Manual

Details of the routines developed. Example Notebook demonstrates their use.

The code is in the 3 libraries :

- lda_alphas_model.py
- lda_alpha_stats.py
- lda_topic_annotate.py

1. The TopicModel object is used in routines , this is initialised when loading the model.

The instance variables .
K - the number of topics
F - the number of Files/samples
alphas - array of F x K
topic_dict - dictionaray key is the topic , value is a dictionary of words
topic_index - index of the order of topics in the array
sampleIDs - this is a list of the samples that were loaded in the same order as they occur in alphas array
groups - [[group1,group2]] list of booleans length F . True indicates sample in group.
groupIDs - [[group1Ids],[group2Ids]] hold the sample ids for each group, used in plotting.
clusters - a list length K , to indicate which cluster a topic is in.

The TopicModel is initialised when loading a model. This can either Joe's or Simon's model or from a pre-saved model.

BvU = TopicModel(dict_file, model_type='nb',alpha_exp = False)
Or from a pickled object.
BvU = TopicModel.load_alphas('BvU.obj')

The parameter for initialisation are

model_type = 'nb' default is Simon's model.
'gs' to load Joe's model.
sampleIDs = [] , This is loaded automatically for Simon's model, otherwise can either add here or l later with set_sampleIDs
alpha_exp = True . if you want to use raw alpha values set to False.

Setting the groups and groupIDs
This will be initialised so that all the samples are in one group.
This can be set by group names for example 'Beer' and 'Urine' or from groupings that come from the statistics routines.

BvU.set_groups (groupings = [], names = []) only use either groupings or names.

2. Plotting routines

Interactive PCA plot:

```
BvU.plot_PCA ( title = '')
```

Boxplots:

```
BvU.topic_boxplots(interesting,group_names = [] , anno = [], title = ''):
```

interesting - a list of the topic/motif numbers (needs changing to use topic_index for flexibility)

group_names a list of 2 strings, if setting groups from Stats then add a meaning ful name for the legend (defaualt is group1,group2)

anno – used for producing a plot to add annotation names for labelling the motifs. Added for Justin's plot of biochemically relevant motifs. (see Notebook

3. Statistic Routine   in lda_alpha_stats.py

```
alpha_stats (BvU.alphas, groups, significant = 0.05):
```

Method to calculate the statistical significance between 2 groups from the alphas

Parameters:

alphas
groups (for 2 groups only) a list of booleasn group1 == true group2 == False)

Returns

a data frame with FC , T-test pvalues, Mann-Whitney p-values
and p-adjusted values for each topic

4. Annotation Routines   in lda_topic_annotate.py

An annotation object is created with:-

```
annoFiles = ['urine.csv','beer.csv']
anno = Annotation(annoFiles, ['U_','B_'])
```

Display interesting motifs with:-

```
anno.display_hits(BvU, interesting)   # interesting is a list of motifs.
```

Save a complete list of matched annotations (and the words only of unmatched motifs)

```
anno.write_csv(hits,'BvU_large.csv')
```

(Needs the parameter used in get_hits (self,model,topics=[],ppm_tolerance = 20, word_threshold = 0.001,score_threshold = 0.3) to be moved so accessible from here.)