TACTICAL tutorial

Jason M. Torres

April 16, 2020

This is a tutorial for the R package **TACTICAL** ("Tissue of ACTion scores for Investigating Complex trait Associations at Loci"). This package takes genetic associations or fine-mapped genetic credible sets and systemically integrates them with functional annotations to obtain *tissue of action* (TOA) scores that can guide mechanistic investigation of genetic signals from genome-wide association studies.

Suggested usage

TACTICAL provides a simple way to systematically integrate genetic information from trait-associated loci with functional genomic annotations and gene expression to obtain *tissue-of-action* (TOA) scores. These scores can be used to:

- classify signals to most likely tissues of action.
- prioritise genetic signals for experimental validation in a particular cell or tissue type.
- guide the identification of causal gene(s) - at specific loci - by informing which tissue(s) are most appropriate for analyses such as eQTL colocalisation or integration with chromatin conformation capture (3C) approaches.
- inform *process-specific* polygenic risk scores.

TACTICAL is simple to run and flexible; you need only provide the input data you are interested in evaluating. The data accepted include:

- `Genetic information`: Credible sets from bayesian fine-mapping or index variants from GWAS. If using index SNPs, we recommended using conditionally independent SNPs or LD-pruned SNPs.
- `Genomic annotations`: Any type of interval data that can be formatted in a BED file. This can include chromatin segmentation states, ChIP-seq peaks, chromatin accessible regions (i.e. DHS or ATAC-seq peaks), lowly/highly methylated regions, coding sequence, untranslated regions, etc.
- `Expression specificity scores`: When comparing across a set of tissues or cell types, you can provide expression specificity scores (described below) that inform the extent of tissue-specific gene expression in each evaluated tissue.

Although there are many possible data combinations the user may wish to explore, we offer a few suggestions:

1. As some annotations are more enriched for genome-wide significant SNPs (or SNP heritability), you may consider explicitly using these enrichment values as weights within TACTICAL. Although this package does not estimate enrichment *per se*, there are many software programs available that do, such as fgwas, GARFIELD, and GoShifter.
2. If you are interested in prioritising genetic signals within a particular tissue, then we suggest using as many relevant annotations as available (i.e. from a variety of molecular assays); though you may not want to include "repressed" or low-signal annotations in your tissue or cell-type of interest.
3. On the other hand, if you are interested in profiling signals across a range of candidate tissues, then we strongly recommend restricting the provided annotations to those are available for all evaluated tissues. For example, you may not want to use ATAC-seq peaks if they are only available for a subset of tissues.

Getting started

Prerequisites

The package was developed in R (version 3.6.0) and requires the following R packages:

- data.table (1.12.8)
- dplyr (0.8.4)
- GenomicRanges (1.36.1)
- devtools (2.2.2)
  - `install_github` function used for package installation
- **Note**: functions from ggplot2 (3.3.0) and gridExtra (2.3) are used to plot PCA output in this tutorial

Installation

The package can be directly installed from GitHub using this R command:

```
devtools::install_github("jmtorres138/TACTICAL")
```

We can then load the `TACTICAL` package within an R session:

```r
library("TACTICAL")
```

Input files

In order to obtain TOA scores from TACTICAL, the user must provide input files that contain the genetic association information and functional data they would like to integrate. For this tutorial, we will use the test datasets provided with the package installation.

SNP file

This text file contains the SNP-level genetic association information and must include columns with the names `SIGNAL`, `SNPID`, `CHR`, `POS`, and `VALUE`.

- **`SIGNAL`**: The unique character string identifier used for each GWAS index variant or fine-mapped credible set.
- **`SNPID`**: The unique character string identifier for each individual SNP.
- **`CHR`**: The chromosome of the SNP. The naming convention used to designate chromosome must be consistent across all input files (i.e. BED files, gene expression specificity file).
- **`POS`**: The physical position of each SNP. It is important that the physical coordinates in the SNP file and annotation files correspond to the same genomic build.

- **`VALUE`**: When using fine-mapped credible sets, this value corresponds to the posterior probability of association (PPA) for each credible SNP. When using (conditionally independent) GWAS index variants, this value must be set to 1.

For this tutorial, we will use the file `credible-input.txt` that includes 99% credible sets from a European GWAS meta-analysis of type 2 diabetes (Mahajan et al. 2018). This file contains all credible SNPs for the 101 99% credible sets with maximum credible set PPA ≥ 0.5. The complete dataset can be downloaded from the DIAGRAM website.

```
SIGNAL   SNPID           CHR      POS         VALUE
3_1      chr1:62579891   chr1     62579891    0.9960100
5_1      chr1:117532790  chr1     117532790   0.9721200
5_1      chr1:117531620  chr1     117531620   0.0131370
5_1      chr1:117529458  chr1     117529458   0.0086275
11_1     chr1:214159256  chr1     214159256   0.9996700
```

```
11_2      chr1:214150821  chr1      214150821   0.6437100
11_2      chr1:214150445  chr1      214150445   0.3562900
13_1      chr1:229672955  chr1      229672955   0.9999900
15_1      chr2:422144     chr2      422144      0.6214900
...       ...             ...       ...         ...
```

Note the convention used to designate credible sets in the `SIGNAL` column, where the number before the underscore indicates the GWAS locus number (i.e. GWAS associated region) and the number after the underscore indicates the signal number (i.e. conditionally indepedent GWAS association within the locus). From this information, we can see that the only signal for the third associated locus ("3_1") from the GWAS has been fine-mapped to a single credible SNP with PPA = 0.996, whereas the only signal for the fifth associated locus ("5_1") has been fine-mapped to three credible SNPs, with one SNP having a much higher PPA than the others. Also note that the eleventh associated locus has both a primary signal ("11_1") and a conditionally independent secondary signal ("11_2"), with the former being fine-mapped to a single SNP and the latter being fine-mapped to two SNPs.

Tissue path file

This text file contains two columns and does not include a header. The first column lists the names of the tissue and/or cell types with functional genomic annotations that will be integrated with GWAS information. It is important the the same character strings used in this column are also included in the tissue annotation file. The second column lists the complete paths the BED files that correspond to each tissue.

For this tutorial, we will incorporate chromatin state annotations for four diabetes relevant tissues that have been previously delineated by the Parker laboratory (Varshney et al. 2017). The complete set of annotations can be downloaded here.

```
liver     Liver.chromatinStates.bed
islet     Islets.chromatinStates.bed
muscle    SkeletalMuscle.chromatinStates.bed
adipose   Adipose.chromatinStates.bed
```

BED file(s)

The expected format for each set of genomic annotations is a BED file. It is important that there is no header and that the first four columns correspond to annotation chromsome, annotation start position, annotation end position, and annotation name. More than one type of annotation can be included in the same file. Also note that this file is 0-based.

```
chr1  0       13200   18_Quiescent/low_signal
chr1  13200   15800   6_Weak_transcription
chr1  15800   16800   5_Strong_transcription
chr1  16800   19600   6_Weak_transcription
chr1  19600   567400  18_Quiescent/low_signal
chr1  567400  567800  2_Weak_TSS
...   ...     ...     ...
```

BED files for genomic annotations corresponding to chromatin activity (e.g. ChIP-seq on histone PTMs, chromatin states), chromatin accessibility (e.g. DHS sites and ATAC-seq), and transcription factor binding sites (e.g. ChIP-seq) can be obtained from the ENCODE Project or Roadmap Epigenomics Project.

Tissue annotation file

This text file indicates how each tissue-level annotation should be handled when obtaining tissue score vectors for each SNP. It contains three columns and no header. The first column lists the tissue or cell type, the second column lists in the annotation name, and the third column lists the annotation weight. We have previoulsy used genome-wide fold enrichments for trait-associated variants, obtained from the program `fgwas`, to weight the relative importance of each tissue-level annotation.

```
liver   1_Active_TSS          6.76
liver   9_Active_enhancer_1   3.02
islet   1_Active_TSS          6.90
islet   9_Active_enhancer_1   7.15
muscle  1_Active_TSS          5.50
muscle  9_Active_enhancer_1   2.07
...     ...                   ...
```

There are several programs available that can provide genome-wide enrichment values for genomic annotations, such as fgwas and GARFIELD.

In order to perform an *unweighted* anlysis that handles each tissue-level annotation equally, simply set the third column value to 1.

Genomic path file

This text file is similar to the tissue path file, the key difference is that it lists genomic annotations that do not vary across tissues (e.g. coding sequence). It contains two columns and does not include a header. The first column lists the names of the genomic annotations and the second column lists the complete paths the corresponding BED files.

The provided sample file only contains one entry that corresponds to coding sequence:

```
coding    cds.bed
```

Genomic annotation file

This text contains two columns and does not include a header. The first column lists the names of genomic annotations and the second column lists the annotation weight (i.e. genome-wide fold enrichment values). In order to perform an *unweighted* anlysis, set the second column value to 1.

```
coding  6.01
```

Gene expression specificity score file (Optional)

Although the genomic annotations in the genomic annotation file may not vary across tissues *per se*, they may correspond to features that do vary between tissues, as is the case between coding regions and the expression patterns of their encoded genes. To account for the relative expression levels of genes across relevant tissues, the user can provide an input file of expression specificity scores, that can be obtained using this formula:

$$\varepsilon_{g,t} = \frac{\text{med}(\text{expression}_{g,t})}{\sum_{x \in T} \text{med}(\text{expression}_{g,x})}$$

where $\text{med}(\text{expression}_{g,t})$ is the median expression of gene g in tissue t and T is the set of tissues evaluated in the tissue annotation file. These values range from zero to one, where zero indicates that gene g is not expressed in tissue t, whereas one indicates that gene g is only expressed in tissue t, relative to the other evaluated tissues.

The input text file contains columns of expression specificity scores corresponding to each evaluated tissue, and are accordingly named (the tissue names must be the same as those provided in the tissue annotation file). The first five columns are `FEATURE_ID`, `FEATURE_NAME`, `CHR`, `START`, and `END` as shown in the provided sample file:

```
FEATURE_ID  FEATURE_NAME  CHR   START     END       islet   muscle  adipose   liver
Gene1       ZFYVE28       chr4  2341180   2341358   0.352   0.076   0.428     0.142
Gene2       TTPAL         chr20 43109007  43109084  0.292   0.041   0.281     0.385
Gene3       NRARP         chr9  140196036 140196380 0.521   0.056   0.368     0.054
...         ...           ...   ...       ...       ...     ...     ...       ...
```

Note that this file is not required to calculate TOA scores, but is recommended.

Step 1: Annotating GWAS associations or fine-mapped bayesian credible sets

For this tutorial, we will use the test datasets provided with the package installation. For simplicity, we will change our current working directory to the directory containing the datasets. When running on your own datasets, please make sure that the full paths to your BED files are provided in the annotation path files.

For the first step, we will using the `annotate_snps` function to annotate the SNPs in the SNP file to the annotations provided in the annotation files:

```
data.directory <- system.file("extdata",package="TACTICAL")
setwd(data.directory)

snp.df <- annotate_snps(snp_file = "credible-input.txt",
                        tissue_path_file = "tissue-file-paths.txt",
                        tissue_annotation_file = "tissue-annotations.txt",
                        genomic_path_file = "genomic-file-paths.txt",
                        genomic_annotation_file = "genomic-annotations.txt")
```

The output of this function is a data frame that includes all columns in the snp file in addition to columns corresponding to each tissue-level and genomic annotation. The value in this column is either a zero (SNP does not map to annotation) or one (SNP maps to annotation).

The example output file has 42 columns, 37 of which correspond to the evaluated annotations. Let's also take a look at the first and last annotation column for the first five SNPs:

```
dim(snp.df)
#> [1] 448  42
snp.df[1:5,c(1:6,42)]
#>    SIGNAL          SNPID  CHR       POS     VALUE liver.6_Weak_transcription
#> 1:    3_1  chr1:62579891 chr1  62579891 0.9960100                          1
#> 2:    5_1 chr1:117532790 chr1 117532790 0.9721200                          1
#> 3:    5_1 chr1:117531620 chr1 117531620 0.0131370                          1
#> 4:    5_1 chr1:117529458 chr1 117529458 0.0086275                          1
#> 5:   11_2 chr1:214150821 chr1 214150821 0.6437100                          0
#>    coding
#> 1:      1
#> 2:      0
#> 3:      0
#> 4:      1
#> 5:      0
```

Step 2: Calculating tissue score vectors for each SNP

Once we have annotated our SNPs with the creatively named `annotate_snps` function, we will then calculate tissue vectors for each SNP in the outputted data frame with the `calculate_tissue_vectors` function. The functions arguments include:

- **`snp.annotated.df`**: The output dataframe from the `annotate_snps` function.
- **`tissue_annotation_file`**: The name of the tissue annotation file as a character string.
- **`genomic_annotation_file`**: The name of the genomic annotation file as a character string.
- **`ess.annot`**: If expression specificity scores are provided, the name of the corresponding genomic annotation (e.g. "coding"). Otherwise, this argument is set to `NULL`.
- **`ess.file`**: If expression specificity scores are provided, the name of the expression specificity score file as a character string. Otherwise, this argument is set to `NULL`.

We can calculate tissue vectors using the following command:

```
data.directory <- system.file("extdata",package="TACTICAL")
setwd(data.directory)

tvec.df <- calculate_tissue_vectors(snp.annotated.df = snp.df,
                                    tissue_annotation_file = "tissue-annotations.txt",
                                    genomic_annotation_file = "genomic-annotations.txt",
                                    ess.annot = "coding",
                                    ess.file = "gene-expression-specificity-scores.txt")
```

The output of this function is a dataframe that, for each credible SNP, includes a single tissue score corresponding to each tissue:

```
dim(tvec.df)
#> [1] 448   9
tvec.df[1:5,]
#>    SIGNAL           SNPID  CHR       POS     VALUE     adipose       islet
#> 1:    3_1  chr1:62579891 chr1  62579891 0.9960100 0.153910838 0.448993912
#> 2:    5_1 chr1:117532790 chr1 117532790 0.9721200 0.321864601 0.196646828
#> 3:    5_1 chr1:117531620 chr1 117531620 0.0131370 0.002960162 0.003077626
#> 4:    5_1 chr1:117529458 chr1 117529458 0.0086275 0.001900228 0.005410466
#> 5:   11_2 chr1:214150821 chr1 214150821 0.6437100 0.000000000 0.452565228
#>          liver       muscle
#> 1: 0.168808106 0.2242971438
#> 2: 0.142686309 0.3109222617
#> 3: 0.002233116 0.0048660966
#> 4: 0.001188004 0.0001288016
#> 5: 0.191144772 0.0000000000
```

Step 3: Calculating tissue of action (TOA) scores for each genetic signal (i.e. independent GWAS association or credible set)

We will now take the dataframe outputted from the `calculate_toa_scores` function and input it into the `calculate_toa_scores` function to obtain *tissue of action* (TOA) scores for each genetic signal (i.e. credible set when evaluating fine-mapped loci):

```
tscores.df <- calculate_toa_scores(snp.tissvec.df = tvec.df)
```

The output of this function is another dataframe where each row corresponds to a genetic signal and contains columns indicate the TOA scores for each evaluated tissue:

```
dim(tscores.df)
#> [1] 101   6
tscores.df[1:5,]
#>   SIGNAL adipose  islet  liver muscle unclassified
#> 1    3_1  0.1539 0.4490 0.1688 0.2243            0
#> 2    5_1  0.3267 0.2051 0.1461 0.3159            0
#> 3   11_2  0.0000 0.7031 0.2969 0.0000            0
#> 4   11_1  0.0000 0.5048 0.4948 0.0000            0
#> 5   13_1  0.2205 0.3451 0.2005 0.2339            0
```

Note that this dataframe also includes a column `unclassified` that indicates how much of the signal value is not accounted by by the evaluated annotations. When using fine-mapping data, this column corresponds to the cummulative PPA attributable to credible SNPs that do not map to any of the provided annotations.

Step 4: Apply a rule-based classifier to assign signals to tissue and/or cell types

Once we've obtained TOA scores, you may wish to use them to assign individuals signals to tissues. Although there are numerous classification methods you can apply, we provide a function `tissue_classifier` that applies a simple rule-based classifier for assigning signals to tissues using TOA scores. The three arguments for this function are:

- **`toa.df`**: The dataframe of TOA scores outputted from the `calculate_toa_scores` function.
- **`tissue_threshold`**: The minimum TOA score threshold that must be met for a signal to be assigned to a tissue. The default value for this argument is 0.2.
- **`shared_threshold`**: The difference in TOA score threshold that determines whether a signal will be assigned as a "shared" signal (i.e. two or more TOA scores fall within this range). This value is set to 0.1 by default.

```
classified.df <- tissue_classifier(toa.df=tscores.df, tissue_threshold = 0.2, shared_threshold = 0.1)
```

The output of this function is a dataframe that provides the tissue assignment for each signal. Note that for signals classified as "shared", the set of responsible tissues are indicated in the third column:

```
dim(classified.df)
#> [1] 101   3
classified.df[1:5,]
#>   SIGNAL classification        tissues
#> 1    3_1          islet          islet
#> 2    5_1         shared adipose,muscle
#> 3   11_2          islet          islet
#> 4   11_1         shared    islet,liver
#> 5   13_1          islet          islet
```

Principal components analysis (PCA)

We are now able to further analyse the TOA scores and profile genetic signals. For example, we can perform a principal components anlaysis on the set of classified signals (86/101) using the following code:

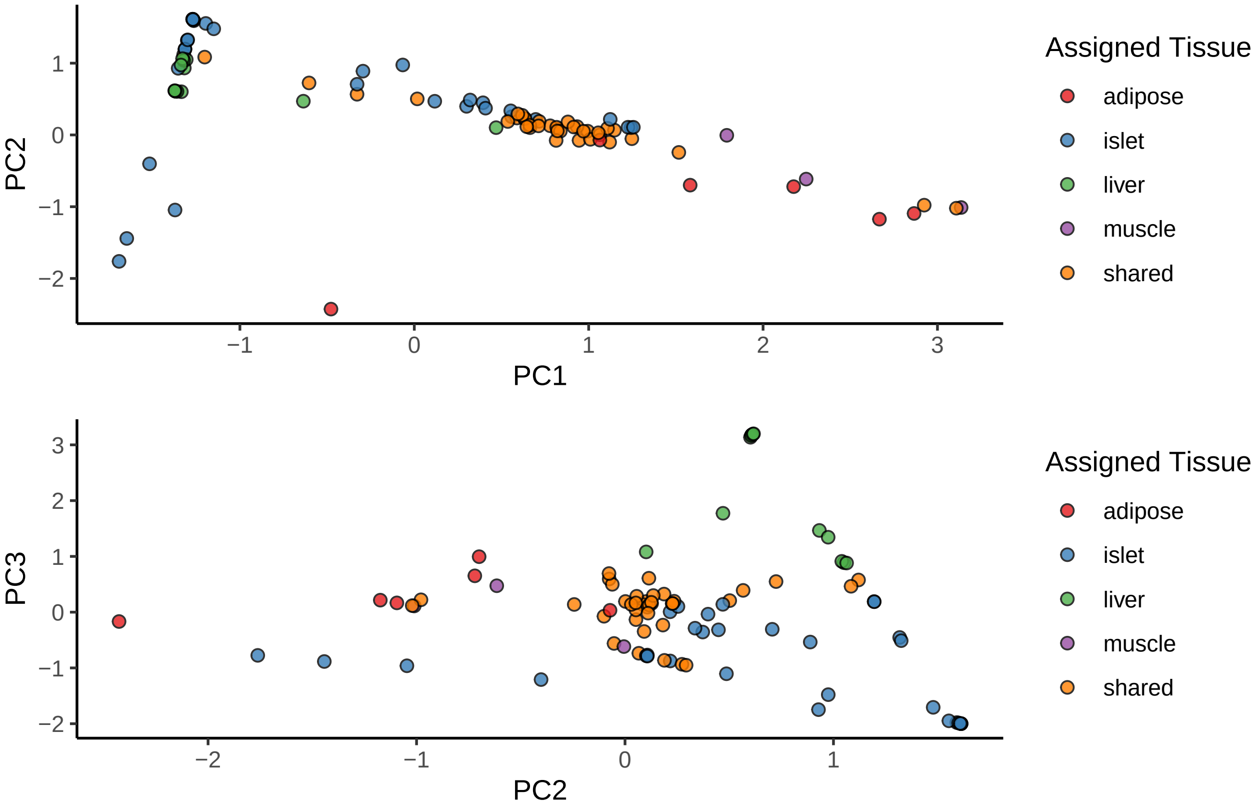
```
library("ggplot2")
library("gridExtra")

pr.out <- prcomp(dplyr::select(tscores.df,-one_of("SIGNAL")),scale=TRUE)
pca.df <- as.data.frame(pr.out$x)
pca.df$classification <- classified.df$classification

pltA <- ggplot(data=dplyr::filter(pca.df,classification!="unclassified"),
                         aes(x=PC1,y=PC2,fill=classification)) +
               geom_point(shape=21,alpha=0.8,color="black",size=2) +
               scale_fill_brewer(palette = "Set1",name="Assigned Tissue") +
               theme_classic()

pltB <- ggplot(data=dplyr::filter(pca.df,classification!="unclassified"),
                         aes(x=PC2,y=PC3,fill=classification)) +
               geom_point(shape=21,alpha=0.8,color="black",size=2) +
               scale_fill_brewer(palette = "Set1",name="Assigned Tissue") +
               theme_classic()

grid.arrange(pltA,pltB,nrow=2)
```



References

Mahajan, Anubha, Daniel Taliun, Matthias Thurner, Neil R Robertson, Jason M Torres, N William Rayner, Anthony J Payne, et al. 2018. "Fine-mapping type 2 diabetes loci to single-variant resolution using high-density imputation and islet-specific epigenome maps." *Nature Genetics* 50 (11): 1505–13. doi:10.1038/s41588-018-0241-6.

Varshney, Arushi, Laura J Scott, Ryan P Welch, Michael R Erdos, Peter S Chines, Narisu Narisu, Ricardo D'O. Albanus, et al. 2017. "Genetic regulatory signatures underlying islet gene expression and type 2 diabetes." *Proceedings of the National Academy of Sciences* 114 (9): 2301–6. doi:10.1073/pnas.1621192114.