Package: twl (via r-universe)

August 24, 2024

Type Package

Title Two-Way Latent Structure Clustering Model

Version 1.0

Date 2018-08-17

Author Michael Swanson

Maintainer Michael Swanson <dms866@mail.harvard.edu>

Description Implementation of a Bayesian two-way latent structure model for integrative genomic clustering. The model clusters samples in relation to distinct data sources, with each subject-dataset receiving a latent cluster label, though cluster labels have across-dataset meaning because of the model formulation. A common scaling across data sources is unneeded, and inference is obtained by a Gibbs Sampler. The model can fit multivariate Gaussian distributed clusters or a heavier-tailed modification of a Gaussian density. Uniquely among integrative clustering models, the formulation makes no nestedness assumptions of samples across data sources -- the user can still fit the model if a study subject only has information from one data source. The package provides a variety of post-processing functions for model examination including ones for quantifying observed alignment of clusterings across genomic data sources. Run time is optimized so that analyses of datasets on the order of thousands of features on fewer than 5 datasets and hundreds of subjects can converge in 1 or 2 days on a single CPU. See ``Swanson DM, Lien T, Bergholtz H, Sorlie T, Frigessi A, Investigating Coordinated Architectures Across Clusters in Integrative Studies: a Bayesian Two-Way Latent Structure Model, 2018, <doi:10.1101/387076>, Cold Spring Harbor Laboratory" at <https://www.biorxiv.org/content/early/2018/08/07/387076.full.pdf> for model details.

License GPL (>= 2)

Imports Rfast

Depends R (>= 2.10), data.table, MCMCpack, corrplot

RoxygenNote 6.0.1

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-24 11:00:03 UTC

Contents

`twl-package` *Two-Way Latent Structure Clustering Model*

Description

Implementation of a Bayesian two-way latent structure model for integrative genomic clustering. The model clusters samples in relation to distinct data sources, with each subject-dataset receiving a latent cluster label, though cluster labels have across-dataset meaning because of the model formulation. A common scaling across data sources is unneeded, and inference is obtained by a Gibbs Sampler. The model can fit multivariate Gaussian distributed clusters or a heavier-tailed modification of a Gaussian density. Uniquely among integrative clustering models, the formulation makes no nestedness assumptions of samples across data sources – the user can still fit the model if a study subject only has information from one data source. The package provides a variety of post-processing functions for model examination including ones for quantifying observed alignment of clusterings across genomic data sources. Run time is optimized so that analyses of datasets on the order of thousands of features on fewer than 5 datasets and hundreds of subjects can converge in 1 or 2 days on a single CPU. See "Swanson DM, Lien T, Bergholtz H, Sorlie T, Frigessi A, Investigating Coordinated Architectures Across Clusters in Integrative Studies: a Bayesian Two-Way Latent Structure Model, 2018, <doi:10.1101/387076>, Cold Spring Harbor Laboratory" at <https://www.biorxiv.org/content/early/2018/08/07/387076.full.pdf> for model details.

Details

The DESCRIPTION file:

Package:	twl
Type:	Package
Title:	Two-Way Latent Structure Clustering Model
Version:	1.0
Date:	2018-08-17
Author:	Michael Swanson
Maintainer:	Michael Swanson <dms866@mail.harvard.edu>
Description:	Implementation of a Bayesian two-way latent structure model for integrative genomic clustering. The model
License:	GPL (>= 2)
Imports:	Rfast
Depends:	data.table, MCMCpack, corrplot
RoxygenNote:	6.0.1
LazyData:	true

Index of help topics:

Author(s)

Michael Swanson

Maintainer: Michael Swanson <dms866@mail.harvard.edu>

References

Swanson DM, Lien T, Bergholtz H, Sorlie T, Frigessi A, Investigating Coordinated Architectures Across Clusters in Integrative Studies: a Bayesian Two-Way Latent Structure Model, 2018, doi: 10.1101/387076, Cold Spring Harbor Laboratory, https://www.biorxiv.org/content/early/2018/08/07/387076.full.pdf.

clus_save *Output samples*

Description

5000 iterations from output of TWLsample function

Usage

```
data(data_and_output)
```

Format

A list of data.tables

Source

output of TWLsample function

Examples

```
data(data_and_output)
ls()
```

cross_dat_analy *Compares clustering across datasets using metrics described in associated TWL manuscript*

Description

Compares clustering across datasets using metrics described in associated TWL manuscript

Usage

```
cross_dat_analy(clus_save, BURNIN)
```

Arguments

clus_save	list of samples outputted from TWLsample function.
BURNIN	number of samples devoted to burn-in. Defaults to 2000.

Value

outpu_lis a list of output metrics. The first element is a list of lists of sample-specific pairwise cluster overlap. The second element is an estimate of across all datasets cluster correspondence by averaging pairwise cluster overlap (the length is the vector therefore is the number of unique samples associated with at least 2 data sources.

Examples

```
data(data_and_output)
## Not run: clus_save <- TWLsample(misaligned_mat,misaligned,output_every=50,num_its=5000,manip=FALSE)
outpu_new <- pairwise_clus(clus_save,BURNIN=2000)
post_analy_cor(outpu_new,c("title1","title2","title3","title4","title5"),
tempfile(),ords='none')
clus_labs <- post_analy_clus(outpu_new,clus_save,c(2:6),rep(0.6,5),c("title1","title2",
"title3","title4","title5"),tempfile())
output_nest <- cross_dat_analy(clus_save,4750)

## End(Not run)
```

---

misaligned *Progressively misaligned cluster annotation*

Description

Example annotation information for simulated data of progressively misaligned clusters

Usage

```
data(data_and_output)
```

Format

A list of data.tables

Source

simulated

Examples

```
data(data_and_output)
ls()
```

---

misaligned_mat *Progressively misaligned cluster data matrices*

Description

Simulated data of progressively misaligned clusters on which to fit a TWL model.

Usage

```
data(data_and_output)
```

Format

A list of matrices

Source

simulated

Examples

```
data(data_and_output)
ls()
```

---

outpu_new *Output PSMs*

Description

Posterior similar matrices, output of pairwise_clus function

Usage

```
data(data_and_output)
```

Format

A list of matrices

Source

output of pairwise_clus function

Examples

```
data(data_and_output)
ls()
```

pairwise_clus *Create posterior similarity matrix from outputted list of clustering samples*

Description

Create posterior similarity matrix from outputted list of clustering samples

Usage

```
pairwise_clus(clus_save, BURNIN = 2000)
```

Arguments

clus_save	list of samples outputted from TWLsample function.
BURNIN	number of samples devoted to burn-in. Defaults to 2000.

Value

outpu a list whose length is the number of datasets being integrated, and each elemnt of which is a posterior similarity matrix. The dimension of each symmetric matrix is the number of samples in the respective dataset, and elements in the matrix are values between 0 and 1, and estimate of the probability 2 samples find themselves in the same clustering.

Examples

```
data(data_and_output)
## Not run: clus_save <- TWLsample(misaligned_mat,misaligned,output_every=50,num_its=5000,manip=FALSE)
outpu_new <- pairwise_clus(clus_save,BURNIN=2000)
post_analy_cor(outpu_new,c("title1","title2","title3","title4","title5"),
tempfile(),ords='none')
clus_labs <- post_analy_clus(outpu_new,clus_save,c(2:6),rep(0.6,5),c("title1","title2",
"title3","title4","title5"),tempfile())
output_nest <- cross_dat_analy(clus_save,4750)

## End(Not run)
```

post_analy_clus *Assigns cluster labels by building dendrogram and thresholding at specified height*

Description

Assigns cluster labels by building dendrogram and thresholding at specified height

Usage

```
post_analy_clus(outpu_new, clus_sav_new, num_clusts, height_clusts_vec = NULL,
  titles, pdf_path)
```

Arguments

outpu_new	the output of the pairwise_clus function, and a list whose length is the number of datasets being integrated, and each elemnt of which is a posterior similarity matrix. The dimension of each symmetric matrix is the number of samples in the respective dataset, and elements in the matrix are values between 0 and 1, and estimate of the probability 2 samples find themselves in the same clustering.
clus_sav_new	list of samples outputted from TWLsample function. See details for additional explanation of this parameter and height_clusts_vec.
num_clusts	a vector of length the number of integrated datasets, specifying the number of cluster labels to be identified from the generated dendrogram for each dataset
height_clusts_vec	vector of dendrogram heights of length the number of integrated datasets (if the analyst prefers manual inspection of outputted dendrograms and specification of the heights at which to threshold, thereby defining cluster membership). Defaults to NULL. See details for additional explanation of this parameter and num_clusts.
titles	Vector of strings of length the number of datasets, used as prefixes in column labels of the outputted list of data.tables.
pdf_path	file path where the dendrogram figures will be saved as a pdf.

Details

At least one of either num_clusts or height_clusts_vec, or both, can be specified. If both are specified, then heights is first used within the dendrogram for preliminary cluster assignment, then the X largest clusters of these receive final, outputted, assignment (the rest receiving a "clus_unknown" label), where X is the corresponding element in the num_clusts argument vector.

Value

post_lab a list of data.tables of 2 columns each with names 'nam' and '*_clus', the nam specifying sample name annotation, and *_clus with the assigned cluster, where * is the corresponding element in the title argument vector.

Examples

```
data(data_and_output)
## Not run: clus_save <- TWLsample(misaligned_mat,misaligned,output_every=50,num_its=5000,manip=FALSE)
outpu_new <- pairwise_clus(clus_save,BURNIN=2000)
post_analy_cor(outpu_new,c("title1","title2","title3","title4","title5"),
tempfile(),ords='none')
clus_labs <- post_analy_clus(outpu_new,clus_save,c(2:6),rep(0.6,5),c("title1","title2",
"title3","title4","title5"),tempfile())
output_nest <- cross_dat_analy(clus_save,4750)
```

```
## End(Not run)
```

---

post_analy_cor *Creates and saves correlation plots based on posterior similarity matrices*

Description

Creates and saves correlation plots based on posterior similarity matrices

Usage

```
post_analy_cor(outpu_new, titles, pdf_path, ords = "none")
```

Arguments

outpu_new	the output of the pairwise_clus function, and a list whose length is the number of datasets being integrated, and each elemnt of which is a posterior similarity matrix. The dimension of each symmetric matrix is the number of samples in the respective dataset, and elements in the matrix are values between 0 and 1, and estimate of the probability 2 samples find themselves in the same clustering.
titles	a vector of strings of length number of integrated datasets. Elements of the vector are titles in the respective correlation plots
pdf_path	file path where the plots will be saved as a pdf.
ords	whether the correlation plots should be reordered according to that of hierarchical clustering for a more comprehensible plot. Defaults to 'none'. Passing any string apart from 'none' (i.e., 'yes') will result in the re-ordering.

Value

dendro_ord regardless of whether correlation plots are reordered according to hierarchical clustering, a list of reorderings is returned of length the number of datasets on which analysis was performed.

Examples

```
data(data_and_output)
## Not run: clus_save <- TWLsample(misaligned_mat,misaligned,output_every=50,num_its=5000,manip=FALSE)
outpu_new <- pairwise_clus(clus_save,BURNIN=2000)
post_analy_cor(outpu_new,c("title1","title2","title3","title4","title5"),
tempfile(),ords='none')
clus_labs <- post_analy_clus(outpu_new,clus_save,c(2:6),rep(0.6,5),c("title1","title2",
"title3","title4","title5"),tempfile())
output_nest <- cross_dat_analy(clus_save,4750)

## End(Not run)
```

`TWLsample` *Main function to obtain posterior samples from a TWL model.*

Description

Main function to obtain posterior samples from a TWL model.

Usage

```
TWLsample(full_dat_mat, full_dat, alpha_re = 7, beta_re = 0.4,
  num_its = 5000, num_all_clus = 30, output_every = 20, manip = TRUE,
  sav_inter = FALSE)
```

Arguments

`full_dat_mat`	list of matrices of the different data types.
`full_dat`	list of data.tables with a single column labelled 'nam', denoting sample annotation. A consistent naming convention of samples must be used across data types.
`alpha_re`	Hyperparameter for the dirichlet prior model within each data type, influencing sparsity of clusterings. A smaller number encourages fewer clusters. Defaults to 7 and should be chosen as a function of sample size.
`beta_re`	Hyperparameter for the dirichlet prior model across datatypes within each sample, influencing the degree to which each data type's sample cluster labels affect those of the other data types. Defaults to 0.4 and should be chosen as a function of the total number of data types being integrated in the analysis.
`num_its`	Number of iterations. Defaults to 5000.
`num_all_clus`	Ceiling on the number of clusters. Defaults to 30. Should be chosen as some factor greater (for example, 5), than maximum number of hypothesized clusters in the data types.
`output_every`	Frequency of sampling log statistics, reporting mixing, cluster distribution, and proportion of cluster sharing across data types. Defaults to once every 20 iterations.
`manip`	TRUE/FALSE for whether likelihood manipulation should be used to increase mixing in situations where cluster means are far from one another in Euclidean distance. This should not influence identified clusters nor parameters associated with them. Defaults to TRUE.
`sav_inter`	A logical indicating whether a temporary file of the samples should be written out in the working directory every 50 iterations. Allows for restarts when sampling is interrupted, and defaults to FALSE.

Value

A list of lists of data.tables. The list length is the number of iterations. The length of each element is the number of data types. The data.tables have 2 columns, sample annotation called 'nam' and cluster assignment called 'clus'.

Examples

```
data(data_and_output)
## Not run: clus_save <- TWLsample(misaligned_mat,misaligned,output_every=50,num_its=5000,manip=FALSE)
outpu_new <- pairwise_clus(clus_save,BURNIN=2000)

## End(Not run)
post_analy_cor(outpu_new,c("title1","title2","title3","title4","title5"),
tempfile(),ords='none')
clus_labs <- post_analy_clus(outpu_new,clus_save,c(2:6),rep(0.6,5),c("title1","title2",
"title3","title4","title5"),tempfile())
output_nest <- cross_dat_analy(clus_save,4900)
```

Index