TReNA: Past, Present, and Future

Matt Richards

5/4/2017

What is TReNA?

It's an R package for:

- Feature selection
- Footprinting
- Functional genomics

Using TReNA

When using TReNA, you'll need an assay matrix of expression data where:

1. Rows are different genes
2. Columns are different samples

The TReNA package also comes with its own example data:

```
suppressMessages(library(TReNA))
load(system.file(package="TReNA",
                 "extdata/ampAD.154genes.mef2cTFs.278sample
mtx.sub[1:3,1:5]
```

```
##       S11344_TCX S11316_TCX S11431_TCX S11341_TCX S1128
## ADNP   94.893216  83.284825  79.511862  80.608077  53.16
## ADNP2  24.279075  18.847705  20.676318  19.959116  17.70
## ALX3    0.874247   1.285071   1.030784   1.388908   0.62
```

Creating and Solving a TReNA Object

```
trena <- TReNA(mtx.assay = mtx.sub, solver = "lasso")
target.gene <- "MEF2C"
tfs <- setdiff(rownames(mtx.sub), target.gene)
tbl.out <- solve(trena, target.gene, tfs)
```

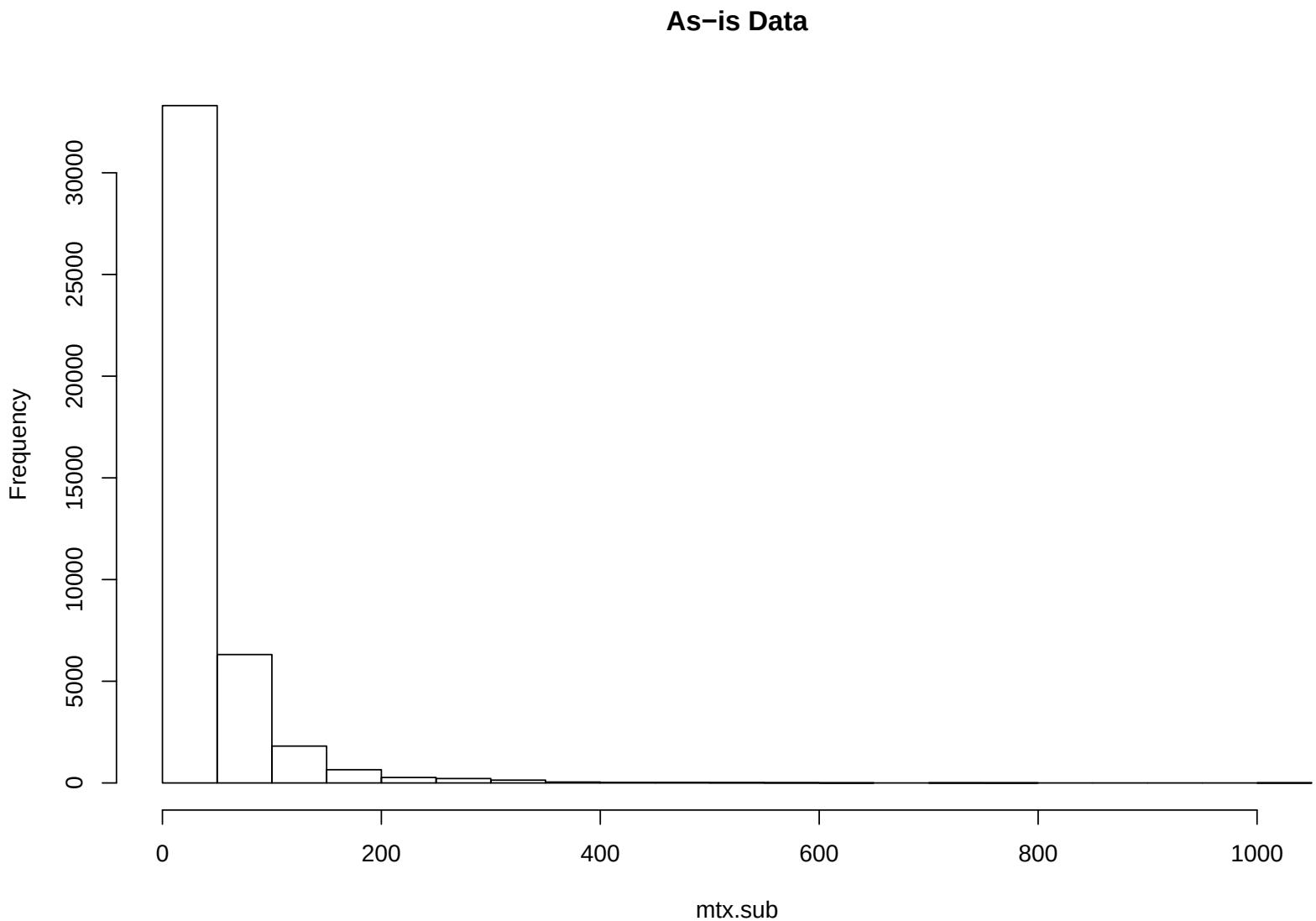
	beta	intercept	gene.cor
HDX	7.290454	11.87699	0.7241906
STAT4	3.769317	11.87699	0.9047599
FOXP2	1.618574	11.87699	0.8196172
LHX6	1.081253	11.87699	0.7146115

TReNA in December 2016

- 3 different "solvers"
  - LASSO
  - Random Forest
  - Bayes Spike
- Footprint Finders
  - Pull out footprints from a:
    - Gene
    - Region

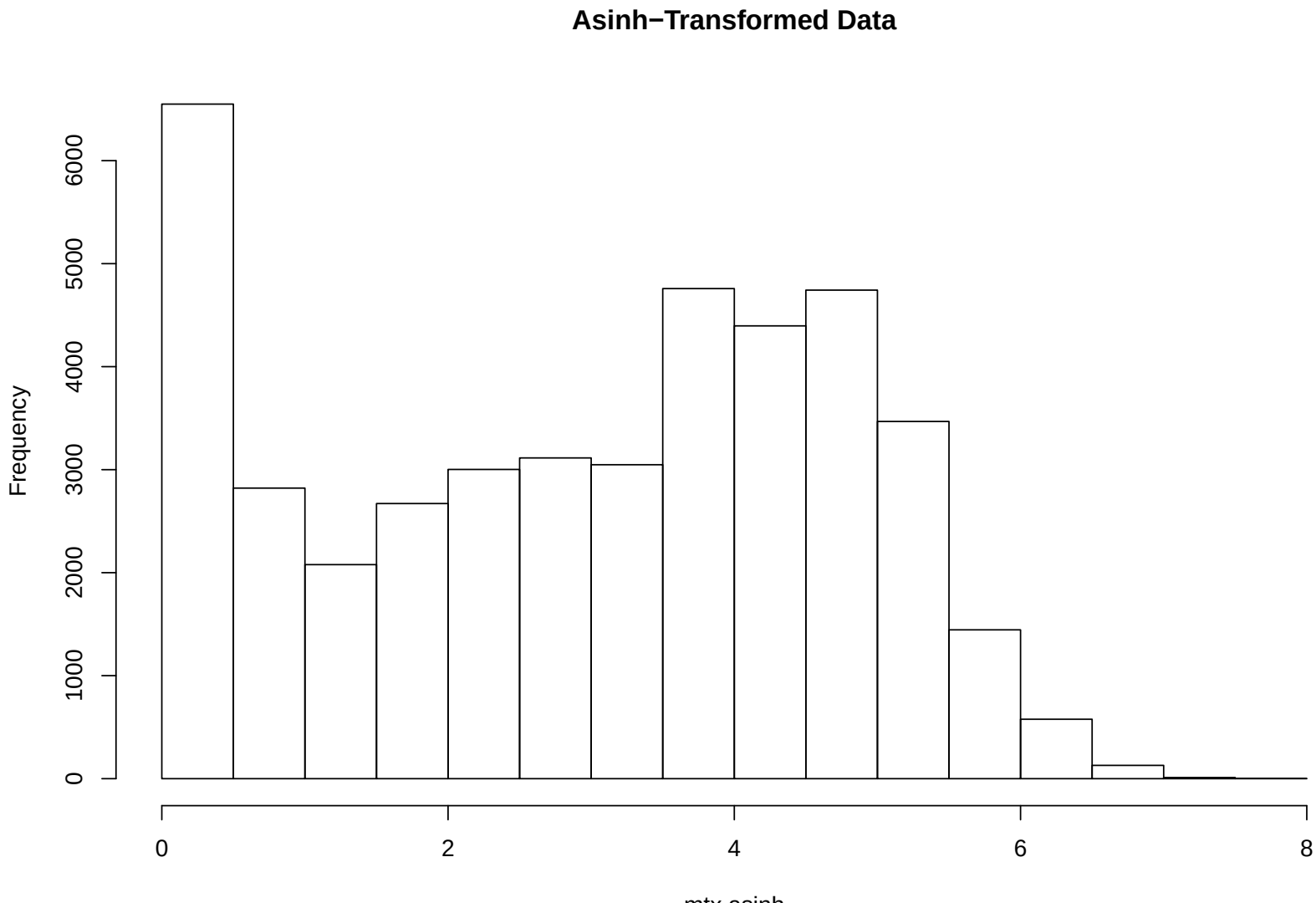
Distribution Effects

```
hist(mtx.sub, main = "As-is Data")
```



Transforming with arcsinh

```
mtx.asinh <- asinh(mtx.sub)
hist(mtx.asinh, main = "Asinh-Transformed Data")
```

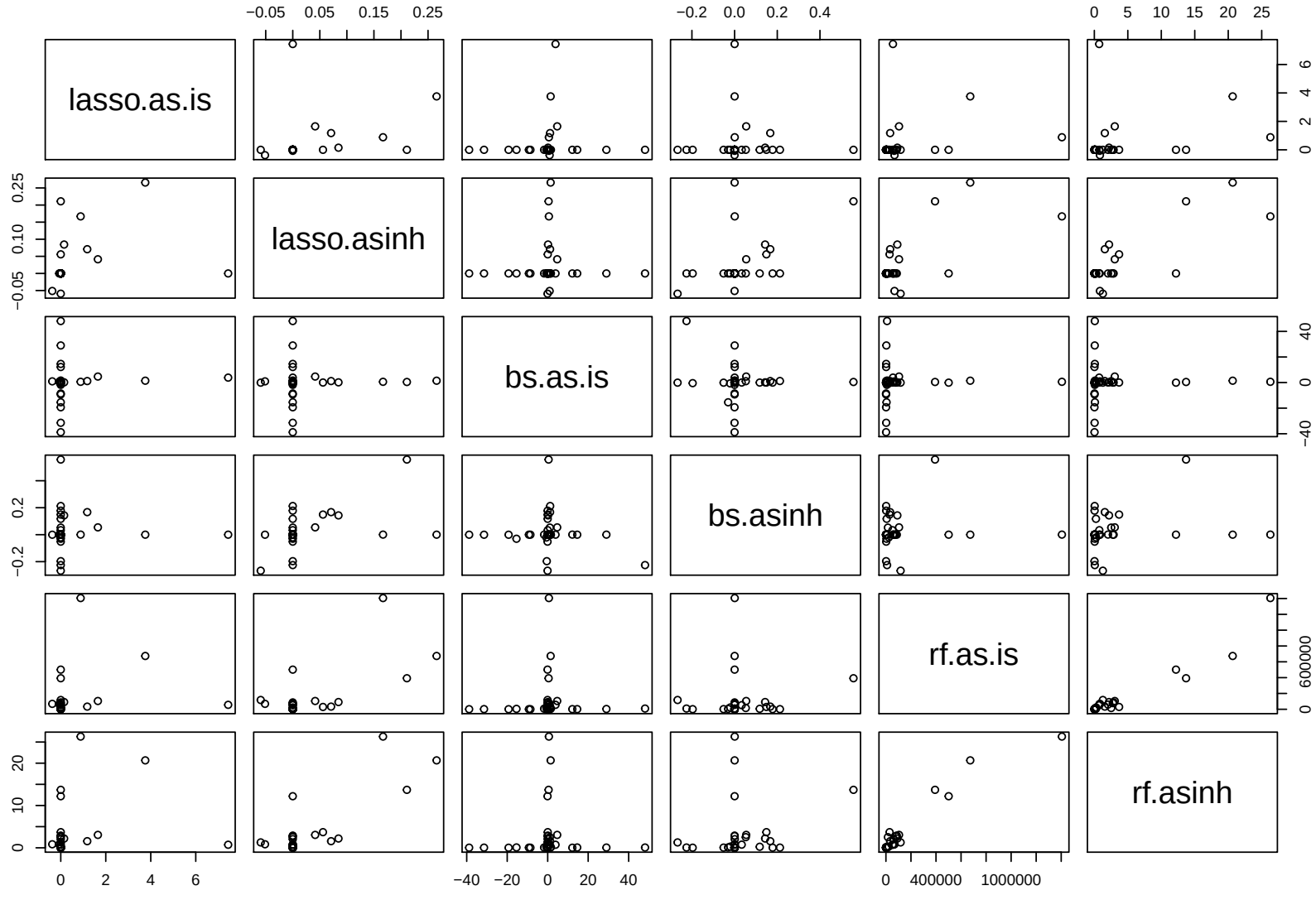


Solvers effects

In addition to the distributions, solver choice can greatly affect results.

```
## [1] --- Testing LASSO
## [1] --- Testing Bayes Spike
## [1] --- Testing Random Forest
## [1] --- Testing Square Root LASSO
```

Pair Plots



Correlations

	lasso.as.is	lasso.asinh	bs.as.is	bs.asinh	rf.as.is	rf.asin
lasso.as.is	1.0000	0.320	0.066	-0.0015	0.220	0.26
lasso.asinh	0.3200	1.000	0.028	0.4900	0.690	0.82
bs.as.is	0.0660	0.028	1.000	-0.1400	0.027	0.02
bs.asinh	-0.0015	0.490	-0.140	1.0000	0.110	0.23
rf.as.is	0.2200	0.690	0.027	0.1100	1.000	0.96
rf.asinh	0.2600	0.820	0.029	0.2300	0.960	1.00

Takeaways from Past TReNA

- Distribution matters
  - As-is data is often quite skewed
  - We recommend asinh or VOOM
- Solver choice matters
  - Are 3 solvers sufficient?
  - What else could/should we add?
  - How do we compare between solvers?

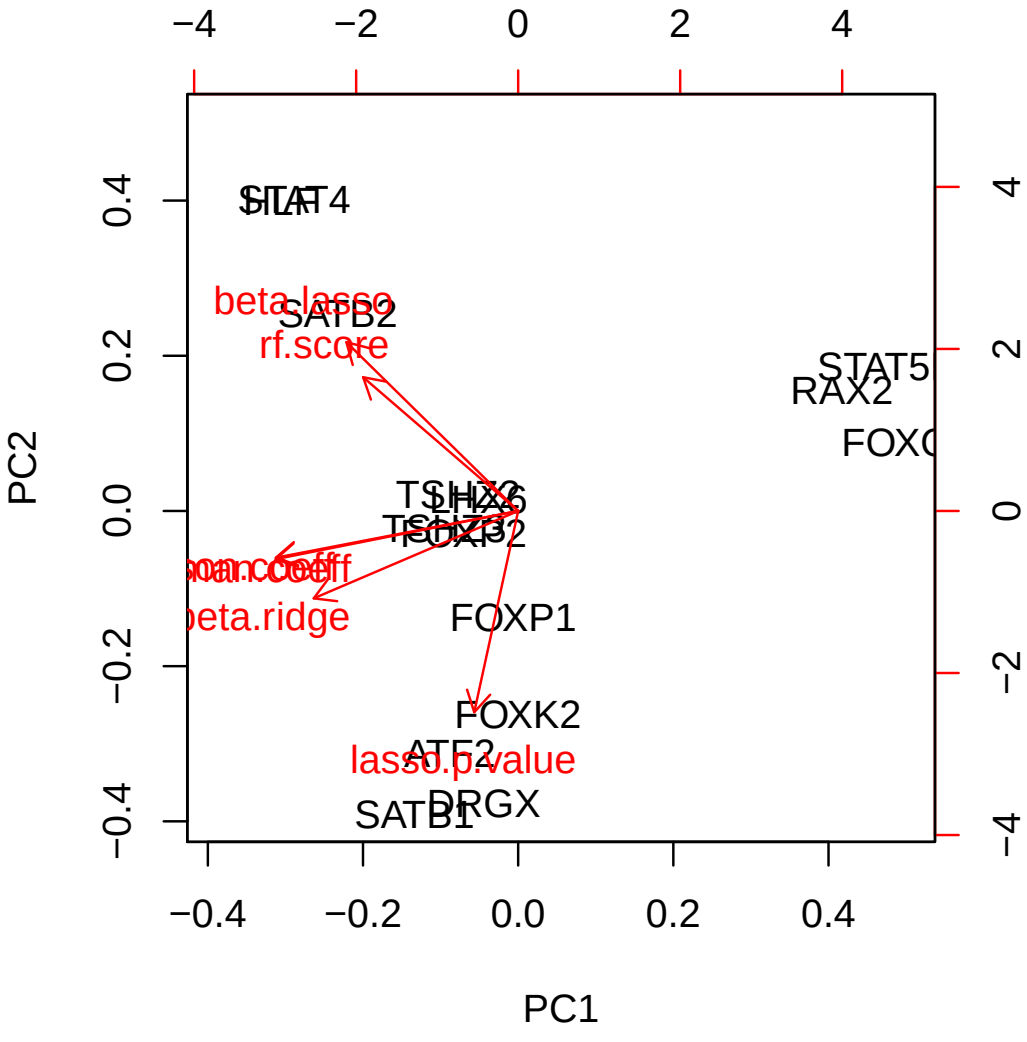
We've Added 5 (6) Solvers

- Square Root LASSO
- Pearson Correlation
- Spearman Correlation
- LASSO P-Values
- Ridge Regression
- Ensemble Solver

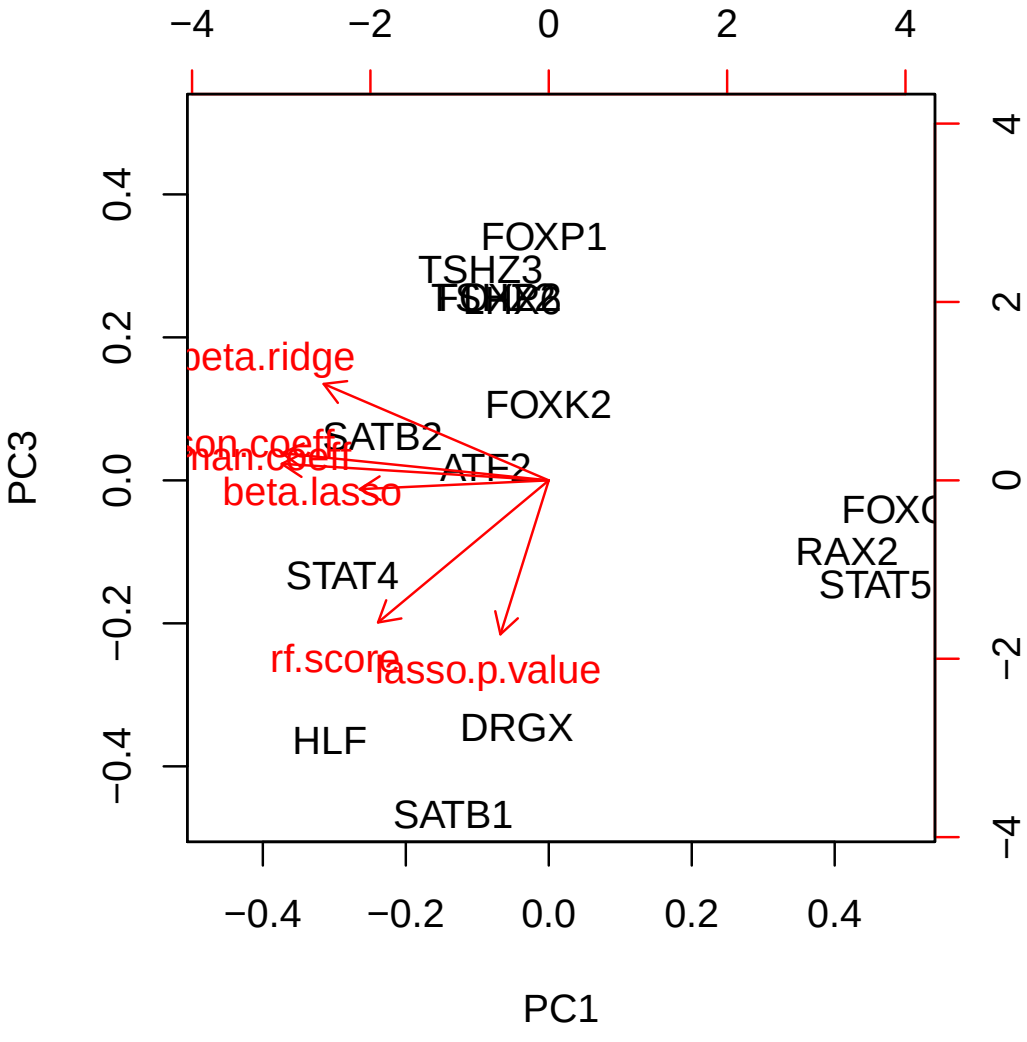
The Ensemble Solver class

	gene	beta.lasso	rf.score	lasso.p.value	beta.ridge	conc
7	HLF	0.1668048	31.308459	0.0000000	0.0445578	0.6
12	STAT4	0.2639761	18.984825	0.0000000	0.0360529	0.5
11	SATB2	0.2125086	11.964488	0.0000000	0.0545548	0.5
14	TSHZ2	0.0851875	2.161280	0.0000000	0.0456957	0.5
15	TSHZ3	0.0599899	3.555202	0.0000000	0.0742151	0.5
10	SATB1	0.0000000	10.151417	0.9571191	0.0482014	0.6

Examining Solver PCA



Examining Solver PCA



The CandidateFilter Class

- Feature selection is nice, but nothing new
- Our value added is largely in how we filter predictors
  - VarianceFilter class
  - **FootprintFilter** class

Using a VarianceFilter

```
variance.filter <- VarianceFilter(mtx.assay = mtx.asinh)
tf.list <- getCandidates(variance.filter, extraArgs = list(
str(tf.list)
```

```
## List of 2
##  $ tfs    : chr [1:36] "BCL6" "BCL6B" "CREB5" "CUX2" ...
##  $ tf.vars: Named num [1:36] 0.23 0.479 0.28 0.501 0.248
##   ..- attr(*, "names")= chr [1:36] "BCL6" "BCL6B" "CREB5
```

Using a FootprintFilter

```
tfs <- getCandidates(footprint.filter, extraArgs)
str(tfs)
```

```
## List of 2
##  $ tfs: chr [1:64] "LHX4" "LHX6" "LHX9" "LHX2" ...
##  $ tbl:'data.frame': 580 obs. of  18 variables:
##   ..$ name      : chr [1:580] "MA0046.2" "MA0046.2" "MA0
##   ..$ loc       : chr [1:580] "chr5:88903305-88903319" "
##   ..$ chrom     : chr [1:580] "chr5" "chr5" "chr5" "chr5
##   ..$ start     : int [1:580] 88903305 88903305 8890338
##   ..$ endpos    : int [1:580] 88903319 88903319 8890339
##   ..$ type      : chr [1:580] "motif.in.footprint" "moti
##   ..$ length    : int [1:580] 15 15 13 13 13 13 13 13 13
##   ..$ strand    : chr [1:580] "+" "-" "+" "+" ...
##   ..$ sample_id : chr [1:580] "ENCSR318PRQ_wellington" "
##   ..$ method    : chr [1:580] "WELLINGTON" "WELLINGTON"
##   ..$ provenance: chr [1:580] "brain.filler.minid" "brai
##   ..$ score1    : num [1:580] -23.1 -23.1 -25.9 -25.9 -2
```

The FootprintFinder Class

We have public databases of footprints (with more on the way)

```
genome.db.uri    <- "postgres://bddsrds.globusgenomics.org/
footprint.db.uri <- "postgres://bddsrds.globusgenomics.org/
fpf <- FootprintFinder(genome.db.uri, footprint.db.uri, qu
tbl.fp <- getFootprintsInRegion(fpf, "chr5", 88822685, 890
str(tbl.fp)
```

Unfortunately, the databases decided to misbehave right now. . .

Summing Up TReNA at Present

Current Workflow:

1. Start with a matrix of expression data
2. Transform the matrix to alter the distribution
3. Use a filter to wean down predictors
4. Run a feature selection with chosen solver

Where do I get TReNA?

- Accepted to Bioconductor Development Branch
- PriceLab Github

What's Next?

- More databases
  - 22 tissue types
  - 3 footprinting methods
  - 2 different alignments
- Structural changes to TReNA
  - More filtering options
    - DHS filter
    - GO filter
- Web application
  - Ability to build TReNA workflows

Thank you

Special thanks to:

- Nathan Price
- Paul Shannon
- Cory Funk
- Max Robinson
- Rory Donovan-Maiye

For More on TReNA, visit the current TReNA webpage