Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Retrieving and Analyzing Taste Colexifications from Lexibank

Olena Shcherbakova[1] and Johann-Mattis List[12]
[1] Department of Linguistic and Cultural Evolution, MPI for Evolutionary Anthropology, Leipzig
[2] Chair of Multilingual Computational Linguistics, University of Passau, Passau

Colexifications have enjoyed a considerable amount of popularity in the recent years. However, there are still many semantic domains, where not much research on colexification patterns has been carried out so far. Here we show, how the recently published Lexibank repository can be queried to yield colexification data on taste colexifications which can in turn be easily plotted on geographic maps.

1 Introduction

English has distinct words for 'salty', 'sweet', 'bitter' and 'sour'. This reflects the human ability to perceive and differentiate 4 basic tastes (ignoring umami). Yet, despite the discriminability of these tastes, many languages do not have distinct words for each of those concepts, so that the same word serves to express 'bitter' and 'sour' (tapay in Wichí, Matacoan, Key and Comrie 2015), 'bitter' and 'salty' (maka in Suboo, Timor-Alor-Pantar), 'sour' and 'salty' (ersi in Tiyei, Timor-Alor-Pantar, Kaiping et al. 2019). These languages are said to colexify basic tastes, and the languages of the world differ greatly in how they lexicalize these concepts.

Unfortunately, the absence of large-scale cross-linguistic data on basic tastes has hindered the investigations of taste colexifications, whereas other sense perception domains, such as color perception and their lexicalization, have received more attention starting with the seminal study by Berlin and Kay (1969, e.g. Haynie and Bowern 2016, Josserand et al. 2021, Dediu 2023). Malt and Majid (2013: 586) describe the lack of studies devoted to taste perception in the following way: "[a]t the turn of the previous century, there were two large-scale questionnaire/observational surveys on taste lexicons which demonstrated comparable variation in this domain. There has been very little pursuit since". The referenced "large-scale questionnaire/observational surveys" are

Computer-Assisted Language Comparison in Practice
Volume 6, Number 10
https://calc.hypotheses.org/6398/

Published on 30/10/2023

Chamberlain (1903) and Myers (1904) that point out intriguing instances of taste colexifications, yet on relatively small convenience samples. These studies give an impression of linguistic diversity in taste colexifications in Oceania, North America, and Eurasia, but with more data becoming increasingly available, more systematic and truly large-scale studies should be feasible nowadays.

Here we demonstrate how to information on taste colexifications can be first extracted automatically from the Lexibank repository (List et al. 2022) and then visualized by plotting the geographic distribution of different taste colexifications across several hundreds of languages in R (R Core Team 2022). These pilot studies on taste colexifications show that more data is necessary to make robust inferences about linguistic diversity of basic taste words and to test hypotheses about the mechanisms underlying the variation in lexicalization of the taste perception domain.

2 Materials

For this initial study on taste colexifications, we use data from the Lexibank repository, recently published in Version 1.0 (List et al. 2023). In contrast to the first version of Lexibank, published in 2022 (List et al. 2022), Version 1.0 now comes with three new additions. First, while the earliest version of Lexibank only provides the results of analyzing all individual datasets available to the repository and then shares these results in the form of automatically created feature collections, Lexibank 1.0 now includes standardized segmented phonetic transcriptions for more than 2000 distinct language varieties. Second, these data can be directly queried in a Lexibank CLLD application (https://lexibank.clld.org/). Third, thanks to the fact that the resulting Lexibank data is all available in Cross-Linguistic Data Formats (Forkel et al. 2018), the data can be easily converted to SQLite (https://sqlite.org, Hipp 2023), which means one can query the data conveniently with SQL statements.

3 Retrieving Colexifications from Lexibank

In order to query data from Lexibank, we first convert the current 1.0 version of the Lexibank repository (available on GitHub at https://github.com/lexibank/lexibank-analysed/tree/v1.0) into SQLite using the pycldf library (Forkel 2023, https://pypi.org/project/pycldf).

```
$ git clone https://github.com/lexibank/lexibank-analysed
$ cd lexibank-analysed
$ git checkout v1.0
$ pip install -e pycldf
$ cldf createdb lexibank-analysed/cldf/wordlist-metadata.json lexibank.sqlite3
```

Having created the SQLite database, colexifications can be easily extracted from the data by means of an SQL query. The query itself may look complex, specifically for those who do not know the internal CLDF data structures. However, the query has several advantages over Python or R scripts. First, it is extremely fast, taking only a few seconds on a "normal" laptop that was used for this study. Second, when comparing the complexity of the query with the amount of code one would have to write to achieve the same with Python or R, the query can be considered as straightforward and small. Third, since SQL queries naturally output tabular data, and since SQLite supports output in CSV format, the data the query produces is given in a form that can be directly and conveniently reused by other scripts, be they written in Python or R or any other programming language.

For convenience, we store the query in a shell script that can be invoked directly from the terminal. Alternatively, one can also enter the interactive SQLite mode and paste the query there. The major strategy of the query is to first select all those words that encode one of the four taste concepts (glossed as SOUR, BITTER, SWEET, SALTY in Concepticon, List et al. 2023, https://concepticon.clld.org). This query can be done with the following query.

```
SELECT
  l.cldf_name as LanguageName,
  l.cldf_glottocode as Language,
  l.family as Family,
  p.cldf_name as Concept,
  f.cldf_segments as Segments
FROM
  formtable as f,
  languagetable as l,
  parametertable as p
WHERE
  p.cldf_id == f.cldf_parameterReference
    AND
  l.cldf_id == f.cldf_languageReference
    AND
  (
    p.cldf_name == 'SOUR'
      OR p.cldf_name == 'BITTER'
      OR p.cldf_name == 'SWEET'
      OR p.cldf_name == 'SALTY'
  );
```

In order to compare within one and the same language, whether the word for SOUR colexifies with the word for BITTER, or the word for SWEET colexifies with a word for SALTY, we can carry out a JOIN (https://de.wikipedia.org/wiki/Join_(SQL)) of this

query with itself, which will give us a new combined table in which we find both SOUR and BITTER or both BITTER and SWEET. All we have to do in order to find out whether these combinations of two expressions colexify is to check their identity.

This results in the following, admittedly long, SQL query, which we have commented in parts (with comments preceded by two dash symbols, --) so that the structure becomes a bit clearer.

```
-- selection of the two joined tables check for colexifications in the last column
SELECT
  ROW_NUMBER() OVER() as ID,
  table_a.LanguageName, table_a.Language, table_a.Latitude, table_a.Longitude,
  table_a.Family, table_a.Concept||'+'||table_b.ConceptB as Parameter,
  table_a.Segments, table_b.SegmentsB, table_a.Segments = table_b.SegmentsB as Value
-- query four words in the first table
FROM
  (SELECT
     l.cldf_name as LanguageName, l.cldf_latitude as Latitude, l.cldf_longitude as Longitude,
     l.cldf_glottocode as Language, l.family as Family, p.cldf_name as Concept,
     f.cldf_segments as Segments
    FROM
     formtable as f, languagetable as l, parametertable as p
    WHERE
     p.cldf_id = f.cldf_parameterReference AND l.cldf_id = f.cldf_languageReference
       AND (p.cldf_name = 'SOUR' OR p.cldf_name = 'BITTER' OR p.cldf_name = 'SWEET'
         OR p.cldf_name = 'SALTY')
) as table_a
-- query the words in the second table to join them
INNER JOIN
  (SELECT
     l2.cldf_glottocode as LanguageB, p2.cldf_name as ConceptB,
     f2.cldf_segments as SegmentsB
    FROM
     formtable as f2, parametertable as p2, languagetable as l2
    WHERE
     f2.cldf_languageReference = l2.cldf_id AND f2.cldf_parameterReference = p2.cldf_id
       AND (ConceptB = 'SOUR'  OR ConceptB = 'BITTER' OR ConceptB = 'SWEET'
         OR ConceptB = 'SALTY')
  ) as table_b
-- conditions for the output, limit to the same language and also to diverging concepts
ON table_a.Language = table_b.LanguageB AND table_a.Concept != table_b.ConceptB
    AND ((table_a.Concept = 'BITTER'  AND table_b.ConceptB == 'SALTY') OR
      ( table_a.Concept = 'BITTER' AND table_b.ConceptB == 'SOUR') OR
      (table_a.Concept = 'BITTER' AND table_b.ConceptB == 'SWEET') OR
      (table_a.Concept = 'SALTY'  AND table_b.ConceptB == 'SOUR') OR
      (table_a.Concept = 'SALTY' AND table_b.ConceptB == 'SWEET') OR
      (table_a.Concept = 'SOUR' AND table_b.ConceptB == 'SWEET'))
-- order to retrieve data for each language in a block
ORDER BY Language, Parameter;
```

In order to wrap this query into a shell script, one just needs to paste it into a text file (that can conveniently be given the ending .sh) and put three lines in the beginning of the file and one in its end, as shown in the following code block, that also makes sure the output generated by SQLite is valid CSV.

```
$ sqlite3 lexibank.sqlite3 <<EOF
.headers on
.mode csv

# SQL QUERY HERE

EOF
```

This shell script can then be directly invoked from the terminal and it will yield the output in CSV format. To store the output in a file, one can use Shell syntax.

```
$ sh query.sh > tc.tsv
```

The CSV file that we can produce in this way contains not only the information on whether two concepts expressing one of the four basic tastes colexify, but also gives us the geolocation of the language, thanks to the fact that the CLDF data in Lexibank is integrated with Glottolog (https://glottolog.org, Hammarstrom et al. 2023). As we will see when plotting the data to geographic maps, having coordinates in the file in this form will come in handy.

The result itself shows that Lexibank does not provide a very large coverage on taste terms. Out of the more than 2000 language varieties in Lexibank, there are only 621 (about one fourth) language varieties in which at last two of the taste terms occur. Given that we have a shell script for our SQLite query, we can easily calculate this number when using the possibility of concatenating multiple commands together in the Shell, using csvkit to query the CSV file (Groskopf and McKinney 2023, https://pypi.org/project/csvkit/).

```
$ sh query.sh | csvcut -c 3 | sort -u | wc -l
621
```

This command first carries out our query, then cuts the third column out of the resulting CSV file (which contains the Glottocodes), then sorts the data by taking only unique values, and finally counting all lines in the resulting data.

As a quick inspection of the CSV file reveals, we do find two major kinds of colexifications, although there are six possibilities in total. There are 41 colexifications for BITTER and SALTY (out of a total of 310 languages for which there are words for

both concepts in Lexibank) and there are 47 colexifications for BITTER and SOUR (out of a total of 610 languages for which there are words for the concepts in Lexibank). We can automatically retrieve this information with a simple Shell command, as shown below (for the case of positive examples for BITTER and SALTY colexifications.

```
$ csvcut tc.csv -c Language,ParameterValue | grep "BITTER+SALTY,1" | wc -l
 41
```

4 Plotting Data with R

Data provided in the form of a CSV file including geocoordinates for individual languages can be plotted in various ways using various programs and techniques. Given the large number of users who use R in their daily work, we illustrate in the following how the data can be plotted with R. The R code itself was modified from code originally provided by Simon J. Greenhill.

4.1 Loading Libraries and Reading Data

Before plotting loading the data in R and plotting colexifications on the world map, three packages with their dependencies must be installed, `tidyverse` (Wickham et al. 2019), `ggplot2` (Wickham 2016), and `maps` (Becker et al. 2022). In order to make sure that the code proposed here can be easily reproduced, we recommend to use the groundhog package (Simonsohn and Gruson 2021, https://groundhogr.com/), which allows to manage the import of library versions based on a certain date. The code shown here can be run both in the interactive R console or from the terminal using the Rscript command. When using groundhog, as we do here, you must first install and load the library in an interactive session and specify the path where the packages should be installed.

```
> library(groundhog)
> set.groundhog.path("rpkg")
```

After entering this command, you will be prompted to confirm the selection. Once this has been done, we can load the libraries in the versions that were available on October 1st in 2023.

```
library(groundhog)
pkgs <- c("tidyverse", "ggplot2", "maps")
groundhog.library(pkgs, "2023-10-01")
```

In the code snippet below, we read in the file, choose the concept pairs of interest (`filter()`), and remove duplicated rows. For now, we are interested in

BITTER/SALTY and BITTER/SOUR concept pairs since they are colexified more frequently than other concepts pairs. We `select()` the columns we no longer need and keep only distinct() rows of the data frame.

Next we `group_by()` the columns LanguageName, Language, Family, Parameter, Latitude, and Longitude before applying the `summarize()` function and saving the results in the new Colexification column. This allows to detect colexifications if multiple word forms are associated with the same concept in the same language. If within a concept pair at least one colexification is detected, this language is coded as having a colexification for this pair of concepts (coded as 1 as opposed to 0). For instance, Chuvash (Turkic) has three words forms for SOUR — [jyçə], [jyçɛk], and [kəʋasak] —, the former of which also expresses BITTER. Thus, in this language, BITTER and SOUR are considered to be colexified even though two other forms — [jyçɛk] and [kəʋasak] — do not express BITTER. Since the rest of the functions do not need to be applied keeping the grouping of the columns, we make sure to use ungroup() before proceeding.

Finally, the `pivot_wider()` function breaks down the Parameter column into two separate ones BITTER+SALTY and BITTER+SOUR filling them with the values from the Value column.

```
taste <- read_csv("tc.csv", show_col_types = FALSE
) %>% filter(
    Parameter == "BITTER+SALTY" | Parameter == "BITTER+SOUR"
) %>% dplyr::select(
    -c(Segments, SegmentsB, ID)
) %>% distinct(.keep_all = TRUE) %>% group_by(
    LanguageName, Language, Family, Parameter, Latitude, Longitude
) %>% summarize(
    Value = ifelse(1 %in% Value, 1, max(Value)), .groups = "keep"
) %>% ungroup(
) %>% pivot_wider(names_from = Parameter, values_from = Value)
```

4.2 Preparing the World Map Background

The function `map_data()` prepares the data from the maps package to be plotted with `ggplot2`.

```
world <- map_data(
    "world", wrap = c(-25, 335), ylim = c(-56, 80),margin = T
)
lakes <- map_data(
    "lakes", wrap = c(-25, 335), col = "white", border = "gray", ylim = c(-55, 65), margin = T
)
```

Since we want the Pacific Ocean to be positioned in the center of the map so that Austronesian languages are not divided as in the more common Atlanic-centered maps, we need to shift the Longitude values.

```
taste <- taste %>% dplyr::mutate(
   Longitude = if_else(Longitude <= -25, Longitude + 360, Longitude))
```

Next, we prepare the plot of the world map (saved as basemap) assembling the previously created world and lakes. This world map will be used as the background for plotting our data in the steps below.

```
basemap <- ggplot(
   taste
) + geom_polygon(
   data = world,
   aes(
      x = long,
      y = lat,
      group = group
   ),
   colour = "gray87",
   fill = "gray87",
   linewidth = 0.5
) + geom_polygon(
   data = lakes,
   aes(x = long, y = lat, group = group),
   colour = "gray87",
   fill = "white",
   linewidth = 0.3
) + theme(
   panel.grid.major = element_blank(),
   panel.grid.minor = element_blank(),
   axis.title.x = element_blank(),
   axis.title.y = element_blank(),
   axis.line = element_blank(),
   panel.border = element_blank(),
   panel.background = element_rect(
      fill = "white"
   ),
   axis.text.x = element_blank(),
   axis.text.y = element_blank(),
   axis.ticks = element_blank()
) + coord_map(
   projection = "vandergrinten",
   ylim = c(-56, 67)
)
```

4.3 Data Preparation

We transform our numeric columns with concept pairs into factors using `as.factor()` and create two separate dataframes that exclude missing values for each concept pair. These new dataframes will be used for plotting the colexifications on the map.

```
taste <- taste %>% dplyr::mutate(
    `BITTER+SOUR` = as.factor(`BITTER+SOUR`),
    `BITTER+SALTY` = as.factor(`BITTER+SALTY`)
)

taste_1 <- taste %>% filter(
    !is.na(`BITTER+SOUR`)
) %>% dplyr::mutate(
    Colexification = `BITTER+SOUR`
)

taste_2 <- taste %>% filter(
    !is.na(`BITTER+SALTY`)
) %>% dplyr::mutate(
    Colexification = `BITTER+SALTY`
)
```

4.4 Plotting the World Map for BITTER and SOUR

We first plot the BITTER and SOUR colexification absence/presence on the previously created world map background (basemap). For absent and present colexifications, we use distinct shapes and colors.

```
p_1 <- basemap + geom_point(
    data = taste_1,
    aes(x = Longitude, y = Latitude, shape = `BITTER+SOUR`, colour = `BITTER+SOUR` ),
    alpha = 0.7,
) + scale_color_manual(
    values = c("#009E73", "#E69F00"),
    labels = c("absent", "present"),
    name = "BITTER+SOUR"
) + scale_shape_manual(
    values = c(19, 17),
    labels = c("absent", "present")
) + ggplot2::theme(
    text = element_text(size = 17),
    plot.margin = unit(c(0,-5, 0, -5), "cm")
)
```

The result is shown in Figure 1 below. As can be seen, the BITTER and SOUR colexification can be found in quite some places in the world, occurring in languages spoken across different language families.

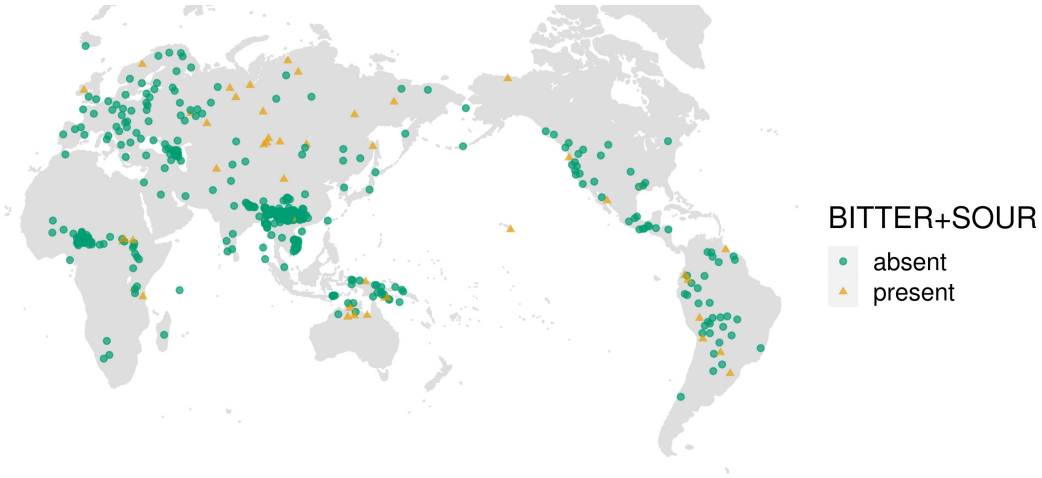

Figure 1: Colexifications of BITTER and SOUR

4.5 Plotting the World Map for BITTER and SALTY

We go through the same steps as for plotting the previous map but make sure to exchange the dataframe name from taste_1 to taste_2 and change the name of the concept pair to BITTER+SALTY.

```
p_2 <- basemap + geom_point(
    data = taste_2,
    aes(x = Longitude, y = Latitude, shape = `BITTER+SALTY`, colour = `BITTER+SALTY`),
    alpha = 0.7,
) + scale_color_manual(values = c("#009E73", "#E69F00"),labels = c("absent", "present"),
    name = "BITTER+SALTY"
) + scale_shape_manual(values = c(19, 17), labels = c("absent", "present")
) + ggplot2::theme(
  text = element_text(size = 17),
  plot.margin = unit(c(0,-5, 0, -5), "cm" )
)
```

As can be seen from Figure 2, the colexification of BITTER and SALTY is much more restricted to a specific region in South-East Asia, where mostly Sino-Tibetan languages are spoken. This shows that it may be useful to look a bit closer at the distribution of BITTER and SALTY colexifications in this area.

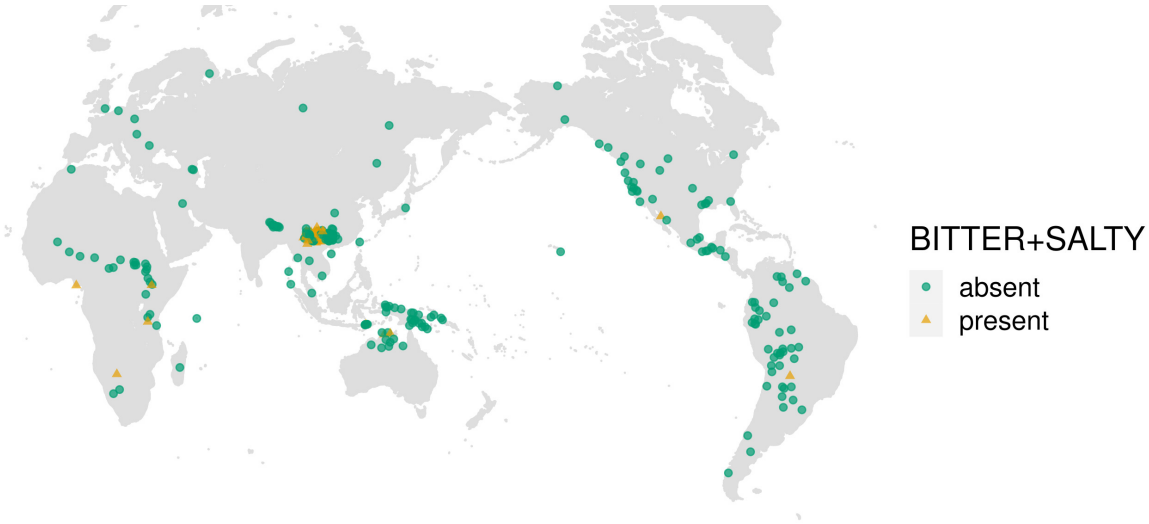

Figure 2: Colexifications of BITTER and SALTY

4.6 Plotting the Map of a Selected Region

To delve into the distribution of colexification in Sino-Tibetan languages, we zoom in on the selected region in Asia by applying new xlim and ylim values within coord_map that correspond to Latitude and Longitude values.

If we wanted to plot only Sino-Tibetan languages, we could filter() the data-frame to discard other languages. However, it might be interesting to keep other closely located languages from the region. So we opt for highlighting the Sino-Tibetan languages in the map by increasing the transparency of dots that represent other language families. For this, we 1) create a new column that tracks whether the language belongs to the Sino_Tibetan language family and turns its values into the factors, 2) add alpha to aes() call so that transparency varies based on our newly created column, and 3) specify the transparency for the values of the column (i.e. no transparency (1) for Sino-Tibetan languages and 0.5 value for alpha for other languages). Finally, we remove the unneeded legend for transparency from the plot.

```
taste_2 <- taste_2 %>% dplyr::mutate(Sino_Tibetan = ifelse(Family == "Sino-Tibetan", 1, 0)
) %>% dplyr::mutate(Sino_Tibetan = as.factor(Sino_Tibetan))

p_3 <- basemap + geom_point(data = taste_2,
  aes(x = Longitude, y = Latitude, shape = `BITTER+SALTY`, colour = `BITTER+SALTY`,
    alpha = Sino_Tibetan)
) + scale_color_manual(values = c("#009E73", "#E69F00"), labels = c("absent", "present"),
    name = "BITTER+SALTY"
) + scale_shape_manual(values = c(19, 17), labels = c("absent", "present")
) + scale_alpha_manual(values = c(0.5, 1), name = "Sino-Tibetan", guide = "none"
) + ggplot2::theme(text = element_text(size = 17 ), plot.margin = unit(c(0, -5, 0, -5), "cm")
) + coord_map(ylim = c(12, 40), xlim = c(82, 115))
```

As can be seen from Figure 3, the distribution is indeed quite peculiar, being restricted to language varieties spoken the middle and western part of China. It seems very likely that the pattern is regional in nature.

4.7 Saving Data to File

Last not least, we have to save the plots to file. This can best be done with the `ggsave()` function of `ggplot2`.

```
print(p_1)
ggsave("bitter-sour.pdf", width=12, height=4)
print(p_2)
ggsave("bitter-salty.pdf", width=12, height=4)
print(p_3)
ggsave("bitter-salty-zoom.pdf", width=8, height=4)
```

5 Conclusion and Outlook

We have demonstrated how to retrieve information on taste colexifications from Lexibank and how these can be plotted onto geographic maps. The results show that after retaining only distinct colexifications per language variety for BITTER+SALTY colexifications, we find 41 BITTER+SALTY colexifications out of a total of 227 languages where both concepts are expressed. Of these, only 10 languages do not belong to the Sino-Tibetan family. Conversely, for distinct colexifications of BITTER and SOUR, we find 46 colexifications out of 466 languages which have words for both concepts in Lexibank. This colexification can be found in quite a few languages from different language families across the globe (but it is most widely represented in Turkic and Uralic languages).

The seemingly wider global distribution of BITTER and SOUR colexifications appears to be in line with the observation that this concept pair is among the most commonly encountered cross-linguistically, in particular due to the "common impalatability" of these two basic tastes (Myers 1904: 126). However, another allegedly frequent pair SALTY and SWEET, which in contrast to the previous basic tastes, is considered "pleasant" and "agreeable" (Myers 1904) is found only in three languages in our dataset. This highlights the importance of documenting linguistic diversity in the domain of taste perception to make robust inferences about the regularities in lexicalization of basic tastes. The map zoomed in on Sino-Tibetan languages and their neighbours showed that that there is only one non-Sino-Tibetan language in the region that colexifies BITTER and SALTY: Chuanqiandian (Hmong-Mien), the easternmost yellow triangle on the map. As for the displayed Sino-Tibetan languages, the next step would be investigating whether the presence of the colexification was mainly due to

inheritance, geographic proximity, or their combination, as well as the potential influence of cultural and environmental factors.

While observing the global distribution of two colexifications in basic taste terms, it is important to keep in mind the sample differences: the sample for BITTER and SOUR is more than twice as large as that for BITTER and SALTY. This is because the reference materials for many languages do not record the word for SALTY and hence it is impossible to determine if it is the same word as for other tastes or not (or whether it is present in certain languages in the first place).

Collecting more lexical data in this domain is crucial for understanding the variation in the conceptualization of basic tastes. It is also a prerequisite for testing hypotheses about the mechanisms underlying the emergence and loss of colexifications in the domain of taste perception. For instance, the matching patterns between 1) taste misidentifications of bitter and sour (their confusion is established in English-speaking populations, for instance (O'Mahony et al. 1979, Doty et al. 2017) and 2) colexifications (the cross-linguistic prevalence of BITTER and SOUR colexifications, so far established only on small convenience samples as in Myers 1904) have been proposed to reflect psychophysiological processes (Majid and Levinson 2008, Osawa and Ellen 2014: 76). On the other hand, colexifications of 'bitter' with other terms could result from lower sensitivity in populations to the bitter taste due to more frequent consumption of bitter foods which prevail in the diet of hunter-gatherers (Sjostrand et al. 2020) or due to genetic factors, such as larger proportions of non-tasters of bitter compounds (Drewnowski 2004, Doty et al. 2017). Rigorous hypothesis testing and the application of an evolutionary framework could reveal the interrelationship between linguistic diversity and language change, on the on hand, and human taste perception, on the other hand.

The source code and data that we used to carry out the experiments shown in this little study have been curated on GitHub where they can be found at https://github.com/clics/taste-colexifications/releases/tag/v1.0 and have been archived with Zenodo (https://zenodo.org/doi/10.5281/zenodo.10046125).

References

Becker, Richard A., Allan R. Wilks R, Ray Brownrigg, Thomas P. Minka & Alex Deckmyn (2022): maps: Draw Geographical Maps. https://CRAN.R-project.org/package=maps.

Berlin, Brent & Paul Kay. 1969. Basic color terms: Their universality and evolution. University of California Press.

Dediu, Dan (2023): Ultraviolet light affects the color vocabulary: evidence from 834 languages. Frontiers in Psychology 14. 1143283. Doty, Richard L, Jonathan H Chen & Jane Overend. 2017. Taste quality confusions: influences of age, smoking, PTC taster status, and other subject characteristics. Perception 46(3–4). 257–267.

Drewnowski, Adam (2004): 6-n-propylthiouracil sensitivity, food choices, and food consumption. In John Prescott & Beverly J. Tepper (eds.), Genetic Variation in Taste Sensitivity, 179–193. New York: Marcel Dekker.

Forkel, Robert, Johann-Mattis List, Simon J. Greenhill, Christoph Rzymski, Sebastian Bank, Michael Cysouw, Harald Hammarstrom, Martin Haspelmath, Gereon Kaiping, & Russell D. Gray (2018): Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics. Scientific Data 5.180205. 1-10. https://doi.org/10.1038/sdata.2018.205

Forkel, Robert (2023): PyCLDF [Software, Version 1.35.0]. https://pypi.org/project/pycldf/

Groskopf, Christopher & James McKinney (2023): CSVKIT [Software, Version 1.3.0]. https://pypi.org/project/csvkit/).

Hammarstrom, Harald, Robert Forkel, Martin Haspelmath & Sebastian Bank (2023): Glottolog [Dataset, Version 4.8] https://doi.org/10.5281/zenodo.8131084

Haynie, Hannah J. & Claire Bowern (2016): Phylogenetic approach to the evolution of color term systems. Proceedings of the National Academy of Sciences 113(48). 13666–13671.

Hipp, Richard (2023): SQLite [Software, Version 3.43.2]. https://www.sqlite.org/src

Josserand, Mathilde, Emma Meeussen, Asifa Majid & Dan Dediu (2021): Environment and culture shape both the colour lexicon and the genetics of colour perception. Scientific Reports 11(1). 19095.

Kaiping, Gereon, Owen Edwards & Marian Klamer (2019): LexiRumah [Dataset Version v3.0.0]. Zenodo. https://doi.org/10.5281/zenodo.3537977

Key, Mary Ritchie & Bernard Comrie (2015): The Intercontinental Dictionary Series. Leipzig: Max Planck Institute for Evolutionary Anthropology. https://ids.clld.org

List, Johann-Mattis, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, & Russell D. Gray (2022): Lexibank, A public repository of standardized wordlists with computed phonological and lexical features. Scientific Data 9.316. 1-31. https://doi.org/10.1038/s41597-022-01432-0

List, Johann-Mattis, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, & Russell D. Gray (2023): Lexibank Analysed [Data set, Version 1.0]. https://zenodo.org/records/7836668

Majid, Asifa & Stephen C Levinson (2008): Language does provide support for basic tastes. Behavioral and Brain Sciences 31(1). 86–87.

Malt, Barbara C & Asifa Majid (2013): How thought is mapped into words. Wiley Interdisciplinary Reviews: Cognitive Science 4(6). 583–597.

Myers, Charles S. (1904): The taste-names of primitive peoples. British Journal of Psychology 1(2). 117–126.

O'Mahony, Michael, M Goldenberg, J Stedmon & J Alford (1979): Confusion in the use of the taste adjectives 'sour' and 'bitter.' Chemical Senses 4(4). 301–318.

Osawa, Yoshimi & Roy Ellen (2014): The cultural cognition of taste term conflation. The Senses and Society 9(1). 72–91.

R Core Team (2022): R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Sjostrand, Agnès E., Per Sjodin, Tatyana Hegay, Anna Nikolaeva, Farhad Shayimkulov, Michael GB Blum, Evelyne Heyer & Mattias Jakobsson (2021): Taste perception and lifestyle: insights from phenotype and genome data among Africans and Asians. European Journal of Human Genetics 29(2). 325–337.

Simonsohn, Uri & Hugo Gruson (2021): Groundhog. Wharton Credibility Lab at the University of Pennsilvania: Philadelphia. https://groundhogr.com/

Wickham, Hadley (2016): ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. (2019): Welcome to the tidyverse. Journal of Open Source Software 4(43). 1686. https://doi.org/10.21105/joss.01686.