

An Introduction to *CCPROMISE*

Xueyuan Cao, Stanley Pounds

October 10, 2016

1 Introduction

CCPROMISE, Canonical correlation with PROjection onto the Most Interesting Statistical Evidence, is a general procedure to integrate two forms of genomic features that exhibit a specific biologically interesting pattern of association with multiple phenotypic endpoint variables. In biology, one type of genomic feature tends to regulate the other types. For example, DNA methylation regulates gene expression. Biological knowledge of the endpoint variables is used to define a vector that represents the biologically most interesting values for a set of association statistics. The CCPROMISE performs one hypothesis test for each gene, and is flexible to accommodate two type of genomic features with various types of endpoints.

In this document, we describe how to perform CCPROMISE procedure using hypothetical example data sets provided with the package.

2 Requirements

The CCPROMISE package extends our former PROMISE package to integrate two forms of molecular data with multiple biologically related endpoints in gene level or probe set level. The understanding of *ExpressionSet* is a prerequisite to perform the CCPROMISE procedure. Due to the internal handling of multiple endpoints, the consistency of *ExpressionSet* is assumed. The detailed requirements are illustrated below.

Load the CCPROMISE package and the example data sets: *exmplESet*, *exmplMSet*, *exmplGeneSet*, and *exmplPat* into R.

```
> library(CCPROMISE)
> data(exmplESet)
> data(exmplMSet)
> data(exmplGeneSet)
> data(exmplPat)
```

The *ExpressionSet* should contain at least two components: *exprs* (array data) and *phenoData* (endpoint data). The subject id and order of *ESet* and

MSet should be same. *exprs* is a data frame with column names representing the array identifiers (IDs) and row names representing the probe (genomic feature) IDs. *phenoData* is an *AnnotatedDataFrame* with column names representing the endpoint variables and row names representing array. The array IDs of *phenoData* and *exprs* should be matched.

The association pattern definition is critical. The prior biological knowledge is required to define the vector that represents the biologically most interesting values for statistics. In this hypothetical example, we are interested in identifying genomic features that are negatively associated with drug level to kill 50% cells, negatively associated with disease, and negatively associated with rate of events. The three endpoints are represented in three rows as shown below:

```
> exmplPat

  stat.coef      stat.func      endpt.vars
1    -0.33 spearman.rstat          LC50
2    -0.33 spearman.rstat          MRD22
3    -0.33      jung.rstat EFSTIME,EFSCENSOR
```

3 CCPROMISE Analysis

As mentioned in section 2, the *ExpressionSet* of two forms of genomic data and pattern definition are required by CCPROMISE procedure. The code below performs a CCPROMISE analysis at gene level with fast permutation based on negative binomial.

```
> test1 <- CCPROMISE(geneSet=exmplGeneSet,
+                   ESet=exmplESet,
+                   MSet=exmplMSet,
+                   promise.pattern=exmplPat,
+                   strat.var=NULL,
+                   prlbl=c('LC50', 'MRD22', 'EFS', 'PR3'),
+                   EMLbl=c("Expr", "Methyl"),
+                   nbperm=TRUE,
+                   max.ntail=10,
+                   nperms=100,
+                   seed=13)
```

Gene level result:

```
> head(test1$PRres)

  Gene Expr_LC50.Stat Expr_MRD22.Stat Expr_EFS.Stat
1  DDR1    0.02537225    0.06515531    0.096304751
2  RFC2   -0.19860902    0.11057641    0.194535613
3  PAX8   -0.17035650   -0.05050530   -0.014700892
4 GUCA1A  -0.11857641    0.03538941    0.002563084
```

5	UBE1L	0.06880280	-0.05714667	-0.041141660
6	THRA	-0.05640794	-0.05223717	-0.071381381
	Expr_PR3.Stat	Methyl_LC50.Stat	Methyl_MRD22.Stat	
1	-0.062277435	-0.006796137	0.02987477	
2	-0.035501001	0.049870701	0.07273351	
3	0.078520899	-0.187767560	-0.06772779	
4	0.026874639	0.034627937	0.15896498	
5	0.009828512	NaN	NaN	
6	0.060008831	0.011165082	-0.04654276	
	Methyl_EFS.Stat	Methyl_PR3.Stat	PR3.Stat	
1	0.01013990	-0.011072845	-0.036675140	
2	0.05561824	-0.059407483	-0.047454242	
3	-0.10627160	0.120588984	0.099554942	
4	0.09232005	-0.095304321	-0.034214841	
5	NaN	NaN	0.009828512	
6	0.01189162	0.007828685	0.033918758	
	Expr_LC50.Pval	Expr_MRD22.Pval	Expr_EFS.Pval	
1	0.7222222	0.3888889	0.2222222	
2	0.1600000	0.0800000	0.0000000	
3	0.1904762	0.4761905	0.9047619	
4	0.7058824	0.9411765	0.9411765	
5	0.6666667	0.7500000	0.5000000	
6	0.6923077	0.5769231	0.1923077	
	Expr_PR3.Pval	Methyl_LC50.Pval	Methyl_MRD22.Pval	
1	0.3333333	0.9444444	0.7222222	
2	0.6000000	0.5600000	0.2400000	
3	0.2619048	0.1904762	0.3571429	
4	0.9411765	0.8823529	0.0000000	
5	0.8333333	NA	NA	
6	0.2307692	1.0000000	0.4230769	
	Methyl_EFS.Pval	Methyl_PR3.Pval	PR3.Pval	nperm
1	0.8888889	0.8333333	0.5555556	18
2	0.6000000	0.3600000	0.4000000	25
3	0.2380952	0.1071429	0.1190476	84
4	0.1764706	0.1176471	0.5882353	17
5	NA	NA	0.8333333	12
6	0.8846154	0.9615385	0.3846154	26

The code below performs a prbPROMISE analysis at probe pair level with fast permutation.

```
> test2 <- PrbPROMISE(geneSet=exmplGeneSet,
+                      ESet=exmplESet,
+                      MSet=exmplMSet,
+                      promise.pattern=exmplPat,
+                      strat.var=NULL,
+                      prlbl=c('LC50', 'MRD22', 'EFS', 'PR3'),
```

```
+           EMlbl=c("Expr", "Methyl"),
+           nbperm=TRUE,
+           max.ntail=10,
+           nperms=100,
+           seed=13)
```

Probe pair level correlation result at p value cut off 0.05:

```
> head(test2$CORres)
```

	Gene	Expr	Methyl
1007_s_at*cg00466425	"DDR1"	"1007_s_at"	"cg00466425"
1007_s_at*cg01386080	"DDR1"	"1007_s_at"	"cg01386080"
1007_s_at*cg01936707	"DDR1"	"1007_s_at"	"cg01936707"
1007_s_at*cg02313535	"DDR1"	"1007_s_at"	"cg02313535"
1007_s_at*cg02376496	"DDR1"	"1007_s_at"	"cg02376496"
1007_s_at*cg03270204	"DDR1"	"1007_s_at"	"cg03270204"
	Spearman.rstat		Spearman.p
1007_s_at*cg00466425	"0.3467"		"1.46642e-05"
1007_s_at*cg01386080	"0.2624"		"0.0011758053"
1007_s_at*cg01936707	"0.4245"		"7.56e-08"
1007_s_at*cg02313535	"0.1978"		"0.0150390162"
1007_s_at*cg02376496	"0.3069"		"0.0001359269"
1007_s_at*cg03270204	"-0.2394"		"0.0031454459"

Probe pair level PROMISE result of probe pair at p value cut off 0.05 as above:

```
> head(test2$PRres)
```

	Gene	Expr_LC50.Stat	Expr_MRD22.Stat
1	1007_s_at*cg00466425	-0.02436901	-0.1044313
2	1007_s_at*cg01386080	-0.02436901	-0.1044313
3	1007_s_at*cg01936707	-0.02436901	-0.1044313
4	1007_s_at*cg02313535	-0.02436901	-0.1044313
5	1007_s_at*cg02376496	-0.02436901	-0.1044313
6	1007_s_at*cg03270204	-0.02436901	-0.1044313
	Expr_EFS.Stat	Expr_PR3.Stat	Methyl_LC50.Stat
1	-0.1031647	0.07732165	-0.03802601
2	-0.1031647	0.07732165	-0.12352788
3	-0.1031647	0.07732165	-0.14957974
4	-0.1031647	0.07732165	0.11877059
5	-0.1031647	0.07732165	-0.02394829
6	-0.1031647	0.07732165	0.05799370
	Methyl_MRD22.Stat	Methyl_EFS.Stat	Methyl_PR3.Stat
1	0.092249971	0.11770019	-0.05730805
2	-0.009996237	0.03141073	0.03403780
3	0.069168489	-0.02279546	0.03440224
4	0.165563303	0.05556954	-0.11330114

5	0.167249598	0.17349992	-0.10560041	
6	-0.133609780	-0.14570372	0.07377326	
	PR3.Stat	Expr_LC50.Pval	Expr_MRD22.Pval	Expr_EFS.Pval
1	0.01000680	1.0000000	0.1666667	0.2500000
2	0.05567973	0.9032258	0.1935484	0.1935484
3	0.05586195	0.9473684	0.2631579	0.3157895
4	-0.01798974	1.0000000	0.1818182	0.2727273
5	-0.01413938	1.0000000	0.1666667	0.2500000
6	0.07554746	0.8620690	0.1896552	0.1724138
	Expr_PR3.Pval	Methyl_LC50.Pval	Methyl_MRD22.Pval	
1	0.1666667	0.7500000	0.1666667	
2	0.2903226	0.2903226	0.8709677	
3	0.3157895	0.3157895	0.4210526	
4	0.1818182	0.5454545	0.0000000	
5	0.1666667	0.7500000	0.0000000	
6	0.2586207	0.6724138	0.1206897	
	Methyl_EFS.Pval	Methyl_PR3.Pval	PR3.Pval	nperm
1	0.08333333	0.5000000	0.8333333	12
2	0.61290323	0.61290323	0.3225806	31
3	0.84210526	0.52631579	0.5263158	19
4	0.54545455	0.18181818	0.9090909	11
5	0.08333333	0.08333333	0.8333333	12
6	0.12068966	0.34482759	0.1724138	58