M2 BIM - RESYS Project:
Master Regulators for Metastatic behavior in Osteosarcoma

By Alexis TRANG and Yann ZHONG

Contents

Introduction

General description, copied from project prompt:

An osteosarcoma (OS) or osteogenic sarcoma (OGS) (or simply bone cancer) is a cancerous tumor in a bone. Specifically, it is an aggressive malignant neoplasm that arises from primitive transformed cells of mesenchymal origin (and thus a sarcoma) and that exhibits osteoblastic differentiation and produces malignant osteoid. Osteosarcoma is the most common histological form of primary bone cancer. It is most prevalent in teenagers and young adults. Overall survival of patients with metastatic disease is approximately twenty percent. Mechanisms behind the development of metastases in osteosarcoma are unknown. To identify gene signatures that play a role in metastasis, a study performed genome-wide gene expression profiling on pre-chemotherapy biopsies of osteosarcoma patients who developed metastases within 5yrs and patients who did not develop metastases within 5yrs.

In genetics, a Master regulator is a gene at the top of a gene regulation hierarchy, particularly in regulatory pathways related to cell fate and differentiation. When analyzing the signature of a specific behavior in a disease, you can obtain the transcription factors that are master regulators for that phenomenon, that is, responsible for the behavior you see. In this case, we're talking about

metastatic behavior. RTN is an R package specialized at inferring gene regulatory networks, based on ARACNe.

Step by step, as according to project description:

1. Download gene expression data.
2. Install RTN package, and infer regulatory network based on downloaded gene expression data.
3. Install snow package for parallel processing in RTN.
4. Install RedeR package to better visualize inferred network.
5. Perform differential gene expression analysis between metastatic and non-metastatic biopsies, obtaining a signature for metastatic behavior of this cancer in the patients.
6. Run a Master Regulator Analysis to infer putative Master Regulators by using inferred network and obtained signatures.
7. Get biological insight into the putative Master Regulators at the Human Protein Atlas and at Gene Cards.

Background

The project revolves around using gene expression data downloaded from GEO (the Gene Expression Omnibus), a large online database for genomics data, notably data from microarray or high throughput sequencing experiments for use in determining gene expression activity.

In this project, the dataset we are interested in is the GSE21257 dataset, which corresponds to the following title: **"Genome-wide gene expression profiling on pre-chemotherapy biopsies of osteosarcoma patients who developed metastases within 5yrs (n=34) and patients who did not develop metastases within 5yrs (n=19)"**[1]. This dataset became public on the 11th of February 2011.

The dataset and conclusions from the paper

The data we are looking at is the gene expression data from a total of 53 patients who were diagnosed with osteosarcoma (bone cancer) and who were due for chemotherapy. Of those patients, 34 developed metastases (the spread of the cancer throughout the rest of the body, which essentially indicates the development of the cancer) within 5 years, and 19 did not. The paper associated to this study is **Buddingh et al. (2011)**, and sequencing was done through Microarray technology (and not next generation sequencing, which is become increasingly the norm) using Illumina silica beadchips.

To analyze said data, Buddingh et al. performed DE (Differential Expression) analysis on the 53 patients – DE consists of looking at a control group (in this case, the patients who did develop metastasis) and a "differential" group (in this case the patients who did no develop metastasis). This analysis brings us to the fundamental concept of gene regulatory networks, notably upregulated and downregulated genes.

An **upregulated** cell will have its gene products such as RNA/proteins more strongly expressed and produced in response to an external stimulus, whereas a **downregulated** gene will have its gene products less expressed. A hallmark of cancer is also DNA damage, which is caused by the inability of the body to repair natural DNA mutations. Such inability is, in the case of cancers such as bladder cancer, stomach cancer, or thyroid cancer, caused by downregulation of the MGMT gene (a DNA repair gene). [2][3][4]

What was found was that from the two cohorts, there were a total of 139 significantly differentially expressed genes, with **125 being upregulated and 14 downregulated**. Example of such genes that the authors further confirmed experimentally through the gold standard of RT-qPCR are

CD14 and HLA-DRA, which were found to be barely present patients who did exhibit metastases. Of all of those, roughly half (20+25 = 45%) were associated (strongly or indirectly) to upregulation of macrophage or immunological functions.

These macrophages are mainly M2 macrophages but can also become M1 – going from tumor's "friend" to "foe" (M1 is better). TAMs are as such a mix of M1 antitumor and M2 protumor macrophages, being a heterogeneous cell population. M2 "alternatively activated TAMs (Tumor Associated Macrophages)" are associated with worse patient prognostics in numerous cancer types. However, here, it seems that they have shown that the larger presence of infiltrating TAMs correlated with better survival odds for patients with osteosarcoma.

The paper's authors conclude that that treatment with MTP (or L-MTP-PE): a macrophage activating agent, could potentially yield beneficial results relative to cancer treatment. The next step for us was then to try and visualize a regulatory network out of this dataset using RTN.

Network inference and Master Regulator inference with RTN

RTN is a Bioconductor package which was developed by Fletcher et al. (2013) [5], in an effort to visualize networks associated to genes and regulators that were likely to play a role in breast cancer development, more notably the FGFR2 (fibroblast growth factor receptor 2) locus and their respective Master regulators (MRs).

Primarily, what interests us in this paper is the background but mainly the methods for the package, as RTN is what we will be using to infer a regulatory network for our own osteosarcoma dataset. With the RTN package, we will be obtaining the required network with a preprocessed set of input files. The first of these required files can be acquired remotely or simply downloaded and contains the entirety of the probes obtained from microarray sequencing of a specific beadchip (such as, in our case, the GPL10295 beadchip [6]). This beadchip can have thousands upon thousands of probes that may be linked to a gene (with sometimes multiple probes being linked to a gene, such as the 63Oiusg8JTF.19NHfo and Nonl3upr2ojFAogAsU probes both being tied to the ILMN_3630 – or ABCA5, an ATP Binding gene).

As such, given the exceedingly large number of probes (a stretch of DNA that has been labeled as shown earlier), we would only like to consider likely transcription/regulatory factors pertaining to the associated probes – this was directly given to us in a file called *dt4rtn_TFs.txt*. After having processed the data and the RFs that we would like to identify within our input data, then, a network of regulons – a group of genes regulated as a unit – can be built, using the TNI class within the RTN package, following the TNI pipeline[7]. Furthermore, using the TNA pipeline, we can advance our analysis into the search for Master regulators while still using the RTN package.

The methods for network inference will thus be detailed below in the relevant section, with the steps broken down in respective subsections.

Methods and discussion

The entirety of our work shall be presented in a R markdown file, which will also be attached a knitted html file.

Installations and preprocessing

We start the code by installing BiocManager if required, and using this Bioconda package in order to install all the rest of the necessary packages in RTN, snow, RedeR, limma, Biobase and GEOquery. Following this, the packages are simply activated by usage of the *library* function.

Then, we need to preprocess the input data before it can be used as input to the RTN package functions. This code, which was provided to us, starts by getting GEO data remotely with the *getGEO* function from the GEOquery package. A *GSE21257_series_matrix.txt* file is acquired, which is then referenced against the GPL10295 platform (the specific Illumina beadchip) in order to get the probes (and thus, the TFs) that interest us in the case of this network construction.

Once all the necessary data has been manipulated to fit our input parameters, we use a TNI (Transcription Regulatory Network) constructor to create our network object, having passed through it all the relevant probe elements and TF names and their associated IDs.

```
rtni <- tni.constructor(expData=ex, regulatoryElements=tfs,
rowAnnotation=gexpIDs, cvfilter=T)
```

Using RTN to infer regulatory networks

Once the input data has been successfully preprocessed, using the RTN vignette as a reference, we start inferring the regulatory networks. We start by running a permutation analysis with 1000 permutations total, as the documentation recommended running this analysis with a number of permutations equal to greater than 1000.

Following this, we use two distinct methods to remove both unstable and weak interactions: bootstrap and ARACNE respectively. Bootstrapping is a statistical technique commonly used in machine learning to estimate quantities about a population through the averaging of small estimates drawn out and put back into the population [9]. While the documentation for the TNI bootstrap does not go into the specifics, we can imagine that by a similar method, bootstrapping in this context would reduce the likelihood of keeping "unstable" links that are very far from the median of the regulon values.

ARACNE [10] is an algorithm covered during lectures that aims to remove possible indirect relationships between nodes through the computation of mutual information. By applying the Data Processing Inequality (which states that no information may be gained from post-processing of a signal – or simply that a signal's quality can only be lost and never be gained when going through "noisy" channels), only the most likely path of information flow (or in this case, regulation) will be kept.

Once both of these "filtering" methods are applied, to get a general overview of the inferred network, we run the tni package *summary* function on our network, indicating a network comprised of 140/141 regulons controlled by different genes.

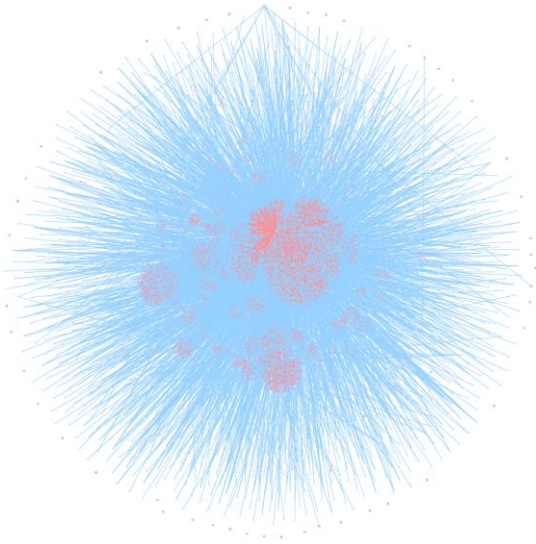
Of note: we both ran the same code but ended up with different number of elements in our regulons object, as well as different numbers of positive and negative links for the genes: for example in Yann's code the YEATS4 object had 1580 links, and in Alexis' code the same object had 1519 links. We believe this variation to be due to the possible permutation and/or bootstrap and ARACNE filters.

Using RedeR to visualize the preliminary output

RedeR is a package which was designed to link R to Java in order to have an interactive network for us to visualize. In order to do so, an Apache server is built with the *RedPort* function and assigned to a rdp object. Then, we call the object to launch the RedeR interface and use *addGraph* as well as *addLegend* in order to plot out our inferred network object. The *relax* function [11] is used to allow the complex network edges to be considered as springs that can exert repulsive or attractive forces based on a set of parameters, such as target length, stiffness, repel factor, cooling factor…

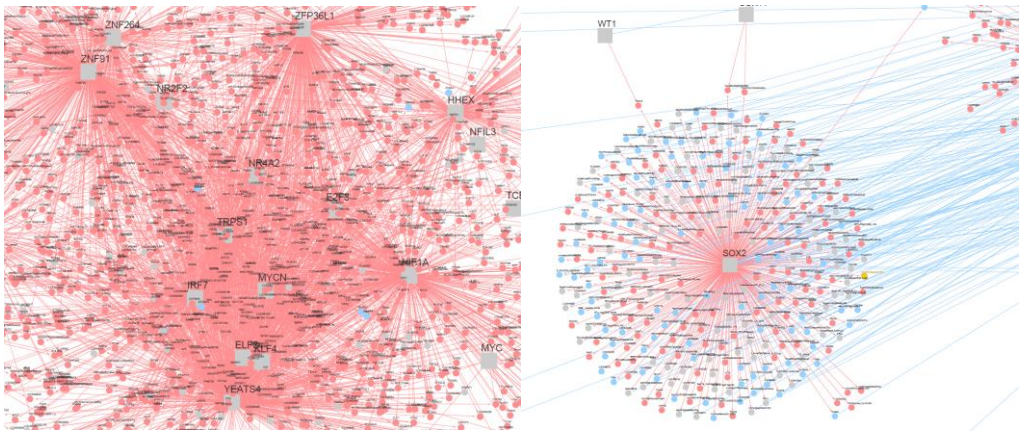
In our case, we plot the full inferred network with the regulons all displayed on one graph, giving a very (very) global view of our network, with the code below:

```
g <- tni.graph(rtni, regulatoryElements = names(regulons))
```



Full view of inferred network after relaxation

With a more zoomed in view on various parts becoming increasingly unreadable:



Left: zoom of central cluster (YEATS4, MYCN, etc.); Right: zoom of leftmost cluster SOX2

In fact, looking at the *regulon* object in R-Studio interface gives us a good indicator of how many elements the top 10 genes actually have a link to, out of the 140 (again, these numbers are based on one of our two codes, with the other one having a slight variation on the number of links per gene).:

Symbol	Full name of protein encoded by gene	Type	Nb of links
YEATS4	YEATS Domain Containing 4	Protein Coding Gene	1580
ZNF91	Zinc Finger Protein 91	Protein Coding Gene	845
HIF1A	Hypoxia Inducible Factor 1 Subunit Alpha	Protein Coding Gene	709
IRF7	Interferon regulatory factor 7	Protein Coding Gene - RF	638
SOX2	SRY-Box Transcription Factor 2	Protein Coding Gene - TF	480
PAX7	Paired Box 7	Protein Coding Gene - TF	480
TRPS1	Transcriptional Repressor GATA Binding 1	Protein Coding Gene - TF	464
MYCN	MYCN Proto-Oncogene, BHLH Transcription Factor	Protein Coding Gene	419
IRF8	Interferon Regulatory Factor 8	Protein Coding Gene - RF	358
ZNF264	Zinc Finger Protein 264	Protein Coding Gene	342

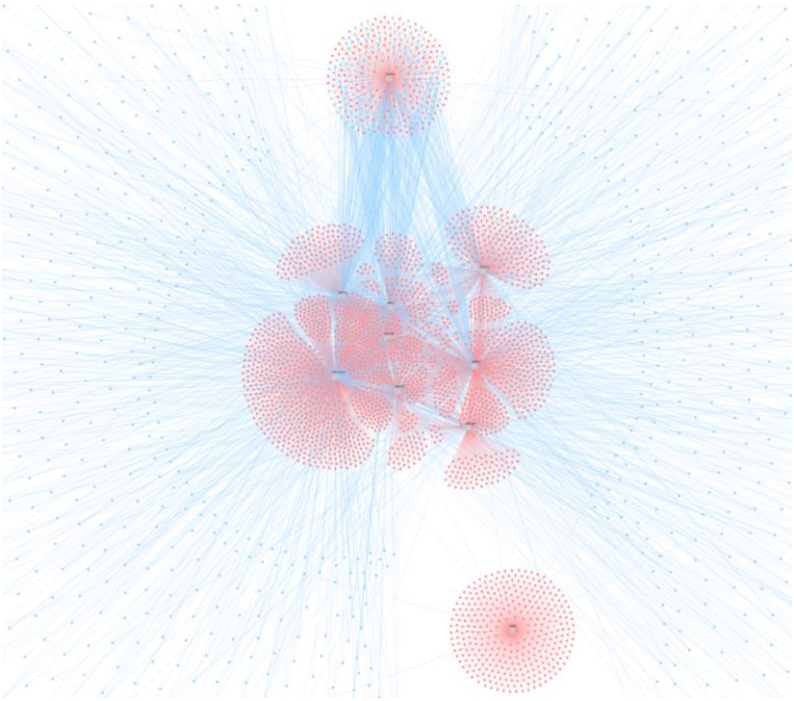
*: RF = Regulatory Factor; TF = Transcriptional Factor

The above information was procured from GeneCards[12], which is a comprehensive database that details the functions of genes as well as links and information relevant to said gene. Of the top 10 hits with the most links (be they positive – upregulated, in red, or negative – downregulated, in blue), 5 are simple genes that code for proteins (many of them commonly found in tumors or have oncogenic links), while 3 others are transcription factors and 2 are regulatory factors.

In order to have a slightly better view on our results, we decided to have an equivalent view with RedeR, only this time, we simply draw out and relax our graph with the top 10 hits:

```
g <- tni.graph(rtni, regulatoryElements =
c("ZNF91","ZNF264","YEATS4","TRPS1","SOX2","PAX7","MYCN","IRF7","IRF8","HIF1A"))
```

A less chaotic look at our top 10 regulons graph is as follows:



Full view of the top 10 regulons with most links

While still very saturated, it becomes a bit clearer – for example, there is a cluster of 8 regulons that group together as the graph relaxes, and there are two regulons farther out from this cluster. The top one is for the gene SOX2, while the bottom one is for the gene IRF8. What's worth noting is also that IRF8 visibly seems to have a sizeable amount of positive regulatory effect – or upregulation, but very few negative regulatory effects – or downregulation.

Differential gene expression analysis with limma

Linear Models for Microarray data (limma) is a package that contains various functions for fitting linear models. In our case, we will use limma to perform a differential expression analysis on our samples. The samples are first log2 transformed, before they're used as input in the limma pipeline. The resulting topTable object provides the necessary data to continue analysis with the RTN package, and more specifically, with the Transcriptional Network Analysis (tna). (See "GEO2R.GSE21257.results")

```
tT <- topTable(fit2, adjust="fdr", sort.by="B", number=nrow(fit2))
```

Using TNA to infer the Master Regulators

Tna is a pipeline that's used to do enrichment analysis on a list of regulons: we filter our data before applying Master Regulator Analysis (MRA) and look for the values of **enrichment score**.

The filtering step is necessary to obtain the "hits", which are our genes of interest. We used different filters (or none at all) to obtain different results from the MRA method. This analysis gives us an MRA object which assesses the overlap between each regulon and the genes listed in the hits (genes that were deemed interesting in the differential expression analysis).

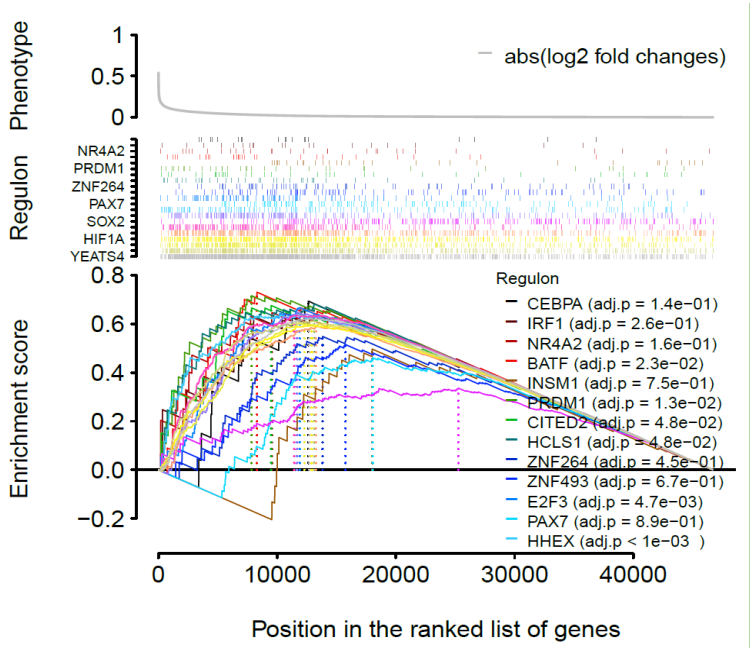
Subsequently, we use the complementary approach **gsea1**, which is a one-tailed gene set enrichment analysis, to find regulons associated with a particular response. This analysis provides us with a tables and graphs giving information on the TFs and their regulons (the following results were obtained with a **filter that only considers the 1000 highest logFC in absolute values**):

	Regulon	Universe.Size	Regulon.Size	Total.Hits	Expected.Hits	Observed.Hits	Pvalue	Adjusted.Pvalue
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
ILMN_3307	BATF	46718	29	46718	29	29	1	1
ILMN_27029	CEBPA	46718	19	46718	19	19	1	1
ILMN_21636	CITED2	46718	37	46718	37	37	1	1
ILMN_12494	E2F3	46718	100	46718	100	100	1	1
ILMN_13615	ELF3	46718	203	46718	203	203	1	1
ILMN_10504	HCLS1	46718	38	46718	38	38	1	1
ILMN_137681	HHEX	46718	149	46718	149	149	1	1
ILMN_9514	HIF1A	46718	443	46718	443	443	1	1
ILMN_22264	INSM1	46718	30	46718	30	30	1	1
ILMN_11739	IRF1	46718	19	46718	19	19	1	1

	Regulon	Universe.Size	Regulon.Size	Total.Hits	Expected.Hits	Observed.Hits	Pvalue	Adjusted.Pvalue
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
ILMN_5194	IRF7	46718	347	46718	347	347	1	1
ILMN_20480	IRF8	46718	233	46718	233	233	1	1
ILMN_28405	NR4A2	46718	28	46718	28	28	1	1
ILMN_6370	PAX7	46718	102	46718	102	102	1	1
ILMN_8384	PRDM1	46718	34	46718	34	34	1	1
ILMN_13292	SOX2	46718	210	46718	210	210	1	1
ILMN_16428	YEATS4	46718	1115	46718	1115	1115	1	1
ILMN_9425	ZFP36L1	46718	531	46718	531	531	1	1
ILMN_7722	ZNF264	46718	65	46718	65	65	1	1
ILMN_15703	ZNF493	46718	89	46718	89	89	1	1

	Regulon	Universe.Size	Regulon.Size	Total.Hits	Expected.Hits	Observed.Hits	Pvalue	Adjusted.Pvalue
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
ILMN_13294	ZNF91	46718	471	46718	471	471	1	1

After which we plotted the graphs for the above table:



The bottom plot shows the running enrichment score for the gene set as the analysis walks down the ranked list. The score at the peak of the dotted line is the enrichment score for the gene set.

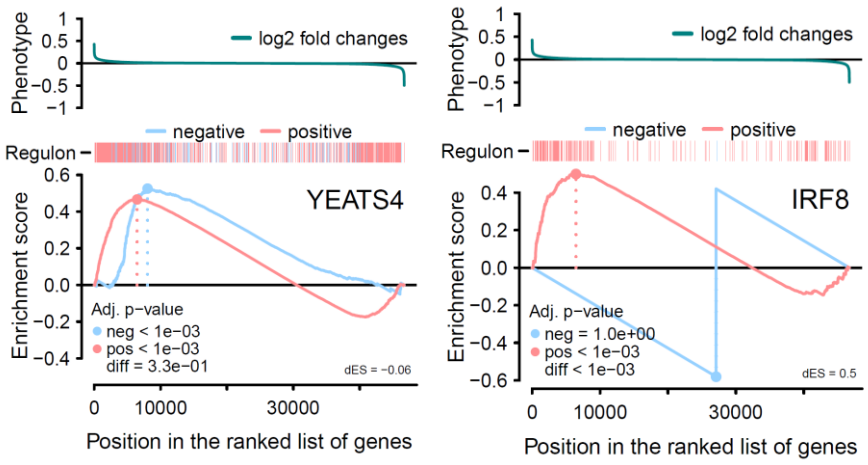
Then we proceed with **gsea2**, which is a two-tailed approach, meaning that it tests whether the regulon is positively or negatively associated with the phenotype, giving us two per-phenotype enrichment scores (ES), whose difference represents the regulon activity. The results are shown in the below tables (the following results were also obtained with a **top 1000 filter**):

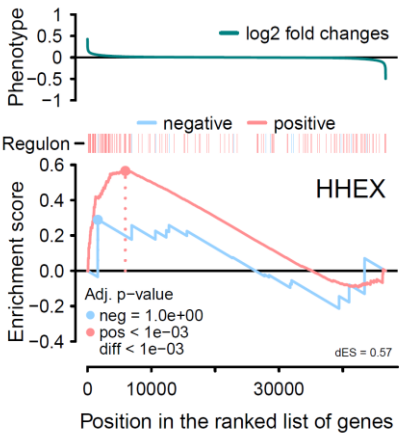
	Regulon <chr>	Regulon.Size <dbl>	Observed.Score <dbl>	Pvalue <dbl>	Adjusted.Pvalue <dbl>
ILMN_13615	ELF3	203	0.65	0.000999	0.0029970
ILMN_137681	HHEX	149	0.64	0.000999	0.0029970
ILMN_9514	HIF1A	443	0.59	0.000999	0.0029970
ILMN_20480	IRF8	233	0.64	0.000999	0.0029970
ILMN_16428	YEATS4	1115	0.62	0.000999	0.0029970
ILMN_9425	ZFP36L1	531	0.61	0.000999	0.0029970
ILMN_13294	ZNF91	471	0.60	0.000999	0.0029970
ILMN_12494	E2F3	100	0.67	0.001998	0.0052448
ILMN_5194	IRF7	347	0.59	0.003996	0.0093240
ILMN_3307	BATF	29	0.73	0.010989	0.0230770

	Regulon <chr>	Regulon.Size <dbl>	Observed.Score <dbl>	Pvalue <dbl>	Adjusted.Pvalue <dbl>
ILMN_8384	PRDM1	34	0.72	0.014985	0.0286080
ILMN_10504	HCLS1	38	0.68	0.024975	0.0437060
ILMN_21636	CITED2	37	0.68	0.035964	0.0580960
ILMN_27029	CEBPA	19	0.69	0.075924	0.1138900
ILMN_28405	NR4A2	28	0.65	0.124880	0.1748300
ILMN_11739	IRF1	19	0.64	0.206790	0.2714200
ILMN_7722	ZNF264	65	0.55	0.340660	0.4208100
ILMN_15703	ZNF493	89	0.51	0.545450	0.6363600
ILMN_22264	INSM1	30	0.50	0.688310	0.7607700
ILMN_6370	PAX7	102	0.46	0.879120	0.9230800

For full results, please refer to the zipped file *all_gsea_results.7z.*

The most interesting results are the following, for YEATS4, IRF8, and HHEX. We can see that one TF is most prominent: YEATS4, though it only appears in gsea2 graphs **if we do not apply a filter** – due perhaps to the fact that previous filters were too specific. We decided to ignore the filter in that situation as YEATS4 still felt important enough to keep and analyze.





Conclusion

According to the Human Protein Atlas (and UniProt), YEATS4 is: “a complex involved in transcriptional activation of select genes principally by acetylation of nucleosome histones H4 and H2A”, its tissue expression cluster resides in bone marrow, and is a prognostic marker of liver cancer. Its involvement in gene regulations is thus certain, and thanks to gsea2, we can also predict that its action is to activate OR deactivate its targets.

By running an alignment algorithm (BLAST) we can see that the sequence of YEATS4 is very well conserved through the evolution:

Meaning, YEATS4 might be of capital importance in the regulation/response against osteosarcoma.

Another interesting TF is IRF8 as it is Involved in CD8(+) dendritic cell differentiation. CD8 is also one of the example genes that has an immunological function and that was found to have a higher expression in patients without metastases after 5 years. 15 genes associated with this kind of function were differentially expressed in that way according to Buddingh and al.’s paper. (Similarly for IRF7)

HHEX is also hypothetical repressor in hematopoietic differentiation functions of the cells. It might be interesting since the papers suggest that MACs infiltrate this kind of cells, and 10 hematopoietic genes were found to have a significantly higher expression in patients without metastasis.

Overall, we have managed to use R packages from Bioconductor in order to infer a regulatory network from Microarray sequencing data. The subsequent Master Regulator analysis of this network then allowed us to plot out (with or without filters) a set of enrichment scores which gives us a relative idea of which genes might play a part in being upstream regulators of patients who showed better survival odds over 5 years. While we did not manage to specifically identify genes given as examples in the paper (CD14, HLA-DRA), we did confirm for example that HHEX, which we found was important, may play a role in hematopoietic differentiation. With more time to test other filters and further explore the functions given by the packages for construction and visualization of the networks, we might be able to further confirm the findings of the paper.

Bibliography

[1] E. P. Buddingh *et al.*, 'Tumor-Infiltrating Macrophages Are Associated with Metastasis Suppression in High-Grade Osteosarcoma: A Rationale for Treatment with Macrophage Activating Agents', *Clin Cancer Res*, vol. 17, no. 8, pp. 2110–2119, Apr. 2011, doi: 10.1158/1078-0432.CCR-10-2047.

[2] 'Gene regulatory network', *Wikipedia*. Oct. 19, 2021. Accessed: Nov. 13, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Gene_regulatory_network&oldid=1050666872

[3] 'Downregulation and upregulation', *Wikipedia*. Jun. 01, 2021. Accessed: Nov. 13, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Downregulation_and_upregulation&oldid=1026374055

[4] 'Regulation of gene expression', *Wikipedia*. Oct. 26, 2021. Accessed: Nov. 13, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Regulation_of_gene_expression&oldid=1051856995

[5] M. N. C. Fletcher *et al.*, 'Master regulators of FGFR2 signalling and breast cancer risk', *Nat Commun*, vol. 4, no. 1, p. 2464, Dec. 2013, doi: 10.1038/ncomms3464.

[6] 'GEO Accession viewer'. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL10295 (accessed Nov. 19, 2021).

[7] 'TNI-class: Class "'TNI'": an S4 class for Transcriptional Network... in RTN: RTN: Reconstruction of Transcriptional regulatory Networks and analysis of regulons'. https://rdrr.io/bioc/RTN/man/TNI-class.html (accessed Nov. 20, 2021).

[8] M. A. A. Castro *et al.*, 'Regulators of genetic risk of breast cancer identified by integrative network analysis', *Nat Genet*, vol. 48, no. 1, pp. 12–21, Jan. 2016, doi: 10.1038/ng.3458.

[9] J. Brownlee, 'A Gentle Introduction to the Bootstrap Method', *Machine Learning Mastery*, May 24, 2018. https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/ (accessed Nov. 20, 2021).

[10] A. A. Margolin *et al.*, 'ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context', *BMC Bioinformatics*, vol. 7, no. S1, p. S7, Mar. 2006, doi: 10.1186/1471-2105-7-S1-S7.

[11] 'relax: relax in RedeR: Interactive visualization and manipulation of nested networks'. https://rdrr.io/bioc/RedeR/man/relax.html (accessed Nov. 15, 2021).

[12] 'GeneCards - Human Genes | Gene Database | Gene Search'. https://www.genecards.org/ (accessed Nov. 15, 2021).