Examples on How to Visualize the Results of P2C2M

Michael Gruenstaeudl

January 28, 2015

This vignette provides two examples of how the results of the R package **P2C2M** can be visualized using ggplot2 [1].

1 Visualization Examples

1.1 The Effects of Simulation Replication

Simulation replication is used to control for high levels of coalescence stochasticity and, by extension, to reduce error rates. In order to assess the effect of simulation replication on the test distributions generated by P2C2M, a comparison between two test distributions generated under different numbers of simulation replicates is conducted.

Example data is loaded into R. An alpha value is set. Several lists are initialized. Gene names are specified.

```
> require(P2C2M)
> require(ggplot2)
> require(grid)
> data(viz_example_1)
> inp = viz_example_1
> alpha = 0.05
> inData = qnts = df = titles = list()
> df$lwr = df$upr = list()
> titles = sprintf("gene%02d", c(1:10))
> colnames(inp$nReps0) = titles
> colnames(inp$nReps10) = titles
```

The set of the example data calculated without simulation replication is converted into a stacked format. Upper and lower quantiles are calculated.

```
> inData$nReps10 = stack(as.data.frame(inp$nReps10))
> inData$nReps10[,3] = "nReps10"
> colnames(inData$nReps10) = c("value", "gene", "n_reps")
> qnts$nReps10 = apply(inp$nReps10, 2, quantile, c(alpha, 1-alpha), na.rm=TRUE)
> df$lwr$nReps10 = data.frame(lwrQntl=qnts$nReps10[1,], gene=names(qnts$nReps10[1,]),
+                              n_reps=rep("nReps10", 10))
> df$upr$nReps10 = data.frame(uprQntl=qnts$nReps10[2,], gene=names(qnts$nReps10[2,]),
+                              n_reps=rep("nReps10", 10))
```

The set of the example data calculated with simulation replication (nReps=10) is converted into a stacked format. Upper and lower quantiles are calculated.

```
> inData$nReps0 = stack(as.data.frame(inp$nReps0))
> inData$nReps0[,3] = "nReps0"
> colnames(inData$nReps0) = c("value", "gene", "n_reps")
> qnts$nReps0 = apply(inp$nReps0, 2, quantile, c(alpha, 1-alpha), na.rm=TRUE)
> df$lwr$nReps0 = data.frame(lwrQntl=qnts$nReps0[1,], gene=names(qnts$nReps0[1,]),
+                             n_reps=rep("nReps0", 10))
```

```
> df$upr$nReps0 = data.frame(uprQntl=qnts$nReps0[2,], gene=names(qnts$nReps0[2,]),
+                            n_reps=rep("nReps0", 10))
```

Both sets of data are combined and ordered via factors.

```
> inData = rbind(inData$nReps0, inData$nReps10)
> dfLwr = rbind(df$lwr$nReps0, df$lwr$nReps10)
> dfUpr = rbind(df$upr$nReps0, df$upr$nReps10)
> inData$gene = factor(inData$gene, levels = sort(titles))
```

The distributions of differences of the summary statistic coal (Liu & Yu 2010) are visualized as a panel of density distributions.

```
> ggplot(data=inData, aes(x=value)) +
+   geom_density() +
+   facet_grid(gene ~ n_reps, scales = "free_y") +
+   labs(x="Difference values") +
+   ggtitle(expression(atop("Distributions under different N of replicates",
+                      atop(italic("Descriptive Statistic: coal (Liu & Yu 2010)"), "")))) +
+
+   theme_bw() +
+   theme(axis.text = element_text(size=5),
+         axis.title.y=element_blank(),
+         strip.text.y=element_text(angle=0),
+         panel.grid.major.x=element_blank(),
+         panel.grid.major.y=element_blank(),
+         strip.background = element_rect(fill="white"),
+         panel.margin = unit(0.5, "lines"),
+         plot.title = element_text(face="bold", size=rel(1.5), vjust=-1)) +
+   # Limits on the x-axis improve the visualization
+   xlim(-500, 500) +
+   geom_vline(xintercept=0, linetype = "dashed") +
+   geom_vline(aes(xintercept=lwrQntl), dfLwr, color="grey") +
+   geom_vline(aes(xintercept=uprQntl), dfUpr, color="grey")
```

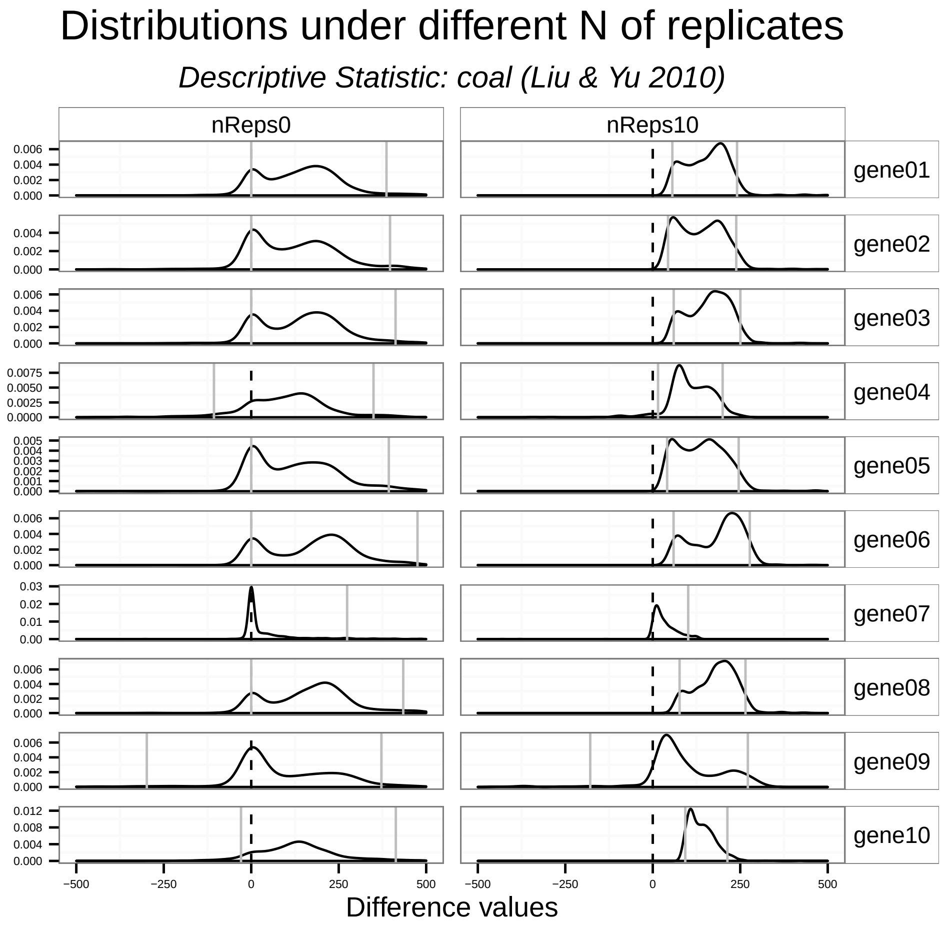

Figure 1: A panel of density distributions illustrating the effect of simulation replication on the test distributions generated by P2C2M

1.2 Visualization of the Distribution of False Positive Results

In order to visualize the sensitivity of P2C2M to false positive results among the different summary statistics, a graphical comparison is generated.

Example data is loaded into R.

```
> require(P2C2M)
> require(ggplot2)
> data(viz_example_2)
> inp = viz_example_2
```

A custom function is specified which converts result matrices into presence/absence matrices, stacks the matrix columns and adds identifier information.

```
> customfunc = function(inData, simNum){
+   handle = inData
+   colnames(handle) = c("gtp", "ray", "ndc", "gsi")
+   # Convert results into presence/absence matrix
+   handle[!grepl("n.s.", handle)] = 1
+   handle[grepl("n.s.", handle)] = 0
+   # Stack the individual descriptive statistics
+   handle = stack(data.frame(handle, stringsAsFactors=FALSE))
+   colnames(handle)[1] = "value"
+   colnames(handle)[2] = "stat"
+   # Add gene identifiers (under the assumption that there are 10 genes)
+   handle[,3] = rep(c(1:10), 4)
+   colnames(handle)[3] = "gene"
+   handle[,4] = simNum
+   colnames(handle)[4] = "sim"
+ return(handle)
+ }
```

The custom function is executed on the example data, which consists of two subsets that are characterized by different substitution rates.

```
> highL = list()
> sims = as.numeric(names(inp$High))
> for (i in 1:length(inp$High)) {highL[[i]] = customfunc(inp$High[[i]], sims[i])}
> High = do.call("rbind", highL)
> High[,ncol(High)+1] = "High_Subst_Rate"
> colnames(High)[ncol(High)] = "ratetype"
> lowL = list()
> sims = as.numeric(names(inp$Low))
> for (i in 1:length(inp$Low)) {lowL[[i]] = customfunc(inp$Low[[i]], sims[i])}
> Low = do.call("rbind", lowL)
> Low[,ncol(Low)+1] = "Low_Subst_Rate"
> colnames(Low)[ncol(Low)] = "ratetype"
> inData = rbind(High, Low)
```

The distribution of false positive results is visualized as a presence/absence plot.

```
> ggplot(data=inData, aes(x=sim,y=gene)) +
+   geom_point(aes(colour=value), size = 3) +
+   scale_colour_manual(values = c(NA,’black’)) +
+   facet_grid(stat˜ratetype) +
+   ggtitle(expression(atop("Distribution of False Positive Results",
+           atop(italic("Alpha=0.1") , "")))) +
+   theme_bw() +
+   theme(axis.text = element_text(size=5),
+         strip.background = element_rect(fill="white")) +
+   scale_x_discrete(breaks=c(1:5), labels=c(1:5)) +
+   scale_y_discrete(breaks=c(10:1), labels=c(10:1))
```

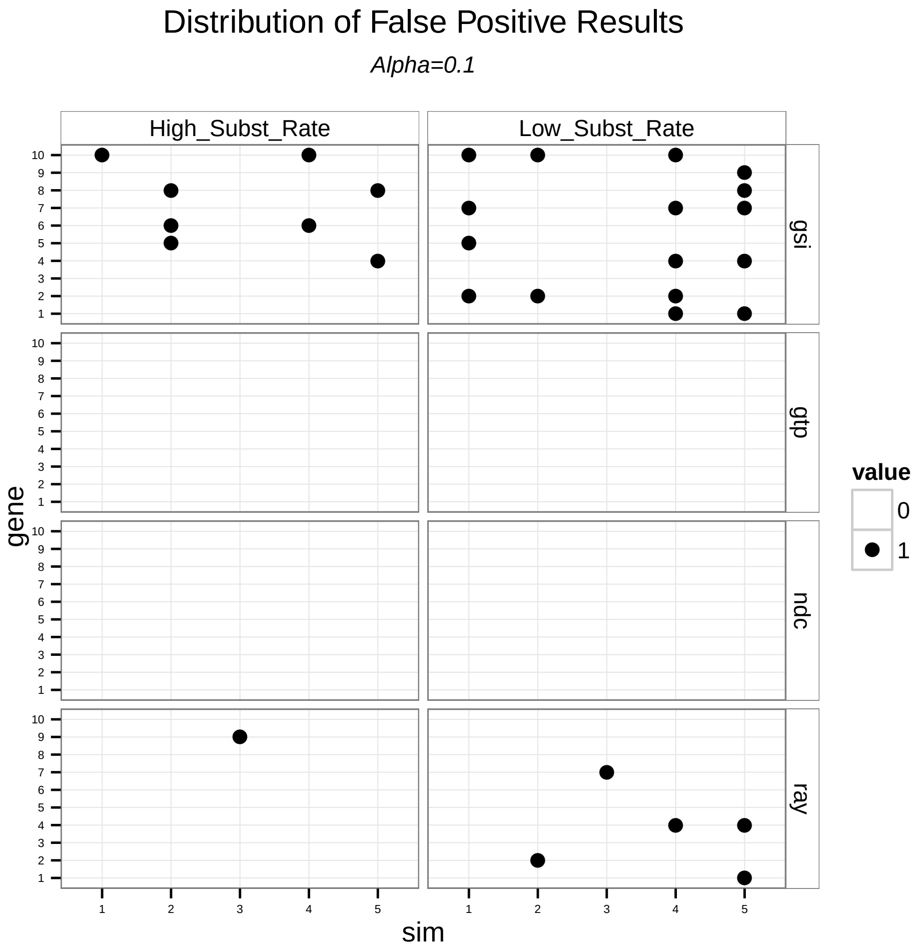

Figure 2: A presence/absence plot illustrating the sensitivity of P2C2M to false positive results among the different summary statistics

References

[1] H Wickham. *ggplot2: elegant graphics for data analysis*. Springer, New York, 2009.