Understanding Outputs

Jishnu Das Lab

2024-02-23

Overview

We will analyze example output found in `SLIDE/test/out/0.01_1_out`

This is the general workflow:

1. Evaluate latent factor performance using `ControlPerformancePlot.png`
2. Look at top features in each latent factor using `plotSigGenes_marginals.png` and `plotSigGenes.png`
3. Look at interactions between latent factors with `plotInteractions.png`
4. Look at gene lists in `gene_list_Z.txt` files
5. View SLIDE results and parameters in `SLIDE_LFs.rds`

Evaluate latent factor performance using `ControlPerformancePlot.png`

This plot shows how the significant latent factors perform relative to their knockoff counterparts:

- The red line is the true performance of the significant latent factors (either correlation for continuous response/Y or AUC for binary response/Y).
- The blue density shows the performance of the knockoffs for the stand-alone/marginal latent factors
- The green density shows the performance of the true marginal latent factors paired with knockoffs for the interaction latent factors

A good model should have:

- **performance red line should be highest** (better performance) and to the right of both the blue and green densities - this indicates that there signal in the significant latent factors.
- **green density should be slightly higher than blue**, since we are using the true marginal latent factors paired with knockoff interaction latent factors - we are trying to evaluate whether the interaction latent factors are meaningful
- **blue density has the lowest performance**, as knockoffs for our marginal latent factors should not have signal

```
knitr::include_graphics("examples/out/0.01_1_out/ControlPerformancePlot.png")
```
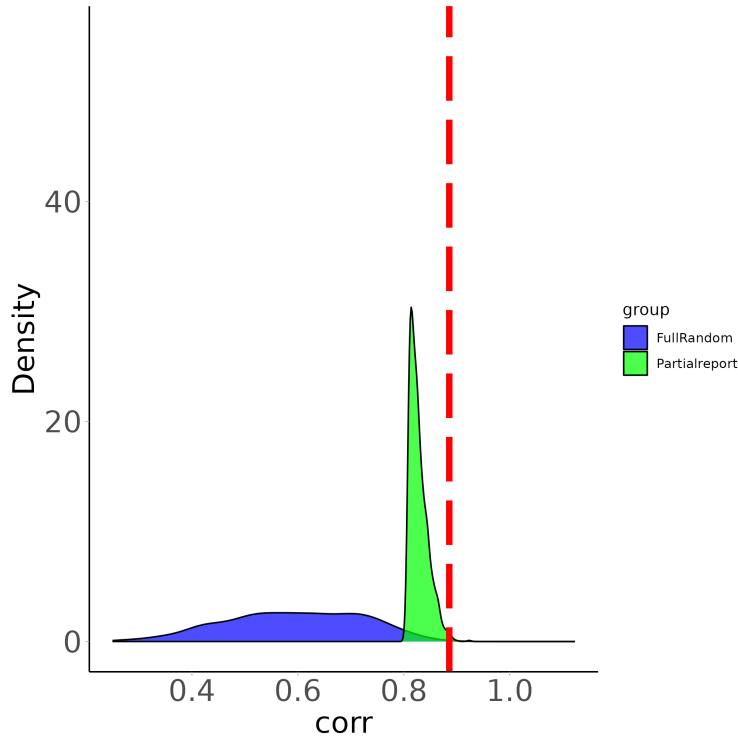
Figure 1: Significant Latent Factor Performance

Above is an example of a model with great performance. In some cases, you may find that the green density overlaps perfectly with the blue - in this case, the interaction latent factors are not meaningful. If you find that all three (blue, green and red) perfectly overlap, then this indicates a lack of signal in the significant latent factors and there are a few options:

1. Increase `spec` parameter (increase FDR control for choosing latent factors)
2. Re-cluster latent factors using different `delta` and `lambda` parameters - it is possible the latent factors are a poor fit to the data (too many latent factors will overfit = increase `delta`; too few latent factors will be over-clustered = decrease `delta`)

The data for this plot is stored in `ControlPerformance.rds`; the `ControlPerformancePlot.rds` has the plot object for the ggplot above.

Look at top features in each latent factor using `plotSigGenes_marginals.png` and `plotSigGenes.png`

The `plotSigGenes_marginals.png` shows the top features in the stand-alone/marginal significant latent factors. By default, we show the top 10 features by loading (features that contribute most to the latent factor), as well as the top 10 univariate features (features that have high correlation or AUC with response/Y). Each latent factor shows 20 features - the bottom 10 are the highest loading and the top 10 are highest univariate.

```
knitr::include_graphics("examples/out/0.01_1_out/plotSigGenes_marginals.png")
```

SLIDE Marginal Variables

Genes Associated with Significant Latent Factors

46	58	97	129	137
c.3.MT1X	c.7.APOE		c.4.TCIM	c.14.ISG15
c.6.MT1X	c.7.APOC1	c.0.KRT6A	c.13.FLNA	c.14.IFI6
c.12.AREG	c.7.S100A9	c.11.KRT6A	c.16.SPINK5	c.14.IFI27
c.5.MT1X	c.17.HSPA6	c.15.CD164	c.9.CCR7	c.13.CNN1
c.17.TOB1	c.10.HSPA6	c.15.CD59	c.4.IER3	c.16.IFI27
c.6.CLU	c.8.HES1	c.15.SOD2	c.6.CLU	c.15.DCD
c.8.MT1X	c.12.ID2	c.0.KRT16	c.15.SLC12A2	c.10.FOSB
c.4.MT1X	c.7.SPP1	c.12.LGALS1	c.3.PTN	c.13.HBB
c.16.IFI27	c.8.POSTN	c.15.LY6D	c.13.MYLK	c.0.LGALS7B
c.8.CTGF	c.16.DCD	c.16.IFI27	c.11.ATF3	c.3.POSTN
c.3.SLPI	c.3.SLPI	c.13.SCGB2A2	c.3.SLPI	c.3.SLPI
c.3.TPPP3	c.3.TPPP3	c.3.SLPI	c.3.TPPP3	c.3.TPPP3
c.10.FOS	c.10.FOS	c.3.TPPP3	c.10.FOS	c.10.FOS
c.10.TXNIP	c.10.TXNIP	c.10.FOS	c.10.TXNIP	c.10.TXNIP
c.3.DIO2	c.3.DIO2	c.10.TXNIP	c.3.DIO2	c.3.DIO2
c.10.KLF6	c.10.KLF6	c.3.DIO2	c.10.KLF6	c.10.KLF6
c.3.TNC	c.3.TNC	c.10.KLF6	c.3.TNC	c.3.TNC
c.4.CXCL3	c.4.CXCL3	c.3.TNC	c.4.CXCL3	c.4.CXCL3
c.3.TNXB	c.3.TNXB	c.3.TNXB	c.3.TNXB	c.3.TNXB
c.5.IGFBP5	c.5.IGFBP5	c.5.IGFBP5	c.5.IGFBP5	c.13.CCDC80

Significant Latent Factor

Figure 2: Marginal Significant Latent Factors

We also include plots showing both the marginal and interaction latent factors in this format; here, the bolded/italicized latent factors are the stand-alone/marginal latent factors and the plain text are the interactions.

```
knitr::include_graphics("examples/out/0.01_1_out/plotSigGenes.png")
```

Figure 3: Marginal (bolded) and Interaction (plain) Significant Latent Factors

Look at interactions between latent factors with `plotInteractions.png`

This plot shows a graphical representation of how the interacting latent factors are paired with the marginals. Here, `Z97` interacts with both `Z147` and `Z109`; `Z46`, `Z58`, `Z129`, and `Z137` have no interactions

```
knitr::include_graphics("examples/out/0.01_1_out/plotInteractions.png")
```
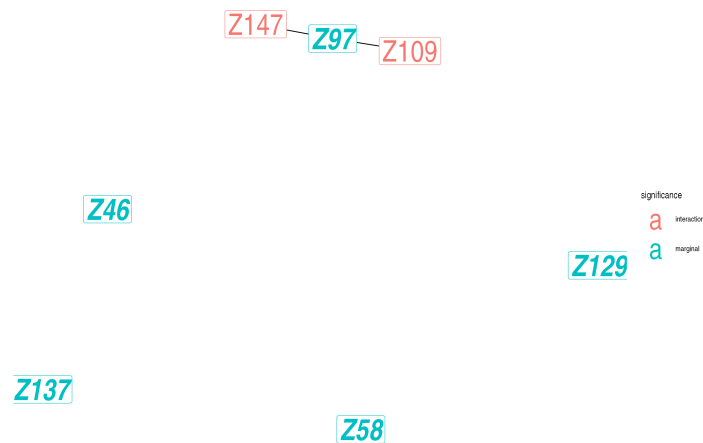

Figure 4: Latent Factor Interactions

Look at gene lists in `gene_list_Z.txt` files

These outputs provide a table of the top 20 features (same as above in the text plots), and provide the associated latent factor loadings `A_loading` and correlation/AUC `corrs`, as well as the association of each feature with the response `color`. For the color, `Red` is associated with a higher response/Y (or Y = 1 for binary) and `Blue` is associated with a lower response/Y (or Y = 0 for binary).

```
gene_list = read.csv("examples/out/0.01_1_out/gene_list_Z97.txt",
    row.names = 1)
knitr::kable(head(gene_list))
```

	A_loading	corrs	color
c.0.KRT6A	1.0000000	0.5470392	Red
c.11.KRT6A	1.0000000	0.6641996	Red
c.15.CD164	0.0322199	-0.3351158	Blue
c.15.CD59	0.0314163	-0.1307760	Blue
c.15.SOD2	0.0254815	0.1766502	Red
c.0.KRT16	0.0244576	0.2317077	Red

View SLIDE results and parameters in `SLIDE_LFs.rds`

We can see the parameters that were used to run SLIDE in `SLIDE_LFs.rds`

```
SLIDE_LFs = readRDS("examples/out/0.01_1_out/SLIDE_LFs.rds")
names(SLIDE_LFs)
```

```
#> [1] "SLIDE_res"     "SLIDE_param"   "marginal_vals" "interaction"
```

We can see the parameters:

```
knitr::kable(SLIDE_LFs$SLIDE_param)
```

	x
method	4.0
spec	0.3
fdr	0.1
niter	20.0
f_size	24.0