Integrative cross-omics and cross-context analysis elucidates molecular links underlying genetic effects on complex traits

Yihao Lu, Meritxell Oliva, Brandon L. Pierce, Jin Liu, and Lin S. Chen

Introduction

This vignette provides an introduction to the `X.ING` package. The R package `X.ING` implements the X-ING method for cross-omics and cross-context analysis.

To install the development version of the `X.ING` package, please load the `devtools` package first. Note that `X.ING` requires the `CCA`, `RGCCA`, `Rcpp`, and `RcppArmadillo` packages.

To install this package, run the following command in R

```
library(devtools)
install_github("ylustat/X.ING")
```

Load the package using the following command:

```
library("X.ING")
```

```
##
## Attaching package: 'X.ING'
```

```
## The following objects are masked from 'package:base':
##
##     scale, unique
```

This vignette depends on R packages `MendelianRandomization`, `ggplot2` and `tidyverse`, you can load these packages using the following command:

```
suppressMessages(library("tidyverse"))
suppressMessages(library("pROC"))
suppressMessages(library("MASS"))
library("ggplot2")
library("MendelianRandomization")
library("RGCCA")
```

Fit XING using simulated data

The main function of the `X.ING` package is `XING()`. In this section, we fit `XING` function using simulated data (provided in the package) as an example to illustrate the basic usage of `XING`. The example data is a list with two elements, where `z` is a list of z-scores and `gamma` is a list of true underlying association status. `1` in `gamma` indicates the presence of non-zero effect and `0` indicates the absence of effect. We have three sets of data available in the example data (L=3). The number of contexts are 10, 8, and 6, respectively.

```
data("example_data")
length(example$z)
```

```
## [1] 3
```

```
sapply(example$z, dim)
```

```
##      [,1] [,2] [,3]
## [1,] 6000 6000 6000
## [2,]   10    8    6
```

The proportion of true associations are ~0.04.

```
sapply(example$gamma, mean)
```

```
## [1] 0.04265000 0.03893750 0.03922222
```

The X-ING takes the z-score list as input. It automatically use CCA (when L=2) for GCCA (when L>2) for the multi-view learning. Afterwards, it applies PCA to extract shared patterns across contexts for each omics. A simplest application where two sets of z-scores are analyzed:

```
z <- example$z
res <- XING(z_list = z[1:2])
```

The output includes posterior probabilities (`alpha`) and posterior means (`mu`):

```
names(res)
```

```
## [1] "mu"    "alpha"
```

```
lengths(res)
```

```
##    mu alpha
##     2     2
```

AUC can be calculated based on the posterior probability and true association status

```
gamma <- example$gamma

mapply(function(g,pp) auc(c(g),c(pp), quiet = T), gamma[1:2], res$alpha)
```

```
## [1] 0.8548316 0.8457119
```

It switches to GCCA (implemented in RGCCA package) if L>2 (note that the computation speed may substantially slowed):

```
z <- example$z
res <- XING(z_list = z)
```

Similar to the previous example, the output also includes posterior probabilities (`alpha`) and posterior means (`mu`):

```
names(res)
```

```
## [1] "mu"    "alpha"
```

```
lengths(res)
```

```
##    mu alpha
##     3     3
```

And the corresponding AUCs:

```
mapply(function(g,pp) auc(c(g),c(pp), quiet = T), gamma, res$alpha)
```

```
## [1] 0.8953075 0.8766113 0.8168692
```

Users may specify the sample overlap matrix $\widehat{\boldsymbol{R}}$ through thr `Lambda_list` parameter. The number of canonical coefficients to keep, and the number of PCs can be specified by `CC` and `PC_list`. For example,

```
res <- XING(z_list = z, CC = 4, PC_list = c(4,3,2), Lambda_list = lapply(sapply(z,ncol),diag))
mapply(function(g,pp) auc(c(g),c(pp), quiet = T), gamma, res$alpha)
```

```
## [1] 0.8249574 0.8323304 0.8175283
```

To reduce computation time, user may specify the iteration time via `iterT`. Also, the `tolerance` parameter controls the criterion for convergence of the GCCA algorithm. A smaller `tolerance` may lead to substantially longer computation time.