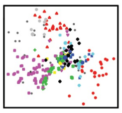
GaussClust  (http://imgur.com/KleBQPV)

codacy A (https://www.codacy.com/app/justincbagley/GaussClust?utm_source=github.com&amp;utm_medium=referral&amp;utm_content=justincbagley/GaussClust
license GPL (>= 2) (LICENSE)

Clustering using Gaussian mixture modeling for species delimitation and classification

LICENSE

All code within the GaussClust v0.1.0 repository is available "AS IS" under a generous GNU license. See the LICENSE (LICENSE) file for more information.

CITATION

If you use scripts from this repository as part of your published research, I require that you cite the repository as follows (also see DOI information below):

- Bagley, J.C. 2017. GaussClust v0.1.0. GitHub repository, Available at: http://github.com/justincbagley/GaussClust.

Alternatively, please provide the following link to this software repository in your manuscript:

- https://github.com/justincbagley/GaussClust

DOI

The DOI for GaussClust v0.1.0, via Zenodo (https://zenodo.org), is as follows: DOI 10.5281/zenodo.231221 (https://zenodo.org/badge/latestdoi/76403119). Here are some examples of citing GaussClust using the DOI:

Bagley, J.C. 2017. GaussClust v0.1.0. GitHub package, Available at: http://doi.org/10.5281/zenodo.231221.

Bagley, J.C. 2017. GaussClust v0.1.0. Zenodo, Available at: http://doi.org/10.5281/zenodo.231221.

CONTENTS

INTRODUCTION

> *"Clustering and discriminant analysis (or classification) methods are among the most important*
> *techniques in multivariate statistical learning." - Lebret et al. (2015)*

This repository focuses on ways to use Gaussian Mixture Models (GMMs) to conduct clustering analyses to address problems in systematics, particularly in the delimitation of species, and classification of individuals to species, using univariate or multivariate data. Recent papers discuss the promise of such methods for species delimitation, with or without multiple data-types (e.g. genetic, morphological, ecological data), and with or without accounting for noise during clustering (Hausdorf & Hennig 2010; Edwards & Knowles 2014). The current development version of the repository focuses on GaussClust.sh, a shell script that customizes and runs R scripts to conduct various GMM analyses, and bgmmSensTest.sh and related code for exploring the sensitivity of belief-based GMM analysis to varying prior probabilities on known observations. As in the case of the author's other software on GitHub (e.g. PIrANHA (https://github.com/justincbagley/PIrANHA)), GaussClust is fully command line-based and is available as open-source software according to the license.

As noted by Lebret et al. (2015), two foci of multivariate approaches related to clustering are **clustering** proper, which aims to group observations (e.g. individuals) into groups or 'clusters' that are more similar to one another than to other clusters, and **classification methods** where a discriminant function is used to assign new data to groups that are known a priori. GaussClust.sh primarily focuses on the former, but the latter also sneaks into clustering analyses and therefore is also included. The basic GaussClust workflow is most broadly divided into accomplishing three goals:

1. setting up and reading in the data,
2. creating an Rscript to conduct a set of user-directed GMM-based analyses,
3. running the Rscript and cleaning up the working directory.

The code run by **GaussClust.sh** in R can be set up to accomplish at least three basic steps:

1. scaling/standardization of the data by converting the raw data to Gower distances (Gower 1971) and conducting non-metric multidimensional scaling (NMDS) on the distances,
2. checking and preparing the data for GMM analyses, and
3. calling different GMM analyses.

There are several types of NMDS and GMM analyses available using GaussClust, including:

- **NMDS:**
    - regular NMDS on a single value specified for the number of clusters in the data
    - NMDS model selection using BIC
- **GMMs (all starting from NMDS results):**

- unsupervised GMM
- supervised or semisupervised discriminant analysis
- semisupervised belief-based GMMs

The bgmm (https://cran.r-project.org/web/packages/bgmm/index.html) R package (Szczurek et al. 2010; Biecek et al. 2012) that is available when using GaussClust implements several GMM variants, including unsupervised, partially supervised (soft- or belief-based GMM), and supervised models. However, the authors develop and implement belief-based GMM, an approach that I believe is likely to be of special interest during species delimitation and taxonomic classification studies. Thus, use of bgmm in GaussClust presently focuses on belief-based modeling. During belief-based GMM analysis, bgmm expects to be passed the number of clusters, an X matrix of unknown observations, a knowns matrix of labeled observations, and a P matrix containing the plausibilities of the known observations (beliefs matrix is set to P), which is taken as prior knowledge that the observations (e.g. individuals) belong to a certain cluster (Biecek et al. 2012). A problem arises that it may be difficult to assign priors for known observations. Moreover, since belief-based GMM uses the labeled and unlabeled data to estimate mixtures, labels could be predicted that are different from the original cluster labels, which should be the most plausible ones (Biecek et al. 2012). The **bgmmSensTest** portion of the repository uses a simple sensitivity analysis to probe the effect of varying the prior probabilities given to known observations (e.g. labeled individuals) during belief-based GMM in bgmm.

In **bgmmSensTest.sh**, bgmm is run while assuming that the diagonal values become prior probabilities for knowns, and other values in a row are zero (or some other minimum value) or are uniformly distributed across the remaining off-diagonal cells along each row. bgmmSensTest.sh goes through a simple series of steps to see how different the matrix of posterior probabilities of assignment to different clusters (named 'tij' in bgmm output) turns out to be when the (on-diagonal) prior probabilities for knowns in P are varied across some range of equally spaced float values, for example from 0.5 to 0.95. The basic steps in a bgmmSensTest run include: (1) set up work environment and read in sensitivity test conditions from 'bgmm_sens_test.cfg' configuration file in working dir, (2) make test input files with prior probs varying across user-specified range, (3) organize files and conduct test runs within separate sub-folders, (4) use results and text organized above to create an Rscript allowing visualization and comparison of results across test conditions, (5) run the Rscript, organize its output, and cleanup unnecessary files.

Hopefully, the results of bgmmSensTest will show that varying prior probabilities on knowns has very little effect on the posterior outcome of their assignment to clusters, and of of course this should also have very little effect on the unknowns! If, for example, results meet an arbitrary value of <<5% difference across all pairs of probability matrices (each derived from running GaussClust with a different 'test value' for the knowns' prior probs), then you can probably feel confident in fixing the prior probabilities of the knowns to a set value within the test range. Alternatively, results might be better presented from across the range of test values used in bgmmSensTest.

NOTE: One difference between bgmmSensTest code and regular GaussClust runs is that bgmmSensTest calls a 'lite' version of GaussClust with reduced to-screen output (printing comments to screen mostly turned off). *Users should only use this 'GaussClust_lite.sh' script with bgmmSensTest.sh, not by itself.*

Which GaussClust script or option is best for my kind of analysis?

Specific examples are given in different Usage sections below, but in general you'll be best off using GaussClust to **delimit species** with an unsupervised GMM or semisupervised belief-based GMM, and it will be best to use the discriminant analysis for **classification**. **GaussClust.sh** is used for straightforward GMM analysis using any method discussed above. **bgmmSensTest.sh** is used with GaussClust_lite.sh to conduct a sensitivity analysis described here and in the script itself. That's pretty much it, though more options are planned for development in the near future!

What's new in GaussClust?

While GaussClust is just passing its initial development stages, I have recently updated this software in several ways, for example to use the 'metaMDS' function available in the vegan (https://cran.r-project.org/web/packages/vegan/index.html) package to conduct NMDS, rather than using 'isoMDS' through labdsv (which turned out to be slightly problematic). **See the TODO list (https://github.com/justincbagley/GaussClust#todo-list) below for other ways I am seeking to improve the software.** ***If it interests you, feel free to jump in and help with development!!*** **:memo::+1:**

GETTING STARTED

Dependencies

Code in the GaussClust repository is largely dependent upon R and a series of R packages, as follows:

- The R (https://www.r-project.org) software environment
    - available from download mirrors from The Comprehensive R Archive Network (CRAN), such as https://cloud.r-project.org
- bgmm (https://cran.r-project.org/web/packages/bgmm/index.html)
- Rmixmod (https://cran.r-project.org/web/packages/Rmixmod/index.html)
- StatMatch (https://cran.r-project.org/web/packages/StatMatch/index.html)
- MASS (https://cran.r-project.org/web/packages/MASS/index.html)
- ggfortify (https://cran.r-project.org/web/packages/ggfortify/index.html)
- vegan (https://cran.r-project.org/web/packages/vegan/index.html)
- tools - a default R package with "base" priority
- grid - a default R package with "base" priority
- gplots (https://cran.r-project.org/web/packages/gplots/index.html)

Installation

:computer: GaussClust uses UNIX shell and R scripts and thus is well suited for running on a variety of operating systems, but especially UNIX/LINUX-like systems. Most of these systems (Mac OSX, Ubuntu, Linux, etc.) come with the shell preinstalled, so GaussClust code should run "out-of-the-box" from most any folder on your machine. To 'install' GaussClust, all you need to do is download the repository, move into the repository folder and enter `$ chmod u+x ./*.sh` into the command line interface (e.g. Mac Terminal). This changes file mode bits in the .sh files to allow the user permission to access and execute them, making the GaussClust code ready to run.

Best practices for input data and format

:warning: GaussClust calls R functions that are capable of robustly handling several weaknesses often present in real working datasets employed by researchers in systematics & biodiversity research, including (1) mixed data-types and (2) missing data. As one of its first steps, GaussClust also standardizes and reduces the dimensionality of the data using non-metric multidimensional scaling (NMDS), prior to running any Gaussian mixture models or discriminant analyses. This is ideal, because NMDS makes fewer assumptions about the data than other ordination methods such as PCA. However, it is critical that the data meet some standards for NMDS.

Perhaps the most important admonition I can make here is that users should be sure that the number of variables in their dataset is (much) greater than the potential/likely number of clusters present in the data, and greater than any range of clusters that they would like to model over. For example, under Another real-world GaussClust example (https://github.com/justincbagley/GaussClust#another-real-world-gaussclust-example-showing-variations) in the Usage section below, the analysis calls for modeling over 5 to 20 clusters, so I would expect that datasets with <20 variables might perform poorly in this case, and if a single value of nbCluster=x were used, you would need to use (many) more than x variables in your dataset. It can be difficult to ensure that such conditions are met or objectively determined; however, this issue highlights how critical it is that systematists have thoroughly explored their data, and are familiar with it, prior to conducting extensive analyses, including those involving NMDS or GMMs.

One suggestion for doing the above is for users to (1) explore basic patterns in the data, (2) conduct outlier analyses and remove any problematic individuals or variables, and (3) conduct pilot analyses with basic multivariate methods and nominal species labels prior to conducting any analyses using GaussClust. However, it would probably be a good idea to collect data for at least two to three times the number of nominal taxa in the dataset, in the case of species delimitation and classification analyses (this is my arbitrary guess; of course, four times that number might be even better!).

GaussClust Usage

> :warning: **WARNING!:** *As of current development build, GaussClust will only work properly if you specify to retain 4 NMDS dimensions (-k 4). Other values for -k will give weird results or cause run(s) to fail! Please excuse this limitation and note that fixing this issue is on the TODO list (https://github.com/justincbagley/GaussClust/blob/master/README.md#todo)!!; however, retaining 4 dimensions is common practice and may be adequate for your purposes (see k flag explanation in Usage below).*

Assuming GaussClust.sh is present in the current working directory, display usage for the script by simply entering its name at the command line and hitting return, as follows:

```
$ ./GaussClust.sh
```

```
##################################################################################
###########
#                              GaussClust v0.1.0, January 2017
#
##################################################################################
###########

INFO      | Wed Jan  4 14:50:29 CST 2017 | STEP #1: SETUP. SETTING OPTIONS AND
PATH VARIABLE...

Usage: ./GaussClust.sh [options] inputFile

Options: -k nmdsDimensions (specify number of dimensions, k, to retain during
NMDS on Gower distances) | -u unsuperGMM (0=no unsupervised GMM is carried out;
1=conduct unsupervised GMM using 'Rmixmod' R package, for comparative or
individual purposes) | -r rangeNumClusters (optional numeric listing of a
range, x:y, of the number of clusters to be modeled over during unsupervised
GMM in Rmixmod) | -d GMMDiscrim (0=(semi-)supervised, GMM-based discriminant
analysis is not carried out with 'mixmodelLearn' in Rmixmod; 1=conduct
discriminant analysis) | -b beliefBasedMM (0=no mixture modeling is carried out
using the 'bgmm' R package; 1=calls 'supervised' GMM analysis, 2=calls
'semisupervised' GMM analysis, and 3=calls both supervised and semisupervised
analyses in bgmm) | -p probsMatrix (specify matrix of plausibilities, or
weights of the prior probabilities for labeled observations, for bgmm; also
sets belief matrix) | -c numComponents (specify number of components (e.g.
Gaussian components) or 'clusters' to assign individuals to during regular GMM
(single value, rather than a range; see -r above) or bgmm modeling)

The -k flag sets the number of k dimensions to be retained during NMDS, which
affects both
regular Gaussian mixture modeling and also the different models that are
implemented in
the bgmm R package. Like file name, there is no default value; however, k=4 is
recommended
by the authors based on discussion in Edwards and Knowles (Proc. Roy. Soc. B.
2014) and
Hausdorf and Hennig (Syst. Biol. 2014). (By contrast, k=2 would be normal for
most other
ecological data, but may not contain sufficient information for interspecific
datasets.)

The -u flag calls the unsupervised Gaussian mixture modeling method implemented
in the
'mixmodCluster' function of the Rmixmod R package. See the Rmixmod R site and
documentation
for additional information on this package (available at:
https://cran.r-project.org/web/packages/Rmixmod/index.html). Set this flag to
'0' to
skip this analysis.

The -r flag gives the user the ability to conduct unsupervised modeling (called
using -u
above) over a range of nbCluster values. In the case that rangeNumClusters is
specified
(e.g. as '5:20'), Rmixmod will calculate unsupervised GMMs over this range and
select the
best model using the Bayesian information criterion (BIC). If a range of values
is not
specified for -r, then a GMM analysis in Rmixmod will use the number of
components/clusters
specified using the -c flag (see below).

The -d flag calls the supervised or semi-supervised discriminant analysis
method implemented
in the 'mixmodLearn' and 'mixmodPredict' functions of Rmixmod. The discriminant
analysis is
based on GMMs and is conducted in a two-step (A, Learning; B, Prediction)
procedure, which
estimates a discriminant function from known labeled data and uses it to
predict (classify)
unknown samples that correspondto the same knowns, i.e. species or clusters.
Set this flag
to '0' to skip this analysis.

The -b flag allows users to request two belief-based Gaussian mixture modeling
options
available in the 'bgmm' R package. The two currently supported models are
specified in
different functions by passing the script a value of '1', which calls the
'supervised'
function for supervised GMM analysis, or '2', which calls the 'semisupervised'
function
for semisupervised GMM analysis. You can also call both of these functions by
passing a
value of '3' to this option. See the bgmm R site and documentation for more
information
(available at: https://cran.r-project.org/web/packages/bgmm/index.html). Set
this flag
to '0' to skip this analysis.

The -p flag specifies the filename of the bgmm 'B' matrix file in the working
dir.

The -c flag specifies the number of components or 'clusters' that will be
modeled during
regular GMM or bgmm modeling (except see other option available using -r flag
above). This
corresponds to 'k' or the number of columns in 'B', based on definitions in the
bgmm
documentation.

Input file: Script expects as inputFile a single plain-text data table with a
header and
```

```
several columns of information followed by columns containing single-type or
mixed data
(e.g. categorical, discrete, or continuous data for different morphological
characters
measured) for the sample. The first column will be named 'samples' and
typically contain
sample IDs/codes for each individual (preferably with a species-specific
abbreviation
followed by a museum voucher number or individual code). The second column is
headed as
'type' and specifies whether each individual ID is 'known' or 'unknown'. The
third column
contains labels (e.g. four-letter codes) for each known individual (e.g. by
species), and
'NA' values for samples of unknown type, assigning individuals to species. The
example
input file contains a header with four-letter codes for each datacolumn, but
users can
make the names a little longer if needed.
```

Real-world GaussClust example #1:

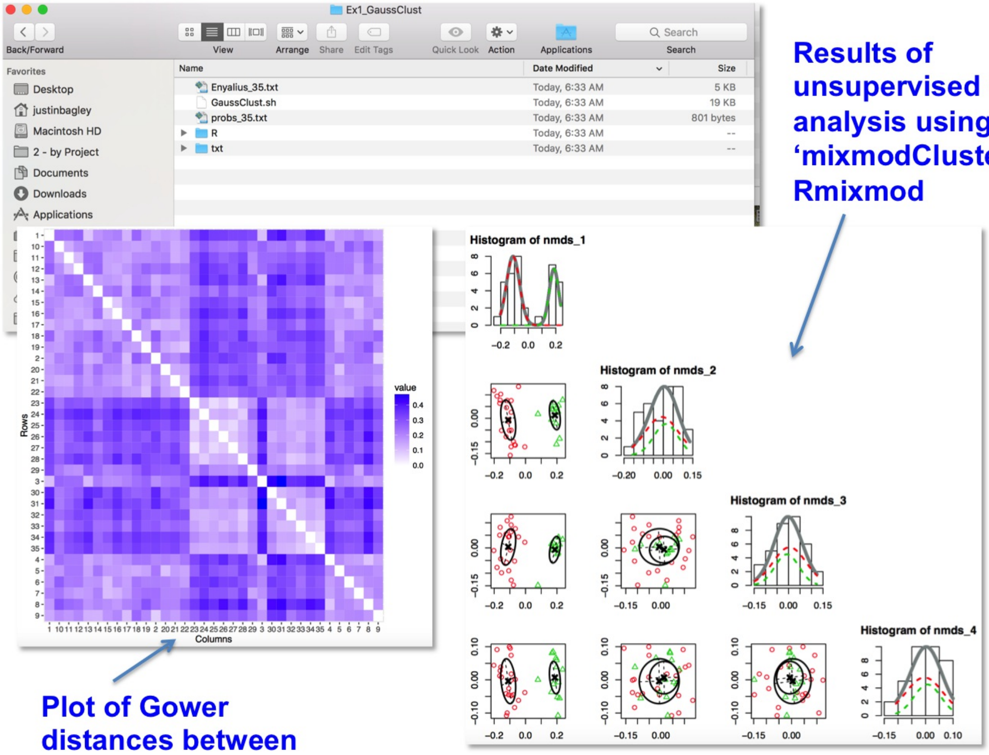
Below is an example of the usage for GaussClust.sh, in which the example data files ("Enyalius_35.txt" and "probs_35.txt"; see "example_data_files" folder of distro) are used to conduct *only* unsupervised Gaussian clustering (-u 1) on 2 clusters (-c 2), based on 4 dimensions of data retained from NMDS (-k 4). All screen output (e.g. to Terminal on mac) from the analysis is shown. When you run GaussClust, 'INFO' and date printed to screen along with information for each of the broader steps of the script (details from R go to an *.Rout file, organized at the end of each run). Question-response lines output to screen are marked 'FLOW' and (to date) requrie yes/no answers. As you can see, the code runs with no error messages.

***This run produced all directories and ouptut files present in the "Ex1_GaussClust" folder provided within the GaussClust distribution*. Some of the folder and graphical output are shown in Figures 1-3 below. Look over these Figures, and through the files in the example folder to learn more about the structure of GaussClust output, including graphical results saved as PDF files.**

```
$ ./GaussClust.sh -k 4 -u 1 -r 0 -d 0 -b 0 -p ./probs_35.txt -c 2
./Enyalius_35.txt

################################################################################
###########
#                             GaussClust v0.1.0, January 2017
#
################################################################################
###########

INFO      | Wed Jan  4 19:11:52 CST 2017 | STEP #1: SETUP. SETTING OPTIONS AND
PATH VARIABLE...
INFO      | Wed Jan  4 19:11:52 CST 2017 | STEP #2: MAKE GAUSSIAN CLUSTERING R
SCRIPT CONTAINING ENVIRONMENTAL VARIABLES AND ANALYSIS CODE...
INFO      | Wed Jan  4 19:11:52 CST 2017 | STEP #3: RUN THE R SCRIPT.
INFO      | Wed Jan  4 19:11:56 CST 2017 | STEP #4: CLEAN UP THE WORKSPACE.
INFO      | Wed Jan  4 19:11:56 CST 2017 |          Moving R output files to
new folder named 'R'...
FLOW      | Wed Jan  4 19:11:56 CST 2017 |          Would you like to keep the
Rscript output by GaussClust? (y/n) : y
FLOW      | Wed Jan  4 19:12:02 CST 2017 |          Would you like to keep text
files output by GaussClust? (y/n) : y
INFO      | Wed Jan  4 19:12:03 CST 2017 |          Moving text files to new
folder named 'txt'...
INFO      | Wed Jan  4 19:12:03 CST 2017 | Done conducting Gaussian clustering
and related analyses using GaussClust.
INFO      | Wed Jan  4 19:12:03 CST 2017 | Bye.
```

Read-world GaussClust example Results:

Figure 1. Structure and examples of GaussClust results, based on "Ex1_GaussClust" example run. Multiple PDF figures are output into the 'R' folder within the working directory, two of which are shown here. The first (*at left*), is a pairwise plot of Gower distances between observations, which shows a pronounced pattern of variation suggestive of multiple groups present in the dataset. The second figure (*at right*) shows the results of the unsupervised GMM analysis, comparing pairwise NMDS plots. Two clusters were strongly supported, and in a species delimitation context, this analysis would be interpreted as recovering two morphologically distinct populations or 'species'. ***Look in the "GaussClust.Rout" file in the R folder for more detailed results, including raw NMDS scores and details on clustering partitions etc.!***



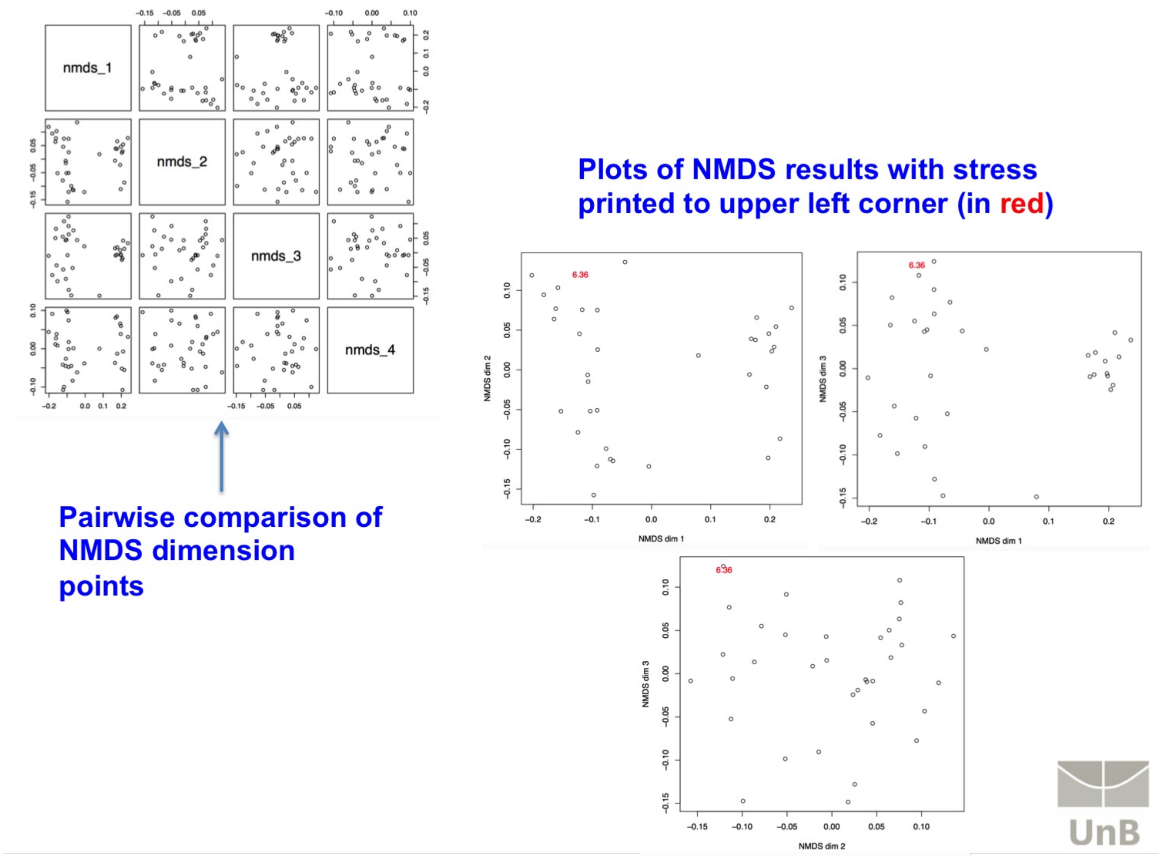
Name	Date Modified	Size
Enyalius_35.txt	Today, 6:33 AM	5 KB
GaussClust.sh	Today, 6:33 AM	19 KB
probs_35.txt	Today, 6:33 AM	801 bytes
R	Today, 6:33 AM	--
txt	Today, 6:33 AM	--

Results of unsupervised GMM analysis using ‘mixmodCluster’ in Rmixmod

Plot of Gower distances between observations (samples)

[(https://cloud.githubusercontent.com/assets/10584087/21687010/1f85da92-d32d-11e6-85b8-9e55e8ccfd56.png)](https://cloud.githubusercontent.com/assets/10584087/21687010/1f85da92-d32d-11e6-85b8-9e55e8ccfd56.png)

Figure 2. Additional pairwise plots of NMDS results from GaussClust, including stress and patterns of variation in the dims with no clusters imposed.

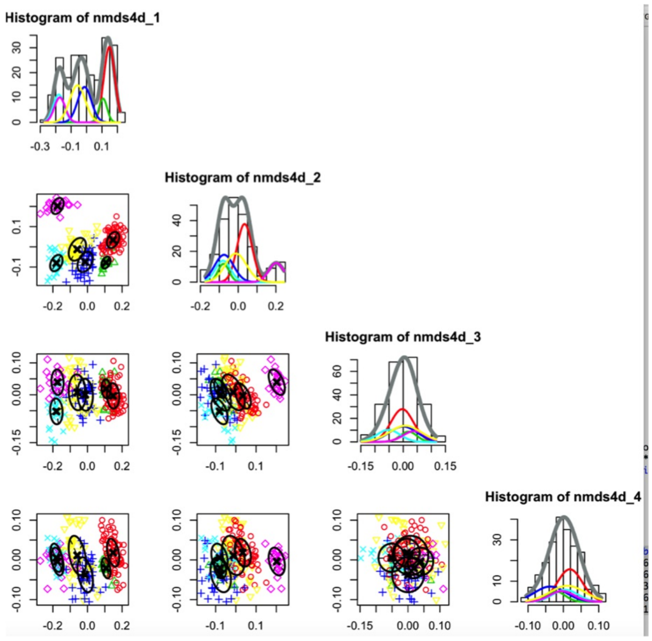


Plots of NMDS results with stress printed to upper left corner (in red)

Pairwise comparison of NMDS dimension points

[(https://cloud.githubusercontent.com/assets/10584087/21687009/1f7d79ce-d32d-11e6-800e-b2e7a0247205.png)](https://cloud.githubusercontent.com/assets/10584087/21687009/1f7d79ce-d32d-11e6-800e-b2e7a0247205.png)

Figure 3. A more complex unsupervised GMM result from GaussClust, produced during an analysis that supported 6 clusters in the data.



[(https://cloud.githubusercontent.com/assets/10584087/21687007/1f7b38b2-d32d-11e6-806b-353471e39ffc.png)](https://cloud.githubusercontent.com/assets/10584087/21687007/1f7b38b2-d32d-11e6-806b-353471e39ffc.png)

Another real-world GaussClust example, showing some variations:

Here, (1) we specify to keep 4 NMDS dimensions (-k 4); (2) conduct a regular unsupervised GMM analysis in the Rmixmod R package (Lebret et al. 2015), using multiple models across 5-20 clusters (-u 1 -r 5:20), which are compared to identify the best model using BIC; (3) call supervised discriminant analysis in Rmixmod (-d 1); (4) attempt supervised and semisupervised analysis in bgmm (-b 3 for 'both'); and (4) specify that analyses use or require 13 clusters (as needed; -c 13). Output is not redirected (e.g. by placing something like '> output.txt' at the end, so all output from the script is printed to screen (except for steps conducted in R). Example files are not provided for this example, because this analysis is part of an on-going project in our lab.

```
$ ./GaussClust.sh -k 4 -u 1 -r 5:20 -d 1 -b 3 -p B_206.txt -c 13 ./mydata.txt
```

bgmmSensTest Usage

> :warning: **WARNING!:** *As of current development build, you should run bgmmSensTest.sh in a new sub-folder created within the GaussClust-master distro folder, so that GaussClust_lite.sh is present one enclosing folder up from the working directory for the analysis.*

The bgmmSensTest script doesn't print out usage options, as GaussClust does; however, assuming you heed the warning above, the bgmm sensitivity test can then be run by adding appropriate values for test conditions following each equals sign in the bgmm_sens_test.cfg configuration file and then entering `$ ./bgmmSensTest.sh` at the command line while the data file (e.g. 'data.txt'), P matrix file (e.g. 'P.txt'), and configuration file are all present in the same working dir folder.

Real-world bgmmSensTest example #1:

I have produced a bgmmSensTest example using the same example data files as those used to test GaussClust above. Below, I provide the code for producing the example configuration file and screen output from a run on these files.

***This run produced all directories and ouptut files present in the "Ex2_bgmmSensTest" folder provided within the GaussClust distribution*. Some of the folder and graphical output are shown in Figures 4-6 below. Look over these Figures, and through the files in the example folder to learn more about the structure of GaussClust output, including graphical results saved as PDF files.**

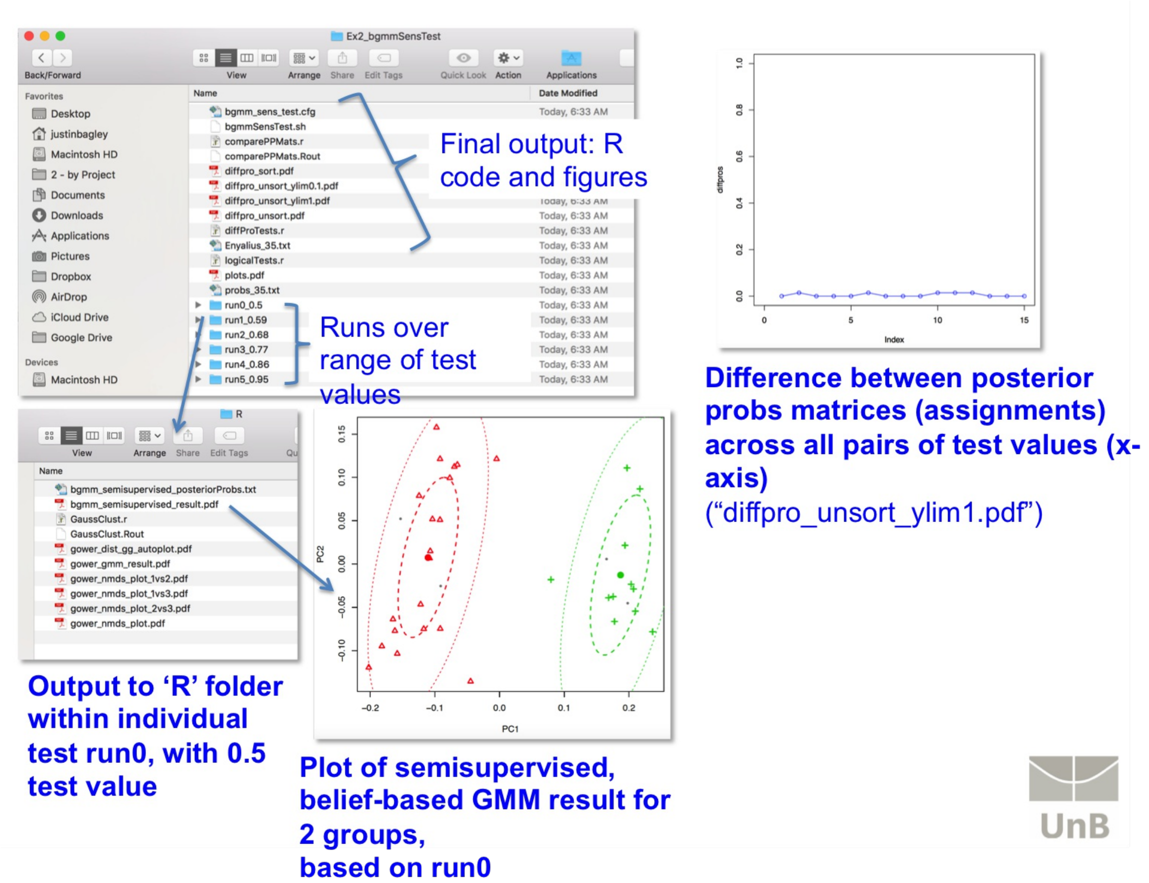
```
$ cd GaussClust-master
$ nano ./bgmm_sens_test.cfg  ## Use nano to edit the configuration file and
save changes.
$ cat ./bgmm_sens_test.cfg  ## Use cat to view the contents of the
configuration file:
######################  bgmmSensTest v0.1.0 Configuration File
##########################

## PARTIAL STRING FOR GaussClust OPTIONS k, u, r, d, and b  ##
##--Must be a string listing each option flag in usual way (hyphen followed by
lowercase
##--letter), followed by a space, and then an integer value. This is a
'partial' string
##--because the flag (-p) and filename info for the matrix of plausibilities /
prior
```

```
##--probabilities on knowns are not included, nor is the -c flag with the
number of
##--clusters (this is handled below) or the original data file name (handled
below as well).
##--Also RECALL: To get Usage for ./GaussClust.sh, simply type the script name
at the CLI
##--and press enter. See the GaussClust script or its README for more
information; however,
##--here are two examples of syntax that could be used here:
##--    -k 4 -u 1 -r 0 -d 0 -b 3
##--    -k 4 -u 1 -r 5:20 -d 1 -b 3

    opt_string_part=-k 4 -u 1 -r 0 -d 0 -b 3

## NUMBER OF COMPONENTS (e.g. GAUSSIAN COMPONENTS), OR CLUSTERS (-c) ##

    num_clusters=2

## FILE NAMES  ##
    data_file=Enyalius_35.txt
    probs_file=probs_35.txt

## SENSITIVITY TEST SETTINGS  ##
    target_diag_value=0.95
    num_test_vals=6
    test_val_min=0.5
    test_val_max=0.95
    test_val_inc=0.09

#
#
#
########################################## END
#############################################
$
$ ## Make a new directory for the example run and move the files into it:
$ mkdir Ex2_bgmmSensTest; cp ./example_data_files/Enyalius_35.txt
./example_data_files/probs_35.txt ./bgmm_sens_test.cfg ./Ex2_bgmmSensTest/;
$ cd ./Ex2_bgmmSensTest/
$ ./bgmmSensTest.sh  ## Run the sensitivity analysis!

################################################################################
###########
#                           bgmmSensTest v0.1.0, December 2016
#
################################################################################
###########

INFO      | Wed Jan  4 19:13:35 CST 2017 | Starting bgmmSensTest analysis...
INFO      | Wed Jan  4 19:13:35 CST 2017 | STEP #1: SETUP AND USER INPUT.
INFO      | Wed Jan  4 19:13:35 CST 2017 |          Setting working directory
to: /Users/justinbagley/Documents/GaussClust/Ex2_bgmmSensTest
```

```
INFO      | Wed Jan  4 19:13:35 CST 2017 |          Reading in sensitivity test
conditions and file names from configuration file.
INFO      | Wed Jan  4 19:13:35 CST 2017 | STEP #2: MAKE DIFFERENT INPUT FILES
FOR THE TEST, VARYING TEST STAT ACROSS A RANGE OF VALUES.
INFO      | Wed Jan  4 19:13:35 CST 2017 | STEP #3: RUN ALGORITHM ACROSS TEST
INPUT FILES, SAVE & COLLATE RESULTS.
INFO      | Wed Jan  4 19:13:35 CST 2017 | Running GaussClust_lite on test
input files generated by 'bgmmSensTest.sh' ...
INFO      | Wed Jan  4 19:13:38 CST 2017 | Running GaussClust_lite on test
input files generated by 'bgmmSensTest.sh' ...
INFO      | Wed Jan  4 19:13:41 CST 2017 | Running GaussClust_lite on test
input files generated by 'bgmmSensTest.sh' ...
INFO      | Wed Jan  4 19:13:44 CST 2017 | Running GaussClust_lite on test
input files generated by 'bgmmSensTest.sh' ...
INFO      | Wed Jan  4 19:13:47 CST 2017 | Running GaussClust_lite on test
input files generated by 'bgmmSensTest.sh' ...
INFO      | Wed Jan  4 19:13:50 CST 2017 | Running GaussClust_lite on test
input files generated by 'bgmmSensTest.sh' ...
INFO      | Wed Jan  4 19:13:53 CST 2017 | STEP #4: MAKE R SCRIPT CONTAINING
ENVIRONMENTAL VARIABLES AND ANALYSIS CODE FOR VISUALIZING AND COMPARING
RESULTS...
INFO      | Wed Jan  4 19:13:53 CST 2017 | STEP #5: RUN THE R SCRIPT AND
CONDUCT CLEANUP.
INFO      | Wed Jan  4 19:13:54 CST 2017 | Done conducting sensitivity test(s)
examining the effect of varying the 'prior' probabilities of known observations
in the beliefs matrix supplied to bgmm, using bgmmSensTest.
INFO      | Wed Jan  4 19:13:54 CST 2017 | Bye.
```

Read-world bgmmSensTest example Results:

Figure 4. Structure and examples of bgmmSensTest results, based on "Ex2_bgmmSensTest" example run. This figures shows the file structure resulting from a sensitivity anlaysis using bgmmSensTest. Runs are conducted over a range of test values, each within a separately labeled folder (here runs 0-5, over 6 test values). The first (*below left*) inset shows output to the R folder within one individual test run folder, that for run0. The "bgmm_semisupervised_result.pdf" figure clearly supports two groups in the data. ***Look in the "comparePPMats.Rout" file for more details about this result (output to R console and saved to file) and comparisons among test runs!*** All the resulting posterior probabilities of assignment are compared by the Rscript generated during the analysis, and output files are placed in the working directory. One of these ("diffpro_unsort_ylim1.pdf") clearly shows that varying the prior probabilities of knowns results in <5% difference across different pairs of result (posterior) matrices, on an appropriate 0-1 scale, suggesting that varying the on-diagonal probs of knowns has little effect on the results.



```
Ex2_bgmmSensTest
Name                            Date Modified
bgmm_sens_test.cfg              Today, 6:33 AM
bgmmSensTest.sh
comparePPMats.r
comparePPMats.Rout
diffpro_sort.pdf
diffpro_unsort_ylim0.1.pdf
diffpro_unsort_ylim1.pdf        Today, 6:33 AM
diffpro_unsort.pdf              Today, 6:33 AM
diffProTests.r                  Today, 6:33 AM
Enyalius_35.txt                 Today, 6:33 AM
logicalTests.r                  Today, 6:33 AM
plots.pdf                       Today, 6:33 AM
probs_35.txt                    Today, 6:33 AM
run0_0.5                        Today, 6:33 AM
run1_0.59                       Today, 6:33 AM
run2_0.68                       Today, 6:33 AM
run3_0.77                       Today, 6:33 AM
run4_0.86                       Today, 6:33 AM
run5_0.95                       Today, 6:33 AM
```

```
R
Name
bgmm_semisupervised_posteriorProbs.txt
bgmm_semisupervised_result.pdf
GaussClust.r
GaussClust.Rout
gower_dist_gg_autoplot.pdf
gower_gmm_result.pdf
gower_nmds_plot_1vs2.pdf
gower_nmds_plot_1vs3.pdf
gower_nmds_plot_2vs3.pdf
gower_nmds_plot.pdf
```

Final output: R code and figures

Runs over range of test values

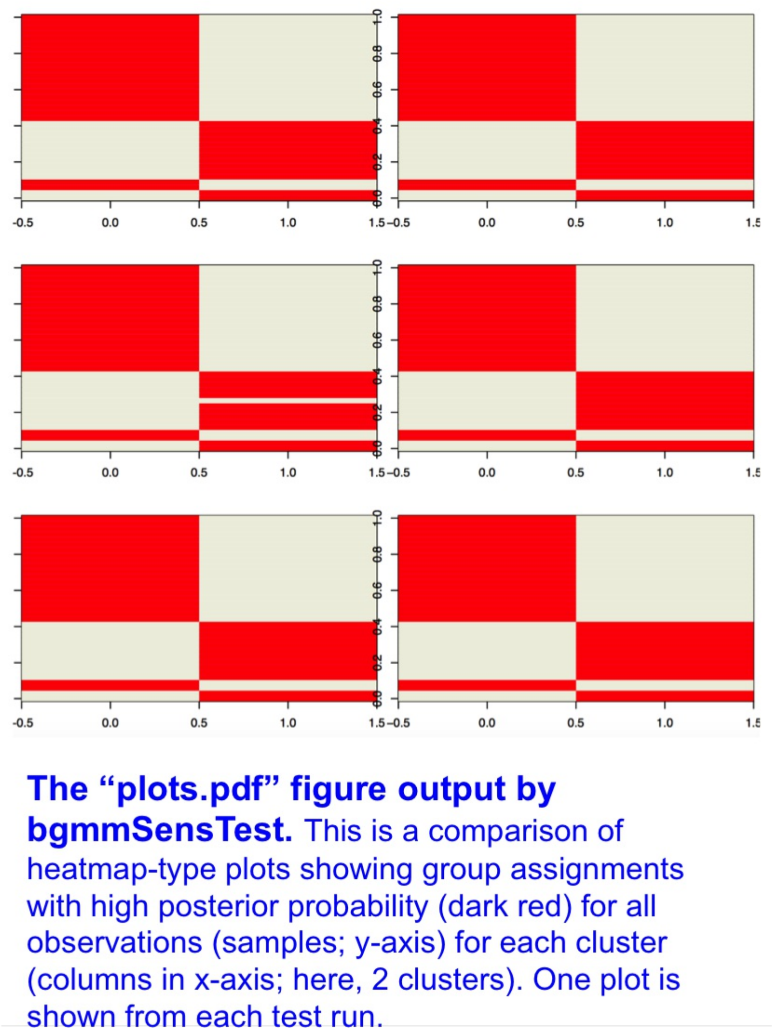
Output to 'R' folder within individual test run0, with 0.5 test value

Plot of semisupervised, belief-based GMM result for 2 groups, based on run0

Difference between posterior probs matrices (assignments) across all pairs of test values (x-axis) ("diffpro_unsort_ylim1.pdf")

(https://cloud.githubusercontent.com/assets/10584087/21687008/1f7b6698-d32d-11e6-8940-4034bdf87657.png)

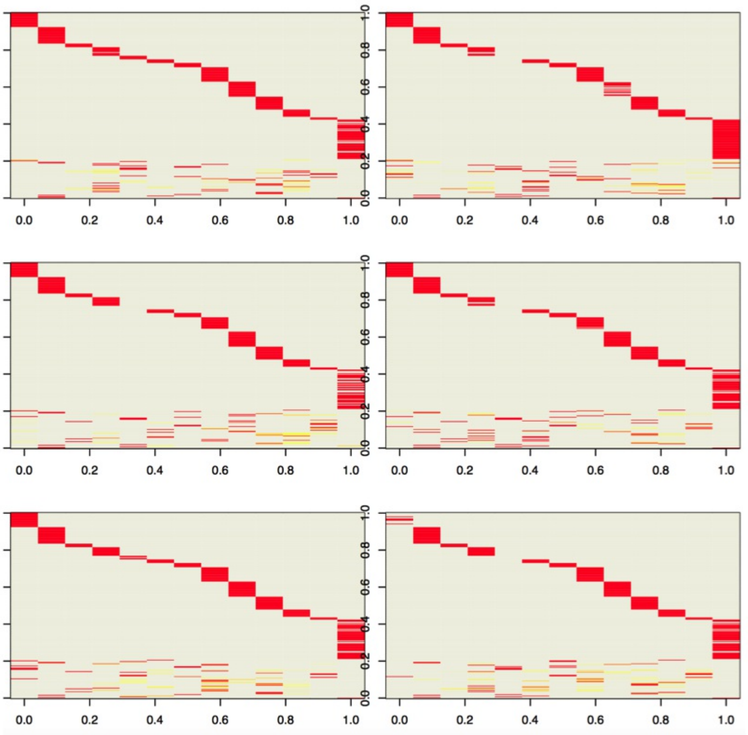
Figure 5. (*see caption below*)



The “plots.pdf” figure output by bgmmSensTest. This is a comparison of heatmap-type plots showing group assignments with high posterior probability (dark red) for all observations (samples; y-axis) for each cluster (columns in x-axis; here, 2 clusters). One plot is shown from each test run.

[(https://cloud.githubusercontent.com/assets/10584087/21687006/1f7993ae-d32d-11e6-9f08-fc426f43bf99.png)](https://cloud.githubusercontent.com/assets/10584087/21687006/1f7993ae-d32d-11e6-9f08-fc426f43bf99.png)

Figure 6. (*see caption below*)



A more complex set of heatmaps, resulting from a sensitivity test over 6 test values, for 13 clusters. This plot shows greater variation in posterior assignments, including lower values shown in yellow or orange.

[(https://cloud.githubusercontent.com/assets/10584087/21687005/1f792a36-d32d-11e6-856c-c534c3f0b1a3.png)](https://cloud.githubusercontent.com/assets/10584087/21687005/1f792a36-d32d-11e6-856c-c534c3f0b1a3.png)

TROUBLESHOOTING

How to troubleshoot some potentially common problems encountered by users:

(**1**) One problem you may run into is the 'permission denied' error returned by the shell when attempting to execute the shell scripts; for example, bash might return "-bash: ./GaussClust.sh: Permission denied". This indicates that permissions for .sh files were not correctly set during [Installation (https://github.com/justincbagley/GaussClust#installation)](https://github.com/justincbagley/GaussClust#installation). Fix this by moving into the repository master folder and entering `$ chmod u+x ./*.sh` at the command line, then using the shell script(s) in the repo folder from that point forward.

(**2**) Sometimes, likelihood or Bayesian calculations in Rmixmod or bgmm fail because they, by chance, use a 'bad'/suboptimal random number seed. This is a rather trivial issue, but it can cause clustering analysis runs to fail. Fixing this requires simply re-running the analysis with a different seed (selected for you each time you run). For example, one common minor issue with bgmmSensTest analyses is that you are attempting to run x number of runs over x different test values, and the code creates separate daughter folders in which each run is conducted, but one run fails because one or all model likelihoods goes to infinity! You will notice this because 1) an error will be output to screen showing that the shell 'mv', or move, command failed to execute properly since it couldn't find the posterior probability txt file resulting from one of the runs (which should be present in a 'txt' folder in the run directory). This causes the bgmmSensTest analyses and figures to be incomplete. For example, if you don't rerun the analysis, then you will have one less heatmap-type plot than expected in the resulting "plots.pdf" output file.

(**3**) If you inadvertently attempted to run bgmmSensTest without passing a value other than "0" or "1" for the -b flag in the configuration file (as in the "bgmm_sens_test-bozo.cfg" file in the distro), you will find a number of errors and problems with the results, because belief-based GMM is not conducted. For example, after running bgmmSensTest with "-b 0" in the first cfg file option, you would find that separate runs were conducted across a range of test values, but that a series of errors were printed to screen starting at the end of STEP #3, in which cp, sed, cat, and rm failed. The first cp failure message will say that the *_posteriorProbs.txt results files could not be found, and later error messages will also state that no ./tij*.txt file could be removed, etc. From the partial screen output below, you can see that the reporting gets a bit messy, indicating several things went wrong:

```
...
...
INFO      | Thu Jan  5 00:50:29 CST 2017 | Running GaussClust_lite on test
input files generated by 'bgmmSensTest.sh' ...
cp: ./run0_0.5/R/*_posteriorProbs.txt: No such file or directory
cp: ./run1_0.59/R/*_posteriorProbs.txt: No such file or directory
cp: ./run2_0.68/R/*_posteriorProbs.txt: No such file or directory
cp: ./run3_0.77/R/*_posteriorProbs.txt: No such file or directory
cp: ./run4_0.86/R/*_posteriorProbs.txt: No such file or directory
cp: ./run5_0.95/R/*_posteriorProbs.txt: No such file or directory
sed: ./pairs.txt: No such file or directory
cat: ./Rtop.tmp: No such file or directory
rm: ./Rtop.tmp: No such file or directory
INFO      | Thu Jan  5 00:50:32 CST 2017 | STEP #4: MAKE R SCRIPT CONTAINING
ENVIRONMENTAL VARIABLES AND ANALYSIS CODE FOR VISUALIZING AND COMPARING
RESULTS...
INFO      | Thu Jan  5 00:50:32 CST 2017 | STEP #5: RUN THE R SCRIPT AND
CONDUCT CLEANUP.
rm: ./tij*.txt: No such file or directory
rm: ./*.tmp: No such file or directory
rm: ./pairs*.txt: No such file or directory
rm: ./diffProTests.txt: No such file or directory
INFO      | Thu Jan  5 00:50:33 CST 2017 | Done conducting sensitivity test(s)
examining the effect of varying the 'prior' probabilities of known observations
in the beliefs matrix supplied to bgmm, using bgmmSensTest.
INFO      | Thu Jan  5 00:50:33 CST 2017 | Bye.
```

Under this scenario, comparisons also will not be made between test runs, and output files (e.g. PDFs) present in the "Ex2_bgmmSensTest" example run folder will not be created during your sensitivity analysis! Avoid these issues by always passing a "2" or "3" option to the script using the -b flag.

ACKNOWLEDGEMENTS

During the development of this software, J.C.B. received stipend support from a Ciência Sem Fronteiras (Science Without Borders) postdoctoral fellowship from the Brazilian Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq; Processo 314724/2014-1). Lab and computer

space was also supplied by The University of Alabama, during an internship in the Lozier Lab in the UA Department of Biological Sciences.

REFERENCES

- Biecek P, Szczurek E, Vingron M, Tiuryn J (2012) The R package bgmm: mixture modeling with uncertain knowledge. Journal of Statistical Software, 47, 1–31.
- Edwards DL, Knowles LL (2014) Species detection and individual assignment in species delimitation: can integrative data increase efficacy? Proceedings of the Royal Society B, 281, 20132765.
- Gower JC (1971) A general coefficient of similarity and some of its properties. Biometrics, 27, 857–871. doi:10.2307/2528823
- Hausdorf B, Hennig C (2010). Species delimitation using dominant and codominant multilocus markers. Systematic Biology, 59, 491–503.
- Lebret R, Iovleff S, Langrognet F, Biernacki C, Celeux G, Govaert G (2015) Rmixmod: the R package of the model-based unsupervised, supervised, and semi-supervised classification Mixmod Library. Journal of Statistical Software, 67(6). doi:10.18637/jss.v067.i06
- Szczurek E, Biecek P, Tiuryn J, Vingron M (2010) Introducing knowledge into differential expression analysis. Journal of Computational Biology, 17, 953–967.

OTHER READINGS

Users should familiarize themselves with NMDS and GMMs before using GaussClust. Here are some useful resources:

- NMDS Wikipedia overview (https://en.wikipedia.org/wiki/Multidimensional_scaling)
- NMDS tutorial (http://strata.uga.edu/software/pdf/mdsTutorial.pdf) from the University of Georgia
- Unconstrained Ordination Techniques lecture (http://www.umass.edu/landeco/teaching/multivariate/schedule/ordination2.pdf) from Univ. Massachusetts
- GMM Wikipedia overview (https://en.wikipedia.org/wiki/Mixture_model)
- What are the true clusters? (https://128.84.21.199/pdf/1502.02555v1.pdf) article by Christian Hennig

R GMM, clustering, and NMDS tutorials:

- http://tinyheero.github.io/2015/10/13/mixture-model.html
- http://tinyheero.github.io/2016/01/03/gmm-em.html
- https://www.r-bloggers.com/fitting-mixture-distributions-with-the-r-package-mixtools/
- https://www.google.com/search?client=safari&rls=en&q=multivariate+partially+supervised+gaussian+mixture+model&ie=8&oe=UTF-8
- http://www.statmethods.net/advstats/cluster.html
- https://rstudio-pubs-static.s3.amazonaws.com/58843_d17d5721f5254ac3bbeea211466c0bb7.html

TODO LIST

Current:

- **Make code work on -k values other than 4, so user is free to specify any number of retained NMDS dimensions!**
- **Add example output graphics / screenshots to README!**
- **For bgmmSensTest.sh, add a simple test of whether originally labeled observations are "re-labeled", or not (ideally would not be; see Biecek et al. 2012)**
- Look into model selection for belief-based GMMs, over range of k values
- Explore different methods for BIC calculation (e.g. in mclust) during unsupervised clustering.
- Explain defaults better.
- Would be nice if not giving a flag meant the corresponding analysis was not run (equivalent to passing '0' for some of the options).
- Provide warnings to user, with a particularly important example being to warn when the number of variables is less than k (number of clusters), in which case NMDS will surely fail to produce good results!

Recently finished/fixed:

- Solve two input file problem. **DONE!** :white_check_mark:
- Make GaussClust script do more with bgmm, including semisupervised analysis using belief probs matrix**. DONE!** :white_check_mark:
- Change Usage section of README to include code for working with example files**. Partially DONE!**
- Find a better NMDS function than isoMDS in labdsv package. **DONE!** :white_check_mark:
- Change bgmmSensTest from interactive to non-interactive, and make it produce PDF plots. **DONE!** :white_check_mark:

January 9, 2017
Justin C. Bagley, Tuscaloosa, AL, USA