Validation: Seneca stylometry

Paschalis Agapitos

```
# install.packages("stylo")
```

Introduction

Aim of this notebook is to validate the methods used in the paper *A Stylometric Analysis of Seneca's Disputed Plays: Authorship Verification of Octavia and Hercules Oetaeus*.

A different version of the main anaysis dataset will be used now. The dataset is contained in a folder called `validation_corpus`. It contains 28 texts in verse written by three authors (in total 287138 tokens). The authors used in this corpus are:

- Publius Ovidius Naso (henceforward: **Ovid**)
  - Ars Amatoria
  - Epistulae
  - Fasti
  - Ibis
  - Medicamina Faciei femineae
  - Metamorphoses
  - Ex Ponto
  - Remedia Amoris
  - Tristia
- Aulus Persius Flaccus (henceforward: **Persius**
  - The six books of *Satires*
- Publius Papinius Statius (henceforward: **Statius**)
  - The 12 books of *Thebaid*

To validate the methods we selected one text from each author (i.e., in total three texts) and we renamed them with the following format: `unknown{n}.txt`. The authors to validate the methods are the following ones:

- *Amores* by Ovid (i.e., `unknown0.txt`)
- *Thebaid* book 1 by Statius (i.e., `unknown1.txt`)
- *Satire* 4 by Persius (i.e., `unknown2.txt`)

The first two texts were randomly chosen to be tested. However, the last one is the trickiest one because it consists of only 342 tokens.

In this notebook we will apply Principal Component Analysis (henceforward: PCA) and Bootstrap Consensus Tree (henceforward: BCT); for the former we will use a covariance and a correlation matrix to visualise the results. The same preprocessing step will be applied to every text that is used here and the results will be generated using Most Frequent Characters (henceforward MFCs) tetragrams and pentagrams; these number of n-grams will be applied to each one of the aforementioned methods and their variations.

```
library(stylo)
```

```
##
## ### stylo version: 0.7.4 ###
```

```
##
## If you plan to cite this software (please do!), use the following reference:
##     Eder, M., Rybicki, J. and Kestemont, M. (2016). Stylometry with R:
##     a package for computational text analysis. R Journal 8(1): 107-121.
##     <https://journal.r-project.org/archive/2016/RJ-2016-007/index.html>
##
## To get full BibTeX entry, type: citation("stylo")
```

```
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(pheatmap)
```

Setting the working directory

Our working directory is set to `validation_PCA_BCT`. This directory holds the data and the code to validate the PCA method and the BCT method.

```
setwd("../validation_PCA_BCT/")
getwd()
```

```
## [1] "/Users/paschalis/Documents/MA_DH/Thesis/seneca_stylometry/analysis/validation/validation_PCA_BC
```

Importing the corpus and tokenization

In this step we import the corpus that we are going to use and consequently we tokenize it. The tokenization follows the rules of the parameter `Latin.corr`. This is done because a lot of texts do not distinguish "u/v" and by setting this parameter to `Latin.corr` we take care of this variation in the letters. Furthermore, we know that since we use texts from the Perseus Digital Library that the distinction between "u/v" should be addressed. Lastly, we change uppercase letters to lowercase because we need to further minimize the variations between words.

```
raw.corpus <- load.corpus(files = "all", corpus.dir = "validation_corpus/",
                          encoding = "UTF-8")

tokenized.corpus <- txt.to.words.ext(raw.corpus, corpus.lang = "Latin.corr",
                                     preserve.case = F)

summary(tokenized.corpus)
```

```
##
## Function call:
## NULL
##
## Number of texts/samples:         28
## Total number of units:           287138
## Number of units in samples:
##
##                 ovid_ars.txt...  14901
##               ovid_epist.txt...  25762
##               ovid_fasti.txt...  31273
##                ovid_ibis.txt...  4032
##             ovid_medicam.txt...  613
##                ovid_meta.txt...  78098
##               ovid_ponto.txt...  21505
##               ovid_remed.txt...  5256
##             ovid_tristia.txt...  22975
##           persius_sati_1.txt...  945
##                             ...  ...
##
## Depending if the corpus has been tokenized or not, the "units" mean
## tokens (words, word pairs, POS-tags, character n-grams, etc.),
## or strings of text (usually paragraphs) ending with a newline char.
```

Remove the pronouns

It was decided to remove the pronouns, since some pronouns are connected to the genre of the text.

```
corpus.no.pronouns <- delete.stop.words(tokenized.corpus,
                                        stop.words = stylo.pronouns(corpus.lang = "Latin"))
# the list with the pronouns removed
stylo.pronouns(corpus.lang = "Latin")
```

```
##  [1] "ea"         "eae"        "eam"        "earum"      "eas"        "ego"
##  [7] "ei"         "eis"        "eius"       "eo"         "eorum"      "eos"
## [13] "eum"        "id"         "illa"       "illae"      "illam"      "illarum"
## [19] "illas"      "ille"       "illi"       "illis"      "illius"     "illo"
## [25] "illorum"    "illos"      "illud"      "illum"      "is"         "me"
## [31] "mea"        "meae"       "meam"       "mearum"     "meas"       "mei"
## [37] "meis"       "meo"        "meos"       "meorum"     "meum"       "meus"
## [43] "mihi"       "nobis"      "nos"        "noster"     "nostra"     "nostrae"
## [49] "nostram"    "nostrarum"  "nostras"    "nostri"     "nostris"    "nostro"
## [55] "nostros"    "nostrorum"  "nostrum"    "sua"        "suae"       "suam"
## [61] "suarum"     "suas"       "sui"        "suis"       "suo"        "suos"
## [67] "suorum"     "suum"       "suus"       "te"         "tibi"       "tu"
## [73] "tua"        "tuae"       "tuam"       "tuarum"     "tuas"       "tui"
## [79] "tuis"       "tuo"        "tuos"       "tuorum"     "tuum"       "tuus"
## [85] "vester"     "vestra"     "vestrae"    "vestram"    "vestrarum"  "vestras"
## [91] "vestri"     "vestris"    "vestro"     "vestros"    "vestrorum"  "vestrum"
## [97] "vobis"      "vos"
```

Character 4-grams

Extracting the features (character 4-grams)

The final step before proceeding to the application of the methods to the corpus is to extract the features that we want to use and add them to a table with frequencies. In our case, we want to extract character 4-grams.

```
corpus.char.4.grams <- txt.to.features(corpus.no.pronouns,
                                       features = "c",
                                       ngram.size = 4) # break the text into character 4-grams

frequent.features.4grams <- make.frequency.list(corpus.char.4.grams,
                                                head = 2000)
```

```
freqs.4grams <- make.table.of.frequencies(corpus.char.4.grams,
                                          features = frequent.features.4grams,
                                          relative = T) # relative=True to compute the relative frequen
```

```
## processing  28  text samples
```

```
## ..
## combining frequencies into a table...
```

Methods - Character 4-grams

Apply Principal Component Analysis

Principal Component Analysis - Correlation matrix (MFCs 4grams)

In this experiment we will apply Principal Component Analysis (henceforward: PCA) using a correlation matrix to visualise the results. The features used in this experiment are Most Frequent Characters (henceforward: MFCs) 4-grams. We will look at the top 100 to 1500 MFCs 4-grams with an increment of 100 in each iteration (no culling will be specified because we want to obtain a sufficient number of features in each iteration (given this corpus, if we set culling to 100% we obtain only 33 MFC)).

Eder's Delta will be used as a distance metric.

```
# PCA correlation - top 100-1500-100 incr.100 MFCs 4-grams
results_pca_4grams_cor = stylo(frequencies = freqs.4grams,
                               analysis.type = "PCR",
                               mfw.min = 100, mfw.max = 1500, increment=100,
                               distance.measure = "eder", # Eder's Delta
                               custom.graph.title = "Who is the author?", # title of the plot
                               pca.visual.flavour="classic", # flavour of the PCA plot
                               write.png.file=T, gui = T) # gui = True to double-check the parameters
```

```
## using current directory...
```

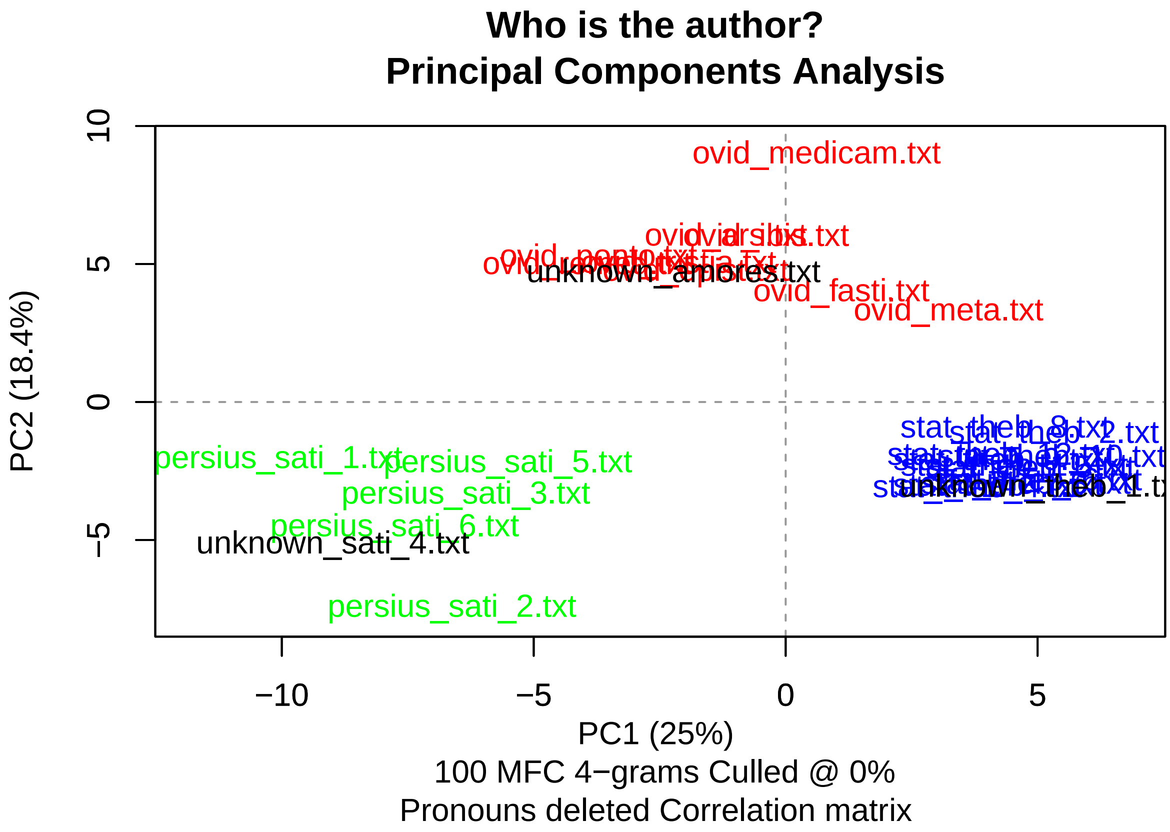
```
## Warning in delete.stop.words(table.with.all.freqs, pronouns): chosen stop words were not found in the
##   please check the language, lower/uppercase issues, etc.
```

```
##
```

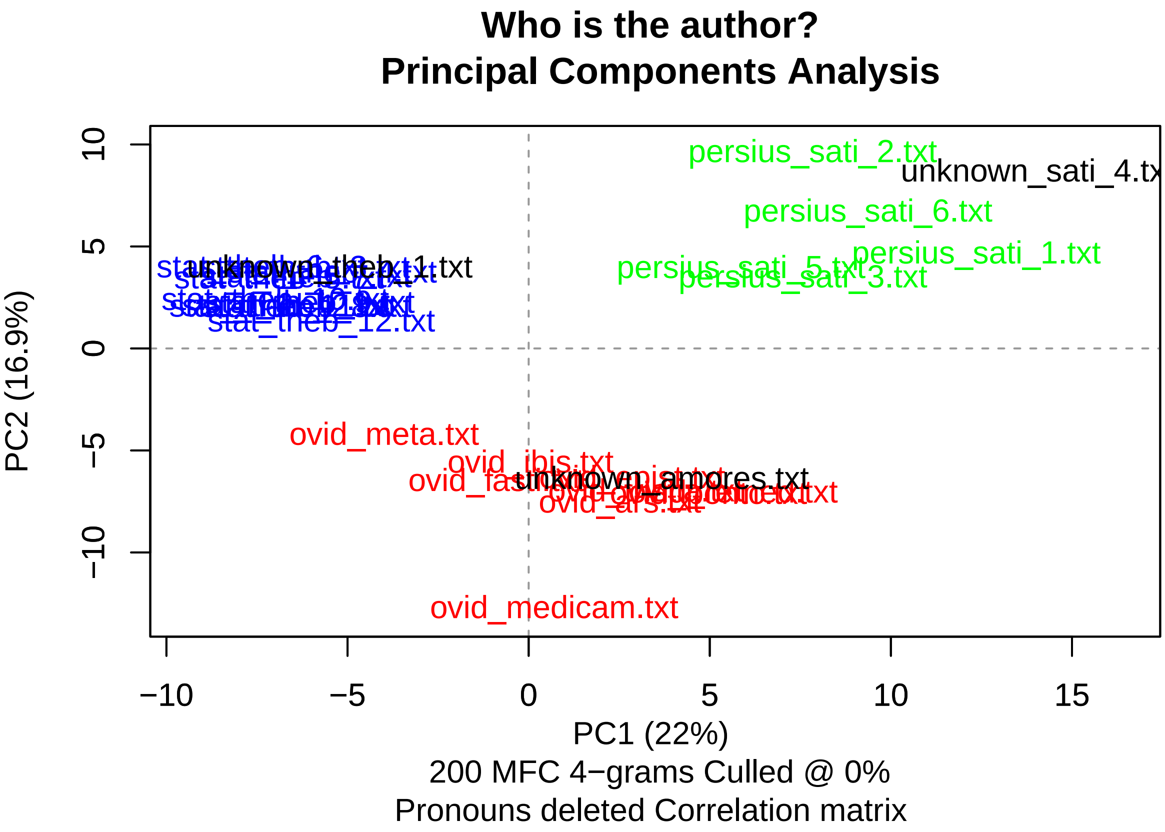
```
## culling @ 0  available features (words) 2000
```

```
## MFW used:
```

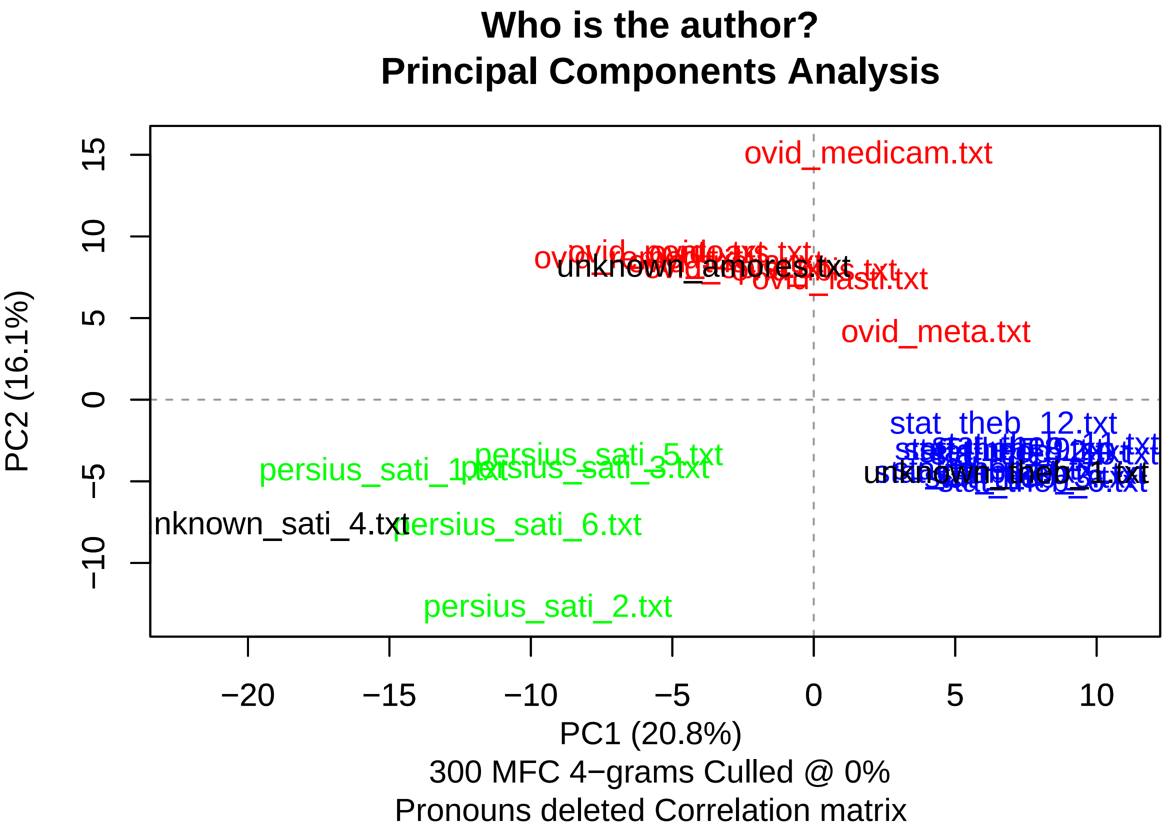
```
## 100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



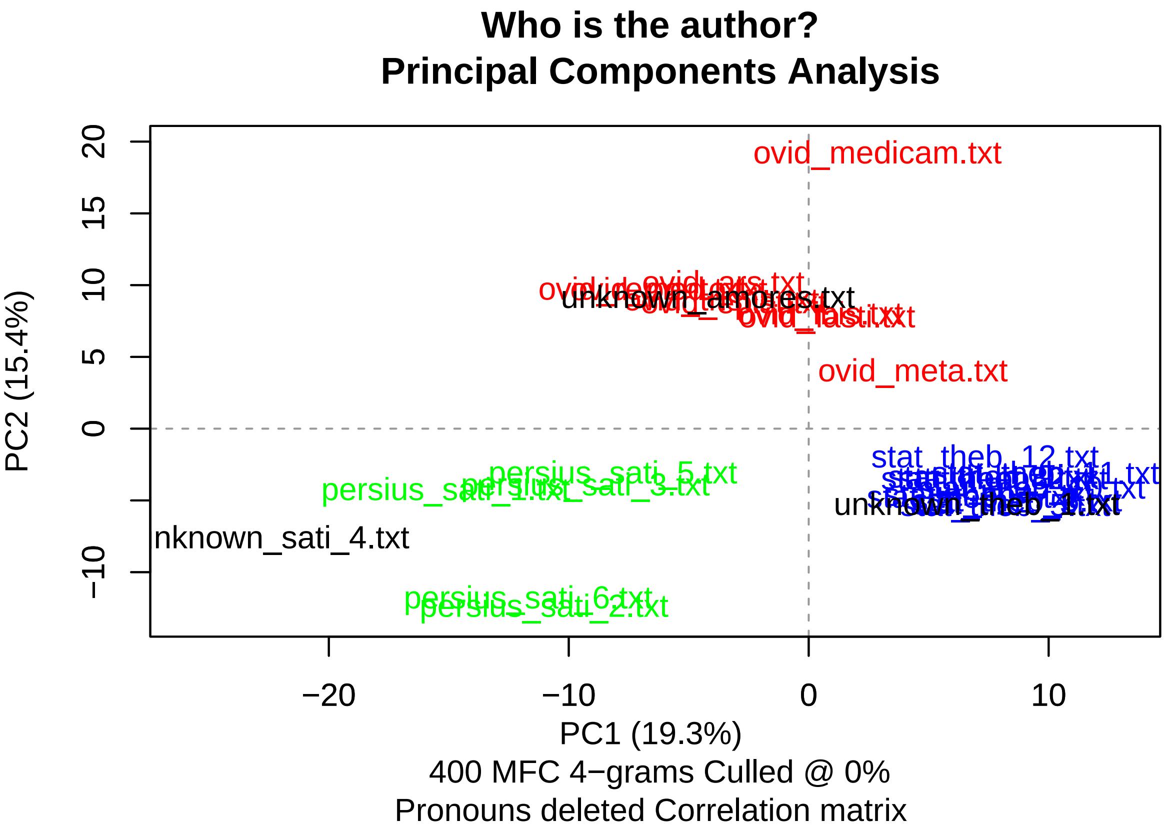
```
## 300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



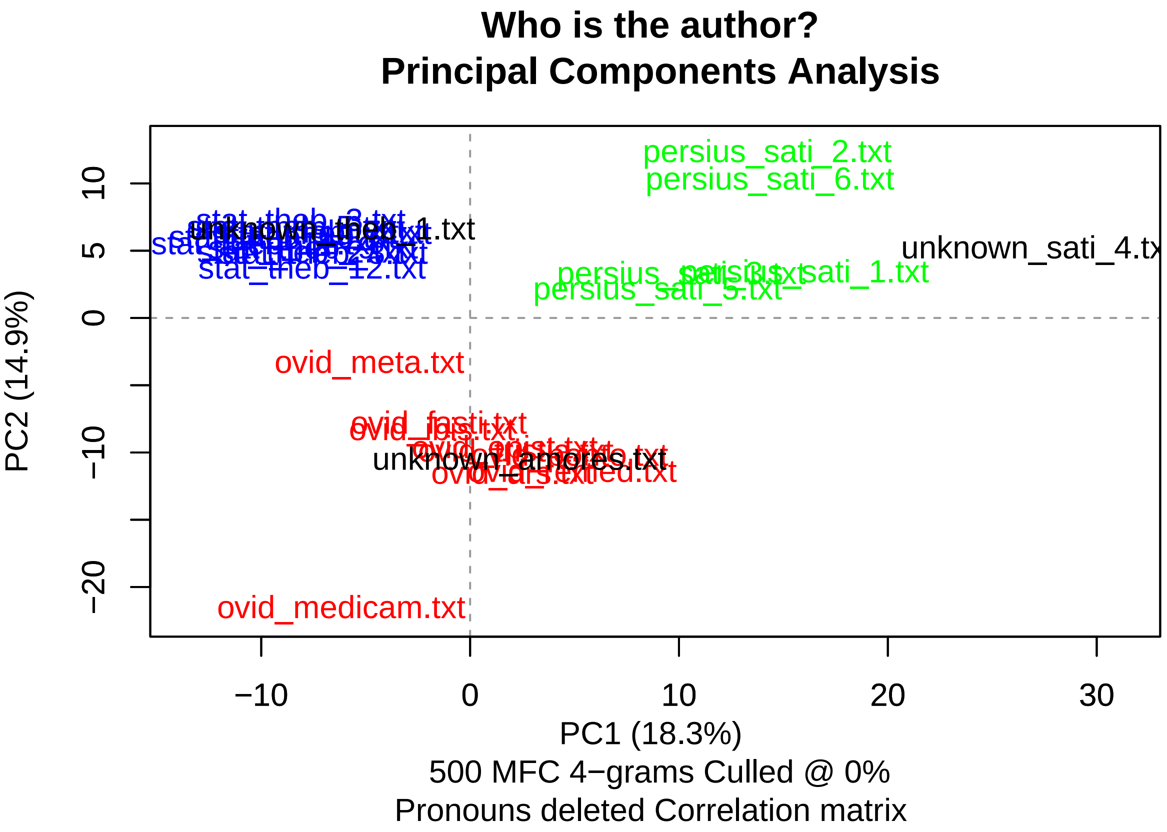
```
## 400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



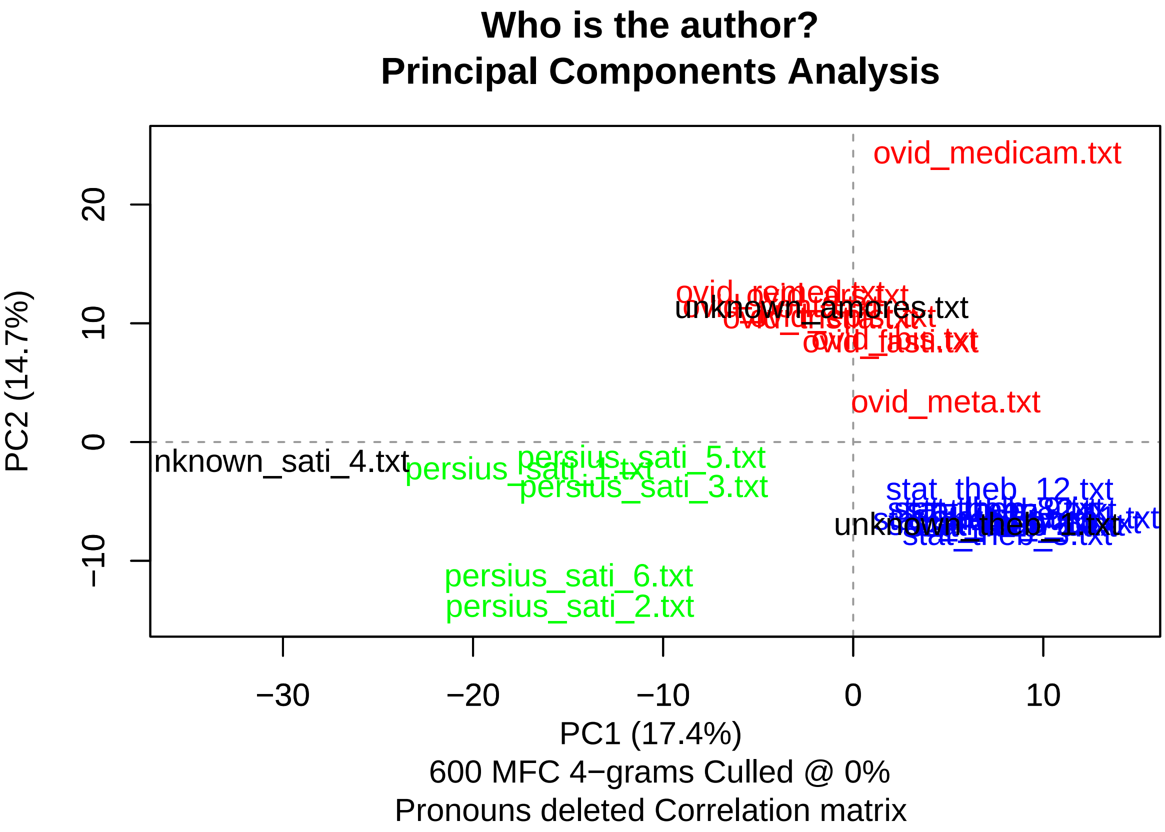
```
## 500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



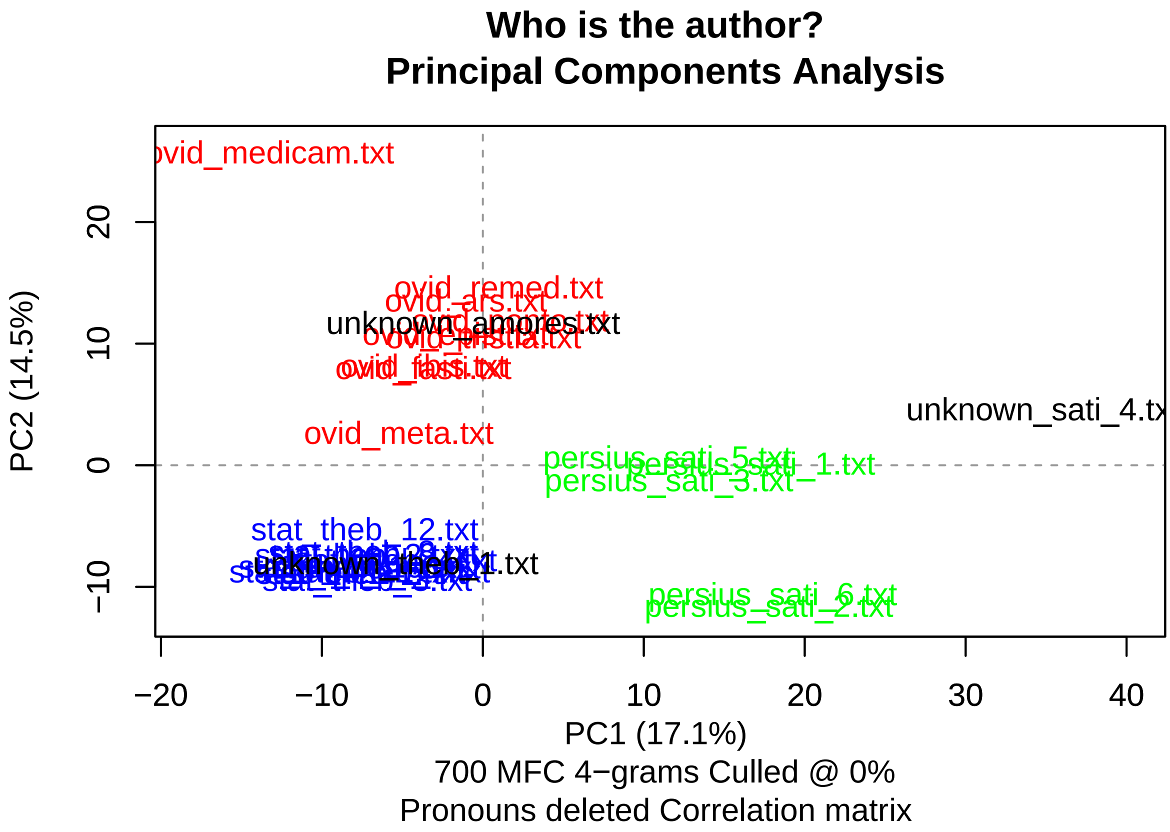
```
## 600
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



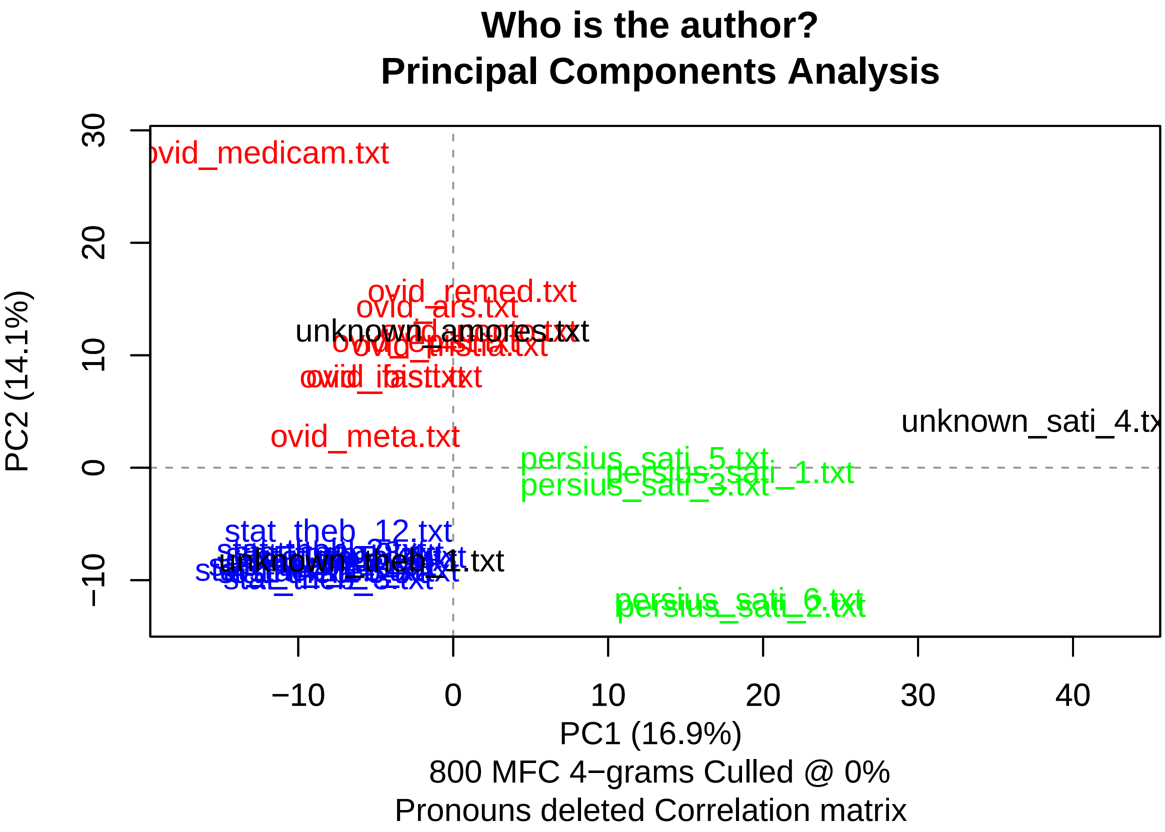
```
## 700
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



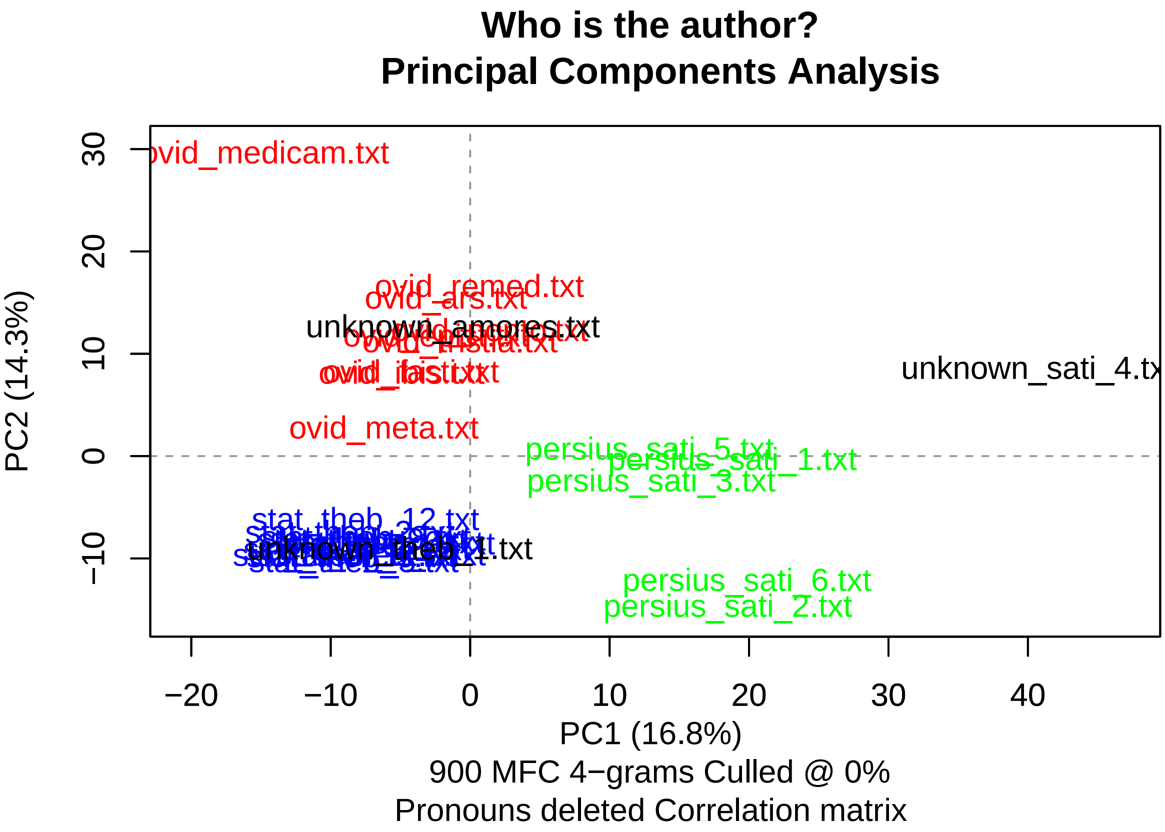
```
## 800
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



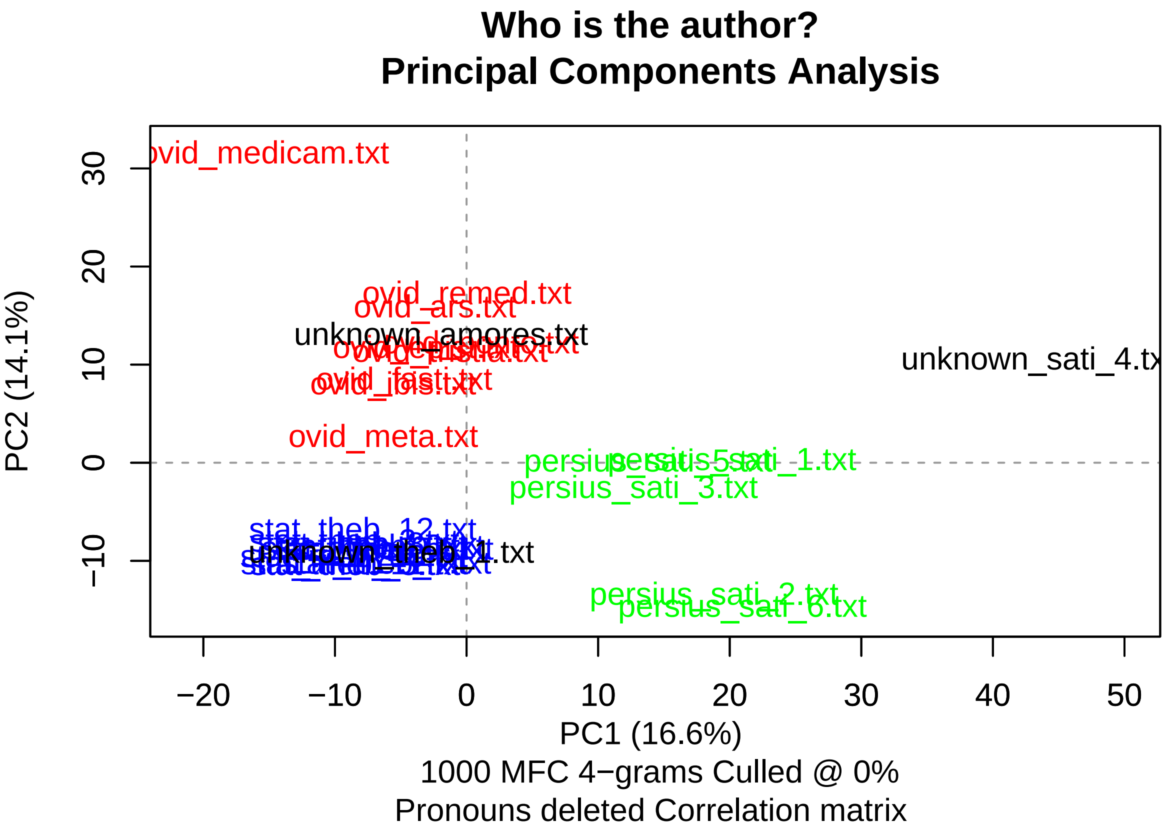
```
## 900
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



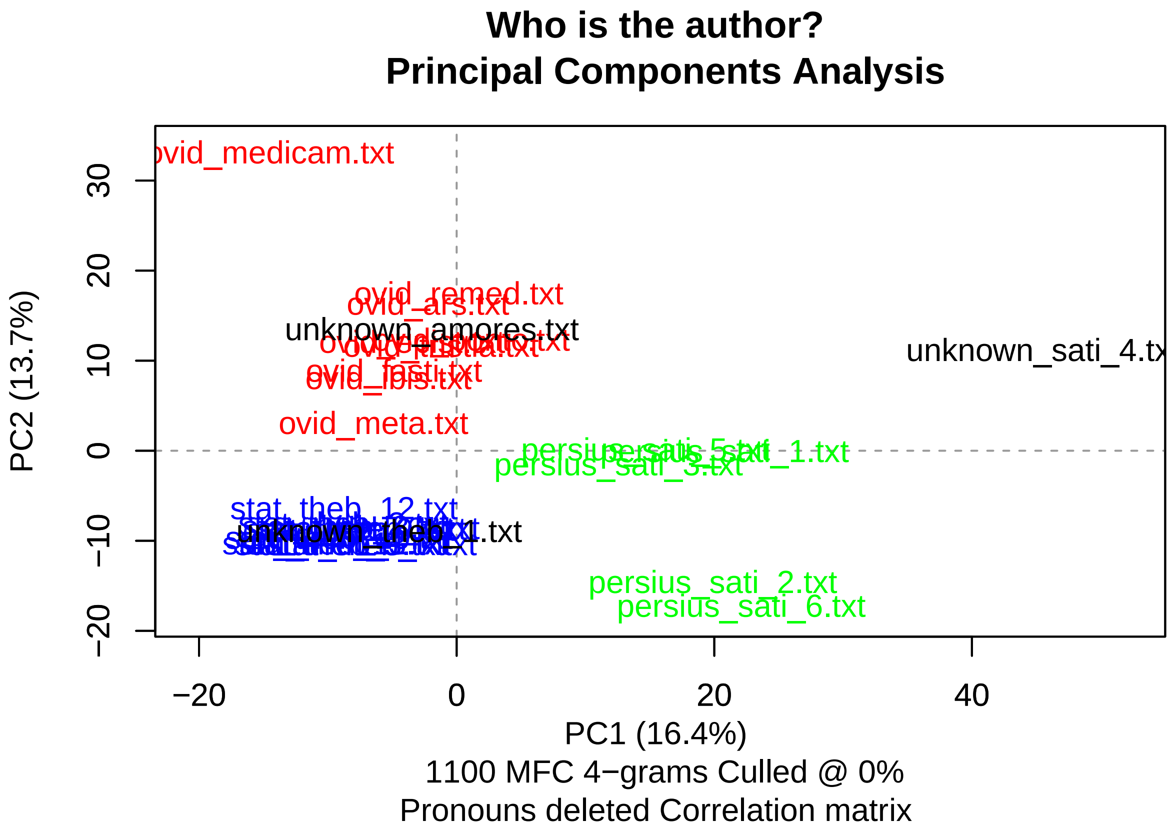
```
## 1000
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



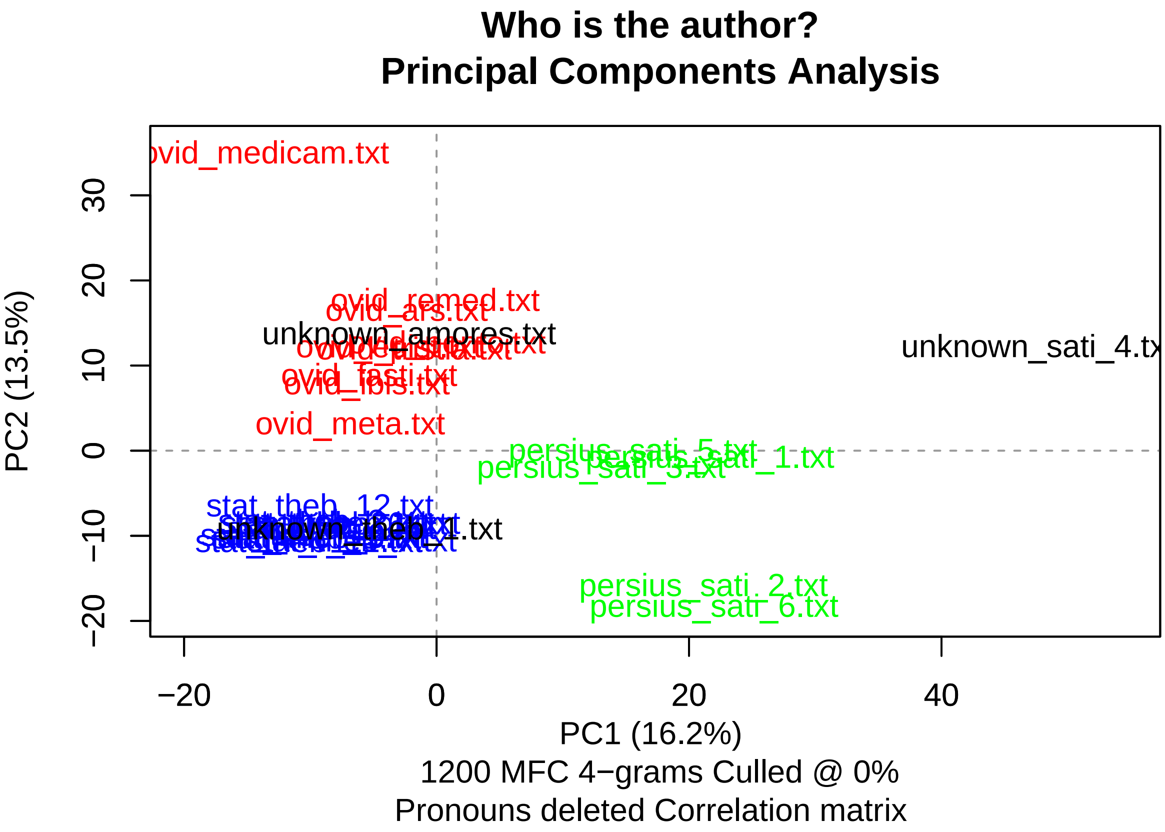
```
## 1100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



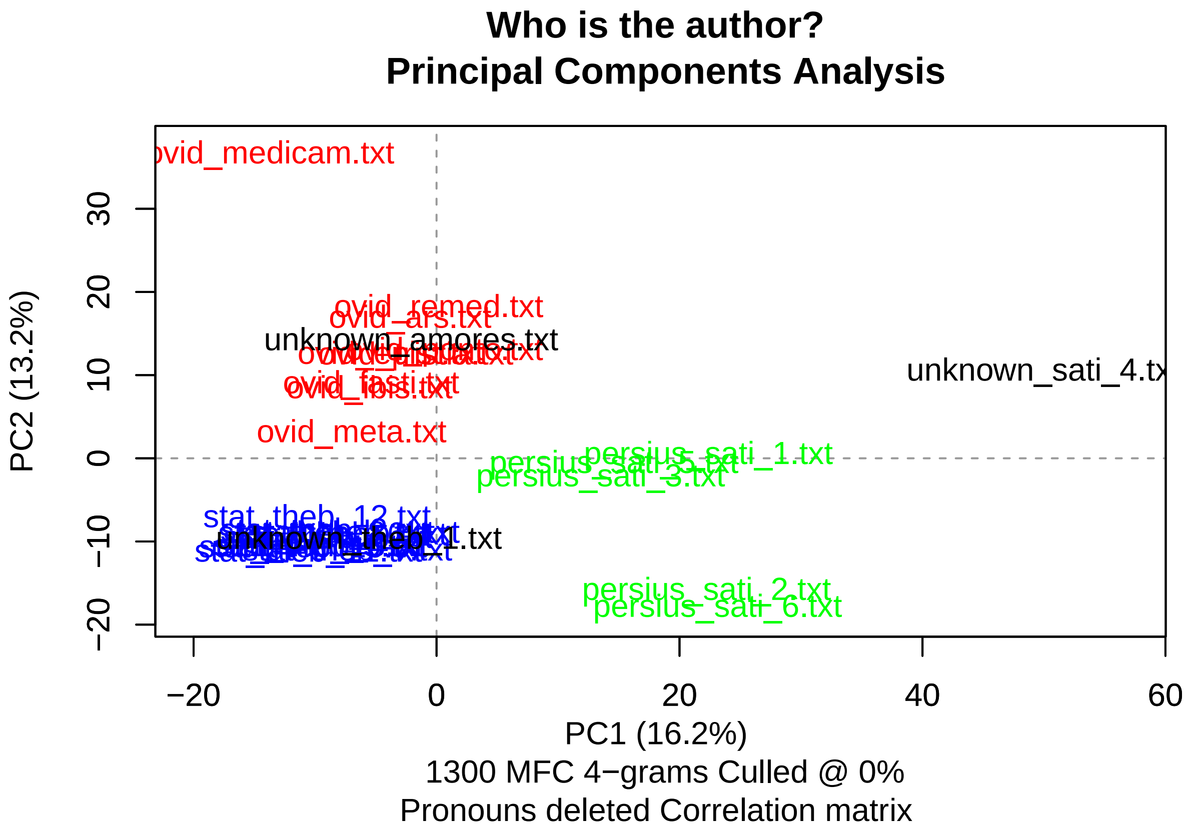
```
## 1200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



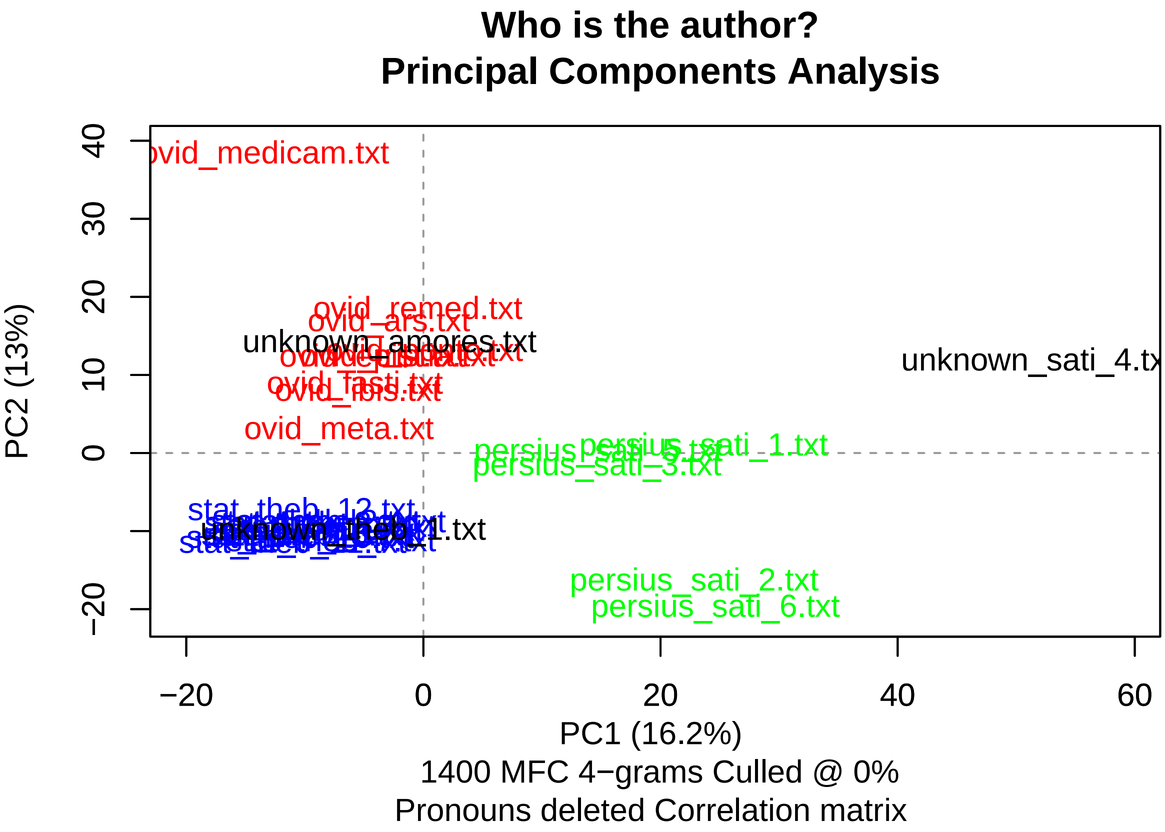
```
## 1300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



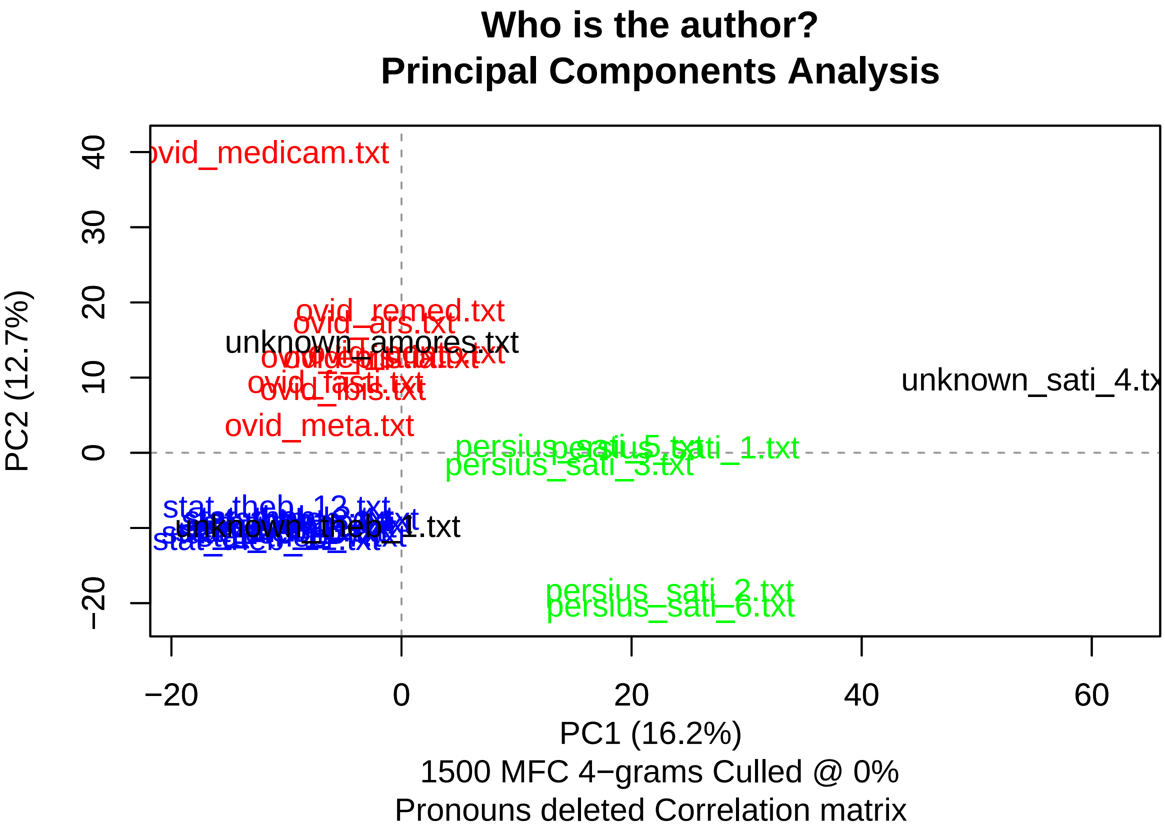
```
## 1400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



```
## 1500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```


Who is the author?
Principal Components Analysis
PC2 (13%)
40
30
20
10
0
−20
−20
0
20
40
60
vid_medicam.txt
ovid_remed.txt
ovid_ars.txt
unknown_amores.txt
ovid_fasti.txt
ovid_ibis.txt
ovid_meta.txt
unknown_sati_4.tx
persius_sati_1.txt
persius_sati_5.txt
persius_sati_3.txt
stat_theb_12.txt
unknown_theb
persius_sati_2.txt
persius_sati_6.txt
PC1 (16.2%)
1400 MFC 4−grams Culled @ 0%
Pronouns deleted Correlation matrix

##

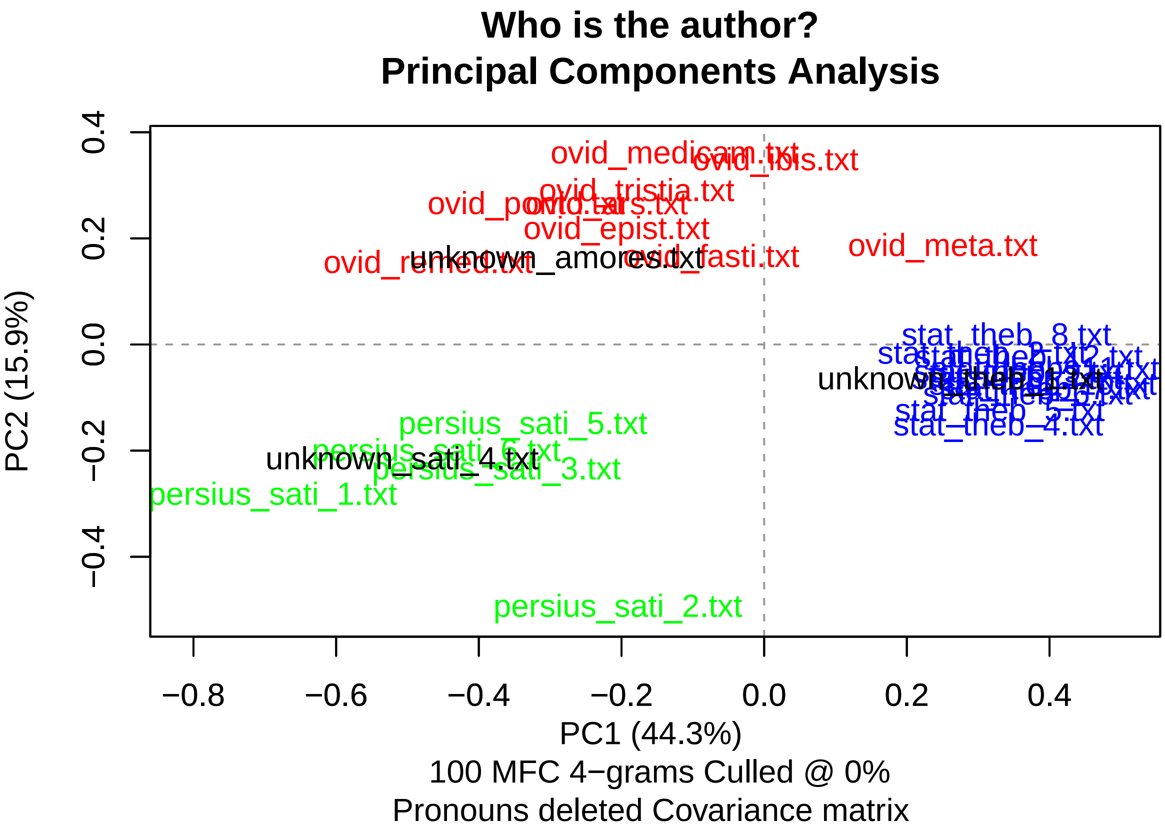

Who is the author?
Principal Components Analysis
PC2 (12.7%)
40
30
20
10
0
−20
−20
0
20
40
60
vid_medicam.txt
ovid_remed.txt
ovid_ars.txt
unknown_amores.txt
ovid_fasti.txt
ovid_ibis.txt
ovid_meta.txt
unknown_sati_4.tx
persius_sati_1.txt
persius_sati_5.txt
persius_sati_3.txt
stat_theb_12.txt
unknown_theb
persius_sati_2.txt
persius_sati_6.txt
PC1 (16.2%)
1500 MFC 4−grams Culled @ 0%
Pronouns deleted Correlation matrix

Principal Component Analysis - Covariance matrix (MFCs 4grams)

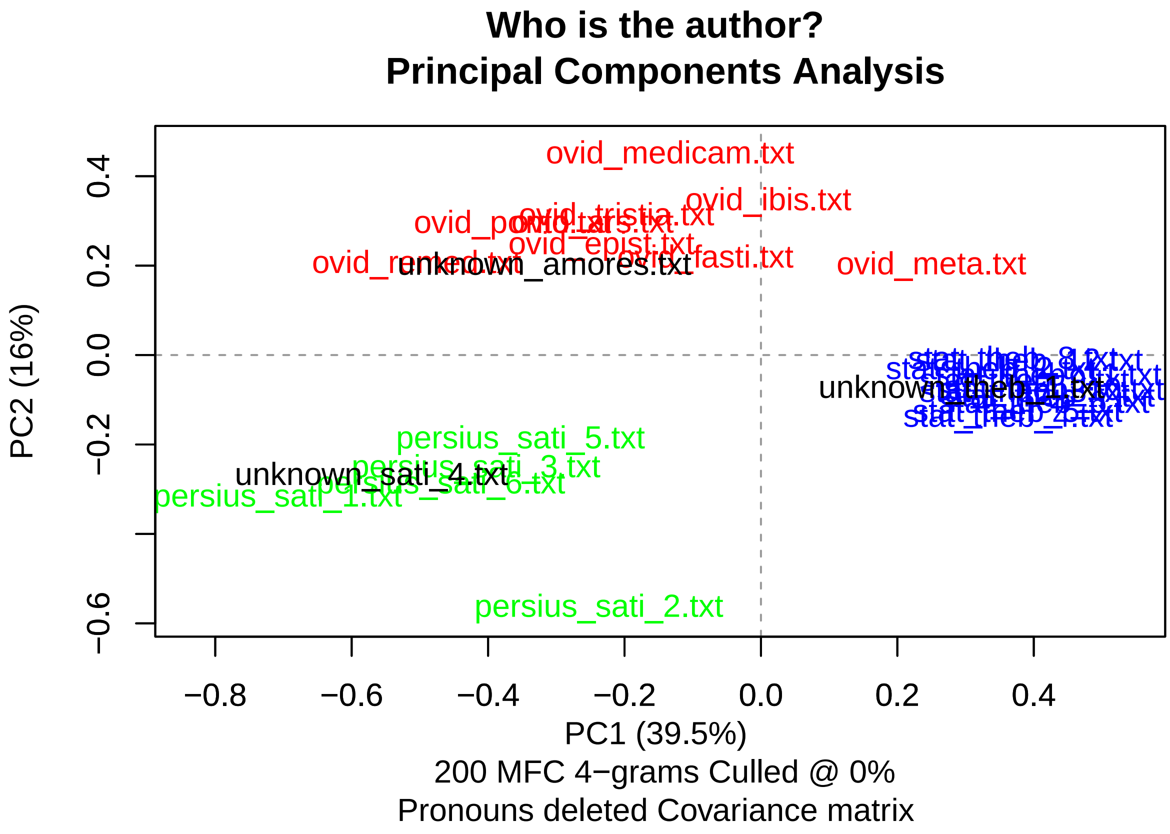
The same as before will be applied here. In this case we will use a covariance matrix, but using the same range of MFC and the same amount of increment in each iteration. Moreover, the same distance metric will be used as before (i.e., Eder's Delta)

```
# PCA covariance - top 1500 incr.100 MFCs 4-grams
results_pca_4grams_cov = stylo(frequencies = freqs.4grams, analysis.type = "PCR",
                               mfw.min = 100, mfw.max = 1500, increment=100, # range of MFC
                               distance.measure = "eder", # Eder's Delta
                               custom.graph.title = "Who is the author?", # title of the plot
                               pca.visual.flavour="classic", # flavour of the plot
                               write.png.file=T, gui = T) # gui=T to double-check the parameters set
```

```
## using current directory...

## Warning in delete.stop.words(table.with.all.freqs, pronouns): chosen stop words were not found in the
##   please check the language, lower/uppercase issues, etc.

##

## culling @ 0  available features (words) 2000

## MFW used:

## 100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```

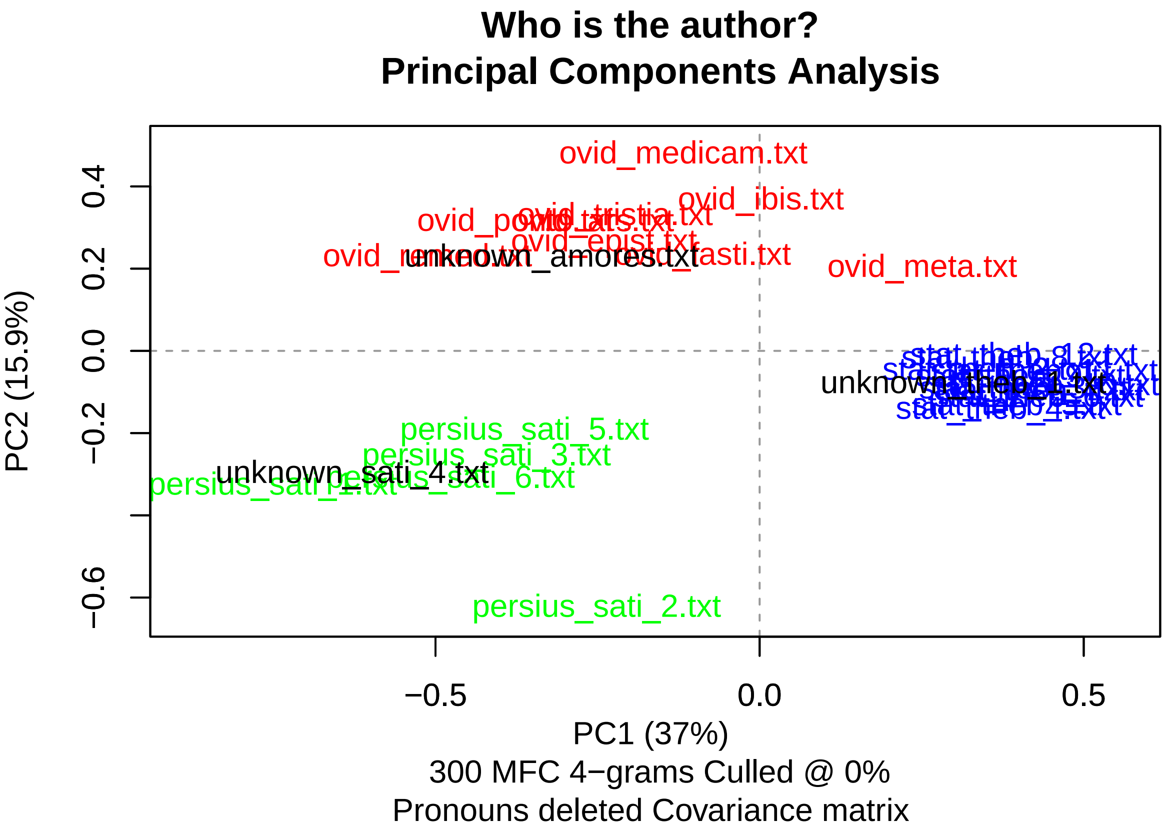


```
## 300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```

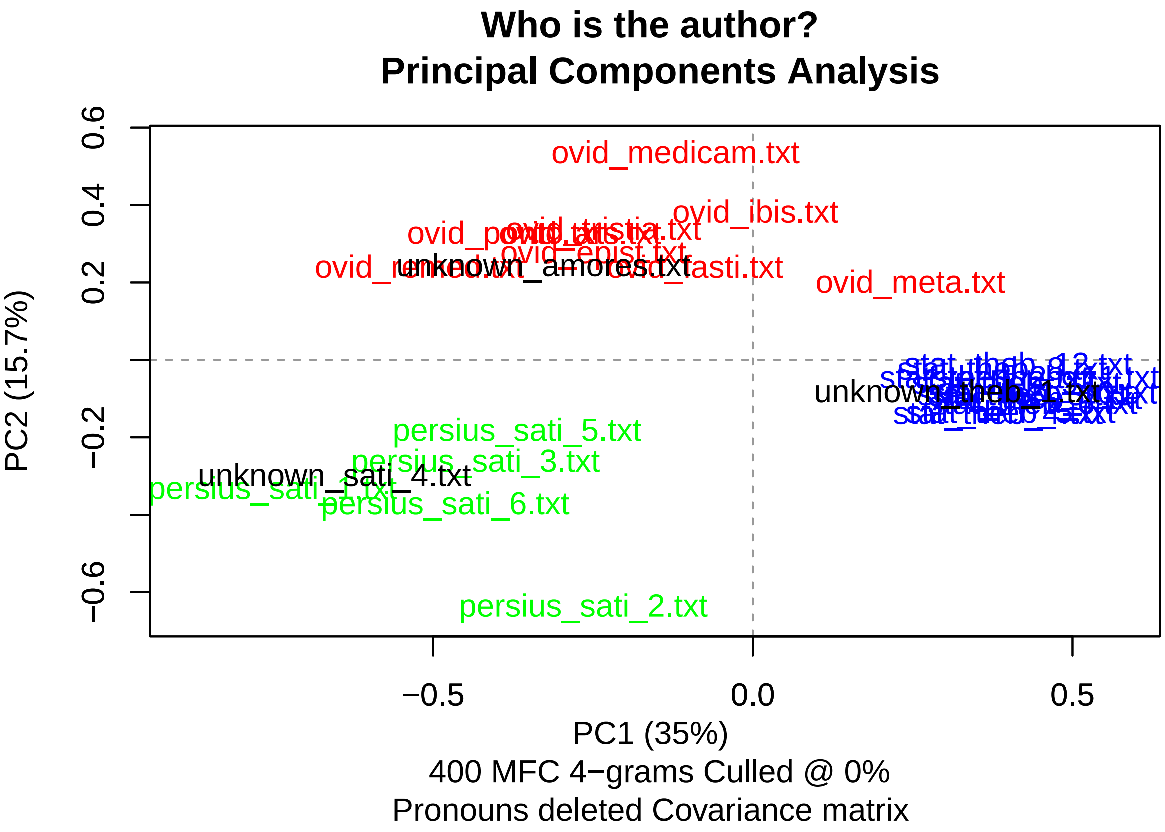


```
## 400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```

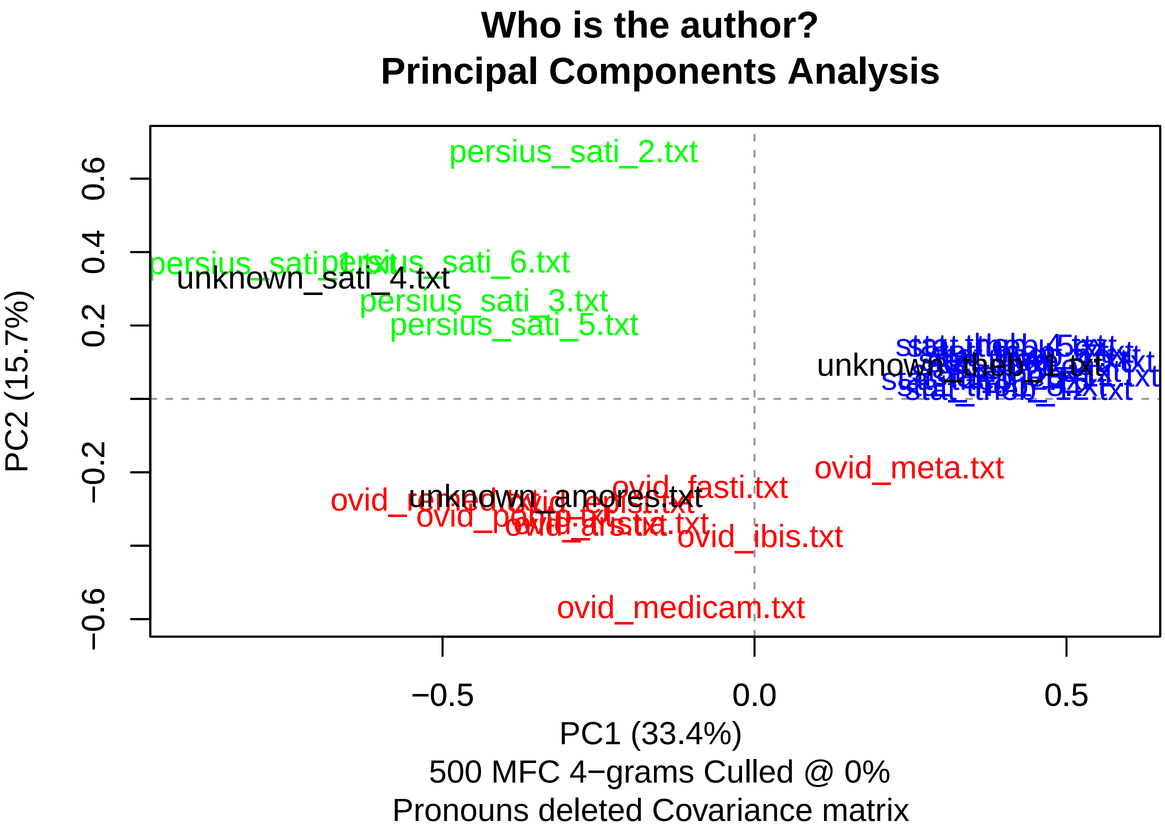
]



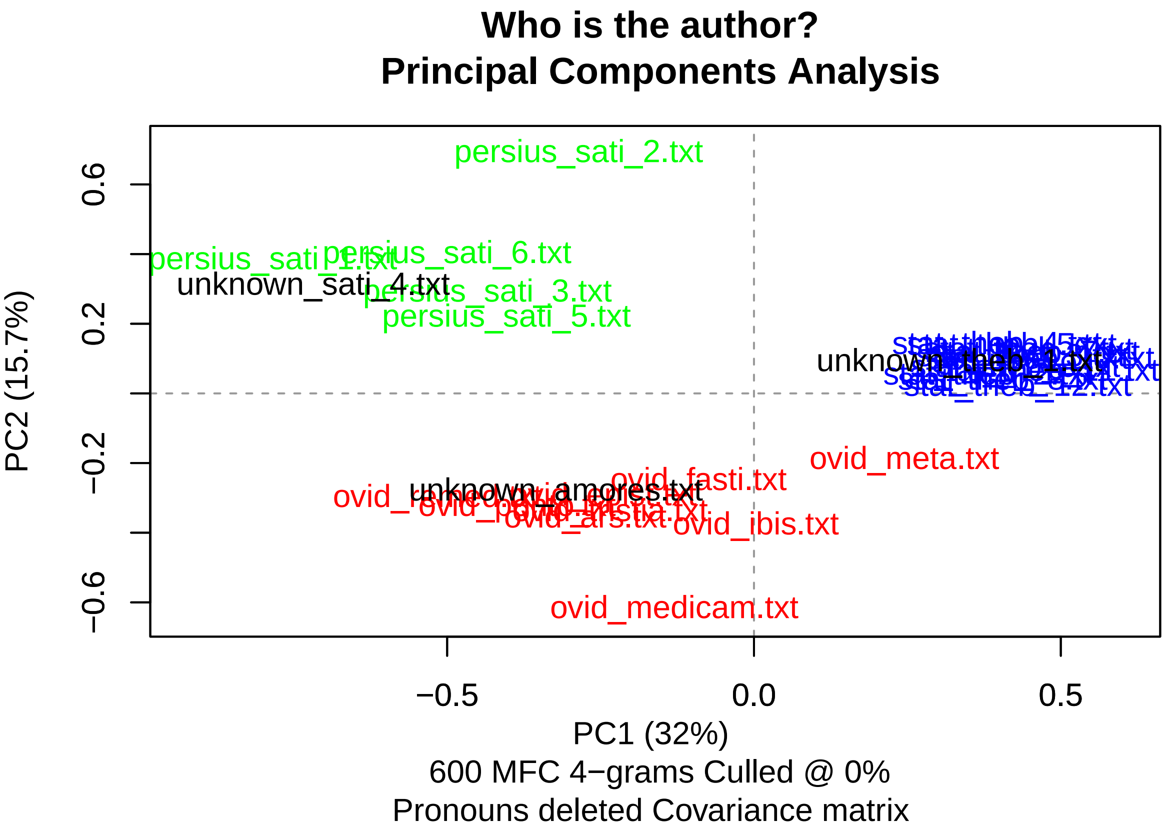
```
## 500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



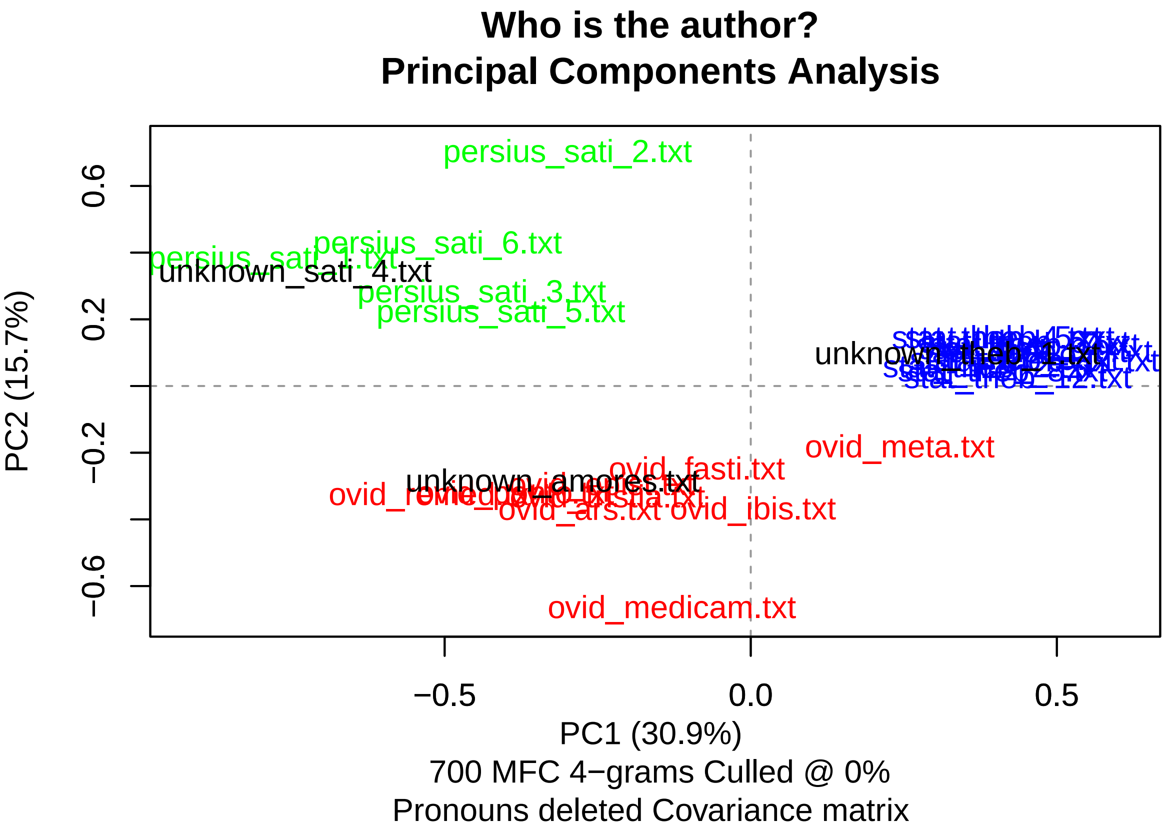
```
## 600
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



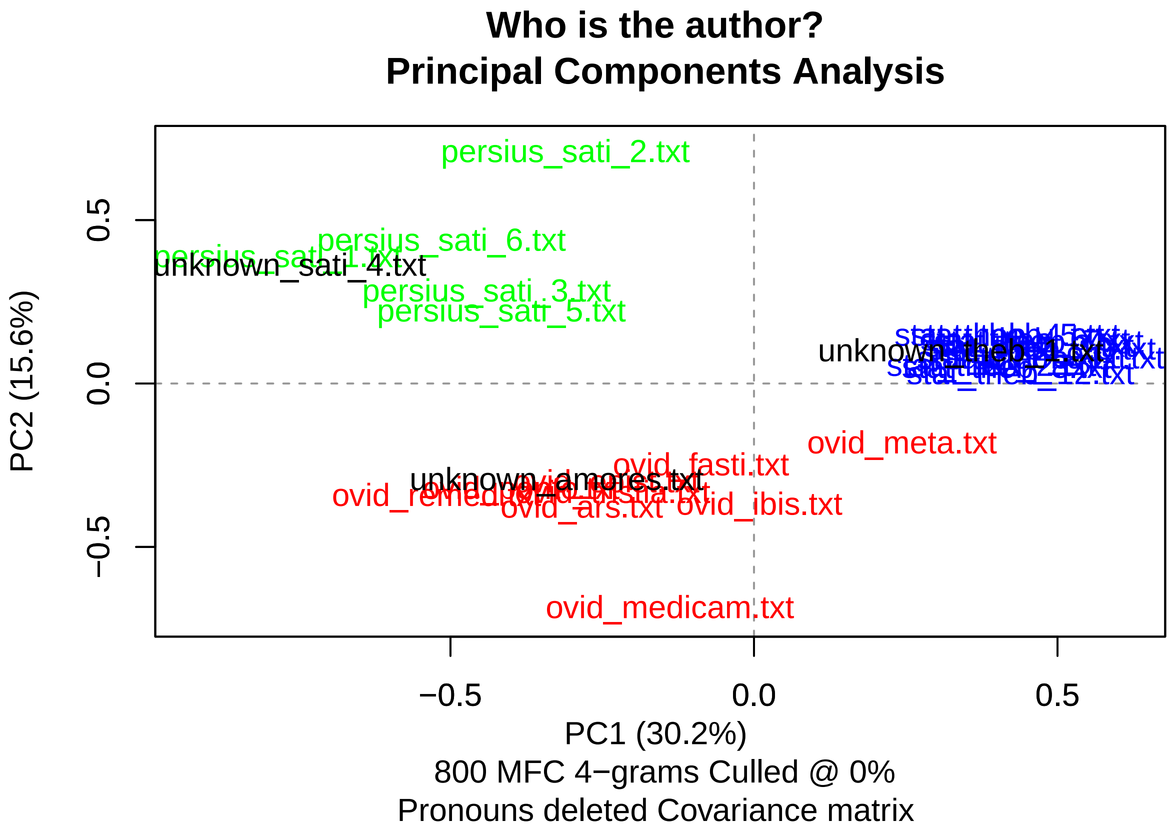
```
## 700
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



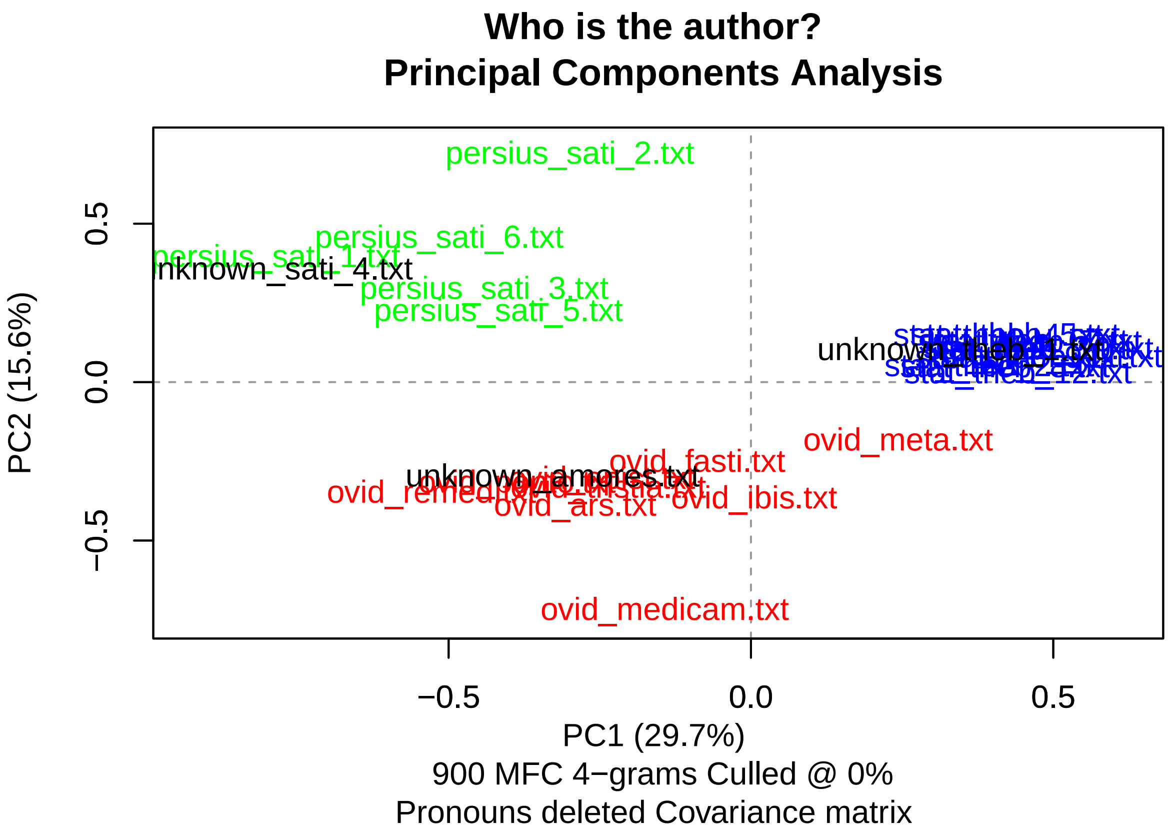
```
## 800
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



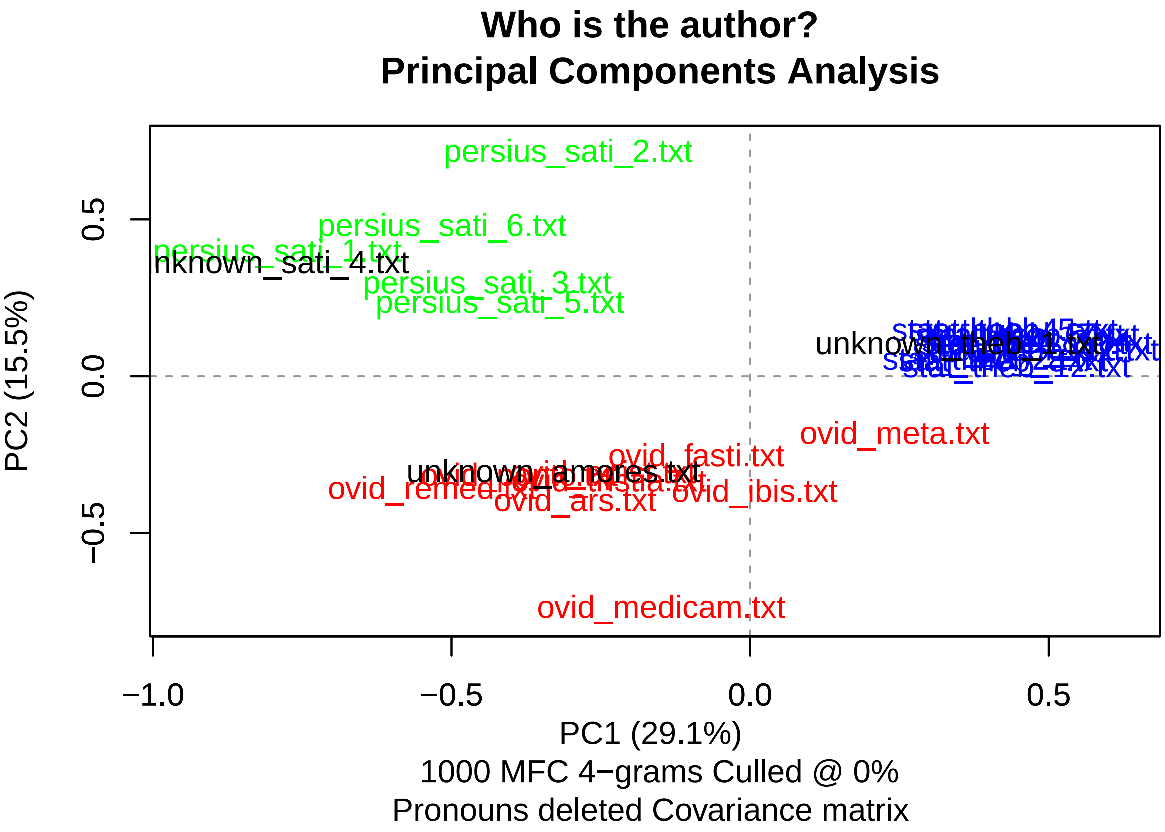
```
## 900
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



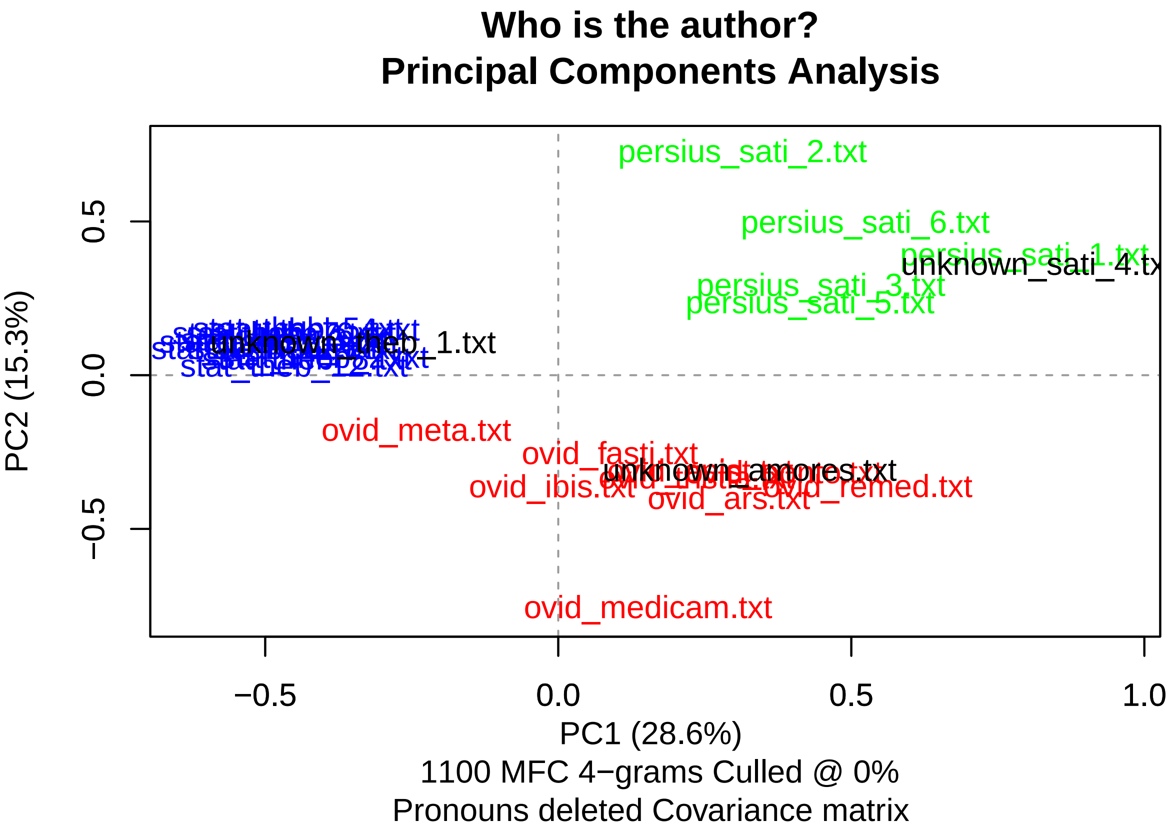
```
## 1000
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



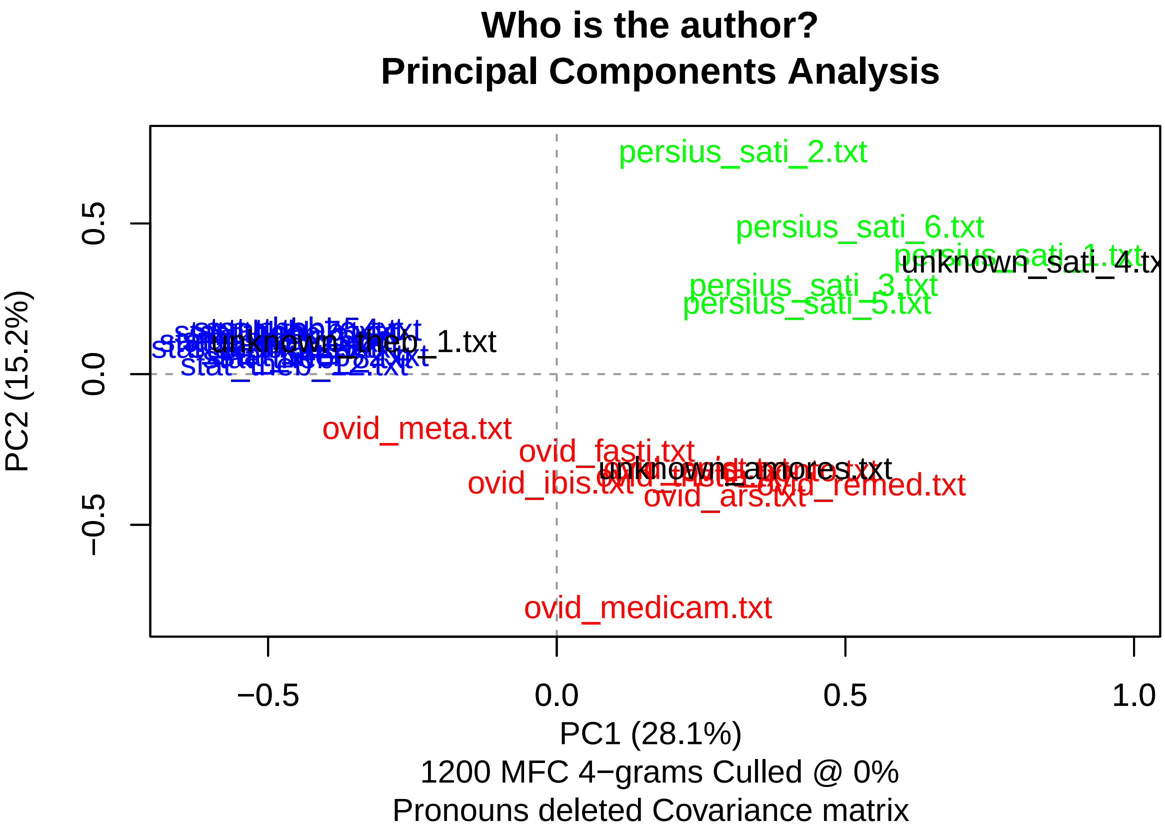
```
## 1100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



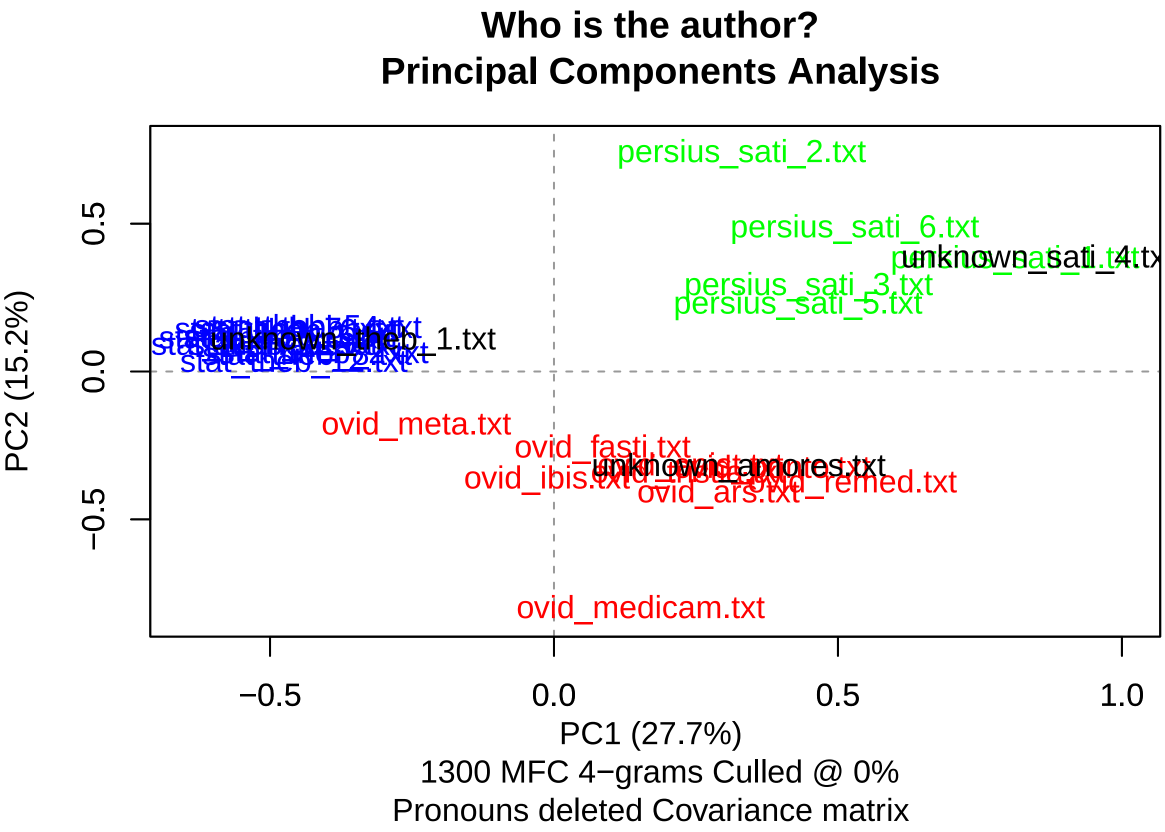
```
## 1200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



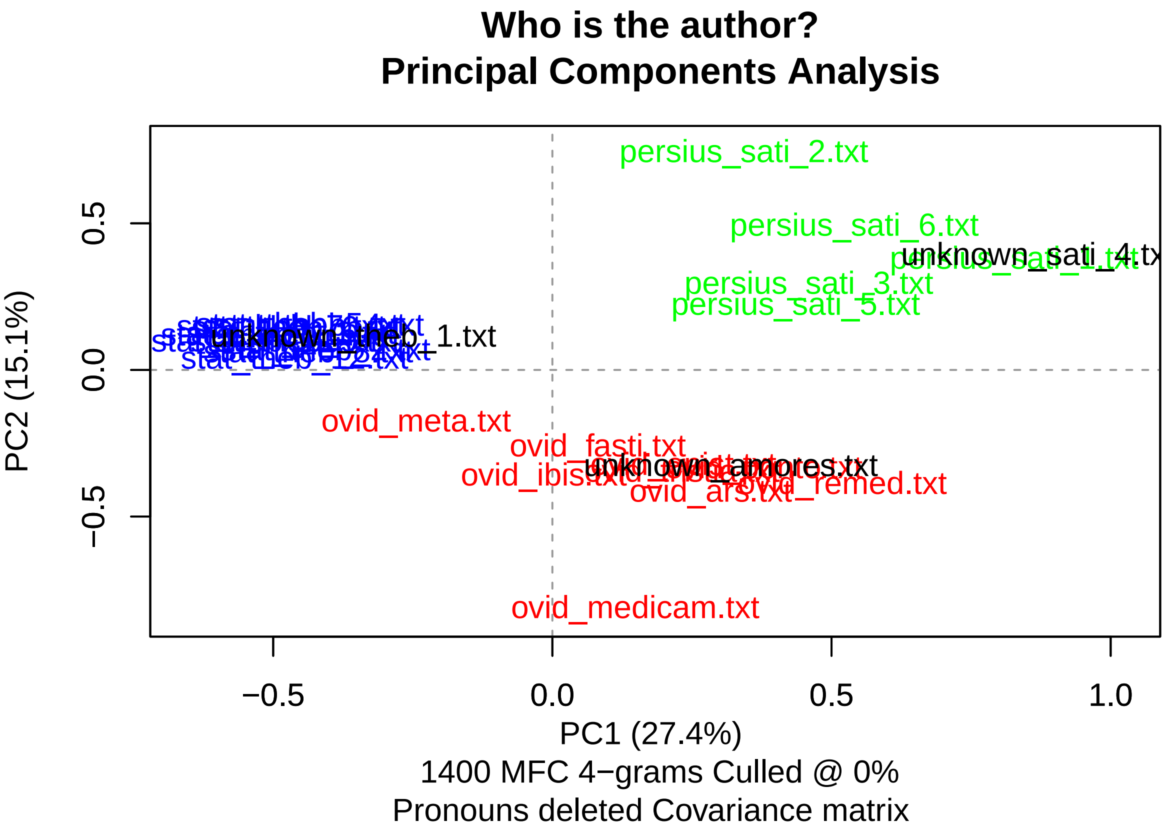
```
## 1300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



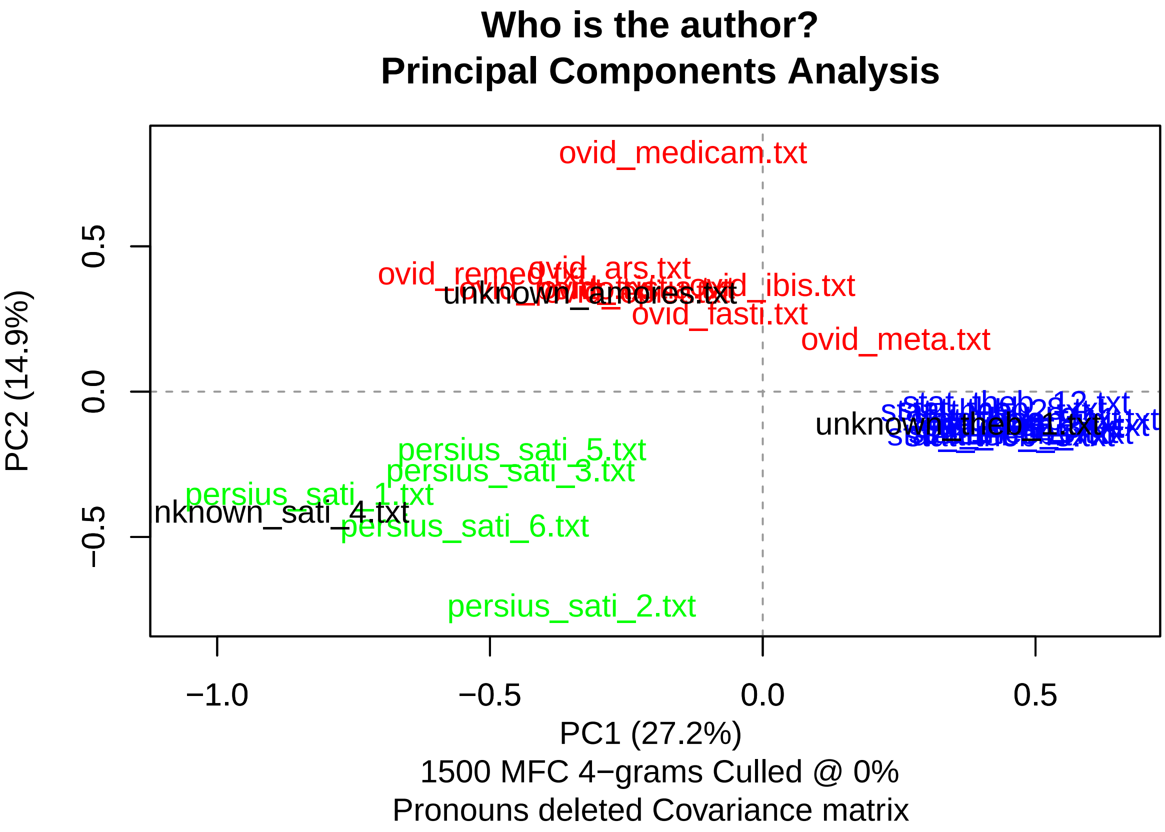
```
## 1400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



```
## 1500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```


Who is the author?
Principal Components Analysis
persius_sati_2.txt
persius_sati_6.txt
unknown_sati_4.tx
persius_sati_1.txt
persius_sati_3.txt
persius_sati_5.txt
unknown_theb_1.txt
stat_theb_12.txt
ovid_meta.txt
ovid_fasti.txt
unknown_amores.txt
ovid_ibis.txt
ovid_remed.txt
ovid_ars.txt
ovid_medicam.txt
PC2 (15.1%)
0.5
0.0
−0.5
−0.5
0.0
0.5
1.0
PC1 (27.4%)
1400 MFC 4−grams Culled @ 0%
Pronouns deleted Covariance matrix

##


Who is the author?
Principal Components Analysis
ovid_medicam.txt
ovid_remed.txt
ovid_ars.txt
ovid_ibis.txt
unknown_amores.txt
ovid_fasti.txt
ovid_meta.txt
stat_theb_12.txt
unknown_theb_1.txt
persius_sati_5.txt
persius_sati_3.txt
persius_sati_1.txt
nknown_sati_4.txt
persius_sati_6.txt
persius_sati_2.txt
PC2 (14.9%)
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
PC1 (27.2%)
1500 MFC 4−grams Culled @ 0%
Pronouns deleted Covariance matrix

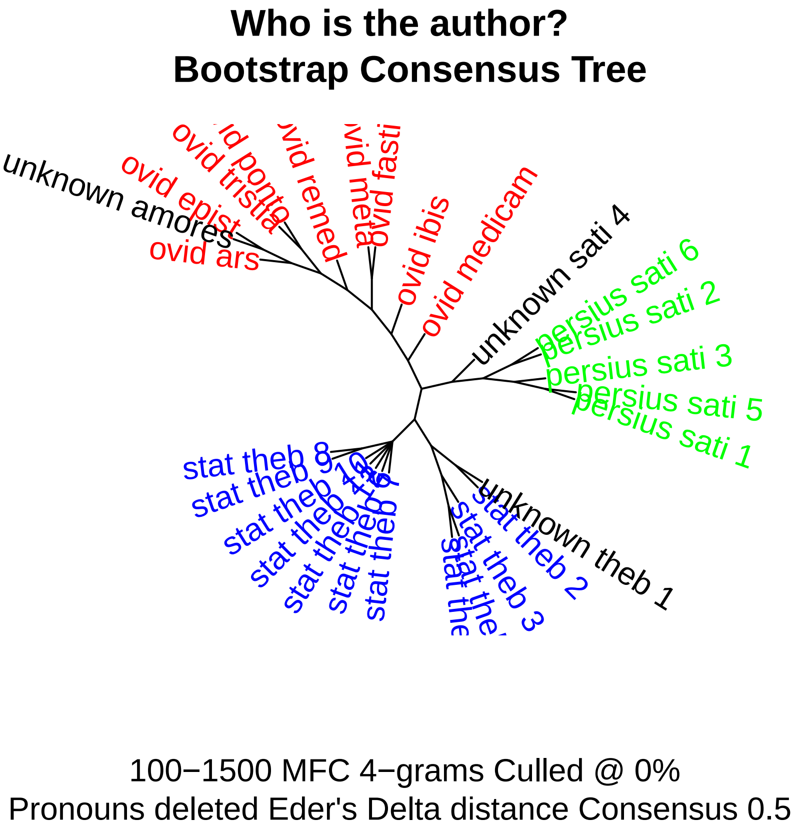
Apply Bootstrap Consensus Tree - MFC 4-grams

```
# BCT 4grams - top 100-1500-100 MFC 4 grams - consensus strength 0.5
bct.results.4grams_100_1500MFC = stylo(corpus.dir = "validation_corpus/", frequencies = freqs.4grams,
                                       distance.measure="eder",
                                       analysis.type = "BCT",
                                       mfw.min = 100, mfw.max = 1500, increment = 100,
                                       custom.graph.title="Who is the author?",
                                       write.png.file=T,
                                       gui = TRUE)
```

```
## using current directory...

## Warning in delete.stop.words(table.with.all.freqs, pronouns): chosen stop words were not found in th
##   please check the language, lower/uppercase issues, etc.

##

## culling @ 0  available features (words) 2000

## Calculating z-scores...

## Calculating Eder's Delta distances...

## MFW used:

## 100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 600
## Processing metadata...
```

```
##
##
## Assigning plot colors according to file names...
##
## 700
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 800
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 900
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1000
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1500
## Processing metadata...
```

```
##
##
## Assigning plot colors according to file names...
##
##
```



Character 5-grams

Extracting the features (character 5-grams)

In this section we will extract a new kind of feature (character 5-grams) and proceed to exactly the same analysis

```
corpus.char.5.grams <- txt.to.features(corpus.no.pronouns,
                                       features = "c",
                                       ngram.size = 5)

frequent.features.5grams <- make.frequency.list(corpus.char.5.grams,
                                                head = 2000)

freqs.5grams <- make.table.of.frequencies(corpus.char.5.grams,
                                          features = frequent.features.5grams,
                                          relative = TRUE)
```

```
## processing  28  text samples
```

```
## ..
## combining frequencies into a table...
```

Methods - Character 5-grams

Apply Principal Component Analysis

Principal Component Analysis - Correlation matrix (MFCs 5grams)

In this experiment we will apply Principal Component Analysis (henceforward: PCA) using a correlation matrix to visualize the results. The features used in this experiment are Most Frequent Characters (henceforward: MFCs) 5-grams. We will look at the top 100 to 1500 MFCs 4-grams with an increment of 100 in each iteration (no culling will be specified because we want to obtain a sufficient number of features in each iteration (given this corpus, if we set culling to 100% we obtain only 33 MFC)).

Eder's Delta will be used as a distance metric.

```
# PCA correlation - top 100-1500-100 | MFCs 5grams
pca_5grams_cor = stylo(frequencies = freqs.5grams, analysis.type = "PCR",
                       mfw.min = 100, mfw.max = 1500, increment=100,
                       distance.measure = "eder",
                       custom.graph.title = "Who is the author?",
                       write.png.file=T,
                       gui = T)
```

```
## using current directory...
```

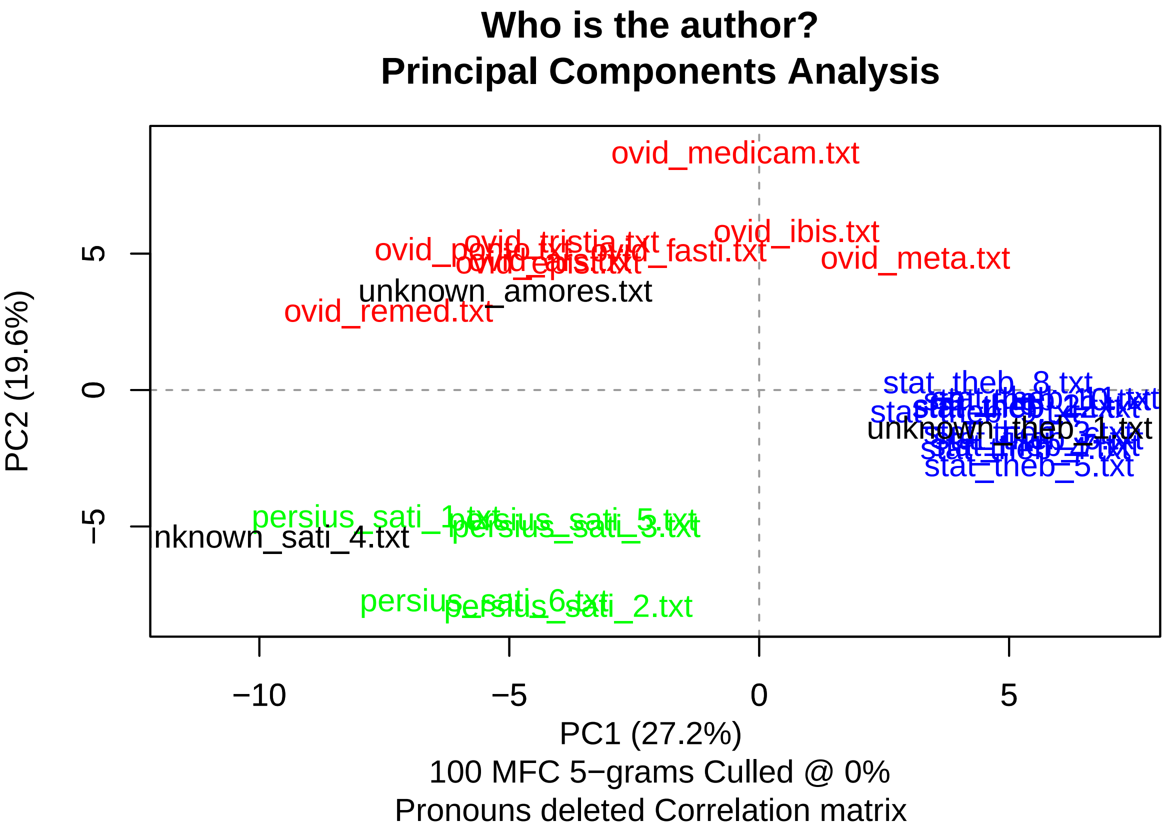
```
## Warning in delete.stop.words(table.with.all.freqs, pronouns): chosen stop words were not found in th
##   please check the language, lower/uppercase issues, etc.
```

```
##
```

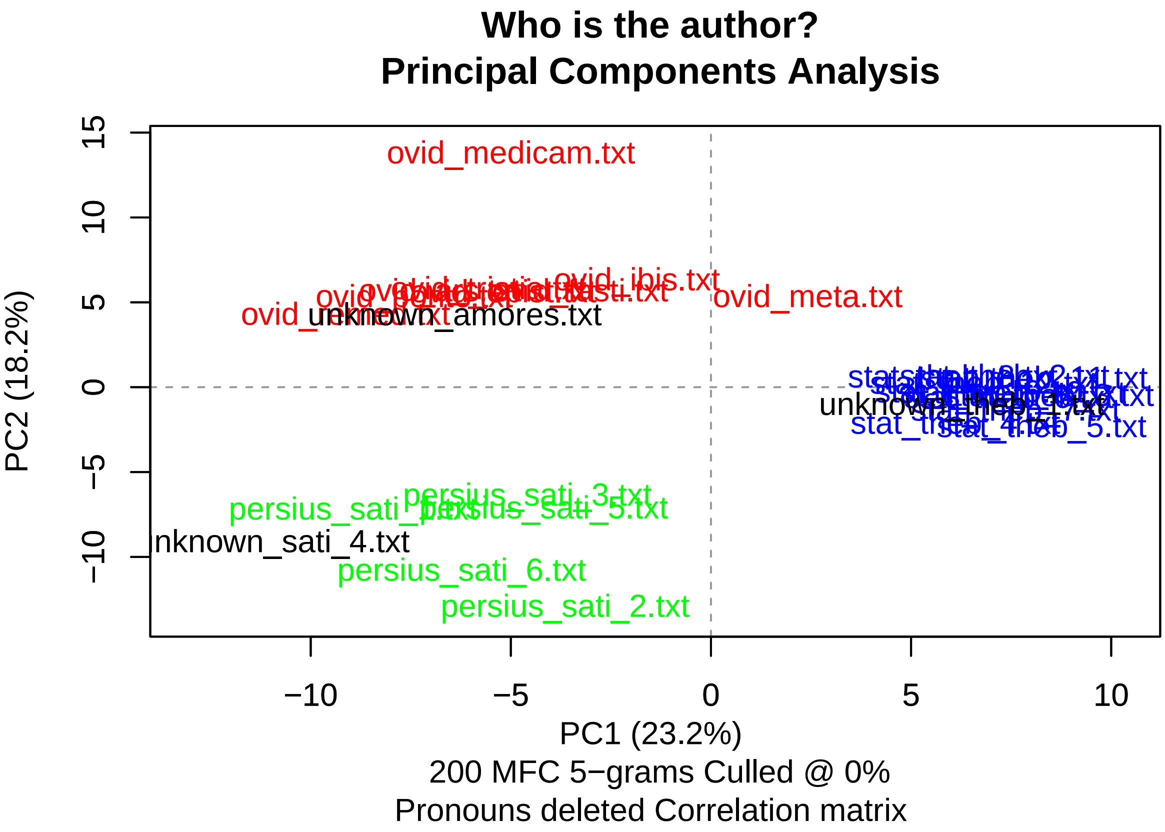
```
## culling @ 0  available features (words) 2000
```

```
## MFW used:
```

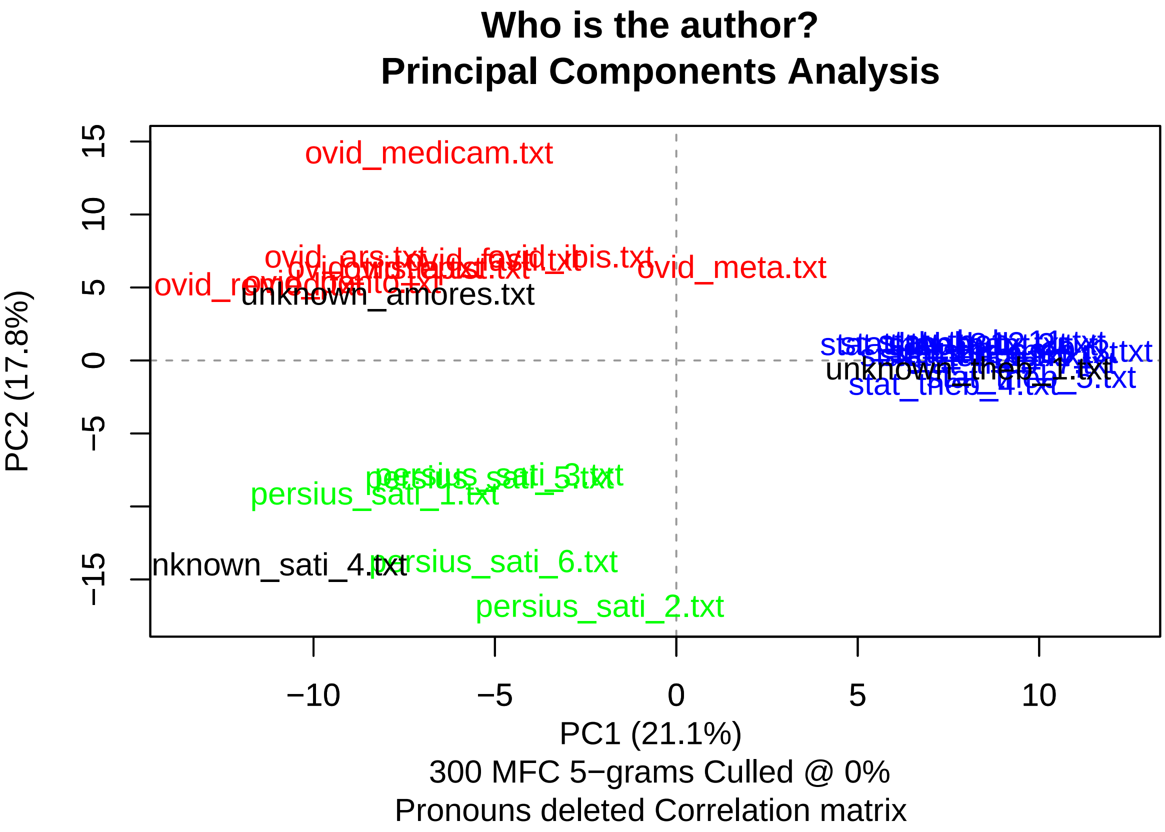
```
## 100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



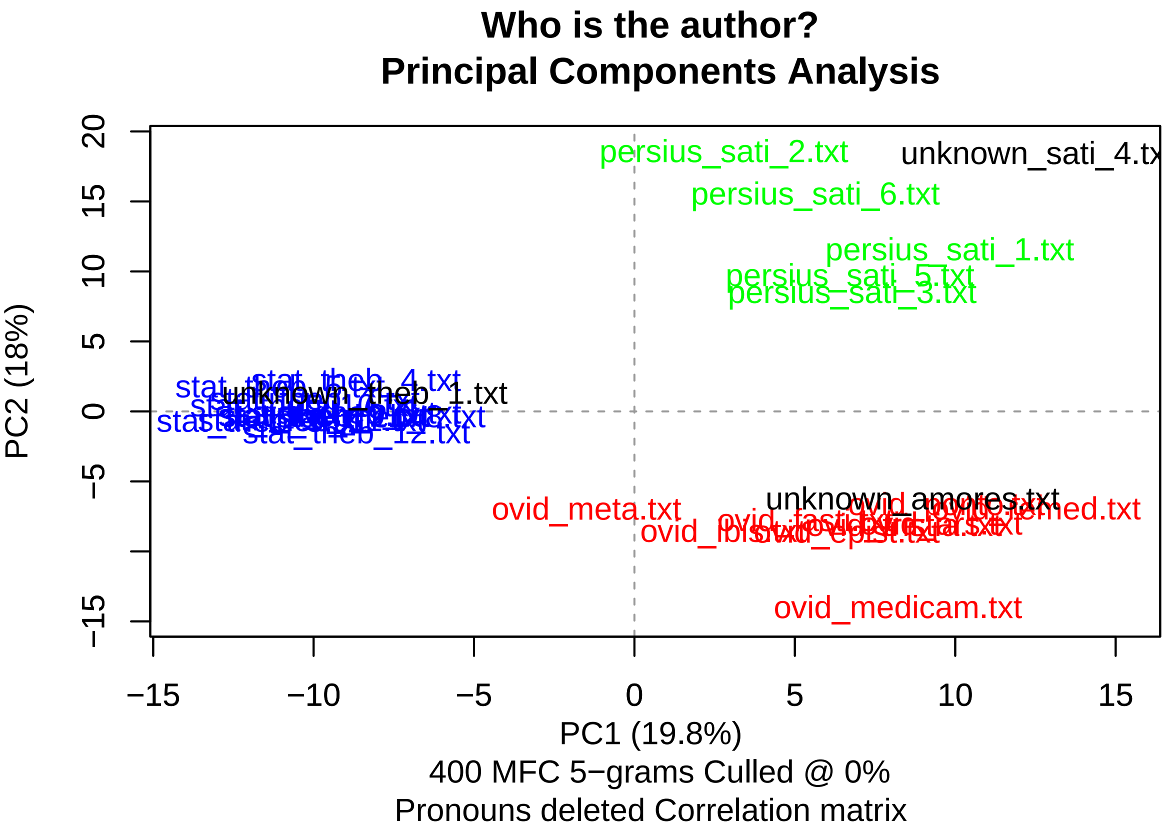
```
## 300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



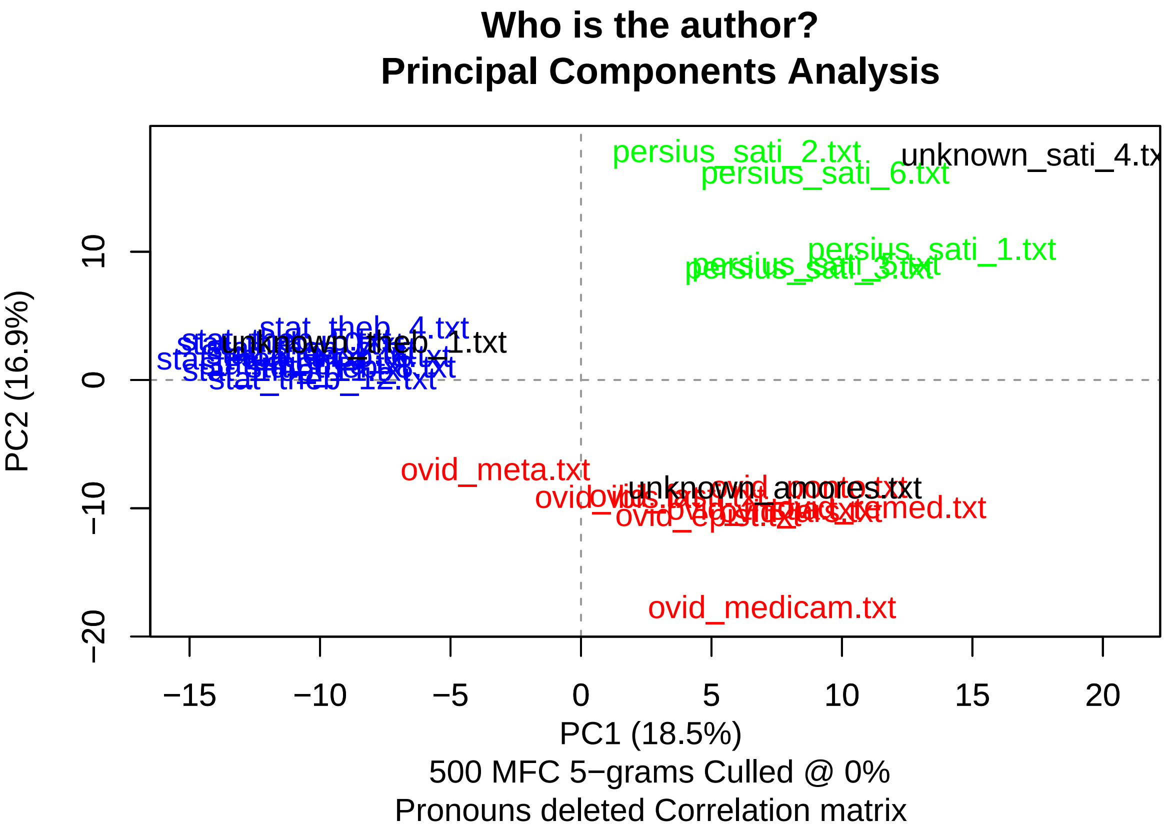
```
## 400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



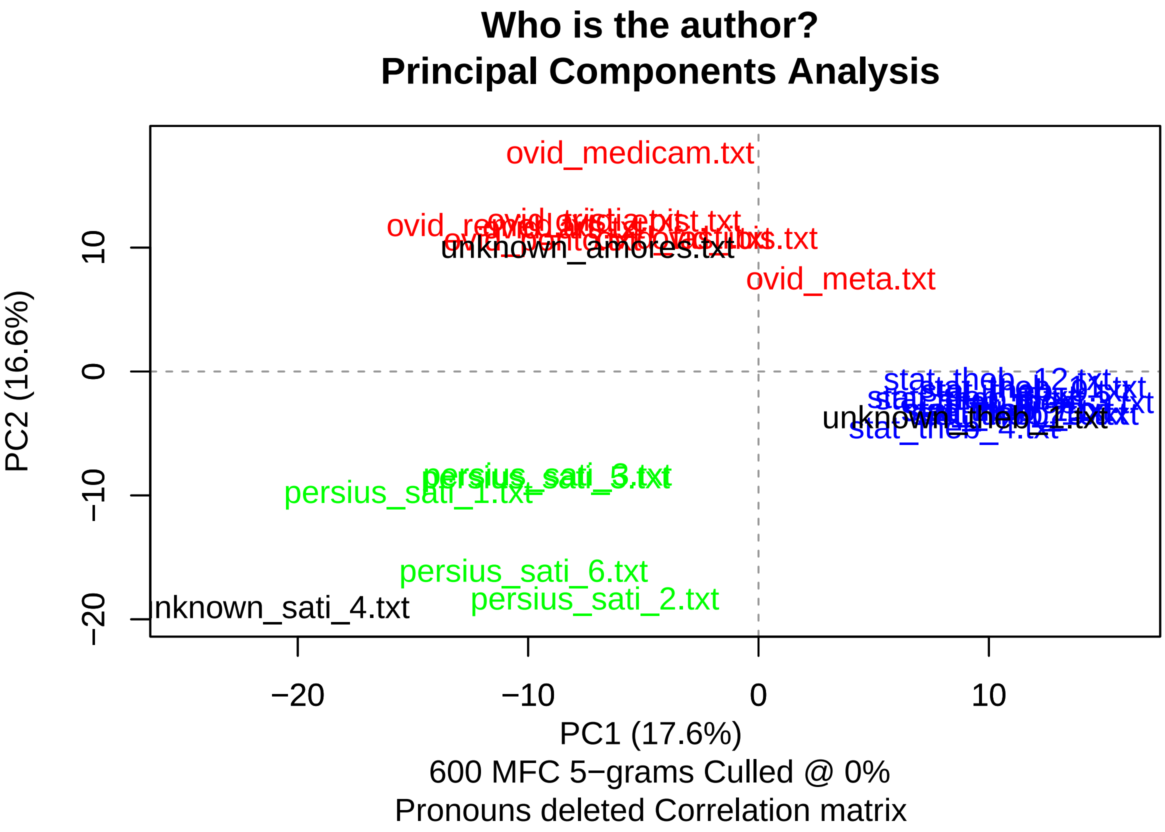
```
## 500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



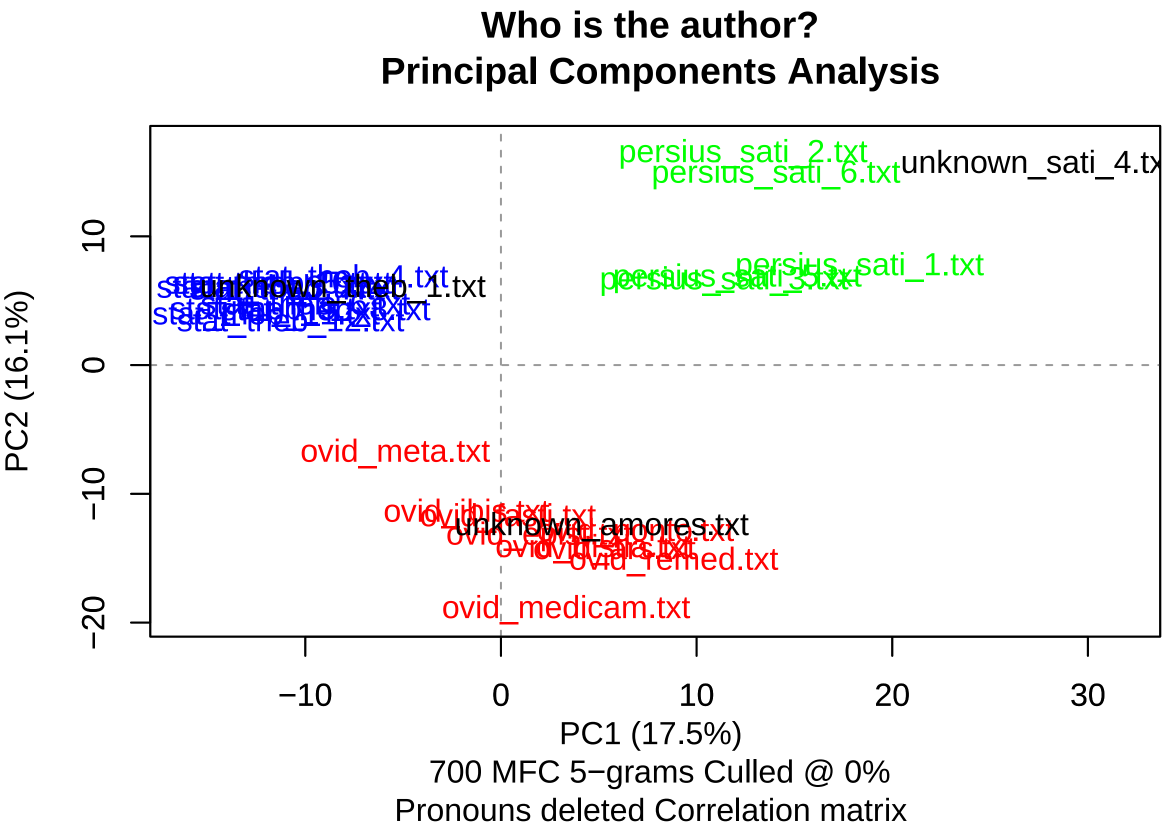
```
## 600
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



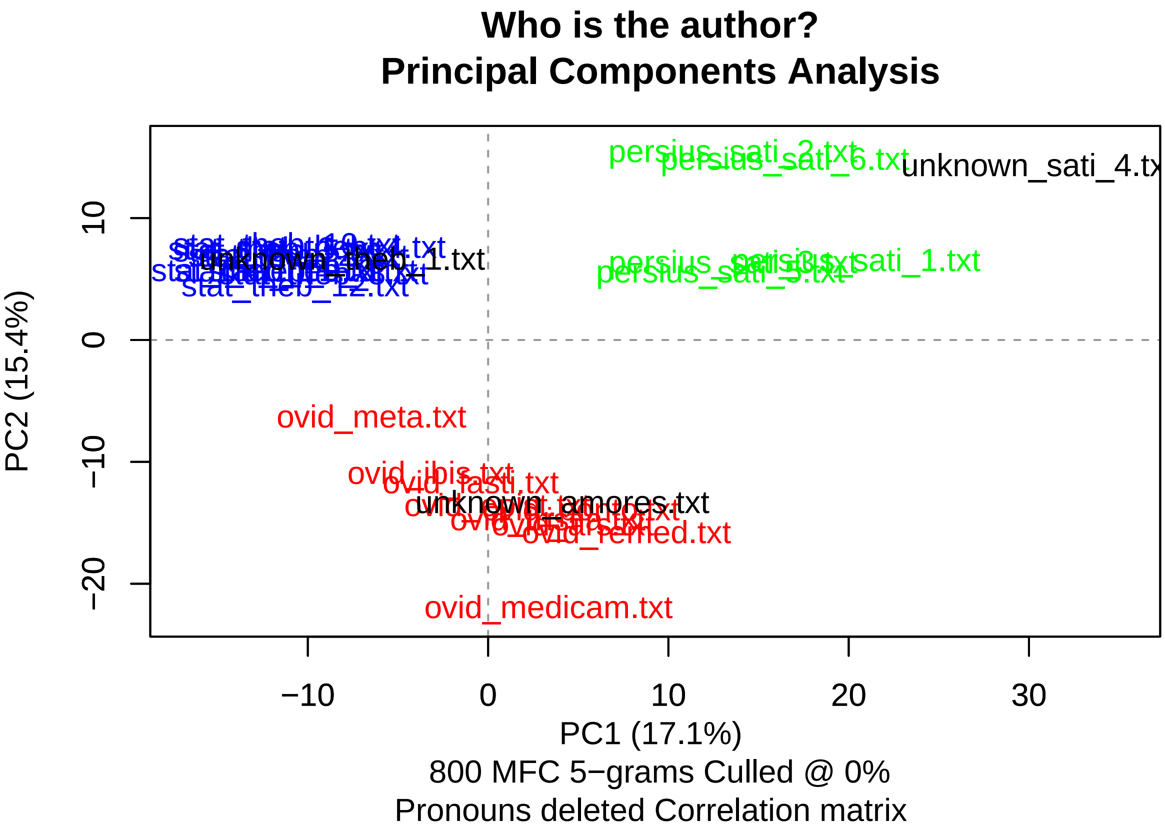
```
## 700
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



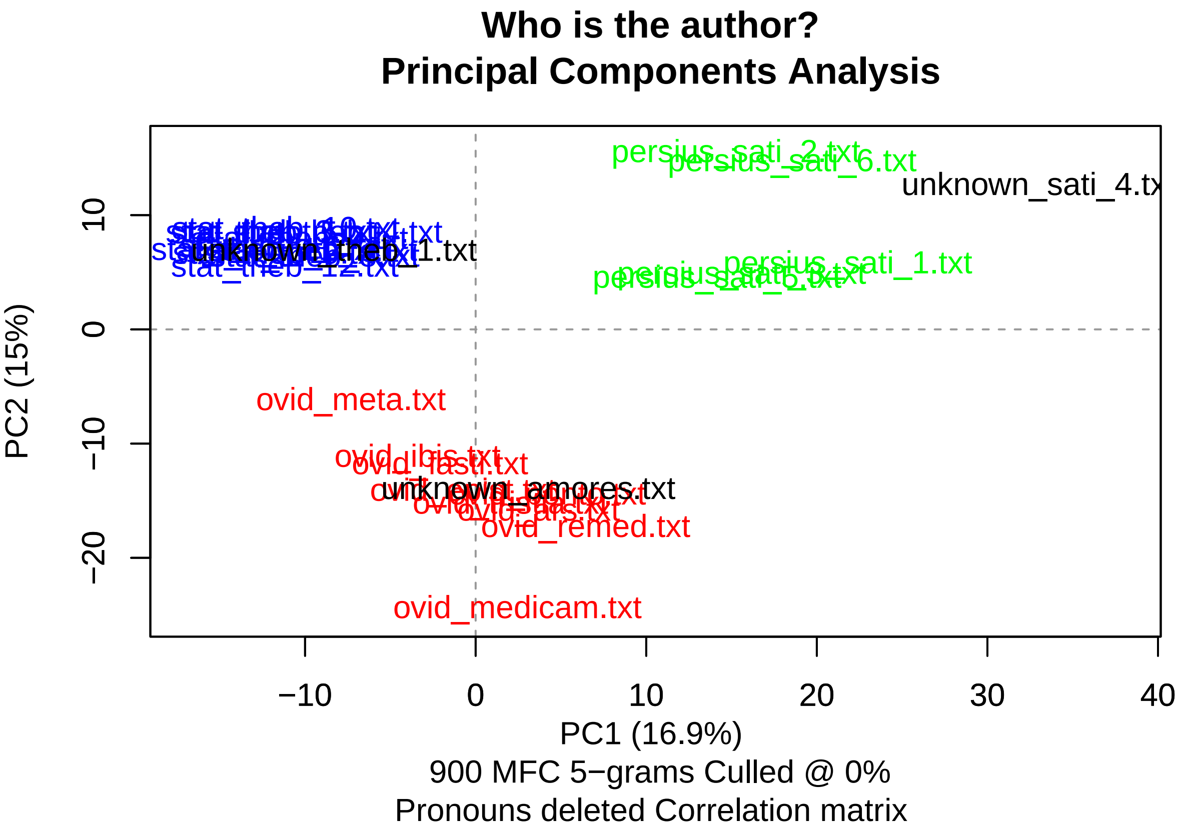
```
## 800
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



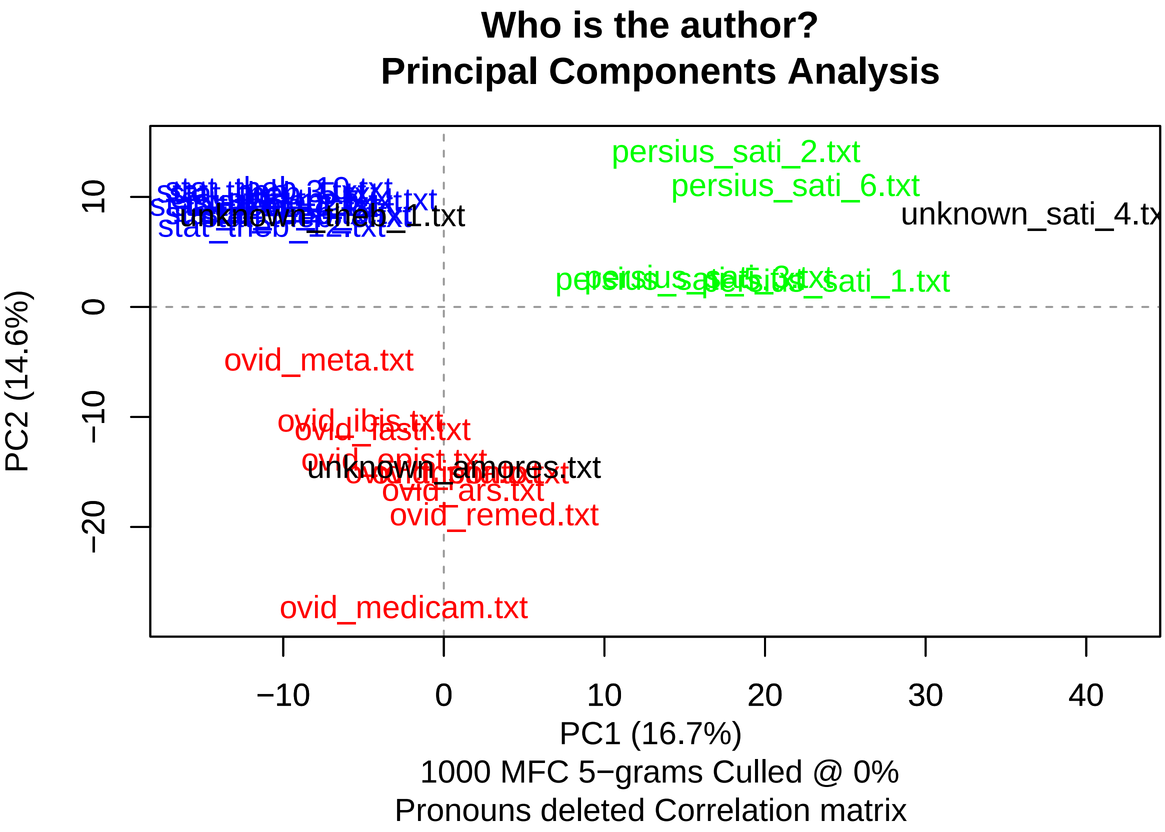
```
## 900
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



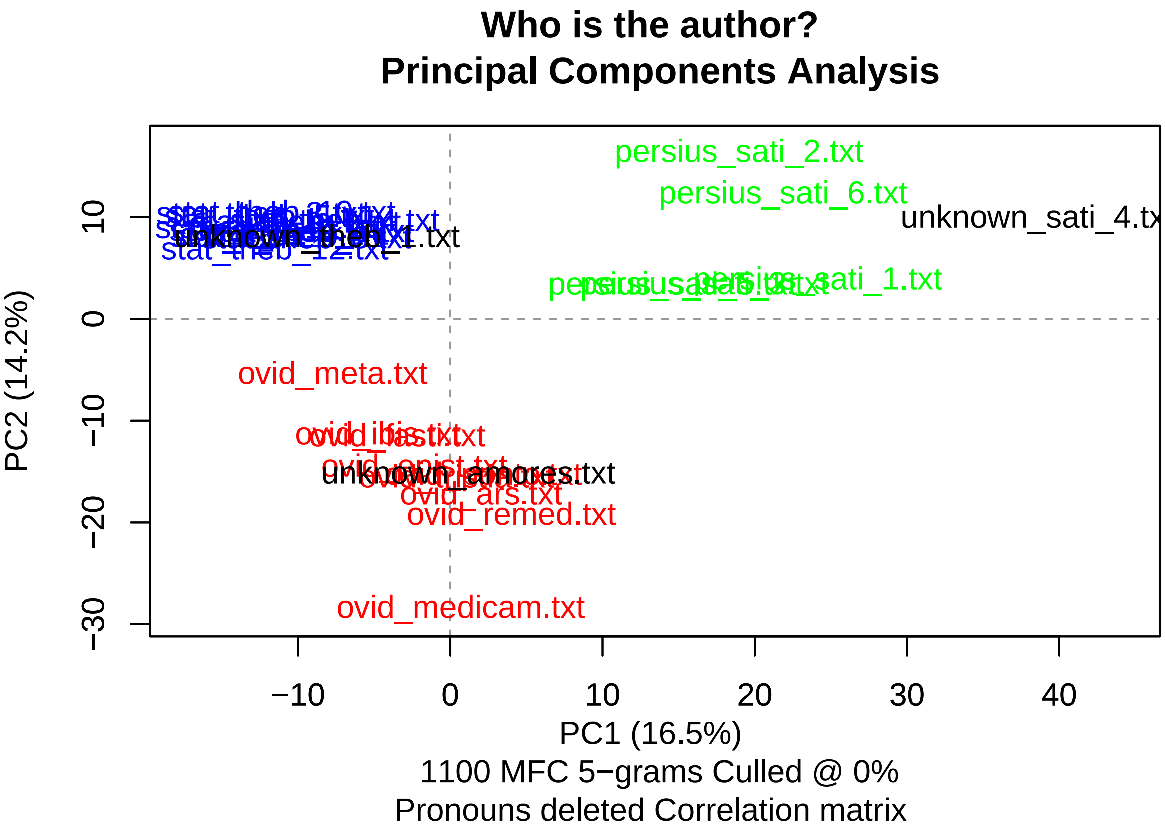
```
## 1000
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



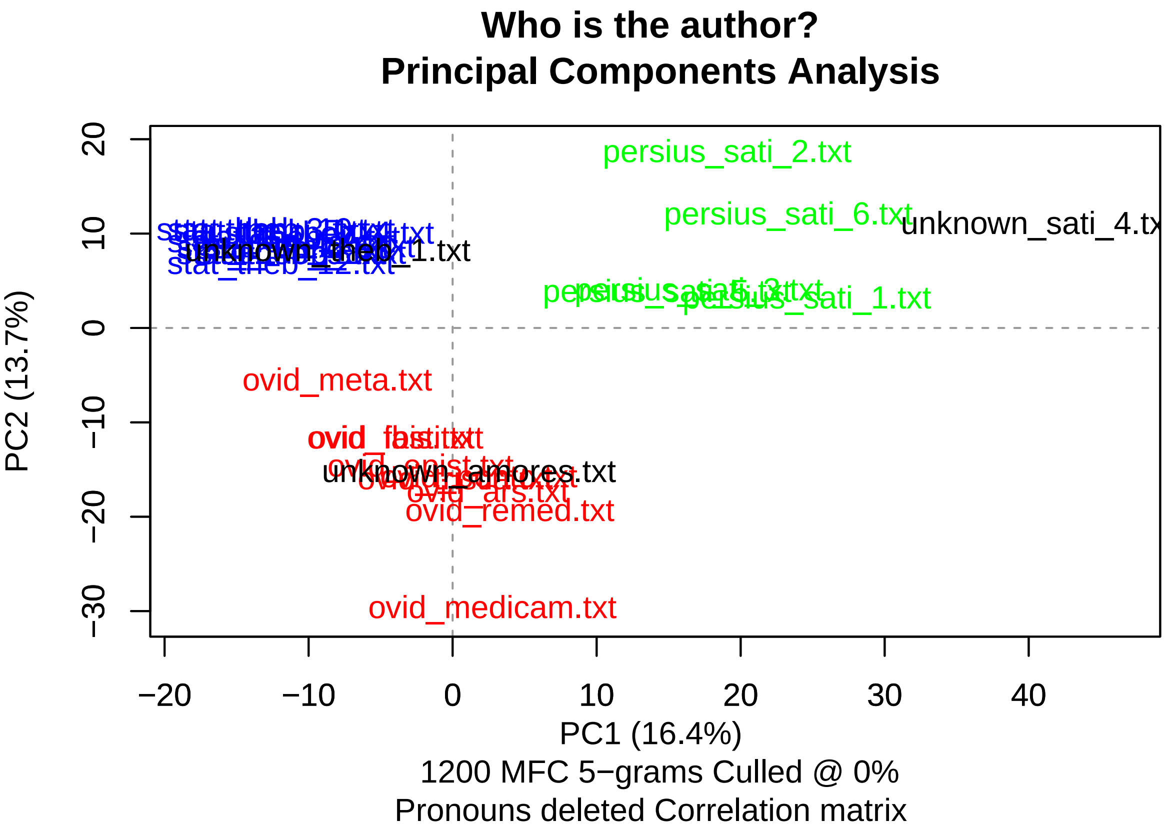
```
## 1100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



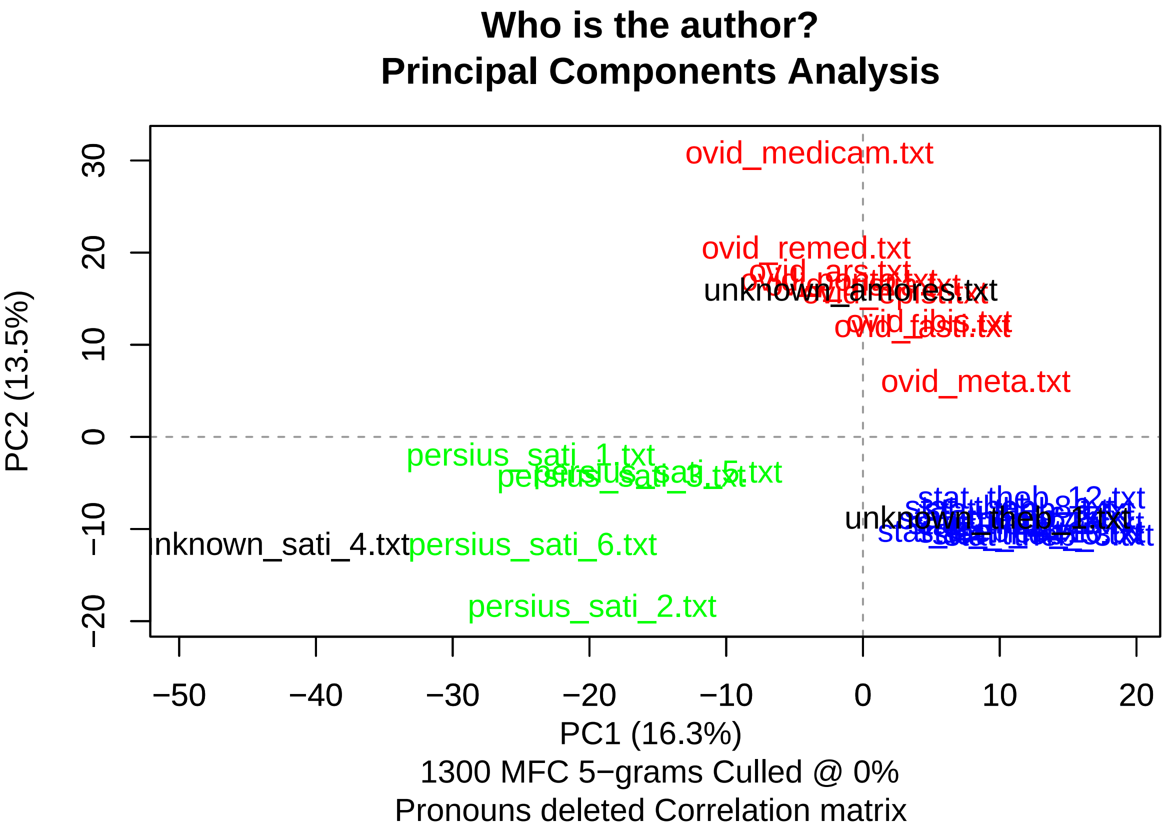
```
## 1200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



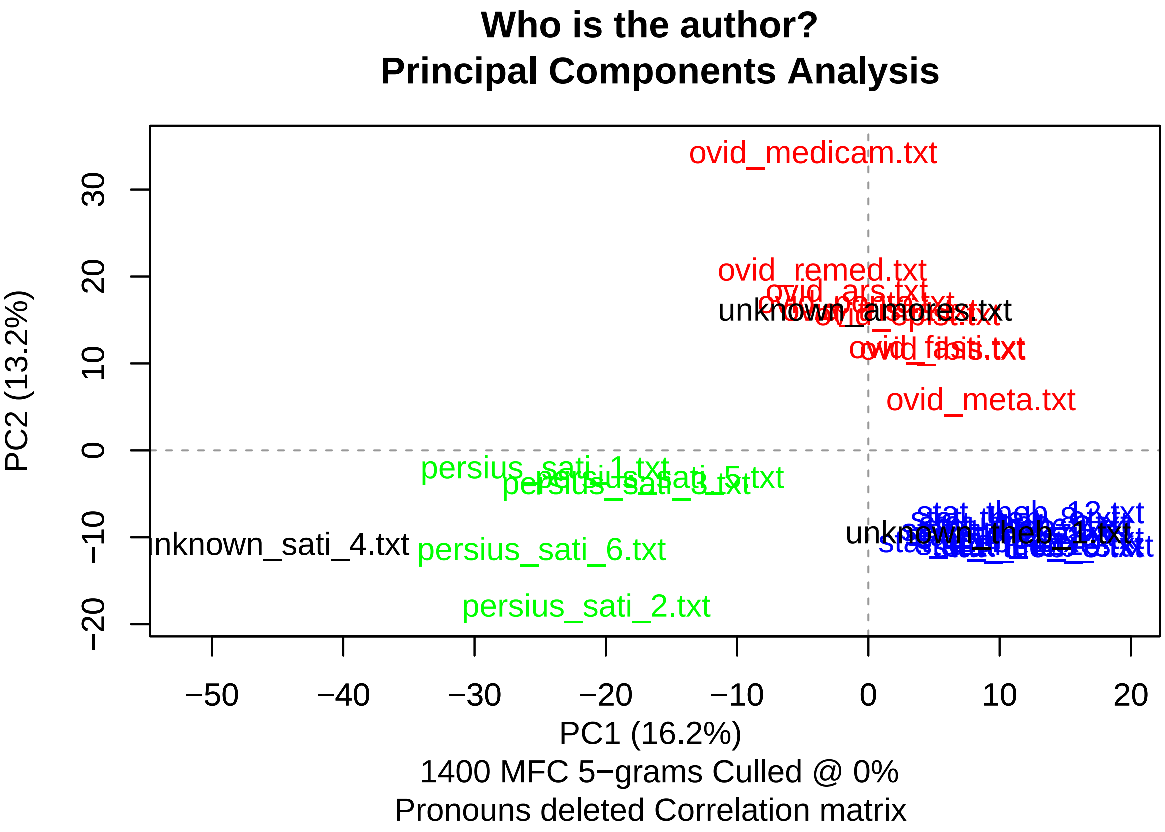
```
## 1300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



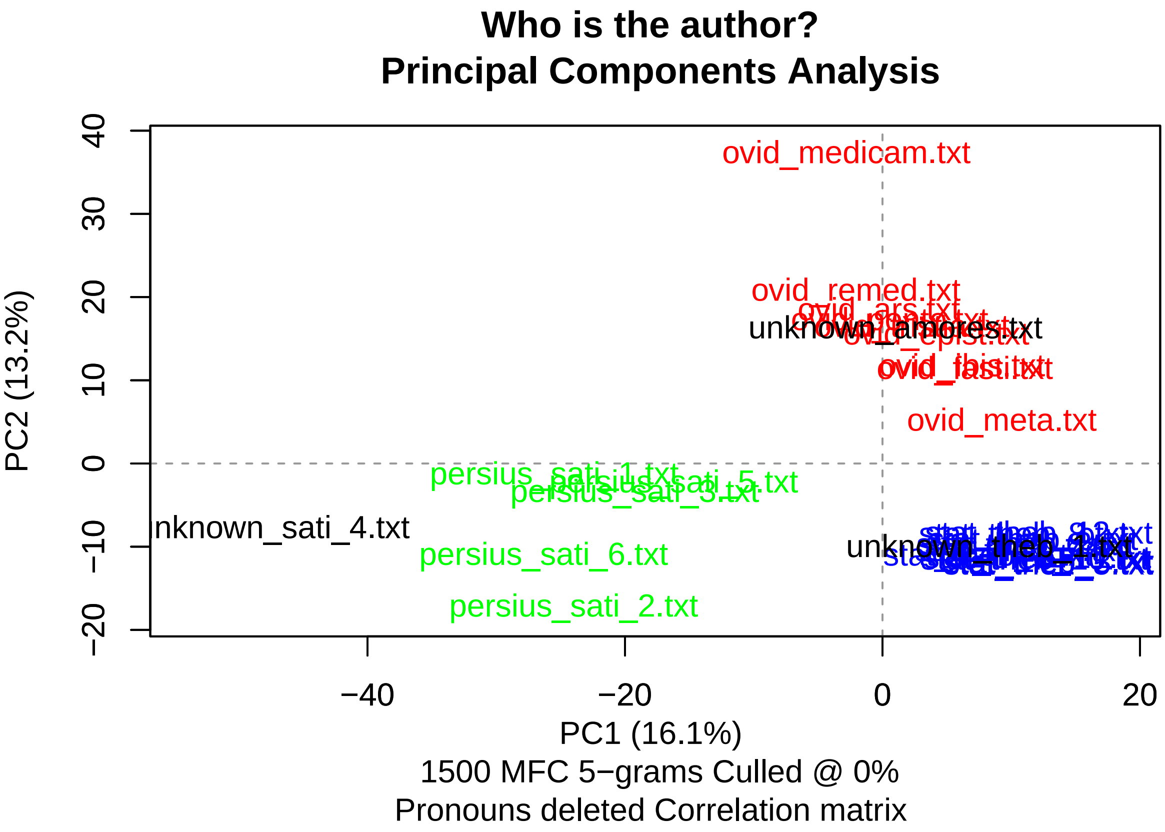
```
## 1400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



```
## 1500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```


Who is the author?
Principal Components Analysis
ovid_medicam.txt
ovid_remed.txt
ovid_ars.txt
unknown_amores.txt
ovid_epist.txt
ovid_fasti.txt
ovid_ibis.txt
ovid_meta.txt
persius_sati_1.txt
persius_sati_5.txt
persius_sati_3.txt
nknown_sati_4.txt
persius_sati_6.txt
persius_sati_2.txt
stat_theb_12.txt
unknown_theb_1.txt
PC2 (13.2%)
30
20
10
0
−10
−20
−50
−40
−30
−20
−10
0
10
20
PC1 (16.2%)
1400 MFC 5−grams Culled @ 0%
Pronouns deleted Correlation matrix

##


Who is the author?
Principal Components Analysis
ovid_medicam.txt
ovid_remed.txt
ovid_ars.txt
unknown_amores.txt
ovid_epist.txt
ovid_fasti.txt
ovid_ibis.txt
ovid_meta.txt
persius_sati_1.txt
persius_sati_5.txt
persius_sati_3.txt
nknown_sati_4.txt
persius_sati_6.txt
persius_sati_2.txt
stat_theb_12.txt
unknown_theb_1.txt
PC2 (13.2%)
40
30
20
10
0
−10
−20
−40
−20
0
20
PC1 (16.1%)
1500 MFC 5−grams Culled @ 0%
Pronouns deleted Correlation matrix

Principal Component Analysis - Covariance matrix (MFCs 5grams)

```
# PCA covariance matrix | 100-1500-100 MFC 5grams
pca_5grams_cov = stylo(frequencies = freqs.5grams, analysis.type = "PCR",
                       mfw.min=100, mfw.max = 1500, incerement=100,
                       distance.measure = "eder",
                       custom.graph.title = "Who is the author?",
                       write.png.file=T,
                       gui = TRUE)
```

```
## using current directory...
```

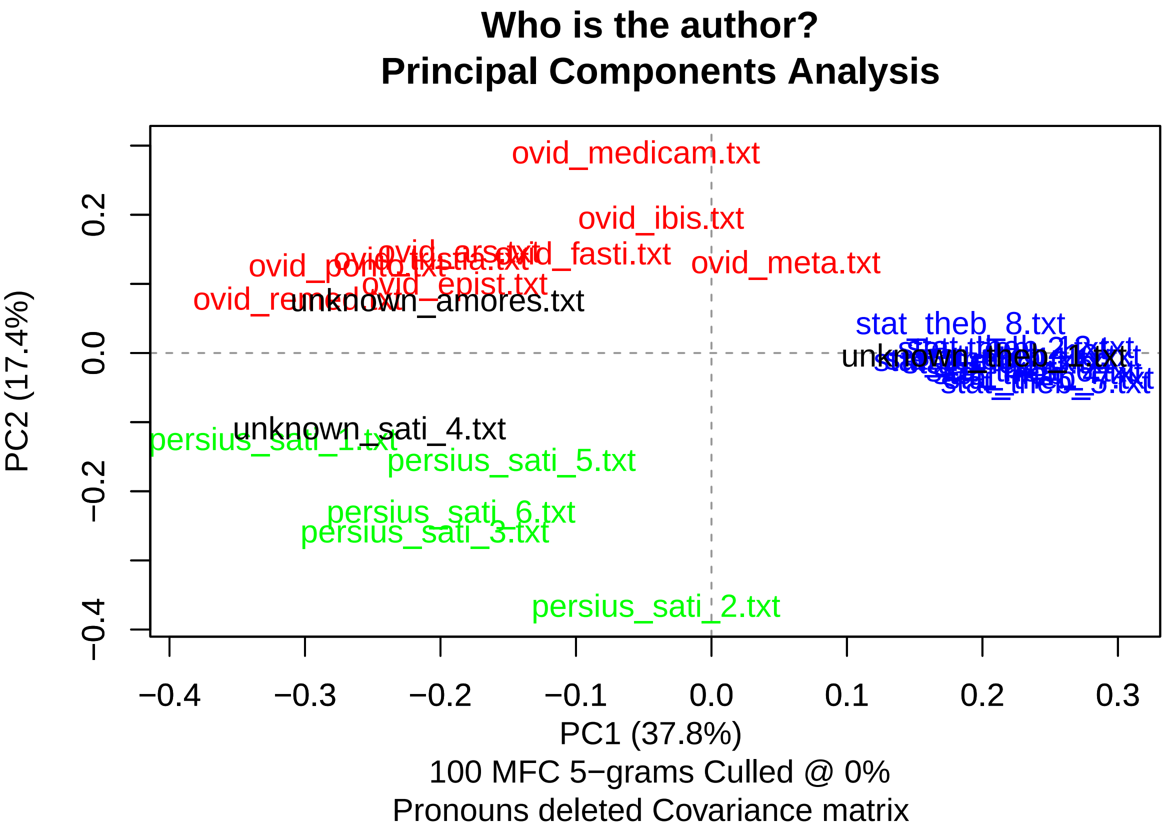
```
## Warning in delete.stop.words(table.with.all.freqs, pronouns): chosen stop words were not found in the
##   please check the language, lower/uppercase issues, etc.
```

```
##
```

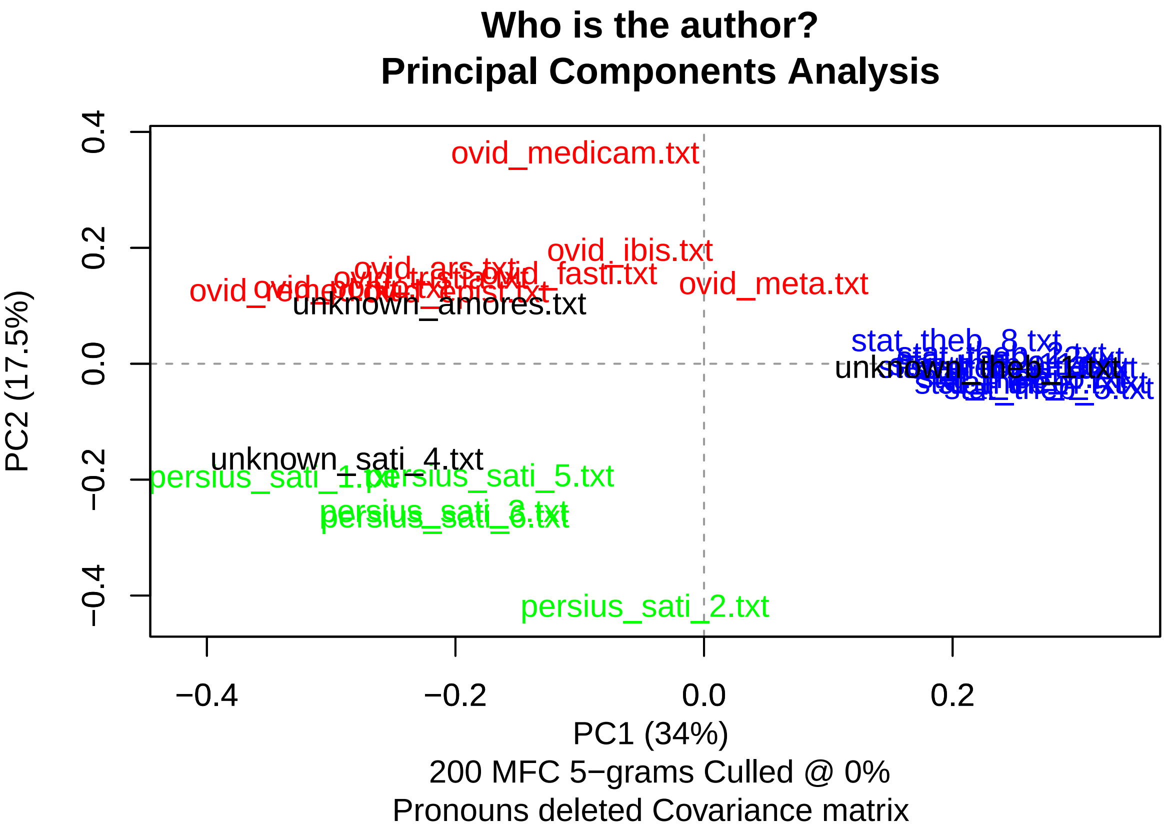
```
## culling @ 0  available features (words) 2000
```

```
## MFW used:
```

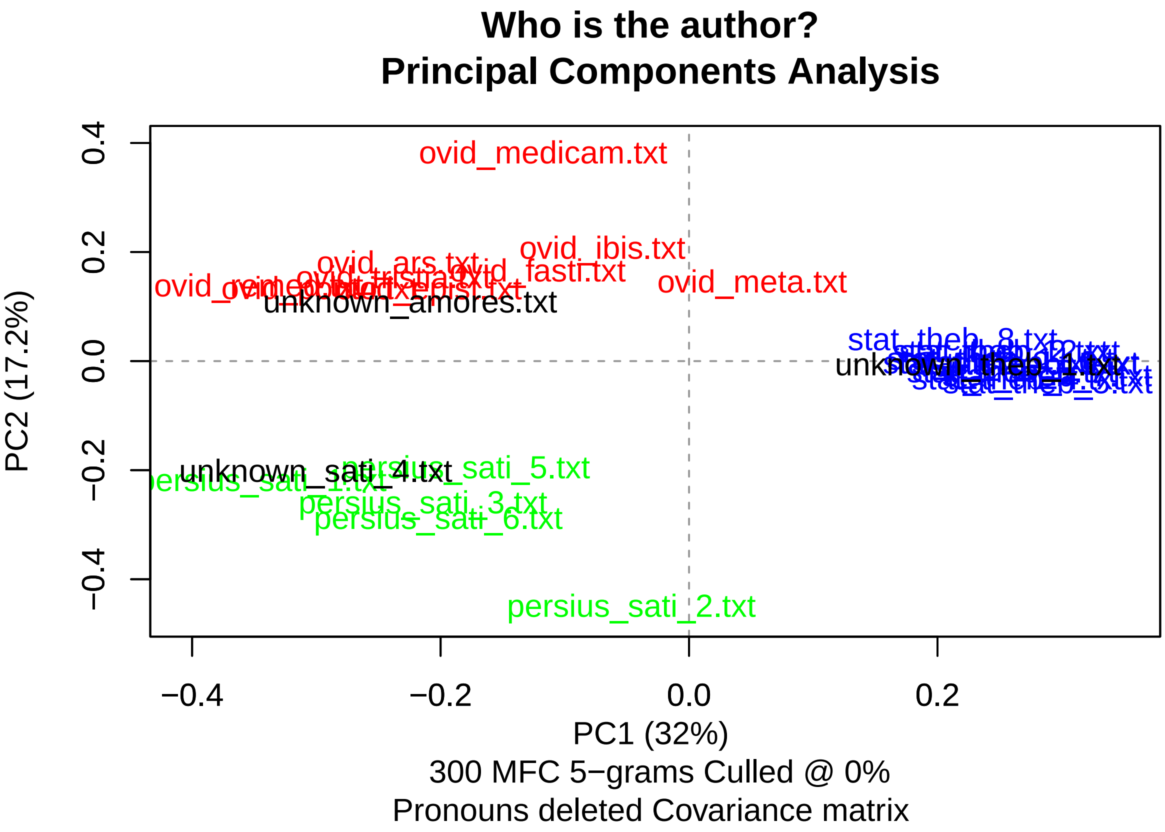
```
## 100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



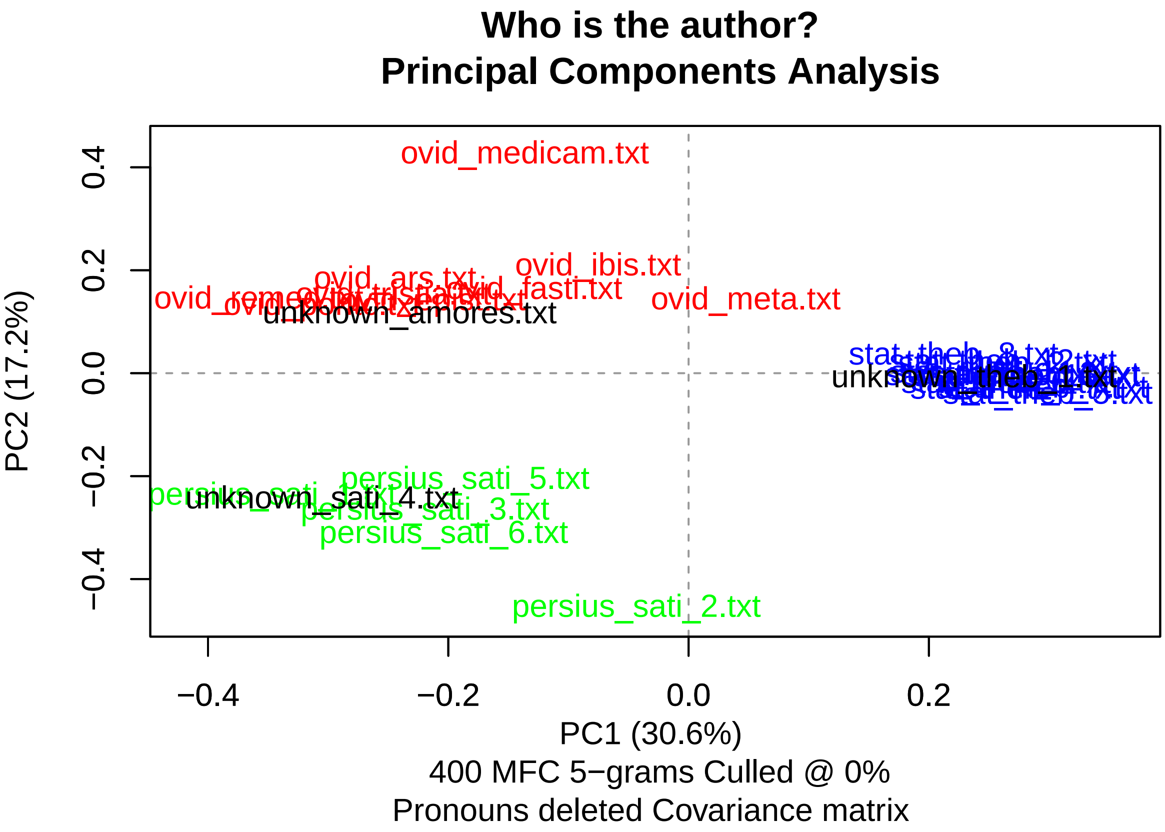
```
## 300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



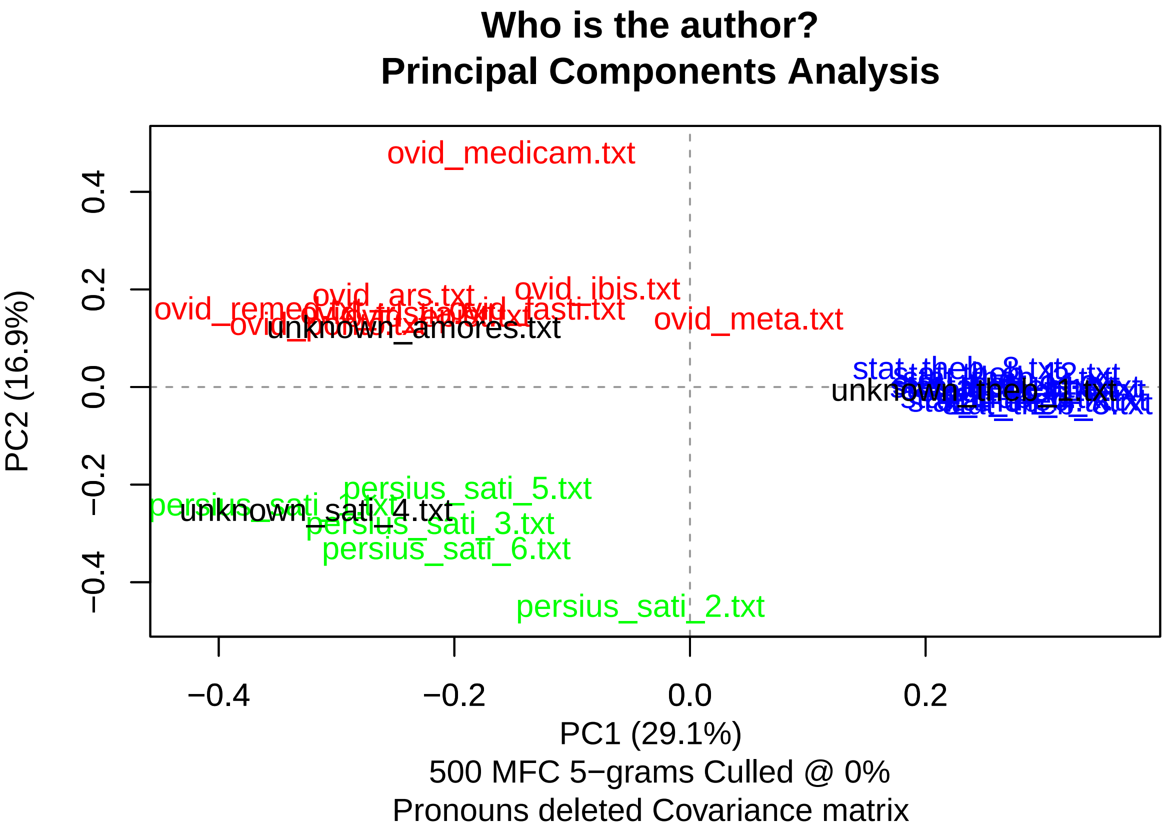
```
## 400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



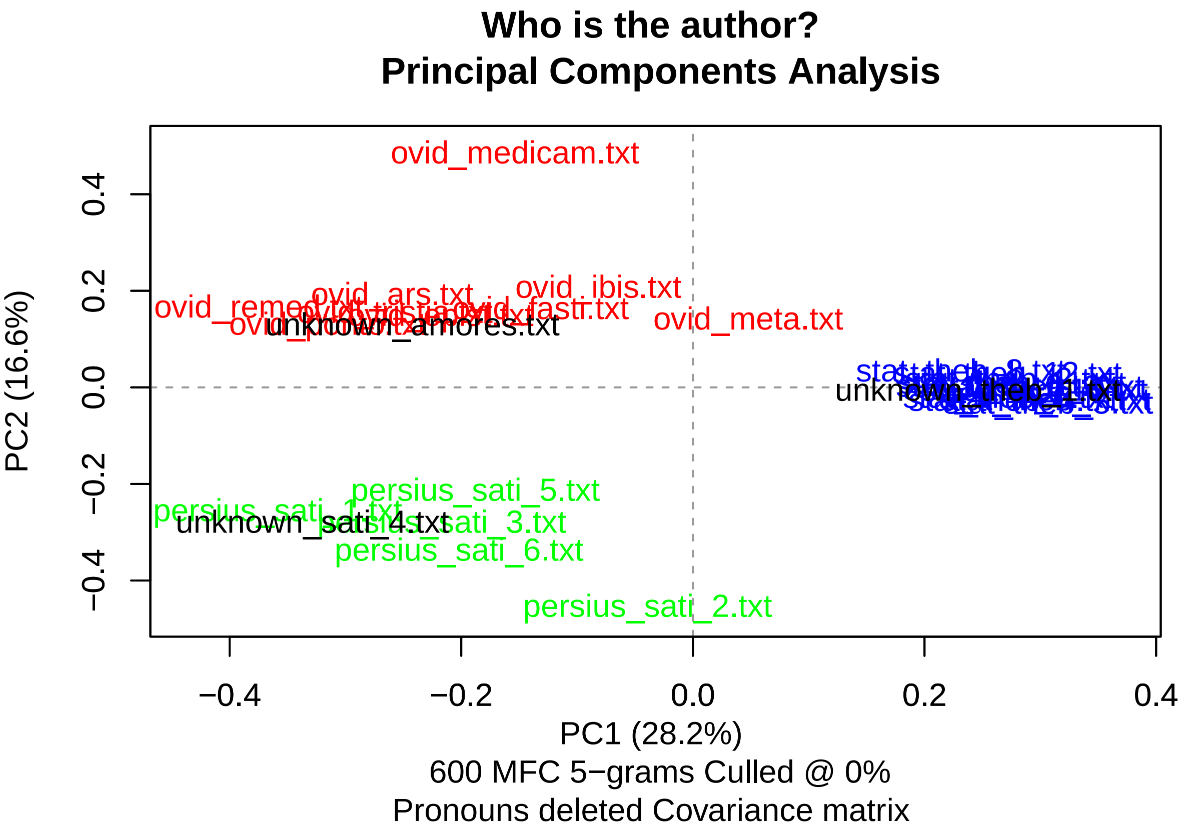
```
## 500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



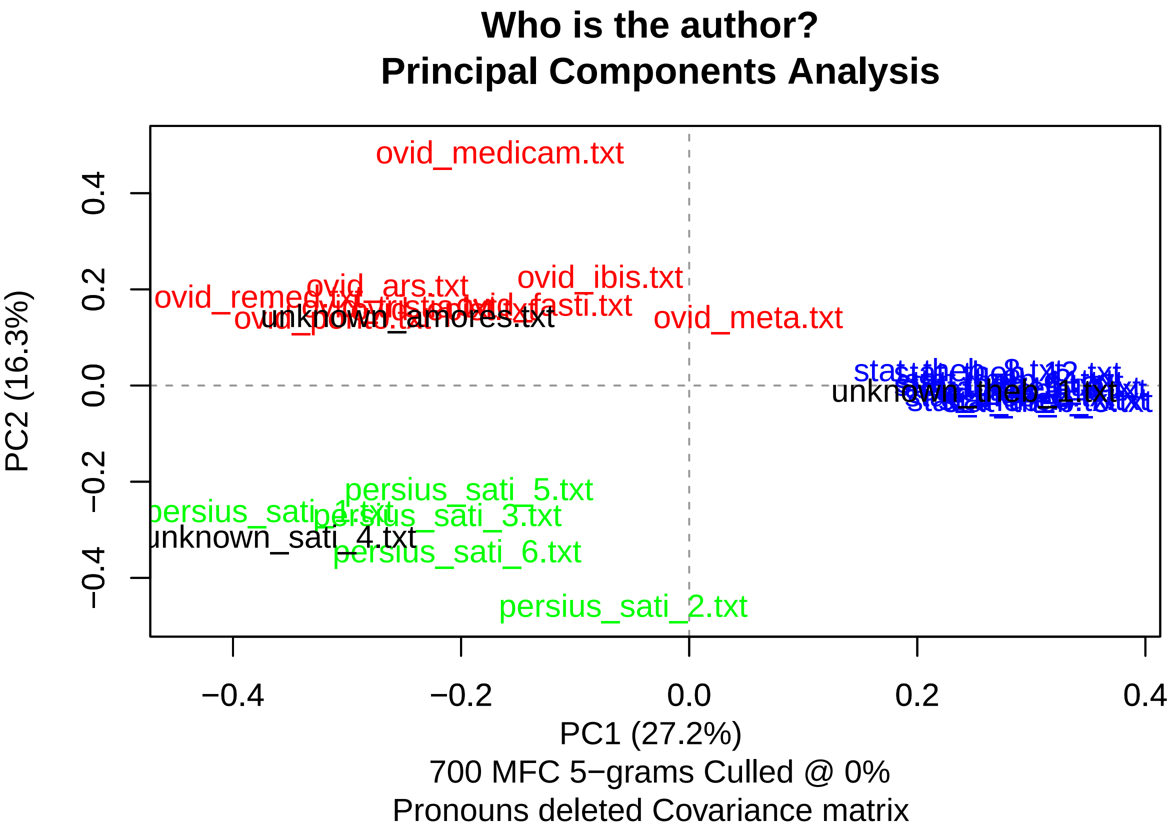
```
## 600
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



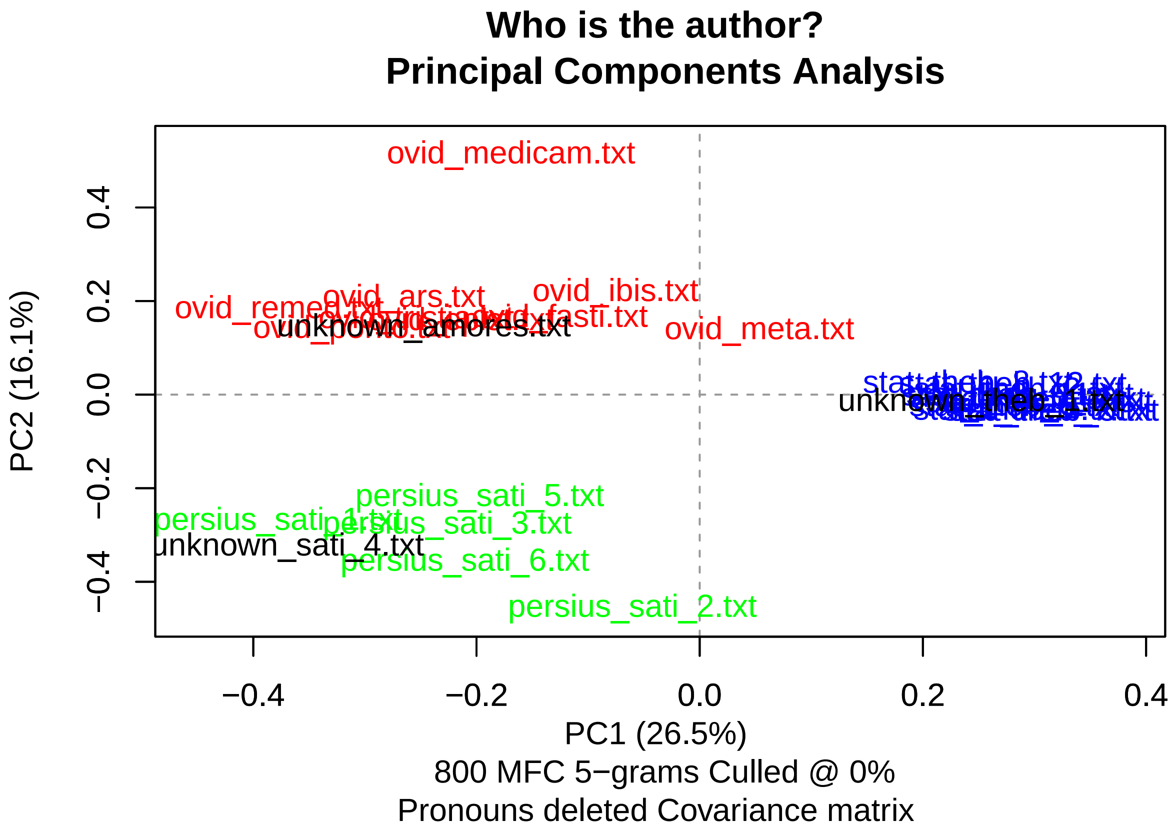
```
## 700
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



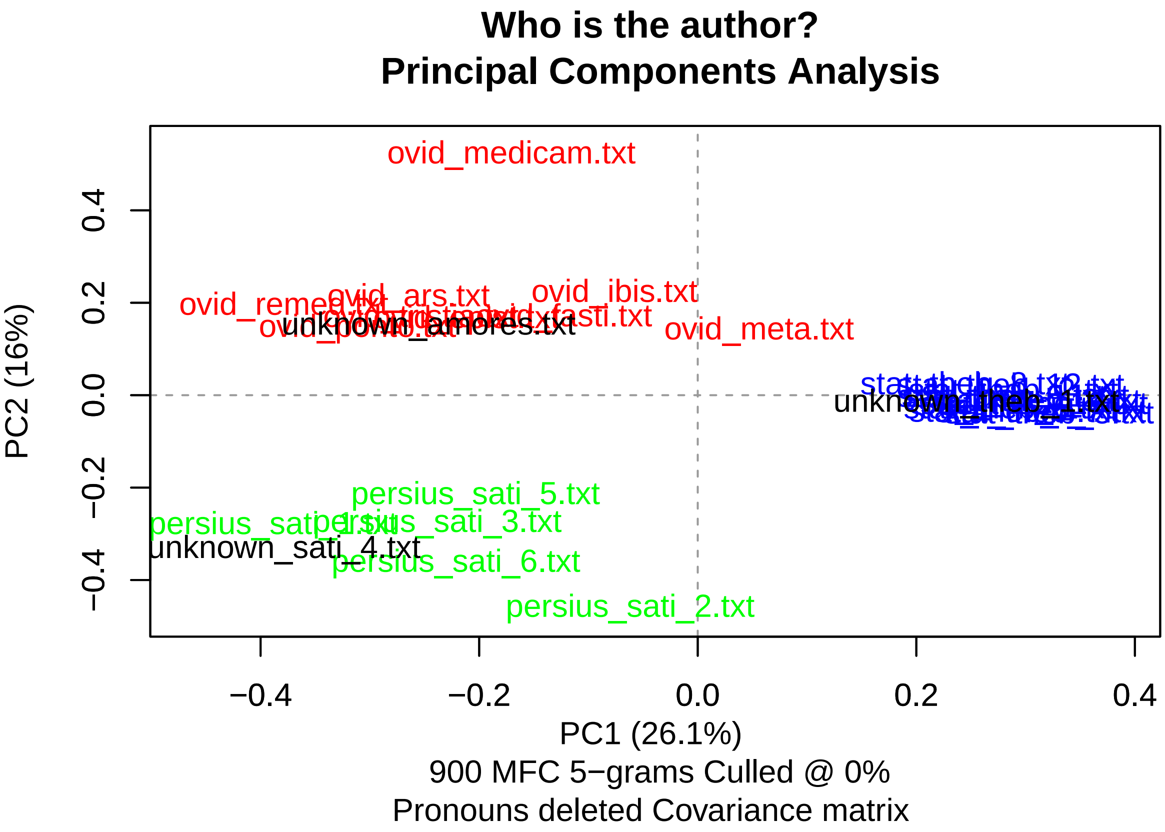
```
## 800
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



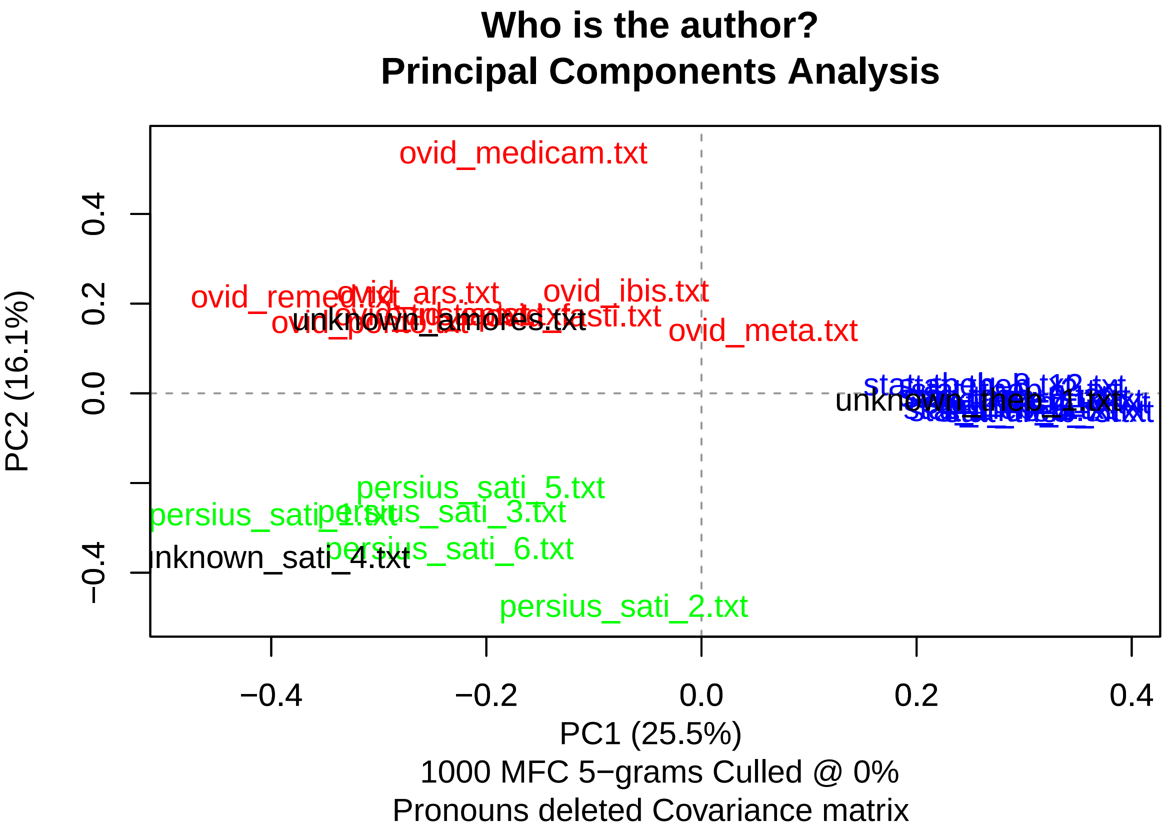
```
## 900
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



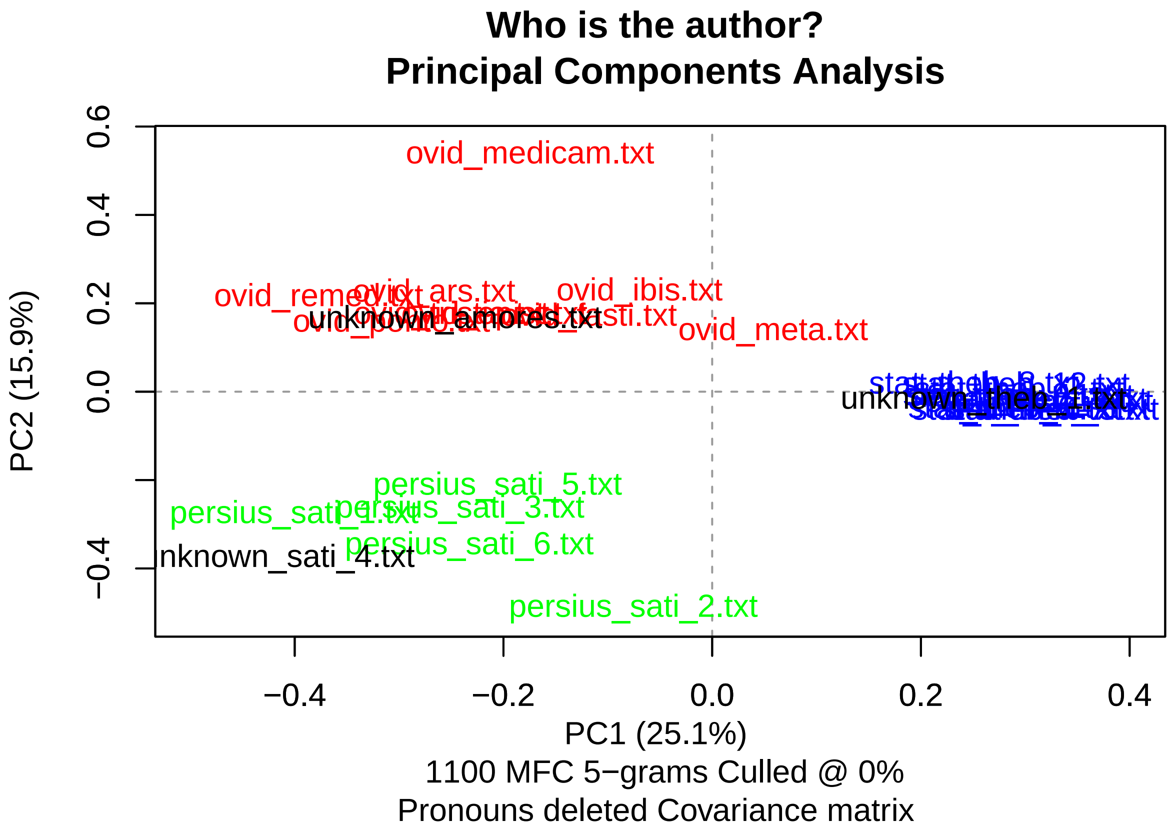
```
## 1000
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



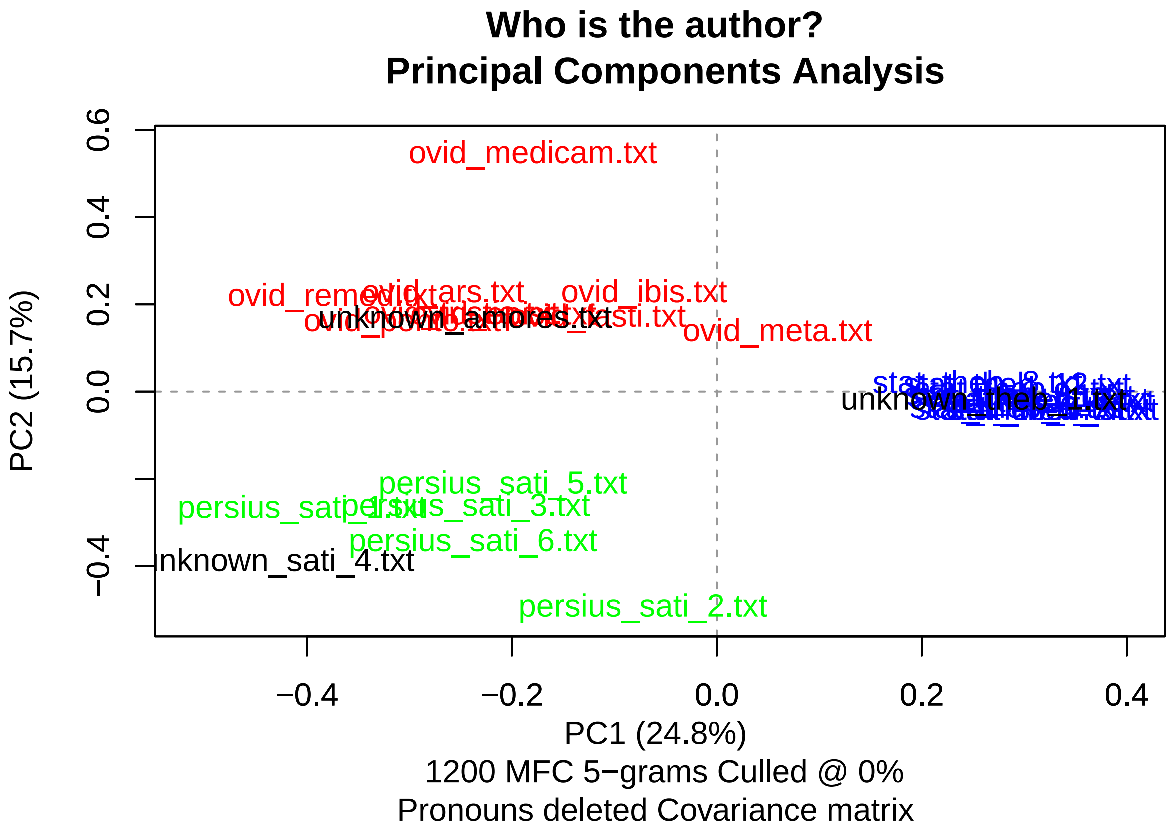
```
## 1100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



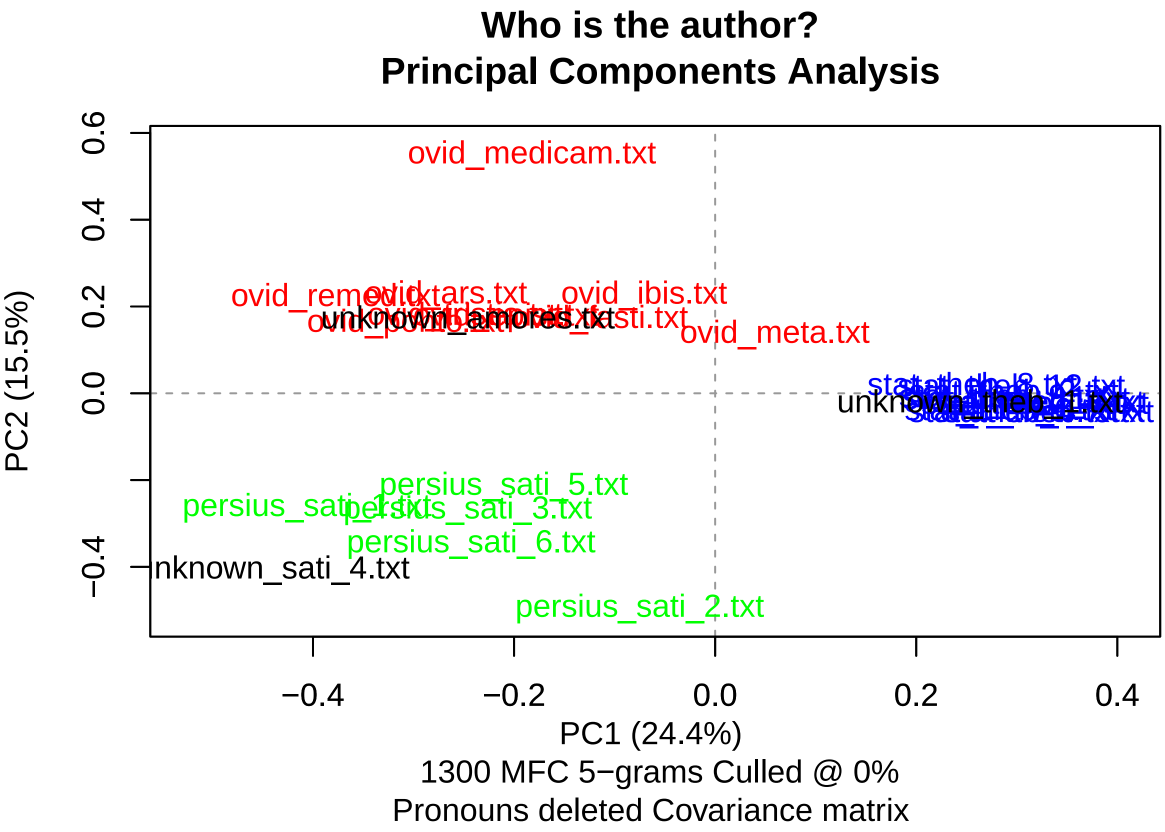
```
## 1200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



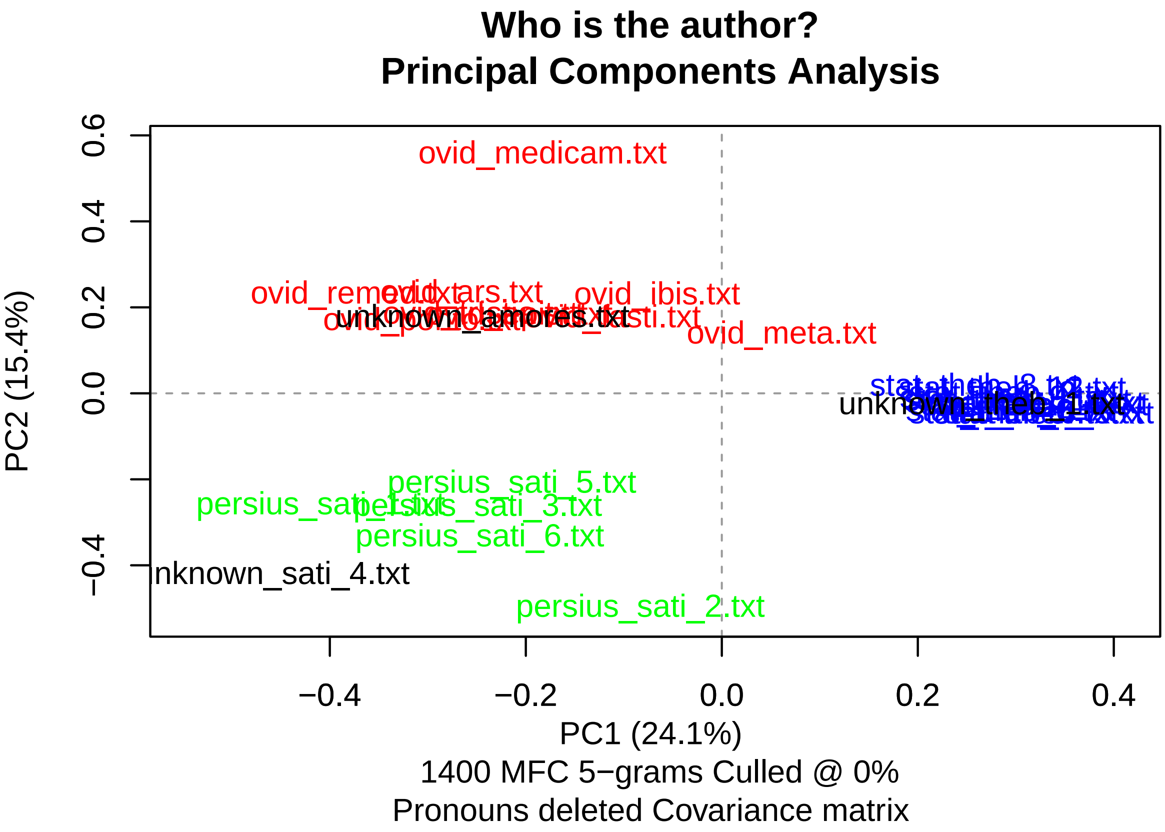
```
## 1300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



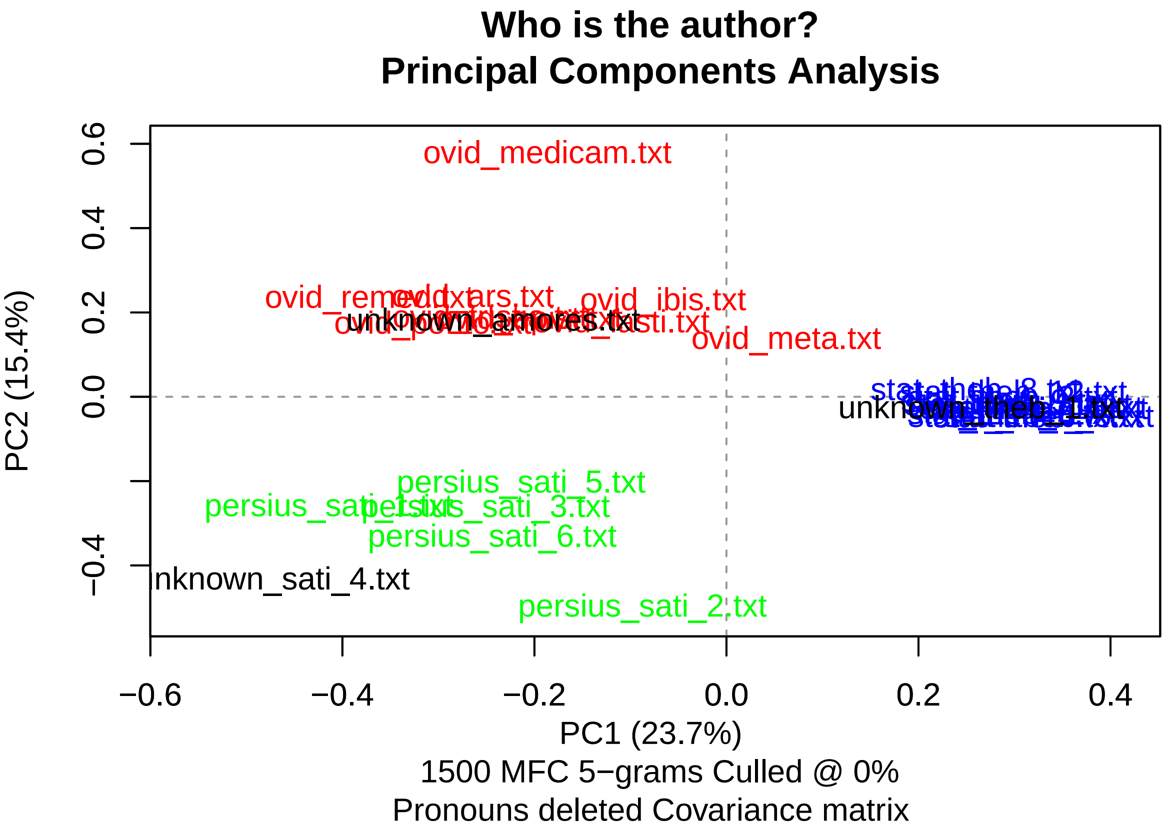
```
## 1400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



```
## 1500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
```



##



##

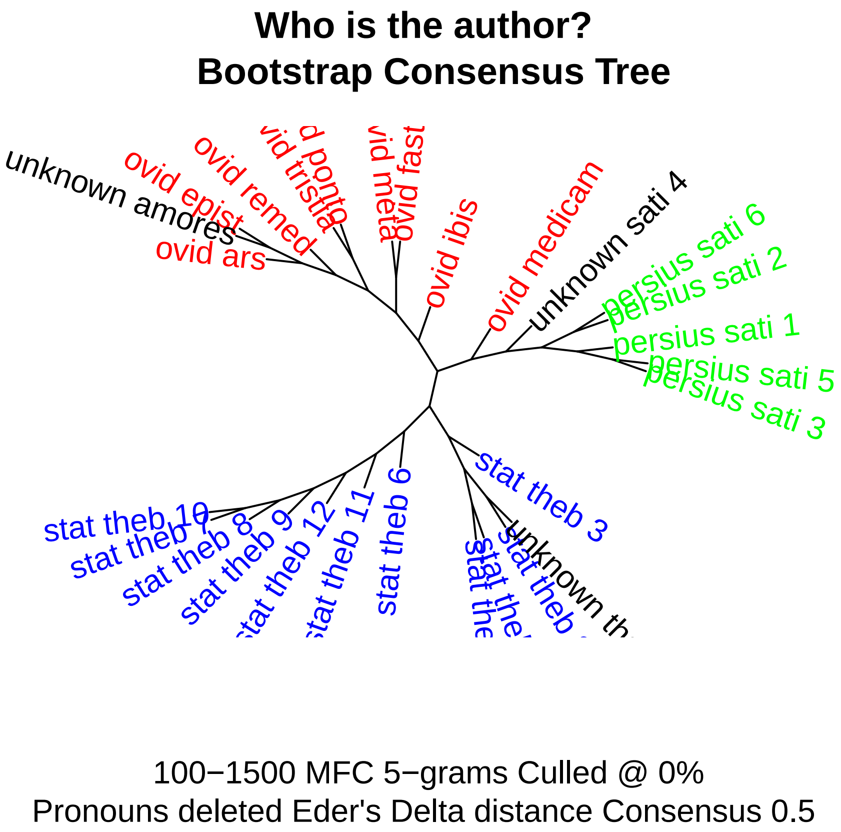
Apply Bootstrap Consensus Tree (MFCs 5grams)

```
# BCT 5grams - top 100-1500-100 MFCs 5grams - consensus strength 0.5
bct.results.5grams_100_1500_MFCS = stylo(corpus.dir = "validation_corpus/", frequencies = freqs.5grams,
                                         distance.measure="eder",
                                         analysis.type = "BCT",
                                         mfw.min = 100, mfw.max = 1500, increment = 100,
                                         custom.graph.title="Who is the author?",
                                         write.png.file=T,
                                         gui = T)
```

```
## using current directory...

## Warning in delete.stop.words(table.with.all.freqs, pronouns): chosen stop words were not found in the
##   please check the language, lower/uppercase issues, etc.

##

## culling @ 0  available features (words) 2000

## Calculating z-scores...

## Calculating Eder's Delta distances...

## MFW used:

## 100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 500
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 600
## Processing metadata...
##
```

```
##
## Assigning plot colors according to file names...
##
## 700
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 800
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 900
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1000
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1100
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1200
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1300
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1400
## Processing metadata...
##
##
## Assigning plot colors according to file names...
##
## 1500
## Processing metadata...
##
```

```
##
## Assigning plot colors according to file names...
##
##
```



Conclusions

From the methods their variants ran above, we can observe that all of the “unknown” texts have been successfully attributed to the “correct” author.

- `unknown0.txt` (i.e., Amores) to Ovid
- `unknown1.txt` (i.e., Thebaid book 1) to Statius
- `unknown2.txt` (i.e., Satire 4) to Persius

Especially the last case, *Satire* 4 by Persius is an interesting case. The length of the text is miniscule compared to the other texts in the corpus; it consists of only 342 tokens, thus the distance from the other texts. The other “not-so-tricky” case is the *Medicamina Faciei Femineae* by Ovid which has only 613 tokens (i.e., the second shortest text in our validation dataset).