mTADA

Code ▾

This notebook descibes steps used to jointly analyze two traits by `mTADA`.

I. Introduction

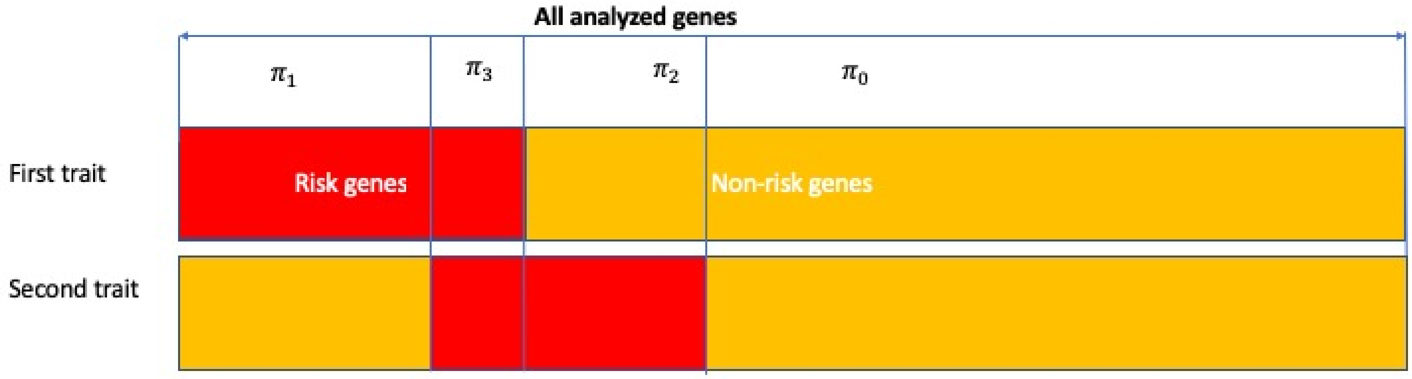
`mTADA` jointly analyze de novo mutations (DNMs) of two traits to 1) estimate the gene-level genetic overlap of the two traits; 2) report shared and specific risk genes; and 3) identify additional risk genes for each analyzed trait.

The method requires genetic parameters from single-trait analyses (the third and fourth columns in Table 1 below). Users can obtain single-trait parameters from `extTADA`/`TADA` methods.

Table 1. `mTADA` model for one variant category at the i^{th} gene.

Hypothesis	Proportion	First trait	Second trait
H_0	π_0	$x_{i1} \sim Poisson(2N_1\mu_i)$	$x_{i2} \sim Poisson(2N_2\mu_i)$
H_1	π_1	$x_{i1} \sim Poisson(2N_1\gamma_1\mu_i)$; $\gamma_1 \sim Gamma(\bar{\gamma}_1\beta_1, \beta_1)$	$x_{i2} \sim Poisson(2N_2\mu_i)$
H_2	π_2	$x_{i1} \sim Poisson(2N_1\mu_i)$	$x_{i2} \sim Poisson(2N_2\gamma_2\mu_i)$; $\gamma_2 \sim Gamma(\bar{\gamma}_2\beta_2, \beta_2)$
H_3	π_3	$x_{i1} \sim Poisson(2N_1\gamma_1\mu_i)$; $\gamma_1 \sim Gamma(\bar{\gamma}_1\beta_1, \beta_1)$	$x_{i2} \sim Poisson(2N_2\gamma_2\mu_i)$; $\gamma_2 \sim Gamma(\bar{\gamma}_2\beta_2, \beta_2)$

Figure 1. `mTADA` framework.



Data for reproducible analyses

Data used in the main manuscript are inside the folder data (data):

1. FullDataSet_DenovoMutations_for_mTADA.txt (data/FullDataSet_DenovoMutations_for_mTADA.txt): all gene-level de novo mutations.
2. SingleTrait_Parameters.txt (data/SingleTrait_Parameters.txt): all single-trait parameters. We used `extTADA` to estimate these parameters.

> *Note*: Users can re-run all these single-trait analyses by following an example here: https://github.com/hoangtn/extTADA (https://github.com/hoangtn/extTADA).

II. Requirements

`mTADA` **is written in** `R`. Other `R` packages are required to run `mTADA`:

- `rstan`: https://mc-stan.org/rstan/ (https://mc-stan.org/rstan/).
- `locfit`: https://cran.r-project.org/web/packages/locfit/index.html (https://cran.r-project.org/web/packages/locfit/index.html).

III. An example: joint analysis of DD and EE DNMs.

Only one function `mTADA` (in the **Run** `mTADA` section) is used to obtain results. However, some additional steps are described here.

Load the source codes

Hide

```
sourceDir <- "./script/"
rFile <- dir(sourceDir, ".R$")
for (ir in 1:length(rFile)){
    source(paste0(sourceDir, rFile[ir]))
}
```

```
locfit 1.5-9.1   2013-03-22
```

Hide

```
dataDir <- "./data/"
```

Read the data and single-trait parameters

Hide

```
## De novo data
data <- read.table(paste0(dataDir, "FullDataSet_DenovoMutations_for_mTADA.txt"), header
 = TRUE, as.is = TRUE)
## Single-trait parameters
sPar <- read.table(paste0(dataDir, "SingleTrait_Parameters.txt"), as.is = TRUE, header =
TRUE)

trait1 = "DD"
trait2 = "EE"
##Take a quick look at the single-trait parameters of DD and EE
sPar[grep(trait1, sPar[, 1]), ] ##Trait 1
```

	Parameter <chr>	EstimatedValue <dbl>
8	DD_pi[1]	0.02936283
9	DD_hyperGammaMeanDN[1]	22.31762802
10	DD_hyperGammaMeanDN[2]	86.03966530
11	DD_hyperBetaDN[1]	0.82594514
12	DD_hyperBetaDN[2]	0.80689775

5 rows

Hide

```
sPar[grep(trait2, sPar[, 1]), ] ##Trait 2
```

	Parameter <chr>	EstimatedValue <dbl>
18	EE_pi[1]	0.01548789
19	EE_hyperGammaMeanDN[1]	51.08181282
20	EE_hyperGammaMeanDN[2]	65.15189031
21	EE_hyperBetaDN[1]	0.80906448
22	EE_hyperBetaDN[2]	0.80774192

5 rows

Set parameters for two traits.

As described above, mTADA needs single-trait parameters:

- the number of trios: *ntrio*;
- the mean and disperson parameters of relative risks: $\bar{\gamma}_j$ and β_j (j=$1, 2$);
- the proportion of risk genes: π_1^S and π_2^S.

All these parameters are shown above.

Hide

```
### Trait-1 INFORMATION
ntrio1 = 4293 #family numbers
p1 = 0.02936283 #The proportion of risk genes, this is p1S
meanGamma1 = c(22.31762802,  86.03966530) #Mean Gamma of two categories
beta1 = c(0.82594514, 0.80689775) #Beta values inside the distribution RR ~ Gamma(meanRR
*beta, beta)
dataT1 <- data[, paste0(c("dn_damaging_", "dn_lof_"), trait1)] #De novo data
muDataT1 <- data[, c("mut_damaging", "mut_lof")] #Mutation data of the first trait
### Trait-2 INFORMATION
ntrio2 = 356
p2 = 0.01548789 #This is p2S
meanGamma2 = c(51.08181282, 65.15189031)
beta2 = c(0.80906448, 0.80774192)
dataT2 <- data[, paste0(c("dn_damaging_", "dn_lof_"), trait2)]
muDataT2 <- muDataT1
```

Run mTADA

In this example, we only use a small number of iterations and two MCMC chains. However, users can change these parameters to obtain more reliable results.

Hide

```
nIteration = 2000 #This should be higher to obtain better results.
nChain = 2 #The number of MCMC chains

##########MAIN ANALYSIS
mTADAresults <- mTADA(geneName = data[, 1],
    #######Trait-1 information
                  ntrio1 = ntrio1, # Trio number of Trait 1
                  p1 = p1, #Risk-gene proportion of Trait 1
                  dataDN1 = data.frame(dataT1), #De novo data of Trait 1
                  mutRate1 = data.frame(muDataT1), # Mutation rates of Trait 1
                  hyperGammaMeanDN1 = c(meanGamma1), # Mean relative risks of Trait 1
                  hyperBetaDN01 = beta1, #NULL, #array(c(1, 1)),
    #######Trait-2 information
                  ntrio2 = ntrio2, # Trio number of Trait 2
                  p2 = p2, #Risk-gene proportion of Trait 2
                  dataDN2 = data.frame(dataT2), # De novo data of Trait 2
                  mutRate2 = data.frame(muDataT2), # Mutation rates of Trait 2
                  hyperGammaMeanDN2 = c(meanGamma2), # Mean relative risks of Trait 2
                  hyperBetaDN02 = beta2, #NULL, #array(c(1, 1)),
    ####Other parameters
                  nIteration = nIteration,
                  useMCMC = TRUE, #If FALSE, it will use the 'Variational Bayes' approac
h.
                  nChain = nChain
                      )
```

```
No information for core numbers (nCore); therefore, nCore = nChain: 2 core(s) is/are use
d

Loading required package: ggplot2
Loading required package: StanHeaders
rstan (Version 2.18.2, GitRev: 2e1f913d3ca3)
For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
===================
Building the model
=================



=======Use MCMC===========
```

```
starting worker pid=6868 on localhost:11020 at 16:38:02.167
starting worker pid=6878 on localhost:11020 at 16:38:02.371

SAMPLING FOR MODEL 'ff6d9758ae6f1964e2f79a593fbb863b' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.061898 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 618.98 secon
ds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:

SAMPLING FOR MODEL 'ff6d9758ae6f1964e2f79a593fbb863b' NOW (CHAIN 2).
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2:
Chain 2: Gradient evaluation took 0.065093 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 650.93 secon
ds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
```

Get results

`mTADA`'s output includes:

1. `data`: main gene-level results (posterior probabilities for the four models as described in the main manuscript: PP0, PP1, PP2 and PP3).
2. `probModel`: a vector of $\pi_j, (j = 0..3)$ in Table 1.
3. `pars`: the estimated value and credible interval of π_3 (described as p12 in the our code).
4. `mcmcData`: MCMC sampling results for π_3.

The most important information is from `data`. **Users can use this information to obtain top prioritized genes for downstream analyses (e.g., top shared/specific genes, top genes for each trait).** However, we will also take a quick look at all these information.

Results for downstream analyses (gene-level posteior probabilities of four models)

Hide

```
fData <- mTADAresults$data ## Full analysis results of the two-trait analysis.
head(fData)
```

	geneN… <fctr>	dn_damaging_… <int>	dn_lof_DD <int>	dn_damaging_EE <int>	dn_lof_EE <int>	NO <dbl>	BOTH <dbl>
1	A1BG	0	0	0	0	0.9784133	0.0028198649
2	A1BG-AS1	0	0	0	0	0.9646577	0.0059529886
3	A1CF	0	0	0	0	0.9893207	0.0006352912
4	A2M	0	0	1	0	0.7697822	0.0024472701
5	A2M-AS1	0	0	0	0	0.9636641	0.0061846227
6	A2ML1	0	0	0	0	0.9919200	0.0002052714

6 rows | 1-9 of 9 columns

Genes with PP3 > 0.8 (Posterior probabilities of Model 3)

Hide

```
fData[fData$BOTH > 0.8, ]
```

	geneN… <fctr>	dn_damaging_… <int>	dn_lof_DD <int>	dn_damaging_EE <int>	dn_lof_EE <int>	NO <dbl>	BOT <db
2348	CACNA1A	5	0	2	0	3.131031e-04	0.993239
3201	CHD2	0	6	0	1	4.633821e-10	0.935622
6254	GABBR2	2	0	2	0	2.527242e-03	0.951377
6265	GABRB3	2	0	2	0	9.265651e-04	0.979521
6610	GNAO1	4	1	2	0	1.609090e-08	0.998367
7165	HECW2	5	1	1	0	1.940937e-06	0.889528
7426	HNRNPU	0	7	0	1	8.809791e-13	0.934957
8283	KCNQ2	9	0	2	0	3.262209e-13	0.998174
8284	KCNQ3	3	0	1	0	4.328467e-03	0.911280
10146	MLL	1	26	1	0	1.485790e-48	0.864694

1-10 of 14 rows | 1-8 of 9 columns Previous **1** 2 Next

Genes with PP1 > 0.8 (Posterior probabilities of Model 1)

Hide

```
fData[fData$FIRST > 0.8, ]
```

	geneN...	dn_damaging_...	dn_lof_DD	dn_damaging_EE	dn_lof_EE	NO	BC
	<fctr>	<int>	<int>	<int>	<int>	<dbl>	<c
347	ADNP	1	19	0	0	4.078191e-37	0.19631
681	ANKRD11	0	32	0	0	2.291183e-60	0.13523
1000	ARID1A	1	2	0	0	8.453220e-02	0.11369
1001	ARID1B	0	30	0	0	1.699967e-56	0.14276
1002	ARID2	0	3	0	0	2.189776e-03	0.17063
1153	ASXL1	0	4	0	0	1.841959e-05	0.16860
1155	ASXL3	0	14	0	0	2.519386e-25	0.19578
1317	AUTS2	0	4	0	0	1.132125e-05	0.18047
1450	BCL11A	2	3	0	0	7.447173e-07	0.18657
1630	BRPF1	0	4	0	0	7.096549e-05	0.12944

1-10 of 72 rows | 1-8 of 9 columns Previous **1** 2 3 4 5 6 ... 8 Next

Genes with PP2 > 0.8 (Posterior probabilities of Model 2)

Hide

```
fData[fData$SECOND > 0.8, ]
```

	geneN...	dn_damaging_...	dn_lof_DD	dn_damaging_EE	dn_lof_EE	NO	BOTH
	<fctr>	<int>	<int>	<int>	<int>	<dbl>	<dbl>
14671	SCN1A	2	0	4	4	2.0179e-12	0.1174299

1 row | 1-8 of 9 columns

Use mTADA's results for single-trait analyses.

We can obtain single-trait results by summing PP1 and PP3 (Trait 1) or PP2 and PP3 (Trait 2).

Trait 1

Hide

```
fData[, 'pTrait1'] <- fData[, 'BOTH'] + fData[, 'FIRST']
fData1 <- fData[fData$pTrait1 > 0.8, ]
head(fData1[, c(1:5, 10)])
```

	geneNa...	dn_damaging_DD	dn_lof_DD	dn_damaging_EE	dn_lof_EE	pTrait1
	<fctr>	<int>	<int>	<int>	<int>	<dbl>
347	ADNP	1	19	0	0	1.0000000
447	AHDC1	0	8	0	0	1.0000000
681	ANKRD11	0	32	0	0	1.0000000
1000	ARID1A	1	2	0	0	0.9150887
1001	ARID1B	0	30	0	0	1.0000000
1002	ARID2	0	3	0	0	0.9977959

6 rows

Trait 2

Hide

```
fData[, 'pTrait2'] <- fData[, 'BOTH'] + fData[, 'SECOND']
fData2 <- fData[fData$pTrait2 > 0.8, ]
head(fData2[, c(1:5, 11)])
```

	geneNa...	dn_damaging_DD	dn_lof_DD	dn_damaging_EE	dn_lof_EE	pTrait2
	<fctr>	<int>	<int>	<int>	<int>	<dbl>
2348	CACNA1A	5	0	2	0	0.9957149
3201	CHD2	0	6	0	1	0.9356220
6254	GABBR2	2	0	2	0	0.9957601
6265	GABRB3	2	0	2	0	0.9974753
6610	GNAO1	4	1	2	0	0.9983683
7165	HECW2	5	1	1	0	0.8895286

6 rows

Other information

Some additional information can be obtained from mTADA's results.

Hide

```
pCI <- mTADAresults$pars ## Genetic parameters
piValue <- mTADAresults$probModel ## Posterior probabilities of genes for four models
mcmcResult <- mTADAresults$mcmcData ##MCMC results
```

The proportions of risk genes

piValue is a vector of π values. In the result below, pNO, pFIRST, pSECOND, and pBOTH are π_0, π_1, π_2 and π_3 respectively in **Table 1**.

Hide

```
piValue
```

```
        pNO      pFIRST     pSECOND       pBOTH 
0.961775704 0.022736406 0.008861466 0.006626424 
```

Estimated information of π_3.

Credible-interval information is from pCI.

Hide

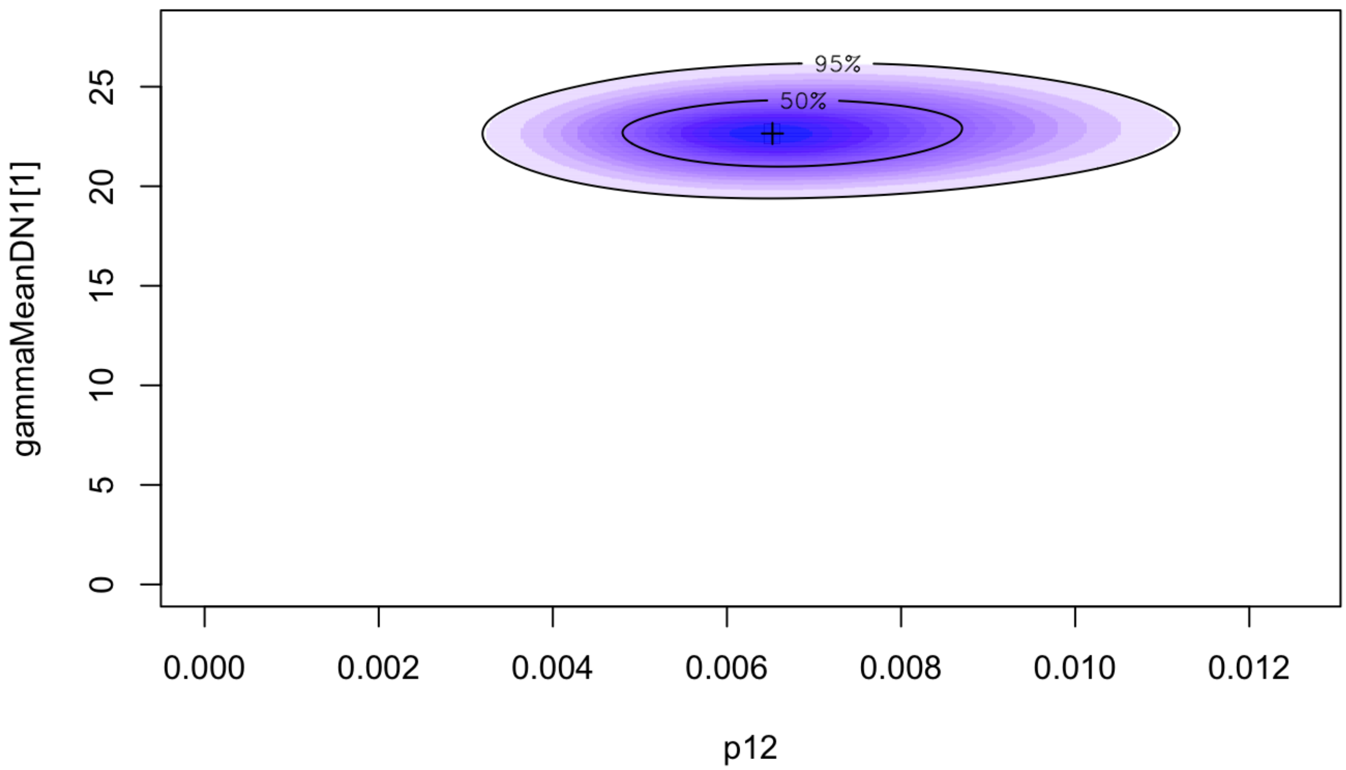
```
pCI ## Mode: estimated values; CI: credible interval with low (l) and upper (u) values
```

```
                         Mode         lCI        uCI
p12               0.006626424  0.003787886  0.0103368
gammaMeanDN1[1] 22.726850922 20.061165377 25.4284814
```

To check the convergent information of π_3 , we can visualize MCMC results.

Hide

```
## p12 is pi3 in the model
plotParHeatmap1(mcmcResult = mcmcResult, pars = c('p12', 'gammaMeanDN1[1]'))
```



Citation

mTADA: a framework for identifying risk genes from de novo mutations in multiple traits. Hoang T. Nguyen, Amanda Dobbyn, Ruth C. Brown, Brien P. Riley, Joseph Buxbaum, Dalila Pinto, Shaun M Purcell, Patrick F Sullivan8, Xin He, Eli A. Stahl.