TargetScore: Infer microRNA targets using microRNA-overexpression data and sequence information (**?**)

Yue Li
yueli@cs.toronto.edu

May 3, 2016

1 Introduction

MicroRNAs (miRNAs) are known to repress gene expression in mammalian species by forming Watson-Crick (WC) base-pairing to the $3'$ UTR regions of the target mRNA transcripts (**?**). The binding primarily occurs at the 2-7 nucleotide (nt) positions from the $5'$ end of the miRNA, which is termed as the "seed" and the binding as the "seed match" (**?**). MicroRNA regulations have been implicated in numerous developmental and pathogenic processes (**?**). Functional characterization of miRNAs depends on precise identification of their corresponding targets. However, accurate identification of miRNA targets remains a challenge with the current state-of-the-art algorithms achieving less than 50% specificity and having poor agreement among them (**??**). On the other hand, overexpression of miRNA coupled with expression profiling of mRNA by either microarray or RNA-seq has been recently developed (**??**). Consequently, genome-wide comparison of differential gene expression holds a new promise to elucidate the global impact of a specific miRNA regulation without solely relying on sequence information.

We demonstrated that target prediction can be improved by integrating expression fold-change and sequence information such as context score and other orthogonal sequence-based features such as probability of conserved targeting (PCT) (**?**) into a probabilistic score (manuscript under peer-review). Our approach differs from previous expression-based target prediction methods in three important aspects. First, our method is able to identify condition-specific miRNA targets. In contrast, well-known methods such as GenMiR++ (**?**) and GroupMiR (**?**) are based on miRNA-target expression correlation, which requires a large set of the expression profiles measured across various tissues or sample conditions. Second, the proposed method is an unsupervised method such that it does not require training data unlike some other (regression-based) methods (**???**). Third, our method operates on the entire gene set to more closely model the overall likelihood rather than only on a pre-filtered set of genes above some arbitrary cutoffs such as TargetScan score (**?**) or sample variance (**?**).

2 *TargetScore* overview

We describe a novel probabilistic method to miRNA target prediction problem using miRNA-overexpression data and sequence scores **?**. As an overview, each score feature is considered as an independent observed variable as input to a Variational Bayesian-Gaussian Mixture Model (VB-GMM). Bayesian is chosen over maximum-likelihood approach to avoid overfitting. Specifically, given expression fold-change (due to miRNA transfection), we use a three-component VB-GMM to infer down-regulated targets accounting for genes with little or positive fold-change (due to off-target effects (**?**)). Otherwise, two-component VB-GMM is applied to unsigned sequence scores. The parameters for the VB-GMM are optimized using Variational Bayesian Expectation-Maximization (VB-EM) algorithm. Presumably, the mixture component with the largest absolute means of observed negative fold-change or sequence score is associated with miRNA targets and denoted as "target component". The other components correspond to the "background component". It follows that inferring miRNA-mRNA interactions most likely explained by the observed data is equivalent to inferring the posterior distribution of the target component given the observed variables. The targetScore is computed as the sigmoid-transformed fold-change weighted by the averaged posteriors of target components over all of the features. Please refer to the manual for specific paramter settings:

```
> library(TargetScore)

> ?TargetScore
```

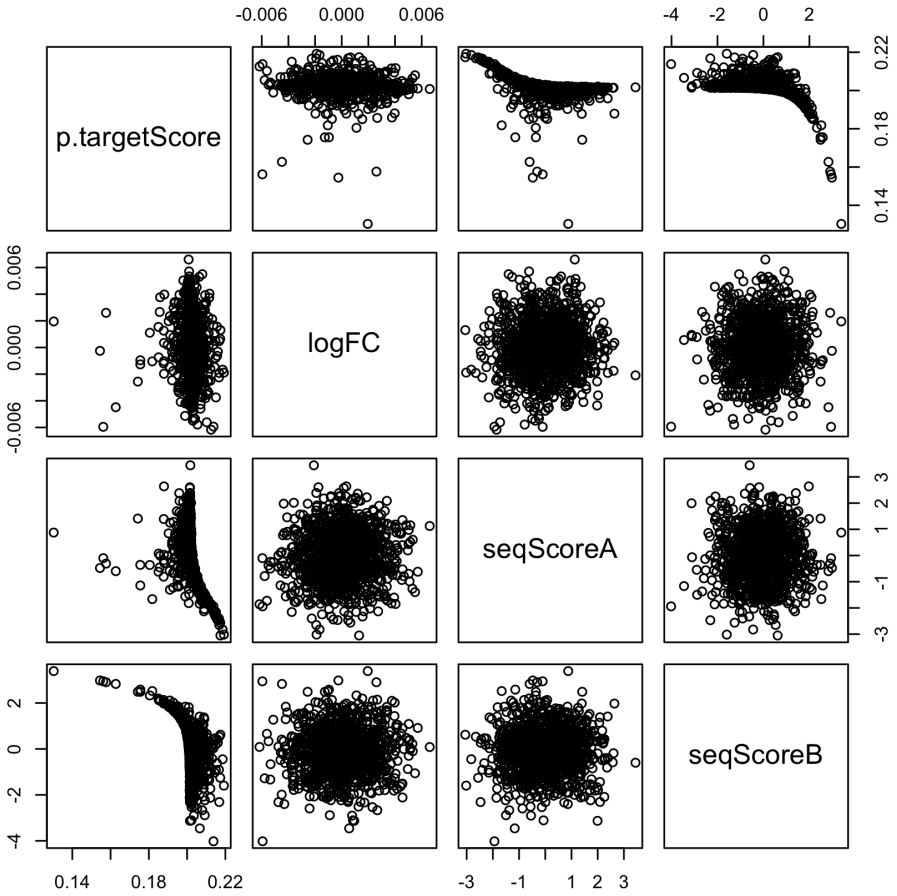
3 TargetScore input data

TargetScore operates on overexpression logarithmic fold-changes (logFC) and (optionally) sequence-based scores. In particular, gene expression in treatment (miRNA overexpressed) and (mock) control can be measured by either normalized signal intensities from microarray (e.g., hsa-miR-1 overexpressed in HeLa vs HeLa treated with mock control; GEO accession: GSE11968)or RPKM (reads per kilobase of exon per million mapped reads) by RNA-seq (e.g., hsa-miR-23b from GSE37918). The logFC for each gene is then defined as log(treatment) - log(control). The sequence scores vary depending on the specific sequence-based predictors used. For instance, user can download the pre-computed TargetScan context+ score and PCT (probabilities of conserved targeting) from TargetScan website (`http://www.targetscan.org/cgi-bin/targetscan/data_download.cgi?db=vert_61`).

4 A toy example

Suppose we overexpress an miRNA miR-X in HEK293 and measured the expression level of 1100 genes in the (un)treated cell. Comparing with control, 10 genes are down-regulated, 1000 unchanged, 90 up-regulated (possible due to off-target effects) in the transfected HEK293. In addition, the8 and 20 genes have sequence score A and B sampled from normal distribution with larger negative means (i.e. more likely to be a target). Our task to infer which genes out of the 1100 genes are targets of miRNA miR-X.

```
> trmt <- c(rnorm(10,mean=0.01), rnorm(1000,mean=1), rnorm(90,mean=2)) + 1e
> ctrl <- c(rnorm(1100,mean=1)) + 1e3
> logFC <- log2(trmt) - log2(ctrl)
> # 8 out of the 10 down-reg genes have prominent seq score A
> seqScoreA <- c(rnorm(8,mean=-2), rnorm(1092,mean=0))
> # 10 down-reg genes plus 10 more genes have prominent seq score B
> seqScoreB <- c(rnorm(20,mean=-2), rnorm(1080,mean=0))
> seqScores <- cbind(seqScoreA, seqScoreB)
> p.targetScore <- targetScore(logFC, seqScores, tol=1e-3)
> # plot relation between targetScore and input variables
> pairs(cbind(p.targetScore, logFC, seqScoreA, seqScoreB))
```



5 Real data on hsa-miR-1 and hsa-miR-17

We are now going to apply TargetScore to predict targets of miRNA hsa-miR-1 and hsa-miR-17 (GEO accession: GSE20745). For hsa-miR-1, we chose from multiple studies on GEO the microarray data that correlate the most with the validated targets (**?**). More details on the data

processing and elaborate testing are described in the manuscript (in peer-review).

```
> extdata.dir <- system.file("extdata", package="TargetScore")
> # load test data
> load(list.files(extdata.dir, "\\.RData$", full.names=TRUE))
> myTargetScores <- lapply(mytestdata, function(x) targetScore(logFC=x$logF
> names(myTargetScores) <- names(mytestdata)
> valid <- lapply(names(myTargetScores), function(x) table((myTargetScores[
> names(valid) <- names(mytestdata)
> # row pred and col validated targets
> valid

$`hsa-miR-1`

       FALSE  TRUE
  FALSE 16976    12
  TRUE   2053   136

$`hsa-miR-17`

       FALSE  TRUE
  FALSE 17762    53
  TRUE   1229   133
```

6 TargetScoreData

To automate the pipeline of calculating targetScore, we precomplied and processed to genergate miRNA-overexpression fold-changes from 84 Gene Expression Omnibus (GEO) series corresponding to 6 platforms, 77 human cells or tissues, and 112 distinct miRNAs. The end result is a data package: *TargetScoreData*. To our knowledge, this is by far the largest miRNA-pertubation data compendium. Accompanied with the data, we also included in this package the sequence feature scores from TargetScanHuman 6.1 including the context+ score and the probabilities of conserved targeting for each miRNA-mRNA interaction. Thus, the user can use these static sequence-based scores together with user-supplied tissue/cell-specific fold-change due to miRNA overexpression to predict miRNA targets using *TargetScore*.

As a convenience function, we included the package a wrapper function called `getTargetScores` that does the following: (1) given a miRNA ID, obtain fold-change(s) from logFC.imputed matrix or use the user-supplied fold-changes; (2) retrives TargetScan context score (CS) and PCT (if found); (3) obtain validated targets from the local mirTarBase file; (4) compute targetScore. We apply `getTargetScores` function using miRNA hsa-miR-1, which we know has all three types of data, namely logFC, targetScan context score, and PCT.

```
> library(TargetScoreData)
> library(gplots)
> myTargetScores <- getTargetScores("hsa-miR-1", tol=1e-3, maxiter=200)
```

```
> table((myTargetScores$targetScore > 0.1), myTargetScores$validated) # a v
> # obtain all of targetScore for all of the 112 miRNA (takes time)
>
> logFC.imputed <- get_precomputed_logFC()
> mirIDs <- unique(colnames(logFC.imputed))
>
> # targetScoreMatrix <- mclapply(mirIDs, getTargetScores)
>
> # names(targetScoreMatrix) <- mirIDs
```

7 Use *limma* package to compute logFC as input to TargetScore

TargetScore can be used easily as a downstream pipeline from a well-established differential analysis package such as *limma* (**?**). The following code provides a tutorial to obtain targetScore for hsa-miR-205. A brief workflow: (1) download the overexpression data from GEO using `getGEO` from (**?**); (2) compute logFC using functions from *limma*; (3) obtain targetScore using function `getTargetScores`.

```
> # Demo using limma
> # download GEO data for hsa-miR-205
> library(limma)
> library(Biobase)
> library(GEOquery)
> library(gplots)
> gset <- getGEO("GSE11701", GSEMatrix =TRUE, AnnotGPL=TRUE)
> if (length(gset) > 1) idx <- grep("GPL6104", attr(gset, "names")) else id
> gset <- gset[[idx]]
> geneInfo <- fData(gset)
> x <- normalizeVSN(exprs(gset))
> pData(gset)$title
> design <- model.matrix(~0+factor(c(1,1,1,1,2,2,2,2)))
> colnames(design) <- c("preNeg_control", "miR_205_transfect")
> fit <- lmFit(x, design)
> #contrast
> contrast.matrix <- makeContrasts(miR_205_transfect-preNeg_control,levels=
> fit2 <- contrasts.fit(fit, contrast.matrix)
> fit2 <- eBayes(fit2)
> limma_stats <- topTable(fit2, coef=1, number=nrow(fit2))
> limma_stats$gene_symbol <- geneInfo[match(limma_stats$ID, geneInfo$ID), "
> logFC <- as.matrix(limma_stats$logFC)
> rownames(logFC) <- limma_stats$gene_symbol
> # targetScore
> myTargetScores <- getTargetScores("hsa-miR-205", logFC, tol=1e-3, maxiter
```

Using the validated targets for hsa-miR-205, we can now evaluate the sensitivity and specificity of using targetScore and p.value from limma based on receiver operating characteristic (ROC) curve as follows.

```
> library(ggplot2)
> library(scales)
> library(ROCR)
> # ROC
> roceval <- function(myscores, labels_true, method) {
+
+         pred <- prediction(myscores, labels_true)
+
+         perf <- performance( pred, "tpr", "fpr" )
+
+         auc <- unlist(slot(performance( pred, "auc" ), "y.values"))
+
+         fpr <- unlist(slot(perf, "x.values"))
+
+         tpr <- unlist(slot(perf, "y.values"))
+
+         cutoffval <- unlist(slot(perf, "alpha.values"))
+
+         rocdf <- data.frame(x= fpr, y=tpr, cutoff=cutoffval, auc=auc, meti
+                 methodScore=sprintf("%s (%s)", method, percent(auc)), curv
+
+         return(rocdf)
+ }
> limma_stats$p.val <- -log10(limma_stats$P.Value)
> limma_stats$p.val[logFC > 0] <- 0
> myeval <- rbind(
+   roceval(myTargetScores$targetScore, myTargetScores$validated, "TargetSco
+         roceval(limma_stats$p.val, myTargetScores$validated, "Limma"))
> ggroc <- ggplot(myeval, aes(x=x, y=y, color=methodScore)) +
+
+         geom_line(aes(linetype=methodScore), size=0.7) +
+
+         scale_x_continuous("False positive rate", labels=percent) +
+
+         scale_y_continuous("True positive rate", labels=percent) +
+
+         theme(legend.title= element_blank())
> print(ggroc)
>
```

8 Use *DESeq* package to compute logFC as input to TargetScore

We can also combine *DESeq* (**?**) with targetScore for RNA-seq miRNA-overexression data analysis as the following simulated tests.

```
> library(DESeq)
> cds <- makeExampleCountDataSet()
> cds <- estimateSizeFactors( cds )
> cds <- estimateDispersions( cds )
> deseq_stats <- nbinomTest( cds, "A", "B" )
> logFC <- deseq_stats$log2FoldChange[1:100]
> # random sequence score
> seqScoreA <- rnorm(length(logFC))
> seqScoreB <- rnorm(length(logFC))
> seqScores <- cbind(seqScoreA, seqScoreB)
> p.targetScore <- targetScore(logFC, seqScores, tol=1e-3)
```

9 Summary

TargetScore is a flexbile package that takes logFC and sequence scores as optional inputs to compute the probability of each gene being the target of the overexpressed miRNA. User can compute logFC using the existing packages such as *limma* and *DESeq* as demonstrated above. The function `getTargetScores` provides user a convenient way to obtain targetScore with either pre-computed input (logFC, targetScan context score, and PCT) or supplied logFC.

10 Session Info

```
> sessionInfo()

R version 3.3.0 RC (2016-04-26 r70550)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] TargetScore_1.10.0 Matrix_1.2-6       pracma_1.8.8

loaded via a namespace (and not attached):
[1] tools_3.3.0     grid_3.3.0      lattice_0.20-33
```