Package 'spruce'

February 21, 2022

Type Package

Title Spatial Random Effects Clustering of Single Cell Data

Version 0.99.1

Date 2022-02-17

Description Allows for identification of cell sub-populations within tissue samples using Bayesian multivariate mixture models with spatial random effects to account for a wide range of spatial gene expression patterns, as described in Allen et. al, 2021 <doi:10.1101/2021.06.23.449615>. Bayesian inference is conducted using efficient Gibbs sampling implemented using 'Rcpp'.

License GPL (>= 2)

Imports Rcpp, mvtnorm, BayesLogit, truncnorm, stats, igraph, MCMCpack, patchwork, tidyr, dplyr, ggplot2, tidyselect, Seurat, rlang

RoxygenNote 7.1.2

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

Depends R (>= 4.0)

NeedsCompilation yes

Author Carter Allen [aut, cre] (<https://orcid.org/0000-0001-6937-7234>), Dongjun Chung [aut]

Maintainer Carter Allen <carter.allen12@gmail.com>

Repository CRAN

Date/Publication 2022-02-21 08:50:02 UTC

R topics documented:

`build_knn_graph` *Make KNN network*

Description

Construct a binary adjacency matrix

Usage

```
build_knn_graph(coords, k)
```

Arguments

`coords`	An n x 2 data frame or matrix of 2d spot coordinates
`k`	The number of neighbors

Value

an adjacency matrix

Examples

```
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
A <- build_knn_graph(coords_df,k = 4)
```

coords_df_sim *stx Mouse brain coordinates*

Description

A data frame with 3 columns. Columns 1-2 give spot coordinates. Column 3 gives simulated ground truth labels.

Usage

```
coords_df_sim
```

Format

A 2696 x 3 data frame

fit_msn *Multivariate skew-normal mixture model clustering*

Description

Implement Gibbs sampling for MSN model with no spatial random effects

Usage

```
fit_msn(Y, K, nsim = 2000, burn = 1000, z_init = NULL)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
K	The number of mixture components to fit.
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Randomly initialized if NULL.

Value

a list of posterior samples

Examples

```
# parameters
n <- 100
g <- 3 # number of features
K <- 3 # number of clusters (mixture components)
pi <- rep(1/K,K) # cluster membership probability
z <- sample(1:K, size = n, replace = TRUE, prob = pi) # cluster indicators
z <- remap_canonical2(z)
t_true <- truncnorm::rtruncnorm(n,0,Inf,0,1)
t <- t_true

# Cluster Specific Parameters
# cluster specific means
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1)
)
# Cluster speficic skewness
Xi <- list(
  Xi1 = rep(2,g),
  Xi2 = rep(0,g),
  Xi3 = rep(-3,g)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]] + t[i]*Xi[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
fit1 <- fit_msn(Y,3,10,0)
```

`fit_msn_PG_smooth` *Multivariate skew normal mixture model clustering - PG multinom regression Spatial smoothing*

Description

Implement Gibbs sampling for MSN model with spatial smoothing prior. Includes fixed effects multinomial regression on cluster indicators using Polya-Gamma data augmentation.

Usage

```
fit_msn_PG_smooth(
  Y,
  W,
  coords_df,
  K,
  r = 3,
  nsim = 2000,
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
W	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)
coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
r	Empirical spatial smoothing
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Initialized with hierarchical clustering if NULL.
verbose	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
```

```
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_msn_PG_smooth(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_msn_smooth` *Spatial multivariate skew normal mixture model clustering*

Description

Implement Gibbs sampling for MSN model with spatial smoothing

Usage

```
fit_msn_smooth(
  Y,
  coords_df,
  K,
  r = 3,
  nsim = 2000,
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
r	Empirical spatial smoothing
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Randomly initialized if NULL.
verbose	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
```

```
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_msn_smooth(Y = Y, coords_df = coords_df, K = K, nsim = 10, burn = 0)
```

`fit_mvn` *Multivariate normal mixture model clustering*

Description

Implement Gibbs sampling for MVN model with no spatial random effects

Usage

```
fit_mvn(Y, K, nsim = 2000, burn = 1000, z_init = NULL)
```

Arguments

`Y`	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
`K`	The number of mixture components to fit.
`nsim`	Number of total MCMC iterations to run.
`burn`	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
`z_init`	Optional initialized allocation vector. Randomly initialized if NULL.

Value

a list of posterior samples

Examples

```
n <- 100 # number of observations
g <- 3 # number of features
K <- 3 # number of clusters (mixture components)
pi <- rep(1/K,K) # cluster membership probability
z <- sample(1:K, size = n, replace = TRUE, prob = pi) # cluster indicators
z <- remap_canonical2(z)

# Cluster Specific Parameters
# cluster specific means
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1)
```

```
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
fit1 <- fit_mvn(Y,3,10,0)
```

`fit_mvn_MCAR` *Multivariate normal spatial mixture model clustering*

Description

Implement Gibbs sampling for MVN model with MCAR spatial random effects

Usage

```
fit_mvn_MCAR(Y, coords_df, K, nsim = 2000, burn = 1000, z_init = NULL)
```

Arguments

`Y`	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
`coords_df`	An n x 2 data frame or matrix of 2d spot coordinates.
`K`	The number of mixture components to fit.
`nsim`	Number of total MCMC iterations to run.
`burn`	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
`z_init`	Optional initialized allocation vector. Randomly initialized if NULL.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)
A <- build_knn_graph(as.matrix(coords_df),k = 4)

n <- nrow(coords_df) # number of observations
g <- 2 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

# generate phi - not cluster specific
# conditional covariance of phi_i given phi_noti
m <- colSums(A)
M <- diag(m)
V <- matrix(0.4,nrow = g, ncol = g) # CAR covariance
diag(V) <- 0.6
V_true <- V
rho <- 0.999999 # Spatial dependence parameter ~ 1 for intrinsic CAR
Q <- diag(m) - rho*A # m is number of neighbors for each spot
covphi <- solve(Q) %x% V # gn x gn covariance of phis
phi <- mvtnorm::rmvnorm(1, sigma=covphi) # gn vector of spatial effects
PHI <- matrix(phi, ncol=g, byrow=TRUE)  # n x g matrix of spatial effects
PHI <- t(scale(t(PHI)))

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]] + PHI[i,],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
```

```
fit_MCAR <- fit_mvn_MCAR(Y = Y, coords_df = coords_df, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG` *Multivariate normal mixture model clustering - PG multinom regression*

Description

Implement Gibbs sampling for MVN model. Includes fixed effects multinomial regression on cluster indicators using Polya-Gamma data augmentation.

Usage

```
fit_mvn_PG(Y, W, K, nsim = 2000, burn = 1000, z_init = NULL, verbose = FALSE)
```

Arguments

`Y`	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
`W`	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)
`K`	The number of mixture components to fit.
`nsim`	Number of total MCMC iterations to run.
`burn`	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
`z_init`	Optional initialized allocation vector. Initialized with hierarchical clustering if NULL.
`verbose`	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
```

```
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG(Y = Y, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG_CAR` *Multivariate normal mixture model clustering - PG multinom regression w/ CAR random effect*

Description

Implement Gibbs sampling for MVN model. Includes fixed effects multinomial regression w/ CAR random intercepts on cluster indicators using Polya-Gamma data augmentation.

Usage

```
fit_mvn_PG_CAR(Y, W, coords_df, K, nsim = 2000, burn = 1000, z_init = NULL)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
W	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)

coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Randomly initialized if NULL.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}
```

```
# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_CAR(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG_CAR_MCAR` *Multivariate normal spatial mixture model clustering w/ PG multinomial regression on membership probabilities*

Description

Implement Gibbs sampling for MVN model with MCAR spatial random effects w/ PG multinomial regression on membership probabilities and CAR random ints in multinomial regression model.

Usage

```
fit_mvn_PG_CAR_MCAR(
  Y,
  W,
  coords_df,
  K,
  nsim = 2000,
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

`Y`	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
`W`	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)
`coords_df`	An n x 2 data frame or matrix of 2d spot coordinates.
`K`	The number of mixture components to fit.
`nsim`	Number of total MCMC iterations to run.
`burn`	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
`z_init`	Optional initialized allocation vector. Randomly initialized if NULL.
`verbose`	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_CAR_MCAR(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG_CAR_MCAR_smooth`

Multivariate normal spatial mixture model clustering w/ PG multinomial regression on membership probabilities with spatial smoothing

---

Description

Implement Gibbs sampling for MVN model with MCAR spatial random effects w/ PG multinomial regression on membership probabilities and CAR random ints in multinomial regression model with spatial smoothing.

Usage

```
fit_mvn_PG_CAR_MCAR_smooth(
  Y,
  W,
  coords_df,
  K,
  r = 3,
  nsim = 2000,
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
W	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)
coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
r	Empirical spatial smoothing
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Randomly initialized if NULL.
verbose	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
```

```
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_CAR_MCAR_smooth(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG_CAR_smooth` *Multivariate normal mixture model clustering - PG multinom regression w/ CAR random effect and spatial smoothing*

Description

Implement Gibbs sampling for MVN model. Includes fixed effects multinomial regression w/ CAR random intercepts on cluster indicators using Polya-Gamma data augmentation and spatial smoothing.

Usage

```
fit_mvn_PG_CAR_smooth(
  Y,
  W,
```

```
  coords_df,
  K,
  r = 3,
  nsim = 2000,
  burn = 1000,
  z_init = NULL
)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
W	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)
coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
r	Empirical spatial smoothing
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Randomly initialized if NULL.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
```

```
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_CAR_smooth(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG_MCAR` *Multivariate normal spatial mixture model clustering w/ PG multinomial regression on membership probabilities*

Description

Implement Gibbs sampling for MVN model with MCAR spatial random effects w/ PG multinomial regression on membership probabilities

Usage

```
fit_mvn_PG_MCAR(
  Y,
  W,
  coords_df,
  K,
  nsim = 2000,
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

`Y`	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
`W`	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)

coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Randomly initialized if NULL.
verbose	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}
```

```
# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_MCAR(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG_MCAR_smooth` *Multivariate normal spatial mixture model clustering w/ PG multinomial regression on membership probabilities and spatial smoothing*

Description

Implement Gibbs sampling for MVN model with MCAR spatial random effects w/ PG multinomial regression on membership probabilities and spatial smoothing

Usage

```
fit_mvn_PG_MCAR_smooth(
  Y,
  W,
  coords_df,
  K,
  r = 3,
  nsim = 2000,
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

`Y`	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
`W`	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)
`coords_df`	An n x 2 data frame or matrix of 2d spot coordinates.
`K`	The number of mixture components to fit.
`r`	Empirical spatial smoothing
`nsim`	Number of total MCMC iterations to run.
`burn`	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
`z_init`	Optional initialized allocation vector. Randomly initialized if NULL.
`verbose`	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_MCAR_smooth(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_PG_smooth` *Multivariate normal mixture model clustering - PG multinom regression Spatial smoothing*

Description

Implement Gibbs sampling for MVN model with spatial smoothing prior. Includes fixed effects multinomial regression on cluster indicators using Polya-Gamma data augmentation.

Usage

```
fit_mvn_PG_smooth(
  Y,
  W,
  coords_df,
  K,
  r = 3,
  nsim = 2000,
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
W	An n x v matrix of covariates to predict cluster membership. Should include an intercept (i.e., first column is 1)
coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
r	Empirical spatial smoothing
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Initialized with hierarchical clustering if NULL.
verbose	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
```

```
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_smooth(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
```

`fit_mvn_smooth` *Spatial multivariate normal mixture model clustering*

Description

Implement Gibbs sampling for MVN model with spatial smoothing

Usage

```
fit_mvn_smooth(
  Y,
  coords_df,
  K,
  r,
  nsim = 2000,
```

```
  burn = 1000,
  z_init = NULL,
  verbose = FALSE
)
```

Arguments

Y	An n x g matrix of gene expression values. n is the number of cell spots and g is the number of features.
coords_df	An n x 2 data frame or matrix of 2d spot coordinates.
K	The number of mixture components to fit.
r	Empirical spatial smoothing
nsim	Number of total MCMC iterations to run.
burn	Number of MCMC iterations to discard as burn in. The number of saved samples is nsim - burn.
z_init	Optional initialized allocation vector. Randomly initialized if NULL.
verbose	Logical for printing cluster allocations at each iteration.

Value

a list of posterior samples

Examples

```
## Not run:
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

# Cluster Specific Parameters
# cluster specific means
Mu <- list(
  Mu1 = rnorm(g,-2,1),
  Mu2 = rnorm(g,-1,1),
  Mu3 = rnorm(g,1,1),
  Mu4 = rnorm(g,2,1)
)
# cluster specific variance-covariance
S <- matrix(0.5,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1
Sig <- list(
  Sig1 = S,
  Sig2 = S,
```

```
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# sometimes helps to initialize using heuristic like kmeans
fitk <- stats::kmeans(Y,4)
z_km <- remap_canonical2(fitk$cluster)

# fit model
# use more iterations in practice
fit1 <- fit_mvn_smooth(Y,coords_df,4,2,10,0,z_km)
## End(Not run)
```

`fit_spruce` *Fit spruce Bayesian spatial mixture model*

Description

This function allows you to detect sub-populations single-sample spatial transcriptomics experiments.

Usage

```
fit_spruce(
  seurat_obj,
  K,
  emb = "PCs",
  n_dim = 8,
  r = 3,
  MCAR = TRUE,
  CAR = TRUE,
  smooth = TRUE,
  nsim = 2000,
  burn = 1000,
  z_init = NULL
)
```

Arguments

`seurat_obj`	An integrated Seurat object
`K`	The number of sub-populations to infer. Each should be present in each sample.

emb	Either one of "PCs", "HVGs", or "SVGs" OR a matrix with custom embeddings. If the latter, rows should be sorted as in meta data of Seurat object.
n_dim	The number of dimensions to use if emb is specified as one of "PCs", "HVGs", or "SVGs". Ignored if emb is a matrix of custom embeddings.
r	Spatial smoothing parameter. Should be greater than 0 with larger values enforcing stronger prior spatial association.
MCAR	Logical. Include multivariate CAR random intercepts in gene expression model?
CAR	Logical. Include univariate CAR random intercepts in multinomial gene expression model?
smooth	Logical. Use manual spatial smoothing controled by r parameter?
nsim	Number of total MCMC iterations to conduct.
burn	Number of initial MCMC iterations to discard as burn in. The number of saved iterations is nsim-burn
z_init	Initialized cluster allocation vector to aid in MCMC convergence. If NULL z_init will be set using hierarchical clustering.

Value

A list of MCMC samples, including the MAP estimate of cluster indicators (z)

---

get_map *Get MAP estimate of cluster indicators*

Description

Compute maximum aposteriori (MAP) estimate of cluster indicators

Usage

```
get_map(z)
```

Arguments

z	All cluster indicator posterior samples from a given cell spot

Value

MAP estimate of cluster labels. Useful applied over columns of posterior samples matrix (see example)

Examples

```
# parameters
n <- 100 # number of observations
g <- 3 # number of features
K <- 3 # number of clusters (mixture components)
pi <- rep(1/K,K) # cluster membership probability
z <- sample(1:K, size = n, replace = TRUE, prob = pi) # cluster indicators
z <- remap_canonical2(z)

# Cluster Specific Parameters
# cluster specific means
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
fit1 <- fit_mvn(Y,3,100,0)

# Apply get_map() to columns of Z (i.e., posterior samples from each cell spot)
z_map <- apply(fit1$Z, 2, get_map)
```

`get_psi_sums` *Sum all neighboring psis*

Description

Sum all neighboring psis

Usage

```
get_psi_sums(Psi, A)
```

Arguments

Psi	an n x 1 vector of component k psis
A	an n x n adjacency matrix

---

get_scores *Calculate cluster uncertainty*

Description

Use posterior estimates to calculate uncertainty scores

Usage

```
get_scores(fit)
```

Arguments

fit	A model fit returned by one of the fit_*_PG model functions

Value

An n x (K + 1) matrix. First K columns are continuous phenotypes, and last column is uncertainty scores

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
```

```
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_smooth(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
scores_df <- get_scores(fit)
```

`plot_deltas` *Plot delta parameters from multinomial regression model*

Description

Allows for visualization of multinomial regression models from spatial or non-spatial models

Usage

```
plot_deltas(fit)
```

Arguments

`fit`	A model fit returned by one of the fit_*_PG model functions

Value

a ggplot

Examples

```
# parameters
data(coords_df_sim)
coords_df <- coords_df_sim[,1:2]
z <- remap_canonical2(coords_df_sim$z)

n <- nrow(coords_df) # number of observations
```

```
g <- 3 # number of features
K <- length(unique(coords_df_sim$z)) # number of clusters (mixture components)
pi <- table(z)/length(z) # cluster membership probability

W <- matrix(0, nrow = n, ncol = 2)
W[,1] <- 1
W[,2] <- sample(c(0,1),size = n, replace = TRUE, prob = c(0.5,0.5))

# Cluster Specific Parameters
Mu <- list(
  Mu1 = rnorm(g,-5,1),
  Mu2 = rnorm(g,0,1),
  Mu3 = rnorm(g,5,1),
  Mu4 = rnorm(g,-2,3)
)
# cluster specific variance-covariance
S <- matrix(1,nrow = g,ncol = g) # y covariance matrix
diag(S) <- 1.5
Sig <- list(
  Sig1 = S,
  Sig2 = S,
  Sig3 = S,
  Sig4 = S
)

Y <- matrix(0, nrow = n, ncol = g)
for(i in 1:n)
{
  Y[i,] <- mvtnorm::rmvnorm(1,mean = Mu[[z[i]]],sigma = Sig[[z[i]]])
}

# fit model
# in practice use more mcmc iterations
fit <- fit_mvn_PG_smooth(Y = Y, coords_df = coords_df, W = W, K = K, nsim = 10, burn = 0)
plot_deltas(fit)
```

`psi_sums` *Sum neighboring psis in spot i*

Description

Sum neighboring psis in spot i

Usage

```
psi_sums(ai, Psi)
```

Arguments

`ai`	the ith row (or column) of the adjacency matrix
`Psi`	an n x 1 vector of component k psis

remap_canonical2 *Canonical re-mapping of mixture component labels*

Description

Avoid label switching by re-mapping sampled mixture component labels at each iteration (Peng and Carvhalo 2016).

Usage

```
remap_canonical2(z)
```

Arguments

z	A length-n vector of discrete mixture component labels

Value

A length-n vector of mixture component labels re-mapped to a canonical sub-space

Examples

```
# parameters
n <- 10 # number of observations
K <- 3 # number of clusters (mixture components)
pi <- rep(1/K,K) # cluster membership probability
z <- sample(1:K, size = n, replace = TRUE, prob = pi) # cluster indicators
z <- remap_canonical2(z)
```

spruce *SPRUCE*

Description

This package fits Bayesian spatial mixture models

spruce functions

The spruce functions ...

Index