coGPS:Cancer Outlier Gene Profile Sets

Yingying Wei, Michael Ochs

May 19, 2021

1 Introduction

Generation of high-throughput genomic data has become routine in modern biology. While the focus remains in many cases on the identification of molecular species, such as mRNA transcripts, that differ between two conditions, the data is often of great interest for elucidating differences in pathway activity. Standard algorithms, such as *limma* [Smyth 2004]and *SAM* [Tusher , Tibshirani and Chu 2001], are widely used for identifying transcripts (hereon referred to genes) that differ statistically between two groups. Later, as multiple datasets became available on the same biological process, people began to combine information from multiple studies to improve the power of statistical analysis in each study [Conlon, Song and Liu 2006], [Kendziorski *et al.* 2003], [Yuan and Kendziorski 2006], [Cui *et al.* 2005], and [Scharpf *et al.* 2009].

It is often the case in cancer studies that the identification of important genes by these approaches fails. The realization that genes are often disregulated in only a subset of tumors led to the development of Cancer Outlier Profile Analysis (*COPA*) [MacDonald, 2006]. Outlier genes are defined to be ones that are over- or under- expressed in a subset of tumor samples compared to normal samples. The difference between *outlier differential expressed gene* and normal differential expressed gene is that *outlier gene* may have the same level of expression in majority of tumor samples as that of normals. A few statistics for finding *outlier genes* have been proposed [MacDonald, 2006], [Tibshirani, R. and Hastie, 2007], and [Wu, 2007].

The data can also be used to elucidate pathway activity through gene set analysis, also termed gene set enrichment analysis. In this case the statistic is used to rank all genes, and the relative rank of genes within a pathway is used, typically in a Wilcoxon Rank Sum Test, to determine if the genes associated with a pathway are distributed non-randomly. The issue for cancer studies is the fact that pathways are often disregulated due to changes in different genes in different samples, making the outlier statistic of great use for ranking genes for gene set analysis. This is the focus of the work here.

In addition, we now have several types of high-throughput data available in a single study, which provides significantly more information if it can be integrated. Therefore, we are interested in capturing those genes that are over-expressed, hypo-methylated and copy number amplified in a subset of data. The subset of samples that are over-expressed for a gene may not be the same as the subset

of samples that are hypo-methylated for that gene. We call this type of outlier pattern *uniform outlier* . On the other hand, we may be interested in genes that are over-expressed in a subset of samples but behave the same as normals in methylation and CNV data. We call this type of outlier pattern *subtype outlier* . Here we follow and generalize Ghosh's approach [Ghosh 2010] of defining a statistic for outlier measure based on the p-value of tumor samples compared to the empirical distribution of gene expression for controls.

2 Method

2.1 Empirical p-values for tumors

We assume that we have totally D studies. For study d, we have N_{d0} control samples and N_{d1} tumor samples. The gene expression data are stored in a matrix, with each row representing a single gene and each column representing one sample. Therefore, the expression value for gene g is denoted as $[X_{g,1,1}, \cdots, X_{g,1,N_{10}}, X_{g,1,N_{10}+1}, \cdots, X_{g,1,N_{10}+N_{11}}, \cdots, X_{g,D,N_{D0}+N_{D1}}]$ And the total expression matrix is G by $(N_{10} + N_{11} + \cdots + N_{D0} + N_{D1})$

The idea is that we compare each observation in our tumor samples to the empirical distribution of expression values of the same gene for normal samples in the same study. Now for gene g in study d, for each expression in tumor samples we calculate the up-tail empirical p-value as

$$\hat{p}_{g,d,l} = \frac{1}{N_{d0}} \sum_{i=1}^{N_{d0}} I(X_{g,d,N_{d0}+l} \leq X_{g,d,i}) \tag{1}$$

The corresponding lower-tail empirical p-value is calculated as

$$\hat{p}_{g,d,l} = \frac{1}{N_{d0}} \sum_{i=1}^{N_{d0}} I(X_{g,d,N_{d0}+l} \geq X_{g,d,i}) \tag{2}$$

Therefore, in either case we come up with $G * (N_{11} + \cdots + N_{D1})$ matrix.

2.2 Uniform Outlier

Now for each gene, we conduct a Bonferroni correction. Setting the significance level to be α, we turn the $\hat{p}_{g,d,l}$ to be a binary matrix $\hat{m}_{g,d,l}$:

$$\hat{m}_{g,d,l} = I(\hat{p}_{g,d,l} \leq \frac{\alpha}{N_{11} + \cdots + N_{D1}}) \tag{3}$$

Finally, we sum over $\hat{m}_{g,d,l}$ for each gene and obtain the summarized statistic S_g, which measures the number of outlier samples for gene g across all studies. The idea is that we treat tumor samples from all studies equally. If there are consistent number of outlier samples for a given gene across all the studies, this gene would be identified as an outlier gene.

2.3 Subtype Outlier

Now for each gene within each study, we conduct a Bonferroni correction. Setting the significance level to be α, we turn the $\hat{p}_{g,d,l}$ to be a binary matrix $\hat{m}_{g,d,l}$

$$\hat{m}_{g,d,l} = I(\hat{p}_{g,d,l} \leq \frac{\alpha}{N_{d1}}) \tag{4}$$

Next, we sum over $\hat{m}_{g,d,l}$ for each gene within each study and obtain the summarized statistics $s_{g,d}$, which measures the number of outlier samples for gene g in study d. Finally, we set $S_g = max(s_{g,d}, d = 1, \cdots, D)$ By taking the maximum number of outlier sample numbers among the studies, we are able to capture genes that are over-expressed in a proportion of tumor samples in one study but behave exactly as normals in all the rest of studies.

3 Data preparation

In order to work with *coGPS*, we need to prepare the apporpriate data input format. The first argument `exprslist` is a list storing expression data and the corresponding class label of samples. As an example, here we have expression data, methylation data and copy number data for 25 normal people and 44 patients.

```
> library(coGPS)
> data(Exon_exprs_matched)
> data(Methy_exprs_matched)
> data(CNV_exprs_matched)
> data(Exon_classlab_matched)
> data(Methy_classlab_matched)
> data(CNV_classlab_matched)
> head(Exon_exprs_matched[,1:5])
```

```
             X1       X2       X3       X4       X5
TTLL10  6.717344 7.037852 6.767719 6.420387 6.327140
B3GALT6 6.391804 7.063906 7.088237 6.923357 6.437467
SCNN1D  6.439294 7.363256 6.960123 6.329735 6.056324
```

```
PUSL1   6.977590 7.238936 7.501508 6.716543 6.929376
VWA1    8.652309 9.483455 9.331383 8.586654 8.117134
ATAD3B  5.597828 6.759150 6.568391 6.101554 6.011183
```

```
> head(Methy_exprs_matched[,1:5])
```

```
               X1         X2         X3         X4         X5
TTLL10  0.92614513 0.92341822 0.74665674 0.89014933 0.92416166
B3GALT6 0.41558269 0.48709422 0.52353576 0.36409685 0.50031962
SCNN1D  0.58554773 0.37659892 0.15732190 0.62241624 0.68238374
PUSL1   0.06496585 0.11335588 0.04692521 0.05516385 0.05518681
VWA1    0.11405662 0.08245048 0.03221226 0.11885276 0.12259966
ATAD3B  0.03805213 0.03612167 0.03043765 0.03785675 0.03026946
```

```
> head(CNV_exprs_matched[,1:5])
```

```
              X1       X2       X3       X4       X5
TTLL10  1.876122 2.618898 1.649337 1.899147 2.413655
B3GALT6 1.297597 1.517815 2.155240 1.845882 1.113260
SCNN1D  1.987613 2.090843 2.472659 1.977962 1.927649
PUSL1   2.231360 2.085793 1.414651 2.402684 2.015025
VWA1    1.786645 2.012630 2.576627 2.050612 2.111320
ATAD3B  2.209892 1.561455 2.962057 2.001006 1.861121
```

Each element of `exprslist` is a list with the first element being `exprs` and the second element being `classlab`. Each row of `exprs` represents one gene and each column represents one sample. `exprs` should have both row names showing gene names and column names showing sample names.`classlab` is a zero-one vector indicating the status of samples. We use 0 for the baseline group, usually the normal group, and 1 for the comparison group, usually the tumor group.

```
> #exprslist[[i]]$exprs should be in matrix format
> Exon_exprs<-as.matrix(Exon_exprs_matched)
> Methy_exprs<-as.matrix(Methy_exprs_matched)
> CNV_exprs<-as.matrix(CNV_exprs_matched)
> #exprslist[[i]]$classlab should be in vector format
> Exon_classlab<-unlist(Exon_classlab_matched)
> Methy_classlab<-unlist(Methy_classlab_matched)
> CNV_classlab<-unlist(CNV_classlab_matched)
> #make an exprslist consisting 3 studies
> trylist<-list()
> trylist[[1]]<-list(exprs=Exon_exprs,classlab=Exon_classlab)
> trylist[[2]]<-list(exprs=Methy_exprs,classlab=Methy_classlab)
> trylist[[3]]<-list(exprs=CNV_exprs,classlab=CNV_classlab)
```

4 Analysis

Once we have specified the input data `exprslist`, we can apply *PCOPA* to obtain the desired statistics. Here suppose we are interested to find outlier genes that tend to have either over-expression, or hypo-methylated, or amplified copy number outlier samples. Therefore we set `side` to be `c("down","up","down")` and `type` to be `"subtype"`.

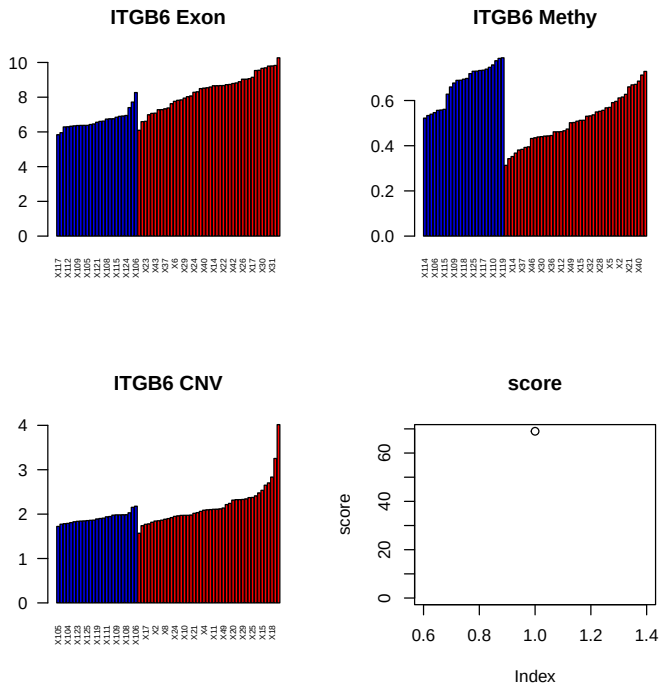
```
> a7<-PCOPA(trylist,0.05,side=c("up","down","up"),type="subtype")
```

If we are interested to find outlier genes that tend to have over-expression, hypo-methylated and amplified copy number outlier samples. Therefore we set `type` to be `"uniform"`.

```
> a8<-PCOPA(trylist,0.05,side=c("up","down","up"),type="uniform")
```

After calculating the statistics, we can use *PlotTopPCOPA* to view the expression patterns of top ranked genes. For study i, *PlotTopPCOPA* first sorts the expression value `exprslist[[i]]$exprs[j,]` among the baseline samples(e.g. normal ones) and comparison group (e.g. tumor ones)seperately for selected gene j, and then plot the sorted expression values. The first argument `exprslist` should be the same one as for *PCOPA* ; the second argument `PCOPAresult` should be an output of PCOPA; the third argument `topcut` determines how far we would go down the top ranked list; and the last argument `typelist` is a vector specifying the titles for each graph corresponds to a specific study.

```
> PlotTopPCOPA(trylist,a8,topcut=1,typelist=c("Exon","Methy","CNV"))
```



We can run permutations to obtain the p-value for PCOPA statistics.

```
> perma7<-permCOPA(trylist,0.05,side=c("up","down","up"),type="subtype",perms=2)
```

For gene specific permutation pvalue:

```
> pvaluea7<-sapply(1:length(a7),function(i)
+         length(which(perma7[i,]>a7[i]))/ncol(perma7))
```

For pvalue calculation using all genes' permutation COPA values:

```
> dista7<-as.vector(perma7)
> pvaluea7<-sapply(1:length(a7),function(i)
+         length(which(dista7>a7[i]))/length(dista7))
```

5 Downstream analysis

Now we can apply gene set enrichment analysis to the obtained PCOPA statistics.

```
> library(limma)
> data(human_c1)
> genename<-rownames(Exon_exprs)
> test_set1_a7<-rep(1,length(Hs.gmtl.c1))
> for(i in 1:length(Hs.gmtl.c1))
+ {
+         set<-Hs.gmtl.c1[[i]]
+          matched<-match(genename,set)
+         index<-is.na(matched)==FALSE
+         if(sum(as.numeric(index))>0)
+         {
+                 test_set1_a7[i]<-wilcoxGST(index,a7)
+         }
+         else
+         {
+                 test_set1_a7[i]<-NA
+         }
+ }
```

6 Patient specific outlier gene list

In clinical settings, people are extremely interested in finding the outlier gene list for each specific patient. Here our package provides such a solution. Usually we focus on only those top ranked outlier genes over all samples. In other words, we want each of the gene on our patient specific outlier gene list to be among the top *PCOPA* scored outlier genes in all samples. We allow the user to set the number of top genes. In practice, the user may pick up a specific number which she or he thinks reasonable. Or the user may first run a permutation test to get the null distribution of *PCOPA* scores and use p-values after bonferroni correction or q-values derived from the p-values to select all significant genes. Caution should be paid to the order of samples in the generation of patient specific outlier gene list. Suppose one only wants to have the *PCOPA* scores and the down stream analysis of *GSE* , no matching of samples in different data types are required. But if one wants to generate patient specific outlier gene list, one has to put all the samples in each data type in the same order. In other words, `exprslist[[i]]$exprs[,j]` should correspond to the sample j in each data type i.

```
> IndividualList7<-PatientSpecificGeneList(trylist,0.05,side=c("down","up","down"),
+ type="subtype",TopGeneNum=100)
```

The individual outlier gene list for patient 1 in exon data, methylation data and CNV data are:

```
> IndividualList7[[1]]
```

```
[[1]]
[1] "CDKN2C" "BOLA1"  "RSPO1"  "GYPC"

[[2]]
[1] "VAMP3"   "FOXD2"   "CGN"     "TBCE"    "TRIM58"  "TRAPPC3" "STX6"
[8] "CPSF3"   "PPM1B"

[[3]]
[1] "KCNAB2"  "PLEKHO1" "ZNF687"  "PSRC1"   "ID2"     "EML4"    "TCF7L1"
[8] "ATOH8"   "HNRPLL"
```

The corresponding ones for patient 33 are:

```
> IndividualList7[[33]]


[[1]]
character(0)

[[2]]
 [1] "PHF13"    "FOXE3"    "ACTN2"    "RYR2"     "C1orf101" "TRIM58"
 [7] "CSF3R"    "GRIK3"    "EPHA10"   "RGS16"    "SOX11"    "MORN2"
[13] "VAMP5"    "IMP4"     "HOXD8"    "SOS1"     "KCNG3"

[[3]]
 [1] "PLEKHO1"  "ZNF687"   "EXO1"     "CLCC1"    "PSRC1"    "TNFSF4"
 [7] "ACBD6"    "DYNC2LI1" "ATOH8"    "PTPN18"   "LRP1B"
```

References

[http://www.ncbi.nlm.nih.gov/geo/]

[Conlon, Song and Liu 2006] Conlon, E. M., Song, J. J., Liu, J.S. (2006), Bayesian models for pooling Microarray studies with multiple sources of replications. *BMC Bioinformatics* **7**, 247

[Cui *et al.* 2005] Cui, X., Hwang, J.T.G.,Qiu, J., Blades, N.J., Churchill, G.A. (2005), Improved statistical tests for differential gene expression by shrinking variance components estimates.. *Biostatistics* **6**,1, 59–75

[Kendziorski *et al.* 2003] Kendziorski, C. M.,Newton, M. A., Lan, H.,Gould, M. N. (2003), On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine* **22**, 3899–3914

[Scharpf *et al.* 2009] Scharpf, R. B.,Tjelmeland, H., Parmigiani, G.,Nobel, A. B. (2009), A Bayesian model for cross-study differential gene expression. *Journal of the American Statistical Association* **104(488)**, 1295–1310

[Smyth 2004] Smyth, G. K. (2004), Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* **3**, Art. 3.

[Tusher , Tibshirani and Chu 2001] Tusher, V.G., Tibshirani, R., and Chu, G. (2001), Significance analysis of microarrays applied to the ionizing radiation response *PNAS* **98(9)**, 5116–5121

[Yuan and Kendziorski 2006] Yuan, M., Kendziorski, C. M. (2006), A Unified Approach for Simultaneous Gene Clustering and Differential Expression Identification. *Biometrics* **62**, 1089–1098

[Ghosh 2010] Ghosh (2010), Discrete nonparametric algorithms for outlier detection with genomic data. *Journal of Biopharmaceutical Statistics*

[Tibshirani, R. and Hastie, 2007] Tibshirani, R. and Hastie (2007), Outlier sums for differential gene expression analysis. *Biostatistics* **8**, 2–8

[Wu, 2007] Wu.B (2007), Cancer outlier differential gene expression detection. *Biostatistics* **8**, 566–575

[MacDonald, 2006] MacDonald, J. W. and Ghosh, D. (2006), COPA–cancer outlier profile analysis *Bioinformatics* **22**, 2950–1