Testing My LDA

Georgie Mansell

02/08/2021

Introduction

MSLDA is now an R package on github (https://github.com/g-l-mansell/MSLDA) which contains my different implementations of LDA:

```
library(MSLDA)
library(tidyverse)
```

- lda_original - as in the Blei 2003 paper, but edited to force $\alpha_i > 0$ - the only implementation that does not run in parallel
- lda_original_par - as above but the E-step is run in parallel - should give the same results with the same seed.
- lda_noalpha - since the alpha update rule in LDA_original is flawed, this version treats alpha as a hyperparameter which should be tuned.
- lda_reshaped - this version is a further adaptation of LDA_original, which uses a count (document-term) matrix as an input rather than the document vectors - this should give the same results as LDA_noalpha with improved speed.
- lda_smoothed - as in the Hoffman 2010 paper (batch LDA section), with edited equation for $\mathcal{L}$ - this version assumes beta is a random variable.

This script will then contain the analysis, so everything is in one place to be easily rerun. Whether to rerun the analyses or just load the results of past runs will be controlled here:

```
rerun_original <- T
rerun_reshaped <- T
rerun_smoothed <- T
```

To do

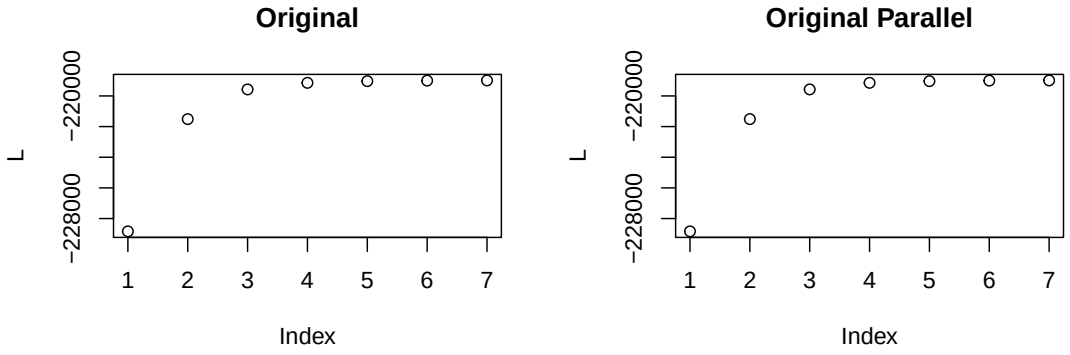
- Try to implement one of the methods in Rcpp (with RcppParallel?)
- Try to follow Colins version with Poison prior
- Read about sparse LDA

Check all 5 implementations work

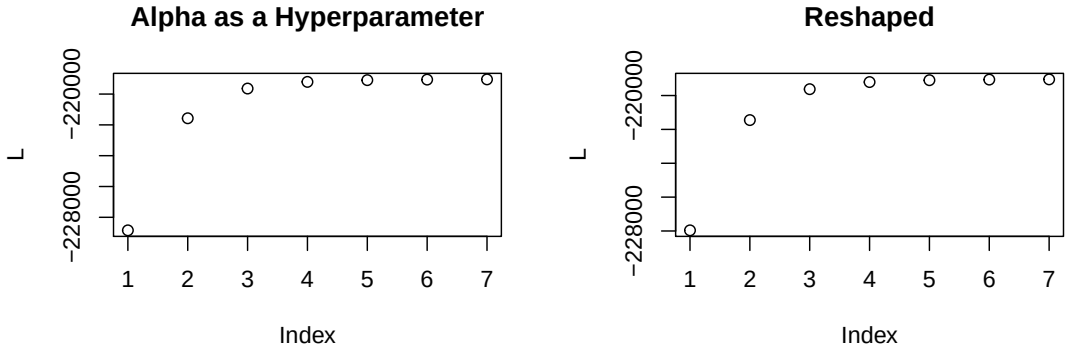
using the simulated dataset of documents and the same seed.

```
load("data/MyCorpus.Rdata")
par(mfrow=c(1, 2))

res1 <- lda_original(docs, K=3, seed=83)
plot(res1$Ls, ylab="L", main="Original")
```

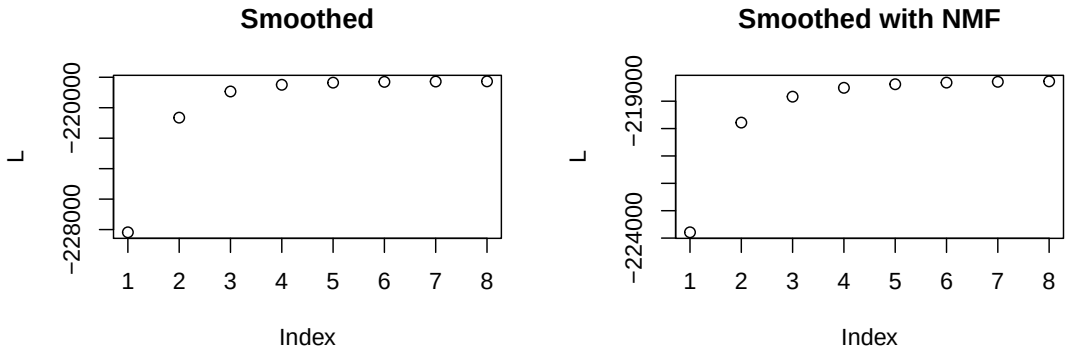
```
res2 <- lda_original_par(docs, K=3, seed=83)
plot(res2$Ls, ylab="L", main="Original Parallel")
```




```
res3 <- lda_noalpha(docs, K=3, seed=83)
plot(res3$Ls, ylab="L", main="Alpha as a Hyperparameter")

res4 <- lda_reshaped(counts, K=3, seed=83)
plot(res4$Ls, ylab="L", main="Reshaped")
```



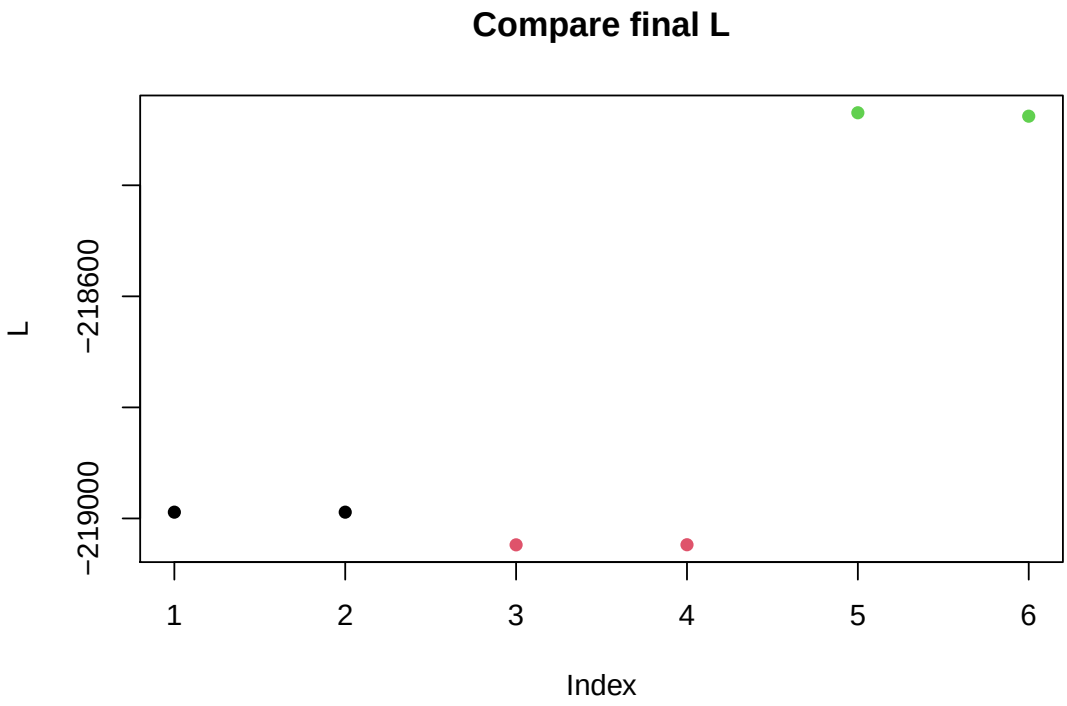

```
res5 <- lda_smoothed(counts, K=3, seed=83, NMF=F)
plot(res5$Ls, ylab="L", main="Smoothed")

res6 <- lda_smoothed(counts, K=3, seed=83, NMF=T)
plot(res6$Ls, ylab="L", main="Smoothed with NMF")
```




```
(Ls <- sapply(list(res1, res2, res3, res4, res5, res6), function(res) res$L))
```

```
## [1] -218988.7 -218988.7 -219047.5 -219047.3 -218269.4 -218275.6
```

```
plot(Ls, ylab="L", main="Compare final L", col=c(1, 1, 2, 2, 3, 3), pch=16)
```




```
#i thought res3$L and res4$L should be exactly the same, but 1dp off isnt too bad?
#interestingly lda_smoothed seems to have performed the best, and nmf initalisation makes it slightly wors
```

Now we know all 5 work as expected, we can just look at the 3 main ones: lda_orginal_par, lda_reshaped, and lda_smoothed.

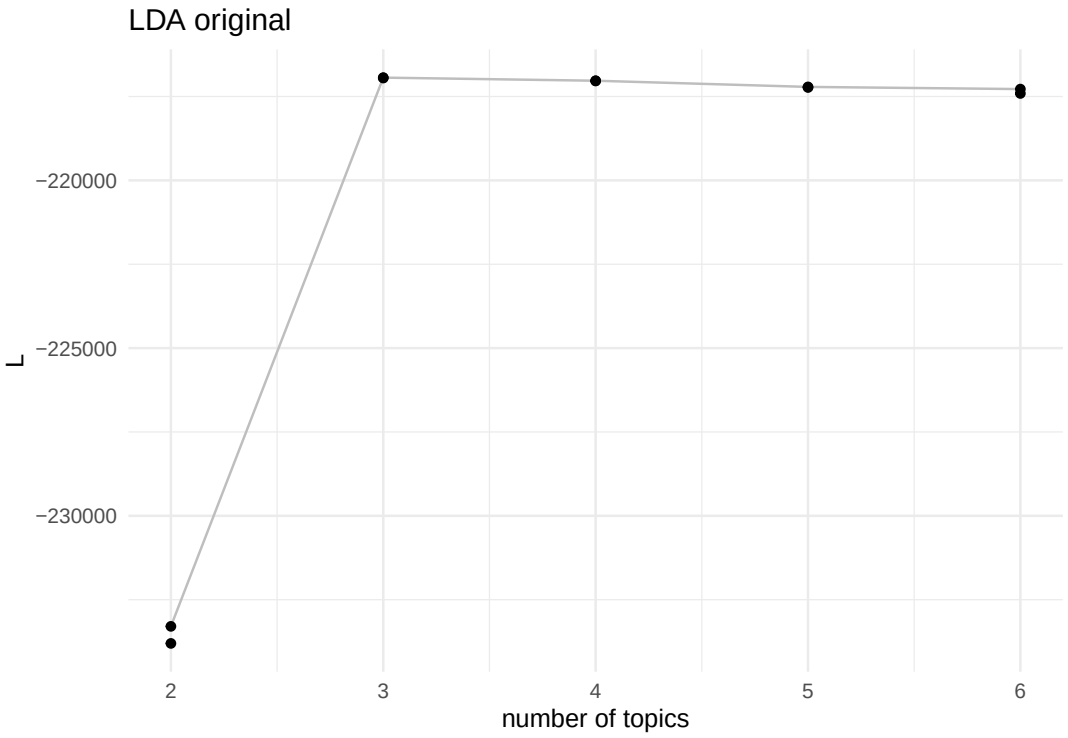
Text data

Running these 3 implementations multiple times with different numbers of topics (expecting a peak at K=3), saving the final values of L to `resK`, and keeping the full results of the runs with the highest L.

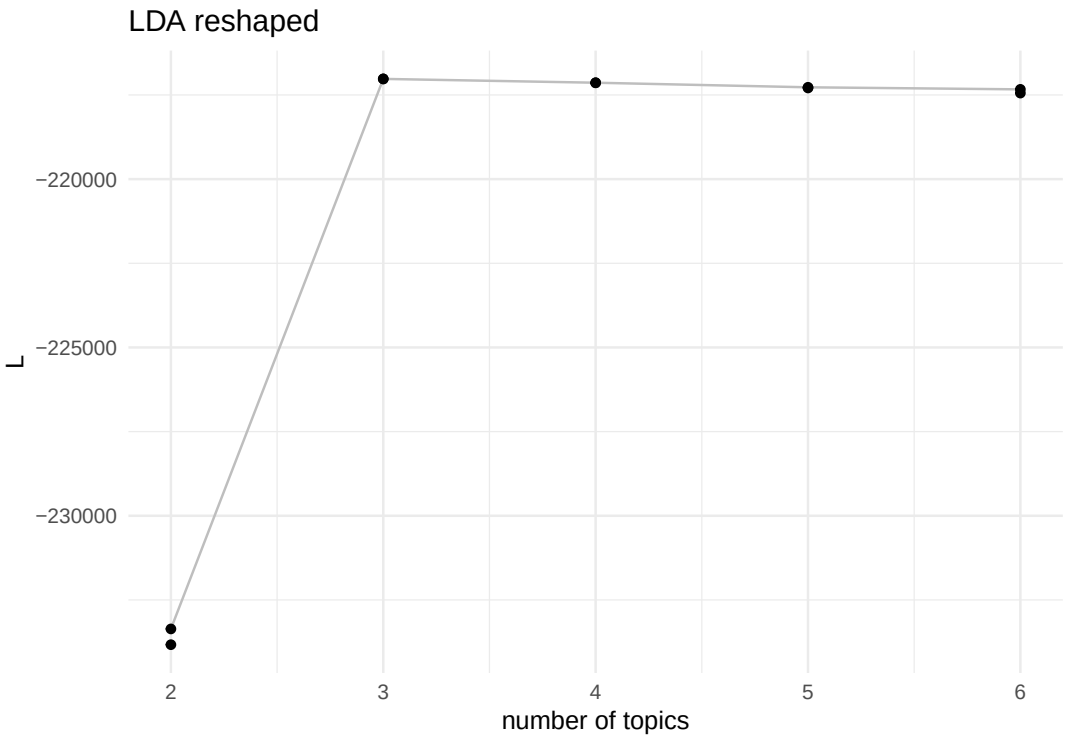
```
plot_LvK <- function(res, Ks){
  max_line <- data.frame(K=Ks, Max=apply(res, 1, max))
  res <- data.frame(K=Ks, res)
  res_lls <- pivot_longer(res, cols=-"K", names_to="Rep", values_to="L")

  p <- ggplot(res_lls, aes(x=K, y=L, group=1))+
    geom_line(data=max_line, aes(x=K, y=Max), colour="grey") +
    geom_point() +
    labs(x="number of topics", y="L") +
    theme_minimal()

  return(p)
}
```

```
Ks <- 2:6
reps <- 2

res1_LvK <- res2_LvK <- res3_LvK <- matrix(NA, length(Ks), reps)
res1_best <- res2_best <- res3_best <- list("L"=-Inf)

for(i in 1:reps){
```

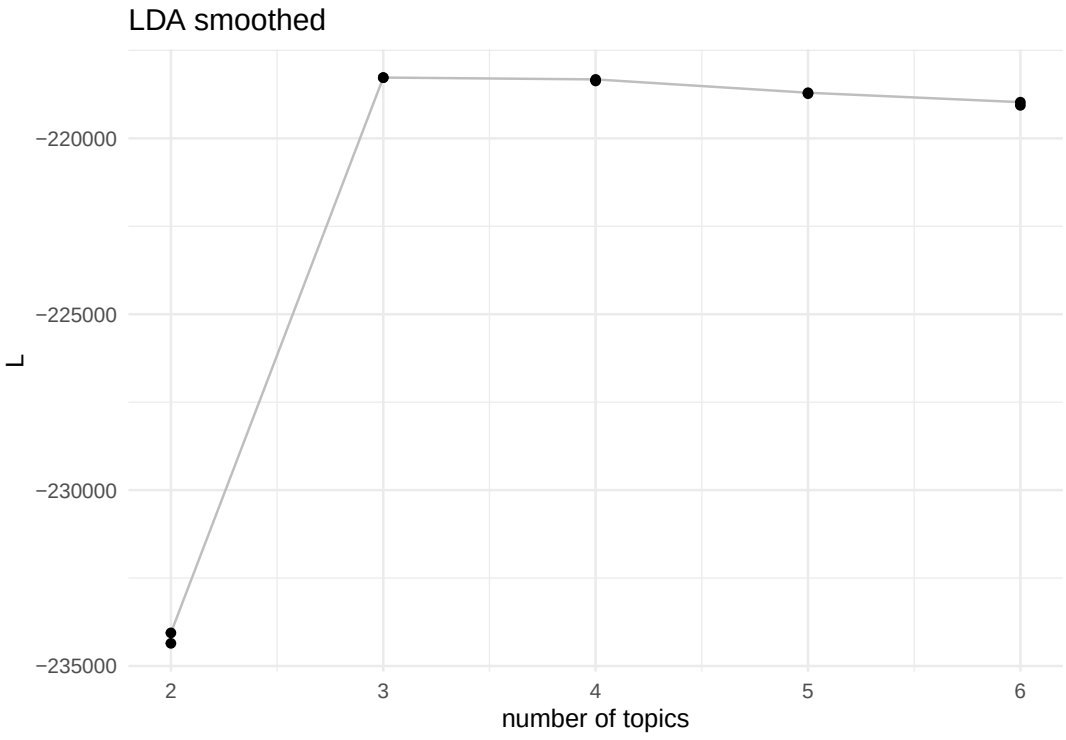
```
  for(j in 1:length(Ks)){
    temp <- lda_original_par(docs, K=Ks[j], seed=i)
    res1_LvK[j, i] <- temp$L
    if(temp$L > res1_best$L) res1_best <- temp

    temp <- lda_reshaped(counts, K=Ks[j], seed=i)
    res2_LvK[j, i] <- temp$L
    if(temp$L > res2_best$L) res2_best <- temp

    temp <- lda_smoothed(counts, K=Ks[j], seed=i)
    res3_LvK[j, i] <- temp$L
    if(temp$L > res3_best$L) res3_best <- temp
  }
}

plot_LvK(res1_LvK, Ks) + labs(title="LDA original")
```




```
plot_LvK(res2_LvK, Ks) + labs(title="LDA reshaped")
```




```
plot_LvK(res3_LvK, Ks) + labs(title="LDA smoothed")
```



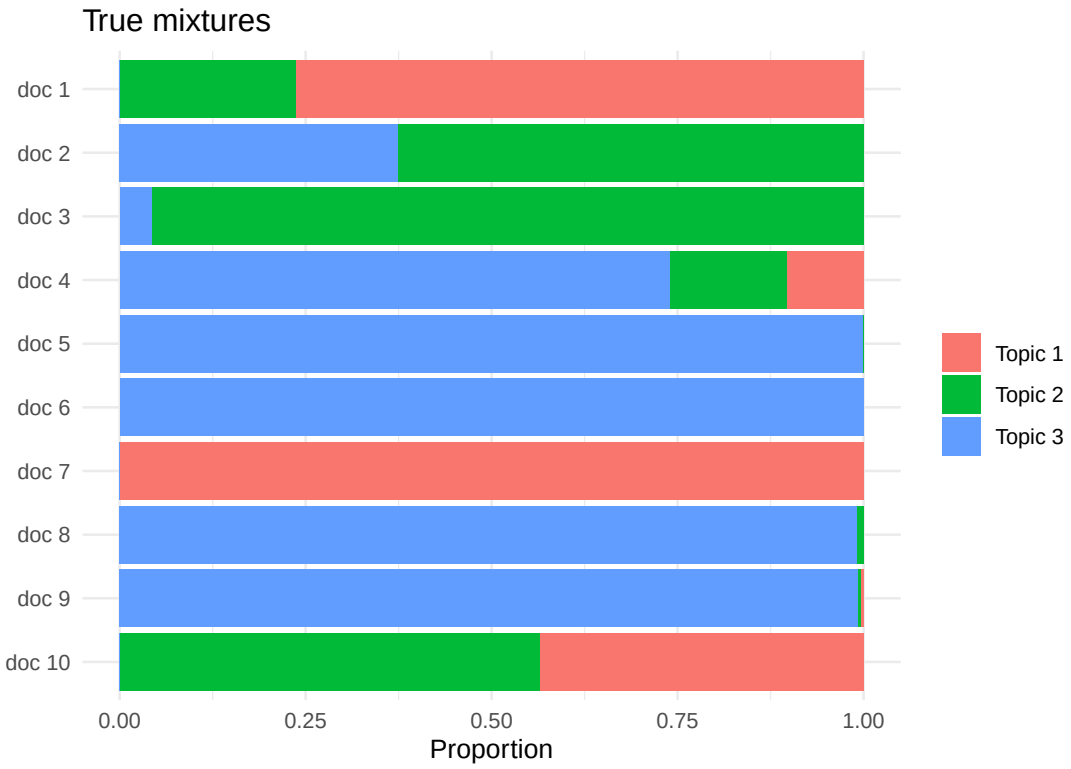
All 3 implementations have a peak at K=3 as expected!

Now comparing the estimated mixing proportions of the best runs

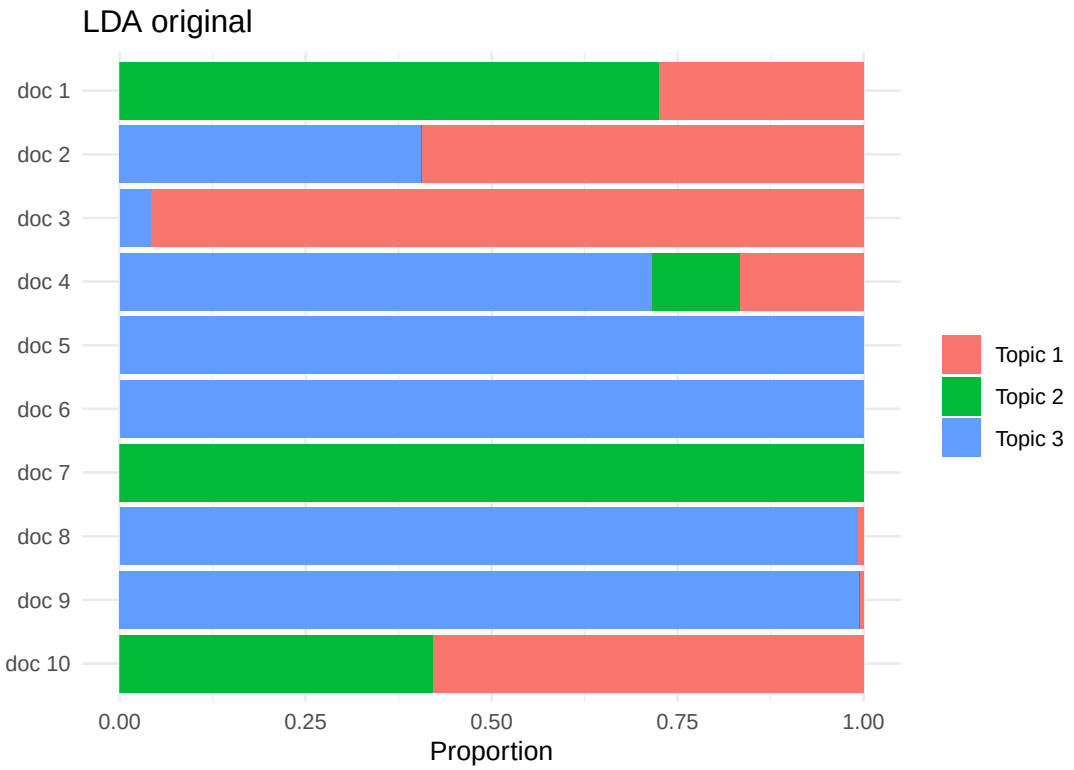
```
plot_mixture <- function(dat, nsamples=10, sample_labels=NULL, topic_label=1:ncol(dat), width=0.9){

  if(is.null(sample_labels)) sample_labels <- paste("doc", 1:nsamples)
  Sample <- factor(sample_labels, levels=rev(unique(sample_labels)))
```

```
  plot_dat <- as.data.frame(dat)
  plot_dat <- plot_dat[1:nsamples,]
  colnames(plot_dat) <- paste("Topic", topic_label)
  plot_dat <- cbind(plot_dat, Sample)

  plot_dat <- plot_dat %>%
    pivot_longer(cols=-"Sample", names_to = "Topic", values_to="Proportion")

  p <- ggplot(plot_dat, aes(fill=Topic, x=Sample, y=Proportion)) +
      geom_bar(position="fill", stat="identity", width=width) +
      coord_flip() +
      labs(x="", fill="") +
      theme_minimal() #+
      #theme(text = element_text(size = 14))

  return(p)
}
```

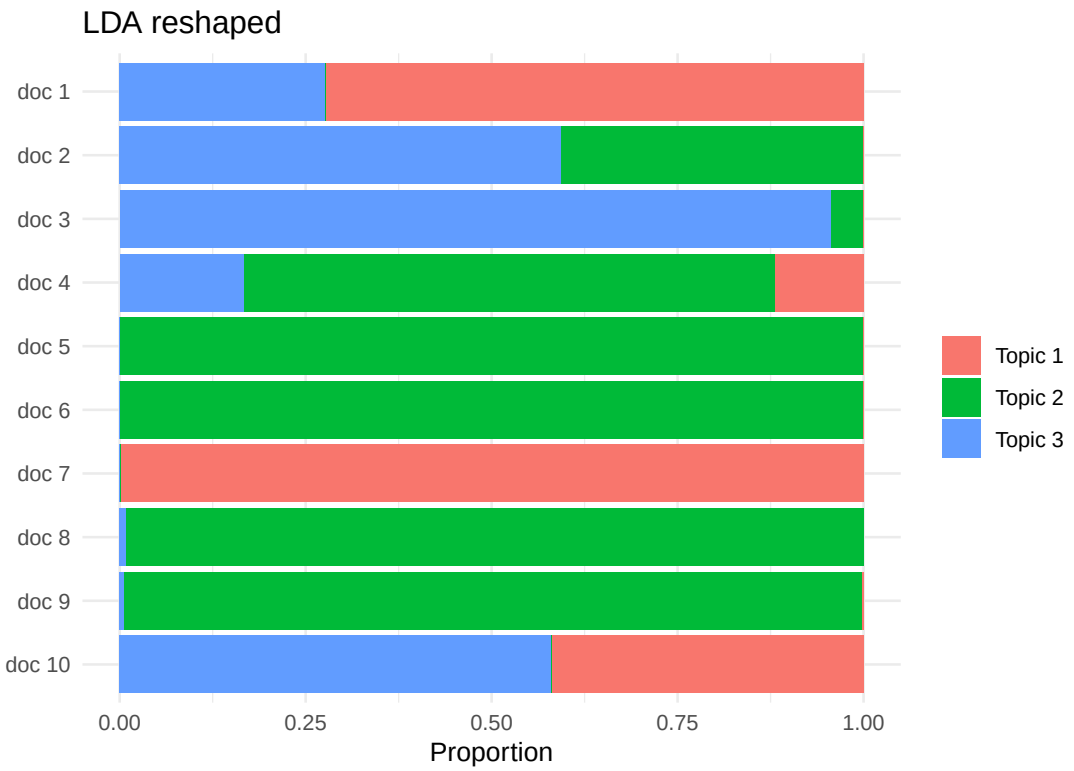
```
plot_mixture(thetas_true) + labs(title="True mixtures")
```



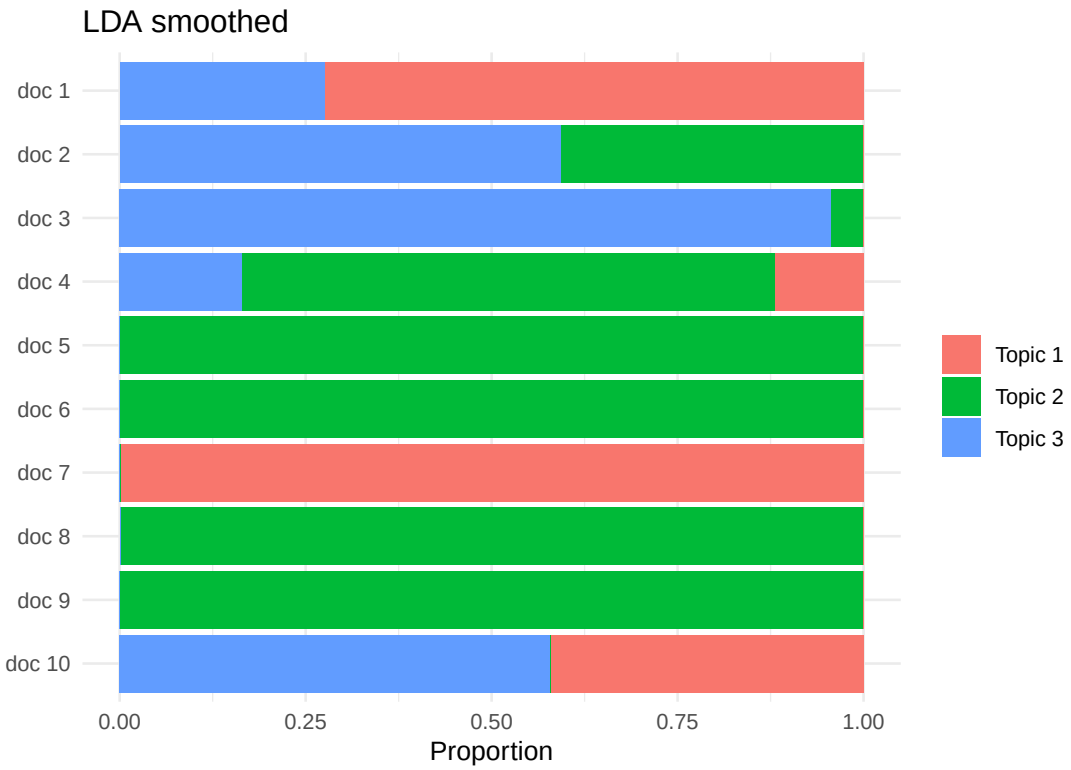

```
plot_mixture(res1_best$thetas) + labs(title="LDA original")
```




```
plot_mixture(res2_best$thetas) + labs(title="LDA reshaped")
```




```
plot_mixture(res3_best$thetas) + labs(title="LDA smoothed")
```

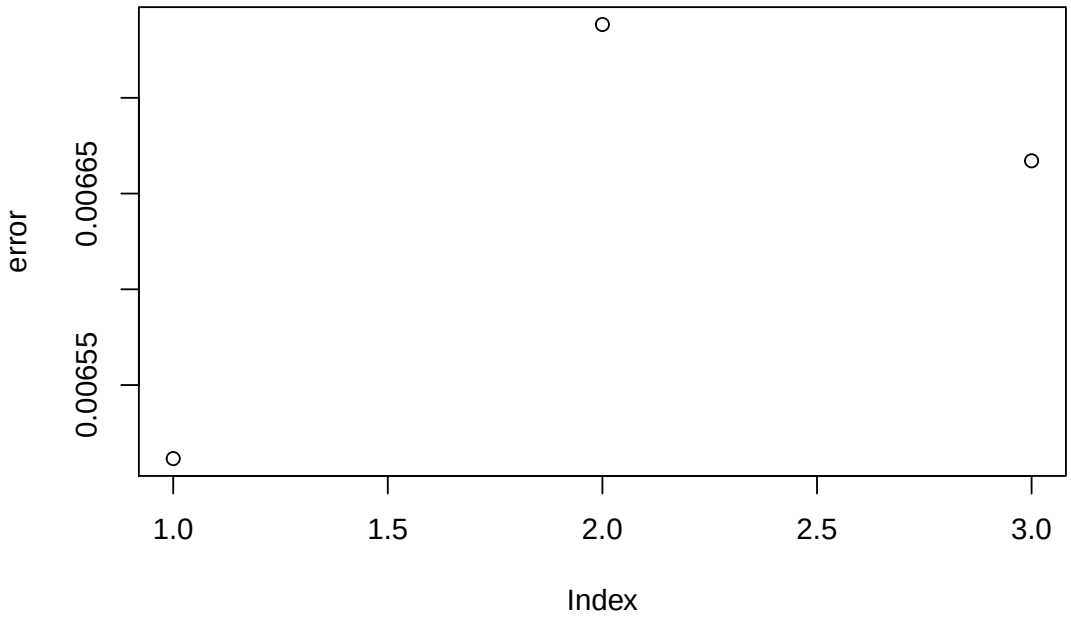


Matching up the topics and comparing the MAE

```
error <- rep(NA, 3)
true_max <- apply(thetas_true, 1, which.max)
mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

for(i in 1:3){
  res <- get(paste0("res", i, "_best"))$thetas
  res <- res / rowSums(res)

  model_max <- apply(res, 1, which.max)
  order <- rep(NA, 3)
  for(j in 1:3){
    samples <- which(true_max == j)
    order[j] <- mode(model_max[samples])
  }
  res <- res[,order]
  error[i] <- mean(abs(thetas_true-res))
}

print(error)
```

```
## [1] 0.006511582 0.006738303 0.006667106
```

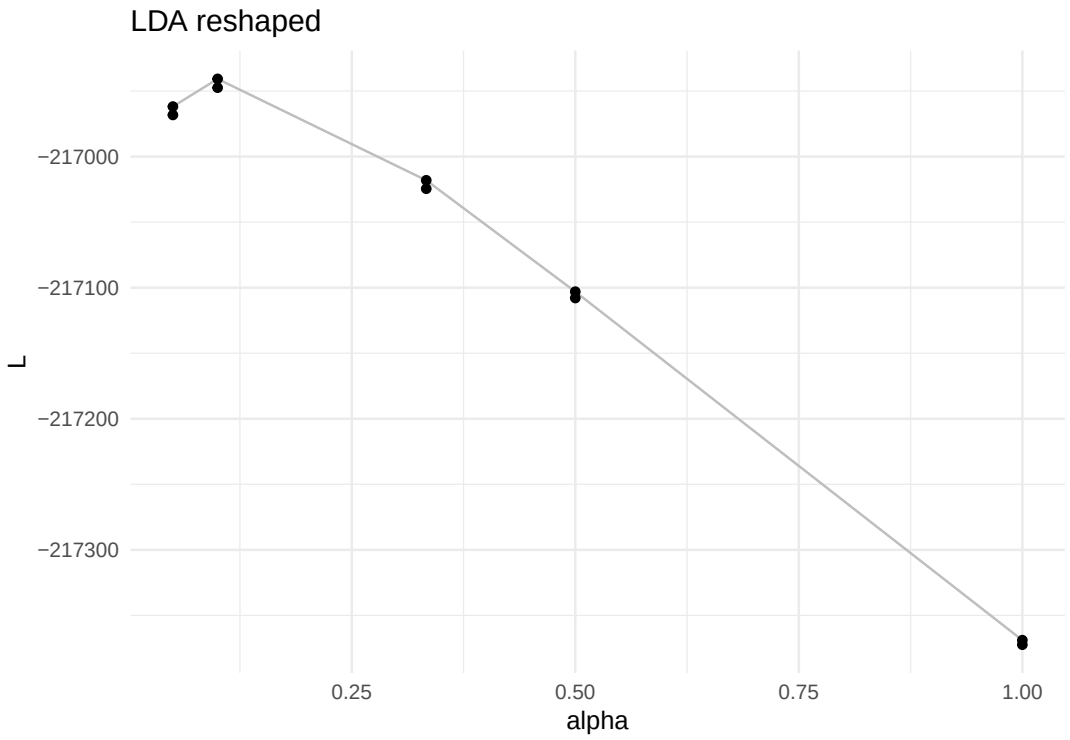
```
plot(error)
```



So LDA original appears to have performed best here. That could be due to it being able to tune α whereas the other two implementations are given a default value of 1/K (and the true value is 0.1).

Testing tuning alpha

```
alphas <- c(0.05, 0.1, 1/3, 0.5, 1)
reps <- 2
res_LvA <- matrix(NA, length(alphas), reps)

for(i in 1:reps){
  for(j in 1:length(alphas)){
    temp <- lda_reshaped(counts, K=3, alpha=alphas[j], seed=i*5)
    res_LvA[j, i] <- temp$L
  }
}

plot_LvK(res_LvA, alphas) + labs(title="LDA reshaped", x="alpha")
```



This has a peak at alpha=0.1 as we'd expect!

lda_smoothed also has another hyperparameter η which can be tuned, although I dont know what value we'd expect.

Spectra

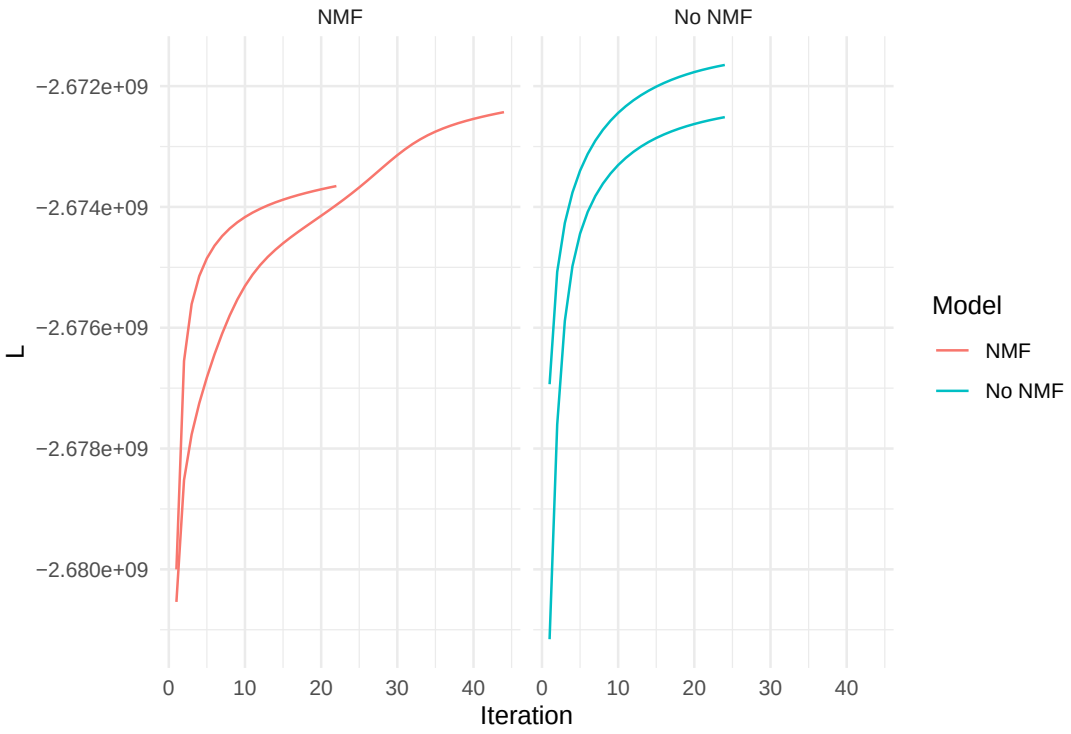
Now moving onto the dataset of bacteria spectra

We can no longer use the lda_original methods as these use document vectors, but we can use lda_reshaped which is very similar

For lda_smoothed, compare running with and without NMF initialisation

```
load("data/Spectra.Rdata")

reps <- 2
K <- 8
res_lls <- matrix(NA, reps*2, 50)
thresh <- 1e-5 #adjusting the threshold because 1e-4 didnt quite seem converged enough

for(i in 1:reps){
  temp <- lda_smoothed(counts, K, seed=i*3, NMF=F, thresh=thresh)
  res_lls[i, 1:length(temp$Ls)] <- temp$Ls

  temp <- lda_smoothed(counts, K, seed=i*6, NMF=T, thresh=thresh)
  res_lls[reps+i, 1:length(temp$Ls)] <- temp$Ls
}

colnames(res_lls) <- 1:50

res_lls2 <- res_lls %>%
  as.data.frame %>%
  mutate(Model=factor(c(rep("No NMF", reps), rep("NMF", reps))),
```

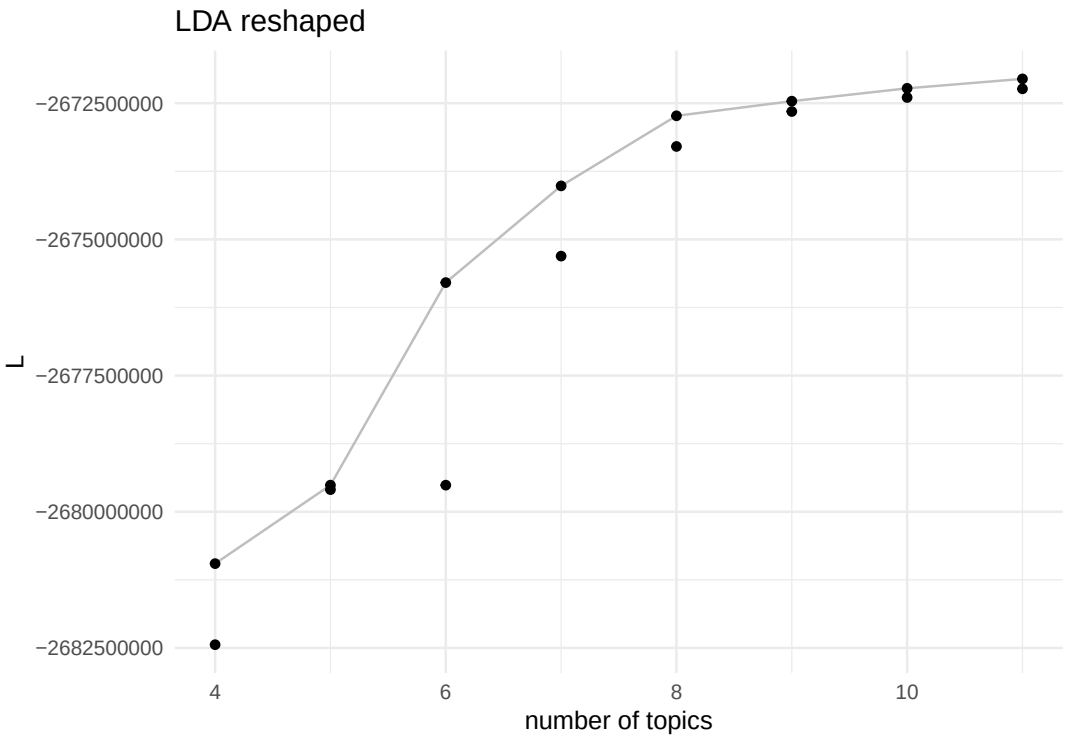
```
        Run=factor(rep(1:reps, 2))) %>%
  pivot_longer(cols=-c("Model", "Run"), values_to="L", names_to="Iteration") %>%
  filter(!is.na(L)) %>%
  mutate(Iteration=as.numeric(Iteration))

ggplot(res_lls2, aes(x=Iteration, y=L, color=Model, group=Run)) +
  geom_line() +
  facet_wrap(~Model) +
  theme_minimal()
```



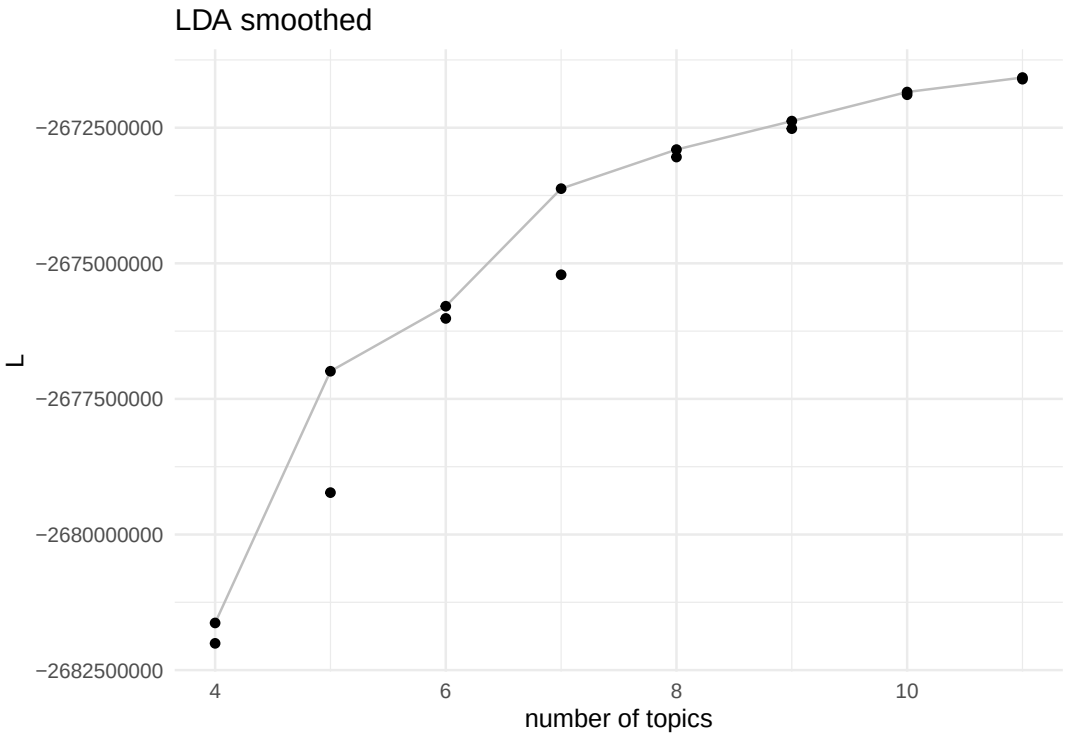
Not sure why NMF makes it perform worse, but we can just continue with the not NMF version.

Try to final optimal K

```
Ks <- 4:11
reps <- 2

res1_LvK <- res2_LvK <- matrix(NA, length(Ks), reps)
res1_best <- res2_best <- list("L"=-Inf)

for(i in 1:reps){
  for(j in 1:length(Ks)){
    temp <- lda_reshaped(counts, K=Ks[j], seed=i)
    res1_LvK[j, i] <- temp$L
    if(temp$L > res1_best$L) res1_best <- temp

    temp <- lda_smoothed(counts, K=Ks[j], seed=i+1)
    res2_LvK[j, i] <- temp$L
    if(temp$L > res2_best$L) res2_best <- temp
  }
}
plot_LvK(res1_LvK, Ks) + labs(title="LDA reshaped")
```




```
plot_LvK(res2_LvK, Ks) + labs(title="LDA smoothed")
```

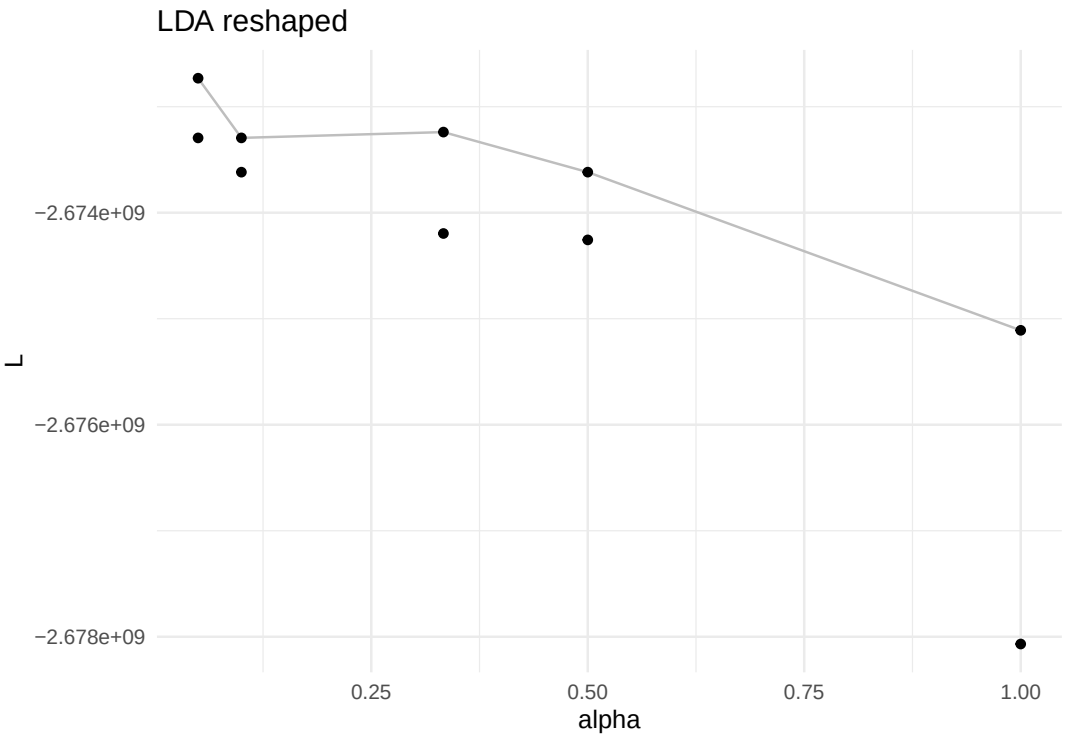


It's good that our results are comparable, but annoying there is not a clear peak. It could be argued that lda_reshaped has an 'elbow' at K=8.

For lda_reshaped try to find optimal alpha

```
alphas <- c(0.05, 0.1, 1/3, 0.5, 1)
reps <- 2
res_LvA <- matrix(NA, length(alphas), reps)
```

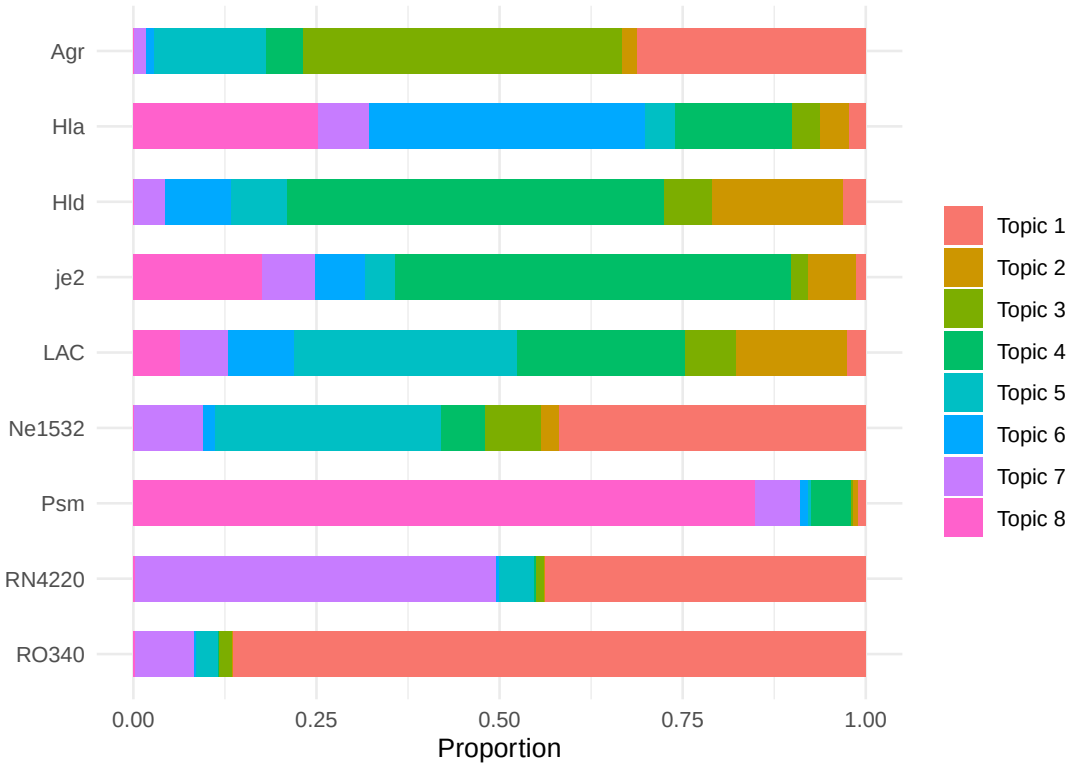
```
res3_best <- list("L"=-Inf)

for(i in 1:reps){
  for(j in 1:length(alphas)){
    temp <- lda_reshaped(counts, K=8, alpha=alphas[j], seed=i*j)
    res_LvA[j, i] <- temp$L
    if(temp$L > res3_best$L) res3_best <- temp
  }
}

plot_LvK(res_LvA, alphas) + labs(title="LDA reshaped", x="alpha")
```



Looks like 0.05 is a good value of alpha.

Visualise the results of this model

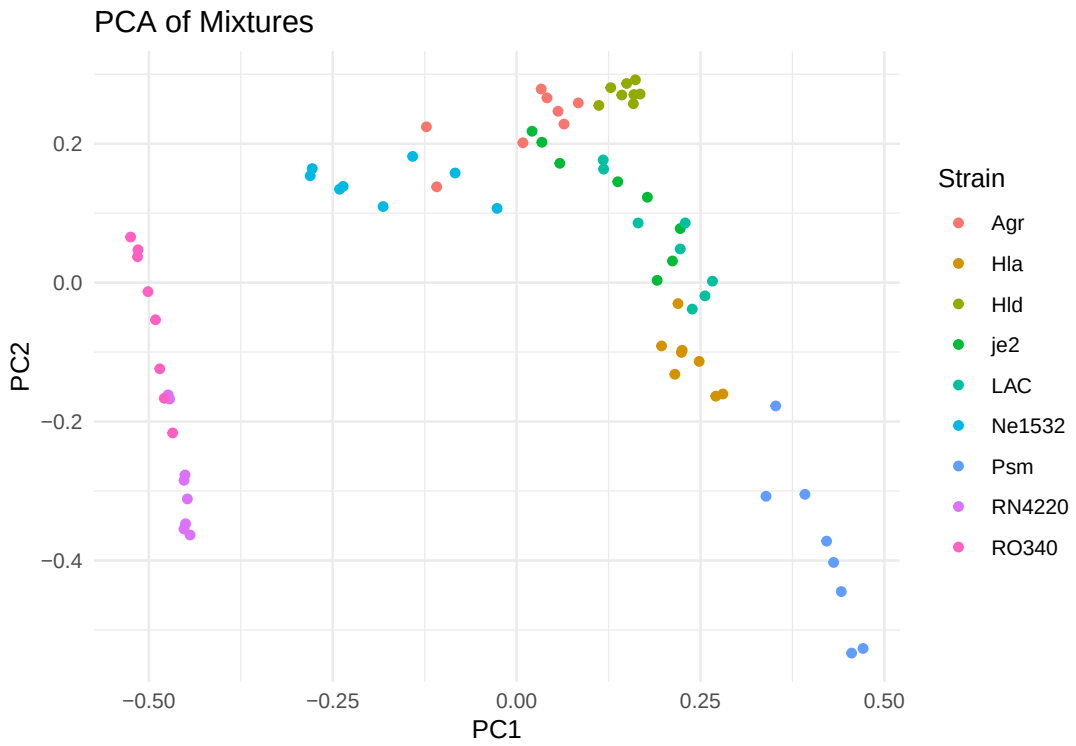
```
thetas <- res3_best$thetas
#plot_mixture(thetas, nsamples=72, sample_labels = paste(idx, rep(1:8, 9)), width=0.4)
plot_mixture(thetas[seq(1, 72, 8),], nsamples=9, sample_labels=idx[seq(1, 72, 8)], width=0.6)
```



Psm and RO340 are clearly associated with Topics 8 and 1 respectively.

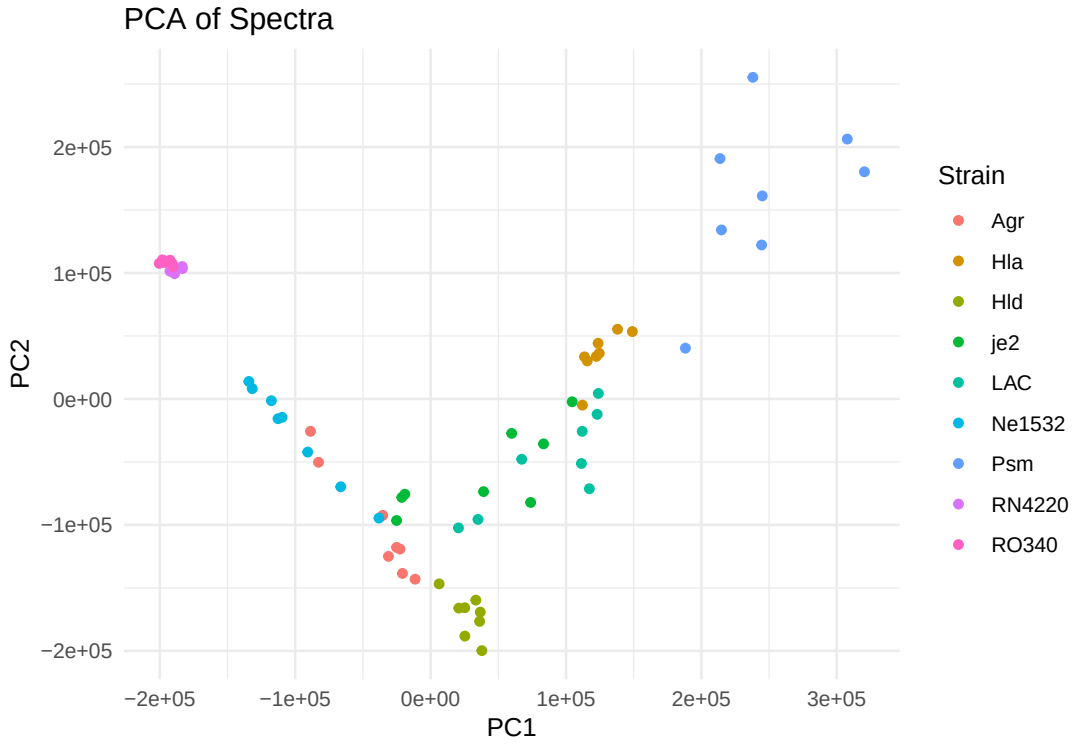
Visualising the results for all the samples using PCA

```
prcomp(thetas)$x[, 1:2] %>%
  as.data.frame %>%
  mutate(Strain=idx) %>%
  ggplot(aes(x=PC1, y=PC2, colour=Strain)) +
    geom_point() +
    theme_minimal() +
    labs(title="PCA of Mixtures")
```



Then compare that to a PCA of the count matrix directly, they are pretty similar...

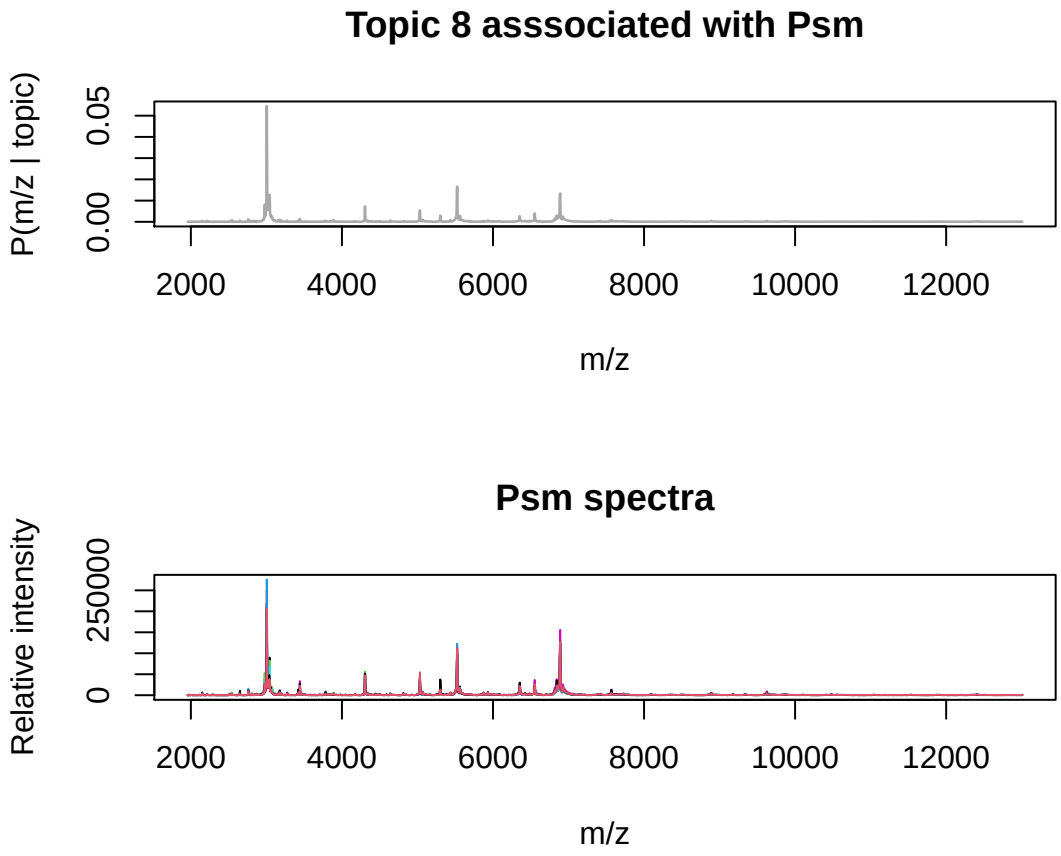
```
prcomp(counts)$x[, 1:2] %>%
  as.data.frame %>%
  mutate(Strain=idx) %>%
  ggplot(aes(x=PC1, y=PC2, colour=Strain)) +
    geom_point() +
    theme_minimal() +
    labs(title="PCA of Spectra")
```



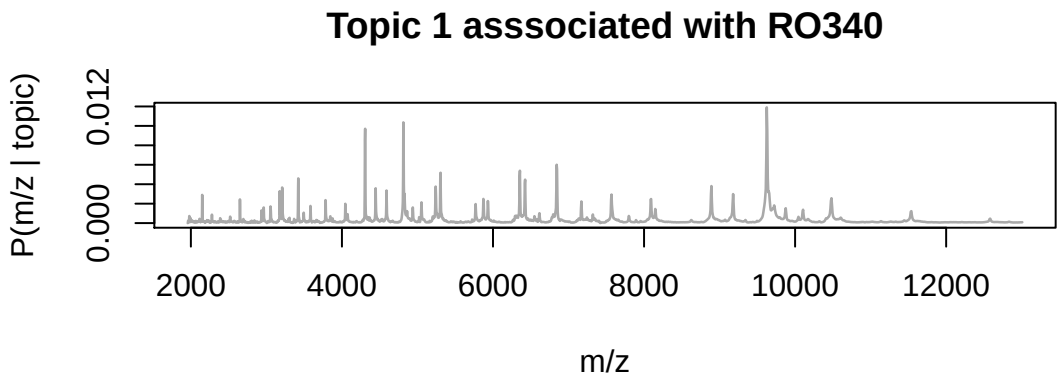
Plot the topic distributions, and compare to the spectra

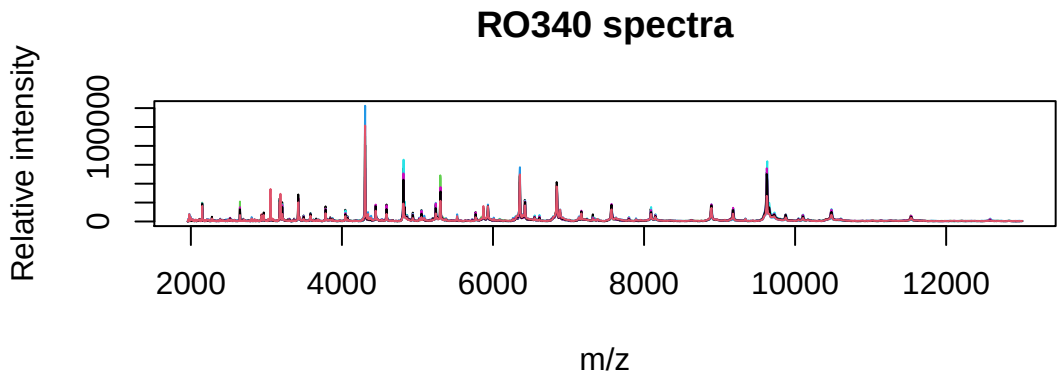
```
compare_topic_spectra <- function(beta, counts, mz, topic, strain){
  par(mfrow=c(2, 1))
  plot(mz, beta[topic,], type="l", xlab="m/z", ylab="P(m/z | topic)",
     col="darkgrey", lwd=1.3, main=paste("Topic", topic, "asssociated with", strain))

  matplot(mz, t(counts[idx==strain,]), type="l", ylab="Relative intensity",
        xlab="m/z", lty=1, main=paste(strain, "spectra"))
}
```

```
beta <- res3_best$beta
compare_topic_spectra(beta, counts, mz_locations, topic=8, strain="Psm")
```

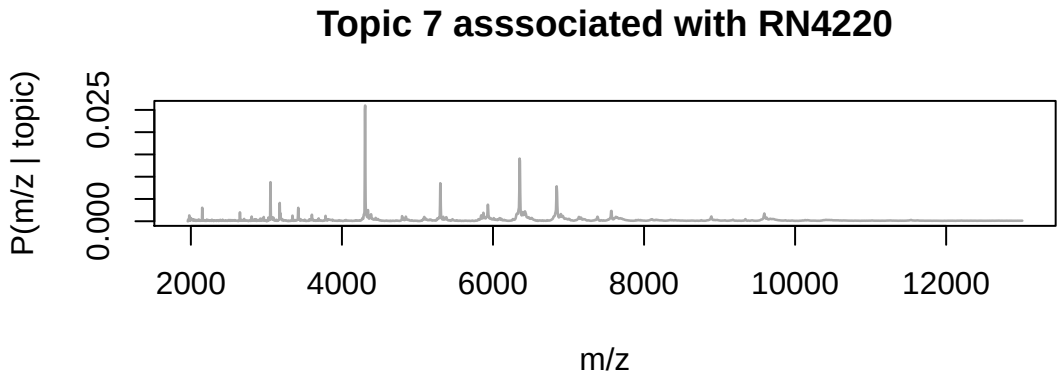


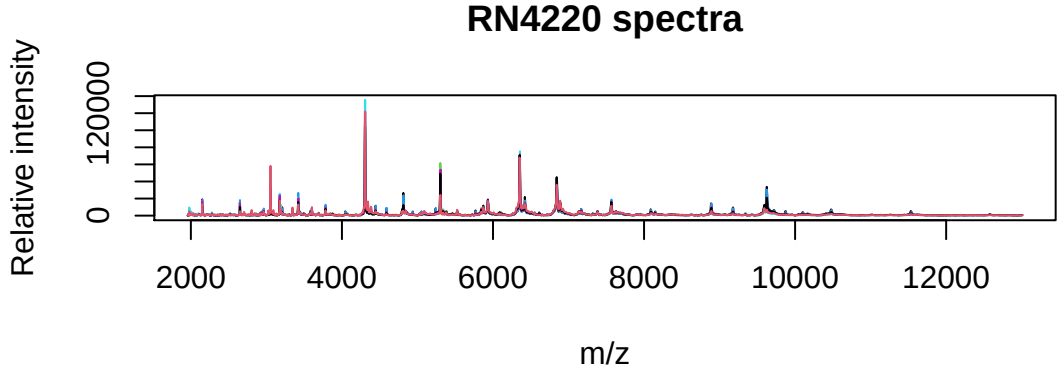

```
compare_topic_spectra(beta, counts, mz_locations, topic=1, strain="RO340")
```






```
compare_topic_spectra(beta, counts, mz_locations, topic=7, strain="RN4220")
```





```
compare_topic_spectra(beta, counts, mz_locations, topic=4, strain="Hld")
```