Package ‘BayClone2’

December 23, 2014

Version 1.0

Date 2014-17-12

Title Bayesian Feature Allocation Model for Tumor Heterogeneity

Author Juhee Lee, Peter Mueller, Subhajit Sengupta, Kamalakar Gulukota, Yuan Ji

Maintainer Subhajit Sengupta <subhajit06@gmail.com>

Depends R (>= 2.0.1), combinat

Description The package implements a Bayesian feature allocation model for inference on tumor heterogeneity using next-generation sequencing data. The model identifies the subclonal copy number and single nucleotide mutations at a selected set of loci and provides inference on genetic tumor variation.

License GPL-3 | file LICENSE

R topics documented:

BayClone2 *BayClone2 function*

Description

This function conducts posterior Markov chain Monte Carlo (MCMC) simulation for BayClone2, a BAYesian feature allocation model for tumor subCLONEs (Lee, et al (2014)) .

Usage

```
BayClone2(min_C, max_C, SS, TT, Burn.in, N.sam, NN, nn, hpara, ave.B)
```

Arguments

`min_C` the minimum value of C including the background subclone (should be >=2, that is, at least one subclone besides the background subclone)

`max_C` the maximum value of C including the background subclone

`SS` the number of loci

`TT` the number of tissue samples

`Burn.in` the number of burn-in iterations in MCMC

`N.sam` the number of MCMC samples for inference after burn-in

`NN` a (SxT) matrix where each element N_st denotes total number of reads at locus s in tissue sample t.

`nn` a (SxT) matrix where each element n_st denotes the number of mutated reads at locus s in tissue sample t.

`hpara` the set of hyper-parameters as a list; r, Q, alpha, beta, gam, a, b, a_z0, b_z0, d0, d. For details, see the below.

`ave.B` a real number between 0 and 1 which denotes the mean fraction of the data for training data (0.025 recommended). It will be used to split training and test dataset.

Details

1. max_C is for computational convenience (not from the modeling). You may set it at an arbitrarily large number but it may take more time for posterior computation.

2. The hyperparameters are passed as a list whose elements are:

r: C ~ GEOMETRIC(r) WHERE E(C)=1/r (r > 0)

Q: MAX NUMBER OF COPIES – q = 0, 1, 2, 3

alpha, beta, gam: PI_C | C ~ BETA-DIRICHLET (alpha/C, beta, gam, ..., gam) (alpha >0, beta >0, gam > 0)

a, b: PHI_T ~ GAMMA(a, b) (a > 0, b > 0)

a_z0, b_z0: P0 ~ BETA(a_z0, b_z0) (a_z0 >0, b_z0 > 0)

d0, d: W_T | C ~ DIRICHLET(d0, d, ..., d) WHERE W_T=(w_t0, w_t1, ..., w_tC) (d0 > 0, d > 0)

Value

The funtion returns a list of posterior samples of random parameters; C, L, Z, w, th, phi, pi, p0_z, M and p:

-C: the number of subclones as a vector

-L: the matrix of copy numbers (S*C) as a list

-Z: the matrix of the number of copies with variant sequence (S*C) as a list

-w: composition weights of samples over subclones (T*C) as a list

-th: unscaled composition weights (T*C) as a list

-phi: samples of average read counts with average sample copy equal to 2 as a matrix

-pi: propbility of being (l_sc=q) as a list

-p0_z: proprotion of SNV in background subclone as a vector

-M: average of subclonal copy number as a matrix

-p: probability of obseving a read with variant sequence as a mtrix.

Author(s)

J. Lee (juheelee@soe.ucsc.edu) and S. Sengupta (subhajit06@gmail.com)

References

J. Lee, P. Mueller, S. Sengupta, K. Gulukota, Y. Ji, Bayesian Inference for Tumor Subclones Accounting for Sequencing and Structural Variants (http://arxiv.org/abs/1409.7158)

Sengupta S, Gulukota K, Lee J, Mueller, P, Y. Ji, BayClone: Bayesian Nonparametric Inference of Tumor Subclones Using NGS Data. Conference paper accepted for PSB 2015 and oral presentation

See Also

export_N_n, fn_post_C, fn_posterior_point

Examples

```
##ILLUSTRATE BayClone2 WITH A SMALL SIMULATION.
###REPRODUCE SIMULATION 1 OF LEE ET AL.
library("BayClone2")

##READ IN DATA
data(BayClone2_Simulation1_mut)
data(BayClone2_Simulation1_tot)
##TOTAL NUMBER OF READS AT LOCUS s IN SAMPLE t
N <- as.matrix(BayClone2_Simulation1_tot)
##NUMBER OF READS WITH VARIANT SEQUENCE AT LOCUS s IN SAMPLE t
n <- as.matrix(BayClone2_Simulation1_mut)

S <- nrow(N)  # THE NUMBER OF LOCI (I.E. NUMBER OF ROWS OF N (AND n))
T <- ncol(N) #THE NUMBER OF TISSUE SAMPLES  (I.E. NUMBER OF COLUMNS OF N (AND n))

###################################
#HYPER-PARAMETER  ----SPECIFYING HYPERPARAMETER VALUES
######################################
#HYPER-PARAMETER
hyper <- NULL

#NUMBER OF SUBCLONES (GEOMETRIC DIST)
### C ~ GEOMETRIC(r) WHERE E(C)=1/r
hyper$r <- 0.2

#PRIOR FOR L
hyper$Q <- 3  #NUMBER OF COPIES -- q = 0, 1, 2, 3

##BETA-DIRICHLET
###PI_C | C ~ BETA-DIRICHLET (ALPHA/C, BETA, GAMMA)
hyper$alpha <- 2
hyper$beta <- 1
hyper$gam <- c(0.5, 0.5, 0.5)

#PRIOR FOR PHI--TOTAL NUMBER OF READS IN SAMPLE T
###PHI_T ~ GAMMA(A, B)
hyper$b <- 3
hyper$a <- median(N)*hyper$b

#PRIOR FOR P_O
```

```
###P0 ~ BETA(a, b)
hyper$a_z0 <- 0.3
hyper$b_z0 <- 5

#PRIOR FOR W
##W_T | L ~ DIRICHLET(D0, D, ..., D) WHERE W_T=(w_t0, w_t1, ..., w_tC)
hyper$d0 <- 0.5
hyper$d <- 1

#WE USE THE MCMC SIMULATION STRATEGY PROPOSED IN LEE AT EL (2014)
n.sam <- 10000;  ##NUMBER OF SAMPLES THAT WILL BE USED FOR INFERENCE
##NUMBER OF SAMPLES FOR BURN-IN
#(USE THIS FOR A TRAINING DATA---FOR DETIALS, SEE THE REFERENCE)
burn.in <- 6000

##############################################
###WE CONSIDER C BETWEEN 1 AND 15 IN ADDITION TO BACKGROUND SUBCLONE
####Max_C AND Min_C SPECIFIES VALUES OF C FOR POSTERIOR EXPLORATION
Min_C <- 2  ##INCLUDING THE BACKGROUND SUBCLONE
Max_C <- 16  ##INCLUDING THE BACKGROUND SUBCLONE


#################################################################
##DO MCMC SAMPLING FROM BAYCLONE2!
#################################################################
##THE LAST ARGUMENT (0.025) IS THE MEAN PROPORTION FOR THE TRAINING DATASET (SPECIFIED BY USERS)
##IT WILL BE USED TO SPLIT INTO TRAINING AND TEST DATASETS
##FOR DETAILS, SEE THE REFERENCE LEE AT EL (2014)
##TO RUN, COMMENT IN THE LINE BELOW (WARNING! THIS MAY TAKE APPROXIMATELY 30 MINUTES)
#set.seed(11615)
#MCMC.sam <- BayClone2(Min_C, Max_C, S, T, burn.in, n.sam, N, n, hyper, 0.025)


#################################################################
#COMPUTE THE POSTERIOR MARGINAL DIST OF C (THE NUMBER OF SUBCLONES)
#################################################################
##TO RUN, COMMENT IN THE LINE BELOW
#post_dist_C <- fn_post_C(MCMC.sam$C, Min_C, Max_C)

######################################################################################
####WE FIND POSTERIOR POINT ESTIMATES OF L, Z, W, PHI, PI, P0 FOR A CHOSEN VALUE OF C
######################################################################################
##THE FIRST ARGUMENT (3) IS A VALUE OF C CHOSEN BY USERS
#C IS THE NUMBER OF SUBCLONES INCLUDING THE BACKGROUND SUBCLONE
##THE CHOSE VALUE OF C SHOULD BE LESS THAN OR EQUAL TO 10 (INCLUDING THE BACKGROUND SUBCLONE)
#DUE TO THE PERMUTATION (FOR DETAILS, SEE SEE THE REFERENCE LEE AT EL (2014))
##TO RUN, COMMENT IN THE LINE BELOW (WARNING! THIS MAY TAKE ARPPOXIMATELY 15 MINUTES)
#point.est <- fn_posterior_point(3, S, T, MCMC.sam)
```

BayClone2_Simulation1_mut

Number of mutated reads (n) for simulation 1 in Lee et al (2014)

Description

This is a (S*T) data matrix for simulation 1 with S=100 and T=4 in Lee, et al (2014). Each element of this data matrix gives the number of reads having variant sequence among the reads mapped at locus s in sample t.

Usage

```
BayClone2_Simulation1_mut
```

Format

A integer matrix with 100 rows and 4 columns.

References

J. Lee, P. Mueller, S. Sengupta, K. Gulukota, Y. Ji, Bayesian Inference for Tumor Subclones Accounting for Sequencing and Structural Variants (http://arxiv.org/abs/1409.7158)

BayClone2_Simulation1_tot
Total number of reads (N) for simulation 1 in Lee et al (2014)

Description

This is a (S*T) data matrix for simulation 1 with S=100 and T=4 in Lee, et al. Each element of this data matrix gives the total number of reads mapped at locus s in sample t.

Usage

```
BayClone2_Simulation1_tot
```

Format

A integer matrix with 100 rows and 4 columns.

References

J. Lee, P. Mueller, S. Sengupta, K. Gulukota, Y. Ji, Bayesian Inference for Tumor Subclones Accounting for Sequencing and Structural Variants (http://arxiv.org/abs/1409.7158)

export_N_n *export_N_n function*

Description

This function takes a VCF file as input and generates two output files (names are specified by users) which contains the total number of reads and the number of reads that bear a mutated sequence, respectively, at a particular locus in a specific tissue sample.

Usage

```
export_N_n(vcf_in_file,N_tot_out_file,n_alt_out_file)
```

Arguments

`vcf_in_file`	the name of the input VCF file.
`N_tot_out_file`	the output file name for total reads count (character string)
`n_alt_out_file`	the output file name for mutant reads count (character string)

Details

Input VCF filename is passed as the first argument. Second and third arguments are the name of the output files.

Second file contains total number of reads at loci in all samples.

Third file contains total number of mutant reads at loci in all samples.

NOTE: Each row in both the output files represents one particular loci for all the samples. The output files may be used as data, N and n, for the function, BayClone2.

Value

The function generates two output files.

-N_tot_out_file: the output file that contains a (S x T) matrix, where S is the number of loci and T is the number of samples. Each element represents total number of reads at a locus in a specific tissue sample.

-n_alt_out_file: the output file that contains a (S x T) matrix, where S is the number of loci and T is the number of samples. Each element represents number of mutated reads at a locus in a tissue sample.

Author(s)

J. Lee (juheelee@soe.ucsc.edu) and S. Sengupta (subhajit06@gmail.com)

References

J. Lee, P. Mueller, S. Sengupta, K. Gulukota, Y. Ji, Bayesian Inference for Tumor Subclones Accounting for Sequencing and Structural Variants (http://arxiv.org/abs/1409.7158)

Sengupta S, Gulukota K, Lee J, Mueller, P, Y. Ji, BayClone: Bayesian Nonparametric Inference of Tumor Subclones Using NGS Data. Conference paper accepted for PSB 2015 and oral presentation

See Also

BayClone2, fn_post_C, fn_posterior_point

Examples

```
### Illustrate the functionality of the function export_N_n with an example
# please put the appropriate file names to run this function.
#library("BayClone2")
##INPUT FILE: test_Data.VCF
##OUTPUT FILES: N_tot.txt and n_alt.txt
#export_N_n("test_Data.VCF","N_tot.txt","n_alt.txt")

##LOAD THE OUTPUT FILES -- THE OUTPUT FILES CAN BE USED AS DATA FOR BAYCLONE2.
#N <- read.table("N_tot.txt")
#n <- read.table("n_alt.txt")
```

fn_posterior_point *fn_posterior_point function*

Description

This functions returns point estimates of the parameters such as a subclonal copy number matrix, a matrix of the number of copies with variant sequence in subclones, a matrix of composition weights of sampels in subclones and the expected read count with two copies (i.e posterior point estimate for L, Z, W, PHI, PI, P0 for a chosen value of C).

Usage

```
fn_posterior_point(CC, SS, TT, sam)
```

Arguments

CC	the number of subclones chosen by users. It should be less than or equal to 10(including background subclone). This limitation is due to the permutaion.
SS	the number of loci
TT	the number of tissue samples
sam	a list of MCMC samples returned from the function, BayClone2

Details

The argument passed to this function, sam is a list returned from BayClone2; sam should be a list of posterior samples of random parameters (returned from the funtion, BayClone2); C, L, Z, w, th, phi, pi, p0_z, M and p

Value

This function returns

C: the value of C passed to the function

L: a posteror point esitmate of L in a S*C matrix

Z: a posteror point esitmate of Z in a S*C matrix

w: a posteror point esitmate of w in a T*C matrix

p0: a posteror point esitmate of p0 as a scalor

phi: a posteror point esitmate of phi in a vector of T

M: a posterior point estimate of M in a S*T matrix

p: a posterior point estimate of p in a S*T matrix

Author(s)

J. Lee (juheelee@soe.ucsc.edu) and S. Sengupta (subhajit06@gmail.com)

References

J. Lee, P. Mueller, S. Sengupta, K. Gulukota, Y. Ji, Bayesian Inference for Tumor Subclones Accounting for Sequencing and Structural Variants (http://arxiv.org/abs/1409.7158)

See Also

export_N_n, BayClone2, fn_post_C

Examples

```
##ILLUSTRATE BayClone2 WITH A SMALL SIMULATION.
###REPRODUCE SIMULATION 1 OF LEE ET AL.
library("BayClone2")

##READ IN DATA
data(BayClone2_Simulation1_mut)
data(BayClone2_Simulation1_tot)
##TOTAL NUMBER OF READS AT LOCUS s IN SAMPLE t
N <- as.matrix(BayClone2_Simulation1_tot)
##NUMBER OF READS WITH VARIANT SEQUENCE AT LOCUS s IN SAMPLE t
n <- as.matrix(BayClone2_Simulation1_mut)

S <- nrow(N)  # THE NUMBER OF LOCI (I.E. NUMBER OF ROWS OF N (AND n))
T <- ncol(N) #THE NUMBER OF TISSUE SAMPLES  (I.E. NUMBER OF COLUMNS OF N (AND n))

###################################
#HYPER-PARAMETER  ----SPECIFYING HYPERPARAMETER VALUES
######################################
#HYPER-PARAMETER
hyper <- NULL

#NUMBER OF SUBCLONES (GEOMETRIC DIST)
### C ~ GEOMETRIC(r) WHERE E(C)=1/r
hyper$r <- 0.2

#PRIOR FOR L
```

```
hyper$Q <- 3  #NUMBER OF COPIES -- q = 0, 1, 2, 3

##BETA-DIRICHLET
###PI_C | C ~ BETA-DIRICHLET (ALPHA/C, BETA, GAMMA)
hyper$alpha <- 2
hyper$beta <- 1
hyper$gam <- c(0.5, 0.5, 0.5)

#PRIOR FOR PHI--TOTAL NUMBER OF READS IN SAMPLE T
###PHI_T ~ GAMMA(A, B)
hyper$b <- 3
hyper$a <- median(N)*hyper$b

#PRIOR FOR P_O
###P0 ~ BETA(a, b)
hyper$a_z0 <- 0.3
hyper$b_z0 <- 5

#PRIOR FOR W
##W_T | L ~ DIRICHLET(D0, D, ..., D) WHERE W_T=(w_t0, w_t1, ..., w_tC)
hyper$d0 <- 0.5
hyper$d <- 1

#WE USE THE MCMC SIMULATION STRATEGY PROPOSED IN LEE AT EL (2014)
n.sam <- 10000;  ##NUMBER OF SAMPLES THAT WILL BE USED FOR INFERENCE
##NUMBER OF SAMPLES FOR BURN-IN
#(USE THIS FOR A TRAINING DATA---FOR DETIALS, SEE THE REFERENCE)
burn.in <- 6000

##############################################
###WE CONSIDER C BETWEEN 1 AND 15 IN ADDITION TO BACKGROUND SUBCLONE
####Max_C AND Min_C SPECIFIES VALUES OF C FOR POSTERIOR EXPLORATION
Min_C <- 2  ##INCLUDING THE BACKGROUND SUBCLONE
Max_C <- 16  ##INCLUDING THE BACKGROUND SUBCLONE


#################################################################
##DO MCMC SAMPLING FROM BAYCLONE2!
#################################################################
##THE LAST ARGUMENT (0.025) IS THE MEAN PROPORTION FOR THE TRAINING DATASET (SPECIFIED BY USERS)
##IT WILL BE USED TO SPLIT INTO TRAINING AND TEST DATASETS
##FOR DETAILS, SEE THE REFERENCE LEE AT EL (2014)
##TO RUN, COMMENT IN THE LINE BELOW (WARNING! THIS MAY TAKE APPROXIMATELY 30 MINUTES)
#set.seed(11615)
#MCMC.sam <- BayClone2(Min_C, Max_C, S, T, burn.in, n.sam, N, n, hyper, 0.025)


#################################################################
#COMPUTE THE POSTERIOR MARGINAL DIST OF C (THE NUMBER OF SUBCLONES)
#################################################################
##TO RUN, COMMENT IN THE LINE BELOW
#post_dist_C <- fn_post_C(MCMC.sam$C, Min_C, Max_C)

######################################################################################
####WE FIND POSTERIOR POINT ESTIMATES OF L, Z, W, PHI, PI, P0 FOR A CHOSEN VALUE OF C
######################################################################################
##THE FIRST ARGUMENT (3) IS A VALUE OF C CHOSEN BY USERS
```

```
#C IS THE NUMBER OF SUBCLONES INCLUDING THE BACKGROUPD SUBCLONE
##THE CHOSE VALUE OF C SHOULD BE LESS THAN OR EQUAL TO 10 (INCLUDING THE BACKGROUND SUBCLONE)
#DUE TO THE PERMUTATION (FOR DETAILS, SEE SEE THE REFERENCE LEE AT EL (2014))
##TO RUN, COMMENT IN THE LINE BELOW (WARNING! THIS MAY TAKE ARPPOXIMATELY 15 MINUTES)
#point.est <- fn_posterior_point(3, S, T, MCMC.sam)
```

fn_post_C *fn_post_C function*

Description

This function computes the posterior marginal distribution of the number of subclones.

Usage

```
fn_post_C(C_sam, min_C, max_C)
```

Arguments

C_sam	the MCMC samples for C in a vecotr
min_C	the minimum value of C (should be >= 2)
max_C	the maximum value of C

Details

You may use the same min_C and max_C used for the function, BayClone2.

Value

This function returns a matrix having two columns. The first column has values of C and the second column has the corresponding posterior probabilities, p(C|data)

Author(s)

J. Lee (juheelee@soe.ucsc.edu) and S. Sengupta (subhajit06@gmail.com)

References

J. Lee, P. Mueller, S. Sengupta, K. Gulukota, Y. Ji, Bayesian Inference for Tumor Subclones Accounting for Sequencing and Structural Variants (http://arxiv.org/abs/1409.7158)

Sengupta S, Gulukota K, Lee J, Mueller, P, Y. Ji, BayClone: Bayesian Nonparametric Inference of Tumor Subclones Using NGS Data. Conference paper accepted for PSB 2015 and oral presentation

See Also

export_N_n, BayClone2, fn_posterior_point

Examples

```
##ILLUSTRATE BayClone2 WITH A SMALL SIMULATION.
###REPRODUCE SIMULATION 1 OF LEE ET AL.
library("BayClone2")

##READ IN DATA
data(BayClone2_Simulation1_mut)
data(BayClone2_Simulation1_tot)
##TOTAL NUMBER OF READS AT LOCUS s IN SAMPLE t
N <- as.matrix(BayClone2_Simulation1_tot)
##NUMBER OF READS WITH VARIANT SEQUENCE AT LOCUS s IN SAMPLE t
n <- as.matrix(BayClone2_Simulation1_mut)

S <- nrow(N)  # THE NUMBER OF LOCI (I.E. NUMBER OF ROWS OF N (AND n))
T <- ncol(N) #THE NUMBER OF TISSUE SAMPLES  (I.E. NUMBER OF COLUMNS OF N (AND n))

###################################
#HYPER-PARAMETER  ----SPECIFYING HYPERPARAMETER VALUES
######################################
#HYPER-PARAMETER
hyper <- NULL

#NUMBER OF SUBCLONES (GEOMETRIC DIST)
### C ~ GEOMETRIC(r) WHERE E(C)=1/r
hyper$r <- 0.2

#PRIOR FOR L
hyper$Q <- 3  #NUMBER OF COPIES -- q = 0, 1, 2, 3

##BETA-DIRICHLET
###PI_C | C ~ BETA-DIRICHLET (ALPHA/C, BETA, GAMMA)
hyper$alpha <- 2
hyper$beta <- 1
hyper$gam <- c(0.5, 0.5, 0.5)

#PRIOR FOR PHI--TOTAL NUMBER OF READS IN SAMPLE T
###PHI_T ~ GAMMA(A, B)
hyper$b <- 3
hyper$a <- median(N)*hyper$b

#PRIOR FOR P_O
###P0 ~ BETA(a, b)
hyper$a_z0 <- 0.3
hyper$b_z0 <- 5

#PRIOR FOR W
##W_T | L ~ DIRICHLET(D0, D, ..., D) WHERE W_T=(w_t0, w_t1, ..., w_tC)
hyper$d0 <- 0.5
hyper$d <- 1

#WE USE THE MCMC SIMULATION STRATEGY PROPOSED IN LEE AT EL (2014)
n.sam <- 10000;  ##NUMBER OF SAMPLES THAT WILL BE USED FOR INFERENCE
##NUMBER OF SAMPLES FOR BURN-IN
#(USE THIS FOR A TRAINING DATA---FOR DETIALS, SEE THE REFERENCE)
burn.in <- 6000
```

```
##############################################
###WE CONSIDER C BETWEEN 1 AND 15 IN ADDITION TO BACKGROUND SUBCLONE
####Max_C AND Min_C SPECIFIES VALUES OF C FOR POSTERIOR EXPLORATION
Min_C <- 2  ##INCLUDING THE BACKGROUND SUBCLONE
Max_C <- 16  ##INCLUDING THE BACKGROUND SUBCLONE


#################################################################
##DO MCMC SAMPLING FROM BAYCLONE2!
#################################################################
##THE LAST ARGUMENT (0.025) IS THE MEAN PROPORTION FOR THE TRAINING DATASET (SPECIFIED BY USERS)
##IT WILL BE USED TO SPLIT INTO TRAINING AND TEST DATASETS
##FOR DETAILS, SEE THE REFERENCE LEE AT EL (2014)
##TO RUN, COMMENT IN THE LINE BELOW (WARNING! THIS MAY TAKE APPROXIMATELY 30 MINUTES)
#set.seed(11615)
#MCMC.sam <- BayClone2(Min_C, Max_C, S, T, burn.in, n.sam, N, n, hyper, 0.025)


#################################################################
#COMPUTE THE POSTERIOR MARGINAL DIST OF C (THE NUMBER OF SUBCLONES)
#################################################################
##TO RUN, COMMENT IN THE LINE BELOW
#post_dist_C <- fn_post_C(MCMC.sam$C, Min_C, Max_C)

######################################################################################
####WE FIND POSTERIOR POINT ESTIMATES OF L, Z, W, PHI, PI, P0 FOR A CHOSEN VALUE OF C
######################################################################################
##THE FIRST ARGUMENT (3) IS A VALUE OF C CHOSEN BY USERS
#C IS THE NUMBER OF SUBCLONES INCLUDING THE BACKGROUPD SUBCLONE
##THE CHOSE VALUE OF C SHOULD BE LESS THAN OR EQUAL TO 10 (INCLUDING THE BACKGROUND SUBCLONE)
#DUE TO THE PERMUTATION (FOR DETAILS, SEE SEE THE REFERENCE LEE AT EL (2014))
##TO RUN, COMMENT IN THE LINE BELOW (WARNING! THIS MAY TAKE ARPPOXIMATELY 15 MINUTES)
#point.est <- fn_posterior_point(3, S, T, MCMC.sam)
```

Index