (*McCall et al., 2010*).

These parameters are available for many microarray platforms in Bioconductor, but only for the default chip definition file (CDF) that maps to *Affymetrix* probeset IDs rather than to genes. In order to use the *gene mapper* CDF from *GATExplorer*, the *fRMA* parameters needed to be recalculated. To do so, we built and used a collection of 1,335 human samples from 163 GEO datasets, grouped into 267 batches of 5 samples each (being these batches groups based on study and tissue, as recommended in the paper).

Once the parameters required to use *fRMA* were ready, Series 1 and Series 2 were preprocessed separately. The remaining batch effect was removed with *InSilicoMerging* using *Combat* (*Johnson et al., 2007*).

Samples and patient information

The samples used in this study come from bone marrow aspirates from patients with different subtypes myelodysplastic syndromes –classified according to the World Health Organization criteria– (summary in *Table 10*). The main subtypes or classes of MDS are refractory cytopenia with uni-lineage or multi-lineage dysplasia (RDUD and RDMD), as representative of low-risk MDS, and refractory anemia with excess of blast cells (RAEB) type 1 (for blasts percentage between 5 and 10%) or type 2 (for blasts percentage between 10 and 20%), as representative of high-risk MDS. Moreover, some MDS present ringed sideroblasts (RS) or alterations in their karyotypes, although the normal karyotype is the most frequent.

In addition, the study also includes samples from AML patients and control samples from donors that may have had other hematological non-malignant pathologies labeled as 'no leukemia' (NoL) samples. In Series 1 and 2 it is unknown whether the AML patients previously had a MDS or not. However, in Series 3, there are 5 samples of AML which are known to be 'secondary AML' (i.e. MDS that transformed into AML).

Series 1 (*hgu133plus2*):

	Karyotype					Total	Blast %
	Normal	Abnormal	Complex	Transloc.	N/A	(n=113)	
NoL	14	0	0	0	0	14	–
RCUD	2	0	0	0	0	2	1.0 -1.2
RARS	10	1	0	0	0	11	0 - 1.0
RCMD	12 (6)	2	0	0	0	14	0.3 - 3.3
RCMD (RS)	1	1	0	0	0	2	0 - 0.8
RAEB 1	8 (3)	1	1	0	0	10	5.0 - 9.0
RAEB 1 (RS)	1	0	0	0	0	1	8.0
RAEB 2	3 (1)	0	2	0	0	5	11.0 - 19.2
AML	33 (6)	9	6	0	2	50	22.0 - 97.0
Not confirmed	4	0	0	0	0	4	

Series 2 (*hgu133plus2*):

	Karyotype					Total	Blast %
	Normal	Abnormal	Complex	Transloc.	N/A	(n=101)	
NoL	11 (11)	0	0	0	0	11	–
RCUD	7 (7)	0	0	1	0	8	0.6 - 2.0
RARS	6	1	0	0	0	7	0 - 3.0
RCMD	11 (11)	3	0	2	0	16	0 - 4.0
RCMD (RS)	10	2	0	0	0	12	0 - 4.0
RAEB 1	1 (1)	0	1	0	1	3	5.2 - 6.0
RAEB 2	5 (4)	0	1	0	0	6	10.0 - 17.0
AML	17 (4)	11	4	0	0	32	20.0 - 92.0
Not confirmed	5	1	0	0	0	6	

Series 3 (*HuEx1.0*):

	Karyotype					Total	Blast %
	Normal	Abnormal	Complex	Transloc.	N/A	(n=87)	
NoL	12 (6)	0	0	0	0	12	–
RCUD	5 (4)	0	0	0	0	5	1.0 - 1.4
RARS	8	2	0	0	0	10	0 - 3.0
RCMD	8 (6)	3	0	0	0	11	0.3 - 4.0
RCMD (RS)	5	1	0	0	0	6	0 - 4.0
RAEB 1	10 (5)	0	2	0	1	13	5.0 - 8.6
RAEB 1 (RS)	1	0	1	0	0	2	6.0 - 8.0
RAEB 2	8 (5)	1	2	0	0	11	10.0 - 19.2
AML	5 (5)	3	0	0	0	8	20.0 - 81.0
AML sec.	5 (5)	3	0	0	1	9	20.0 - 61.0

Table 10. Phenotypic information of the samples available in the three series included in the study. The numbers in brackets indicate the samples that were collected as mononuclear bone marrow cells (isolated using Ficoll gradient). The rest of the samples were just whole bone marrow cell extracts. (For the final selection of samples see *Table 11*)

Exploration of all available samples

Out of all the available samples, this study will focus on four specific myelodysplastic syndromes subtypes: RCUD, RCMD, RAEB 1 and RAEB 2, plus the NoL samples –as non-malignant controls– and the AML samples –as the late acute stage of the disease– to do a comparison between them all.

The MDS samples with ringed sideroblasts (RS) were discarded because the ringed sideroblasts are known to have a very strong expression signature and they usually do not evolve to AML (*Vandelmolen et al., 1988*). However, since the MDS samples presented some heterogeneity with regard to karyotype and whether the cells had been selected by Ficoll. It was needed to decide if it was better to use all the available samples (and thus, to have the highest amount of samples per class), or to discard the ones not done with Ficoll or specific karyotypes to have a more homogeneous population. In addition, it was also needed to determine whether it was worth keeping RCUD and RDMD as separate classes, since it was not clear if the expression profiles of these subtypes would be different.

In order to resolve these issues, some analyses were done previous to the study:

1. **Karyotype**:

Most of the available samples included in the study present normal karyotype, but some of them do have some chromosomal abnormalities. It is known that chromosomal translocations have a strong effect on expression patterns (*Harewood et al., 2010*). However, other alterations could have a lower influence. In order to decide whether these alterations were just one more aspect within the standard heterogeneity of the samples, or if we should use only the samples with normal karyotype (in spite of reducing the cohort of samples using for the study), we explored the differences between normal karyotype and other karyotypes within the MDS samples and AML samples.

To test whether there are clear differences between normal karyotype samples and other karyotypes, multiple differential expression contrasts were done using *SAM* and *Limma* algorithms. Although no statistical differences were found within the MDS subtypes, the results indicated that there were not enough samples to provide a conclusive FDR (ie. *Delta versus FDR* and *Delta versus significant genes* plots). However, there were very clear differences between AML normal karyotype *versus* non-normal karyotypes (442 differentially expressed genes at 0.05 FDR comparing 42 *versus* 26 samples from Series 1+2), and some of specific karyotypes (with *Limma*: 123g differentially expressed in the 16 abnormal karyotype samples, and 159 genes in the 10 samples with complex karyotype). In this way, it was concluded that only samples with normal karyotype would be used. However, in Series 3, due to the reduced number of AML samples with normal karyotype, that would imply using AML samples with higher percentages of blast cells.

2. **Cell types**:

All the samples analyzed in the study were obtained by hybridizing the microarrays with RNA from cells from bone marrow aspirates. However, while some samples used the whole aspirate, which includes all the bone marrow cells, others had selected only the mononuclear cells by density gradient centrifugation in Ficoll (*Figure 2*), which excludes cells like granulocytes and erythrocytes. This information was not considered at the beginning of the study, and it was not known for many samples (they might include all cells or only mononuclear cells). Therefore, it was needed to decide whether it was better to use all available samples, or just those which were known to include only mononuclear cells.

After comparing the samples probably including all the bone marrow cells *versus* the ones including only the mononuclear cells, it was clear that there were differences between both types (i.e. 70 genes differentially expressed within the 11 'Ficoll' and the 13 'unknown' no-leukemia samples from Series 1+2, and 1,005 genes in the 6 *versus* 5 samples from Series 3). Therefore, only the samples confirmed to be from mononucleated cells were selected for our study.

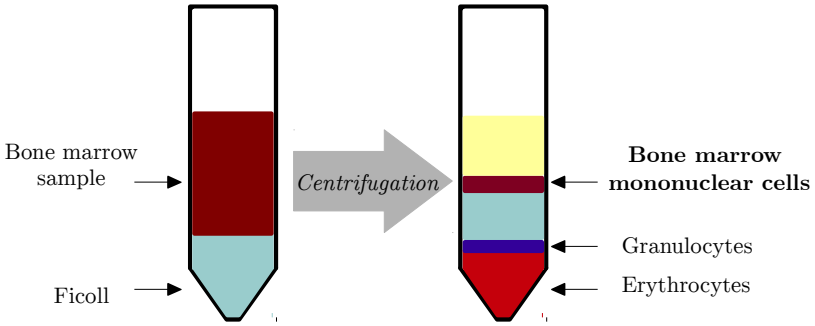


Figure 2. Scheme of isolation of mononuclear cells from human bone marrow aspirates by density gradient centrifugation (Ficoll)
(Source: Based on the figure from Miltenyi Biotec protocol)

3. **RCUD *versus* RCMD**:

MDS subtypes RCUD and RCMD are both very similar, presenting the same range of 0-5% of blast cells. Their main difference is how many cellular lineages are affected by the dysplasia (only one, or more than one). In this way, it was needed to check their expression profiles to see if they were different enough to be worth keeping them as separate classes or, instead, enclose them as low-risk MDS.

In order to check whether there are very obvious differences between them, SAM was used to compare the available samples from RCUD *versus* RCMD in both datasets. Since none of the contrasts provided clear significant genes (FDR<0.05), we also compared only the RA samples within RCUD, just in case the combination of the different lineages would sum up to be equivalent to RCMD. This contrast did not identify any clear significant gene. However, the result seemed to be quite unstable (typical of analyses with too few samples), with 4 genes at FDR<0.086, and 44 genes at FDR<0.12. Since a more exhaustive analysis would have been required to determine the differences between RCUD and RCDM, it was decided to keep RCUD and RCMD as separate classes. In any case, most of the analyses were done considering the MDS stages by low or high risk, and in this way RCUD and RCMD are grouped together as low-risk subtypes.

Final selection of samples

After the exploration on the clinical and experimental characteristics of the samples, a final set of samples was selected to be included in the study (*Table 11*) which corresponded to the ones that fulfill the following requirements:

1. Samples with a clear/confirmed diagnosis
2. Samples without ringed sideroblasts
3. Samples of normal karyotype
4. Samples isolated in Ficoll corresponding to bone marrow mononuclear cells
5. AML samples with percentage of blast cells ≤50% (only in Series 1+2)
   In Series 3 there are only four samples with normal karyotype and 50% or lower blasts. Therefore, all available AML samples in Series 3 will be used.

	NoL	RCUD	RCMD	RAEB 1	RAEB 2	AML	AML sec
Series 3	6	4	6	5	5	5	5
		10		10		10	
Series 1+2	11	7	17	4	5	10	0
		24		9		10	

Table 11. Samples selected for the study: number of samples from each class in the two series. On the second row, grouped by risk levels

	NoL	**RCUD**	**RCMD**	**RAEB 1**	**RAEB 2**	**AML**	Total
Series 1	0	0	6	3	1	6	16
Series 2	11	7	11	1	4	4	38

Table 12. Number of samples from Series 1 and Series 2 integrated into Series 1+2

	NoL	**RCUD**	**RCMD**	**RAEB 1**	**RAEB 2**	**AML**	**AML sec**	Total
Series 3:								
Female	4	2	2	1	2	1	1	13 (36 %)
Male	2	2	4	4	3	4	4	23 (64 %)
Series 1+2:								
Female	7	5	5	1	2	5	0	25 (46 %)
Male	4	2	12	3	3	5	0	29 (53 %)

Table 13. Number of Male & Female samples per series and class

	RCUD	**NoL** (non-leukemia related cytopenias)
Series 3	2 anemia 2 other/unknown	1 neutropenia and anemia 1 pancytopenia 1 trombocitopenia 3 other/unknown
Series 1+2	4 anemia 3 other/unknown	1 neutropenia and anemia 1 pancytopenia 1 neutropenia 2 thrombocytopenia 2 thrombocytopenia ITP* 4 other/unknown

Table 14. Cytopenia subtypes within the NoL and the RCUD samples

*ITP: Idiopathic thrombocytopenic purpura

2.2. Strategies to find genes that characterize disease subtypes

The first analyses in this study tried to characterize the different subtypes of myelodysplastic syndromes by using the bioinformatic tool developed and presented in Chapter 1: *geNetClassifier*. From the biomedical point of view, low-risk MDS are sometimes difficult to differentiate from other mild hematological diseases –like some of the ones included within the 'no leukemia' controls– that are often detected in the clinic by anemia symptoms. For this reason, it would be interesting to identify a set of genes to be used as makers for diagnosis of the different MDS subtypes, or at least, to identify the main biological processes altered in the initial stages of the disease *versus* the late stages. However, when including all the MDS subtypes –and NoL and AML as reference–, there were too few specific genes assigned as characteristic of each subtype. Therefore, after multiple comparisons and combinations, it was concluded that using 'differential expression profiling' (DEP) was not the right approach. The search for specific genes did not provide enough signal to distinguish the early stages of MDS from other non-malignant mild disorders with similar symptomatology. As some previous classification studies had noticed, it seems that MDS subtypes do not present the well-defined discrete states required for automatic separation and classification (*Rhrissorrakrai et al., 2014*). In this situation we took a different approach: instead of using differential expression profiling to look for specific markers of each subtype of MDS, we use co-expression profiling along the different stages of MDS towards AML in order to find similarities and identify specific patterns for the subtypes of MDS.

Since the main hypothesis of this study was that MDS could develop into leukemia by passing through the different stages, the focus would be to search for these similarities but always considering this *evolutionary* or *disease-progressive* point of view. The first step to achieve this was to set up a comparison of each of the individual MDS diseases and AML *versus* the control samples (NoL). Once the genes affected in each subtype are identified, the genes that are 'activated' or 'lost' at each stage of the progression would be identified by selecting the ones that are also in the upcoming contrasts, i.e.: a gene activated from RAEB 1 would mean that while the gene was not differentially expressed in low-risk MDS (RA, RCUD or RCMD), it was in RAEB 1 and the subsequent steps (RAEB 2 and AML). In this way, the genes and functions that are altered in a continuous evolutionary way in the different stages of the progression of the disease could be identified.

With this new methodological approach, we soon observed that, in fact, the expression of some of the genes tended to have an increasing or decreasing trend when looking at the different stages of the disease. These trends could be the kind of common characteristic linking the different MDS subtypes, therefore they became the focus of this part of the study.

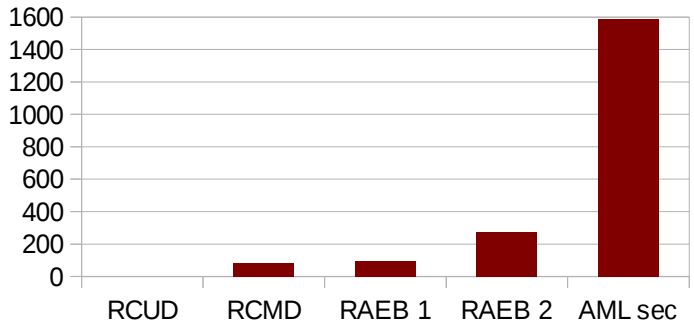


Figure 3. Number of differentially expressed genes in each disease when compared to NoL samples

With these two approaches we noticed that the expression of some genes tended to have an increasing or decreasing trend when looking at the different stages of the disease. These trends could be the kind of common characteristic linking the different MDS subtypes, therefore they became the focus of the study. A summary of the results of this initial analysis with *geNetClassifier* are shown in results *Section 3.2* as illustration of the process that led us to the study of the progression of the gene expression in the MDS stages.

2.3. Genes that correlate with the progression of the disease

Genes associated to the malignancy of the disease

In order to find genes whose expression might be associated to the level of malignancy – being lower at earlier stages (lower risk) and higher at later stages (higher risk), or vice-versa– we searched for genes whose expression evolves following and increasing or decreasing trend in the different stages of MDS (*Figure 4*). To keep the context of the disease progression, in addition to the MDS ordered from a lower to a higher level of malignancy, the no-leukemia samples were used as reference of the origin stage, and the AML samples as reference of the malignant stage they are heading to. Afterwards, we calculated the correlation between the expression of each gene and a variable that represents the level/stage of the disease (*Figure 5*).

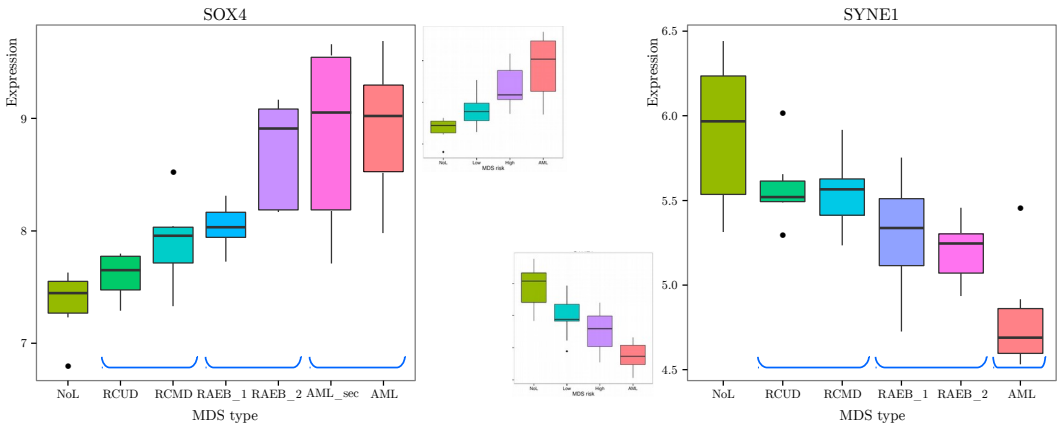


Figure 4. Example of expression patterns associated to the malignancy of the disease. The box-plots represent the expression values (y axis) of the samples grouped by class (x axis) for two different genes: SOX4 and SYNE1. The thumbnails show the same expression data, but grouping the MDS samples in four classes by risk level.

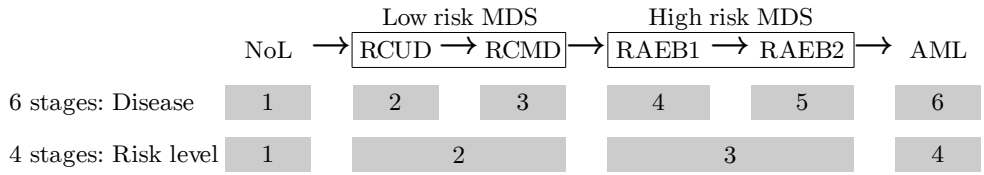


Figure 5. Conceptual design. The expression level was tested to correlate with the 'stages'. Two different subdivision of the samples were used: in 6 stages (named *disease* stages) or in 4 stages (*risk-level* stages).

Gamma correlation

To calculate the correlation between gene expression and the stage variable, which is a categorical/ordinal variable, a rank-based correlation was used. In contrast to other correlations, such as *Pearson's* correlation –which assumes a linear relationship between the variables–, rank-based correlations are non-parametric –they do not assume a specific model or distribution of the variables– since they only measure to which extent there is a monotonic relationship[6] between the two observed variables. The most common rank-based correlation methods are *Spearman's Rho* (*Spearman, 1904, 1907*) and *Kendall's Tau* (*Kendall, 1938*). However, in case there are many ties in any of the variables, it is better to use *Goodman and Kruskal's Gamma* (*Goodman and Kruskal, 1954*). *Gamma* correlation is very similar to *Kruskall's* correlation in regards to calculation, assumptions, and interpretation. Its values also range from -1 (perfect inversion) to 1 (perfect relation), being 0 or very close to 0 if the variables are independent under the given assumptions. In case of rank-based correlations, value 1 is obtained if the rankings of both variables match, and -1 when a ranking is the inverse of the other.

Implementation in R

The Gamma correlation was calculated using the R package Rococo (*Bodenhofer and Krone, 2011*), which provides the implementation of several rank correlation measures, but taking into account some peculiarities of noisy data.

Gamma is calculated based on two variables: The number of cases with the same order in both variables (number of concordant pairs, $\widetilde{C}$), and the number of cases with different order (number of discordant pairs, $\widetilde{D}$). *Gamma* is then calculated as the subtraction of the concordant pairs minus the discordant pairs, divided by the total number of concordant and discordant pairs (*Formula 1*). In this way, tied cases are discarded.

$$\widetilde{\gamma} = \frac{\widetilde{C} - \widetilde{D}}{\widetilde{C} + \widetilde{D}}$$

Formula 1. *Gamma correlation*

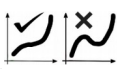
In most methods, the ranking for the variables is constructed in a strict manner: looking for each case whether the value is strictly bigger than the next. Therefore, 1.300000001 is considered bigger than 1.30. Since in case of noisy data, this can distort the results (*Bodenhofer and Klawonn, 2008*), the authors of the package included an additional parameter, *r*, which determines the margin in which both values will be considered equal, and therefore tied in the ranking (*Bodenhofer et al., 2013*).

In this way, the number of concordant and discordant pairs, $\widetilde{C}$ and $\widetilde{D}$, for n pairs of observations (n $\geq$2), $(x_i, y_i)_{i=1}^{n}$, is calculated as:

$$\widetilde{C} = \text{Number of concordant pairs} = \sum_{i=1}^{n} \sum_{i \neq j} \overline{T}\left(R_X(x_i, x_j), R_Y(y_i, y_j)\right)$$

$$\widetilde{D} = \text{Number of discordant pairs} = \sum_{i=1}^{n} \sum_{i \neq j} \overline{T}\left(R_X(x_i, x_j), R_Y(y_j, y_i)\right)$$

[6] Monotonic functions are those that either increase or decrease, but not both.



In a monotonically increasing (or non-decreasing) function: if $x \leq y$ then $f(x) \leq f(y)$

In addition to the parameter r, the package also includes two other arguments, R and $\bar{T}$, to which it offers several alternatives. In our case, we used the default values:

1. 'linear' family of similarities to compare the order of the variables (R):

$$R(x,x') = max(0, min(1, \frac{1}{r}(x - x')))$$

2. 10% of the interquartile range of the variable as the margin that determines if two nearby values are considered equal (r)
3. 'min' as t-norm (triangular function) to determine the aggregation of the ordering measures ($\bar{T}$):

$$\bar{T} = T_M(x,y) = min(x,y)$$

With these parameters, the correlation will be calculated between the gene expression and the categorical value representing the disease stage. This disease stage can be defined using two approaches, since two different subdivision of the samples were defined (*Figure 5*):

a) 6 stage contrast, including a stage for each MDS disease subtype:

NoL – RCUD – RCMD – RAEB 1 – RAEB 2 – AML

b) 4 stage contrast, grouping the MDS samples by risk level (RCUD and RCMD into low-risk, and RAEB 1 and 2 into high-risk):

NoL – Low-Risk – High-Risk – AML

In this way, two correlation values were obtained for each gene, one that represents the correlation of its expression with the disease subtypes (6 stages) and another with the risk level subtypes (4 stages). The results obtained using Series 3 and Series 1+2 will be compared by setting a threshold for *Gamma* and checking the intersection between the different contrasts. From these, two main lists were chosen: the *Full List* of genes correlated in Series 3, and the subset of this list which was confirmed in Series 1+2, named the *Core List* (see results, *Figure 15*).

In order to consider the genes correlated with the MDS stage, the *Gamma* threshold used was 0.50. This number is a standard value used as *Gamma* threshold because, although it is not very restrictive, it indicates that there is an association between the variables. To confirm whether this *Gamma* value is significant in our data, we also took into account the p-value calculated for the *Gamma* correlations. To avoid doing prior assumptions regarding the data or the statistic's distributions, *Rococo* package uses permutation testing for estimating these p-values: the distribution for the null hypothesis (H_0) is estimated by calculating the rank correlation coefficient for X shuffles of the data (and thus, simulating independence between the variables). The p-value is then calculated as the relative frequency –percentage of times– that the shuffled test exceeded the absolute value of the real unshuffled data (in case of a two-sided test).

In our analyses, the p-value was calculated for a distribution with 1000 permutations (the default value), which was enough to test whether the association was significant at 95% while keeping a reasonable execution time. Moreover, the p-values were further adjusted for multiple-testing for the total number of genes in the chip (20,172 for *hgu133plus2* and 38,408 for *HuEx1.0*) using False Discovery Rate (FDR, *Benjamini and Hochberg, 1995*), implemented in *p.adjust* function in R.

To explore further the significance of the 0.50 *Gamma* threshold, the p-value estimation was also done with 100,000 shuffles for a few random genes in each dataset. *Figure 6* shows a representative example of the distribution obtained for these 100,000 random shuffles for a gene in Series 3. Out of these shuffles, only 14 were over 0.50 or under -0.50, which would correspond to an empirical p-value of 0.00014. In the case of Series 1+2, this empirical p-value was about 0.00001. The absolute values of *Gamma* that would correspond to the extreme 5% (2.5% each

side) were 0.27 in Series 3 and 0.23 in Series 1+2, which made us quite confident that the 0.50 *Gamma* threshold set up in these analyses would indeed imply that there is relationship between the gene coexpression profile and the disease stage.

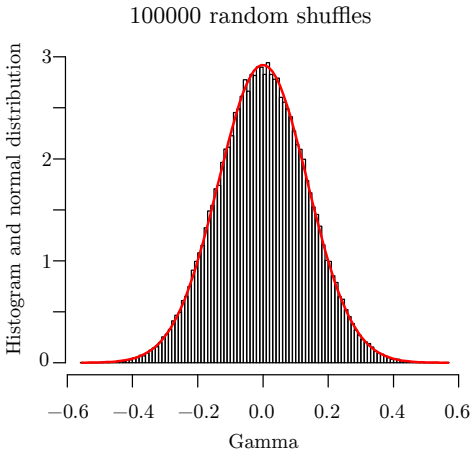


Figure 6. Distribution of the 100,000 *Gamma* values obtained for the shuffled expression of a gene from Series 3.

All these studies were done using all AML samples, ignoring whether they are secondary to MDS or not. The secondary AML samples from Series 3 were used to study the possible difference with primary AML, and explore whether MDS samples had more genes in common to secondary AML. However, the analyses done just with secondary AML did not seem to provide much difference. Although the *SAM* between the secondary AML samples and the likely-primary AML samples identified 584 differentially expressed genes (0.05 FDR), it might be due to the lower percentage of blast cells in the secondary AML samples. Using only the secondary AML samples with the *Gamma* correlation approach, the results were very similar to the ones obtained using all AML samples: 843 genes in the 6-stage contrast, and 1291 genes in the 4-stage contrast, confirming 210 genes out of the 266 genes in the *Core List* (see results Section 3.3). Therefore, all AML samples were used for the analyses, but when exploring the results (i.e. expression box-plots of selected genes) in Series 3, AML and secondary AML were normally kept separate.

2.4. Expression patterns associated to the stages of the disease

Within the genes identified as correlated to the disease stage genes, it was observed that there were genes whose expression changed mostly in the last stage: i.e. in the transition to the most severe MDS (RAEB 2) to AML. By contrast, in other cases, the expression change was more linear, or occurring mostly during MDS stages (*Figure 7*).

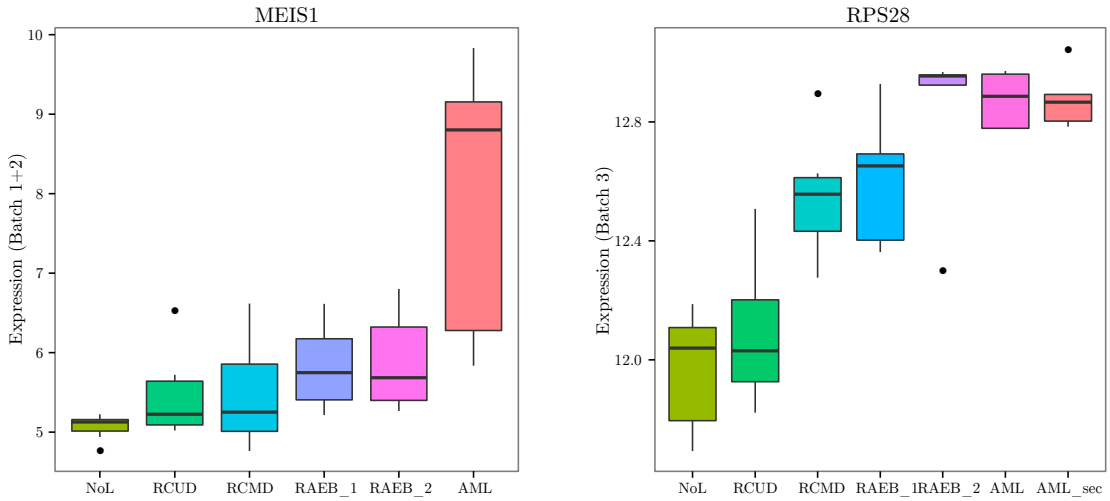


Figure 7. Example of two different patterns or behaviors detected within the correlated genes

In order to find out the groups of genes following these patterns, we tried an approach based on the percentage of the global change that took place within MDS stages and the transition to AML. This approach allowed to confirm that there were big groups of genes showing such transition to AML. However, this approach used an arbitrary threshold. A proper analysis should be done, using an unsupervised method, to find common repetitive patterns within the expression profiles of the genes along the disease stages. For this reason, we used a *Self Organizing Map* (*SOM, Kohonen, 1982*), a robust unsupervised clustering and dimensionality reduction method that allows to search for common expression patterns among groups of genes.

To perform the clustering, the expression of all genes was normalized by subtracting its mean and dividing by the standard deviation. In this way, the expression of all the genes was within the same scale. Then, for each gene, the expression values were sorted in ascending or descending order (ascending if the mean expression in AML samples was higher than in NoL, and descending otherwise). Although this reordering switched the position of the samples, most AML samples were still kept on the right, NoL on the left and MDS samples in the middle (*Figure 8* and figures in results Section 3.4). With this normalized and sorted gene expression data, the genes were clustered with the *SOM* implementation from *Kohonen* package (*Wehrens and Buydens, 2007*) using a 3x3 grid with rectangular topology to allocate up to 9 possible classes.

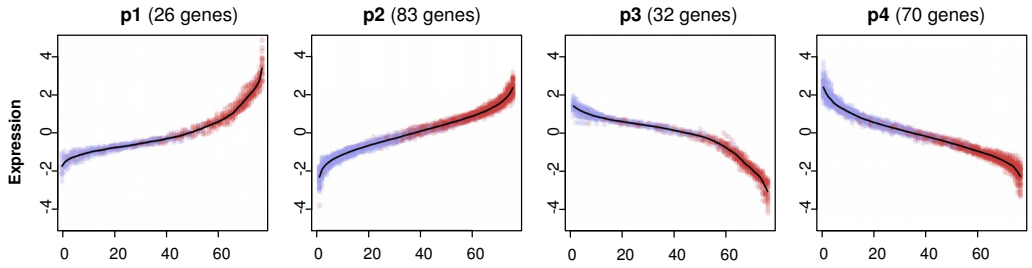


Figure 8. Gene expression of the NoL samples (blue) and AML samples (red) in the reordered dataset

After applying this work-flow to the *Core List* of correlated genes, we obtained similar clusterings in both datasets. Within the nine (3x3) groups provided by the clusterings there were four major patterns including a significant number of genes (*Figure 9*): two of the patterns, p1 and p3, clearly had a bigger increase or decrease in expression in the samples on the far right, while the patterns in the opposite corners, p2 and p4, had a more linear behavior (or even a slight stronger change in the location where the NoL to low-risk MDS transition occurs). The three groups in the middle row were not relevant since they had not been assigned any genes.

The four patterns found were in clear agreement with the trends that we had previously observed for some specific genes (Figure 7): the 'largest change towards AML' or the 'largest change towards MDS'. This was also confirmed by calculating the $\delta/2$ in each pattern (the point that locates the 50% change in the expression range).

In order to obtain the list of genes that consistently fit each of the patterns, the gene needs to be assigned to the same pattern in both datasets, or to one of the main patterns in one of the datasets and to the related intermediate pattern in the other dataset.

To split the *Full List* of 1,163 genes correlated in Series 3, the same approach was followed but taking into account only the pattern assigned in Series 3. Since the *Full List* was obtained through the dataset in *HuEx1.0*, not all the genes are available in *hgu133plus2* to do the validation prior to the pattern assignment (see results Section 3.4).

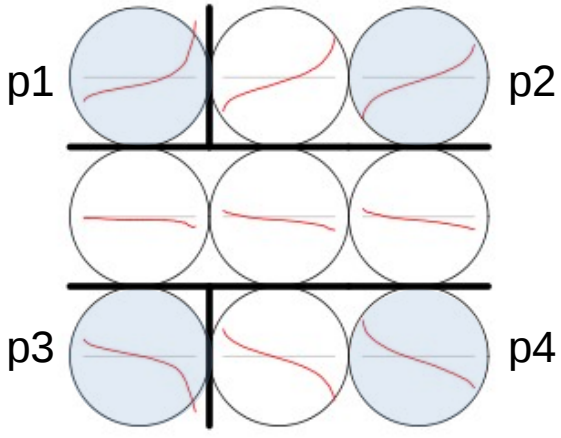


Figure 9. Expression patterns found through *SOM*. Highlighted in blue the 4 patterns selected as representative of the profiles with 'largest change towards AML' (p1 and p3) and 'largest change towards MDS' (p2 and p4).

2.5. Correlation with the percentage of blast cells in the bone marrow

One of the defining characteristics of MDS as a malignant disease is the presence of blasts in the bone marrow and peripheral blood. Since the percentage of blasts in the bone marrow is the main diagnostic criteria for the MDS subtypes in this study, it also follows the kind of increasing/decreasing trend detected by the *Gamma* correlation (*Figure 10*).

In order to further explore the possible relationship between the expression patterns that we found and the percentage of blast cells, we decided to look at the correlation between the expression signal and such percentage.

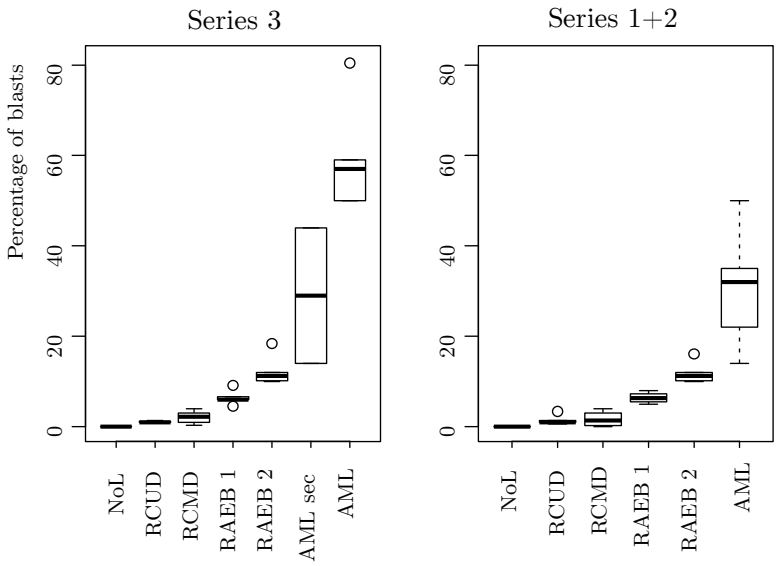


Figure 10. Boxplot of the percentage of blast cells in the bone marrow per disease

Pearson correlation was calculated for each gene and the blasts percentage using the subset of selected samples for which the blast percentage was known (*Table 15*). All no-leukemia samples were assumed to have 0% blasts. The concordance between the correlation with blasts in the different datasets was checked through the intersections of the correlated genes (absolute correlation over 0.50, and FDR adjusted p-value lower than 0.05).

	NoL	RCUD	RCMD	RAEB 1	RAEB 2	AML	Total
Series 3	6	3	6	5	5	5 + 2 sec.	32
Series 1+2	11	6	17	4	5	10	53

Table 15. Samples used for calculating the correlation with blasts percentage

2.6. Functional enrichment analysis of the genes included in the patterns

The initial *Functional Enrichment Analysis* (*FEA*) of the gene lists was done using the package *FGNet* with *GeneTerm Linker*. These initial analyses allowed to confirm that the two main lists –the *Core List* of 266 genes and the *Full List* of 1,163 genes–, and the patterns within them, were indeed associated to functions altered in MDS. However, the clustering of these results complicated the integration and comparison of the output obtained for the different lists.

In order to identify the altered biological processes that were unique or more representative of each pattern, a standard singular *FEA* approach –without additional filtering or clustering– was used. For each of the gene lists, an independent analysis was performed through *RDAVIDWebService* using *DAVID's Functional Annotation Chart* with the main annotations related to pathways and biological processes (GO-BP, KEGG, REACTOME, PANTHER and BIOCARTA). These analyses resulted in ten tables containing the enriched terms (threshold 0.50) for each gene list: the two main lists (the *Core List* and the *Full List*) and the four patterns within each. The output of each main list was merged with the output from its patterns into a single table containing the whole list of terms that appeared in any of the analyses, and the p-value of the contrast in which it appeared (i.e. the pattern). To ease the analysis, the genes annotated to each term were also split into up- and down-regulated and the terminal GO terms were labeled as *leaves* using *plotGoAncestors* from *FGNet*.

The functional network built following this strategy and the most enriched terms within these analyses are shown in results Section 3.6.

2.7. Validation with independent external datasets

As validation for the results obtained in the study, the same approach was applied to two independent expression datasets from MDS samples: a dataset from pediatric MDS that uses the same type of cells as done in our study (bone marrow mononuclear cells) (*Bresolin et al., 2012*) and another dataset that uses only CD34+ bone marrow cells (*Pellagatti et al., 2006*). In this way, although both datasets contain equivalent MDS subtypes to the ones in our study, they also have important differences. Pediatric MDS may show some different characteristics to adult MDS; and CD34+ cells correspond to a small fraction of the bone marrow cells (i.e. an average of 1-2% –range 0.5-5%–).

With these datasets, the *Gamma* correlation between expression along MDS classes of different risk level was calculated in the same way as with the data in our study (adjusting to the available classes as shown in *Table 16*). This provided a list of genes with absolute value of *Gamma* over 0.50 and p-value$<$0.05 for each dataset. Then it was checked which of the genes in the *Core List* were confirmed with these datasets (intersection requiring the same sign of correlation). In the same way, the *FEA* of the lists of genes correlated in these datasets was carried on and compared to the *FEA* of the *Core List*. Since these datasets were missing either NoL and AML, it was not possible to apply the SOM-based method to identify the patterns.

Dataset from Bresolin *et al.*: Pediatric MDS

Availability: GEO GSE29326 (*Bresolin et al., 2012*)

Tissue and cell type: Bone marrow mononuclear cells

Microarray platform: *Affymetrix* GeneChip Human Genome U133 Plus 2.0 Array

Available MDS subtypes/stages:

1. Refractory cytopenia of childhood (RCC)
2. Refractory anemia with excess of blasts (RAEB)
3. RAEB in transformation (RAEB-t)

Main differences with our data:

- It does not include the reference (NoL) state.
- Karyotype:
  - Normal karyotype: 15 patients
  - Monosomy of chromosome 7 or del(7q): 10 patients (5 RCC, 2 RAEB and 3 RAEB-t)
  - Trisomy of chromosome 8: 1 patient (RCC)

  It is not specified which samples are in each group, therefore it is not possible to select only the samples of normal karyotype.
- Class division in pediatric MDS is established at different blasts percentages. In particular, patients with 20-30% of blasts might be classified as MDS instead of AML depending on the remaining morphological characteristics.
- The low-risk MDS subtype is RCC (there is no separation between RAEB 1 and RAEB 2, therefore it is only possible to do the contrasts by risk level).

Dataset from Pellagatti *et al.*: Bone marrow CD34+ cells

Availability: GEO GSE19429 (*Pellagatti et al., 2010*)

Tissue and cell type: Bone marrow CD34+ cells

Microarray platform: *Affymetrix* GeneChip Human Genome U133 Plus 2.0 Array

Available MDS subtypes/stages:

1. Control
2. Refractory anemia (RA)
3. Refractory anemia with excess of blasts 1 (RAEB 1)
4. Refractory anemia with excess of blasts 2 (RAEB 2)

   All samples are normal karyotype

Main differences with our data:

- Does not include AML (the destination stage will be missing)
- It uses a previous WHO classification which includes RCUD and RCMD as refractory anemia (RA)

dataset	NOL	RCUD+RCMD	RAEB 1	RAEB 2	AML
Series 3	6	10 (4+6)	5	5	10 (5+5)
Series 1+2	11	24 (7+17)	4	5	10

	-	RCC	RAEB	RAEB-t
Pediatric	-	11	5	4

	Control	RA	RAEB 1	RAEB 2	-
CD34+	17	21	21	20	-

Table 16. MDS stages available in each dataset

2.8. Gene regulation: Transcription factors associated to the gene patterns

Since the groups of genes with a similar increasing or decreasing trend could have common regulators, and it had been noticed that there were many transcription factors (TFs) within the gene lists, we decided to explore the possible relationship between both. This could be done by looking for enriched transcription factor binding motifs (TFBM) in the promoters of the genes in the different lists. Several methods were initially explored (i.e. *DIRE*, *Opossum*). However, the method finally chosen was *iRegulon* (*Janky et al., 2014*), a Cytoscape plugin that includes a method to detect enriched transcription factor motifs and their direct targets.

Enriched transcription factor motifs –and their probable regulators– were identified for the genes in the four main patterns from the *Core List* and the protein coding genes within the *Full List* of 1,163 genes (splitting them into the 575 up-regulated and 352 down-regulated). For each list, the enriched motifs were searched in the close promoter region (*up to 500bp upstream)* and also at a much wider region (*±10kbp from the translation start site*). Within the transcription factors with enriched motifs, the focus would be on those included within the *Full List*.

3. Results

3.1. Data pre-processing steps: Integration of Series 1 and Series 2

Frozen RMA and *Combat* –through *inSilicoMerging*– were used to merge Series 1 and 2 into a single dataset. *Figure 11* shows the density curves after each pre-processing step. For comparison, the *batch effect* when using only *RMA* is also shown. This batch effect is clearly reduced by normalizing with *Frozen RMA*, and the remaining effect is completely removed after applying *COMBAT*.

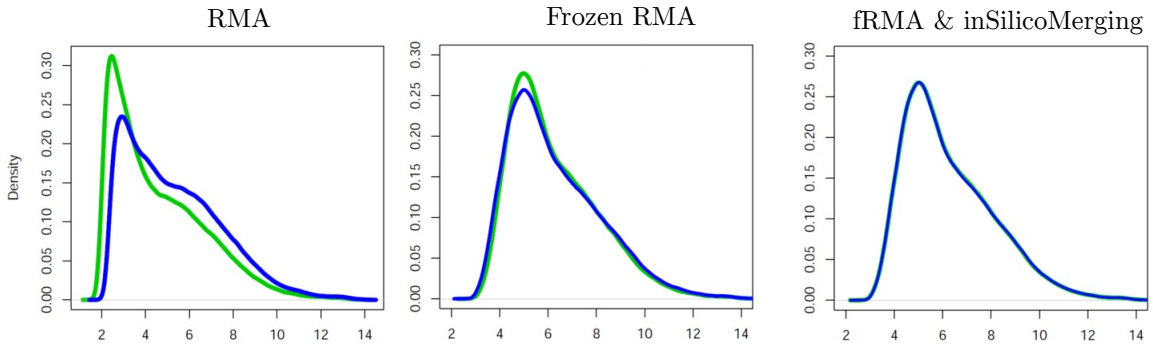


Figure 11. Density curves of Series 1 (blue) and Series 2 (green) after pre-processing with different methods. Preprocessed independently with *RMA* (left); Preprocessed with *Frozen RMA* (center); Preprocessed with *Frozen RMA* and merged with *InSilicoMerging* (right)

The merged datasets were checked through the *multidimensional scaling* (unsupervised clustering) provided by *inSilicoMerging* (*Figure 12 - right*) and a standard complete linkage clustering using as distance *1-correlation* (*Figure 12 - left*). None of them suggested that there was any obvious batch effect (the samples from Series 1 and Series 2 seem to be shuffled rather than grouped by the original series).

To make sure that the shuffling in the clustering was not provided by the disease subtypes, which could be hiding the batch effect, the samples between series within the same type were analyzed using *SAM* to find differences. There were no significant differences when comparing samples from Series 1 vs samples from Series 2 for leukemia or any of the MDS subtypes.

Merging Series 1 and Series 2 through *fRMA* and *COMBAT*/*inSilicoMerging* efficiently removed the batch effect between both datasets. Therefore, they will be used as a single dataset: *Series 1+2*.

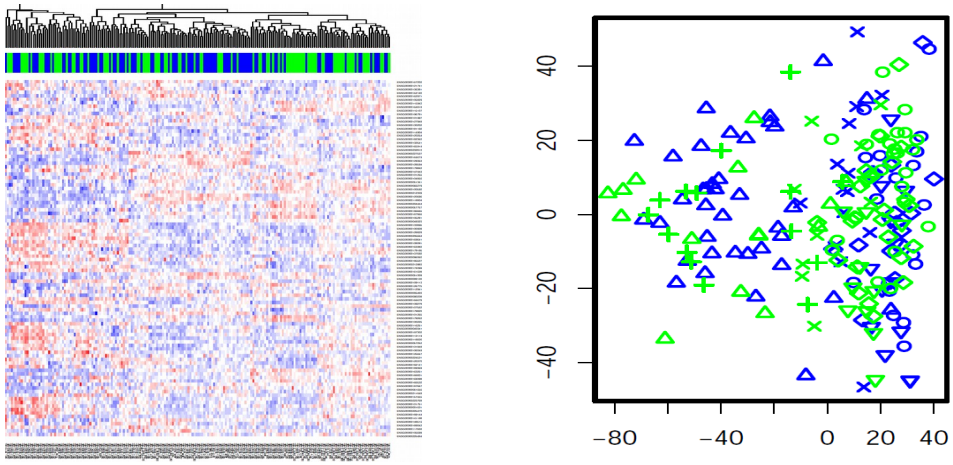


Figure 12. Non supervised clusterings of the merged Series 1+2. In blue samples originally from Series 1 (batch 1), in green from Series 2 (batch 2). Right: Clustering using *1-correlation* as distance. Left: *Multidimensional scaling plot* from *inSilicoMerging* (dot shapes indicate the sample type)

3.2. Genes that characterize each disease: use of *geNetClassifier*

An overview of different contrasts using *geNetClassifier* is shown in *Table 17* and *Figure 13*, and the protein coding genes associated to each MDS subtype are available in *Table 18*.

As it can be observed, the number of genes found for the extreme classes (NoL, the control; and AML, the most malignant disease) are relatively high, indicating that they are clearly different from any other class. By contrast, very few genes are assigned to the intermediate states that correspond to MDS subtypes. In addition these genes do not segregate the MDS classes efficiently, indicating that they are not good for characterizing the disease.

By disease subtype:

	NoL	RCUD	RCMD	RAEB1	RAEB2	AML sec	AML
Series 3	512	62	21	43	25	209	810
Series 3 (prot.cod)*	225	23	9	12	17	129	847
Series 3 (gr.AML)	534	58	22	52	31	695	
Series 1+2	234	16	80	21	23	-	1408

By risk level:

	NoL	Low risk	High risk	AML_sec	AML
Series 3 *	516	40	27	209	799
Series 3 (gr.AML)	543	30	30	693	
Series 1+2	240	118	37	-	1373

Table 17. Number of genes with posterior probability over 0.95 for different contrasts

(*gr.AML*: grouping AML_sec and AML, *prot cod*: only protein coding genes)

* Plot for the contrast by risk in *Figure 13*, protein coding genes assigned to each disease in *Table 18*

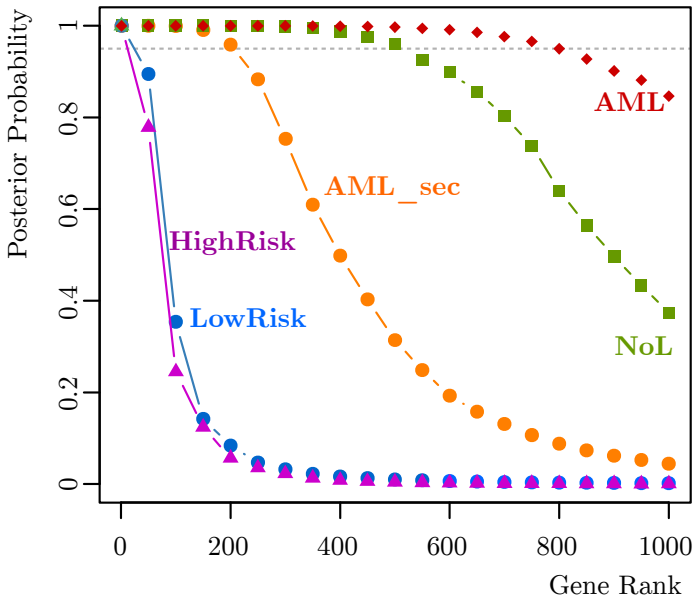


Figure 13. Posterior probability plot for the genes identified in Series 3 using the grouping *by risk level* as classes.

Ranking	NoL	RCUD	RCMD	RAEB_1	RAEB_2	AML_sec	AML
1	AC008964.1	CCL7	HBZ	CDR1	PROM1	CDH2	LTF
2	CD3G	AC130365.2	INPP4B	CLEC1B	DNTT	DDIT4L	CRISP3
3	TARM1	AL137145.1	DSG2	OR8G5	MACC1	CT45A5	CEACAM6
4	HIST2H2BF	OR6N2	AC022532.2	TMEM78	PTH2R	NT5E	CEACAM8
5	GZMK	AL953854.1	BAMBI	AP002512.2	FABP4	CLEC9A	TCN1
6	KCNJ15	AC136443.2	LRRC2	AC124319.3	CP	SCGB3A2	MMP8
7	AC008948.1	CCIN	SAMHD1	NBPF16	SMC1B	LOX	AP003970.1
8	ARHGEF5L	OCM2	KRTAP2-4	MMP1	AC008088.2	ABCA6	MMP9
9	DDX3Y	PGA3	TNFSF10	RP11-568G20.1	LAX1	SLCO1B3	CD24L4
10	OR6K3	DOCK4	DYNLL1P1	PF4	FNBP1L	CCNA1	BPI
11	RPS4Y1	AL049872.1	TLR7	PDE5A	SYCP2	RFPL4A	SLPI
12	FAR2	CCL4	MERTK	OR2M5	CA8	MAMDC2	HP
13	KLRF1	AL355273.1	C20orf175	OR2D2	FMO2	AC008749.3	ARG1
14	AKAP5	SDHD	OR2T33	AL365229.1	HLA-DQA1	CD36	ANXA3
15	NRIP3	OR1N1	SPINK8	AC134878.1	KCNK1	RHAG	CLEC4D
16	PLK2	PROKR2	C9orf150	KHDC1L	AL138968.2	SPTA1	IL8RA
17	JUN	OCLN	ZNF675	AC010518.1	BEX1	ZNF521	GJB6
18	ORM2	ARG2	GBP4	AC005296.1	IFNG	CES1	FPR2
19	DDIT3	C4orf34	CXCL10	OR13C4	HIST1H4D	SPON1	FPR1
20	CD3E	CD300E	KIT	AC091565.4	LEPR	ANKRD20A1	CHI3L1
21	EIF1AY	CCR1	ZNF737	CR381653.1	HIST1H1T	GYPA	GPR97
22	KBTBD7	AC092139.2	TXK	GNG11	BBS7	IL3RA	PLBD1
23	NELL2	SDS	PYHIN1	SELP	AC104758.3	RPS13	OLFM4
24	GZMA	MAP2K6	AC010329.1	PROS1	C6orf192	CAPRIN2	LCN2
25	FGFBP2	CMTM1	OPN1MW2	OR2T5	PRG4	CD34	DEFA4
26	MANSC1	DZIP3	DEFB133	AC007204.1	PRG2	DNAJA4	PGLYRP1
27	CD96	C13orf18	TLR6	AC006994.1	RELN	IFI6	GLT1D1
28	CXCL9	AC111170.3	RHD	FN1	C7orf53	HBG1	VSTM1
29	AL645728.1	AC132872.5	IFNA10	KRT5	HSPH1	AC007956.1	OLR1
30	GCNT4	CPA3	P2RY14	AL589743.1	CENPI	RPS3A	AD000685.1
Total:	225	23	9	12	17	129	847

Table 18. List of genes assigned using *geNetClassifier* to each one of the 7 classes (defined in Table 1 for Series 3 using only protein coding genes). The table shows the top genes in the ranking.

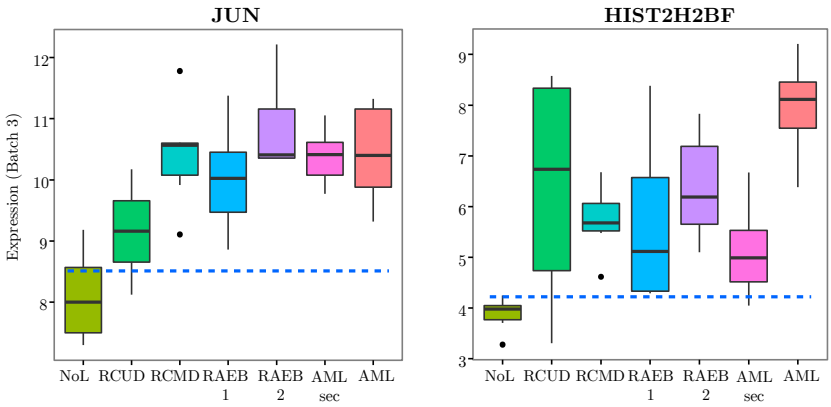


Figure 14. Expression profile in the different stages for some genes selected for NoL: Jun and an histone H2B.

3.3. Genes that correlate with the progression of the disease

The genes that correlate with the progression of the disease based on the risk level include mostly all the genes that correlate with the *MDS subtypes* (*Table 19* and *Figure 15*). Therefore, the correlation with *risk level* was selected as the main contrast for the study. This contrast provides two gene lists:

1. **The *Full List* of genes correlated in Series 3 (1,163 genes)**: The union of the genes that appear in the contrast with 4 stages (grouping by risk level) and the contrast with 6 stages (using all MDS subtypes). The contrast including all MDS subtypes only adds 15 genes to the contrast grouping by risk. Therefore, the genes correlated in Series 3 are mostly the genes that correlate with the risk level. This list can be subset to obtain the 804 genes that also correlate with the MDS subtype.
2. **The *Core List* (266 genes)**: The subset of the genes correlated in Series 3, which is also confirmed in Series 1+2. This list is the intersection of genes correlated with risk level in Series 1+2 and Series 3 (263 genes, p-value<0.00001). However, since there were only three missing genes from the contrast with all MDS subtypes (UFC1, SCD5 and TMED1), they were also included. In this way, this list makes a total of 266 genes (263g ∪ 167g = 266 g).

Both lists are available in the supplementary file *GeneLists_corr_with_disease.xls*. This file also includes the results from the following sections: patterns and correlation with percentage of blasts cells.

	4 stages (Risk level)	**6 stages** (MDS subtype)	Union (4st ∪ 6st)	Intersection (4st ∩ 6st)
Series 3	1,148	804	**1,163**	789
Series 1+2	663	400	671	392
Intersection (Series 3 ∩ 1+2)	**263**	167	269	

Table 19. Number of correlated genes in each contrast and the corresponding union and intersections (Absolute *Gamma* > 0.50 and p < 0.05)

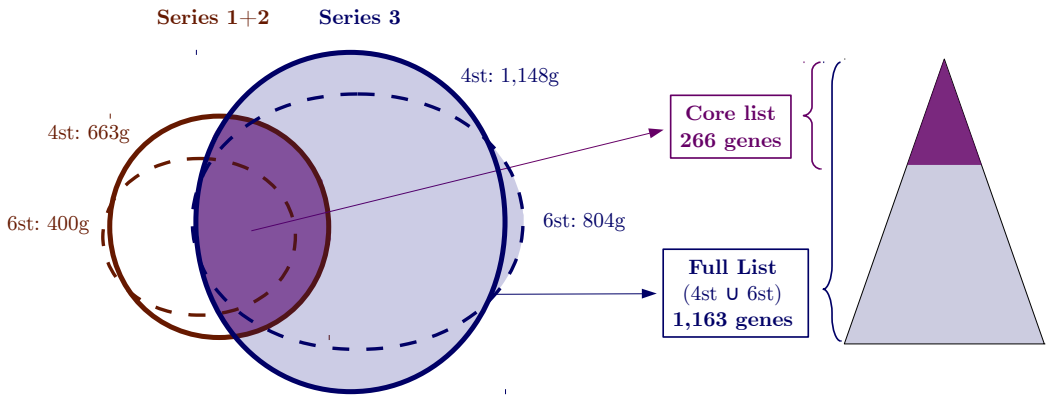


Figure 15. Scheme illustrating the genes included in the *Core List* and the *Full List*.

3.4. Expression patterns associated to the stages of the disease

The 266 genes from the *Core List* were split into four patterns according to the *SOM* clustering (Additional files *Patterns.pdf* and *GeneLists_Patterns.xls*). Pattern 1 and Pattern 3 include 58 genes in which the biggest expression change is clearly in the late stages ($\delta/2$ in AML), and Pattern 2 and 4 include the 153 genes with the biggest change during the MDS stages ($\delta/2$ in MDS). The first pattern of each group contains the genes with an increasing trend, and the second pattern the genes with decreasing trend. As a way to remember the trends, each pattern was labeled according to the location of the $\delta/2$ and the direction of the change: **Pattern 1** (**p1**) was labeled 'AML_up', **Pattern 3** (**p3**) 'AML_dw', **Pattern 2** (**p2**) 'MDS_up' and **Pattern 4** (**p4**) 'MDS_dw'. Note that out of the 266 genes in the *Core List*, only 211 genes were assigned to a pattern. The remaining genes either had intermediate profiles or the assigned pattern did not coincide in the *SOM* clustering with Series 3 and with Series 1+2.

The same approach was followed to calculate the patterns within the *Full List* of 1,163 genes correlated with malignancy in Series 3. However, the assignment to the patterns was done based only on the data in Series 3. Since Series 3 is based on the microarray platform *HuEx1.0,* many of the genes in the *Full List* are not available in the array *hgu133plus2*. Therefore, it was not possible to calculate the patterns for the *Full List* using Series 1+2 to confirm whether they match the pattern assigned in Series 3. Instead, we checked the patterns assigned to the genes in the *Core List* –since it uses both datasets– in the *Full List* –that uses only Series 3– (*Table 20*). Out of the genes that were assigned to a pattern in both clusterings, only 4 genes do not match. These are four genes assigned to **Pattern 4** ('MDS_dw') in the *Core List*, that were assigned to **Pattern 3** ('AML_dw') in the clustering of the 1,163 genes from Series 3. The pattern to which each gene was assigned is added as a column in the additional file *GeneLists_corr_with_disease.xls.*

		Pattern in Full List						
		p1	**p2**	**p3**	**p4**	intermDw	intermUp	Not Assigned
List Pattern in Core	**p1** (AML_up)	**23**	0	0	0	0	2	1
	p2 (MDS_up)	0	**63**	0	0	0	17	3
	p3 (AML_dw)	0	0	**32**	0	0	0	0
	p4 (MDS_dw)	0	0	4	**39**	25	0	2
	Not Assigned	4	3	22	0	3	22	1

Table 20. Genes of the patterns assigned by *SOM* from the *Full List* overlapping with the genes of the patterns assigned to the *Core List*. The maximum overlapping possible is 211 and the overlap observed is 74.4 % (considering only the 4 defined patterns and not the genes that show intermediate profiles).

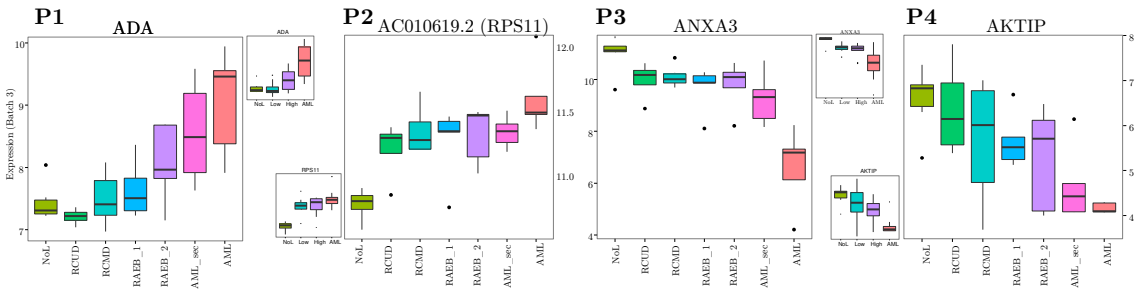


Figure 16. Boxplot of the expression of four genes – one from each pattern

3.5. Correlation with the percentage of blast cells

The *Pearson* correlation (r) between gene expression and percentage of blasts was calculated for every gene in Series 3 and Series 1+2. *Figure 17* shows the distribution of these values. In Series 3 there are 3,477 genes with absolute correlation (R)>0.50 and adjusted p-value<0.05. Out of these, 489 genes follow the same trend as the blasts (showing a positive correlation) and, therefore, are overexpressed in MDS/AML compared to NoL; and 255 genes follow the inverse direction (negative correlation). All these 744 genes (489 positive and 255 negative) are confirmed in Series 1+2, having the same sign of correlation and included within the 1,373 genes (905 positive and 468 negative) that correlate with the percentage of blast cells in Series 1+2.

For the genes in the *Full List* and the *Core List*, the correlation of the gene expression with the percentage of blasts is added as a column in the additional file (*GeneLists_corr_with_disease.xls*). Table 21 shows the 48 genes that have correlations over 0.70 –in absolute value– in both lists. This intersection is mostly limited by Series 1+2 (with only 41 genes correlated positive and 62 negative), rather than Series 3 (223 genes positive and 119 negative).

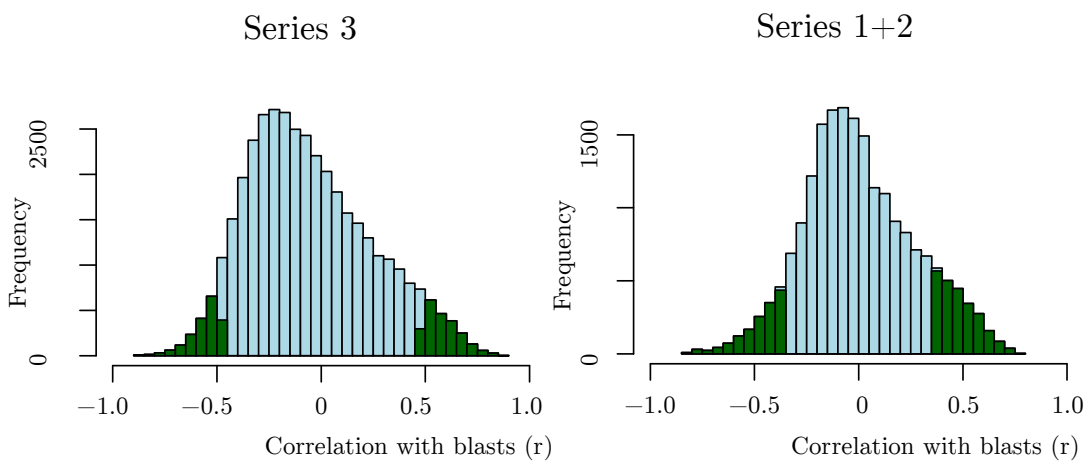


Figure 17. Histogram of the correlation (r) between gene expression and the percentage of blasts for the genes in Series 3 and Series 1+2. Green bars indicate adjusted p-value (FDR) < 0.05

The correlation between gene expression and blast percentage was calculated mainly to explore the possible relationship between the patterns and the blast percentage. The patterns **p1** and **p3** changing mainly in the late stages were confirmed to have a strong association with the blasts percentage: 88% of the 25 genes in **p1**, and 97% of the genes in **p3** are within the 744 genes that are correlated with blasts in both datasets. This is also confirmed when the correlation of all genes in the patterns is compared. The correlation of expression with blast percentage in **pattern p1** and **p3** is clearly different from the average correlation in patterns 2 and 4 (Wilcoxon test: $5{\cdot}10^{-07}$ p-value between **p1** and **p2** from the *Core List*, and p-value$<3{\cdot}10^{-14}$ between **p3** and **p4**, *Figure 18*). A high correlation between the percentage of blast cells and the genes in the patterns characteristic of AML (i.e. **p1** 'AML_up' and **p3** 'AML_dw') would be expected and does not provide relevant news about our current knowledge about MDS as pre-leukemia, but at the same time this correlation provides strong support to the existence of the patterns discovered and brings about the interest for the meaning and value of the other patterns.

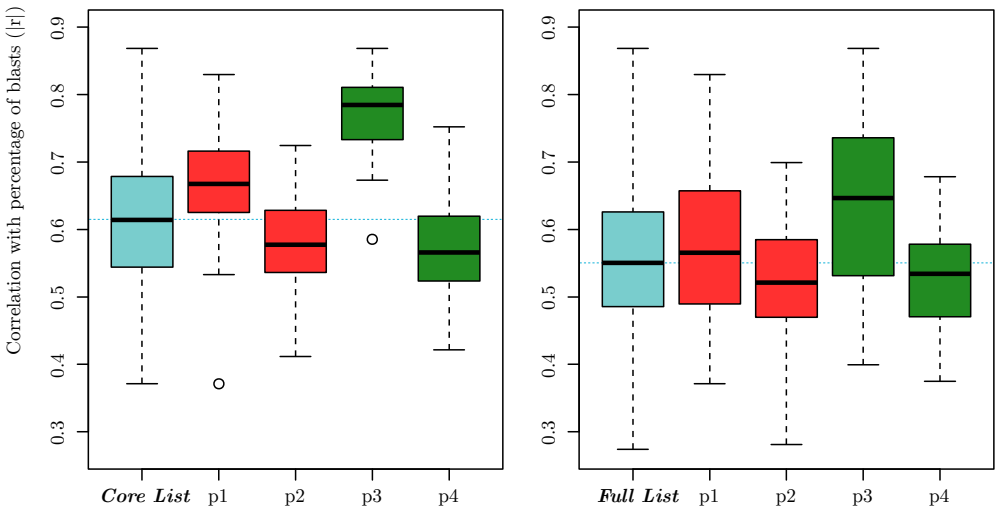


Figure 18. Correlation between the percentage of blasts and the expression for the genes, in each pattern (green and red) and the whole list (blue). **Left**: *Core List* (266g), **right**: *Full List* (1,163g). For each gene, the value is the average correlation from both datasets (if they have the same sign of correlation).

Gene	Pattern	*r* (Series 3)	*r* (Series 1+2)	Gene description
ANGPT1	p1	0.75	0.75	angiopoietin 1
DFFA	-	0.72	0.72	DNA fragmentation factor, 45kDa, alpha polypeptide
FLT3	p1	0.87	0.79	fms-related tyrosine kinase 3
GMDS	-	0.72	0.74	GDP-mannose 4,6-dehydratase
ITPR1	p1	0.87	0.70	inositol 1,4,5-triphosphate receptor, type 1
KAT2A	-	0.81	0.74	K(lysine) acetyltransferase 2A
LYRM4	-	0.73	0.75	LYR motif containing 4
SCARB1	-	0.74	0.72	scavenger receptor class B, member 1
SLC7A6	-	0.72	0.71	solute carrier family 7 (cationic amino acid transp...)
STAR	p1	0.70	0.71	steroidogenic acute regulatory protein
TMTC4	p1	0.75	0.71	transmembrane and tetratricopeptide repeat cont. 4
TSPAN3	-	0.71	0.72	tetraspanin 3
ANXA3	p3	-0.80	-0.78	annexin A3
ARG1	p3	-0.87	-0.77	arginase, liver
CAMP	p3	-0.86	-0.76	cathelicidin antimicrobial peptide
CD24L4	p3	-0.83	-0.76	CD24 molecule
CEACAM6	p3	-0.8	-0.70	carcinoembryonic antigen-related cell adhesion mol.6
CEACAM8	p3	-0.85	-0.79	carcinoembryonic antigen-related cell adhesion mol.8
CHI3L1	p3	-0.76	-0.79	chitinase 3-like 1 (cartilage glycoprotein-39)
CKAP4	p3	-0.78	-0.84	cytoskeleton-associated protein 4
CLEC4D	p3	-0.76	-0.82	C-type lectin domain family 4, member D
CRISP3	p3	**-0.86**	**-0.81**	cysteine-rich secretory protein 3
CXCL1	-	-0.77	-0.75	chemokine (C-X-C motif) ligand 1 (melanoma...)
GCA	p3	-0.76	-0.74	grancalcin, EF-hand calcium binding protein
GGTA1	-	-0.78	-0.72	glycoprotein, alpha-galactosyltransferase 1
GJB6	-	-0.71	-0.77	gap junction protein, beta 6, 30kDa
GLT1D1	p3	-0.83	-0.78	glycosyltransferase 1 domain containing 1
GNG2	p3	-0.73	-0.72	guanine nucleotide binding protein (G protein), ...
GPR160	-	-0.72	-0.71	G protein-coupled receptor 160
HP	p3	-0.74	-0.73	haptoglobin
IL18RAP	-	-0.78	-0.75	interleukin 18 receptor accessory protein
LCN2	p3	-0.83	-0.78	lipocalin 2
LTF	p3	**-0.89**	**-0.85**	lactotransferrin
MGAM	p3	-0.79	-0.75	maltase-glucoamylase (alpha-glucosidase)
MMP25	-	-0.75	-0.75	matrix metallopeptidase 25
MMP8	p3	-0.80	-0.75	matrix metallopeptidase 8 (neutrophil collagenase)
MMP9	p3	**-0.87**	**-0.81**	matrix metallopeptidase 9 (gelatinase B, 92kDa...)
PADI4	p3	-0.74	-0.70	peptidyl arginine deiminase, type IV
PGLYRP1	p3	-0.76	-0.72	peptidoglycan recognition protein 1
QPCT	p3	-0.78	-0.82	glutaminyl-peptide cyclotransferase
S100A12	p3	**-0.83**	**-0.83**	S100 calcium binding protein A12
S100A8	-	-0.78	-0.79	S100 calcium binding protein A8
S100P	-	-0.81	-0.76	S100 calcium binding protein P
SLAIN1	p3	-0.73	-0.72	SLAIN motif family, member 1
SLPI	-	-0.73	-0.81	secretory leukocyte peptidase inhibitor
STOM	p3	-0.88	-0.76	stomatin
TCN1	p3	-0.83	-0.80	transcobalamin I (vitamin B12 binding protein, ...)
WIPI1	p3	-0.79	-0.76	WD repeat domain, phosphoinositide interacting 1

Table 21. Genes which highly correlate with the percentage of blast cells in both datasets ($|r|>0.70$). In bold, the four genes with $|r|> 0.80$ in both datasets. Pattern in the *Full List,* if available.

3.6. Functional enrichment analysis of the genes included in the patterns

The whole table containing the functional enrichment analysis (*FEA*) of the *Full List* of 1,163 genes correlated with malignancy is provided as additional file (*fea_ cor1163gTable.xls*). This table also contains the enrichment of the GO-CC and GO-MF in addition to the pathways according to KEGG and Reactome. For each gene list (i.e. pattern) the terms with p-value<0.10 are highlighted.

As summary, *Table 22* shows the most significant terms in each pattern within the *Full List*, and *Table 23* other functions enriched in the *Full List* but not assigned to any specific pattern (both tables include terms enriched with p-value < 0.05).

The functional enrichment of the patterns from the *Core List* are shown in the functional networks in *Figure 19* (MDS-patterns **p2** and **p4**) and *Figure 20* (AML-patterns **p1** and **p3**). These networks show the terms enriched for the genes in each pattern with p-value<0.10, grouping the closest/overlapping gene-term sets with the *clustersDistance* function from *FGNet* package (for example the 6 terms related to ribosome in **pattern 2**). For GO terms, only the terminal terms (leaves) are shown. In **pattern 1** and **3** there is not any enriched pathway at p-value<0.10, therefore, all terms in the network are GO-BP.

This functional enrichment analysis provides several intriguing results. According to this analysis, *apoptosis* is clearly affected –apoptosis is one of the main functions altered in MDS, and it is mainly lost in the transition to AML (*Raza et al., 1995*)–. However, appearing only on the global list (*Table 23*) the automatic functional analysis does not provide a clear indication regarding in which pattern it is altered the most, or whether it is increased or decreased. In the same way, when looking at the initial gene lists, it was observed that **pattern 1** had several pro-proliferation genes associated to AML (FLT3, ANGPT1 and HOXA-genes). However, *cell proliferation* (GO:0008283) and *cell growth* (GO:0016049) appear as enriched in **pattern 2** (and in **pattern 4** with p-value < 0.08), not related to these known AML genes. *Differentiation* is also in a similar situation. There are several down-regulated genes known to be associated to differentiation in **pattern 3** (LTF, MMP 8 and 9, CAMP, CRISP and CEACAM-), but most terms related to differentiation seem to be associated to either the whole list or **pattern 1**. This is probably a good illustration on how functional enrichment analyses might not match the interpretation of the gene lists by human experts. While *FEA* can be useful and provide extra knowledge, it should always be taken into account that it is an automated approach limited by the knowledge and annotations included in the databases.

On the other hand, there are some recurrent functions significantly associated to **pattern 2** (MDS_up). These include terms related to *nucleosome assembly* (histones), *ribosome* and *spliceosome*, which are functions known to be altered in MDS. For example, ribosomal biogenesis was initially discovered altered in MDS with 5q-, but later linked to p53 stress response and apoptosis (*Raza and Galili, 2012*). In this way, although the relevance of these functions still needs to be assessed, they are a good validation for the method.

Another interesting observation was the profile of the histones in the different stages. Although only a few specific histone genes appear within the list of genes with increasing patterns in all stages, histones had often appeared in the contrasts with *geNetClassifier*. Looking deeper into them, all histones tend to be over expressed in MDS: *wilcoxon-test* between the mean expression of all the histone genes from each type in NoL *versus* low-risk MDS is significant for all histone types in both series, especially for RCMD. However, many of them do not follow the increasing pattern in AML. In this way, they are a good example of a specific group of genes that is clearly altered in MDS, but not on the upcoming AML stage (*Figure 21*).

P1 (AML_up)	**P2** (MDS_up)
Anterior/posterior pattern formation, Skeletal system development & morphogenesis (GO:0009952, GO:0001501, GO:0048705, GO:0048704)	Aminoacyl-tRNA biosynthesis (hsa00970), TRNA aminoacylation for protein translation (GO:0006418), TRNA processing (GO:0008033),
Blood vessel morphogenesis (GO:0048514)	Cell growth (GO:0016049)
Definitive hemopoiesis (GO:0060216)	Cell proliferation (GO:0008283)
Glossopharyngeal nerve morphogenesis (GO:0021615)	De novo purine biosynthesis (P02738)
Myeloid progenitor cell differentiation (GO:0002318)	FGF signaling pathway (P00021)
Notch signaling pathway (GO:0007219)	Metabolism of nucleotides (REACT_1698)
Response to hypoxia (GO:0001666)	Nucleobase, nucleoside and nucleotide biosynthetic process (GO:0034404), Nitrogen compound biosynthetic process (GO:0044271)
	Nucleosome assembly (GO:0006334)
	Ribosome (hsa03010), 3' -UTR-mediated translational regulation (REACT_1762), Gene Expression (REACT_71), Translational elongation (GO:0006414), ...
	Spliceosome (hsa03040)

P3 (AML_dw)	**P4** (MDS_dw)
Carbon dioxide transport (GO:0015670)	ATP biosynthetic process (GO:0006754)
Cytoskeleton organization (GO:0007010)	Cerebellar Purkinje cell differentiation (GO:0021702)
Defense response to bacterium (GO:0042742)	Cell proliferation (GO:0008283)
Hematopoietic cell lineage (hsa04640)	Dopamine receptor mediated signaling pathway (P05912)
Inflammatory response (GO:0006954), Respiratory burst (GO:0045730)	Extracellular matrix organization (GO:0030198)
Leukocyte adhesion (GO:0007159)	Purine nucleotide/ribonucleotide biosynthetic process (GO:0006164, GO:0009152), Nucleobase, nucleoside and nucleotide biosynthetic process (GO:0034404), Nitrogen compound biosynthetic process (GO:0044271)
Plasma membrane repair (GO:0001778)	Response to toxin (GO:0009636)
Polysaccharide catabolic process (GO:0000272)	Signaling in Immune system (REACT_6900)
Regulation of cell adhesion mediated by integrin (GO:0033628)	

Table 22. Terms enriched with p-value<0.05 in each pattern from the *Full List*

Terms	n	Genes (up & down)
Induction of programmed cell death (GO:0012502); Positive regulation of apoptosis (GO:0043065)	31	BAK1,CD24L4, CDK5R1, CEBPG, DAPK2, DFFA, DYRK2, FGD4, MAPK1, MMP9, MUL1, NLRC4, NME3, PDCD7, PLAGL2, PMAIP1, PPP2R1A, PSEN2, PTEN, RP11-142G7.1, RPS6, RXRA, SH3RF1, SORT1, SOX4, SQSTM1, TNFRSF10B, TRIO, UTP11L, WWOX, ZMAT3
Lymphocyte differentiation (GO:0030098); B cell differentiation (GO:0030183); B cell activation (GO:0042113)	18	ADA, ADAM17,ATP7A, BAK1, BST2, CBFB, CD24L4, CD24L4, CD40LG, CEBPG, CTNNB1, FLT3, IGBP1, KLF6, LAT2, MINK1, RPL22, SOX4
Purine metabolism (hsa00230)	16	ADA, AMPD3, APRT, ENTPD1, GMPR2, GUCY1A3, IMPDH2, ITPA, NME3, NT5C, PDE7B, PKLR, POLR2J, POLR3D, POLR3K, PPAT
Response to oxidative stress (GO:0006979)	15	ADA, ATP7A, CA3, GCLM, IPCEF1, OLR1, PDLIM1, PEBP1, PPP2CB, PRDX2, PRNP, PXN, RP11-142G7.1, STAR, TPM1
Alzheimer disease-presenilin pathway (P00004)	12	ADAM17, CTNNB1,GSK3A, LRP10, LRP5L, MMP25, MMP27, MMP8, MMP9, PSEN2, PVRL1, RNF152
Aminoglycan metabolic process (GO:0006022)	8	B4GALT7,CHI3L1, CHIT1, CHST11, CSGALNACT1, EXT2, MAMDC2, PGLYRP1
Purine ribonucleoside biosynthetic process (GO:0046129); Nucleobase metabolic process (GO:0009112)	8	ADA, AMPD3, APRT, CDA, MAPK1, PPAT, QTRT1, QTRTD1
Heme biosynthesis (P02746)	4	COX10,FECH, QARS, RP11-319G5.1
Nuclear receptors coordinate the activities of chromatin remodeling complexes and coactivators to facilitate initiation of transcription in carcinoma cells (h_rarrxrPathway)	4	GTF2F1,KAT2B, NCOA2, RXRA

Table 23. Other functions enriched with p-value<0.05 in the *Full List* but not assigned to any specific pattern

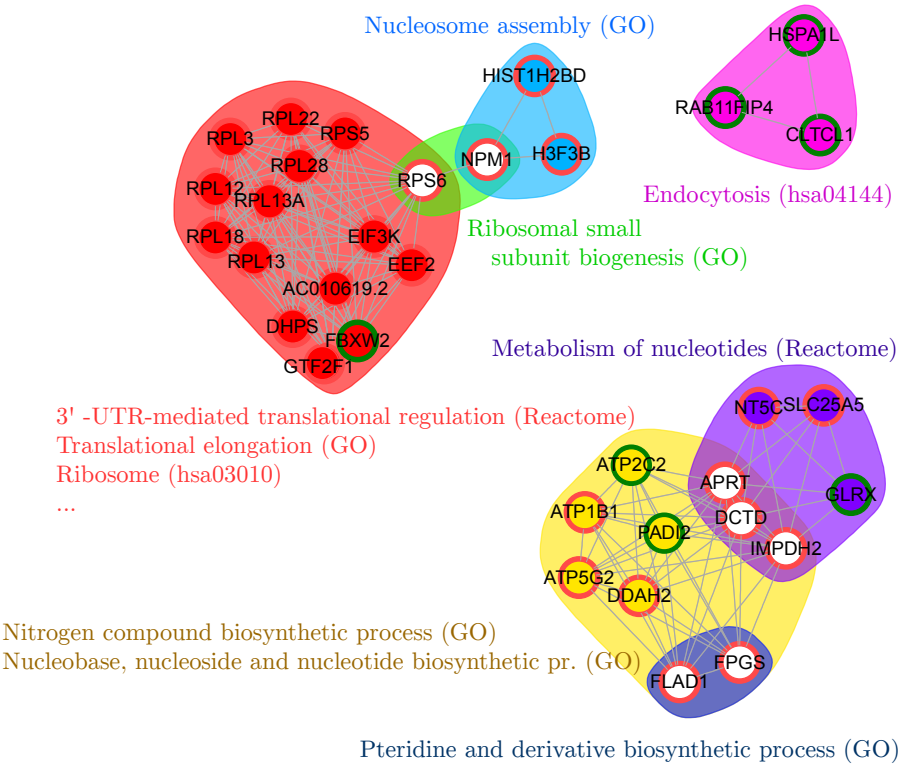


Figure 19. Most enriched pathways and biological processes in **pattern 2** and **4** from the *Core List* (the network includes the 27 genes from **p2** and the 7 genes from **p4** annotated to these terms)

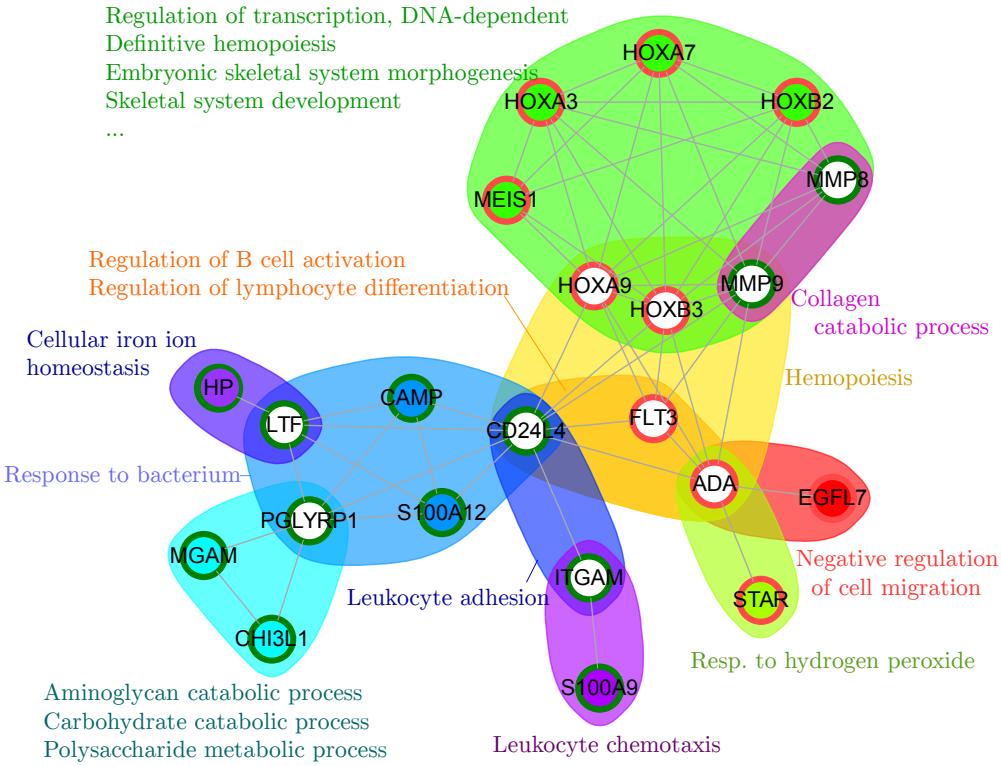


Figure 20. Functional network of the most enriched biological processes (GO) in **pattern 1** and **3** from the *Core List* (the network includes the 10 genes from **p1** and the 12 genes from **p3** annotated to these terms)

	p1 (AML_up)	p2 (MDS_up)	p3 (AML_dw)	p4 (MDS_dw)	Full list	Total
Core List	27	14	11	4	34	65
Full List	31	49	18	31	73	157

Table 24. Number of terms enriched at p-val<0.05 in each of the contrasts

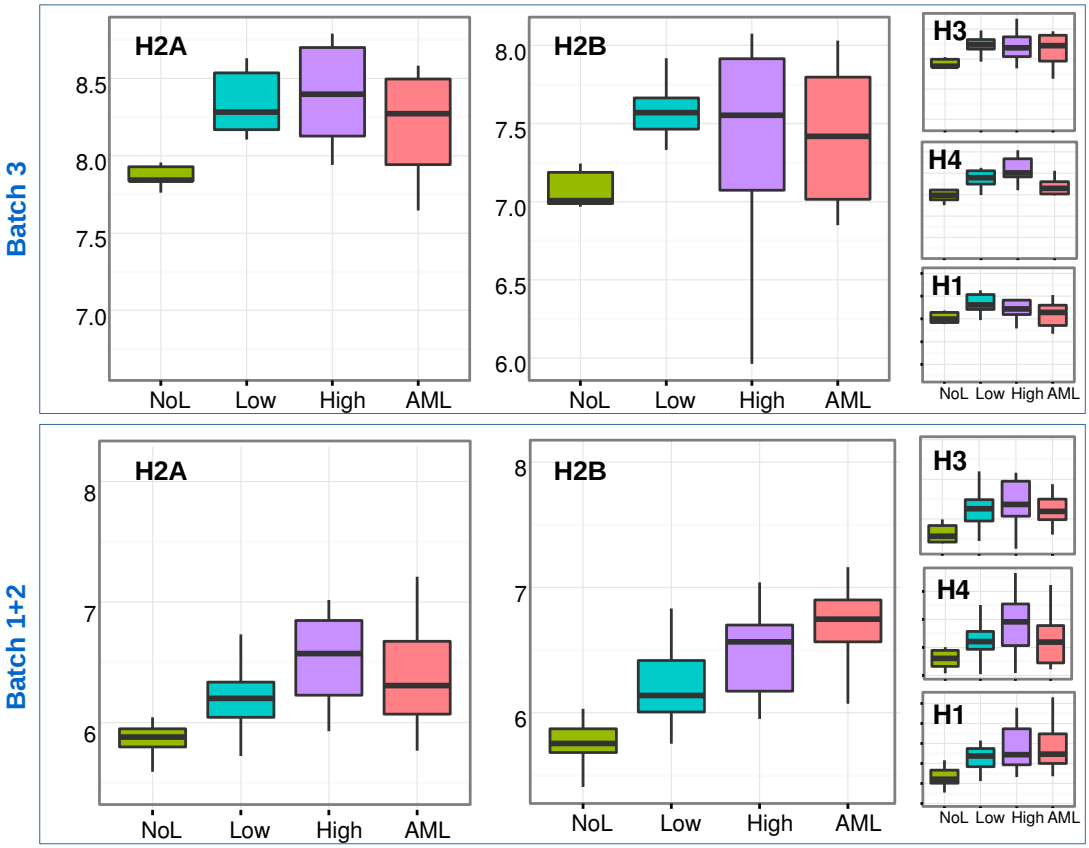


Figure 21. Boxplot of the expression of the histones by disease stage (mean expression of all the genes of the histone type in each sample)

3.7. Validation with independent external datasets

Comparison of gene lists

In Pellagatti *et al.* dataset, 566 genes correlate with the progression of the disease ($|Gamma|>0.05$ and adjusted p-value<0.05). Out of these, 23 genes are within the *Core List* (empirical p-value$<10^{-5}$).

In Bresolin *et al.* dataset, out of the 2,495 genes with *Gamma* over 0.50 and p-value $<$ 0.05, 120 genes are within the 266 genes in the *Core List* (empirical p-value$<10^{-5}$). However, while in the other datasets all genes with $|Gamma|>0.05$ kept their significance after adjusting by FDR, only 450 of the 2,495 genes have a FDR adjusted p-value $<$ 0.05, which reduces the intersection with the *Core List* to 42 genes.

In both external datasets, most of the confirmed genes (i.e. the 23 and the 42 overlapping genes) correspond to the Pattern 1 (AML_up). Moreover, this pattern presents a strong agreement with a MDS gene signature already published and reported in the work by Bresolin *et al.* named *AML-like signature*. This signature detected in MDS samples included 11 genes, and out of them, 6 are found within our Pattern 1: ANGPT1, FLT3, HOXA (at least two genes of this cluster), HOXB3 and MEIS1.

Comparison of enriched terms

Out of the 65 significantly enriched terms in the *Core List*, 37% are annotated to the 566 genes correlated in Pellagatti dataset, and 51 (78%) to the 2,495 genes correlated in Bresolin dataset (taking into account all terms annotated to the list even if they are not significant). Of these, 24 terms (36%) are significantly enriched (enrichment p-value<0.05) in the most restrictive list of 450 genes significant after FDR adjustment in Bresolin dataset.

Within the confirmed terms are the functions related to ribosome (i.e. translational elongation, GO:0006414) and HOXA (i.e. embryonic skeletal system morphogenesis, GO:0048704). The main functions that do not appear in the genes correlated in Bresolin dataset are shown in *Table 25*. Although the functions are not enriched, many of these genes are actually correlated in Bresolin dataset (ADA, ATP2C2, ATP5G2, CHI3L1, DDAH2, IMPDH2, MAPK1, MMP9, and PGLYRP1).

Term	Genes in *Core List*
Polysaccharide catabolic process (GO:0000272)	CHI3L1, CHIT1, MGAM, PGLYRP1 (Down)
Cytosine metabolic process (GO:0019858)	CDA, MAPK1 (Down)
Collagen catabolic process (GO:0030574)	MMP27, MMP8, MMP9 (Down)
Cellular amino acid derivative biosynthetic process (GO:0042398)	FPGS, GAMT (Up); PADI2, PADI4 (Down)
Nitrogen compound biosynthetic process (GO:0044271)	ADA, APRT, ATP1B1, ATP5G2, DCTD, DDAH2, FLAD1, FPGS, IMPDH2 (Up); ARG1, ATP2C2, CDA, PADI2, PADI4 (Down)

Table 25. Functions significantly enriched in the *Core List* (GO leaves) that do not appear as enriched for the 2,495 genes correlated in Bresolin dataset

3.8. Gene regulation: Transcription factors associated to the gene patterns

We looked for enriched transcription factor binding motifs (TFBM) near the genes in each of the patterns from the C*ore List* and of the protein coding genes in the *Full List* (*up to 500 bp* upstream *and ±10k bp* from the gene transcription start site (TSS)). This search provided a total set of 660 transcription factors (TFs) with enriched binding motifs near the TSS of the genes in any of the lists (*Table 26*, additional file *TF_ summary.xls*). From these TFs, about half of them –i.e. 321 genes– were present in at least three or more of the contrasts done. Moreover, from the total set of TFs identified there were 42 which were also TFs within the *Full List* of 1,163 genes (*Table 27*). We focused on exploring these 42 TFs.

Figure 22 shows the genes in **patterns 1** and **2** from the *Core List* potentially up-regulated by TFs from the *Full List*. Interestingly, many of the 84 genes from **pattern 2** (MDS_up) seem to be regulated by only 7 TFs. By contrast, the 25 genes in **pattern 1** (AML_up) have motifs for many more TFs, some of them within the list, which leads to a very connected network. *Figure 23* shows an equivalent network, illustrating how just 4 up-regulated transcription factors (ATF2, CDK2AP1, ETS2 and TAF7) could regulate many of the 575 up-regulated protein coding genes identified as correlated with malignancy in Series 3. Out of these 4 TFs –according to the contrasts in which they appear– TAF7 and ATF2 seem to be the most specific for MDS. TAF7 has the main expression change during the early stages in the disease (boxplot in *Figure 24*), and it has 22 ribosome-related genes as potential targets. On the other hand, ATF2 (activating transcription factor 2, also known as CREB2) is a moonlighting protein that functions as TF but also as histone acetyltransferase (HAT), which together with CDK2AP1 (subunit of the nucleosome remodeling and histone deacetylation) could be associated to the alterations of histones and the effects of HDAC-inhibitors in MDS. ATF2 has also been involved in regulation of apoptosis, cell growth, and DNA damage response; and ETS2 in development and apoptosis.

Another interesting TF found is CEBPG. Although it is over-expressed in MDS and leukemia, it has motifs in all of the down-regulated gene lists (133 down-regulated targets), suggesting it could be a repressor. CEBPG has been reported to interact with CEBPA –one of the commonly mutated genes in AML–, and being highly up-regulated in patients with hyper-methylated CEBPA (*Alberich-Jordà et al., 2012*). According to the methylation data from Chapter 3, CEBPG could also be hypomethylated in RA (0.10 p-value).

In addition, some TFs with enriched motifs in **pattern 2** (MDS_up) are genes often mutated in MDS. Since point mutations are sometimes associated with specific clinical features and survival, we checked in the *Catalog of Somatic Mutations in Cancer* (COSMIC, *Forbes et al., 2014*) which of these TFs appeared within the 80 genes with common mutations in MDS. RUNX1 (mutated in 0.089% of the tested samples), TP53 (0,095%), EP300 (0,051%) and ETV6 (0,007%) had enriched motifs in **pattern 2**. The somatic point mutations in RUNX1, TP53 and ETV6 are known as predictors of poor overall survival independently of other established risk factors (*Bejar et al., 2011*).

	P1 *(AML_up)* *25g*	**P2** *(MDS_up)* *84g*	**Protein coding UP** *575g*	**P3** *(AML_dw)* *32 g*	**P4** *(MDS_dw)* *70g*	**Protein coding DOWN** *352g*
500bp upstream	176	88	67	147	171	82
±10kbp from TSS	190	185	111	244	148	118

Table 26. Number of TFs with enriched motifs in the patterns from the *Core List*, and in the protein coding genes from the *Full List* (splitted into UP-regulated and DOWN-regulated)

	TF	Pattern	FC.R	p1	p2	ProtCod UP	p3	p4	ProtCod DW
1	ATF2	p1	1,06		500, 20k	20k	20k		
2	CBFB	p2	1,06	500, 20k	20k		500, 20k	20k	500
3	ETS2	p2	1,11		500, 20k	500, 20k	500, 20k	20k	
4	NR1H3		1,06		20k		20k		
5	TAF7	p2	1,12		500, 20k	500, 20k			
6	TGIF2	p2	1,23		20k		20k		
7	XBP1		1,09	20k	500		20k		
8	CEBPG		1,17	500			500, 20k	500, 20k	500, 20k
9	GTF2F1	p2 *	1,09	500					
10	HESX1	p1	1,18	20k				500	
11	HOXA10	p1	1,09	20k					
12	HOXA3	p1 *	1,09	500, 20k					
13	HOXA7	p1 *	1,14	500				500	
14	HOXA9	p1 *	1,17	20k					
15	HOXB2	p1 *	1,21	20k				500	
16	HOXB3	p1 *	1,09	500, 20k				500	
17	KDM4A		1,06	20k			500		
18	KLF6		1,13	20k					500, 20k
19	MEIS1		1,1	20k			500		500, 20k
20	SOX4		1,21	500				20k	20k
21	CDK2AP1	p2	1,13			20k			
22	E2F6		1,12					500	
23	GALK1		1,05						20k
24	ID1	p1	1,26					20k	
25	RBM8A		1,06				500		500
26	RRN3	p2	1,1				20k		
1	GATA1		-1,1	20k		500, 20k	500, 20k	20k	500, 20k
2	GFI1B		-1,15	20k				20k	
3	LTF	p3 *	-1,07					500	
4	NFIA	p3	-1,19					20k	
5	RCOR1		-1,13				500	500	
6	RXRA	p3	-1,06	500	20k		500, 20k	20k	
7	FOXJ2		-1,09		20k		20k		
8	KLF3		-1,11				500		500, 20k
9	E2F4		-1,05						500
10	E2F7		-1,14						500
11	KLF7		-1,12	20k					500, 20k
12	SOX6		-1,48						20k
13	TFDP1		-1,11						500
14	FOXM1	p3	-1,09	500					
15	JAZF1		-1,21	500, 20k					
16	YOD1	p3	-1,14	500					

Table 27. Transcription factors from the *Full List* (rows) which appeared in any of the TFBM enrichment analyses (columns). The cell content indicates in which analysis it appeared: 500 bp upstream or 20k bp centered in TSS (within ±10k bp from TSS). Next to the TF, the pattern and fold change in the *Full List*/Series 3 (the asterisk indicates it was also in the same pattern in the *Core List*). The top half of the table shows the TFs that are up-regulated in MDS/Leukemia and the bottom half the ones down-regulated. The TFs that seem to be unique to a specific contrast (and have the same expression direction) are highlighted with the color of the corresponding contrast.

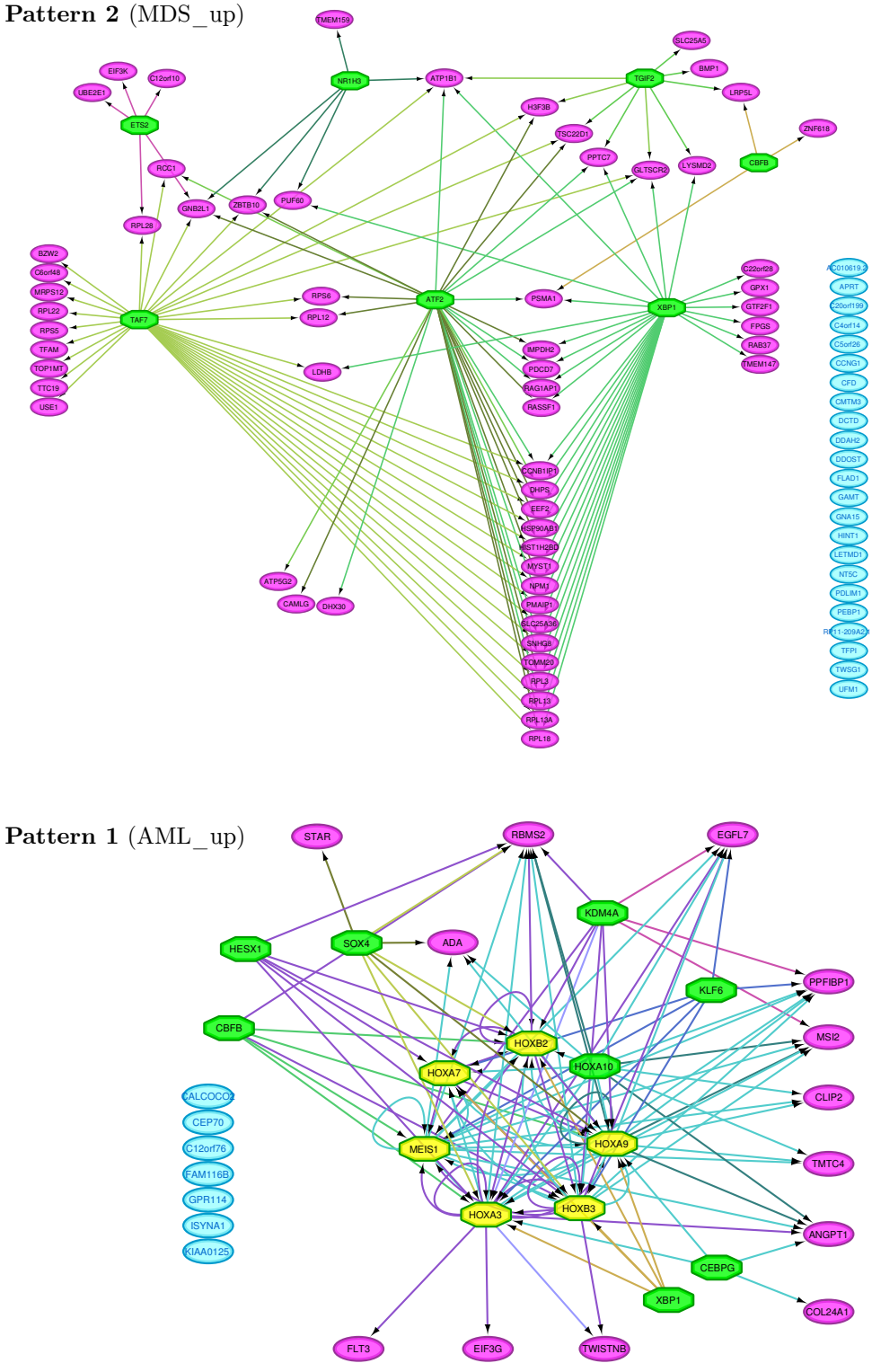


Figure 22. Regulation network of genes in patterns 1 and 2 from the *Core List* (pink and blue nodes), and the up-regulated transcription factors from the *Full List* (green) (TFs with enriched motifs up to 500 bp upstream and ±10k bp of the TSS of the genes in the pattern). Yellow indicates the transcription factors that belong to the pattern.

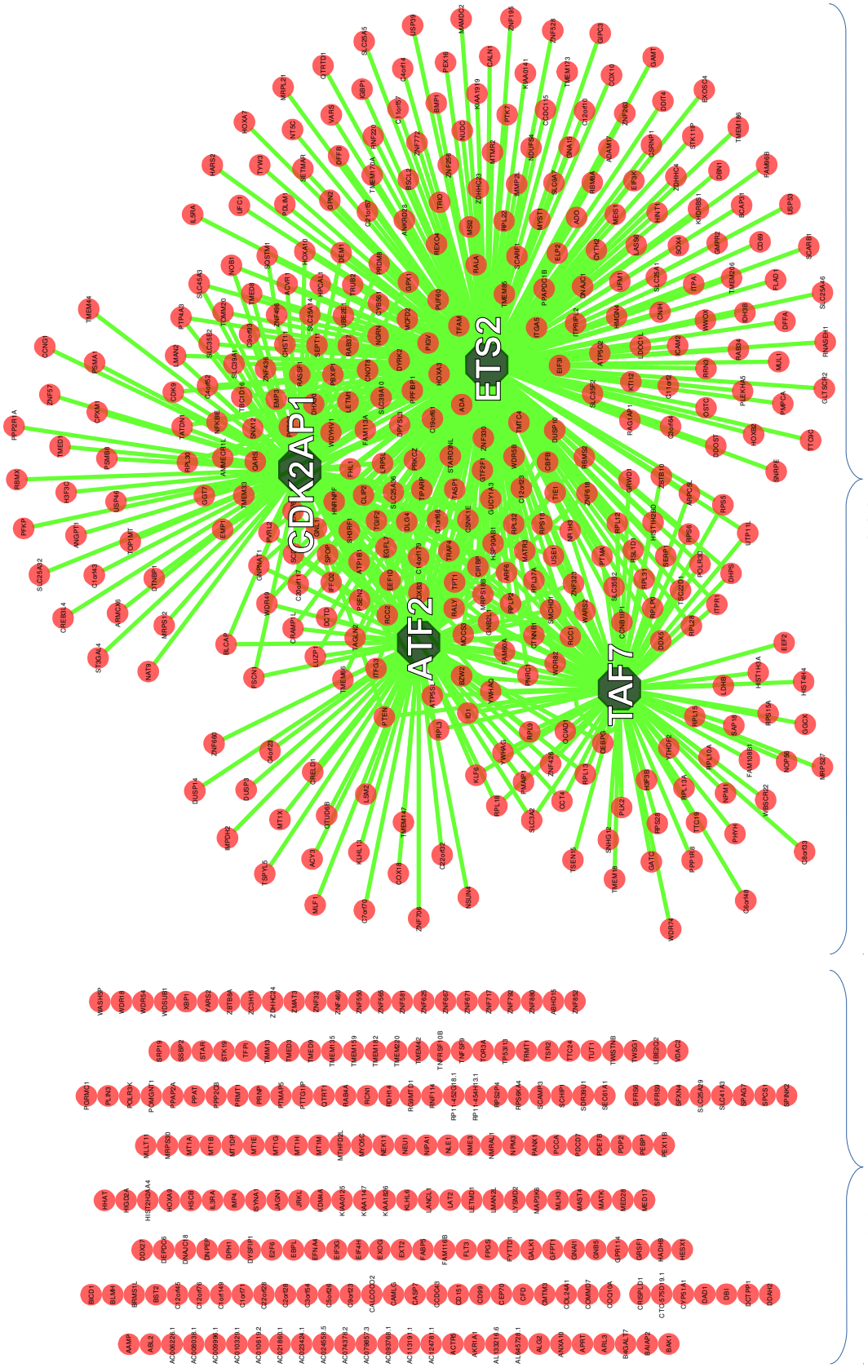


Figure 23. Up-regulated protein coding genes from the *Full List* and the TFs with enriched motifs within the list

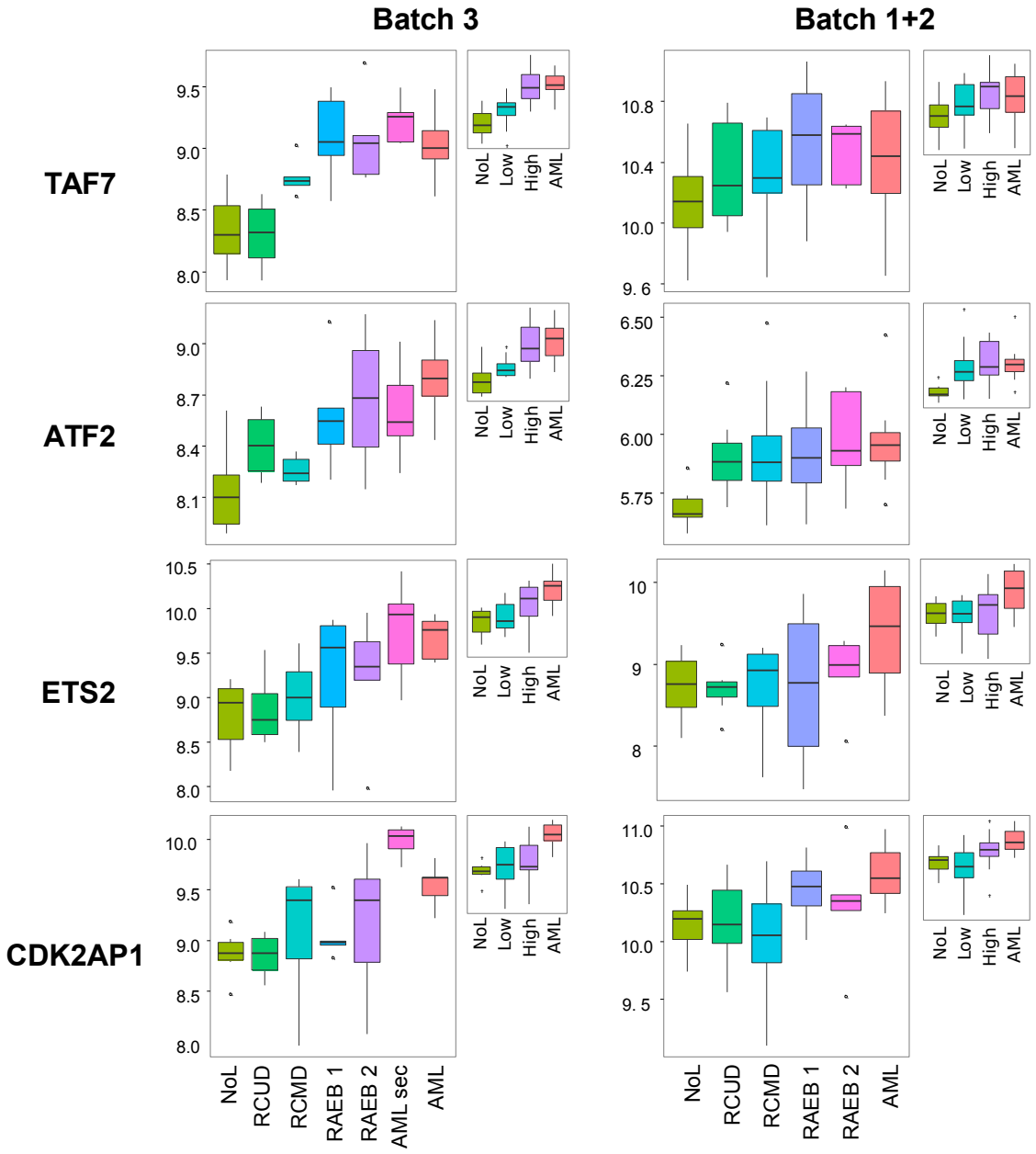


Figure 24. Boxplot of the expression of the four up-regulated TFs in *Figure 23*

4. Discussion

This study analyzed the differences, similarities and evolution of genome-wide expression profiles of subtypes of a malignant hematopathy (myelodysplastic syndromes, MDS) in a comparative frame with non-malignant controls and acute leukemia (as most malignant stage). The work presented the use of multiple bioinformatic methods and strategies to obtain a robust analysis of multiple datasets. The integration of the two independent datasets from the same platform worked very well, probably because both datasets had good quality and were collected by the same laboratories, from the same tissue, and from patients with similar well-controlled diagnosis. The unification of different series allowed to have a big cohort of patients and validation datasets to study the disease subtypes in depth. After a proper selection of samples, our study did not focus on identifying independent unique gene markers for each disease subtype, but rather the identification of common transcriptomic profiles that allowed the investigation of the different stages of MDS and their evolution or progression towards acute leukemia. This methodological approach allowed to identify a set of genes that start getting altered in early stages of MDS and continue that trend in the malignant progression to leukemia. In this way, a key point in this study was the availability of AML samples in addition to the standard control samples –reference of the *original* healthy state–. Although AML was not the objective of the study, having it as reference of the *destination* malignant stage strengthened the analysis by helping to unveil traits normally hidden by the frequent heterogeneity of myelodysplastic syndromes. The availability of samples placed in multiple stages also allowed to discover a series of expression patterns within the genes. We observed that some genes start getting altered at earlier stages and continue the trend at a constant pace in all later stages, while other genes drastically change in the most late stages.

The genes found that make a drastic change in the last stage were already broadly known as related to AML (e.g. HOXA, FLT3, MEIS1). By contrast, the genes mostly altered during the MDS stages are the newly discovered and the most interesting results of this study. The functional enrichment analysis of the genes in these patterns also allowed to further characterize which of the biological functions altered in MDS are somehow a precursor of leukemia, and which are more specific of the early low-risk stages of this disease (e.g. ribosome, histones, splicing). Finally, the analysis of transcription factors, allowed to identify a subset of the altered genes which could potentially be leading the transcriptomic trends observed. The network formed by these transcription factors and their targets also revealed important differences between the patterns. The pattern associated to AML reveals strong inter-regulation between TFs, suggesting a meta-controlled situation with a small subset of regulators very much within a feedback loop to provide some functions (such as continuous proliferation) but stopping many other functions of the cells. On the other hand, the pattern associated to MDS seems to be regulated initially by just a few TFs (e.g. TAF7, ATF2, CEBPG). Although the regulation by these transcription factors will need to be validated experimentally, they seem to be good candidates by having binding motifs within the gene promoters, and being previously associated to functions altered or distorted in the initial stages of MDS.

In conclusion, this work is a good example of a bioinformatics study applied to unravel the molecular basis of complex disease subtypes, done through identification of coexpression transcriptomic profiles along different states. Since the subtypes in this study were stages within a progression, we focused on the expression patterns that tend to increase, decrease or change as the disease evolves. This approach was successfully applied to MDS, a disease known for its heterogeneity, difficulty in the classification of its subtypes, and even difficulty for diagnosis in its early stages. In this way, the method proposed here can be specially useful for analysing heterogeneous diseases without clear subdivisions, or enclosing subtypes that still have to be defined at genomic level, as long as they are stages within a progression.

Conclusions

Conclusions

1. The application of methods and bioinformatic techniques to genomic and transcriptomic data from patients with cancer has been proved efficient to achieve a better characterization of the molecular profiles for each studied disease. In particular, for the recognition of different pathological states sometimes previously loosely defined. This has been proved in this work, through the usage of artificial intelligence and computational techniques which explore the genomic information in a comprehensive and efficient way.

2. Two of the bioinformatic methods presented in this Thesis have been implemented as bioinformatic tools that are now ready for public usage: *geNetClassifier* and *FGNet*.

3. The study of disease subtypes can focus on identifying the genes and characteristics that distinguish the subtypes, but also on what they have in common. The gene profiles that are unique for a specific state are potential disease biomarkers and help to understand the underlying biological process in each of the disease stages. On the other side, the common characteristics allow obtaining a global view of the disease and a better understanding of its development and progression.

4. The integration and bioinformatic analysis of different layers of omic data, not only allows to obtain a better understanding of the diseases, but can also help to unravel the underlying causes.

5. Functional enrichment analyses help to find and understand the biological processes in which the genes altered in a disease are involved. In addition, they also help compare the results between analyses. The altered functions tend to be more stable between studies than the individual genes.

6. The application of the bioinformatic methodologies presented in this Doctoral Thesis to onco-hematologic diseases has allowed to improve their molecular characterization based on omic data. In particular, a transcriptomic profile has been obtained for the evolution and progression of myelodysplastic syndromes (MDS) from their most benign forms, to the higher risk of leukemic transformation. In addition, the integration of complementary expression and methylation profiles has allowed to identify new genomic regulators not previously described in low risk MDS.

Abreviaturas / List of abbreviations

ALL	*Acute lymphoblastic leukemia* Leucemia linfoblástica aguda
AML	*Acute myeloid leukemia* Leucemia mieloblástica aguda
aRNA	*Antisense RNA*
BP	*Biological process (GO)* Proceso biológico
BM	*Bone marrow* Médula ósea
CC	*Cellular component (GO)* Componente celular
CDF	*Chip Definition File* Archivo de definición del chip
cDNA	*Copy DNA* DNA copia
CLL	*Chronic lymphocytic leukemia* Leucemia linfocítica crónica
CML	*Chronic myeloid leukemia* Leucemia mieloide crónica
CpG	*Dinucleotide CG* Dinucleótido CG
FEA	*Functional enrichment analysis* Análisis de enriquecimiento funcional
FDR	*False discovery rate*
FPKM	*Fragments per kilobase of exon model per million mapped* Fragmentos por kilobase de exon por millón de fragmentos mapeados
GEO	*Gene Expression Omnibus (Database)*
GO	*Gene Ontology (Database)*
GSA/GSEA	*Gene Set Enrichment Analysis*
GWAS	*Genome-wide association study* Estudio de asociación del genoma completo
hgu133plus2	*Affymetrix GeneChip Human Genome U133 Plus 2.0 Array*
HuEx1.0	*Affymetrix GeneChip Human Exon 1.0 ST Array*
IQR	*Inter-quartile range* Rango intercuartílico
MCA	*Methylated CpG island amplification* Amplificación de islas CpG metiladas
MCAM	*Methylated CpG island amplification microarray*
MDS	*Myelodysplastic syndrome* Síndrome mielodisplásico
MF	*Molecular function (GO)* Función molecular

miRNA	*Micro RNA*
MM	*Mismatch*
mRNA	*Messenger RNA* RNA mensajero
NoL	*No leukemia*
PCR	*Polymerase chain reaction* Reacción en cadena de polimerasa
PM	*Perfect match*
PPI	*Protein-Protein Interactions* Interacciones entre proteínas
RA	*Refractory anemia* Anemia refractaria
RAEB	*Refractory anemia with excess blasts* Anemia refractaria con exceso de blastos
RARS	*Refractory anemia with ringed sideroblasts* Anemia refractaria con sideroblastos en anillo
RCMD	*Refractory cytopenia with multilineage dysplasia* Citopenia refractaria con displasia multilinaje
RCUD	*Refractory cytopenia with unilineage dysplasia* Citopenia refractaria con displasia unilinaje
RMA	*Robust Multi-array Average*
RNA-seq	*RNA Sequencing*
RPKM	*Reads per kilobase of exon model per million mapped reads* Lecturas por kilobase de exon por millón de lecturas mapeadas
RS	*Ringed sideroblasts* Sideroblastos en anillo
SAM	*Significance Analysis of Microarrays*
SNP	*Single-nucleotide polymorphism* Polimorfismo de un único nucleótido
SOM	*Self Organizing Map*
SVM	*Support Vector Machine* Máquina de soporte vectorial
TF	*Transcription factor* Factor de transcripción
TFBS	*Transcription factor binding sites* Sitios de unión de factor de transcripción
TSS	*Transcription start site* Punto de inicio de transcripción
VST	*Variance-Stabilizing-Transformed*
3C	*Chromosome conformation capture* Captura de la conformación cromosómica
4C	*Circularized chromosome conformation capture*

Bibliografía / References

Alberich-Jordà M, Wouters B, Balastik M, Shapiro-Koss C, Zhang H, DiRuscio A, Radomska HS, Ebralidze AK, Amabile G, Ye M, Zhang J, Lowers I, Avellino R, Melnick A, Figueroa ME, Valk PJM, Delwel R, and Tenen DG (2012) **C/EBPγ deregulation results in differentiation arrest in acute myeloid leukemia**. *J. Clin. Invest. 122*, 4490–4504.

Alexa A, and Rahnenfuhrer J (2010) **topGO: Enrichment analysis for Gene Ontology**. *Bioconductor Package.*

Allocco DJ, Kohane IS, and Butte AJ (2004) **Quantifying the relationship between co-expression, co-regulation and gene function.** *BMC Bioinformatics 5*, 18.

Ambroise C, and McLachlan GJ (2002) **Selection bias in gene extraction on the basis of microarray gene-expression data.** *Proc. Natl. Acad. Sci. USA 99*, 6562–6566.

Barabási A-L, and Oltvai ZN (2004) **Network biology: understanding the cell's functional organization.** *Nat. Rev. Genet. 5*, 101–113.

Barrier A, Lemoine A, Boelle P-Y, Tse C, Brault D, Chiappini F, Breittschneider J, Lacaine F, Houry S, Huguier M, Van der Laan MJ, Speed T, Debuire B, Flahault A, and Dudoit S (2005) **Colon cancer prognosis prediction by gene expression profiling.** *Oncogene 24*, 6155–6164.

Bauer SR, Kudo A, and Melchers F (1988) **Structure and pre-B lymphocyte restricted expression of the VpreB gene in humans and conservation of its structure in other mammalian species Mouw Vpre B 1**. *EMBO J. 7*, 111–116.

Bejar R, Stevenson K, Abdel-Wahab O, Galili N, Nilsson B, Garcia-Manero G, Kantarjian H, Raza A, Levine RL, Neuberg D, and Ebert BL (2011) **Clinical effect of point mutations in myelodysplastic syndromes.** *N. Engl. J. Med. 364*, 2496–2506.

Benjamini Y, and Hochberg Y (1995) **Controlling the false discovery rate: a practical and powerful approach to multiple testing**. *J. R. Stat. Soc. 57*, 289–300.

Bindea G, Mlecnik B, Hackl H, Charoentong P, Tosolini M, Kirilovsky A, Fridman WH, Pagès F, Trajanoski Z, and Galon J (2009) **ClueGO: A Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks**. *Bioinformatics 25*, 1091–1093.

Bodenhofer U, and Klawonn F (2008) **Robust rank correlation coefficients on the basis of fuzzy orderings: initial steps**. *Mathw. Soft Comput. 15*, 5–20.

Bodenhofer U, and Krone M (2011) **RoCoCo: An R package implementing a robust rank correlation coefficient and corresponding test**. *Bioconductor Package.*

Bodenhofer U, Krone M, and Klawonn F (2013) **Testing noisy numerical data for monotonic association**. *Inf. Sci. (Ny). 245*, 21–37.

Bresolin S, Trentin L, Zecca M, Giordan M, Sainati L, Locatelli F, Basso G, and te Kronnie G (2012) **Gene expression signatures of pediatric myelodysplastic syndromes**

are associated with risk of evolution into acute myeloid leukemia. *Leukemia 26*, 1717–1719.

Bullinger L, Döhner K, Bair E, Fröhling S, Schlenk RF, Tibshirani R, Ph D, Döhner H, and Pollack JR (2004) **Use of Gene-Expression Profiling to Identify Prognostic Subclasses in Adult Acute Myeloid Leukemia**. *N. Engl. J. Med. 350*, 1605–1616.

Bulycheva E, Rauner M, Medyouf H, Theurl I, Bornhäuser M, Hofbauer LC, and Platzbecker U (2015) **Myelodysplasia is in the niche: novel concepts and emerging therapies**. *Leukemia 29*, 259–268.

Cawley S, Bekiranov S, Ng HH, Kapranov P, Sekinger E a., Kampa D, Piccolboni A, Sementchenko V, Cheng J, Williams AJ, Wheeler R, Wong B, Drenkow J, Yamanaka M, Patel S, Brubaker S, Tammana H, Helt G, Struhl K, et al. (2004) **Unbiased Mapping of Transcription Factor Binding Sites along Human Chromosomes 21 and 22 Points to Widespread Regulation of Noncoding RNAs**. *Cell 116*, 499–509.

Cruz JA, and Wishart DS (2006) **Applications of machine learning in cancer prediction and prognosis**. *Cancer Inform. 2*, 59–77.

Culhane AC, Schröder MS, Sultana R, Picard SC, Martinelli EN, Kelly C, Haibe-Kains B, Kapushesky M, St Pierre A-A, Flahive W, Picard KC, Gusenleitner D, Papenhausen G, O'Connor N, Correll M, and Quackenbush J (2012) **GeneSigDB: a manually curated database and resource for analysis of gene expression signatures.** *Nucleic Acids Res 40*, D1060–D1066.

Djebali S, Davis C a, Merkel A, Dobin A, Lassmann T, Mortazavi A, Tanzer A, Lagarde J, Lin W, Schlesinger F, and others (2012) **Landscape of transcription in human cells.** *Nature 489*, 101–108.

Ebert BL, and Golub TR (2004) **Genomic approaches to hematologic malignancies**. *Blood 104*, 923–932.

Erdogan B, Facey C, Qualtieri J, Tedesco J, Rinker E, Isett RB, Tobias J, Baldwin DA, Thompson JE, Carroll M, and Kim AS (2011) **Diagnostic microRNAs in myelodysplastic syndrome**. *Exp. Hematol. 39*, 915–926.

Estécio MRH, Yan PS, Ibrahim AEK, Tellez CS, Shen L, Huang TH, and Issa JJ (2007) **High-throughput methylation profiling by MCA coupled to CpG island microarray**. *Genome Res. 17*, 1529–1536.

Figueroa ME, Skrabanek L, Li Y, Jiemjit A, Fandy TE, Paietta E, Fernandez H, Tallman MS, Greally JM, Carraway H, Licht JD, Steven D, Melnick A, and Gore SD (2009) **MDS and secondary AML display unique patterns and abundance of aberrant DNA methylation**. *Blood 114*, 3448–3458.

Fontanillo C, Nogales-Cadenas R, Pascual-Montano A, and De las Rivas J (2011) **Functional analysis beyond enrichment: non-redundant reciprocal linkage of genes and biological terms.** *PLoS One 6*, e24289.

Forbes S a., Beare D, Gunasekaran P, Leung K, Bindal N, Boutselakis H, Ding M, Bamford S, Cole C, Ward S, Kok CY, Jia M, De T, Teague JW, Stratton MR, McDermott U, and Campbell PJ (2014) **COSMIC: exploring the world's knowledge of somatic mutations in human cancer**. *Nucleic Acids Res. 43*, D805–D811.

Franceschini A, Szklarczyk D, Frankild S, Kuhn M, Simonovic M, Roth A, Lin J, Minguez P, Bork P, Von Mering C, and Jensen LJ (2013) **STRING v9.1: Protein-protein interaction networks, with increased coverage and integration**. *Nucleic Acids Res. 41*, 808–815.

Galili N, Qasim SA, and Raza A (2009) **Defective ribosome biogenesis in myelodysplastic syndromes**. *Haematologica 94*, 1336–1338.

Gevaert O (2015) **MethylMix: an R package for identifying DNA methylation driven genes.** *Bioconductor Package.*

Giorgi FM, Del Fabbro C, and Licausi F (2013) **Comparative study of RNA-seq- and Microarray-derived coexpression networks in Arabidopsis thaliana**. *Bioinformatics 29*, 717–724.

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, and Lander ES (1999) **Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring**. *Science 286*, 531–537.

Goodman LA, and Kruskal WH (1954) **Measures of association for cross classifications**. *J. Am. Stat. Assoc. 49*, 732–764.

Grubach L, Juhl-Christensen C, Rethmeier A, Olesen LH, Aggerholm A, Hokland P, and Østergaard M (2008) **Gene expression profiling of Polycomb, Hox and Meis genes in patients with acute myeloid leukaemia**. *Eur. J. Haematol. 81*, 112–122.

Haferlach T, Kohlmann A, Wieczorek L, Basso G, Kronnie G Te, Béné M-C, De Vos J, Hernández JM, Hofmann W-K, Mills KI, Gilkes A, Chiaretti S, Shurtleff S a, Kipps TJ, Rassenti LZ, Yeoh AE, Papenhausen PR, Liu W-M, Williams PM, et al. (2010) **Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the International Microarray Innovations in Leukemia Study Group.** *J. Clin. Oncol. 28*, 2529–2537.

Hanahan D, and Weinberg RA (2000) **The hallmarks of cancer**. *Cell 100*, 57–70.

Harewood L, Schütz F, Boyle S, Perry P, Delorenzi M, Bickmore W a., and Reymond A (2010) **The effect of translocation-induced nuclear reorganization on gene expression**. *Genome Res. 20*, 554–564.

Harrow J, Frankish A, Gonzalez JM, Tapanari E, Diekhans M, Kokocinski F, Aken BL, Barrell D, Zadissa A, Searle S, and others (2012) **GENCODE: The reference human genome annotation for The ENCODE Project**. *Genome Res. 22*, 1760–1774.

Ho Sui SJ, Mortimer JR, Arenillas DJ, Brumm J, Walsh CJ, Kennedy BP, and Wasserman WW (2005) **oPOSSUM: Identification of over-represented transcription factor binding sites in co-expressed genes**. *Nucleic Acids Res. 33*, 3154–3164.

Huang DW, Sherman BT, and Lempicki R a (2009a) **Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists.** *Nucleic Acids Res. 37*, 1–13.

Huang DW, Sherman BT, and Lempicki R a (2009b) **Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources.** *Nat. Protoc. 4*, 44–57.

Hussein K, Theophile K, Büsche G, Schlegelberger B, Göhring G, Kreipe H, and Bock O (2010) **Aberrant microRNA expression pattern in myelodysplastic bone marrow cells**. *Leuk. Res. 34*, 1169–1174.

Irizarry R a, Bolstad BM, Collin F, Cope LM, Hobbs B, and Speed TP (2003a) **Summaries of Affymetrix GeneChip probe level data.** *Nucleic Acids Res. 31*, e15.

Irizarry RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, and Speed TP (2003b) **Exploration, normalization, and summaries of high density oligonucleotide array probe level data**. *Biostatistics 4*, 249–264.

Janky R, Verfaillie A, Imrichová H, Van de Sande B, Standaert L, Christiaens V, Hulselmans G, Herten K, Naval Sanchez M, Potier D, Svetlichnyy D, Kalender Atak Z, Fiers M, Marine JC, and Aerts S (2014) **iRegulon: From a gene list to a gene regulatory network using large motif and track collections**. *PLoS Comput. Biol. 10*, e1003731.

Jiang Y, Dunbar A, Gondek LP, Mohan S, Rataul M, Keefe CO, Sekeres M, Saunthararajah Y, Maciejewski JP, and Dc W (2009) **Aberrant DNA methylation is a dominant mechanism in MDS progression to AML**. *Blood 113*, 1315–1325.

Johnson WE, Li C, and Rabinovic A (2007) **Adjusting batch effects in microarray expression data using empirical Bayes methods**. *Biostatistics 8*, 118–127.

Jones MJ, Fejes AP, and Kobor MS (2013) **DNA methylation, genotype and gene expression: who is driving and who is along for the ride?**. *Genome Biol. Biol. 14*, 5–7.

Kauffmann A, Gentleman R, and Huber W (2009) **arrayQualityMetrics - A bioconductor package for quality assessment of microarray data**. *Bioinformatics 25*, 415–416.

Kendall MG (1938) **A new measure of rank correlation**. *Biometrika 30*, 81–93.

Kendziorski CM, Newton MA, Lan H, and Gould MN (2003) **On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles**. *Stat. Med. 22*, 3899–3914.

Khatri P, Sirota M, and Butte AJ (2012) **Ten years of pathway analysis: current approaches and outstanding challenges.** *PLoS Comput. Biol. 8*, e1002375.

Kiełbasa SM, Klein H, Roider HG, Vingron M, and Blüthgen N (2010) **TransFind - predicting transcriptional regulators for gene sets**. *Nucleic Acids Res. 38*, 275–280.

Kohonen T (1982) **Self-organized formation of topologically correct feature maps**. *Biol. Cybern. 43*, 59–69.

Kristensen VN, Lingjærde OC, Russnes HG, Vollan HKM, and Frigessi A (2014) **Principles and methods of integrative genomic analyses in cancer**. *Nat. Publ. Gr. 14*, 299–313.

Laird PW (2003) **The power and the promise of DNA methylation markers.** *Nat. Rev. Cancer 3*, 253–266.

Laird PW (2010) **Principles and challenges of genome-wide DNA methylation analysis**. *Nat. Rev. Genet. 11*, 191–203.

Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano JA, Armañanzas R, Santafé G, Pérez A, and Robles V (2006) **Machine learning in bioinformatics**. *Brief. Bioinform. 7*, 86–112.

De Las Rivas J, and Fontanillo C (2010) **Protein–Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks**. *PLoS Comput. Biol. 6*, e1000807.

Ley TJ, Miller C, and The Cancer Genome Atlas Research Network (2013) **Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia**. *N. Engl. J. Med. 368*, 2059–2074.

Lokk K, Modhukur V, Rajashekar B, Märtens K, Mägi R, Kolde R, Kolt Ina M, Nilsson TK, Vilo J, Salumets A, and Tõnisson N (2014) **DNA methylome profiling of human tissues identifies global and tissue-specific methylation patterns.** *Genome Biol. 15*, R54.

Luo W, Friedman MS, Shedden K, Hankenson KD, and Woolf PJ (2009) **GAGE: generally applicable gene set enrichment for pathway analysis.** *BMC Bioinformatics 10*, 161.

Luscombe NM, Babu MM, Yu H, Snyder M, Teichmann SA, and Gerstein M (2004) **Genomic analysis of regulatory network dynamics reveals large topological changes**. *Nature 431*, 714–717.

Mani K, Sandgren S, Lilja J, Cheng F, Svensson K, Persson L, and Belting M (2007) **HIV-Tat protein transduction domain specifically attenuates growth of polyamine deprived tumor cells.** *Mol. Cancer Ther. 6*, 782–788.

Marguerat S, Schmidt A, Codlin S, Chen W, Aebersold R, and Bähler J (2012) **Quantitative analysis of fission yeast transcriptomes and proteomes in proliferating and quiescent cells**. *Cell 151*, 671–683.

McCall MN, Bolstad BM, and Irizarry R a (2010) **Frozen robust multiarray analysis (fRMA).** *Biostatistics 11*, 242–253.

Merico D, Isserlin R, Stueker O, Emili A, and Bader GD (2010) **Enrichment Map: a network-based method for gene-set enrichment visualization and interpretation.** *PLoS One 5*, e13984.

Merkerova MD, Krejcik Z, Votavova H, Belickova M, Vasikova A, and Cermak J (2011) **Distinctive microRNA expression profiles in CD34+ bone marrow cells from patients with myelodysplastic syndrome.** *Eur. J. Hum. Genet. 19*, 313–319.

Merkerova MD, Krejcik Z, Belickova M, Hrustincova A, Klema J, Stara E, Zemanova Z, Michalova K, Cermak J, and Jonasova A (2014) **Genome-wide miRNA profiling in myelodysplastic syndrome with del(5q) treated with lenalidomide**. *Eur. J. Haematol. Adv.Access.*

Meyer PE, Lafitte F, and Bontempi G (2008) **minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information.** *BMC Bioinformatics 9*, 461.

Millenaar FF, Okyere J, May ST, van Zanten M, Voesenek L a CJ, and Peeters AJM (2006) **How to decide? Different methods of calculating gene expression from short oligonucleotide array data will give different results.** *BMC Bioinformatics 7*, 137.

Morris C (1983) **Parametric Empirical Bayes Inference: Theory and Applications**. *J. Am. Stat. Assoc. 78*, 47–65.

Noushmehr H, Weisenberger DJ, Diefes K, Phillips HS, Pujara K, Berman BP, Pan F, Pelloski CE, Sulman EP, Bhat KP, Verhaak RGW, Hoadley K a., Hayes DN, Perou CM, Schmidt HK, Ding L, Wilson RK, Van Den Berg D, Shen H, et al. (2010) **Identification of a CpG Island methylator phenotype that defines a distinct subgroup of glioma**. *Cancer Cell 17*, 510–522.

Obayashi T, Okamura Y, Ito S, Tadaka S, Motoike IN, and Kinoshita K (2013) **COXPRESdb: A database of comparative gene coexpression networks of eleven species for mammals**. *Nucleic Acids Res. 41*, 1014–1020.

Orlovsky K, Kalinkovich A, Rozovskaia T, Shezen E, Itkin T, Alder H, Ozer HG, Carramusa L, Avigdor A, Volinia S, Buchberg A, Mazo A, Kollet O, Largman C, Croce CM, Nakamura T, Lapidot T, and Canaani E (2011) **Down-regulation of homeobox genes MEIS1 and HOXA in MLL-rearranged acute leukemia impairs engraftment and reduces proliferation**. *Proc. Natl. Acad. Sci. USA 108*, 7956–7961.

Papaemmanuil E, Gerstung M, Malcovati L, Tauro S, Gundem G, Loo P Van, Yoon CJ, Ellis P, Wedge DC, Pellagatti A, Shlien A, Groves MJ, Forbes S a, Raine K, Hinton J, Mudie LJ, Mclaren S, Hardy C, Latimer C, et al. (2013) **Clinical and biological implications of driver mutations in myelodysplastic syndromes**. *Blood 122*, 3616–3627.

Parker JE, Fishlock KL, Mijovic a., Czepulkowski B, Pagliuca a., and Mufti GJ (1998) "**Low-risk" myelodysplastic syndrome is associated with excessive apoptosis and an increased ratio of pro- versus anti-apoptotic bcl-2-related proteins**. *Br. J. Haematol. 103*, 1075–1082.

Patel AP, Tirosh I, Trombetta JJ, Shalek AK, Gillespie SM, Wakimoto H, Cahill DP, Nahed B V., Curry WT, Martuza RL, Louis DN, Rozenblatt-Rosen O, Suvà ML, Regev A, and Bernstein BE (2014) **Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma**. *Science 344*, 1396–1401.

Pellagatti A, Cazzola M, Giagounidis AAN, Malcovati L, Porta MG Della, Killick S, Campbell LJ, Wang L, Langford CF, Fidler C, Oscier D, Aul C, Wainscoat JS, and Boultwood J (2006) **Gene expression profiles of CD34+ cells in myelodysplastic syndromes: involvement of interferon-stimulated genes and correlation to FABsubtype and karyotype**. *Blood 108*, 337–346.

Pellagatti A, Cazzola M, Giaggounidis A, Perry J, Malcovati L, Della Porta M, Jädersten M, Killick S, Verma A, Norbury C, Hellström-Lindberg E, Wainscoat J, and Boultwood J (2010) **Deregulated gene expression pathways in myelodysplastic syndrome hematopoietic stem cells**. *Leukemia 24*, 756–764.

Pirooznia M, Yang JY, Yang MQ, and Deng Y (2008) **A comparative study of different machine learning methods on microarray gene expression data**. *BMC Genomics 9*, S13.

Raza A, and Galili N (2012) **The genetic basis of phenotypic heterogeneity in myelodysplastic syndromes.** *Nat. Rev. Cancer 12*, 849–859.

Raza a, Gezer S, Mundle S, Gao XZ, Alvi S, Borok R, Rifkin S, Iftikhar a, Shetty V, and Parcharidou a (1995) **Apoptosis in bone marrow biopsy samples involving stromal and hematopoietic cells in 50 patients with myelodysplastic syndromes.** *Blood 86*, 268–276.

Reid JF, Gariboldi M, Sokolova V, Capobianco P, Lampis A, Perrone F, Signoroni S, Costa A, Leo E, Pilotti S, and Pierotti MA (2009) **Integrative Approach for Prioritizing Cancer Genes in Sporadic Colon Cancer**. *Genes Chromosomes Cancer 48*, 953–962.

Del Rey M, O'Hagan K, Dellett M, Aibar S, Colyer H, Alonso M, Diez-Campelo M, Armstrong RN, Sharpe DJ, Gutierrez N, García JL, De Las Rivas J, Mills KI, and Hernandez-Rivas J (2013) **Genome-wide profiling of methylation identifies novel targets with aberrant hypermethylation and reduced expression in low-risk myelodysplastic syndromes**. *Leukemia 27*, 610–618.

Rhrissorrakrai K, Rice JJ, Boue S, Talikka M, Bilal E, Martin F, Meyer P, Norel R, Xiang Y, Stolovitzky G, Hoeng J, and Peitsch MC (2014) **sbv IMPROVER Diagnostic Signature Challenge**. *Syst. Biomed. 1*, 196–207.

Rhyasen G, and Starczynowski D (2012) **Deregulation of microRNAs in myelodysplastic syndrome**. *Leukemia 26*, 13–22.

Ridder D de, Ridder J de, and Reinders M (2013) **Pattern recognition in bioinformatics**. *Brief. Bioinform. 14*, 633–647.

Risueño A, Fontanillo C, Dinger ME, and De Las Rivas J (2010) **GATExplorer: genomic and transcriptomic explorer; mapping expression probes to gene loci, transcripts, exons and ncRNAs**. *BMC Bioinformatics 11, 221.*

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, and Smyth GK (2015) **limma powers differential expression analyses for RNA-sequencing and microarray studies**. *Nucleic Acids Res. Adv.Access.*

Sandoval J, and Esteller M (2012) **Cancer epigenomics: beyond genomics.** *Curr. Opin. Genet. Dev. 22*, 50–55.

Sanger F, Air G, Barrell B, BrownL N, Coulson A, Fiddes C, Hutchison C, Slocombe P, and Smith M (1977) **Nucleotide sequence of bacteriophage phi X174 DNA**. *Nature 265*, 687–695.

Schena W, Shalon D, Davis R, and Brown P (1995) **Quantitative monitoring of gene expression patterns with a complementary DNA microarray**. *Science 270*, 467470.

Schliemann C, Bieker R, Padro T, Kessler T, Hintelmann H, Buchner T, Berdel WE, and Mesters RM (2006) **Expression of angiopoietins and their receptor Tie2 in the bone marrow of patients with acute myeloid leukemia**. *Hemtaologica 91*, 1203–1211.

Seth A, and Watson DK (2005) **ETS transcription factors and their emerging roles in human cancer**. *Eur. J. Cancer 41*, 2462–2478.

Shen L, Kantarjian H, Guo Y, Lin E, Shan J, Huang X, Berry D, Ahmed S, Zhu W, Pierce S, Kondo Y, Oki Y, Jelinek J, Saba H, Estey E, and Issa J-PJ (2010) **DNA methylation predicts survival and response to therapy in patients with myelodysplastic syndromes.** *J. Clin. Oncol. 28*, 605–613.

Siegmund KD (2012) **Statistical Approaches for the Analysis of DNA Methylation Microarray Data**. *Hum. Genet. 129*, 585–595.

Slawski M, Daumer M, and Boulesteix A-L (2008) **CMA: a comprehensive Bioconductor package for supervised classification with high dimensional data.** *BMC Bioinformatics 9*, 439.

Spearman CE (1904) **The proof and measurement of association betweentwo things**. *Am. J. Psychol. 15*, 72–101.

Spearman CE (1907) **Demonstration of formulae for true measurement of correlation**. *Am. J. Psychol. 18*, 161–169.

Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, and Levy S (2005) **A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis**. *Bioinformatics 21*, 631–643.

Statnikov A, Wang L, and Aliferis CF (2008) **A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification.** *BMC Bioinformatics 9*, 319.

Stuart JM, Segal E, Koller D, Stuart SKKM, Segal E, Koller D, and Kim SK (2003) **A gene-coexpression network for global discovery of conserved genetic modules**. *Science 302*, 249–255.

Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, and Mesirov JP (2005) **Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles.** *Proc. Natl. Acad. Sci. USA 102*, 15545–15550.

Tabas-Madrid D, Nogales-Cadenas R, and Pascual-Montano A (2012) **GeneCodis3: A non-redundant and modular enrichment analysis tool for functional genomics**. *Nucleic Acids Res. 40*, 478–483.

Taminau J, Meganck S, Lazar C, Steenhoff D, Coletta A, Molter C, Duque R, de Schaetzen V, Weiss D, Bersini H, and Nowe A (2012) **Unlocking the potential of publicly available microarray data using inSilicoDb and inSilicoMerging R/Bioconductor packages**. *BMC Bioinformatics 13*, 335.

Teichmann SA, and Babu MM (2002) **Conservation of gene co-regulation in prokaryotes and eukaryotes**. *Trends Biotechnol. 20*, 407–410.

Theilgaard-Mönch K, Boultwood J, Ferrari S, Giannopoulos K, Hernandez-Rivas JM, Kohlmann A, Morgan M, Porse B, Tagliafico E, Zwaan CM, Wainscoat J, Van den Heuvel-Eibrink MM, Mills K, and Bullinger L (2011) **Gene expression profiling in MDS and AML: potential and future avenues.** *Leukemia 6*, 909–920.

Tonon L, Touzet H, and Varré JS (2010) **TFM-Explorer: Mining cis-regulatory regions in genomes**. *Nucleic Acids Res. 38*, 286–292.

Toyota M, Ho C, Ahuja N, Jair KW, Li Q, Ohe-Toyota M, Baylin SB, and Issa JPJ (1999) **Identification of differentially methylated sequences in colorectal cancer by methylated CpG island amplification**. *Cancer Res. 59*, 2307–2312.

Trapnell C, Williams B a, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, and Pachter L (2011) **Transcript assembly and abundance estimation from RNA-Seq reveals thousands of new transcripts and switching among isoforms**. *Nat. Biotechnol. 28*, 511–515.

Tusher VG, Tibshirani R, and Chu G (2001) **Significance analysis of microarrays applied to the ionizing radiation response.** *Proc. Natl. Acad. Sci. USA 98*, 5116–5121.

Vandelmolen L, Rice L, Rose MA, and Lynch EC (1988) **Ringed Sideroblasts in Primary MyelodysplasiaLeukemic Propensity and Prognostic Factors**. *Arch. Intern. Med. 148*, 653–656.

Vaquerizas JM, Kummerfeld SK, Teichmann SA, and Luscombe NM (2009) **A census of human transcription factors: function, expression and evolution**. *Nat. Rev. Genet. 10*, 252–263.

Vardiman J, Thiele J, Arber D, Brunning R, Borowitz M, Porwit A, Harris N, Le Beau M, Hellström-Lindberg E, Tefferi A, and Bloomfield C (2009) **The 2008 revision of the WHO classification of myeloid neoplasms and acute leukemia: rationale and important changes.** *Blood 114*, 937–952.

Venet D, Dumont JE, and Detours V (2011) **Most Random Gene Expression Signatures Are Significantly Associated with Breast Cancer Outcome**. *PLoS Comput. Biol. 7*, e1002240.

Warde-Farley D, Donaldson SL, Comes O, Zuberi K, Badrawi R, Chao P, Franz M, Grouios C, Kazi F, Lopes CT, Maitland A, Mostafavi S, Montojo J, Shao Q, Wright G, Bader GD, and Morris Q (2010) **The GeneMANIA prediction server: Biological network integration for gene prioritization and predicting gene function**. *Nucleic Acids Res. 38*, 214–220.

Warden CD, Lee H, Tompkins JD, Li X, Wang C, Riggs AD, Yu H, Jove R, and Yuan YC (2013) **COHCAP: An integrative genomic pipeline for single-nucleotide resolution DNA methylation analysis**. *Nucleic Acids Res. 41*, 1–11.

Wehrens R, and Buydens LMC (2007) **Self-and super-organizing maps in R: the Kohonen package**. *J. Stat. Softw. 21*, 5.

Weidner CI, Lin Q, Koch CM, Eisele L, Beier F, Ziegler P, Bauerschlag DO, Jöckel K-H, Erbel R, Mühleisen TW, and others (2014) **Aging of blood can be tracked by DNA methylation changes at just three CpG sites.** *Genome Biol. 15*, R24.

Woolthuis CM, Han L, Verkaik-Schakel RN, Gosliga D van, Kluin PM, Vellenga E, Schuringa JJ, and Huls G (2012) **Downregulation of MEIS1 impairs long-term expansion of CD34+ NPM1-mutated acute myeloid leukemia cells**. *Leukemia 26*, 848–853.

Yuan M, Wang P, Newton M, and Kendziorski C (2007) **EBarrays. Unified approach for simultaneous gene clustering and differential expression identification**. *Bioconductor Package*.

Zambelli F, Pesole G, and Pavesi G (2009) **Pscan: Finding over-represented transcription factor binding site motifs in sequences from co-regulated or co-expressed genes**. *Nucleic Acids Res. 37*, 247–252.

Zhang K, Pirooznia M, Arabnia HR, Yang JY, Wang L, Luo Z, and Deng Y (2011) **Genomic signatures and gene networking: challenges and promises.** *BMC Genomics 12 Suppl 5*, I1.

Zhang L, Padron E, and Lancet J (2015) **The molecular basis and clinical significance of genetic mutations identified in myelodysplastic syndromes**. *Leuk. Res. 39*, 6–17.

Zhu X, Gerstein M, and Snyder M (2007) **Getting connected: analysis and principles of biological networks.** *Genes Dev. 21*, 1010–1024.