Transrate: exploring expression-dependent contig score penalty

Richard Smith-Unna

23 November 2015

Contents

Creating a test dataset

To explore the effect of different contig penalty schemes, we create a test dataset. It consists of four sets of transcripts, each representing a toy transcriptome with four transcripts. The contig scores are randomised across the entire range of possible scores. The transcript expression profile is dominated by a few transcripts.

```r
library(data.table)
library(dplyr)

# n contigs in reps assemblies
n <- 7
reps <- 100
total <- n * reps

# differentiate the contigs and replicates
id <- 1:total
set <- rep(1:reps, each = n)

# expression distribution dominated by a few contigs
expr_single_raw <- c(0.01, 0.01, 0.02, 0.1, 0.5, 0.99, 4)
expr_single_norm <- expr_single_raw / sum(expr_single_raw)
expr <- rep(expr_single_norm, reps)

# uniform sample scores
raw_score <- runif(total)

# calculate the cdf
cdf_single <- c(0, cumsum(expr_single_norm[1:length(expr_single_norm)-1]))
cdf <- rep(cdf_single, reps)

table <- data.table(
  id = id,
  expr = expr,
  raw_score = raw_score,
  cdf = cdf
)
```

Developing the penalty functions

We want to take the existing compositional error (raw) score and add a penalty. The penalty should penalise contigs that are highly expressed and poorly assembled, but never penalise contigs with low expression.

We can capture the ideal outcome in a table:

Expression	Original score	Penalty	Adjusted score
low	high	none	high
low	medium	none	medium
low	low	none	low
medium	high	none	medium
medium	medium	low	medium
medium	low	medium	low
high	high	low	high
high	medium	medium	low
high	low	high	low

The penalty must thus proportional to the raw score, and to some function of the expression of the contig. We always want the score to remain between 0 and 1, so the penalty is constrained to the difference between the raw score and 1.

Here we consider two penalty functions that differ in how they incorporate contig expression.

- `penalise_by_expr` in which the penalty is proportional to the relative contig abundance
- `penalise_by_cdf` in which the penalty is proportional to the sum of relative expression for all lower-expressed contigs

```
# penalise by the relative abundance of the contig
penalise_by_expr <- function(txp_data) {
  raw <- txp_data$raw_score
  expr <- expr
  dist <- (1 - raw)
  penalty <- dist * raw * expr
  return(1 - (raw + penalty))
}

# penalise by cumulative sum of contig relative abundance
penalise_by_cdf <- function(txp_data) {
  raw <- txp_data$raw_score
  cdf <- txp_data$cdf
  dist <- (1 - raw)
  penalty <- dist * raw * cdf
  return(1 - (raw + penalty))
}

table$'Expression penalty' <- penalise_by_expr(table)
table$'CDF penalty' <- penalise_by_cdf(table)
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

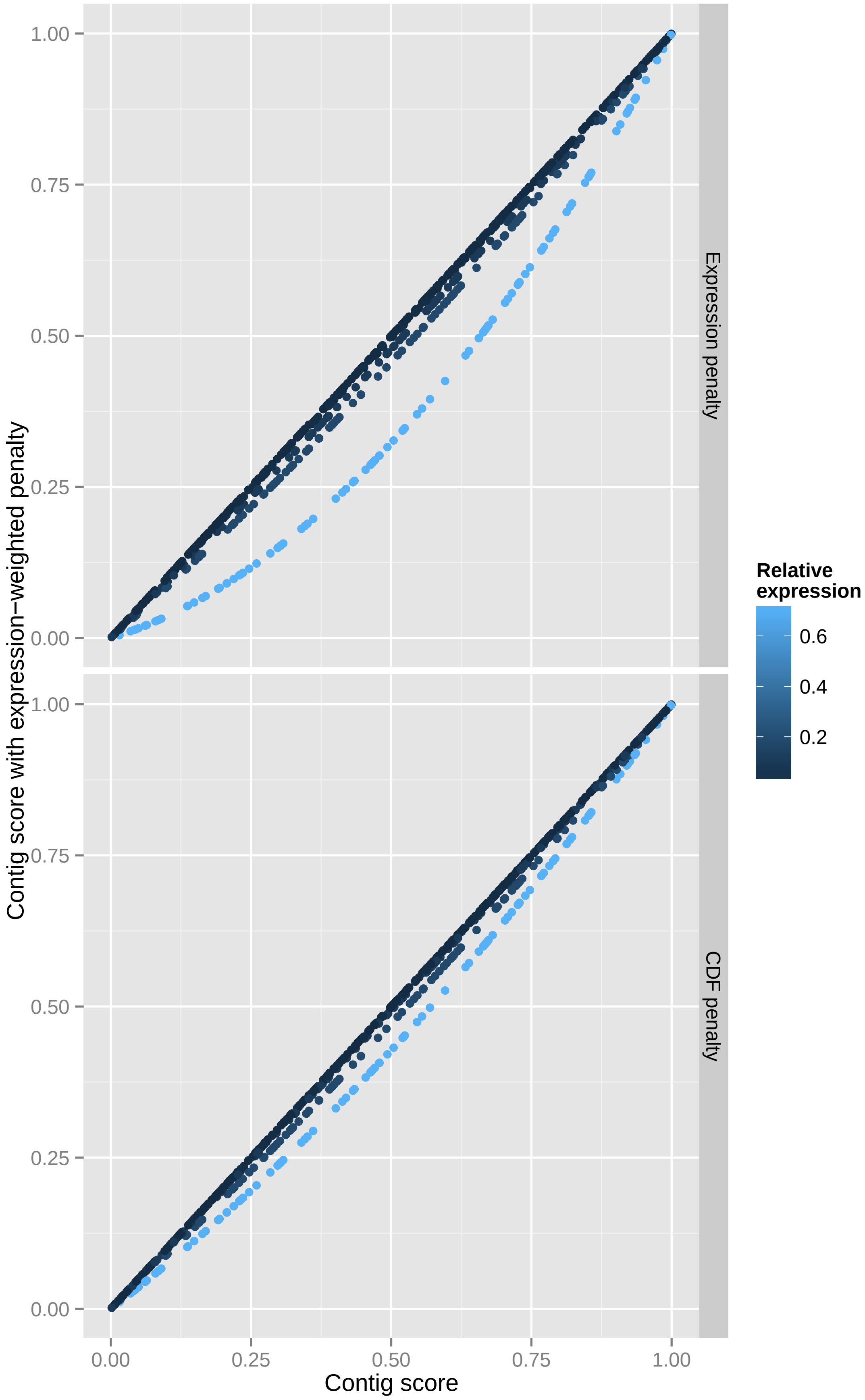
```
table_long <- melt(table, id.vars = names(table)[1:4])
```

Visualising the effect of the penalty

We compare the two functions to the original transrate contig score, colouring the contigs by expression. It can be seen that dominant contigs are penalised more in the first scheme, based on relative abundance.

```
library(ggplot2)
library(grid)
library(gtable)

ggplot(table_long,
       aes(x=value, y=1-raw_score, colour=expr)) +
  geom_point() +
  facet_grid(variable~.) +
  ylab('Contig score') +
  xlab('Contig score with expression-weighted penalty') +
  coord_flip() +
  scale_colour_continuous() +
  guides(colour = guide_colourbar(title='Relative\nexpression'))
```


Contig score with expression−weighted penalty
1.00
0.75
0.50
0.25
0.00
Expression penalty
CDF penalty
0.00
0.25
0.50
0.75
1.00
Contig score
Relative expression
0.6
0.4
0.2