GCalignR: An R package for aligning Gas-Chromatography data

Meinolf Ottensmann [1,*], Martin A. Stoffel [1], Joseph I. Hoffman [1],

1 Department of Animal Behaviour, Bielefeld University, Bielefeld

* meinolf.ottensmann@web.de

Abstract

Chemical signals are among the most fundamental and oldest means of animal communication. The desire to unravel broader patterns of chemical communication in birds and mammals paved the way for two not entirely new techniques, gas-chromatography and mass-spectrometry, in the fields of ecology and evolution. Comparing chemical profiles or chromatograms across many individuals yields some major obstacles as even the newest GC machines have an inherent error when measuring the retention times of chemical substances. Here we present GCalignR, an R package for the alignment of chromatography peaks among samples prior to hypothesis testing using multivariate statistics. GCalignR is specifically designed to be used by non-chemists by providing easy to use functions to check and align gas-chromatography data based on retention times. In addition, the package implements heatmaps and other plots to evaluate and potentially adjust the peak alignment. We hope that GCalignR will provide a tool that fits into a common biologist's workflow in R and that the package will contribute to the standardization and reproducibility of studies on chemical communication.

Introduction

Chemical cues are arguably the most common mode of communication among animals [1]. Patterns in complex chemical signatures can therefore yield information about phylogenetic relatedness [2], sexual maturation [3], kinship [4–6] and genetic quality [6–8]. One of the most common approaches for resolving the chemical composition of samples is gas-chromatography (GC), which can rapidly detect and quantify molecules within a sample to generate a characteristic chromatogram or chemical profile [9]. Although GC is relatively rapid and inexpensive, making it attractive for studies of non-model organisms, individual molecules are characterised according to their retention times, making them effectively anonymous. An additional mass-spectrometry step (GC-MS) can provide further details of the chemical composition of individual molecules, allowing them to be compared to existing databases where available.

GC provides a fast and effective means of resolving broad patterns of chemical similarity, but relies heavily on the correct alignment of homologous substances, represented by specific peaks, across samples. However, peak alignment is not necessarily straightforward as it is necessary to account for perturbations in retention times caused by subtle, random and often unavoidable experimental variation including changes in ambient temperature, flow rate of the carrier gas and column ageing [10,11].

Variation in peak intensities both within and among samples can also contribute   35
towards errors in characterising chemical profiles.   36

GC is widely used in behavioural and ecological studies of (usually) non model   37
organisms, such as birds, mammals and insects, where the goal is often to investigate   38
broad chemical patterns. These studies tend to use GC-MS less often than studies of   39
humans and other model organisms, both because GC-MS is relatively expensive and   40
because many of the chemicals that are resolved often reveal limited homology to   41
currently available databases. However, aligning GC data is non-trivial due to the   42
anonymous nature of the many peaks. Consequently, although a number of programs   43
are available for aligning GC-MS data, which make use of mass spectrograms, we are   44
only aware of a single program that can handle GC data [12]. As a result, most studies   45
of mammalian and avian chemical communication have relied on manual alignment and   46
peak calling [13], which is time-consuming, particularly for large samples of individuals,   47
can be biased and subjective, and is not strictly reproducible.   48

Here, we introduce `GCalignR`, an R package that implements a simple algorithm to   49
align peaks based on retention time data obtained by GC and provides sophisticated   50
visualisations for the evaluation of alignment quality. First of all, the `check_input`   51
function is used to ensure that the data are formatted correctly (Fig 1, step 1). Second,   52
the `align__chromatograms` function is used to align the data (Fig 1, step 2) as follows: 53
(i) systematic shifts of chromatograms are corrected by applying appropriate linear   54
shifts to whole chromatograms based on a single reference sample; (ii) retention times of   55
individual peaks are grouped iteratively together with homologous peaks of other   56
samples and aligned within the same row in a retention time matrix; and (iii) rows with   57
similar retention times are merged where appropriate. Third, diagnostic plots allow the   58
resulting alignments to be visually inspected (Fig 1, step 3), thereby facilitating   59
optional pre-processing or re-alignment of the data (Fig 1, step 4). Finally, to   60
compensate for differences in total chemical concentrations among samples, measures of   61
peak abundance (e.g. peak area or peak height) can be normalised using the function   62
`norm_peaks` (Fig 1, step 5).   63

Fig 1. GCalignR workflow.
In addition to the alignment of substances across samples, the package provides
functions for checking and inspecting the data. The aligned data are ready to use for
analyses in conjunction with other packages. Each function is explained within the main
text.

Implementing GC alignment and checking within R brings several advantages over   64
currently available stand-alone programs. First, the code is open source, which   65
facilitates flexibility and transparency in data analysis. Second, all computational steps   66
can be integrated into R Markdown documents [14], thereby enhancing reproducibility.   67
Finally, our package provides a seamless transition from the processing of the peak data   68
through to downstream analysis within other widely used R packages for multivariate   69
analysis, e.g. vegan [15].   70

The package   71

GCalignR contains functions to align peaks from GC and GC-MS data based on   72
retention times and evaluate the resulting alignments. The main aim of the package is   73
to provide a simple tool that guides the user through the alignment of large datasets   74
prior to the statistical analysis of multivariate chemical data. A typical workflow for the   75
analysis of chemical signatures in GCalignR is shown in Fig 1 and described below. The   76
package vignette provides a detailed description of all of the functions and their   77

arguments and can be accessed via `browseVignettes("GCalignR")` after the package has been installed. The workflow is described below and the standard input format of GCalignR is a tab-delimited text file, as illustrated in the vignette.

Example dataset

The functionality of GCalignR is illustrated using GC data from skin swabs of 41 Antarctic fur seal *Arctocephalus gazella* mother-pup pairs from two neighbouring breeding colonies at South Georgia in the South Atlantic [6]. The chemical data associated with these samples are provided in the file `peak_data.txt`, which is distributed with the package. Additional data on colony membership and age-class are provided in the data frame `peak_factors.RData`.

Alignment of GC peaks among samples

As outlined briefly above, the core algorithm in `align_chromatograms` (see Table 1 for a list of parameters) implements three consecutive manipulations of the peak data (Fig 2 i-iii) together with two further optional manipulations (Fig 2 iv and v). These are described in detail below.

(i) Linear adjustments of chromatograms

First, all peaks within a chromatogram are shifted with respect to a reference chromatogram to account for systematic shifts in retention times among homologous chemicals shared among samples (Fig 2 i). This procedure is implemented for all of the samples in such a way that the number of shared peaks is maximised. The parameter `max_linear_shift` defines the maximum range of linear shifts that are considered by the function. This approach clearly relies on their being a sufficient number of substances shared among the samples. In the absence of shared substances, the function will be unlikely to find a suitable shift and consequently the chromatograms will remain untransformed. By default, a reference is selected automatically by searching for the sample with the highest average similarity to all other samples based on the number of shared peaks prior to alignment. Optionally, the alignment can be implemented using an internal standard (labelled "reference") containing substances that are known *a priori* to occur in most or all of the samples.

Fig 2. Overview of the alignment algorithm implemented in GCalignR using a hypothetical dataset. Within each matrix, rows correspond to substances and columns correspond to samples. Consecutive manipulations of the matrices are shown from left to right. Zeros indicate absence of peaks and are therefore not considered in computations. **i**. Chromatograms are linearly shifted with respect to a reference (here S2). Linear adjustments in this example are conducted based on two peaks coloured in blue and red. Colouring of cells in **ii-v** refer to the substance identity in the final alignment and black rectangles indicate conflicts that are solved by manipulations of the matrices. **ii**. Peaks are aligned row by row. Initially, always the second sample is compared to the first. Then the next sample is compared to all of the samples in previous columns until the last column is reached. **iii**. If merging does not result in the loss of any data, rows are merged. **iv**. If specified, all peaks found in one or more blanks (negative controls) are removed as well as the blank itself. **v**. Optionally, unique peaks present in a single sample can be removed as well.

Table 1. Parameters for the function `align_chromatograms`

Paramter	Description
blanks	Character vector containing the names of negative control samples (blanks) that are used to identify and remove contaminants
data	Path to a tab-delimited text file list containing the chemical data. See the vignette for an example and alternative input formats
delete_single_peak	Logical that implements the optional functionality to remove unique substances from the aligned dataset
max_diff_peak2mean	Numeric value defining the allowed deviation of the retention time of a focal peak from the mean of the corresponding row during peak alignment
max_linear_shift	Numeric value (in minutes) that defines the range that is considered for the adjustment of linear shifts in peak retention times among samples
min_diff_peak2peak	Numeric values defining the expected minimum difference in retention times among substances. Rows that are more similar than the threshold value will be merged, if no conflict emerges due to the presence of peaks in more than one row within a single sample.
rt_cutoff_low	Threshold value defining the minimum retention time in minutes. All peaks with retention times below the threshold value will be removed from the chemical dataset prior to alignment
rt_cutoff_high	Threshold value defining the maximum retention time in minutes. All peaks with retention times below the threshold value will be removed from the chemical dataset prior to alignment
rt_col_name	Name of the variable containing retention times of peaks. The name needs to correspond to a variable included in the chemical data
reference	Name of a sample that will be used as reference to adjust linear shifts in peak retention times across samples. By default, a reference is automatically selected
sep	Field separator character. See `?read.table` for a list of separators
write_output	Character vector of variable names that correspond to variables included in the chemical dataset. If specified, aligned datasets are exported as tab-delimited text files for each of the variables

(ii) Peak alignment

Individual peaks are aligned across samples by comparing the peak retention times of each sample consecutively with the mean of all previous samples (Fig 2 ii). If the focal cell within the matrix contains a retention time that is larger than the mean retention time of all previous cells within the same row plus a user-defined threshold (Eq (1)), that cell is moved to the next row.

$$rt_m > \left(\frac{\sum_{i=1}^{m-1} rt_i}{m-1} \right) + \text{max\_diff\_peak2mean} \tag{1}$$

where rt = retention time; m = focal cell and max_diff_peak2mean defines the [113] user-defined threshold deviation from the mean retention time. [114]

If the focal cell contains a retention time that is smaller than the mean retention [115] time of all previous cells within the same row minus a user-defined threshold (Eq (2)), [116] all previous retention times are then moved to the next row. [117]

$$rt_m < \left(\frac{\sum_{i=1}^{m-1} rt_i}{m-1} \right) - \text{max\_diff\_peak2mean} \tag{2}$$

After the last retention time of a row has been evaluated, this procedure is repeated [118] for the next row until the end of the retention time matrix is reached. [119]

(iii) Merging rows [120]

Occasionally, due to minor variation in retention times, homologous peaks can be sorted [121] into different, but adjacent, rows in different samples. However, this results in a clear [122] pattern whereby some of the samples will have a retention time in one of the rows while [123] the other samples will have a retention time in an adjacent row. Consequently, the [124] function merges adjacent rows when this does not cause any loss of information (i.e. no [125] sample exists that contains substances in both rows, (Fig 2 iii). Again, the user can [126] define the threshold for the minimal difference in the retention time between two [127] mergeable peaks with `min_diff_peak2peak`. [128]

(iv) Removal of contaminants [129]

After aligning peaks, the package offers several optional post-processing steps for [130] cleaning up the data. First of all, negative control samples, if available, can be used to [131] remove potential contaminants, including unwanted chemical substances in laboratory [132] reagents or within the gas chromatography column. Chemical data for negative controls [133] can be included in the input file and, by specifying these samples as blanks, [134] `align_chromatograms` will remove all substances present in the controls from the [135] aligned dataset (Fig 2 iv). [136]

(v) Removal of single peaks [137]

Frequently, substances occur only within a single sample. For comparative approaches [138] based on similarity matrices, these substances are often not informative and can be [139] removed from the dataset. `align_chromatograms` implements the removal of unique [140] substances (Fig 2 v) when the `delete_single_peak` argument is set to `TRUE`. [141]

Workflow [142]

Here, we demonstrate a typical workflow in GCalignR using the fur seal dataset as an [143] example. All of the alignment steps described above are implemented within the [144] function align_chromatograms. A list of user-defined parameters and their descriptions [145] can be accessed from the documentation in the helpfile by typing [146] `?align_chromatograms`. Prior to peak alignment, the `check_input` function [147]

interrogates the input file for typical formatting errors and missing data. We encourage 148
the use of unique names for samples consisting only of letters, numbers and underscores. 149
If the data fail to pass this quality test, indicative warnings will be returned to assist in 150
error correction. As this function is executed internally prior to any alignment, the data 151
need to pass this check before the alignment can begin. 152

```
library(GCalignR) # loads the package                          153
fpath <- system.file(dir = "extdata",                          154
                     file = "peak_data.txt",                    155
                     package = "GCalignR")                      156
check_input(fpath) # checks the data                           157
```

In order to begin the alignment procedure, the following code needs to be executed: 158

```
aligned_peak_data <- align_chromatograms(data = peak_data,     159
       rt_col_name = "time",                                   160
       max_diff_peak2mean = 0.02,                              161
       min_diff_peak2peak = 0.08,                              162
       max_linear_shift = 0.05,                                163
       delete_single_peak = TRUE,                              164
       blanks = c("C2","C3"))                                  165
```

Afterwards, a summary of the alignment process can be retrieved using the printing 166
method, which summarises the function call including defaults that were not altered by 167
the user. This provides all of the relevant information to retrace every step of the 168
alignment procedure. 169

```
print(aligned_peak_data) # Verbal summary of the alignment     170
```

As alignment quality may vary with the parameter values selected by the user, the 171
plot function can be used to output four diagnostic plots. These allow the user to 172
explore how the parameter values affect the resulting alignment and can help flag issues 173
with the raw data. 174

```
plot(aligned_peak_data) # Creates Fig. 3                       175
```

Fig 3 a shows the distribution of peak numbers across samples both before and after 176
the alignment. In this example dataset, we specified two negative controls (blanks): 177
"C2" and "C3". After the removal of substances shared with the controls as well as 178
unique substances, the number of post-alignment peaks is significantly reduced across 179
samples. For details of the number of removed substances, type 180
`print(aligned_peak_data)`. Fig 3 b shows a histogram of linear shifts of entire 181
samples implemented by the function `align_chromatograms`. The majority of samples 182
were not shifted at all, whereas a small number were shifted between -0.01 and 0.01 183
minutes. Fig 3 c shows a histogram of the amount of variation among the samples in 184
the retention times of single substances (defined as retention times that were aligned 185
based on user-defined criteria). In this example, the distribution shows a left-skew, 186
indicating that the majority of substances vary less than 0.05 minutes. Finally, Fig 3 d 187
shows the extent of sharing of substances across samples. This shows a typical pattern 188
whereby most of the substances are found in a small number of samples but there a 189
number of substances are also present in most of the samples. 190

Fig 3. Diagnostic plots summarise aligned datasets.

Additionally, the full alignment can be visualised using a heat map with the function 191
`gc_heatmap`. 192

```
gc_heatmap(aligned_peak_data, type = "binary", threshold = 0.05)   193
```

The resulting heatmap for the example dataset (Fig 4) shows all of the
post-alignment peaks across all samples and substances. Filled cells represent peaks
whereas empty cells indicate the absence of a given substance within a sample.
Substances that deviate by less than a defined threshold value (in this case, 0.05
minutes) from the mean retention time across all samples are shown in light blue. Red
cells indicate a deviation that is larger than 0.05 minutes and thereby flag potentially
problematic alignments. For the example dataset, only a scattering of larger deviations
are observed and these do not appear to be clustered within samples or substances.

**Fig 4. Heatmaps allow to inspect the distribution of substances across
samples.**
Heatmaps show the variability of the retention times. Substances on the horizontal axis
are ordered by retention times, increasing from left to right. Parameters of the function
are explained in the corresponding helpfile and the package vignette.

Peak normalisation

In order to account for differences in the total concentration of samples, we provide an
additional function `normalise_peaks` that can be used to normalise peak abundances.
The abundance measure (e.g. peak area) needs to be specified as `conc_col_name` in the
function call. By default, the output is returned in the format of a data frame.

```
scent <- norm_peaks(data = aligned_peak_data,
                    rt_col_name = "time",
                    conc_col_name = "area",
                    out = "data.frame" )
```

Downstream analyses

The output of GCalignR is compatible with other functionalities in R, thereby providing
a seamless transition between packages. For instance, multivariate analyses can be
conduced using the package vegan. To visualise patterns of chemical similarity within
the fur seal dataset in relation to breeding colony membership, we implemented
non-metric-multidimensional scaling (NMDS) based on a Bray-Curtis dissimilarity
matrix in vegan:

```
# sort data frame by row names
scent <- scent[match(row.names(peak_factors),
                     row.names(scent)),] # sort data
# Logarithmic transformation
scent <- log(scent + 1)
# NMDS
scent_nmds <- vegan::metaMDS(comm = scent, distance = "bray")
scent_nmds <- as.data.frame(scent_nmds[["points"]])
scent_nmds <- cbind(scent_nmds,
                    colony = peak_factors[["colony"]])
```

The results of the NMDS are stored in the data frame `scent_nmds` and can be
visualised using ggplot2 [16].

```
library(ggplot2) # load package ggplot2
ggplot(data = scent_nmds, aes(MDS1,MDS2, color = colony)) +
    geom_point(size = 2) +
    theme_void() +
```

```
scale_color_manual(values = c("blue","red"),                    234
                name="Breeding_colony",                          235
                breaks=c("SSB", "FWB"),                          236
                labels=c("Special_study_beach",                  237
                    "Freshwater_beach")) +                       238
theme(panel.background = element_rect(colour = "black",          239
                    size = 1.5, fill = NA),                      240
        aspect.ratio = 1,legend.justification=c(0,0),            241
        legend.position=c(0.01,0.01),                            242
        legend.title = element_text(size=12, face="bold"),       243
        legend.background = element_rect(color = "black",        244
                    fill = NA, size = 1,                         245
                    linetype = "solid"))                         246
```

Fig 5 reveals a clear pattern in which seals from the two colonies cluster separately   247
based on their chemical profiles, as shown also by Stoffel et al. (2015). Although a   248
sufficient number of standards were lacking to calculate the internal error rate (as   249
shown below for the bumblebee datasets), the strength of this pattern, together with   250
the rarity of substances invoking large deviations from the mean retention time (Fig 4), 251
suggests that the alignment implemented by GCalignR is of high quality.   252

Fig 5. A NMDS plot shows the similarity of individuals within colonies.
Individuals are colour coded based on the breeding colony.

To explore this further, we introduced random errors into the raw dataset, re-aligned   253
the resulting datasets, and then quantified the strength of clustering by colony using the 254
function adonis from the package vegan, which performs a permutation-based   255
multivariate analysis of variance ("permutational manova" [17]). The introduced errors 256
followed a Gaussian profile with -0.02 to 0.02. For each error rate value, defined as the 257
proportion of peaks within each sample with random errors, we generated 10 datasets.   258

Validation based on error rates of known substances and strength of effects

Introducing errors into the fur seal dataset affected both the total number of scored   261
substances as well as the strength of the detected pattern (Fig 6). The total number of 262
substances scored gradually increased with increasing error rate (Fig 6 a) as higher   263
variation in retention times caused peaks to be split into more than one substance. In   264
parallel, Adonis R2 values fell linearly up to an error rate of 0.5 before levelling off at 265
around 0.07 (say something about significance, still highly significant!). This suggests 266
both that the original dataset is clearly structured and that the results of the alignment 267
procedure within GCalignR are relatively robust to low to moderate (i.e. ¡ 0.2) rates of 268
peak calling error.   269

Fig 6. Influence of additional noise on the detectability of patterns.
Random errors following a Gaussian profile were added to the raw chemical dataset on
the total number of scored substances. The strength of the colony effect was determined
using the function adonis of the vegan package.

To further assess the performance of GCalignR, we calculated alignment error rates   270
based on three previously published bumblebee datasets comprising known substances   271
identified using GC-MS [12]. The first dataset comprises 24 *Bombus bimaculatus*   272
individuals characterised for 32 substances (total = 717 retention times). The second   273
comprises 20 *B. ephippiatus* individuals characterised for 42 substances (total = 782   274

retention times) and the third comprises 11 *B. flavifrons* individuals characterised for 44   275
substances (total = 457 retention times). We calculated the error rate as the ratio   276
between incorrectly assigned retention times and the total number of retention times   277
(Eq (1)).   278

$$\text{Error} = \left[\frac{\text{N\_missaligned}}{\text{N\_total}} \right] \tag{3}$$

where N denotes the number of retention times defined as being misaligned (i.e. not   279
assigned to the row that defines the mode of a given substance). By systematically   280
changing the two parameters `max_diff_peak2mean` and `min_diff_peak2peak`, we   281
explored 100 parameter combinations to explore how parameter values affect the   282
alignment accuracy. All three datasets show generally low error rates, typically between   283
3–5% for most parameter combinations, although low values of the parameter   284
`min_diff_peak2peak` tend to be associated with somewhat higher error rates, especially   285
when `max_diff_peak2mean` is also low (Fig 7).   286

Fig 7. Effects of alignment parameters on error rates
The three datasets [12] are comprised of known substances. Each point shows the error
in aligning substances for a combination of `max_diff_peak2mean` and
`min_diff_peak2peak`.

Additionally, we simulated the effect of noise by introducing errors at random into   287
each of the datasets as described for the fur seal dataset above. As expected, error rates   288
were initially low for all three species but increased with progressively increasing levels   289
of noise (Fig 8). *B. bimaculatus* and *B. flavifrons* both showed approximately linear   290
responses whereas *B. ephippiatus* began to level off when exceeding an additional noise   291
level of around 0.7. This probably reflects the fact that chemical diversity is lower in *B.*   292
ephippiatus (stats).   293

**Fig 8. Additional noise in peak retention times increases the error rate
substantially.**
Therefore, optimal alignments require clearly resolved peaks that need to be extracted
prior to using GCalignR.

Final remarks   294

GcalignR is primarily intended as a pre-processing tool in the analysis of complex   295
chemical signatures of animals where overall patterns of chemical similarity are of   296
interest as opposed to specific (i.e. known) chemicals. We have therefore prioritised an   297
objective and fast alignment procedure that is not claimed to be free of error. However,   298
error rate calculations suggest that the algorithm performs well, at least for most   299
parameter combinations, while realistically low levels of noise appear to have a modest   300
effect on the resulting alignments. Importantly, GCaligner also implements a suite of   301
diagnostic plots that allow the user to visualise the influence of parameter settings on   302
the resulting alignments, allowing fine-tuning of both the pre-processing and alignment   303
steps (Fig 1).   304

Availability   305

The current stable version requires at least R 3.2.5 and is available on CRAN.   306

```
install.packages("GCalignR")
```
   307

We aim to extend the functionalities of GCalignR in future and the developmental   308
version can be downloaded from GitHub within R using devtools [18].   309

```
library(devtools)
install_github("mastoffel/GCalignR", build_vignettes = TRUE)
```
  310
  311

The raw fur seal chemical dataset is included in this R package and the bumblebee   312
datasets [12] can be downloaded here `http://onlinelibrary.wiley.com/store/10.`   313
`1002/jssc.201300388/asset/supinfo/jssc3437-sup-0001-TableS1.zip?v=1&s=`   314
`57d5d58273d1d4207e70c72cecd5bba4b1fe95a1`.   315

Supporting information   316

S1 File. Code used for simulations. The code used to generate these results and   317
accompanying documentation are provided in a single PDF file written in Rmarkdown.   318

S2 File. Files used to generate the results presented in this manuscript.   319
This is a compressed zip archive that includes all the data files that have been used to   320
for the simulations presented in this manuscript along with the accompanying   321
documentation written in Rmarkdown   322

Acknowledgments   323

References

1. Wyatt TD. Pheromones and animal behavior: chemical signals and signatures. Cambridge University Press; 2014.

2. de Meulemeester T, Gerbaux P, Boulvin M, Coppée A, Rasmont P. A simplified protocol for bumble bee species identification by cephalic secretion analysis. Insectes Sociaux. 2011;58(2):227–236. doi:10.1007/s00040-011-0146-1.

3. Caspers BA, Schroeder FC, Franke S, Voigt CC. Scents of adolescence: the maturation of the olfactory phenotype in a free-ranging mammal. PloS one. 2011;6(6):e21162.

4. Bonadonna F, Sanz-Aguilar A. Kin recognition and inbreeding avoidance in wild birds: The first evidence for individual kin-related odour recognition. Animal Behaviour. 2012;84(3):509–513. doi:10.1016/j.anbehav.2012.06.014.

5. Krause ET, Krüger O, Kohlmeier P, Caspers BA. Olfactory kin recognition in a songbird. Biology letters. 2012;8(3):327–329.

6. Stoffel MA, Caspers BA, Forcada J, Giannakara A, Baier M, Eberhart-Phillips L, et al. Chemical fingerprints encode mother–offspring similarity, colony membership, relatedness, and genetic quality in fur seals. Proceedings of the National Academy of Sciences. 2015;112(36):E5005–E5012.

7. Charpentier MJE, Crawford JC, Boulet M, Drea CM. Message 'scent': Lemurs detect the genetic relatedness and quality of conspecifics via olfactory cues. Animal Behaviour. 2010;80(1):101–108. doi:10.1016/j.anbehav.2010.04.005.

8. Leclaire S, Merkling T, Raynaud C, Mulard H, Bessière JM, Lhuillier É, et al. Semiochemical compounds of preen secretion reflect genetic make-up in a seabird species. Proceedings of the Royal Society of London B: Biological Sciences. 2012;279(1731):1185–1193.

9. McNair HM, Miller JM. Basic gas chromatography. John Wiley & Sons; 2011.

10. Scott RP. Principles and practice of chromatography. Chrom-Ed Book Series. 2003;1.

11. Pierce KM, Hope JL, Johnson KJ, Wright BW, Synovec RE. Classification of gasoline data obtained by gas chromatography using a piecewise alignment algorithm combined with feature selection and principal component analysis. Journal of Chromatography A. 2005;1096(1):101–110.

12. Dellicour S, Lecocq T. GCALIGNER 1.0: an alignment program to compute a multiple sample comparison data matrix from large eco-chemical datasets obtained by GC. Journal of separation science. 2013;36(19):3206–3209. doi:10.1002/jssc.201300388.

13. Drea CM, Boulet M, DELBARCO-TRILLO J, Greene LK, Sacha CR, Goodwin TE, et al. The "secret" in secretions: methodological considerations in deciphering primate olfactory communication. American journal of primatology. 2013;75(7):621–642.

14. Allaire JJ, Cheng J, Xie Y, McPherson J, Chang W, Allen J, et al.. rmarkdown: Dynamic Documents for R; 2016. Available from: https://CRAN.R-project.org/package=rmarkdown.

15. Oksanen J, Blanchet FG, Friendly M, Kindt R, Legendre P, McGlinn D, et al.. vegan: Community Ecology Package; 2016. Available from: https://CRAN.R-project.org/package=vegan.

16. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York; 2009. Available from: http://ggplot2.org.

17. Anderson MJ. A new method for non-parametric multivariate analysis of variance. Austral Ecology. 2001;26(1):32–46. doi:10.1111/j.1442-9993.2001.01070.pp.x.

18. Wickham H, Chang W. devtools: Tools to Make Developing R Packages Easier; 2016. Available from: https://CRAN.R-project.org/package=devtools.