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 arguably the oldest and most fundamental means of animal communication and play a fundamental role in mate choice and kin selection. A key tool for uncovering broad patterns of chemical similarity is gas chromatography (GC). However, downstream analyses rely on the correct alignment of homologous substances, represented by peaks, across samples. Here we present GCalignR, a user-friendly R package for aligning GC data based on retention times. The package also implements a suite of dynamic visualisation tools to facilitate inspection of the resulting alignments and can be integrated within a broader workflow in R to facilitate downstream multivariate analyses. We demonstrate an example workflow using a chemical dataset from Antarctic fur seals, show that the resulting alignments are relatively insensitive to realistic levels of randomly introduced noise, and also test the package on three pre-validated datasets to reveal generally rather low alignment error rates. We hope that GCalignR will help to simplify the processing of chemical datasets and contribute towards improved standardization and reproducibility.

Introduction

Chemical cues are arguably the most common mode of communication among animals [18]. Patterns in complex chemical signatures can therefore yield information about phylogenetic relatedness [11], sexual maturation [4], kinship [3, 8, 15] and genetic quality [5, 9, 15]. 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 [10]. 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 [13, 14]. Variation in peak intensities both within and among samples can also contribute towards errors in characterising chemical profiles.

GC is widely used in behavioural and ecological studies of (usually) non model organisms, such as birds, mammals and insects, where the goal is often to investigate broad chemical patterns. These studies tend to use GC-MS less often than studies of humans and other model organisms, both because GC-MS is relatively expensive and because many of the chemicals that are resolved often reveal limited homology to currently available databases. However, aligning GC data is non-trivial due to the anonymous nature of the many peaks. Consequently, although a number of programs are available for aligning GC-MS data, which make use of mass spectrograms, we are only aware of a single program that can handle GC data [6]. As a result, most studies of mammalian and avian chemical communication have relied on manual alignment and peak calling [7], which is time-consuming, particularly for large samples of individuals, can be biased and subjective, and is not strictly reproducible.

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

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

Implementing GC alignment and checking within R brings several advantages over currently available stand-alone programs. First, the code is open source, which facilitates flexibility and transparency in data analysis. Second, all computational steps can be integrated into R Markdown documents [1], thereby enhancing reproducibility. Finally, our package provides a seamless transition from the processing of the peak data through to downstream analysis within other widely used R packages for multivariate analysis, e.g. vegan [12].

The package

GCalignR contains functions to align peaks from GC and GC-MS data based on retention times and evaluate the resulting alignments. The main aim of the package is to provide a simple tool that guides the user through the alignment of large datasets prior to the statistical analysis of multivariate chemical data. A typical workflow for the analysis of chemical signatures in GCalignR is shown in Fig 1 and described below. The package vignette provides a detailed description of all of the functions and their 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 [15]. 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 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.

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 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 (in minutes) 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 value (in minutes) 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 exceeding 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

(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 and the colouring of cells refers to the substance identity in the final alignment. Consecutive manipulations of the matrices are shown from left to right. Here, black rectangles indicate conflicts that are solved by manipulations of the matrices. Zeros indicate absence of peaks and are therefore not considered in computations. **i**. Chromatograms are linearly shifted with respect to a reference (here S2). **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.

(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.

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

where rt = retention time; m = focal cell and max_diff_peak2mean defines the user-defined threshold deviation from the mean retention time (see Table 1).

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

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

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

(iii) Merging rows

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

(iv) Removal of contaminants

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

(v) Removal of single peaks

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

Workflow

Here, we demonstrate a typical workflow in GCalignR using the fur seal dataset as an example. All of the alignment steps described above are implemented within the function `align_chromatograms`. A list of user-defined parameters and their descriptions can be accessed from the documentation in the helpfile by typing `?align_chromatograms`. Prior to peak alignment, the `check_input` function interrogates the input file for typical formatting errors and missing data. We encourage the use of unique names for samples consisting only of letters, numbers and underscores. If the data fail to pass this quality test, indicative warnings will be returned to assist the user in error correction. As this function is executed internally prior to any alignment, the data need to pass this check before the alignment can begin.

```
library(GCalignR)
fpath <- system.file(dir = ”extdata”,
                     file = ”peak_data.txt”,
                     package = ”GCalignR”)
check_input(fpath)
```

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

```
aligned_peak_data <- align_chromatograms(data = peak_data,
        rt_col_name = ”time”,
        max_diff_peak2mean = 0.02,
        min_diff_peak2peak = 0.08,
        max_linear_shift = 0.05,
        delete_single_peak = TRUE,
        blanks = c(”C2”,”C3”))
```

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

```
print(aligned_peak_data) # verbal summary of the alignment
```

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

```
plot(aligned_peak_data) # creates Fig. 3
```

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

Fig 3. Diagnostic plots summarising the aligned dataset. A shows the number of peaks both prior to and after alignment; **B** shows a histogram of linear shifts across all samples; **C** shows the variation across samples in peak retention times defined by the difference between maximum and minimum retention time; and **D** shows a frequency distribution of substances shared across samples.

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

```
gc_heatmap(aligned_peak_data,type = ”binary”,threshold = 0.05)
```

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 user-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.

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.

Fig 4. Heatmap of the final alignment highlighting potentially problematic alignments. Substances on the horizontal axis are ordered by retention times, increasing from left to right. Blue cells indicate aligned peaks with retention times that deviate from the mean retention time of that substance by less than 0.05 minutes. Red cells highlight aligned peaks with retention times that fall outside the threshold. Parameters of the function are explained in the corresponding helpfile and the package vignette.

```
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, downstream multivariate analyses can be conducted using the package vegan [12]. 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 using the normalised and log-transformed chemical data:

```
# log + 1 transformation
scent <- log(scent + 1)
# sorting by row names
scent <- scent[match(row.names(peak_factors),
                     row.names(scent)),]
# 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 approach along with the accompanying factors are stored in the data frame `scent_nmds` and can be visualised using ggplot2 [16].

```
library(ggplot2)
ggplot(data = scent_nmds,aes(MDS1,MDS2,color = colony)) +
    geom_point() +
    theme_void() +
    scale_color_manual(values = c(”blue”,”red”)) +
    theme(panel.background = element_rect(colour = ”black”,
     size = 1.25,  fill = NA),
     aspect.ratio = 1,
     legend.position = ”none”)
```

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

Fig 5. A NMDS plot shows the similarity of individuals within colonies. Individuals are colour coded based on the breeding colony. Blue and red points refer to the 'special study beach' and 'freshwater beach' respectively (see [15] for details).

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

We next explored the ability of the algorithm to cope with randomly introduced noise in the fur seal chemical dataset. Errors were introduced at random into the raw dataset following a Gaussian profile with -0.02 to 0.02. The resulting datasets were re-aligned and we then quantified the strength of clustering by colony using the function adonis from the package vegan, which performs a permutation-based multivariate analysis of variance ("permutational manova" [2]). For each error rate value, defined as the proportion of peaks within each sample with random errors, we generated 10 datasets. The frequency of introduced errors affected both the total number of scored substances as well as the strength of the detected pattern (Fig 6). The total number of substances scored gradually increased with increasing error rate (Fig 6 a) as higher variation in retention times caused peaks to be split into more than one substance. In parallel, Adonis R2 values fell linearly up to an error rate of 0.5 before levelling off at around 0.07. This suggests both that the original dataset is clearly structured and that the results of the alignment procedure within GCalignR are relatively robust to low to moderate (i.e. < 0.2) rates of peak calling error.

Fig 6. The influence of additional random noise on the detectability of patterns within the fur seal chemical dataset. Random errors following a Gaussian profile were introduced into the raw fur seal chemical dataset. **A**. The total number of scored substances in the aligned dataset increased linearly with additional noise levels. **B**. Contrastingly, the strength of the colony effect, determined by Adonis R2 values, decreased in response to increased noise levels.

To further assess the performance of GCalignR, we calculated alignment error rates based on three previously published bumblebee datasets comprising known substances identified using GC-MS [6]. The first dataset comprises 24 *Bombus bimaculatus* individuals characterised for 32 substances (total = 717 retention times). The second comprises 20 *B. ephippiatus* individuals characterised for 42 substances (total = 782 retention times) and the third comprises 11 *B. flavifrons* individuals characterised for 44 substances (total = 457 retention times). We calculated the error rate as the ratio of the number of incorrectly assigned retention times to the total number of retention times (Eq (3)).

$$\text{Error} = \left[\frac{\text{Number of missaligned retention times}}{\text{Total number of retention times}}\right] \tag{3}$$

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

Additionally, we simulated the effect of noise by introducing errors at random into each of the bumblebee datasets as described above for the fur seal dataset. As expected,

Fig 7. Effects of alignment parameters on error rates Error rates were calculated for three bumblebee datasets [6] (**A-C**) based on known substances. Each point shows the aligment error rate for a given combination of `max_diff_peak2mean` and `min_diff_peak2peak`.

error rates were initially low for all three species but increased with progressively increasing levels of noise (Fig 8). *B. bimaculatus* and *B. ephippiatus* both showed approximately linear responses whereas *B. flavifrons* appeared to be relatively insensitive until and additional noise level of around 0.7 was exceeded.

Fig 8. The effect of introduced random noise on alignment error rates for three bumblebee datasets.

Final remarks

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

Availability

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

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

We aim to extend the functionalities of GCalignR in future and the developmental version can be downloaded from GitHub within R using devtools [17].

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

The raw fur seal chemical dataset is included in this R package and the bumblebee datasets [6] can be downloaded here `http://onlinelibrary.wiley.com/store/10.1002/jssc.201300388/asset/supinfo/jssc3437-sup-0001-TableS1.zip?v=1&s=57d5d58273d1d4207e70c72cecd5bba4b1fe95a1`.

Supporting information

File S1. R code. The code and accompanying documentation for all simulations presented in this manuscript are provided in a PDF file.

Data S2. Datasets used to generate the results presented in this manuscript. This is a compressed zip archive that includes all the raw data that were used for the simulations presented in this manuscript.

Acknowledgments

We are grateful to Barbara Caspers and Sarah Golüke for helpful discussions during the development of the package. This work was funded by a Deutsche Forschungsgemeinschaft standard Grant HO 5122/3-1 (to J.I.H.).

References

1. 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.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

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

9. 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.

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

11. 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.

12. 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.

13. 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.

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

15. 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.

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

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

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