Package: GenomicSEM (via r-universe)

October 4, 2024

Type Package

Title Structural equation modeling based on GWAS summary statistics

Version 0.0.5

Author Andrew Grotzinger, Matthijs van der Zee, Mijke Rhemtulla, Hill Ip, Michel Nivard, Elliot Tucker-Drob

Maintainer Andrew Grotzinger <agrotzin@utexas.edu>

Description Later

License GPL-3.0

Encoding UTF-8

LazyData true

Imports plyr,e1071,readr,gdata,lavaan,doParallel,parallel,foreach,data.table,stringr,splitstackshape,Matrix,R.utils,dplyr,iterators,Rc

Repository https://mrcieu.r-universe.dev

RemoteUrl https://github.com/GenomicSEM/GenomicSEM

RemoteRef HEAD

RemoteSha 199d16f5e45c040a25ee0c885338ebd667277135

Contents

addSNPs *Combine LDSC and summary statistic output for multivariate GWAS using GenomicSEM*

Description

Function to expand the S and V matrices to include SNP effects for multivariate GWAS in GenomicSEM

Usage

```
addSNPs(covstruc, SNPs,SNPSE=FALSE,parallel=TRUE,cores=NULL,GC="standard", ...)
```

Arguments

`covstruc` Output from Genomic SEM ‘ldsc‘ function

`SNPs` Summary statistics file created using the ’sumstats’ function

`SNPSE` Whether the user wants to provide a different standard error (SE) of the SNP variance than the package default. The default is to use .0005 to reflect the fact that the SNP SE is assumed to be population fixed.

`parallel` addSNPs automatically uses mclapply to create the S and V matrices in parallel. Sometimes running in parallel can cause memory issues within the computing cores. If this is the case, the parallel argument can be set to FALSE, and addSNPs will create the S and V matrices serially.

`cores` addSNPs automatically uses mclapply to create the S and V matrices in parallel. If the user does not provide an argument to the cores option, then addSNPs will automatically use one less than the total number of cores available.

`GC` Level of Genomic Control (GC) you want the function to use. The default is ’standard’ which adjusts the univariate GWAS standard errors by multiplying them by the square root of the univariate LDSC intercept. Additional options include ’conserv’ which corrects standard errors using the univariate LDSC intercept, and ’none’ which does not correct the standard errors.

Value

The function expands the S and V matrices to include SNP effects. As many S and V matrices will be created as there are rows in the summary statistics file (i.e., one S and V matrix per SNP). The function returns a list with 3 named entries:

`V_Full` variance covariance matrix of the parameter estimates in S that includes an individual SNP effect

S_Full genetic covaraiance matrix including individual SNP effect

RS A list containing relevant genetic information (e.g., rsID, basepair, A1/A2) to be appended to the output from other functions (e.g., userGWAS)

commonfactor *Run common factor model on genetic covariance and sampling covariance matrix*

Description

Function to run a common factor model based on output from multivariable LDSC

Usage

```
commonfactor(covstruc,estimation="DWLS", ...)
```

Arguments

covstruc Output from the multivariable LDSC function of Genomic SEM

estimation Options are either Diagonally Weighted Least Squares ("DWLS"; the default) or Maximum Likelihood ("ML")

Value

The function estimates a common factor model, along with model fit indices, using output from GenomicSEM LDSC. The function returns a list with 2 named entries

modelfit The model fit results (e.g., model chi-square, AIC, CFI) from the specified model.

results Parameter estimates and sandwich corrected standard errors from the specified model.

commonfactorGWAS *Estimate SNP effects on a single common factor*

Description

Function to obtain SNP effects on common factor along with index of SNP heterogeneity

Usage

```
commonfactorGWAS(covstruc=NULL,SNPs=NULL,estimation="DWLS",cores=NULL,toler=FALSE,SNPSE=FALSE,paral
```

Arguments

`covstruc` Output from Genomic SEM ‘ldsc‘ function

`SNPs` Summary statistics file created using the ’sumstats’ function

`estimation` The estimation method to be used when running the factor model. The options are Diagonally Weighted Least Squares ("DWLS", this is the default) or Maximum Likelihood ("ML")

`cores` The number of cores to use on your computer for parallel processing. If no number is provided, the default is to use one less core then is available on your computer

`toler` The tolerance to use for matrix inversion.

`SNPSE` Whether the user wants to provide a different standard error (SE) of the SNP variance than the package default. The default is to use .0005 to reflect the fact that the SNP SE is assumed to be population fixed.

`parallel` Whether the function should run using parallel or serial processing. Default = TRUE

`GC` Level of Genomic Control (GC) you want the function to use. The default is ’standard’ which adjusts the univariate GWAS standard errors by multiplying them by the square root of the univariate LDSC intercept. Additional options include ’conserv’ which corrects standard errors using the univariate LDSC intercept, and ’none’ which does not correct the standard errors.

`MPI` Whether the function should use multi-node processing (i.e., MPI). Please note that this should only be used on a computing cluster on which the R package Rmpi is already installed.

`TWAS` Whether the function is being used to estimate a multivariate TWAS using read_fusion output for the SNPs argument.

`smooth_check` Whether the function should save the consequent largest Zstatistic difference between the pre and post-smooth matrices.

Value

The function outputs a series of SNP effects with their SEs and estimate of QSNP (the heterogeneity index). The output is a single object.

`enrich` *Estimate enrichment of model parameter for a user specified model*

Description

Function to take output from multivariable S-LDSC and estimate enrichment of model parameter for user specified model

Usage

```
enrich(s_covstruc, model = "",params,fix= "regressions",std.lv=FALSE,rm_flank=TRUE,tau=FALSE,base=TRU
```

Arguments

`s_covstruc` Output from the multivariable S-LDSC function of Genomic SEM (s_ldsc)

`model` Model to be specified using lavaan notation

`params` Parameters of interest to be examined for enrichment (e.g., factor variances).

`fix` What components of the model should be fixed for follow-up enrichment models. Default = "regressions", which will fix all regression parameters.

`std.lv` Optional argument to denote whether all latent variables are standardized using unit variance identification (default = FALSE)

`rm_flank` Optional argument to denote whether flanking window annotations should automatically be removed from output (default = TRUE)

`tau` Optional argument to denote whether the user wants to use the tau genetic covariance matrices, as opposed to the default zero-order matrices, for estimation of enrichment (default = FALSE)

`base` Optional argument to denote whether the user wants to include the full model output from the genome-wide (i.e., baseline) matrix (default = TRUE)

`toler` Optional argument to manually set tolerance for matrix inverison.

`fixparam` Optional argument to manually fix paramters when estimating the model within annotations

Value

Function to take output from multivariable S-LDSC and estimate enrichment of model parameter for user specified model

`hdl` *estimate a genetic covariance matrix using High Definition Likelihood (HDL) estimation in R*

Description

Function to run HDL (https://github.com/zhenin/HDL) to compute the genetic covariance between a series of traits based on genome wide summary statistics obtained from GWAS. The results generate by this function are necessary and sufficient to facilitate the fit of structural equation models (or other multivariate models) to the genetic covariance matrix. HDL is more powerfull than LDSC but if the LD structure in the reference file mismatches the GWAS LD structure, LDSC seems to perfrom better, expescially for estiamtes of heritability. For medium samples (N > 50.000) with moderate SNP-h2 (snp h2 > 0.07) where the LD structure isnt similar we would recomend ldsc, especially for GWAS. If you have small GWAS (N < 25.000) the extra power HDL provides is worth the downward bias in snp h2 estimates relative to ldsc.

Usage

```
hdl(traits,sample.prev,population.prev,LDpath,Nref,trait.names=NULL,method, ...)
```

Arguments

`traits` A vector of strings which point to munged files for trait you want to include in a Genomic SEM model. the HDL function works with standard munged files

`sample.prev` A vector of sample prevalences for dichotomous traits and `NA` for continous traits

`population.prev` A vector of population prevalences for dichotomous traits and `NA` for continous traits

`LDpath` String which contains the path to the folder in which the LD matrices used in the analysis are located. Expects LD matirices formated as required by the original HDL software.

`Nref` Sample size of the reference file, default is 335265

`trait.names` A character vector specifying how the traits should be named in the genetic covariance (S) matrix. These variable names can subsequently be used in later steps for model specification. If no value is provided, the function will automatically name the variables using the generic from of V1-VX.

`method` sting, either "piecewise" which estimates the heritability or genetic covariance locally in chunks across the genome and then sums these estimates, or "jackknife" which uses a genoem wide estiamte and uses a jackknife estimator for the variance of the parameter. defaults to "piecewise" the original HDL implementation is equal to "jackknife"

Value

The function returns a list with 3 named entries

`S` estimated genetic covariance matrix

`V` variance covariance matrix of the parameter estimates in S

`I` matrix containing the "cross trait intercepts", or the error covariance between traits induced by overlap, in terms of subjects, between the GWASes on which the analyses are based

References

Ning, Z., Pawitan, Y. & Shen, X. High-definition likelihood inference of genetic correlations across human complex traits. Nat Genet (2020).

`ldsc` *build a convariance structure using LD score regression in R*

Description

Function to run LD score regression (https://github.com/bulik/ldsc) to compute the genetic covariance between a series of traits based on genome wide summary statistics obtained from GWAS. The results generate by this function are necessary and sufficient to facilitate the fit of structural equation models (or other multivariate models) to the genetic covariance matrix.

Usage

```
ldsc(traits,sample.prev,population.prev,ld,wld,trait.names=NULL, sep_weights = FALSE,chr=22,n.blocks
```

Arguments

`traits` A vector of strings which point to LDSC munged files for trait you want to include in a Genomic SEM model.

`sample.prev` A vector of sample prevalences for dichotomous traits and `NA` for continous traits

`population.prev`
A vector of population prevalences for dichotomous traits and `NA` for continous traits

`ld` String which contains the path to the folder in which the LD scores used in the analysis are located. Expects LD scores formated as required by the original LD score regression software.

`wld` String which contains the path to the folder in which the LD score weights used in the analysis are located. Expects LD scores formated as required by the original LD score regression software.

`trait.names` A character vector specifying how the traits should be named in the genetic covariance (S) matrix. These variable names can subsequently be used in later steps for model specification. If no value is provided, the function will automatically name the variables using the generic from of V1-VX.

`sep_weights` Logical which indicates wheter the weights are different form the LD scores used for the regression, defaults to FALSE.

`chr` number of chromosomes over which the LDSC weights are split, defalts to 22 (Human) but can be switched for other species

`n.blocks` Number of blocks to use for the jacknive procedure which is used to estiamte entries in V, higher values will be optimal if you have a large number of variables and also slower, defaults to 200

`ldsc.log` How you want to name your .log file for ldsc. The default is NULL as the package will automatically name the log based in the file names unless a log file name is provided to this arugment.

`stand` Whether you want the package to also output a genetic correlation and sampling correlation matrix. Default is FALSE.

`select` Whether you want the package to estimate LDSC using only even or odd chromosomes by setting select to "ODD" and "EVEN" respectively. It can also be set to a set of numbers, such as c(1,3,10), to run ldsc on a specific chromosome or chromosomes. Default is FALSE, in which case LDSC is estimated using all chromosomes.

`chisq.max` Maximum value of the squared Z statistics for a SNP that is considered in the LD-score regression. Default behaviour is to set to the maximum of 80 and N*0.001

Value

The function returns a list with 5 named entries

`S` estimated genetic covariance matrix

`V` variance covariance matrix of the parameter estimates in S

`I` matrix containing the "cross trait intercepts", or the error covariance between traits induced by overlap, in terms of subjects, between the GWASes on which the analyses are based

`N` a vector contsaining the sample size (for the genetic variances) and the geometric mean of sample sizes (i.e. sqrt(N1,N2)) between two samples for the covariances

`m` number of SNPs used to compute the LD scores with.

`localSRMD` *localSRMD for Genomic measurement invariance models*

Description

Given a set of parameter values from Genomic SEM models, calculate the extent to which groups differ on these parameters.

Usage

```
localSRMD(unconstrained, constrained, lhsvar, rhsvar, ...)
```

Arguments

`unconstrained` A vector of parameter values from an unconstrained structural equation model, where focal parameters are estimated freely in each group.

`constrained` A vector of parameter values from a constrained structural equation model, where focal parameters are constrained to be equal across groups.

`lhsvar` A list containing the variances, in each group, of the variables in the usermodel() results column lhs

`rhsvar` A list containing the variances, in each group, of the variables in the usermodel() results column rhs

Value

The function returns the average standardized extent to which estimates from a constrained set of structural equation model parameters differ from those obtained when the same set of parameters are freely estimated.

multiSNP *Combine LDSC, summary statistic output, and LD information for models including multiple SNPs*

Description

Function to expand the S and V matrices to include multiple SNP effects in a single matrix, along with LD information across these SNPs.

Usage

```
multiSNP (covstruc, SNPs, LD, SNPSE = FALSE, SNPlist = NA, ...)
```

Arguments

`covstruc`	Output from Genomic SEM multivariable LDSC
`SNPs`	Summary statistics file created using the sumstats function
`LD`	Matrix of LD information across the SNPs. If only independent SNPs are being provided a matrix of 0s can be entered. Note that the function requires that A1 and A2 be included in the LD matrix column names (e.g., rs12345_A_T)
`SNPSE`	User provided SE of the SNP variance for entry in the V matrix. If no number is provided the package defaults to using .0005 to reflect a practically fixed population value taken from a reference panel
`SNPlist`	List of rsIDs if the user wishes to subset out a set of SNPs from a full set of summary statistics

Value

The function expands the S and V matrices to include multiple SNP effects. These matrices include LD information across the SNPs.

munge *Clean and munge files to enable LD score regression*

Description

Function to process GWAS summary statistis files and prepair them for LD score regression

Usage

```
munge(files,hm3,trait.names=NULL,N,info.filter = .9,maf.filter=0.01, column.names=list(),parallel=FA
```

Arguments

`files` A vector of file names, files must be located in the working directory, or a path must be provided.

`hm3` A file of SNPs with A1, A2 and rsID used to allign alleles across traits. We suggest using an (UNZIPPED) file of HAPMAP3 SNPs with some basic cleaning applied (e.g., MHC region removed) that is supplied and created by the original LD score regression developers and available here: https://data.broadinstitute.org/alkesgroup/LDSCORE/

`trait.names` A vector of trait names which will be used as names for the munged files

`N` A vector of sample size

`info.filter` Numeric value which is used as a lower bound for inputation quality (INFO)

`maf.filter` Numeric value used as a lower bound for minor allel frequency

`column.names` Optional list detailing which columns represent, SNP, MAF, etc. e.g. list(SNP=my_snp_column)

`parallel` Indicates whether munge should process the summary statistics files in parallel or serial fashion. Default is TRUE, indicating that it will run in parallel.

`cores` Indicates how many cores to use when running in parallel. The default is NULL, in which case munge will use 1 less than the total number of cores available in the local environment.

`overwrite` Indicates whether existing .sumstats.gz files should be overwritten

Value

The function writes files of the ".sumstats" format, which can be used to estimate SNP heritability and genetic covariance using the ldsc() function. The function will also output a .log file that should be examined to ensure that column names are being interpret correctly.

`read_fusion` *Format univariate FUSION TWAS output across multiple traits for subsequent use in a multivariate TWAS [T-SEM]*

Description

Function to format TWAS summary statistics for T-SEM

Usage

```
read_fusion(files,trait.names=NULL,binary=NULL,N=NULL,perm=FALSE, ...)
```

Arguments

`files` List of FUSION files.

`trait.names` What names to use when naming the beta and SE columns for each trait

`binary` Vector specifying whether each trait is binary (TRUE) or continuous (FALSE)

`N` The sample size to use when backing out the betas and SEs from FUSION Z-statistics. This should reflect the sum of effective sample size for binary traits

`perm` Whether you want to use the permutation test p-values from FUSION. Default = FALSE

Value

The function combines and formats FUSION TWAS summary statistics for subsequent T-SEM analyses. The function returns a single data.frame.

`sumstats` *Allign summary statistics from univariate GWAS for a GWAS in GenomicSEM*

Description

Function to process GWAS summary statistics files and prepare them for a GWAS in genomicSEM

Usage

```
sumstats(files,ref,trait.names=NULL,se.logit,OLS=NULL,linprob=NULL,N=NULL,betas=NULL,info.filter = .
      keep.indel=FALSE,parallel=FALSE,cores=NULL,ambig=FALSE,direct.filter=FALSE, ...)
```

Arguments

`files` a vector of file names, files must be located in the working directory, or a path must be provided.

`ref` A reference file of SNPs to keep in your GWAS, one based on 1000 genomes phase 3 is provided.

`trait.names` a vector of trait names which will be used as names for the munged files

`se.logit` a logical vector indicating whether the standard errors in each set of summary statistics is on the logit scale

`OLS` a logical vector indicating whether the GWAS was for a continuous trait and used OLS (or a LMM)

`linprob` a logical vector indicating whether the GWAS is a binary outcome with only Z-statistics or was analyzed using a linear probability model i.e. a dichotomous trait using OLS (or a LMM)

`N` A vector of total sample sizes for continuous traits and the sum of effective sample sizes for binary traits `betas` A vector of column names of betas for continuous traits that are known to have been standardized prior to running the GWAS

`N` A vector of sample size

`info.filter` Numeric value which is used as a lower bound for imputation quality (INFO)

`maf.filter` Numeric value used as a lower bound for minor allele frequency

`keep.indel` Indicates whether insertion-deletion mutations (indels) should be included in your summary statistics. The default is FALSE.

`parallel` Indicates whether sumstats should process the summary statistics files in parallel or serial fashion. Default is TRUE, indicating that it will run in parallel.

`cores` Indicates how many cores to use when running in parallel. The default is NULL, in which case sumstats will use 1 less than the total number of cores available in the local environment.

`ambig` Indicates whether strand ambiguous SNPs should be removed from output.

`direct.filter` Indicates whether SNPs that have missing information for more than half of contributing cohorts, as indicated by missing information in the direction column, should be removed.

Value

The function ensures the SNPs in each file are aligned to the same reference allele, it attempts to filter strand issues, it retains SNPs present in the reference file. The function can deal with GWAS of continous traits, dichotomous traits using logistic regression and even dichotomous traits using (misspecified) OLS regression or a mixed model. The function returns .log files that should be inspected to ensure that all column names were appropriately interpreted.

s_ldsc *Estimate genetic covariance matrices within functional annotations using multivariable Stratified LD score regression*

Description

Function to run Stratified LD score regression.

Usage

```
s_ldsc(traits,sample.prev=NULL,population.prev=NULL,ld,wld,frq,trait.names=NULL,n.blocks=200,ldsc.l
```

Arguments

`traits` A vector of file names which point to LDSC munged files for trait you want to include.

`sample.prev` A vector of sample prevalences for dichotomous traits and `NA` for continous traits. Default = NULL.

population.prev

A vector of population prevalences for dichotomous traits and NA for continous traits. Default = NULL.

ld
A folder (or folders) of partitioned LD scores used as the independent variable in S-LDSC.

wld
A folder of non-partitioned LD scores used as regression weights.

frq
A folder of allele frequency files.

trait.names
A character vector specifying how the traits should be named. These variable names can subsequently be used in later steps for model specification.

n.blocks
Number of blocks to use for the jacknive procedure which is used to estiamte entries in V, higher values will be optimal if you have a large number of variables and also slower, defaults to 200.

ldsc.log
What to name the .log file if you want to overrride default to name file based on file names used as input.

exclude_cont
Whether to exclude continuous annotations from S-LDSC estimation.

Value

The function returns a list with 9 named entries

S
The zero-order genetic covariance matrices for each annotation.

V
The zero-order sampling covariance matrices for each annotation.

S_Tau
The tau matrices for each annotation.

V_Tau
The tau sampling covariance matrices for each annotation.

I
matrix containing the "cross trait intercepts", or the error covariance between traits induced by overlap, in terms of subjects, between the GWASes on which the analyses are based

N
a vector contsaining the sample size (for the genetic variances) and the geometric mean of sample sizes (i.e. sqrt(N1,N2)) between two samples for the covariances

m
number of SNPs used to compute the LD scores with.

Prop
The proportional size of each annotation relative to the annotation containing all SNPs.

Select
A data.frame that codes flanking window and continuous annotations as 2 and all other annotations as 1. This is used by the ‘enrich‘ function to exclude the flanking window and continuous annotations from enrichment estimates.

userGWAS — *Create genetic covariance matrices for individual SNPs and estimate SNP effects for a user specified multivariate GWAS*

Description

Function to obtain model estimates for a user-specified model across SNPs.

Usage

```
userGWAS(covstruc=NULL,SNPs=NULL,estimation="DWLS",model="",printwarn=TRUE,sub=FALSE,cores=NULL,tol
      parallel=TRUE,GC="standard",MPI=FALSE,smooth_check=FALSE,TWAS=FALSE,std.lv=FALSE,fix_measurem
```

Arguments

`covstruc` Output from Genomic SEM ‘ldsc‘ function

`SNPs` Summary statistics file created using the ’sumstats’ function

`estimation` The estimation method to be used when running the factor model. The options are Diagonally Weighted Least Squares ("DWLS", this is the default) or Maximum Likelihood ("ML")

`model` The user-specified model to use in model estimation using lavaan syntax. The SNP is referred to as ’SNP’ in the model.

`printwarn` Whether you want warnings and errors printed for each run. This can take up significant space across all SNPs, but the default is set to TRUE as these warnings may not be safe to ignore.

`sub` Whether you want to only output a piece of the model results (e.g., F1 ~ SNP). The argument takes a vector, as multiple pieces of the model result can be output.

`cores` Indicates how many cores to use when running in parallel. The default is NULL, in which case sumstats will use 1 less than the total number of cores available in the local environment.

`toler` The tolerance to use for matrix inversion.

`SNPSE` Whether the user wants to provide a different standard error (SE) of the SNP variance than the package default. The default is to use .0005 to reflect the fact that the SNP SE is assumed to be population fixed.

`parallel` Whether the function should run using parallel or serial processing. Default = TRUE

`GC` Level of Genomic Control (GC) you want the function to use. The default is ’standard’ which adjusts the univariate GWAS standard errors by multiplying them by the square root of the univariate LDSC intercept. Additional options include ’conserv’ which corrects standard errors using the univariate LDSC intercept, and ’none’ which does not correct the standard errors.

`MPI` Whether the function should use multi-node processing (i.e., MPI). Please note that this should only be used on a computing cluster on which the R package Rmpi is already installed.

`smooth_check` Whether the function should save the consequent largest Z-statistic difference between the pre and post-smooth matrices.

`TWAS` Whether the function is being used to estimate a multivariate TWAS using read_fusion output for the SNPs argument.

`std.lv` Optional argument to denote whether all latent variables are standardized using unit variance identification (default = FALSE)

`fix_measurement` Optional argument to denote whether the measurement model should be fixed across all SNPs (default = TRUE)

Value

The function outputs results from the multivariate GWAS. If the sub argument is used, it will output as many list objects as there are sub objects requested. If the sub argument is FALSE (as is the package default), the function will ouput as many list objects as there are SNPs.

`usermodel` *Run user specified model on LDSC output*

Description

Function to run a user specified model based on output from multivariable LDSC

Usage

```
usermodel(covstruc,estimation="DWLS", model = "", CFIcalc=TRUE,std.lv=FALSE,imp_cov=FALSE,fix_resid=T
```

Arguments

`covstruc` Output from the multivariable LDSC function of Genomic SEM

`estimation` Options are either Diagonally Weighted Least Squares ("DWLS"; the default) or Maximum Likelihood ("ML")

`model` Model to be specified using lavaan notation

`CFIcalc` Optional argument to denote whether CFI is being requested (default = TRUE). In some cases the estimation of the independent (i.e., Null) model for calculation of CFI can be time consuming. If the funciton seems to be stuck on this step, we would suggest re-running with this option set to FALSE

`std.lv` Optional argument to denote whether all latent variables are standardized using unit variance identification (default = FALSE)

`imp_cov` Optional argument to denote whether the user wants the model implied and residual covariance matrix included in the usermodel output (default = FALSE)

`fix_resid`	Optional argument to denote whether the user wants Genomic SEM to try troubleshooting a model that does not converge by fixing residual variances to be above 0 (default = TRUE)
`toler`	Optional argument to set lower tolerance for matrix inversion used to produce sadnwich corrected standard errors. (default = FALSE)

Value

The function estimates a user-specified model, along with model fit indices, using output from GenomicSEM LDSC.

`write.model` *Automate writing model syntax using EFA output*

Description

Function to automate writing model syntax based on EFA loadings. This is most likely to be useful when examining larger numbers of traits (e.g., > 10).

Usage

```
write.model(Loadings,S_LD,cutoff,fix_resid=TRUE,bifactor=FALSE,mustload=FALSE,common=FALSE, ...)
```

Arguments

`Loadings`	The matrix of EFA loadings. Note that the number of columns in this matrix determines how many factors are specifeid in the model.
`S_LD`	The LDSC genetic covariance matrix
`cutoff`	The EFA standardized loadings cutoff to determine which traits should load on a factor
`fix_resid`	Whether to apply constraint on all variables to keep residual variances above .001. Default is TRUE.
`bifactor`	Whether to specify a bifactor model in which a general factor predicts all included traits and the remaining factors are specifided to be orthogonal of one another.
`mustload`	Whether all variables should load on at least one factor, even if they dont meet the threshold specified using the cutoff argument.
`common`	Whether to specify a common factor model.

Value

The function outputs model syntax that can be used to run the model using the usermodel function in Genomic SEM.

Index