NMF

William Zhang, Eva, Jerry

2024-09-01

```
# load the packages
library(NMF)
library(tidyverse)
library(gridExtra)
library(readxl)
```

Procedure

1. Remove hourly observation with missing observation for any chemical
2. Remove background noise level using min values (except for chemicals with minimum value < 2*LOD and maximum value > 100*LOD)
3. Zero values are converted to a random value between 0 and 0.5*LOD
4. Normalize using 0th & 99th quantile
5. Compute weight matrix according to Guha's paper, without LOQ

Reading the data

```
# read the radon data
# Old:
# hourly_radon <- readRDS("hourly_radon.rds")
# New:
hourly_data <- readRDS("../DataProcessing/Trailer_hourly_merge_20240905.rds")
```

```
# remove NAs
hourly_nona <- hourly_data %>% select(-c(temp_bb,rhi, esf_bb, distToLovi,inv_dist,
                                          distToLovi_wells, monthly_oil, monthly_gas)) %>% na.omit()

# retrieving the vocs, removing everything else except the vocs
hourly_vocs <- hourly_nona %>% select(c("ethane", "ethene", "propane", "propene",
                                        "1_3-butadiene", "i-butane", "n-butane",
                                        "acetylene", "cyclopentane", "i-pentane",
                                        "n-pentane", "n-hexane", "isoprene", "n-heptane",
                                        "benzene", "n-octane", "toluene", "ethyl-benzene",
                                        "m&p-xylene", "o-xylene"))

# retrieving the non-vocs: co2_ppm, nox, ch4, h2s, so2, o3
# double check this
non_vocs <- c('ch4', 'co2_ppm', 'co', 'h2s', 'so2', 'nox', 'o3')
hourly_non_vocs <- hourly_nona %>% select(all_of(non_vocs))

hourly_full_nona <- cbind(hourly_non_vocs, hourly_vocs)
```

```r
# retrive a vector of yearmonth
hourly_dates <- hourly_nona %>%
  mutate(yearmonth = substring(day, 0, 7)) %>%
  pull(yearmonth)
```

Data preprocessing

```r
# Define LOD for each chemical
LOD_non_voc <- c('ch4' = 0.9,
             'co2_ppm' = 0.0433,
             'co' = 40,
             'h2s' = 0.4,
             'so2' = 0.4,
             'nox' = 0.05,
             'o3' = 1)

LOD_voc_monthly <- read_csv('../data/LNM_VOC_LOD_Rounded.csv') %>% select(-1)
# extract the yearmonth from date variables
LOD_voc_monthly <- LOD_voc_monthly %>%
  mutate(yearmonth = strftime(as.POSIXct(start_date, format = '%Y-%m-%d %H:%M:%S', tz = 'UTC'), '%Y-%m'

LOD_voc_monthly <- LOD_voc_monthly %>%
  select(-c(start_date, end_date)) %>%
  select(!any_of(ends_with('half_ldl')))

colnames(LOD_voc_monthly) <- str_replace_all(names(LOD_voc_monthly), '_ldl', '')

LOD_voc_avg <- read_xlsx('../data/LNM_VOC_Uncertainties.xlsx', skip = 1)
LOD_voc_avg <- LOD_voc_avg %>%
  select(1, 4) %>%
  rename('LOD' = 2, 'chemical' = 1) %>%
  head(20)
```

```r
# find the min for background-levels
background_levels <- sapply(hourly_full_nona, min)
background_levels
```

```
##           ch4       co2_ppm            co           h2s           so2
##      1928.000       411.300        61.630         0.200         0.200
##           nox            o3        ethane        ethene       propane
##         0.025         0.500         0.916         0.011         0.224
##       propene 1_3-butadiene      i-butane      n-butane     acetylene
##         0.009         0.007         0.035         0.090         0.019
##  cyclopentane     i-pentane     n-pentane      n-hexane      isoprene
##         0.005         0.038         0.042         0.021         0.005
##     n-heptane       benzene      n-octane       toluene ethyl-benzene
##         0.004         0.017         0.004         0.004         0.004
##    m&p-xylene      o-xylene
##         0.004         0.004
```

```r
get_info <- function(column) {
  N <- length(column)
  background <- quantile(column, 0)
  quantile1 <- quantile(column, 0.01)
  quantile99 <- quantile(column, 0.99)
  return(c(N, quantile1, quantile99, background))
}

info_table <- hourly_full_nona %>%
  reframe(across(everything(), ~ get_info(.x)))

info_table <- info_table %>%
  mutate(rownames = c('N', '1st percentile', '99th percentile', 'Background')) %>%
  pivot_longer(-rownames) %>%
  pivot_wider(names_from = rownames, values_from = value)

knitr::kable(info_table)
```

name	N	1st percentile	99th percentile	Background
ch4	4497	1963.40000	6318.81200	1928.000
co2_ppm	4497	417.09000	457.87120	411.300
co	4497	84.90720	444.04320	61.630
h2s	4497	0.20000	5.18084	0.200
so2	4497	0.20000	1.83896	0.200
nox	4497	0.22700	92.01080	0.025
o3	4497	0.50000	72.11200	0.500
ethane	4497	1.80852	536.67200	0.916
ethene	4497	0.01100	3.52212	0.011
propane	4497	0.81700	305.54000	0.224
propene	4497	0.00900	0.70228	0.009
1_3-butadiene	4497	0.00700	0.05904	0.007
i-butane	4497	0.14496	63.53760	0.035
n-butane	4497	0.34792	171.37600	0.090
acetylene	4497	0.04900	2.66204	0.019
cyclopentane	4497	0.00500	3.12356	0.005
i-pentane	4497	0.10396	51.02080	0.038
n-pentane	4497	0.10300	58.10280	0.042
n-hexane	4497	0.04196	18.32640	0.021
isoprene	4497	0.00500	0.03204	0.005
n-heptane	4497	0.01500	6.58924	0.004
benzene	4497	0.02700	3.87512	0.017
n-octane	4497	0.00400	2.01452	0.004
toluene	4497	0.01296	3.53640	0.004
ethyl-benzene	4497	0.00400	0.31604	0.004
m&p-xylene	4497	0.00400	1.31824	0.004
o-xylene	4497	0.00400	0.45912	0.004

```r
#adjustments that were made according to paper
#William: I'm guessing this refers to Gunnar's paper section 2.2 and Guha 3.3
adjusting_neg_bg_from_lod <- function(chemical, LOD, background, hourly_data){
    # get min and max
    min_value <- min(hourly_data[chemical], na.rm = TRUE)
    max_value <- max(hourly_data[chemical], na.rm = TRUE)
    # if min less than double LOD or max > 100 times LOD
    # adjust to -100 (for entire column???)
    if (min_value < 2 * LOD & max_value > 100 * LOD ){
      return (0)
    }
  return (background)
}
```

```
# Check if background is negligible for non voc
# merge background and LOD
background_lod_non_voc <- tibble(chemical = non_vocs,
                                 LOD = LOD_non_voc,
                                 background = unname(background_levels[non_vocs]))
adjusted_background_non_voc <- background_lod_non_voc %>%
  rowwise() %>%
  mutate(min = min(hourly_data[chemical], na.rm = TRUE),
         LODx2 = 2 * LOD,
         criterion1 = min(hourly_data[chemical], na.rm = TRUE) < 2 * LOD,
         max = max(hourly_data[chemical], na.rm = TRUE),
         LODx100 = 100 * LOD,
         criterion2 = max(hourly_data[chemical], na.rm = TRUE) > 100 * LOD,
         adjusted_background = adjusting_neg_bg_from_lod(chemical, LOD, background, hourly_data))
```

```
# Check if background is negligible for voc
# merge background and LOD
background_lod_voc <- LOD_voc_avg %>%
  left_join(tibble(chemical = setdiff(names(background_levels), non_vocs),
                   background = background_levels[setdiff(names(background_levels), non_vocs)]))
```

```
## Joining with `by = join_by(chemical)`
```

```
adjusted_background_voc <- background_lod_voc %>%
  rowwise() %>%
  mutate(min = min(hourly_data[chemical], na.rm = TRUE),
         LODx2 = 2 * LOD,
         criterion1 = min(hourly_data[chemical], na.rm = TRUE) < 2 * LOD,
         max = max(hourly_data[chemical], na.rm = TRUE),
         LODx100 = 100 * LOD,
         criterion2 = max(hourly_data[chemical], na.rm = TRUE) > 100 * LOD,
         adjusted_background = adjusting_neg_bg_from_lod(chemical, LOD, background, hourly_data))
```

```
# So now we have the adjusted background concentrations
subtract_adj_bg <- function(column, chemical) {
  print(chemical)
  result <-
  return (result)
}
hourly_nona_bgrm <- hourly_full_nona %>%
  mutate(across(adjusted_background_non_voc$chemical, ~  .x - adjusted_background_non_voc$adjusted_backg
hourly_nona_bgrm <- hourly_nona_bgrm %>%
  mutate(across(adjusted_background_voc$chemical, ~  .x - adjusted_background_voc$adjusted_background[ad
```

```
# look at zero values
colSums(hourly_nona_bgrm == 0)
```

```
##         ch4     co2_ppm          co         h2s         so2
##           1           1           1         777        3065
##         nox          o3      ethane      ethene     propane
##           0           0           1           0           1
##       propene 1_3-butadiene      i-butane      n-butane     acetylene
##             0          3126             1             1             0
##  cyclopentane     i-pentane     n-pentane      n-hexane      isoprene
##             0             1             1             0          2815
##     n-heptane       benzene      n-octane       toluene ethyl-benzene
##             0             0             0             0             0
##    m&p-xylene      o-xylene
##             0             0
```

```
# replace negative values with random values between 0 and 0.5*LOD
set.seed(123)
replace_zero_with_random <- function(column, name, LOD_df){
  LOD <- LOD_df$LOD[LOD_df$chemical == name]
  column <- if_else(column == 0, round(runif(length(column), 0, 0.5 * LOD), 3), column)
  return (column)
}

hourly_nona_bgrm_zerorepl <- hourly_nona_bgrm %>%
  mutate(across(adjusted_background_non_voc$chemical, ~ replace_zero_with_random(.x, cur_column(), adjus

hourly_nona_bgrm_zerorepl <- hourly_nona_bgrm_zerorepl %>%
  mutate(across(adjusted_background_voc$chemical, ~ replace_zero_with_random(.x, cur_column(), adjusted_
```

Normalize the non-vocs

```
#normalizing function
normalize_column <- function(column){
  background <- quantile(column, 0)
  max <- quantile(column, 0.99)
  return ((column - background)/(max - background))
}
```

```
# normalize all
hourly_nona_bgrm_zerorepl_norm <- as_tibble(sapply(as.list(hourly_nona_bgrm_zerorepl), normalize_column
summary(hourly_nona_bgrm_zerorepl_norm)
```

```
##       ch4             co2_ppm             co              h2s         
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  
##  1st Qu.:0.04163   1st Qu.:0.2781   1st Qu.:0.1523   1st Qu.:0.05642  
##  Median :0.10679   Median :0.3635   Median :0.2377   Median :0.12909  
##  Mean   :0.19848   Mean   :0.3976   Mean   :0.2953   Mean   :0.19418  
##  3rd Qu.:0.27300   3rd Qu.:0.4808   3rd Qu.:0.3721   3rd Qu.:0.25197  
##  Max.   :7.30686   Max.   :1.9903   Max.   :6.4901   Max.   :5.52116  
##       so2               nox                o3              ethane      
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.04088   1st Qu.:0.03208   1st Qu.:0.1355   1st Qu.:0.0314  
##  Median :0.08054   Median :0.10033   Median :0.3589   Median :0.1007  
##  Mean   :0.13442   Mean   :0.18181   Mean   :0.3732   Mean   :0.1976  
##  3rd Qu.:0.12081   3rd Qu.:0.24866   3rd Qu.:0.5739   3rd Qu.:0.2917  
##  Max.   :5.11178   Max.   :4.92396   Max.   :1.2121   Max.   :3.8434  
##      ethene           propane           propene        1_3-butadiene    
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   : 0.00000  
##  1st Qu.:0.06294   1st Qu.:0.03571   1st Qu.:0.04616   1st Qu.: 0.03843  
##  Median :0.17459   Median :0.11131   Median :0.14857   Median : 0.09608  
##  Mean   :0.24709   Mean   :0.21527   Mean   :0.23032   Mean   : 0.17294  
##  3rd Qu.:0.35373   3rd Qu.:0.32152   3rd Qu.:0.34474   3rd Qu.: 0.17294
##  Max.   :4.83008   Max.   :3.96568   Max.   :7.96071   Max.   :23.05918
##     i-butane          n-butane          acetylene         cyclopentane
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.02802   1st Qu.:0.02682   1st Qu.:0.08853   1st Qu.:0.03143
##  Median :0.08893   Median :0.08557   Median :0.16837   Median :0.09684
##  Mean   :0.18128   Mean   :0.17406   Mean   :0.24305   Mean   :0.18948
##  3rd Qu.:0.25229   3rd Qu.:0.24489   3rd Qu.:0.33257   3rd Qu.:0.26903
##  Max.   :4.67041   Max.   :3.13405   Max.   :3.19783   Max.   :4.31449
##    i-pentane         n-pentane          n-hexane           isoprene
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   : 0.00000
##  1st Qu.:0.02579   1st Qu.:0.02494   1st Qu.:0.02376   1st Qu.: 0.03698
##  Median :0.08349   Median :0.08124   Median :0.08052   Median : 0.07396
##  Mean   :0.17550   Mean   :0.17463   Mean   :0.17907   Mean   : 0.13700
##  3rd Qu.:0.24496   3rd Qu.:0.24418   3rd Qu.:0.25118   3rd Qu.: 0.14793
##  Max.   :4.23427   Max.   :4.45673   Max.   :5.09899   Max.   :13.20266
##    n-heptane          benzene           n-octane           toluene
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.02460   1st Qu.:0.03914   1st Qu.:0.02736   1st Qu.:0.03454
##  Median :0.08276   Median :0.10186   Median :0.08704   Median :0.10588
##  Mean   :0.18168   Mean   :0.18957   Mean   :0.18465   Mean   :0.19932
##  3rd Qu.:0.25481   3rd Qu.:0.26619   3rd Qu.:0.26013   3rd Qu.:0.28677
##  Max.   :4.62641   Max.   :2.48644   Max.   :3.41355   Max.   :2.56851
##  ethyl-benzene       m&p-xylene          o-xylene
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.01923   1st Qu.:0.01598   1st Qu.:0.0000
##  Median :0.09614   Median :0.08902   Median :0.0813
##  Mean   :0.18206   Mean   :0.18234   Mean   :0.1730
##  3rd Qu.:0.26279   3rd Qu.:0.26860   3rd Qu.:0.2571
##  Max.   :2.97077   Max.   :2.37323   Max.   :2.0171
```

Combine and Transpose

```
normalized_matrix <- as.matrix(hourly_nona_bgrm_zerorepl_norm) #important: using the normalized VOCs fo

# Transpose <- cbind(Normalized_Data, Merged_VOCs)
# rownames(Transpose) <- as.character(Transpose[,1]) # I'm not able to run this line, but it shouldn't 
# transpose_normalized_matrix <- t(as.matrix(normalized_matrix))

number_row<- dim(normalized_matrix)[1] #store number of rows (used for checking)
number_column<- dim(normalized_matrix)[2] #store number of columns
```

NMF section

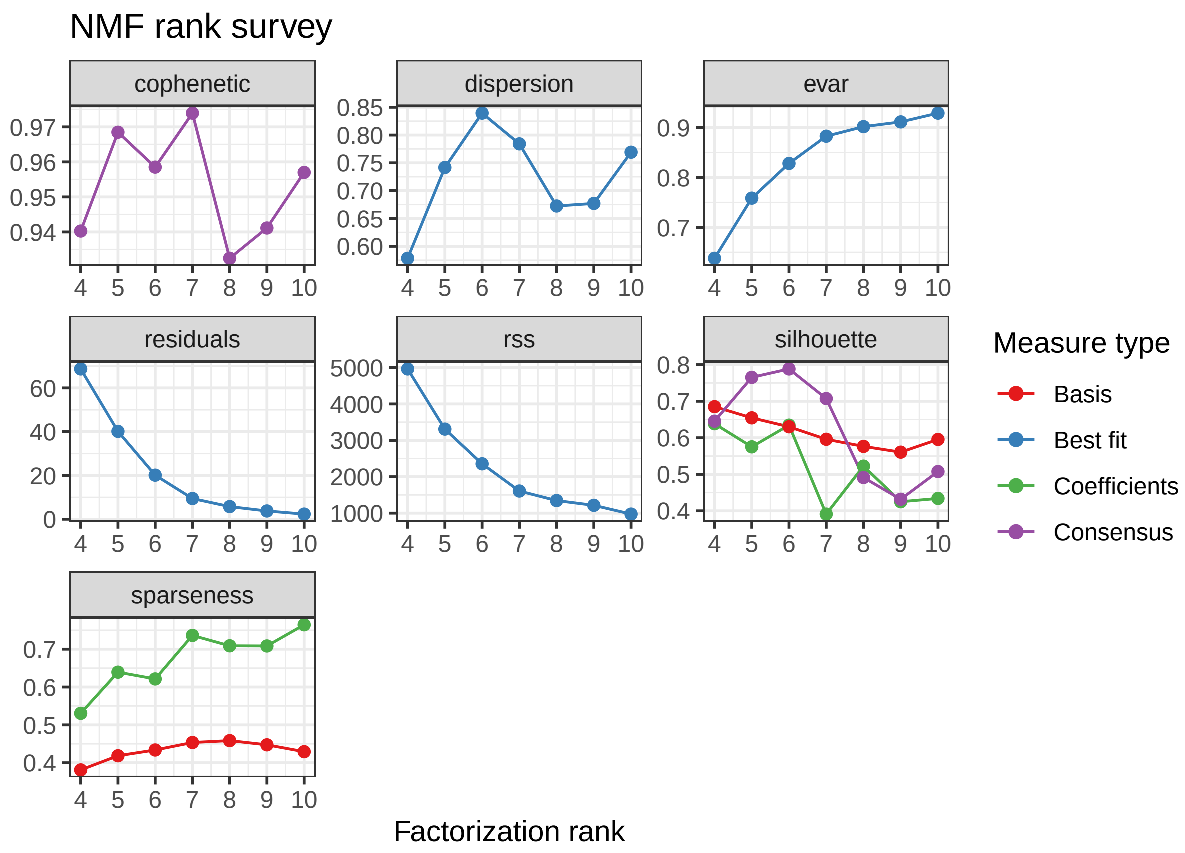
```
# compute weight matrix (uncertainties)
# Based on the Guha paper
# next comment is from the other nmf R file
weight_matrix <- matrix(0, nrow = nrow(normalized_matrix), ncol = ncol(normalized_matrix))
LOD_merged <- tibble(chemical = c(adjusted_background_non_voc$chemical, adjusted_background_voc$chemica
                     LOD = c(adjusted_background_non_voc$LOD, adjusted_background_voc$LOD))

LOD_merged <- tibble(chemical = names(hourly_nona_bgrm_zerorepl_norm)) %>%
  left_join(LOD_merged)
```

```
## Joining with `by = join_by(chemical)`
```

```r
# creating uncertainty Matrix
for (i in 1:number_row) {
  for (j in 1:number_column) {
    xij <- normalized_matrix[i, j]
    LOD <- LOD_merged$LOD[[j]]
    # Get LOD value for this row
    if (j == 1) {
      # based on equation 6, we sqrt ch4 (at column = 1) and times by 1
      weight_matrix[i, j] <- sqrt(xij)
    } else if (j == 2) {
      # 0.25 for co2
      weight_matrix[i, j] <- 0.25 * sqrt(xij)
    } else if (j == 3) {
      # 0.5 for CO
      weight_matrix[i, j] <- 0.5 * sqrt(xij)
    } else if (xij <= LOD) {
      weight_matrix[i, j] <- 2 * LOD # equation 5a) in reference paper
    } else {
      weight_matrix[i, j] <- sqrt(((0.1 * xij)**2 + LOD**2))  #equation 5c) in reference paper
    }
  }
}
```

```r
# set a seed for nmf
# set.seed(123)
# #function below used to estimate the optimal rank and will be used in the nmf() function.
# # takes around 20-30 mins to run
# estimate_rank <- nmfEstimateRank(normalized_matrix, 4:10, method = "ls-nmf", weight = weight_matrix,
# # changing the range of rank to 2:20 from 4:20
# saveRDS(estimate_rank, 'estimate_rank.rds')

estimate_rank <- readRDS('estimate_rank.rds')
measures <- estimate_rank$measures
fit <- estimate_rank$fit
consensus <- estimate_rank$consensus
```

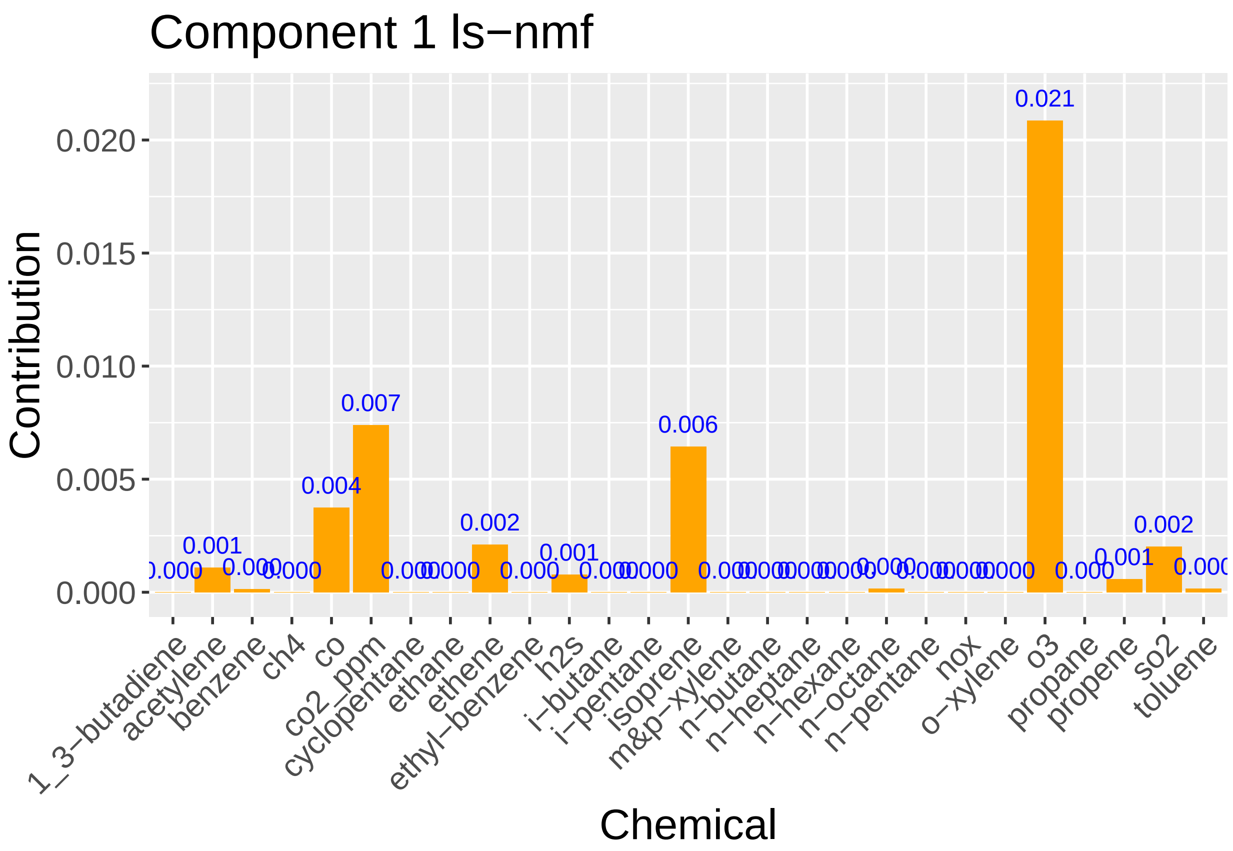
```r
# plots the NMF rank survey
plot(estimate_rank)
```



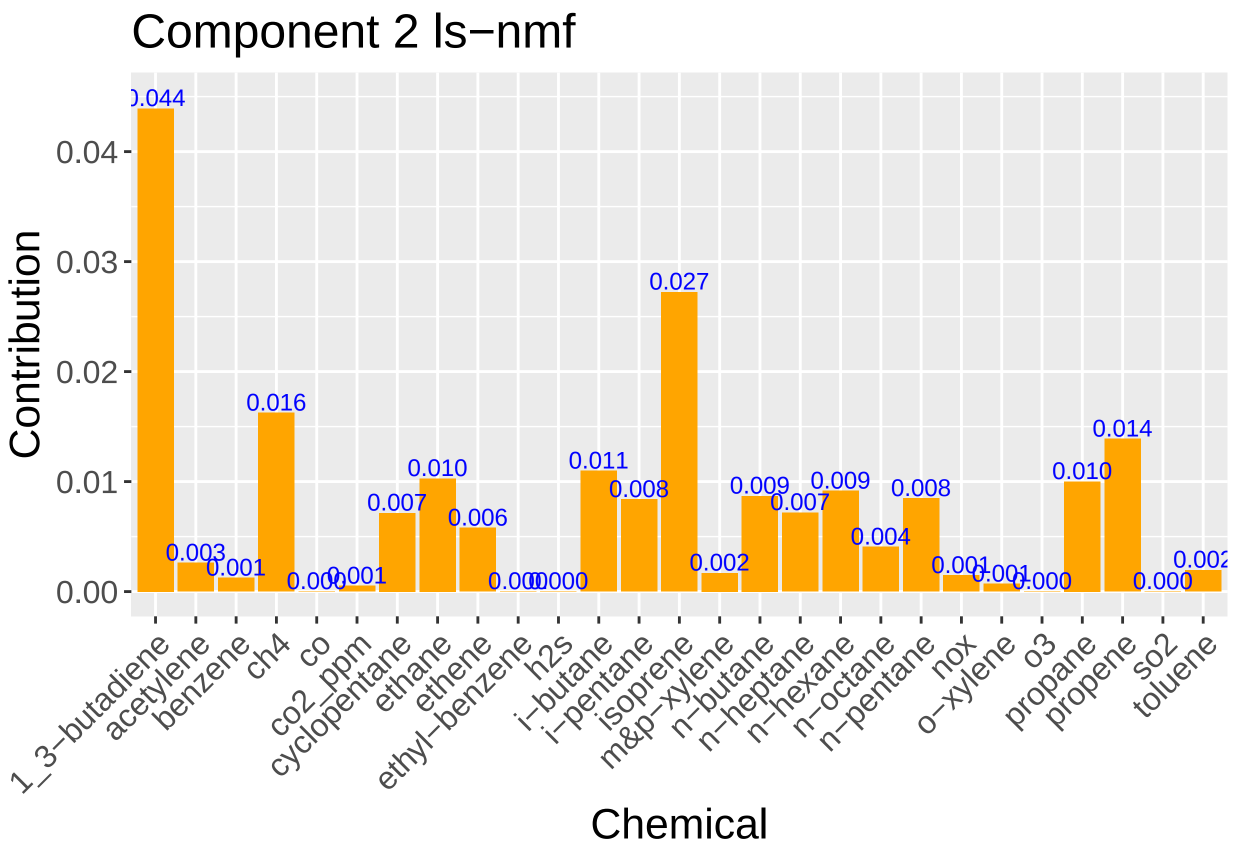
```
# fitting the optimal rank based on the above plots
# the choice of the optimal rank needs to be discussed
output <- nmf(normalized_matrix, rank = 4, weight = weight_matrix, method = "ls-nmf")
W <- basis(output)
H <- coef(output)
```

Source contributions

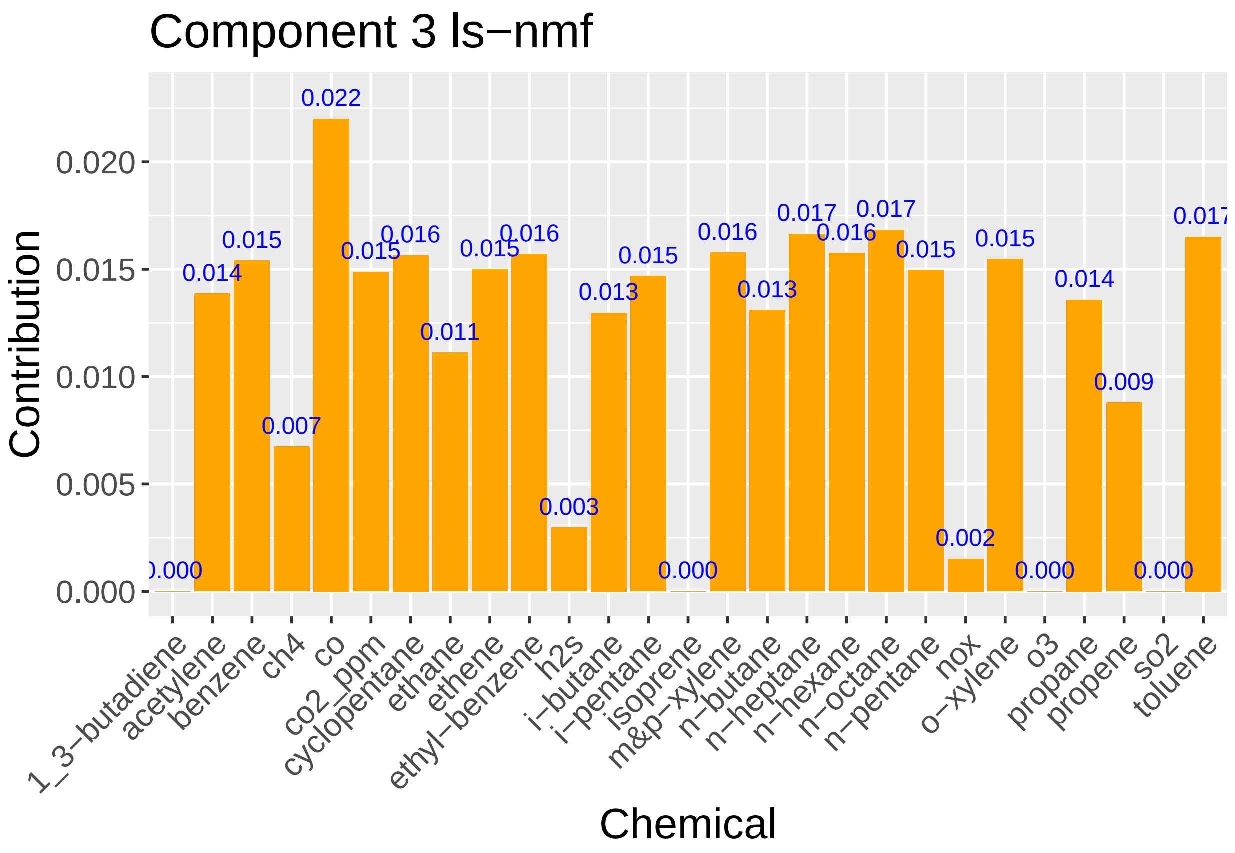
```
# Convert H to a data frame for ggplot
H_df <- as.data.frame(H)
# Add a column for component
H_df$Component <- names(as.data.frame(W))

# Reshape data to long format
H_long <- pivot_longer(H_df, cols = -Component, names_to = "Chemical", values_to = "Contribution")

NFM1 <- subset(H_long, Component == 'V1')
# Plot
nmfplt_1_ls <- ggplot(NFM1, aes(x = Chemical, y = Contribution)) +
  geom_bar(stat = "identity", position = "dodge", fill = "orange") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_text(aes(label = sprintf("%.3f", Contribution)), color = "blue", size = 3, nudge_y = 0.001) +
  labs(x = "Chemical", y = "Contribution", title = "Component 1 ls-nmf")+
  theme(
  text = element_text(size = 14), # Base text size for all text elements
  axis.title = element_text(size = 16), # Size of axis titles
  axis.text = element_text(size = 12), # Size of axis text (tick labels)
  plot.title = element_text(size = 18) # Size of the plot title
)
nmfplt_1_ls
```



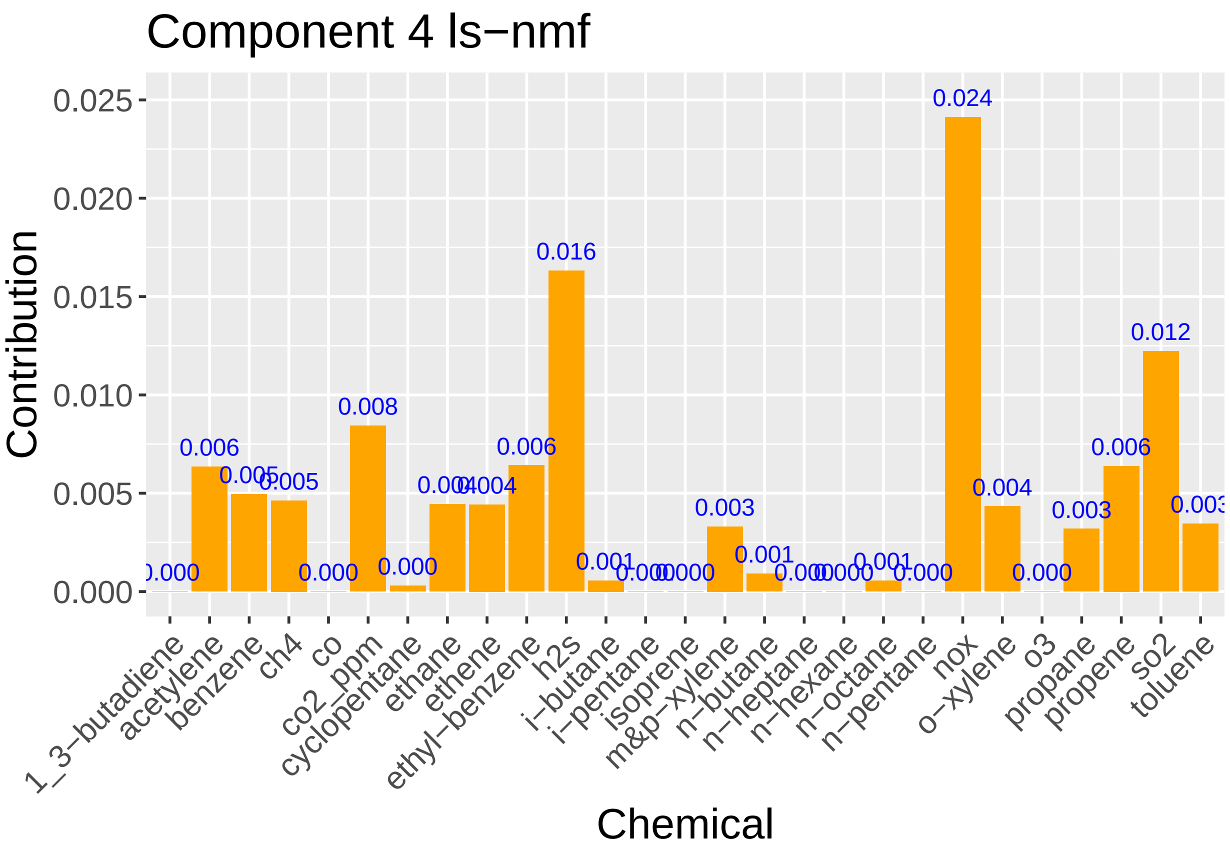
```
NFM2 <- subset(H_long, Component == 'V2')
# Plot
nmfplt_2_ls <- ggplot(NFM2, aes(x = Chemical, y = Contribution)) +
  geom_bar(stat = "identity", position = "dodge", fill = "orange") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_text(aes(label = sprintf("%.3f", Contribution)), color = "blue", size = 3, nudge_y = 0.001) +
  labs(x = "Chemical", y = "Contribution", title = "Component 2 ls-nmf")+
  theme(
  text = element_text(size = 14), # Base text size for all text elements
  axis.title = element_text(size = 16), # Size of axis titles
  axis.text = element_text(size = 12), # Size of axis text (tick labels)
  plot.title = element_text(size = 18) # Size of the plot title
)
nmfplt_2_ls
```

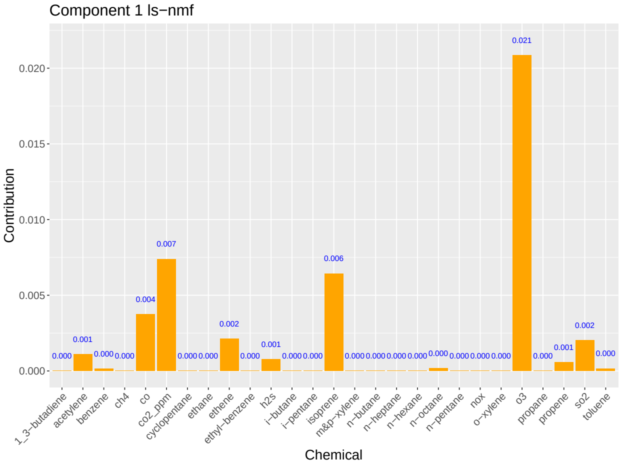


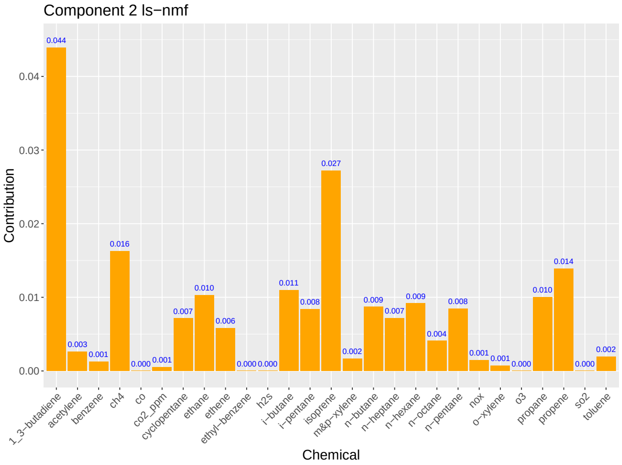
```r
NFM3 <- subset(H_long, Component == 'V3')
# Plot
nmfplt_3_ls <- ggplot(NFM3, aes(x = Chemical, y = Contribution)) +
  geom_bar(stat = "identity", position = "dodge", fill = "orange") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_text(aes(label = sprintf("%.3f", Contribution)), color = "blue", size = 3, nudge_y = 0.001) +
  labs(x = "Chemical", y = "Contribution", title = "Component 3 ls-nmf")+
  theme(
  text = element_text(size = 14), # Base text size for all text elements
  axis.title = element_text(size = 16), # Size of axis titles
  axis.text = element_text(size = 12), # Size of axis text (tick labels)
  plot.title = element_text(size = 18) # Size of the plot title
)
nmfplt_3_ls
```

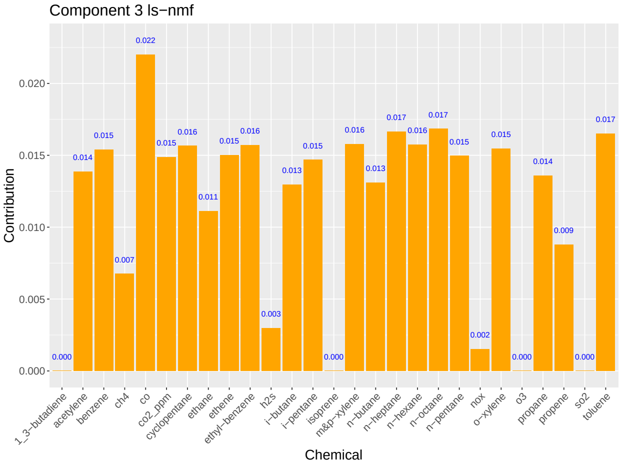


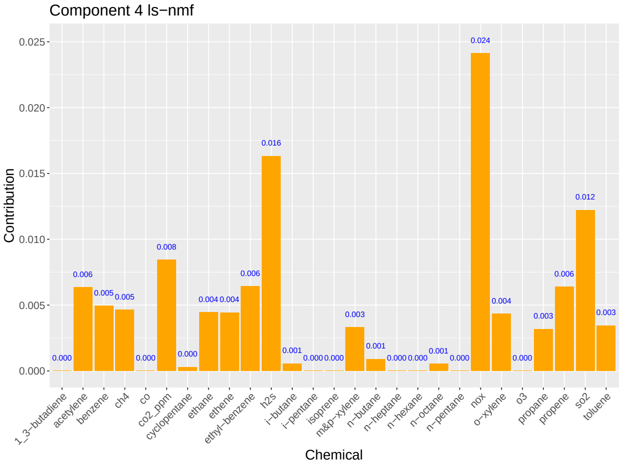
```
NFM4 <- subset(H_long, Component == 'V4')
# Plot
nmfplt_4_ls <- ggplot(NFM4, aes(x = Chemical, y = Contribution)) +
  geom_bar(stat = "identity", position = "dodge", fill = "orange") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_text(aes(label = sprintf("%.3f", Contribution)), color = "blue", size = 3, nudge_y = 0.001) +
  labs(x = "Chemical", y = "Contribution", title = "Component 4 ls-nmf")+
  theme(
  text = element_text(size = 14), # Base text size for all text elements
  axis.title = element_text(size = 16), # Size of axis titles
  axis.text = element_text(size = 12), # Size of axis text (tick labels)
  plot.title = element_text(size = 18) # Size of the plot title
)
nmfplt_4_ls
```


Component 4 ls−nmf
Contribution
0.025
0.020
0.015
0.010
0.005
0.000
0.000
0.006
0.005
0.005
0.000
0.008
0.000
0.004
0.004
0.006
0.016
0.001
0.000
0.000
0.003
0.001
0.000
0.000
0.001
0.000
0.024
0.004
0.000
0.003
0.006
0.012
0.003
1_3−butadiene
acetylene
benzene
ch4
co
co2_ppm
cyclopentane
ethane
ethene
ethyl−benzene
h2s
i−butane
i−pentane
isoprene
m&p−xylene
n−butane
n−heptane
n−hexane
n−octane
n−pentane
nox
o−xylene
o3
propane
propene
so2
toluene
Chemical


Component 1 ls−nmf

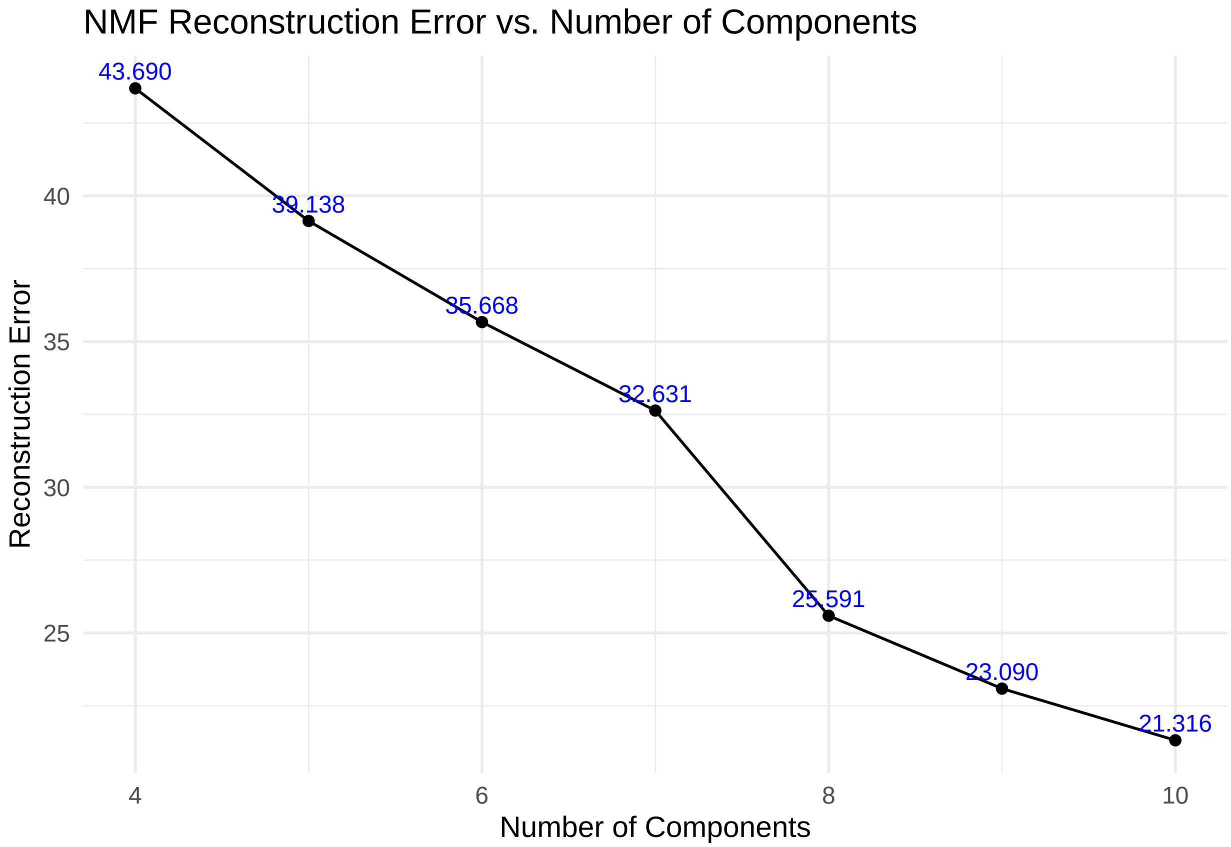

Component 2 ls−nmf


Component 3 ls−nmf

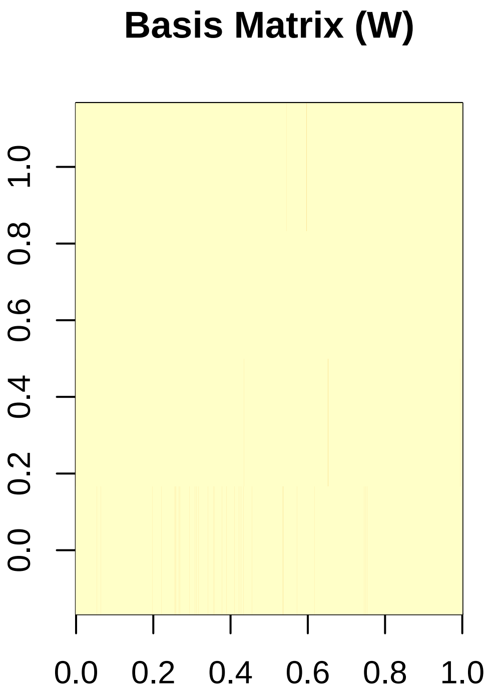

Component 4 ls−nmf

NMF - Eva

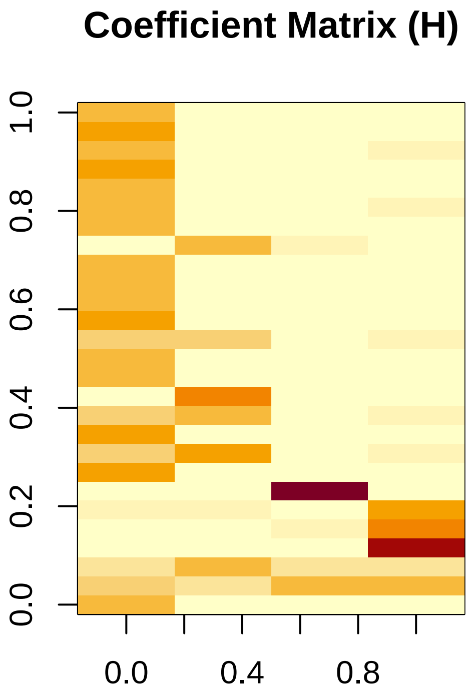
```r
# Try Eva's approach
components <- 4:10
errors <- numeric(length(components) - 4)

# Loop over the number of components
# for (n in components) {
#   nmf_result <- nmf(normalized_matrix, rank = n, method = "KL", seed='nndsvd')
#   reconstruction <- basis(nmf_result) %*% coef(nmf_result)
#   error <- norm(normalized_matrix - reconstruction, type = "F")
#   errors[n-3] <- error
#   print(paste0('Completed ', n - 3, ' out of 7'))
# }
#
# saveRDS(errors, 'errors_norm.rds')

errors <- readRDS('errors_norm.rds')
```



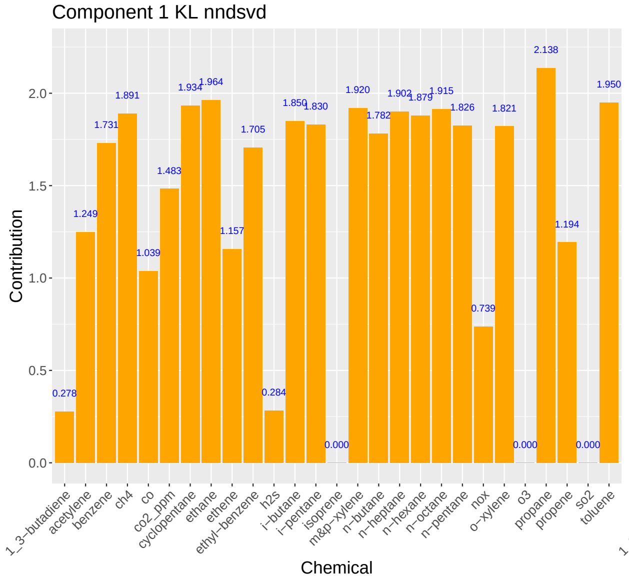
```
## [1] 43.69019

## Warning in sqrt(S[i] * termn) * uun: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termn) * vvn: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termp) * uup: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termp) * vvp: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termn) * uun: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.

## Warning in sqrt(S[i] * termn) * vvn: Recycling array of length 1 in array-vector arithmetic is depre
##   Use c() or as.vector() instead.
```



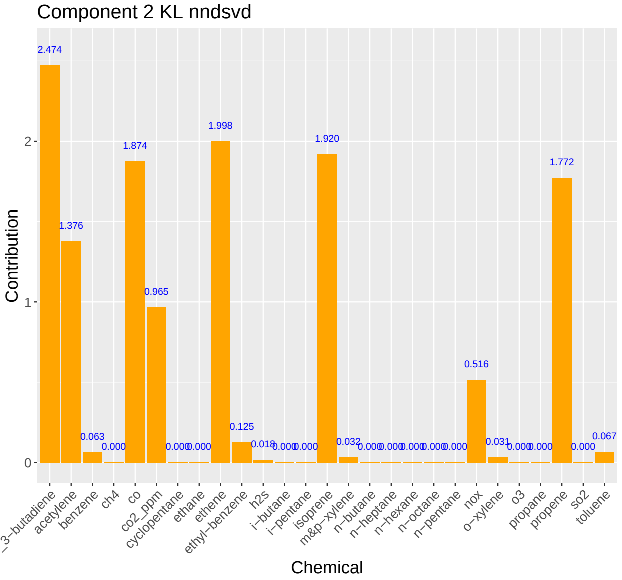
Basis Matrix (W)



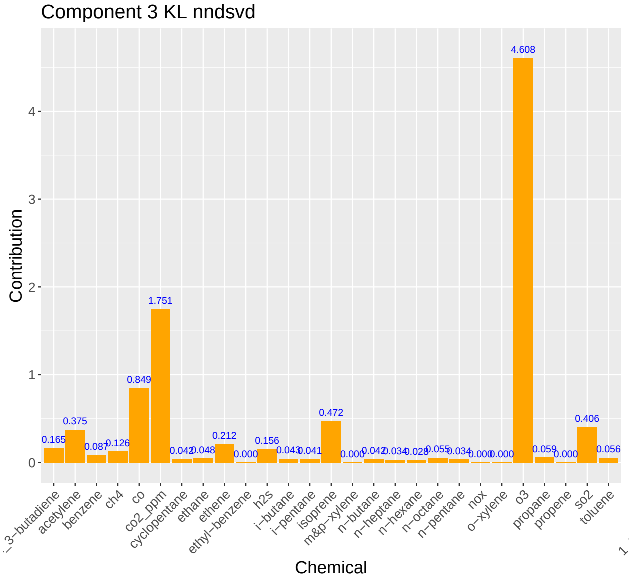
Coefficient Matrix (H)



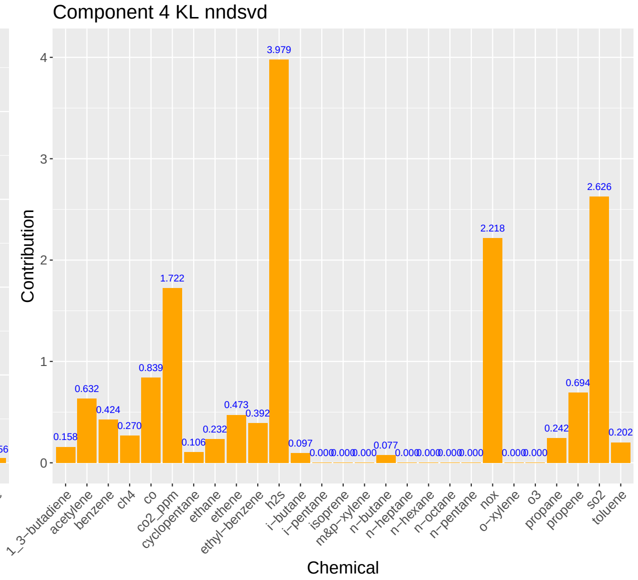
Component 1 KL nndsvd



Component 2 KL nndsvd



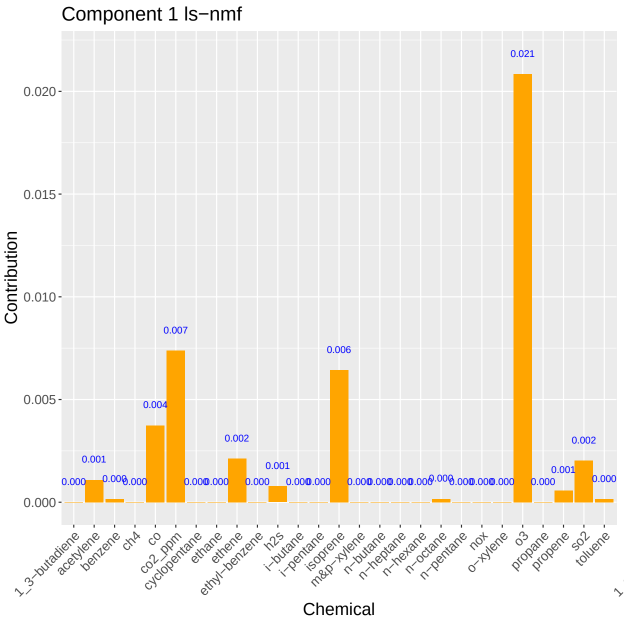
Component 3 KL nndsvd

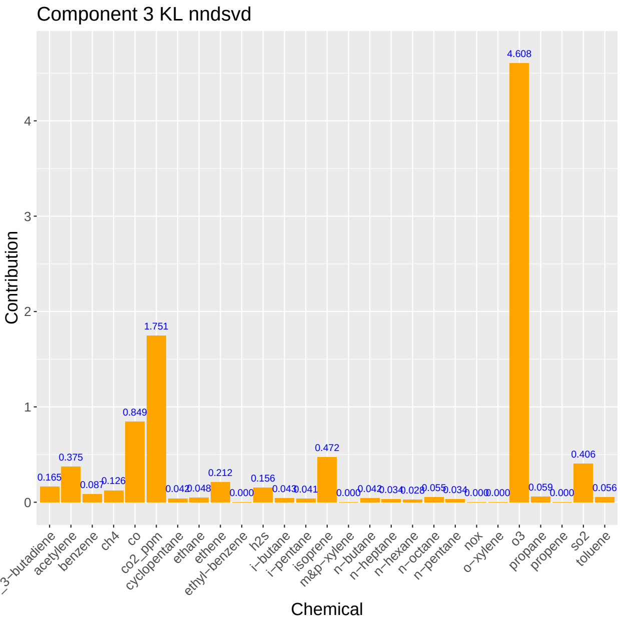


Component 4 KL nndsvd

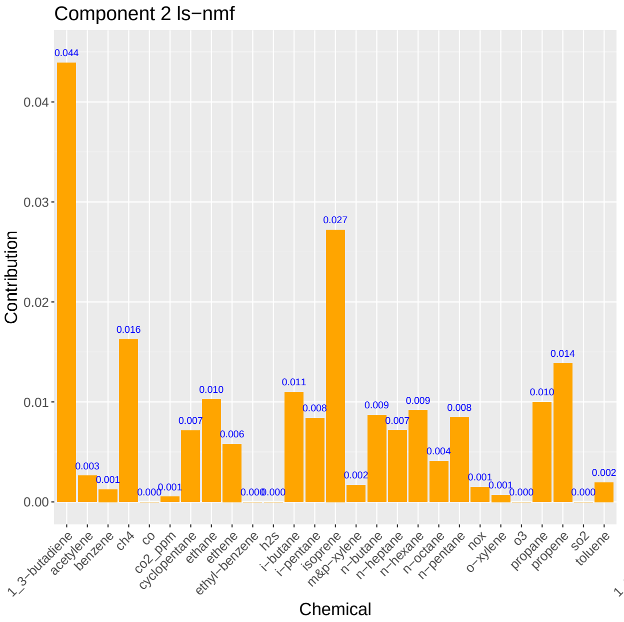
Method comparisons

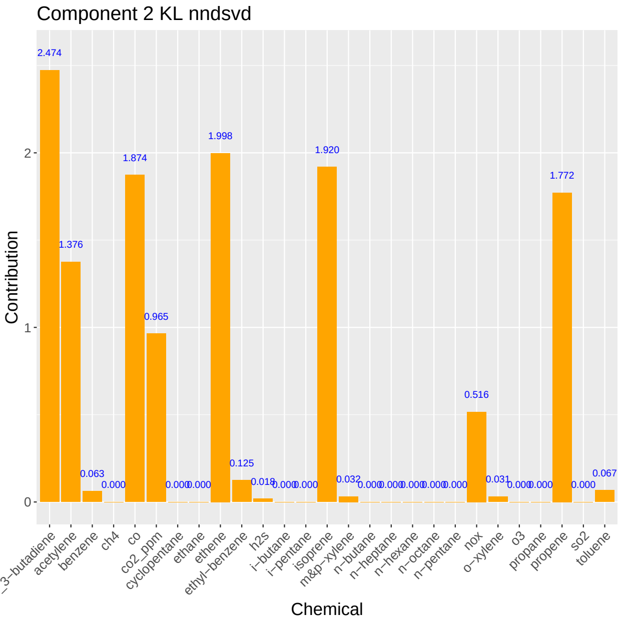
```r
grid.arrange(nmfplt_1_ls, nmfplt_3_svd, ncol = 2)
```



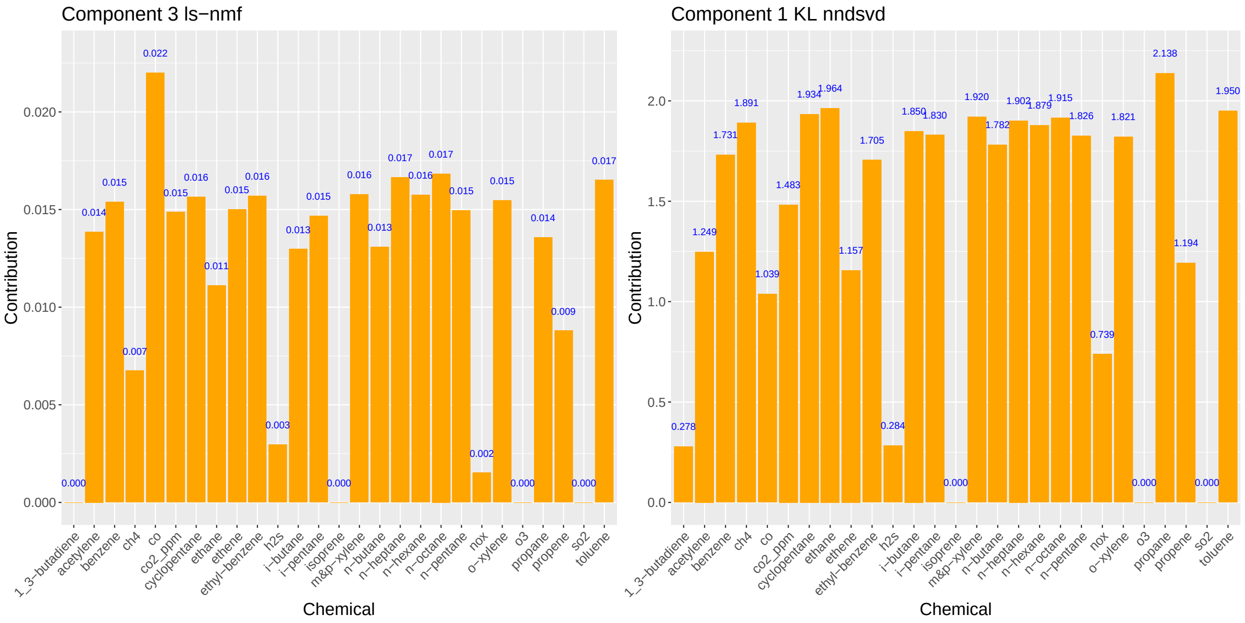


```r
grid.arrange(nmfplt_2_ls, nmfplt_2_svd, ncol = 2)
```

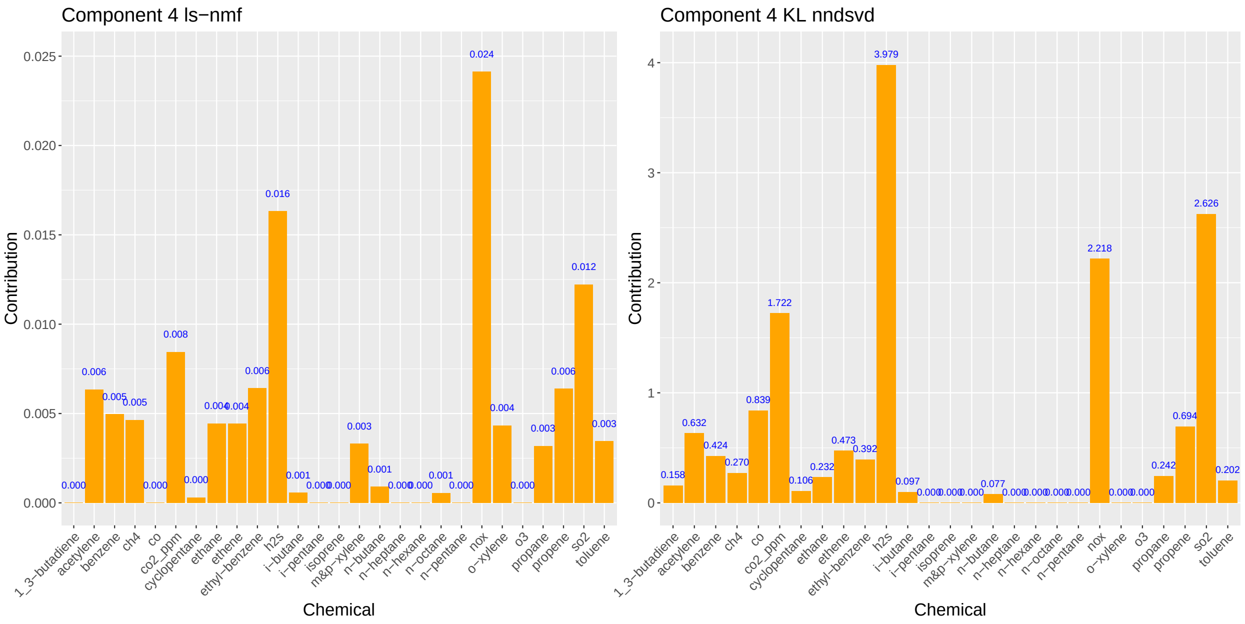




```r
grid.arrange(nmfplt_3_ls, nmfplt_1_svd, ncol = 2)
```



```r
grid.arrange(nmfplt_4_ls, nmfplt_4_svd, ncol = 2)
```


Component 4 ls−nmf
Contribution
Chemical
1_3−butadiene
acetylene
benzene
ch4
co
co2_ppm
cyclopentane
ethane
ethene
ethyl−benzene
h2s
i−butane
i−pentane
isoprene
m&p−xylene
n−butane
n−heptane
n−hexane
n−octane
n−pentane
nox
o−xylene
o3
propane
propene
so2
toluene
Component 4 KL nndsvd
Contribution
Chemical