DBR: an R package for Differential binding of DNA transcription motifs.

Justin Chumbley[*], *Brandt Levitt*[†]

Abstract

We revisit the task of identifying the regulatory drivers of observed differential RNA gene expression. Our R package provides both up-to-data DNA transcription factor binding motif data and tools to correlate them with RNA transcription variation over genes. Specifically, the package "dbr", named for differential binding in R, seeks to modernize the important tool introduced in Cole et al. (2004). It is freely available at https://github.com/chumbleycode/dbr.

Introduction

Revisiting TeLiS

The goal of TeLiS (Cole et al. 2004) was to implicate some gene expression regulator - say CREB3 - in the observed pattern of gene expression over treatment groups. The approach has three steps. In step 1 we identify a sampling frame of n_g genes and label a subset of $k < n_g$ of these genes "differentially expressed"[1]. Step 2 uses external bioinformatic information to evaluate the average number of CREB3 DNA binding motifs within our subset of "differentially expressed" genes[2]. In step 3 we compare this latter with the average number of CREB3 motifs in a *random* subset or sample of k genes from the entire population of n_g genes. This assumes a parametric, frequentist hypothesis testing framework and yields a p-value about the representativeness of our k sample. This p-value reveals whether our gene set contains a suprising number of motifs, which is itself taken as circumstantial evidence that CREB3 mediated differential expression. TeLiS therefore complements and extends traditional gene-by-gene analyses with DNA information. Steps 2 and 3

[*]justin.chumbley@pm.me
[†]belevitt@email.unc.edu

[1]This is the set of genes whose differentially expression we can estimate. We therefore ignore, for example, genes with little or no variation in expression over subjects.

[2]We here define as "motif" any binding site with high affinity for either one or two transcription factors cooperatively. We use the notation a, b and $a :: b$ for motifs binding to a,b or both respectively.

exploit publically available DNA data repositories that allow us to calculate the number of binding motifs, m_{ij}, for each transcription factor j within each gene i.

Statistical assumptions implicit in TeLiS

The DNA motif data can be arranged as a gene x motif matrix $\mathbf{M} = (m_{ij})$. $\mathbf{M}$ is an $n_g \times n_b$ matrix linking our gene population $i \in \{1, 2, ..., n_g\}$ to the set of binding motifs $j \in \{1, 2, ..., n_b\}$[3].

Let ω denote our set of k "differentially expressed" genes. For example, we define ω to be the top ranking $k = 100$ differentially expressed genes. Then $\{m_{ij} : i \in \omega\}$ is the corresponding set of k motif counts for regulator j in each of these differentially expressed genes. This is simply the relevent k rows of the jth column of $\mathbf{M}$. Let $T_{\omega j} := \frac{1}{k} \sum_{i \in \omega} m_{ij}$ denote the per-gene *mean* number of motif binding sites in ω. We will use T_j as notational shorthand for this latter quantity, but the dependence on ω is implied. If T_j is "relatively large", then the gene set is relatively enriched with motif j. Intuitively, this motif is then implicated as a possible root cause of the observed differential gene expression which characterizes ω. In the sense discussed above, T_j may then be interpreted as a test of differential regulation by j.

The original parametric null distribution for T_j implicitly assumed that the number of binding sites within each of our k genes is independently and identically distributed: each being the (random) number of sites found in a single gene sampled uniformly at random with replacement from the entire gene population. We postpone the question of biological dependence between genes and note that sampling is only approximately independent or identically distributed because genes are in fact sampled without replacement from a finite sample frame. This may be relevant when the sample frame is relatively small, or in calculating statistics other than the average motif enrichment discussed above. In fact one can easily use (non-parametric) Monte Carlo sampling without replacement to calculate the null distribution of any statistic, e.g. the mean differential effect within all genes targeted by a transcription $\sum_i \theta_i [m_{ij} > 0]$, where θ_i is an estimate of the per-gene differential expression measure, m_{ij} in the motif count of regulator j in gene i and $[\cdot]$ is Iverson notation for the indicator function.

Simple random sampling without replacement from the gene population provides one possible outcome ω from the sample space $\Omega_{n_g,k}$ of $\binom{n_g}{k}$ possible gene sets of cardinality k within a gene population of size n_g. Now T_j is a random variable[4] whose null distribution is therefore functionally determined by our null hypothesis on $\Omega_{n_g,k}$. The null hypothesis we consider is equivalent to $\omega \sim U(\Omega_{n_g,k})$: i.e. the gene set is chosen uniformly

[3]Note that $m_{ij} = kp_{ij}$ where k is the total number of positions on the genome that host a regulator and p_{ij} is the share of all these sites that bind regulator j in gene i. For more details on the original parametric, and non-parametric model and implementation, see appendix.

[4]$T_j : \Omega_{n_g,k} \to \mathbb{Q}_{\geq 0}$ with domain $\Omega_{n_g,k}$ and codomain the set of non-negative rational numbers.

at random from $\Omega_{n_g,k}$. Then the chance of each an every gene set is just $1/|\Omega_{n_g,k}| = 1/\binom{n_g}{k}$, where $|\cdot|$ is the set cardinality function. Then $P(T_j = t|n_g, k) = \frac{1}{|\Omega_{n_g,k}|}\sum_{\omega\in\Omega_{n_g,k}}[T_{\omega j} = t]$, where $[\cdot]$ is the boolean indicator function. Classical tests and p-values then follow simply, for example by finding a critical value $t : P(T_j \geq t) = \alpha$ or, recalling that we have already defined a set of enrichment scores $\{T_1, T_2, ..T_k\}$, choosing $t : P(\cup_j\{T_j \geq t\}) = \alpha$ which controls the family-wise error rate, etc.

Because $|\Omega_{n_g,k}|$ is too large to conveniently enumerate, one might pursue a simple Monte Carlo approximation: sample 100000 times from $\Omega_{n_g,k}$ and determine the corresponding approximate null distribution for T_j. Computationally, we calculate 100000 samples from the null distribution as (the collumns of) $(\mathbf{Q} * \mathbf{M})/k$, where $\mathbf{M}$ is still our $n_g \times n_b$ gene x binding motif matrix and $\mathbf{Q}$ is a $10000 \times n_g$ matrix whose boolean rows each indicate one of 100000 elements sampled uniformly at random without replacement from $\Omega_{n_g,k}$. If we observe a relatively high value t_j, relative to this Monte Carlo null distribution we conclude that there is "over-representation" of a regulator in promoters of differentially expressed genes, and we report a one-tailed p-value gauging statistical significance.

Recall our two assumptions, fixed n and random sampling of gene sets ω. The former assumption is inconvenient when n is itself chosen adaptively based on the data. To avoid redefining our sample space, so that n is formally a random variable, we follow Cole et al. (2004) and simply condition all inference on n (see below for details on defining ω). The latter assumption seems strong: expression of genes is clustered or correlated, so it would seem that not all ω are equally likely as assumed by $\omega \sim U(\Omega_{n_g,k})$. Yet recall that ω is selected based on differential expression, so it is only relevant whether the gene-specific estimators of differential expression are clustered or correlated. We assume that two estimators are correlated if and only if some co-regulator is responsible for their coordinated differential expression. But then there will be no correlation or clustering between estimators under the null hypothesis, all ω will be equally likely and $\omega \sim U(\Omega_{n_g,k})$ appears defensible.

Possible extensions of TeLiS

We now discuss one limitation of TeLiS: if two regulators A and B are declared significant, A may be causally responsible for ω while B may be spurious. B simply has a similar (confounded) motif count pattern across genes to A. TeLiS does not offer such adjustments. We propose an alternative which both reduces these false positives - due to this confounding - and false negatives, due to insufficient power. The latter is possible by pooling data over *all* n_g genes in the gene population - not just the k genes in TeLiS - into the estimation of a single scalar parameter.

To do this we relax the requirement that genes be first labeled as DE (or not): this classification is heuristic and looses information. We instead attempt to predict variation in the estimated differential expression θ_i across *all* n_g genes in the gene population. For a fixed our regulator of interest j, we correlate $j's$ motif count over genes m_{ij} with the estimated gene-by-gene DE estimates θ_i. This can be achieved via some non-parametric or linear regression model $E(\theta_i|m_{ij}) = a + b_j m_{ij}$. The parameter b_j amounts to an interaction term, wherein the effect of some exposure on gene expression itself depends on the motif density, i.e. larger in genes with more motifs for j. From a multilevel modeling perspective, the parameter b_j might also be viewed as a second level regression parameter controlling the effect of motif density on the (first level) gene-specific random effects which relate between-subject exposure variation to between-subject variation in the expression of the specific gene. The regression function for subject s of gene i on exposure x_s is $E(y_{is}|x_s, z_s) = \alpha + \theta_i x_s + C$, with $C = \sum_k b_{ki} z_{ks}$ the linear effect of some covariates $\{z_k\}$. One benefit of a linear approach is that we can easily infer the partial effect of transcription factor j on differential expression, adjusting for the potentially confounding effect of other transcription factor(s). This is just $E(\theta_i|M) = a + \sum_j b_j m_{ij}$. This may help avoid falsely attributing differential expression to one regulator j when there exist other canditates with a similar/confounded motif counts over genes. Such motif covariation can be directly inspected in this linear framework, while it is harder to study in any approach based on preselecting genes (see appendix). Note also the increased power: this approach pools data over all genes and subjects into the estimation of a single scalar parameter b_j.

Modern TeLiS implementation

We have written the "dbr" software package to assess differential binding in R. As suggested above, the goal of dbr is to implicate a gene regulator - typically an upstream transcription factor - in the differential RNA expression observed between treatment groups. We then say that there is "differential binding" (DB) of the regulator over treatments. In practice, dbr asks whether the pattern of differential RNA expression over genes reflects (the per-gene count of DNA binding-site motifs for) some upstream gene regulator.

In addition to the raw gene-by-motif count matrices, the package currently provides some functions to augment the popular limma package. Our reimplementation of TeLiS incorporates the most up-to-date motif binding data, offers a non-parametric version of TeLiS (when the sampling frame of genes is small), and provides smooth compatibility with limma. The package is available at https://github.com/chumbleycode/dbr. It includes the possible extensions of TeLiS which we have discussed above.

This new functionality that aims to eschews the need to heuristically categorize genes, prior to DB analysis

proper, as differentially expressed or not. The simplest - cheap and cheerful - approach is to simply regress gene-specific DE estimates on gene-specific binding-site counts over the entire relevant genome (genes for which it is possible to estimate DE over treatment or exposure groups). This approach will be validated and extended to multilevel modeling.

Installing dbr and viewing DNA binding matrices

You can install dbr on from the R console with:

```
install.packages("devtools")
devtools::install_github("chumbleycode/dbr")
library(dbr)
```

There are currently three TFBM matrices: utr1, exonic1, exonic_utr1. Type "utr1" etc into the R console to see these. Get more info for each via ?utr1, ?exonic1, etc. Look for your DNA regulatory motifs of interest in the columns of these matrices. For example, recent literature has examined "a pre-specified set of TFs involved in inflammation (NF-kB and AP-1), IFN response (interferon-stimulated response elements; ISRE), SNS activity (CREB, which mediates SNS-induced b-adrenergic signaling), and glucocorticoid signaling (glucocorticoid receptor; GR)." In biomart nomenclature, "NF-kB" is is identified with NFKB1 or NFKB2. AP-1 is called JUN. ISRE is identified with the set of motifs including IRF2, IRF3, IRF4, 5, 7, 8, 9. CREB is identified with CREB3 or CREB3L1. GR is called NR3C1. This leaves us with 13 regulators plus one complex CEBPG::CREB3L1 (CEBPG_CREB3L1), as follows. Examine the gene-by-motif count matrices in the R console with:

```
immune_tfbms = c("CEBPG_CREB3L1", "CREB3", "CREB3L1", "IRF2", "IRF3",
                 "IRF4", "IRF5", "IRF7", "IRF8", "IRF9", "JUN", "NFKB1",
                 "NFKB2", "NR3C1")
utr1[, immune_tfbms] # the gene-by-motif matrix for immune motifs of interest
```

A simple analysis

We examine DB of some immune regulators amoung people with early-life stress (relative to unstressed) using data from Cole et al. (2016). Such analyses generally have two steps, first differential expression then differential binding.

DE

1. Differential expression (DE): Estimate differential RNA expression across exposure groups. Here we use a linear model: the exposure must currently be a single column of "design" matrix of this linear model (dbr cannot currently handle treatments defined across multiple collumns, e.g. factors with many levels).
2. Differential binding (DB): Infer dependence of the above, per-gene, estimates on the binding-site count of some regulator(s) of interest.

```
# Load packages
library(tidyverse)
library(limma)


# Download open source data then specify gene-by-gene regression model
dat = GEOquery::getGEO("GSE77164")[[1]]


# Specify whole-genome regression of rna on design
y <- dat %>% Biobase::exprs()
X <- dat %>%
  Biobase::pData() %>%
  select(age = `age:ch1`,
         soldier = `childsoldier:ch1`,
         edu = `educationlevel:ch1`)
X <- model.matrix(~ soldier + edu + age, data = X)


# Estimate DE using standard limmma/edger pipeline.
ttT <-
  lmFit(y, X) %>%
  eBayes %>%
  tidy_topTable(of_in = "soldier1") # "soldier1" is one column of X
```

DB

Perhaps the simplest DB analysis is just a regression of gene-wise DE estimates on motif site count. This is an approximation to a full multilevel model.

```
# regress DE on one motif of interest
summary(lm(logFC ~ NR3C1, data = append_db(ttT)))


# Or, use dbr to regress logFC on motif site count on all immune_motifs: beware multiple testing
ttT %>%
  infer_db(which_tfbms = immune_tfbms) %>%
  extract_db
```

Here p_uni is the univariate p-value from a set of simple univariate regressions of logFC on each motif, and p_cov is the corresponding p-value from a multivariate regression. The latter relates the partial or conditional relation between a motif and DE, adjusting for the remaining motifs. Any NA's in the output for this column reflect colinearities in the design matrix (i.e. motifs are too highly related to be individually estimated).

The traditional TeLiS approach

This approach requires that we first filter some genes to label as categorically DE, e.g. those with a high logFC. This filtering is not, in itself, a statistical inference. We give three examples of how to do this below.

```
# 1. genes showing > 20% difference in expression
# (Recalling that logFC is the estimated log2-fold-change of our effect)
ttT_sub = filter(ttT, logFC >= log2(1.2))


# 2. top and bottom deciles (most extreme 20%)
ttT_sub = filter(ttT, ntile(logFC, 10) %in% c(1,10))


# 3. genes whose uncorrected p-values below 0.05 (not an inference):
ttT_sub = filter(ttT, P.Value <= 0.05)
```

Having chosen one of these, or defined your own, the filtered gene-subset enters as the first argument to infer_db() below, like so:

```
ttT %>%
  infer_db(ttT_sub        = ttT_sub,
           which_tfbms    = immune_tfbms) %>%
```

```
  extract_db
```

Here "p_par" is the 2 sided p-value for parametric TeLiS (par_p_over and par_p_under are the corresponding one-tailed values for over and motif-underrepresentation). If perm_telis = TRUE, then p_npar will give a (computationally costly) permutation p-value for TeLiS.

Chi-squared test

Other ways to relate DE labels to transcription factor motif, e.g. CREB3 motif. Having categorized genes as "DE" or not, we can examine the relation between this label and the motif count as follows, for example.

```
# Chi-squared
append_db(ttT,ttT_sub = ttT_sub) %>%
  select(gene_subset, CREB3) %>%
  table %>%
  chisq.test
```

Appendix

The TFBM matrix

The RNA expression data of interest is the set of genes for which we have a meaningful estimate of differential RNA expression between exposure groups. The DNA binding motif loci data of interest is the set of loci in open chromatin which host any motif targeted by at least one known whole-blood factor[5]. These loci can be further divided according to the exact region of the gene in which the loci resides: in the start region, in some exonic region, or both.

In this work we only require that some part of the regulator binding motif j overlaps with the 1000 base pairs upstream of gene $i's$ transcription start site, yet other conditions might be equally applied (such as whether the regulator is in an exon).

[5]'Whole blood factor' in this context refers to the fact that we only considered transcription factor binding sites for transcription factors that are expressed in cells found in whole blood. 'Whole blood' is blood containing plasma and cells. The cell types we included were ('white','unclassifiable (Cell Type)','T-lymphocyte','platelet','natural killer cell','monocyte','mast cell','macrophage','lymphocyte','leukocyte','dendritic cell','B-lymphocyte'). Some transcription factors are not found in any of these cells so we reasoned that it didn't make sense to see if they were regulating gene expression patterns in these cells.

REFERENCES

Cole, Steve W, Weihong Yan, Zoran Galic, Jesusa Arevalo, and Jerome A Zack. 2004. "Expression-Based Monitoring of Transcription Factor Activity: The Telis Database." *Bioinformatics* 21 (6). Oxford University Press: 803–10.