Chapter X. Electropalatography

In The Phonetic Analysis of Speech Corpora. (Forthcoming). Blackwell.

Jonathan Harrington, IPDS, University of Kiel, Germany.
1. Palatography and electropalatography

Palatography is the general term given to the experimental technique for obtaining records of where the tongue makes a contact with the roof of the mouth. The earliest types of palatographic techniques were static allowing recordings to be made of a single consonant typically produced between vowels. In this technique, which is still very useful especially in fieldwork (e.g., Ladefoged, 2003), the roof of the mouth is coated in a mixture of olive oil and powdered charcoal and the subject produces a consonant. Details of the consonant's place of articulation and stricture are obtained from a photograph taken of the roof of the mouth showing where the powder was wiped-off and sometimes also of the tongue (which is coated in the powder at the point where tongue-palate contact was made). Dynamic electropalatography is an extension of this technique in which tongue-palate contacts are recorded as a function of time. In this technique, each subject has to have a custom-made acrylic palate that is fixed to the roof of the mouth using clasps placed over the teeth. The palate is very thin and contains a number of electrodes that are exposed to the surface of the tongue (Fig. X.1).

Each electrode is connected to a wire and all the wires from the electrodes are passed out of the corner of the subject's mouth in two bundles. The wires are fed into a processing unit whose job it is to detect whether or not there is electrical activity in any of the electrodes. The choice in binary in all cases: either there is activity or there is not. Electrical activity is registered whenever the tongue surface touches an electrode because this closes an electrical circuit that is created by means of a small electrical current passed through the subject's body via a hand-held electrode.

Three EPG systems that have been commercially available include the Reading EPG3 system developed at the University of Reading and now sold by Articulate Instruments; a Japanese system produced by the Rion corporation and an American system that has been sold by Kay Elemetrics Corporation (see Gibbon & Nicolaidis, 1999 for a comparison of the three systems).

The palate of the Reading EPG3 system, which is the system that is compatible with EMU-R, contains 62 electrodes as shown in Fig. X.1 that are arranged in eight rows. The first row, at the front of the palate and just behind the upper front teeth contains six electrodes, and the remaining rows each have 8 electrodes. There is a greater density of electrodes in the dental-alveolar region than in the dorsal region to ensure that the fine detail of lingual activity that is possible in the dental, alveolar, and post-alveolar zones can be recorded. The last row is generally positioned at the junction between the subject's hard and soft-palate.

Fig 1 also shows the type of display produced by the EPG-system; the cells are either black (1) when the corresponding electrode is touched by the tongue surface or white (0) when it is not. This type of display is known as a palatogram and the EPG3 system typically produces palatograms at a sampling frequency of 100 Hz, i.e., one palatogram every 10 ms. As Fig. X.1 shows, the palate is designed to register contacts extending from the alveolar to velar articulations with divisions broadly into alveolar (rows 1-2), post-alveolar (rows 3-4), palatal (rows 5-7) and velar (row 8).

Electropalatography, as we shall see in this Chapter, is an excellent tool for studying consonant cluster overlap and timing. It is also an invaluable tool for use in both the diagnosis and the treatment of speech disorders. Another major advantage of EPG is that
there is often a reasonably transparent relationship between phonetic quality and EPG output: a [t] really does show up as contacts in the alveolar zone, the different groove widths between [s] and [ʃ] are usually very clearly manifested in EPG displays, and coarticulatory and assimilatory influences can be seen and quite easily quantified. (See Gibbon, 2005, for a bibliography of electropalatographic studies since 1957).

At the same time, it is important to be clear about some of the limitations of this technique:

- A separate palate (involving a visit to the dentist for a plaster-cast impression of the roof of the mouth) has to be made for each subject which can be both time-consuming and expensive.
- As with any articulatory technique, subject-to-subject variation can be considerable. One subject's production of the stop in 'key' can show up as palatal and lateral contact, for another there may be only limited lateral contact and fewer rows may be contacted. This variation can come about not only because subjects may invoke different articulatory strategies for producing the same phonetic segment, but also because the rows of electrodes are not always aligned with exactly the same articulatory landmarks across subjects.
- EPG can obviously give no direct information about labial consonants (apart from coarticulatory effects induced by other segments) and in my experience there is usually only limited information for places of articulation beyond a post-palatal or pre-velar articulation: that is, /k/ in English shows up clearly in 'key', but for many subjects there is scarcely any recorded activity for the retracted /k/ 'call'.
- EPG can only give limited information about vowels. It does register the lateral contact in non-low front vowels, but provides little information about tongue position and velocity – for which electromagnetometry (Chapter Y) is more appropriate.
- The EPG3 system has a fixed EPG sampling rate of 100 Hz and the synchronised acoustic signal is fixed at a sampling frequency of 10000 Hz. At both of the laboratories where I have used EPG (SHLRC Macquarie and the IPDS, Kiel), I have had the system modified so that these parameters can be changed. A 100 Hz palatogram rate is often too slow to record the details especially in apical articulations; a 10000 Hz sampling frequency with the associated 5000 Hz cutoff is often too low for carrying out articulatory-acoustic modelling of fricatives.

2. An overview of electropalatography in EMU-R

There are four main components to the analysis of EPG data in R (Fig. X.2).

1. **A database** must obviously contain EPG signal files – these are signal files from the EPG3 system (usually together with .wav audio files) that have been converted into a format that can be read by EMU (see Appendix X for details).

2. **EPG Objects.** The EPG-data from the database is read into R in the same way as for any signal file data as a trackdata object using the `emu.track()` function. It is in fact an **EPG-compressed trackdata** object (Fig. X.2, box 2, A) because each palatogram consisting of 62 values over an 8 x 8 matrix is compressed into a vector of just 8 values. Since this is a trackdata object, then it is amenable to `dcut()` for obtaining an **EPG-compressed matrix at a single time point** (Fig. X.2, box 2, B). Both of these
EPG-compressed objects can be uncompressed in R (using the `palate()` function) to produce a **3D palatographic array** (Fig. X.2, box 2, C): that is, an array of palatograms containing 0s and 1s in an 8 x 8 matrix.

Any of these three objects are then amenable to two kinds of analysis: plotting or further parameterisation, as follows:

3. **EPG Plots.** Two kinds of plots are possible: either the palatograms showing their time-stamps, or a three-dimensional grey-scale plot that represents the frequency of contact over two or more palatograms.

4. **EPG data-reduced objects.** There reduce the 62 palatographic values to a single value. As we shall see later in this Chapter, they can be very useful for quantifying consonantal overlap and coarticulation.

It will be helpful to begin by looking in some further detail the types of R objects in box 2 (EPG Objects) of Fig. X.2, because they are central to all the other forms of EPG analysis, as the figure shows. All of the EPG-databases that are prestored and accessible within R and used as examples in this Chapter are initially in the form of EPG-compressed-trackdata objects (A. in Fig. X.2) and this is also always the way that you would first encounter EPG data in R if you are using your own EPG database. One of the available EPG-database fragments is `coutts` and it includes the following R objects:

- `coutts` word segment list (of the sentence: 'just relax said Coutts'; one segment per word)
- `coutts.sam` sampled speech trackdata object of `coutts`
- `coutts.epg` EPG-compressed-trackdata object of `coutts` (frame rate 5 ms)

The segment list, `coutts`, consists of four words of a sentence produced by a female speaker of Australian English and the sentence forms part of a passage that was constructed by Hewlett & Shockey (1992) for investigating (acoustically) coarticulation in /k/ and /t/. Here is the segment list:

```
coutts
segment  list from database:  epgcoutts
query was:  [Word!=x ^ Utterance=u1]  
labels    start     end          utts
1   just 16018.8 16348.8 spstoryfast01
2  relax 16348.8 16692.0 spstoryfast01
3   said 16692.0 16840.2 spstoryfast01
4 Coutts 16840.2 17413.7 spstoryfast01
```

The EPG-compressed trackdata object `coutts.epg` therefore also necessarily consists of four segments, as can be verified with `nrow(coutts.epg)`. Thus the EPG data for the first word in the segment list, 'just', is contained in `coutts.epg[1,]` and it looks like this:
coutts.epg[1,
trackdata from track: epg
index:
  left right
[1,]   1    65
ftime:
  start   end
[1,] 16020 16340
data:
  T1  T2  T3  T4  T5  T6  T7  T8
16020 195 195 195 131 199 231 255  62
16025 195 199 195 135 199 231 255  62
16030 195 199 195 135 199 239 255  62
16035 199 199 199 199 199 255 255  62
... etc.

As always, the information in $index gives the information about how many frames (in this case how many separate palatograms) of data there are in the segment and $ftime gives the times of the first and last (65th) of these frames respectively. $data is a 65 x 8 matrix: 65 rows because there are 65 frames of data and 8 columns which provide the information about palatographic contacts in columns 8-1 respectively. As for all trackdata objects, the times at which the EPG-frames of data occur are stored in the row dimension names and for this example they show that palatographic frames occur at intervals of 5 ms (i.e. at times 16020 ms, 16025 ms, etc.).

Each of the values in $data can be unpacked into a series of zeros and ones corresponding to the absence and presence of contact in the palatogram. The 'unpacking' is done by converting these values into binary numbers after adding 1 (one). More specifically, the first entry in column 1 is 195. So to get the corresponding palatographic contacts for row 8, 195 + 1 = 196 is converted into binary numbers. 196 in binary form is 11000011 and so this is the contact pattern for the last (8th row) of the palate at time 16020 ms (i.e., there is lateral contact and no contact at the centre of the palate). Since the values in row 1 of columns 2 and 3 are also 195, we can deduce that this is the contact pattern for the last three rows of the palate at this time.

This job of converting the values in $data into binary values and hence palatographic contacts is done by the palate() function. So the palatogram for all 65 rows of data in coutts.epg[1,] i.e., of the word 'just' extending in time from 16020 ms to 16340 ms is obtained as follows:

p <- palate(coutts.epg[1,])

p is a three-dimensional array of palatograms, as is shown by the following:

dim(p)
[1]  8  8  65

The first element that is returned by dim(p) refers to the number of palatographic rows and the second to the number of palatographic columns: these are therefore always both 8 because each palatogram contains contacts defined over an 8 x 8 grid. The third entry is the number of palatograms. The result here is 65 because, as we have just seen, this is the number of frames (palatograms) for the first segment 'just' of this trackdata object.

A three-dimensional palatographic array is indexed in R with [r, c, n] where r and c are the row and column number of the palatogram and n is the frame number (from 1-
In order to get at the entire palatogram, omit the \( r \) and \( c \) arguments. So the first palatogram at the onset of the word 'just' (at time 16020 ms corresponding to the first row in `coutts.epg$data` is):

\[
p[][,,1] \\
\begin{array}{cccccccc}
C1 & C2 & C3 & C4 & C5 & C6 & C7 & C8 \\
R1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
R2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
R3 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
R4 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
R5 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
R6 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
R7 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
R8 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
\end{array}
\]

In this type of array, the row and column numbers are given as the respective dimension names. Since the first row of the EPG3 palate has 6 contacts (i.e., it is missing the two most lateral contacts), the values in row 1 column 1 and row 1 column 8 are always 0.

The indexing on the palatograms works as for vectors, but since this is a 3D-array, two preceding commas have to be included to get at the palatogram number: so \( p[][,,1:3] \) refers to the first three palatograms, \( p[][,,c(2, 4)] \), to palatograms 2 and 4, \( p[][,,1] \) to all palatograms except the first one, and so on. It is worthwhile getting used to manipulating these kinds of palatographic arrays because this is often the primary data that you will have to work with, if you ever need to write your own functions for analysing EPG data (all of the functions for EPG plotting and EPG data reduction in boxes 3 and 4 of Fig. X.2 are operations on these kinds of arrays). A useful way in which to become familiar with them is to make up some palatographic data. For example:

```r
# create 4 empty palatograms
fake = array(0, c(8, 8, 4))
# give appropriate row and dimension names for a palatogram
dimnames(fake) = list(paste("R", 1:8, sep=""), paste("C", 1:8, sep=""), NULL)
# fill up row 2 of the 3\textsuperscript{rd} palatogram with contacts
fake[2,,3] = 1
# fill up row 1, columns 3-6, of the 3\textsuperscript{rd} palatogram only with contacts
fake[1,3:6,3] = 1
# look at the 3\textsuperscript{rd} palatogram
fake[,,3]
  
C1 C2 C3 C4 C5 C6 C7 C8
R1 0 0 1 1 1 1 0 0
R2 1 1 1 1 1 1 1 1
R3 0 0 0 0 0 0 0 0
R4 0 0 0 0 0 0 0 0
R5 0 0 0 0 0 0 0 0
R6 0 0 0 0 0 0 0 0
R7 0 0 0 0 0 0 0 0
R8 0 0 0 0 0 0 0 0

# give contacts to rows 7-8, columns 1, 2, 7, 8 of palatograms 1, 2, 4
fake[7:8, c(1, 2, 7, 8), c(1, 2, 4)] = 1
# or
fake[7:8, c(1, 2, 7, 8), -3] = 1
# Look at rows 5 and 7, columns 6 and 8, of the palatograms 2 and 4:
fake[c(5,7), c(6, 8), c(2,4)]
```
As we shall see later on, the times at which palatograms occur are stored as the names of the third dimension and they can be set as follows:

```r
# assume that these four palatograms occur at times 0, 5, 10, 15 ms
times <- seq(0, by=5, length=4)
# store these times as dimension names of fake
dimnames(fake)[[3]] = times
```

This causes the time values to appear instead of the index number. So the same instruction as the previous one now looks like this:

```
, , 5
   C6 C8
 R5  0  0
 R7  0  1
, , 15
   C6 C8
 R5  0  0
 R7  0  1
```

These two (pieces of) palatograms occur at 5 ms and 15 ms

Functions can be applied to the separate components of arrays in R using the `apply()` function. For 3D-arrays, 1 and 2 in the second argument to `apply()` refer to the rows and columns (as they do for matrices) and 3 to the 3rd dimension of the array, for example:

```r
# sum the number of contacts in the 4 palatograms
apply(fake, 3, sum)
```

```
[1]  8  8 12  8
```

```r
# sum the number of contacts in the columns
apply(fake, c(2,3), sum)
```

```
 0 5 10 15
 C1 2 2  1  2
 C2 2 2  1  2
 C3 0 0  2  0
 C4 0 0  2  0
 C5 0 0  2  0
 C6 0 0  2  0
 C7 2 2  1  2
 C8 2 2  1  2
```

Notice that the above command returns a matrix whose columns refer to palatograms 1-4 respectively (at times 0, 5, 10, 15 ms) and whose show the summed values per

1 But the times do not appear as dimension names if you look at only a single palatogram – because in this special case, an array is turned into a matrix (which has only 2 dimensions, hence nowhere to put the 3rd dimension name).
palatographic column. So the entries in row 1 means: the number of contacts in column 1 of the palatograms occurring at 0, 5, 10, 15 ms are 2, 2, 1, 2 respectively. If you want to sum (or to apply any meaningful function) by row or column without differentiating by palatogram, then the second argument has to be 1 (for rows) of 2 (for columns) on its own. Thus:

```
apply(fake, 1, sum)
```

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

The first returned entry under R1 means that the sum of the number of contacts in row 1 of all four palatograms together is 4 (which is also given by `sum(fake[1,,])`).

As already mentioned, arrays can be combined with logical vectors in the usual way – but take care where to place the comma! For example, suppose that these are four palatograms corresponding to the labels "k", "k", "t", "k" respectively. Then the palatograms for "k" can be given by:

```
lab = c("k", "k", "t", "k")
which = lab=="k"
fake[,,which]
```

and rows 1-4 of the palatograms for "t" are:

```
fake[1:4,,!which]
```

and so on. Finally, if you want to be able to apply the functions in boxes 3 and 4 of Fig. X.2 to made-up data of this kind, then you have to declare the data to be of class "EPG" (this tells the functions that these are EPG-objects). This is done straightforwardly as:

```
class(fake) = "EPG"
```

Having established some basic attributes of EPG objects in R, we can now consider the two functions for plotting palatograms. As Fig. X.2 shows, palatograms can be plotted directly from epg-compressed trackdata objects or from time slices extracted from these using `dcut()`, or else from the 3D palatographic arrays of the kind discussed above. We will begin by looking at EPG data from the third and fourth segments 'said Coutts'. This is given by `epgplot(coutts.epg[3:4,])` (or by `epgplot(palate(coutts.epg[3:4,]))`) and the corresponding waveform from which the palatograms are derived by `plot(coutts.sam[3:4,, type="l"]`.

Some of the main characteristics of the resulting palatograms shown in Fig. X.3a are:

- The alveolar constriction for the fricative [s] of 'said' is in evidence in the first 7 palatograms between 16690 ms and 16720 ms.
- The alveolar constriction for [d] of 'said' begins to form at 16880 ms and there is a complete alveolar closure for 8 palatograms, i.e., for 40 ms.
- There is clear evidence of a doubly-articulated [ð̆] in 'said Coutts' (i.e., a stop produced with simultaneous alveolar and velar closures) between 16825 ms and 16835 ms.
- [k] of 'Coutts' is released at 16920 ms.
- The aspiration of 'Coutts' and the following [u] vowel extend through to about 17105 ms.
• The closure for the final alveolar [t] of 'Coutts' is first completed at 17120 ms. The release of this stop into the final [s] is at 17205 ms.

The interval including at least the doubly-articulated [d̪k] has been marked by vertical lines on the waveform in Fig. X.3b which was done with the locator() function that allows any number of points on a plot to be selected and the values in either x- or y-dimension to be stored (these commands must be entered after those used to plot Fig. X.3b):

```r
# select two time points at store the x-coordinates
times <- locator(2)$x
# the vertical boundaries in Fig. X.3b are at these times
times
 [1] 16828.48 16932.20
```

Use the xlim argument to plot the palatograms over this time interval and optionally the mfrow argument to set the number of rows and columns (you will also often need to sweep out the graphics window in R to get an approximately square shape for the palatograms):

```r
# Palatograms plotted between the interval defined by times and displayed in 2 x 10
epgplot(coutts.epg, xlim=times, mfrow=c(2,10))
```

The next example of manipulating and plotting electropalatographic data is taken from a fragment of a database of Polish fricatives that was collected in Guzik & Harrington (submitted). This database was used to investigate the relative stability of fricatives in word-final and word-initial position. Four fricatives were investigated: the alveolar [s], a post-alveolar [S], an alveolo-palatal [ć], and a velar [x]. They were produced in word-pairs across all possible combinations, including the homorganic sequences [s#s], [S#S], [ć#ć], [x#x]. The database fragment polhom is of the homorganic sequences produced by one native, adult male speaker of Polish. The palatographic data was sampled at 100 Hz:

```r
polhom  Segment list of Polish homorganic fricatives
polhom.l  A parallel vector of labels ("s", "S", "ć", "x", for [s#s], [ś#ś], [ć#ć], [x#x])
polhom.epg Parallel EPG trackdata
```

As `table(polhom.l)` shows, there are 10 homorganic fricatives in each category. If you have accessed the corresponding database epgpolish from the EMU website, then you will
see that the segment boundaries in the segment list `polhom` extend approximately from the acoustic onset to the acoustic offset of each of these homorganic fricatives.

We will begin by comparing `[s]` with `[ʃ]` as far as differences and similarities in palatographic contact patterns are concerned and this will be done by extracting the palatographic frames closest to the temporal midpoint of the fricatives. The data for "s" and "S" is accessed with a logical vector, and `dcut()` is used for extracting the frames at the midpoint:

```r
# logical vector to identify "s" and "S"
temp = polhom.l %in% c("s", "S")
# EPG-compressed trackdata for "s" and "S"
cor.epg = polhom.epg[temp]
# Matrix of EPG-compressed data for "s" and "S" at the temporal midpoint
cor.epg.5 = dcut(cor.epg, 0.5, prop=T)
# Labels for the above
cor.l = polhom.l[temp]
```

`sum(temp)` shows that there are 20 fricatives and `table(cor.l)` confirms that there are 10 fricatives per category. The following produces a plot of the palatograms at the temporal midpoint, firstly for `[s]`, then for `[ʃ]`. Rather than displaying the times at which they occur, the palatograms are numbered with the `numbering=T` argument:

```r
# logical vector: T when `cor.l` is "s", F when `cor.l` is "S"
temp = cor.l == "s"
# palatograms for "s"
epgplot(cor.epg.5[temp,], numbering=T)
# palatograms for "S"
epgplot(cor.epg.5[!temp,], numbering=T)
```

As expected, the primary stricture for `[s]` is further forward than for `[ʃ]` as shown by the presence of contacts for `[s]` but not for `[ʃ]` in row 1. A three-dimensional, grey-scale image can be a useful way of summarising the differences between two different types of segments and in R, the function for doing this is `epggs()`:

```r
par(mfrow=c(1,2))
epggs(cor.epg.5[temp,], main="s")
epggs(cor.epg.5[!temp,], main="S")
```

At the core of the `epggs()` function is a calculation of the **proportional number of times a cell was contacted.** When a cell is black, then it means that it was contacted in all the palatograms over which the function was calculated, and when a cell is white, then there were no contacts. Thus for `[s]` in Fig. X.6, the entire first column is black in this three-dimensional display because, as Fig. X.5 shows, all ten palatograms for `[s]` have their contacts on in
column 1; and column 5 of rows 1 and 2 for [s] are dark grey, because, while most [s] palatograms had a contact for this cell (numbers 2, 5, 6, 7, 9, 10 in Fig. X.5), others did not.

3. EPG data reduced objects

As discussed earlier, various functions can be applied to EPG-data that reduce each palatogram to a single value (Fig. X.2, box 4). The most basic of these is a function for producing a contact profile in which the contacts per palate are summed (X.3.1). The other data reduction functions which are discussed in X.3.2 are essentially further operations on contact profiles. In the final part of the Chapter (section 4) some of these data reduction functions are put to use for measuring the extent of overlap in consonant clusters and vowel-induced consonantal coarticulation.

All data reduction functions work on the same kinds of EPG-objects as those for plotting electropalatographic data in X.2. Thus, they can be applied to EPG-compressed trackdata objects, a matrix of EPG-compressed data extracted at a single time slice, or to a 3D-palatographic array. In all cases, the output is a single value per palatogram: if the data reduction functions are applied to an EPG-compressed trackdata object, these values are structured into a trackdata object. These points are elaborated further in the next section.

3.1 Contact profiles

A contact profile is a data reduction technique involving a summation of palatographic data by row(s) and/or by column(s). Contact profiles have a number of applications in phonetics: they can be used to distinguish between stops and fricatives at the same place of articulation (by summing the number of contacts in certain rows) or between different places of articulation (by summing contacts in different rows).

The function for calculating a contact profile is `epgsum()` and its default is to sum all the contacts per palate. Thus for the 3D-array `fake` created earlier, `epgsum(fake)` gives the same result as the operation applied in X.2 for summing contacts in the four palatograms, `apply(fake, 3, sum)`. But `epgsum()` can also be used to sum selectively by row and column. So `epgsum(fake, rows=1:4)` sums the contacts in rows 1-4, `epgsum(fake, rows=1:4, columns=c(1, 2, 7, 8))` sums contacts in rows 1-4 of columns 1, 2, 7 and 8. The additional argument `inactive=T` can be used to sum the inactive electrodes (also by row and by column), i.e., the 0s of the palatograms. The default is to sum the entire palatogram (in selected rows and or columns) but it is also possible to show the summations for the separate rows or columns using a second argument of 1 (for rows) or 2 (for columns). For example, in the previous section it was shown how `apply(fake, c(2,3), sum)` gives the sum of the contacts in the columns: an equivalent way of doing this is `epgsum(fake, 2)`. Some further examples are given in the help pages.

In Fig. X.3a, the separate palatograms at 5 ms intervals were shown for the words 'said Coutts'. By making a display of the summed contacts in rows 1-3, the articulations in the front part of the palate should become very clearly visible, while a summation in the back two rows over columns 3-6 should produce a display which is associated with the degree of tongue-dorsum contact in /k/ of 'Coutts'. Here are these two contact profiles:

```r
# sum rows 1-3 of the EPG-trackdata object over 'said Coutts'
fsum <- epgsum(coutts.epg[3:4,], rows=1:3)

# sum rows 7-8, columns 3-6 of the EPG-trackdata object over 'said Coutts'
bsum <- epgsum(coutts.epg[3:4,], rows=7:8, columns=3:6)
```

A plot of the contact profiles superimposed on each other together with the waveform is shown in Fig. X.7 and can be produced as follows:
The synchronised contact profiles in Fig. X.7 provide a great deal of information about the overlap and lenition of the alveolar and velar articulations, for example:

- The tongue dorsum for [k] already begins to rise during [e] of 'said'.
- The maximum overlap between [d] and [k] is at the point of the final stop release in 'said'.
- The [t] of 'Coutts' is less lenited compared with [d] of 'said', as shown by the greater number of contacts for the former extending over a greater duration.

Contact profiles could be used to distinguish between the Polish [s, ź] fricatives discussed earlier according to the central groove width which could be defined as the smallest number of inactive electrodes in any row over the central columns 3-6. For example, in the first five palatograms of [s] in Fig. X.5, this central groove width is 3, 1, 2, 2, 1 respectively; for the first 5 [ź] palatograms in Fig. X.5., the central groove width is usually at least one inactive contact greater: 3, 3, 2, 3, 2.

In order to obtain groove widths for the data in Fig. X.5, the first step is to count the number of inactive electrodes (i.e., those with a value of zero) over a particular row and column range: we will restrict this to the first four rows and to columns 3-6, since, as Fig. X.5 shows, this is the region of the palate within which the point of maximum narrowing occurs:

```r
# Commands repeated from before
temp = polhom.l %in% c("s", "S")
cor.epg = polhom.epg[temp,]
cor.epg.5 = dcut(cor.epg, 0.5, prop=T)
cor.l = polhom.l[temp]

# count the number of inactive electrodes in rows 1-4, columns 3-6
# and display the result separately by row
in.sum = epgsum(cor.epg.5, 1, rows=1:4, columns=3:6, inactive=T)
# show the first two rows of in.sum
in.sum[1:2,]
R1 R2 R3 R4
2120 3 3 4 4
1160 1 2 3 4
```
So that it is completely clear what is being counted, the first two palatograms of the array are listed below. The count on the right is of the zeros in bold:

\[
p = \text{palate}(\text{cor.epg.5})
p[,,1:2]
\]

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>R4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

A function is needed to get the \textit{minimum} groove width – that is the function should return 3 and 1 respectively for the above two palatograms. Since \texttt{in.sum} is a matrix, this can be done with the \texttt{apply()} function:

\[
\text{min.groove} = \text{apply}(\text{in.sum}, 1, \text{min})
\]

# minima for the first two palatograms above: this is correct (see the palatograms above)

\[
\text{min.groove}[1:2]
\]

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2120</td>
<td>1160</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A boxplot shows that the minimum groove width is less for \([s]\) than for for \([\text{S}]\):

```r
data = data.frame(min.groove = min.groove, cor.l = factor(cor.l))
boxplot(min.groove ~ factor(cor.l), ylab="Minimum groove width")
```

-------------------------------------------
Figure X.8 about here
-------------------------------------------

The above analysis was for one single palatogram per segment extracted at the temporal midpoint. The same kind of analysis could be carried out for every palatogram between the onset and offset of these fricatives. This would allow us to see not only if there was a difference in minimum groove width between \([s,\text{S}]\), but also whether groove width decreased from the fricative margins towards the fricatives’ temporal midpoint (this is to be expected given that the homorganic fricatives were flanked by vowels and given that the
extent of stricture in fricative production tends to increase from the margins towards the temporal midpoint).

The first step is to count the number of inactive electrodes in rows 1-4 and columns 3-6 as before, but this time for all the palatograms contained in the entire EPG-compressed trackdata object:

```r
# sum the number of inactive electrodes from the onset to
# offset for all segments in the EPG-trackdata object polhom.epg
# and store the count separately by row
in.sum.all = epgsum(polhom.epg, 1, rows=1:4, columns=3:6, inactive=T)
```

`in.sum.all` is a four-dimensional trackdata object that consists of the sum of inactive electrodes in rows 1-4 of columns 3-6 for every palatogram between the onset and offset of each fricative. So that it is clear what has just been calculated, Fig. X.9 shows the EPG data for the 10th segment (given by `epgplot(polhom.epg[10,])`), together with the corresponding minimum groove width values (given by `in.sum.all[10,]$data`).

Thus the values of the rows at 1290 ms in the matrix on the right are 1, 1, 3, 4 because this is the count of inactive electrodes in rows 1-4, columns 3-6 of the palatogram shown on the left at that time. A function is now needed similar to the one before to find the minimum value per row of `$data`:

```r
minfun <- function(contacts)
{
  # find the minimum per row
  apply(contacts, 1, min)
}
```

When this function is applied to the data of the 10th segment, the minimum groove widths of the palatograms at intervals of 10 ms between the start and end time of the 10th segment are returned:

```r
minfun(in.sum.all[10,]$data)
1260 1270 1280 1290 1300 1310 1320 1330 1340 1350 1360 1370
  2   1   1   1   1   1   1   1   1   1   2   3
```

This function must now be applied to every segment which can be done using the `by()` function whose output is passed to `buildtrack()` to build a corresponding trackdata object:

```r
groove.min = buildtrack(by(in.sum.all, minfun))
```

A plot of the 10th segment of this trackdata object should give the same values that were returned by `minfun(in.sum.all[10,]$data)`, which is indeed the case (Fig. X.10).

```r
plot(groove.min[10,], type="b", ylab="Minimum groove width", xlab="Time (ms)")
```
Finally, a plot from segment onset to segment offset should show both the differences on this parameter between [s] and [ʃ] and also a progressively decreasing minimum groove width towards the temporal midpoint of the segments, as the fricative's stricture is increased. Such a plot can be produced with `dplot()` and in this example, the 10 fricatives per category are averaged after linear time normalisation:

```r
# Figure X.11
temp = polhom.l %in% c("s", "S")
dplot(groove.min[temp,], polhom.l[temp], normal=T, average=T, ylab="Minimum groove width", xlab="Normalised time", col=c(1, "slategray"), lwd=c(1,2))
```

Evidently, the groove width, on average, decreases towards the temporal midpoint for [ʃ] and somewhat after the temporal midpoint for [s] Fig. X.11 also shows that the groove width for [s] is, on average, well below that of [ʃ] at equal proportional time points from segment onset to segment offset.

### 3.2 Contact distribution indices

As discussed in Gibbon & Nicolaidis (1999), various EPG parameters have been devised for quantifying both the distribution and the extent of tongue palate contacts. Almost all of these are based on some form of summation of the palates (see e.g., Recasens et al., 1993; Pallares & Recasens, 1999; Hardcastle, Gibbon and Nicolaidis, 1991 for details). These are the anteriority index (AI), the centrality index (CI), the dorsopalatal index (DI) and the centre of gravity (COG). The first three of these all vary between 0 and 1 and COG varies between 0.5 and 7.5. The R functions in the EMU library for calculating them are `epgai()`, `epgci()`, `epgdi()`, and `epgcog()` respectively.

The anteriority index quantifies how far forward the contacts are on the palate in rows 1-5. Rows 6-8 are not taken into account in this calculation. AI is especially useful for quantifying the place of articulation back as far as the post-alveolar zone (row 5) and can also be used to quantify the degree of stricture for two consonants at the same place of articulation. The data in Fig. X.11 shows AI for various made-up palatograms. (Details of how to produce these are given in the exercises).

Four general principles involved in calculating AI (Fig. X.12):

1. The further forward the contacts in any row, the higher AI. Thus, the palatogram with the filled row of contacts in row 1 in (a) has a higher AI value than (c) for which the contacts are filled in row 2. AI decreases from 0.9822 (filled row of contacts in row 1)
to 0.05 (filled row of contacts in row 5). Any palatogram with contacts exclusively in rows 6-8 has an AI of 0.

2. Any single contact in row $i$ always has a higher AI than any number of contacts in row $j$, where $i < j$. So the AIs for palatograms (b) and (d) that each have a single contact in row 2 are greater than the AI of palatogram (e) in which all contacts are filled in a lower row number, row 3.

3. The same number of contacts in any row has the same AI irrespective of their lateral distribution (distribution by column). So the fact that the lateral distribution of the single contact is different in palatograms (b) and (d) makes no difference as far as AI is concerned, since both palatograms have a single contact in row 2.

4. The greater the number of contacts, the higher AI – but only up to the limit specified by 2. above. So palatogram (f) which has rows 3-5 completely filled has a higher AI than palatogram (e), in which only row 3 is filled; but since palatogram (f) has no contacts forward of row 3, its AI is lower than those of (b) or (c) that have a single contact in row 2.

The centrality index (CI), as its name suggests, measures the extent of contact at the centre of the palate and varies between 0 and 1. In general, the more the contacts are laterally distributed, the lower the value of CI. This parameter could be used to distinguish between consonants that have a narrow vs. wide central groove, as in the [s,ʃ] fricatives discussed earlier. The actual calculation of CI can be explained in terms of a set of principles that are very similar to those of AI, except that they are based on columns and the relative lateralisation of contacts:

---

Figure X.13 about here
---

1. In the case of a single filled column of contacts, CI is higher nearer the centre of the palate: thus higher for filled columns 4 or 5 (palatograms (b), (e) in Fig. X.13) than for the more laterally filled columns 3 or 6 (palatograms (a), (d)).

2. Any single contact in a given column has a higher CI than a palatogram filled with any number of contacts in more lateral columns. So the CIs for palatograms (g) and (h) which have a single contact in columns 4 and 5 are higher than those of palatograms (a) and (d) in which all contacts are filled in the more lateral columns 3 and 6.

3. The same number of contacts in any column has the same CI irrespective of their distribution by row: thus, palatograms (g) and (h) have the same CI.

4. The greater the number of contacts, the higher CI – but only up to the limit specified by 2. above.

The dorsopalatal index (DI) also varies between 0 and 1 and is a measure of the extent of contact in the last three rows, i.e. in the palatal and post-palatal region. It is a simple
proportional measure: when all 24 electrodes are contacted in rows 6-8, then DI has a value of 1; if 12 are contacted, then DP is 0.5, etc.

Finally, the **centre of gravity index** (COG) is a measure of the extent to which the place of articulation is distributed between the front and back of the palate: further forward/backward places of articulation are associated with higher/lower COG values. COG varies between 7.5 (when row 1 alone is filled) to 0.5 (when row 8 alone is filled). COG is calculated from a weighted average of the sum of contacts in the rows, where the weights on rows 1-8 are 7.5, 6.5…0.5. For example, for palatogram (c) in Fig. X.13, COG is calculated as follows:

```r
# sum of the contacts in rows 1-8 for (c) in Fig. X.14
contacts = c(0, 0, 0, 2, 2, 3, 4, 4)
# weights on rows 1-8
weights = seq(7.5, 0.5, by = -1)
# COG for (c)
sum(contacts * weights)/sum(contacts)
[1] 2.1
```

In Fig 14, (a) and (b) have the same contact patterns, except that in (b) some contacts in the first row are missing. Thus, the overall distribution of contacts is further towards the front in (a) than in (b) and so COG is higher. (c) and (d) have no contacts in the first three rows and so have lower COG values than those of either (a) or (b). Finally (c) and (d) have the same pattern of contacts except that in (d) the last row is filled: consequently the overall distribution of contacts in (d) is furthest towards the back of all the palatograms and so it has the lowest COG of all.

An example of how AI, DI and COG vary is shown for the first two words 'just relax' from the 'coutts' database fragment considered earlier in Fig. X.15 below. AI, DI, and COG for the first two segments are obtained as follows:

```r
ai = epgai(coutts.epg[1:2,])
di = epgdi(coutts.epg[1:2,])
cog = epgcog(coutts.epg[1:2,])
```

The plot in Fig. X.15 is then given by:

```r
par(mfrow=c(3,1)); par(mar=c(1, 2, 1, 1))
plot(coutts.sam[1:2,], type="l", axes=F, ylab="Amplitude") # waveform
axis(side=1)
mtext("Time (ms)", side=1, at=16600, line=-1)
plot(chind(ai, di), type="l", axes=F, col=c(1, "slategray")) # AI and DI
axis(side=2, line=-1)
# some superimposed labels
text(c(16048, 16250, 16434, 16616, 16674), c(0.80, 0.88, 0.60, 0.69, 0.82), c("dZ", "t", "l", "k", "s"))
plot(cog, type="l", axes=F, ylab="COG") # COG
axis(side=1)
times = c(16050, 16250, 16442, 16600, 16650) # mark in some time values
abline(v=times)
```
The contact profiles in Fig. X.15 lead to the following conclusions:

- **AI** is somewhat lower for [l] of 'relax' than either [dʒ] ("dZ") or [st] of 'just' because, in contrast to these segments, [l] has only one contact in the first row, as the palatogram at 16440 ms shows.

- **DI** has a high value during [dʒ] and this is because, as the palatogram at 16050 ms shows, there is quite a lot of contact in the back three rows.

- **COG** often tends to track AI quite closely and this is also evident for the data in figure X.15. However, COG unlike AI, takes account of the overall distribution of the contacts from front to back; and unlike AI, COG is not biased towards giving a higher ranking if there is a single contact in a low row number. Therefore, because the two leftmost palatograms in Fig 15 have contacts in the first row, they have high AI values and higher than those of the 3rd palatogram from the left at 16440 ms during [l]. But of these three, the leftmost palatogram at 16050 ms has the lowest COG because of the large number of contacts in the back rows.

Finally, we will consider how effectively some of these data reduction parameters distinguish between the Polish fricatives considered earlier. For this analysis, the data from the alveolo-palatal [ç] is included as well as from [s,ʃ]. Here again is a greyscale palatographic display, this time averaged over the middle third of each fricative:

```r
par(mfrow=c(1,3))
for(j in c("s", "S", "ç")){
  temp = polhom.l == j
  epggs(dcut(polhom.epg[temp,], .33, .67, prop=T), main=j)
}
```

These greyscale displays in Fig. X.16 can be used to make various predictions about how these three places of articulation might be separated on some of the EPG data reduction parameters:

- **AI**: highest for [s] (greatest number of contacts in rows 1 and 2) and possibly higher for [ç] than for [ʃ] (more contacts in rows 1-2).
- **DI**: highest for [ç] (greatest number of contacts in rows 5-8)
- **CI**: lowest for [ʃ] (least number of contacts medially in columns 4-5)
- **COG**: highest for [s], possibly with little distinction between [ʃ] and [ç] since the distribution of contacts from front to back is about the same for these fricatives.
In the example below, these parameters were calculated across the entire temporal extent of the homorganic fricatives. Since the fricatives were flanked by vowels, then the parameters might be expected to rise towards the temporal midpoint in most cases.

### AE, DI, CI, COG

```r
eai = epgai(polhom.epg);  di = epgdi(polhom.epg)
ci = epgci(polhom.epg);   cog = epgcog(polhom.epg)
```

# logical vector to identify the three fricatives
```r
temp = polhom.l %in% c("s", "S", "c")
```

```r
par(mfrow=c(2,2)); par(mar=c(1,2,1,1))
col=c(1, "gray", "slategray")
dplot(eai[temp,], polhom.l[temp], offset=.5, axes=F, main="AI", col=col)
axis(side=2)
dplot(di[temp,], polhom.l[temp], offset=.5, axes=F, main="DI", col=col)
axis(side=2)
dplot(ci[temp,], polhom.l[temp], offset=.5, axes=F, main="CI", col=col)
axis(side=2)
dplot(cog[temp,], polhom.l[temp], offset=.5, axes=F, main="COG", col=col)
axis(side=1, line=-1); axis(side=2)
```  

Figure X.17 about here

We can see that three of the parameters distinguish one fricative category from the other two: thus DI separates [c] from [s,S], CI separates [S] from [s,c], COG separates [s] from [c,S] while AI produces a clear distinction between all three categories.

4. Analysis of EPG data

We now have the machinery in place to carry out many different kinds of analysis using electropalatographic data. Two common kinds of investigation to which an EPG-analysis is particularly suited are presented in this remaining section: an investigation into the extent of consonant-overlap in alveolar-velar consonant clusters (X.4.1); and vowel-induced place of articulation variation in dorsal fricatives and stops (X.4.2).

4.1 Consonant overlap

The database fragment in this section is part of a larger database that was collected and analysed by Lisa Stephenson (Stephenson, 2003, 2004, 2005; Stephenson & Harrington, 2002) in studying consonant overlap in the production of blends in English and Japanese. In her experiments, subjects saw two hypothetical town names on a screen and had to produce a blend from the two words as quickly as possible after seeing them. They might see for example 'Randon' and 'Pressgate' and the task was to produce a blend by combining the first syllable of the first word with the second syllable of the second word, thus 'Rangate'.

Stephenson's database included a number of blends formed with combinations of /n/ and a following consonant and in the analysis in this section, two of these types will be compared: blends formed with /nk, ng/ with blends formed with /sk, sg/ clusters. No differentiation will be made between the voicing status of the final consonant: so the comparison is between /nK/ vs. /nK/ where /K/ stands for either /k/ or /g/. The question that is addressed is the following: is the extent of alveolar-velar overlap the same in /nK/ and /sK/?
As an initial hypothesis, it is reasonable to expect more overlap in /nK/ for at least two reasons. Firstly because of the well-known tendency for /n/ to assimilate in this context (see e.g., Hardcastle, 1994) whereas /s/ does not audibly retract its place of articulation in e.g., 'mascot' or 'must get' and is often resistant to coarticulatory influences (e.g., Recasens, 2004). Secondly, whereas it is quite possible to sustain an alveolar [n] production when there is tongue-dorsum contact at the velum for [k] or for [g], this type of overlapping or double-articulation is likely to be more difficult in [sk] or [sg]: this is because if there is substantial velar closure during the production of the alveolar, then the airflow through the oral cavity will be inhibited as a result of which it will be difficult to sustain the high aerodynamic power required for the production of the sibilant fricative [s].

The available database fragment is engassim and there are the usual sets of parallel R objects associated with this:

- **engassim**: segment list from the acoustic onset to the acoustic offset of the entire [nk, ng, sk, sg] sequences.
- **engassim.l**: label vector of the above. "nK" for [nk, ng] vs. "sK" for [sk, sg].
- **engassim.w**: label vector of the words from which the sequence were derived.
- **engassim.epg**: parallel EPG trackdata at a frame rate of 5 ms.

We saw in X.3 that the anteriority and dorsopalatal indices tend to provide positive evidence for productions at alveolar and velar places of articulation respectively. The data will therefore be analysed for these parameters, but as with any more complicated parametric analysis, it is always a good idea to look at some samples of the data first. A plot of all of the "nK" data separately per segment and from the onset of the [n] to the offset of the velar can be produced as follows. (Use the left-mouse button to advance through each plot; you will have to do this 17 times, since there are 17 "nK" segments. Use the same commands to get the corresponding "sK" data, but replace `which` with its logical inverse `!which`). The EPG-frames from the first run through the loop (for the first "nK" and "sK" segments) are shown in Fig. X.18:

```r
which = engassim.l == "nK"
for(j in 1:sum(which)){
  # show palate numbers rather than times
  epgplot(engassim.epg[which,][j,, numbering=T)
  # left mouse button to advance
  locator(1)
}
```

-----------------------------

Figure X.18 about here

-----------------------------

For the "nK" sequence on the left of Fig. X.18, the alveolar closure builds up palatographically from row 2; the maximum alveolar closure is complete by palatogram 6 and the release of the alveolar occurs 70 ms later by frame 17. The same display shows how the velar closure begins to form during this interval such that the maximum visible extent of velar closure takes place by frame 16. Evidently then, although the alveolar and velar articulations are not simultaneous i.e. are not completely doubly articulated, they overlap a good deal. Consider now the "sK" data on the left. The alveolar constriction for the [s] extends approximately over 115 ms between roughly palatograms 10 and 23, but the greatest degree of narrowing for the velar stop /k/ does not take place until well after this at frame 31.
We will now see whether there is evidence for a greater extent of alveolar-velar overlap in "sK" in all of the data using anteriority and dorsopalatal indices to parameterise the extent of contact at the front and at the back of the palate respectively:
# anteriority and dorsopalatal indices for all of the data from
# segment onset to segment offset
ai = epgai(engassim.epg); di = epgdi(engassim.epg)

par(mfrow=c(1,2))
which = engassim.l == "nK"
# data for "nK"
dplot(ai[which,], ylim=c(0,1), main="/nK/")
par(new=T); dplot(di[which,], ylim=c(0,1), col="slategray")
# data for "sK"
dplot(ai[!which,], ylim=c(0,1), main="/sK/")
par(new=T); dplot(di[!which,], ylim=c(0,1), col="slategray")

Figure X.19 about here

It is apparent from Fig. X.19 that the tongue-dorsum activity for [k] is timed to occur a good deal earlier relative to the preceding consonant in the clusters with [n] compared with those of [s]. In particular, the left panel of Fig. X.19 shows how the dorsopalatal index rises throughout the AI-plateau for [n]; by contrast for most of the AI-plateau for [s] between roughly 40 ms and 100 ms on the right, there is a dorsopalatal trough.

The differences in the extent of alveolar-velar overlap could be further highlighted by producing greyscale EPG-images at about 50 ms after the acoustic onset of the consonant cluster – which, as Fig. X.19 shows, is roughly the time at which the AI maxima are first attained in /nK/ and /sK/.

par(mfrow=c(1,2))
which = engassim.l == "nK"
epggs(dcut(engassim.epg[which,], start(engassim[which,])+50), main="/nK/")
epggs(dcut(engassim.epg[!which,], start(engassim[!which,])+50),
main="/sK/")

Figure X.20 about here

The greyscale images in Fig. X.20 show greater evidence of alveolar-velar overlap for /nK/ which, in contrast to /sK/ has more filled cells in the last two rows.

4.2 VC coarticulation in German dorsal fricatives

The analysis in this section is concerned with dorsal fricative assimilation in German and more specifically with whether the influence of a vowel on the following consonant is greater when the consonant is a dorsal fricative, which, for compatibility with the MRPA, will be denoted phonemically as /x/, compared with an oral stop, /k/. This analysis was carried out in a seminar at the IPDS, University of Kiel, and then further developed in a paper by Ambrazaitis & John (2004).

In German, a post-vocalic dorsal fricative varies in place of articulation depending largely on the backness of a preceding tautomorphemic vowel. After front vowels, /x/ is
produced in Standard German and many German dialects as a palatal fricative (e.g., [riːç], [lɪ:tʃ], [peç]; ‘riech’/smell, ‘Licht’/light’; ‘pech’/bad luck’ respectively), as a velar fricative after high back vowels (e.g., [bu:x], ‘Buch’/book’) and quite possibly as a uvular fricative after central or back non-high vowels (e.g., [maχ], ‘make’; [lɔx], ‘Loch’/hole’) – see e.g., Kohler, (1995). In his extensive analysis of German phonology, Wiese (1996) raises the interesting point that, while this type of vowel-dependent place of articulation in the fricative is both audible and well-documented, the same cannot be said for analogous contexts with /k/. Thus, there are tautomorphic sequences of /iːk, ik, ekształˈflie̯g’/fly’, ‘Blick’/view’, ‘Fleck’/stain’), and of /uːk, ak/ (‘Pflug’/plough’, ‘Stock’/stick’) and of /ak/ (‘Lack’/paint’), but it not so clear either auditorily nor from any experimental analysis whether there is the same extent of allophonic variation between palatal and uvular places of articulation.

We can consider two hypotheses as far as these possible differences in coarticulatory influences between /x/ and /k/ are concerned. Firstly, if the size of coarticulatory effects is entirely determined by the phonetic quality of the preceding vowel, then there is indeed no reason to expect there to be any differences in the variation of place of articulation between the fricative and the stop: that is, the extent of vowel-on-consonant coarticulation is the same for both, but perhaps the coarticulatory variation is simply much less audible in the case of /k/ because the release of the stop, which together with the burst contains most of the acoustic cues to place of articulation, is so much shorter than in the fricative. The alternative hypothesis is that there is a categorical distinction between the fricative, but not the /k/ allophones. Under this hypothesis, we might expect not only a sharper distinction in speech production between the front and back allophones of the fricative but also that the variation within the front or back allophones might be less for the fricative than the stop.

It is certainly difficult to answer this question completely with the available fragment of the database of a single speaker, but it is nevertheless possible to develop a methodology that could be applied to many speakers in subsequent experiments. The database fragment here is dorsal that forms part of the database recorded by phonetics students at the IPDS Kiel in 2003 and also part of the study by Ambrazaitis & John (2004). The EPG data was recorded at a frame rate of 10 ms at the Zentrum für allgemeine Sprachwissenschaft in Berlin. For the recording, the subject had to create and produce a street name by forming a blend of a hypothetical town name and a suffix that were shown simultaneously on a screen. For example, the subject was shown RIEKEN and –UNTERWEG at the same time, and had to produce RIEKUNTERWEG as quickly as possible after these words were flashed up on the screen. In this example, the underlined part of the blend includes /iːkU/>. Blends were formed in an analogous way to create /V₁CV₂/ sequences where V₁ included vowels varying in backness and height, C = /k, ç/, and V₂ = /i, u/. In all cases, primary stress is necessarily on V₂. To take another example, the subject produced RECHIUSTERWEG in response to RECHEN and –INSTERWEG resulting in a blend containing /ɛxtI/> over the underlined segments.

For the database fragment to be examined here, there are seven different V₁ vowels whose qualities are close to IPA [i, ɪ, e, ø, a, o, u] (MRPA "i", "I", "E", "a", "O", "U" respectively in this database) and that vary phonetically in backness more or less in the order shown. So assuming that the following V₂ = /i, u/ has much less influence on the dorsal fricative than the preceding vowel, which was indeed shown to be the case in Ambrazaitis & John (2004), we can expect a relatively front allophone of the fricative or stop after the front vowels [i, ɪ, e], but a back allophone after [ø, u].

The database dorsal contains the following parallel R-objects that, with the exception of dorsal.bound which marks the time of V₁C acoustic boundary, extend from the
acoustic onset $V_1$ to the acoustic offset of $C$ (the acoustic offset of the dorsal fricative or of the stop closure of /k/):

dorsal  Segment list of $V_1C$ ($C = /k, x/$)
dorsal.epg  EPG-compressed trackdata of dorsal
dorsal.sam  sampled waveform trackdata of dorsal
dorsal.fm  Formant trackdata of dorsal
dorsal.vlab  Label vector of $V_1$ ("i", "I", "E", "a", "O", "U")
dorsal.clab  Label vector of $C$ ("k" or "x")
dorsal.bound  Event times of the acoustic $V_1C$ boundary

There were 2 tokens per /$V_1CV_2/$ category (2 tokens each of /i:kI/, /i:kuI/, /i:xi/, /i:xiu/… etc.) giving 4 tokens for each separate $V_1$ in $V_1k$ and 4 tokens for $V_1x$, (although since $V_1$ was not always realised in the way that was intended - e.g., /i/ was sometimes produced instead of /i:/ - there is some deviation from this number, as table(label(dorsal)) shows). In order to be clear about how the above R objects are related, Fig. X.21 shows the sampled waveform with superimposed VC boundary and electropalatographic data over the first segment in the database which (as dorsal.l[1] shows) was "ak":

```r
plot(dorsal.sam[1,], type="l", main=paste("/", dorsal.vlab[1], dorsal.clab[1], sep="", ", "/"), ylab="Amplitude", xlab="Time (ms)")
abline(v=dorsal.bound[1], lwd=2, col="slategray")
epgplot(dorsal.epg[1,], lwd=2, col="slategray")
```

Figure X.21 about here

For the investigation of the variation in place of articulation in dorsal consonants, the anteriority index is not appropriate because this only registers contact in rows 1-5. The dorsopalatal index might shed more light on place of articulation variation – however, given that it is based on summing the number of contacts in the back three rows, it is likely to distinguish between the lesser stricture of the fricatives than the stops. But this is not what is needed. Instead, we need a parameter that is affected mostly by shifting the tongue from front to back along the palate and which does so in more or less the same way for the fricative and the stop categories.

The parameter that is most likely to be useful here is the EPG centre of gravity which should show decreasing values as the primary dorsal stricture moves back along the palate. COG should also show a predictable relationship by vowel category. It should be highest for a high front vowel like [i:] that tends to have a good deal of contact laterally in the palatal region and decrease for [i,ε] where there is still palatal contact but it is weakened. It should have the lowest values for [u,ɔ] in which, to the extent that any tongue-palate contact shows up at all, contact it expected at the back of the palate.

The EPG-COG parameter should show some relationship to the vowel's second formant frequency, since $F2$ of [i:] is higher than $F2$ of [i,ε] and since of course $F2$ of front vowels is greater than $F2$ of low, central and back vowels. These relationships between COG, vowel category and $F2$ can be examined during the interval for which sensible formant data is available, i.e., during the voiced part of the vowel. Given that the interest in this analysis is in the influence of the vowel on the following consonant, we will consider data extracted at the vowel-consonant boundary close the vowel’s last glottal pulse, i.e. close to
the time point at which the voiced vowel gives way to the (voiceless) fricative or stop. We will also consider two different types of COG. In one, COG is calculated as in section X.3 over the entire palate: in the other, which I will call the posterior centre of gravity (P-COG), the COG calculations are restricted to rows 5-8. P-COG is relevant for the present investigation because we are dealing exclusively with sounds made in the dorsal region, i.e., with vowels followed by dorsal consonants. It should be mentioned at this point that this version of P-COG is not quite the same as the one in Gibbon & Nicolaidis (1999) who restrict the calculations not only to rows 5-8 but also to columns 3-6 (see the picture on the jacket cover of Hardcastle & Hewlett, 1999), i.e. to a central region of the palate. However, this parameter is likely to exclude much of the information that is relevant in the present investigation, given that the distinction between high front and back vowels often shows up as differences in lateral tongue-palate contact (present for high front vowels, absent for back vowels), i.e. at the palatographic margins.

The relationship between the centre of gravity parameters and F2 at the acoustic vowel offset is shown in Fig. X.22 which was created with the following commands:

```r
# COG and PCOG, from the onset to the offset of VC
cog = epgcog(dorsal.epg); pcog = epgcog(dorsal.epg, rows=5:8);
# COG and PCOG at the VC boundary
cog.voffset = dcut(cog, dorsal.bound)
pcog.voffset = dcut(pcog, dorsal.bound)
# F2 at the VC boundary
f2.voffset = dcut(dorsal.fm[,2], dorsal.bound)
par(mfrow=c(1,2))
plot(f2.voffset, cog.voffset, pch=dorsal.vlab, xlab = "F2 (Hz)", ylab = "COG")
plot(f2.voffset, pcog.voffset, pch=dorsal.vlab, xlab = "F2 (Hz)", ylab = "PCOG")
```

As Fig. X.22 shows, both COG and PCOG show a fairly linear relationship to the second formant frequency at the vowel offset, as well as a clear separation between vowel categories, with the low back vowels appearing at the bottom left of the display and the high and mid-high front vowel in the top right. For this particular speaker, these relationships between acoustic data, articulatory data and vowel category emerge especially clearly. It must be emphasised that this will not always be so for all speakers! PCOG shows a slightly better correlation with the F2-data than COG (as `cor.test(f2.voffset, pcog.voffset)` show) but then COG shows a clearer distinction within the front vowel categories [i; I; e] – and this could be important in determining whether the coarticulatory influences of the vowel on the consonant are more categorical for /x/ than for /k/ (in which case, we would expect less variation in /x/ following these different front vowels, if /x/ is realised as basically the same front allophone in all three cases). In the subsequent analysis we will work with COG – some further analysis with PCOG is given in the exercises.

----------------------------
Figure X.22 about here
----------------------------

In order to get some insight into how /k,x/ vary with the preceding vowel context, we will begin with a plot of COG 30 ms on either side of the vowel boundary. This is shown in Fig. X.23 and was produced as follows:
# cut the EPG-data to +/-30 ms either side of V,C boundary
epg30 = dcut(dorsal.epg, dorsal.bound-30, dorsal.bound+30)
# calculate COG
cog30 = epgcog(epg30)

# logical vector that's true when the consonant is /k/ as opposed to /x/
which = dorsal.clab=="k"
ylim = c(0.5, 3.5); par(mfrow=c(1,2))
col = c(1, "slategray", "slategray", 1, 1, "slategray")
linetype=c(1,1,5,5,1,1) ; lwd=c(2,2,1,1,1,1)
dplot(cog30[which,], dorsal.vlab[which], offset=.5, ylim=ylim,
legend="topleft", ylab="EPG COG", main="/k/", col=col, linetype=linetype,
lwd=lwd)
dplot(cog30[!which,], dorsal.vlab[!which], offset=.5, ylim=ylim, legend=F,
main="/x/", col=col, linetype=linetype, lwd=lwd)

Figure X.23 about here

As Fig. X.23 shows, there is a clearer separation (for this speaker at least) on this parameter for /x/ compared with /k/ following the front vowels [i, ɛ, ɪ] on the one hand and the non-front /a, œ, u/ on the other. A histogram plot of the EPG-COG values at 30 ms after the acoustic VC boundary brings out the differences quite clearly.

# COG values at 30 ms after the VC boundary. Either:
cog30end = dcut(cog30, 1, prop=T)
# or
cog30end = dcut(cog30, dorsal.bound+30)

# logical vector, T when clab is /k/, F when clab is /x/
which = dorsal.clab=="k"
par(mfrow=c(1,2))
# Histogram of EPG-COG 30 ms after the VC boundary for /k/
hplot(cog30end[which], main="/k/", xlab="EPG-COG at t = 30 ms")
# as above but for /x/
hplot(cog30end[!which], main="/x/", xlab="EPG-COG at t = 30 ms")

Figure X.24 about here

There is evidently a bimodal distribution of EPG-COG 30 ms after the VC boundary for both /x/ and /k/, but this is somewhat more pronounced for /x/ such a finding is consistent with the view that there may be a more marked separation into front and non-front allophones for /x/ than for /k/. In order to test this hypothesis further, the EPG-COG data are plotted over the extent of the consonant (over the fricative or the stop closure) in Fig. X.25:

# Centre of gravity from acoustic onset to offset of the consonant
cogcons = epgcog(dcut(dorsal.epg, dorsal.bound, end(dorsal.epg)))
# logical vector that's True when dorsal.clab is "k"
which = dorsal.clab=="k"
par(mfrow=c(1,2)); ylim = c(0.5, 3.5)
There is once again a clearer separation of EPG-COG in /x/ depending on whether the preceding vowel is front or back. Notice in particular how EPG COG seems to climb to a target for /ex/, to reach a position similar to that for /ix/ and /ik/.

For this single speaker, the data does indeed suggest a greater categorical allophonic distinction for /x/ than for /k/.
Exercises

A. Create a 3D palatographic array for creating the figure in Fig. X.12, then plot Fig. X.12 and use the made-up array to verify the values for the anteriority index.

B. Write R commands to plot the 1\textsuperscript{st}, 4\textsuperscript{th}, and 7\textsuperscript{th} palatograms at the acoustic temporal midpoint of \[c\] (MRPA "c") in the \texttt{polhom} database fragment.

C. The database fragment \texttt{coutts2} contains the same utterance produced by the same speaker as \texttt{coutts} but at a slower rate. The R-objects for \texttt{coutts2} are:

\begin{verbatim}
coutts2          segment list of words
coutts2.l        vector of word labels
coutts2.epg      EPG-compressed trackdata object
coutts2.sam      Trackdata of the acoustic waveform
\end{verbatim}

Produce palatographic plots over a comparable extent as in Fig. X.4 from the /d/ of 'said' up to the release of 'k' in 'said Coutts'. Comment on the main ways the timing of /d/ and /k/ differ in the normal and slow database fragments.

D. For the \texttt{engassim} database fragment, the AI and DI were calculated as follows:

\begin{verbatim}
ai = epgai(engassim.epg); di = epgdi(engassim.epg)
\end{verbatim}

Calculate (a) the time at which AI first reaches a maximum value (b) the time at which DI first reaches a maximum value, i.e. the two times shown in Fig. X.26 but for every segment:

\begin{verbatim}
\end{verbatim}  

By making a boxplot of the difference between these times, (b) – (a), show that the duration between these two maxima is greater for "sK" than for "nK".

Answers

A.

\begin{verbatim}
palai = array(0, c(8, 8, 8))
palai[1,2:7,1] = 1
palai[2,4,2] = 1
palai[2,,3] = 1
palai[2,8,4] = 1
palai[3,,5] = 1
palai[3:5,,6] = 1
palai[4,,7] = 1
palai[5,,8] = 1
class(palai) = "EPG"
aivals = round(epgai(palai), 4)
epgplot(palai, mfrow=c(1,8), numbering=as.character(aivals))
\end{verbatim}
B.  
# EPG data at the midpoint
polhom.epg.5 = dcut(polhom.epg, 0.5, prop=T)
# EPG data at the midpoint of "c"
which = polhom.l == "c"
polhom.epg.c.5 = polhom.epg.5[which,]
# plot of the 1st, 11th, 15th "c" segments at the midpoint
epgplot(polhom.epg.c.5[c(1,4,7),], mfrow=c(1,3))

-----------------------------
Figure X.27 about here
-----------------------------

C.  
epgplot(coutts2.epg, xlim=c(14840, 15010))

-----------------------------
Figure X.28 about here
-----------------------------

The main difference is that /d/ is released (at 14190 ms), well before the maximum extent of
dorsal closure is formed (at 14935 ms), i.e., the stops are not doubly articulated.

D.  
# function for calculating the time at which the maximum first occurs
targ <- function(dat)
{
times = tracktimes(dat)
which = dat==max(dat)
times[which][1]
}

ai = epgai(engassim.epg); di = epgdi(engassim.epg)
aimax = by(ai, targ, simplify=T); dimax = by(di, targ, simplify=T)
boxplot(dimax - aimax ~ factor(engassim.l))
Some references (incomplete)


Fig. X.1: The palate of the EPG3 system in a plaster cast impression of the subject's upper teeth and roof of the mouth (left) and fixed in the mouth (right). Pictures from the Speech Science Research centre, Queen Margaret University College, Edinburgh, [http://www.qmuc.ac.uk/ssrc/DownSyndrome/EPG.htm](http://www.qmuc.ac.uk/ssrc/DownSyndrome/EPG.htm). Bottom left is a figure of the palatographic array as it appears in R showing 6 contacts in the second row. The relationship to phonetic zones and to the row (R1-R8) and column (C1-C8) numbers are also shown.
Fig. X.2: Schematic outline of the relationship between electropalatographic objects and functions in R.
Fig. X.3a: Palatograms of 'said Coutts' showing the times (ms) at which they occurred.
Fig. X.3b: Waveform over the same time interval as the palatograms in Fig. X.3a. The vertical lines mark the interval that is selected in Fig. X.4.
Fig. X.4: Palatograms over the interval marked by the vertical lines in Fig. X.3b.
Fig. X.5: Palatograms for 10 [s] (left) and 10 [ʃ] (right) Polish fricatives extracted at the temporal midpoint from homorganic [s#s] and [ʃ#ʃ] sequences produced by an adult male speaker of Polish.
Fig. X.6: Greyscale images of the data in Fig. X.5. The darkness of a cell is proportional to the number of times that the cell was contacted.
Fig. X.7: Sum of the contacts in rows 1-3 (black) and in and rows 6-8 (gray) synchronised with an acoustic waveform in 'said Coutts' produced by an adult female speaker of Australian English. Some phonetic landmarks are superimposed on the top trajectory.
Fig. X.8: A boxplot of the minimum groove width shown separately for palatograms of Polish [s,ʃ] shown in Fig. X.5. The minimum groove width is obtained by finding whichever row over rows 1-5, columns 3-6 has the fewest number of inactive electrodes and then summing these.
Fig. X.9: Palatograms between the acoustic onset and offset of a Polish [s]. On the right is the number of inactive electrodes for each palatogram in rows 1-4. The count of inactive electrodes for the palatogram 1290 ms is highlighted.
Fig. X.10: Minimum groove width between the acoustic onset and offset of a Polish [s].
Fig. X.11: Minimum groove width between the acoustic onset and offset of Polish [s] (black) and [ʃ] (gray) averaged after linear time normalisation.
Fig. X.12: Palatograms with corresponding values on the anterioty index shown above.
Fig. X.13: Palatograms with corresponding values on the centrality index shown above.
Fig. X.14: Palatograms with corresponding centre of gravity values shown above.
Fig. X.15: Synchronised waveform (top) anteriority index (middle panel, black), dorsopalatal index (middle panel, gray), centre of gravity (lower panel) for 'just relax'. Some palatograms that occur closest to the time points marked by the vertical lines in the lower panel (in \([d\ddot{z}]\) and \([t]\) of 'just' and \([l], [k], [s]\) of 'relax' respectively) are shown below.
Fig. X.16: Greyscale images for 10 tokens each of the Polish fricatives [s,ʃ,ɕ].
Fig. X.17: Anteriority (AI), dorsopalatal (DI), centrality (CI), and centre of gravity (COG) indices for 10 tokens of the Polish fricatives [s,ʃ,ɕ] (black, pale-gray, dark-gray respectively) synchronised at their temporal midpoints.
Fig. X.18: Palatograms from the acoustic onset to the acoustic offset of /nk/ (left) in the blend 'duncourt' and /sk/ (right) in the blend 'bescan' produced by an adult female speaker of Australian English.
Fig. X.19: Anteriority (black) and dorsopalatal (gray) indices for 17 /nK/ (left) and 15 /sK/ sequences (K= /k, g/) produced by an adult female speaker of Australian English.
Fig. X.20: Greyscale EPG images for the /nK/ (left) and the /sK/ (right) data from Fig. X.19, 50 ms after the acoustic segment onset of the cluster.
Fig. X.21: Acoustic waveform (top) of /ak/ produced by an adult male speaker of standard German and the corresponding palatograms at 10 ms intervals.
Fig. X.22: The EPG centre of gravity (left) and EPG posterior centre of gravity (right) plotted as a function of F2 for data taken at the acoustic vowel offset for data pooled across /x/ and /k/.
Fig. X.23: the EPG centre of gravity calculated 30 ms on either side of the acoustic $V_1C$ boundary for /k/ (left) and /x/ (right) shown separately as a function of time by $V_1$ category.
Fig. X.24: EPG-centre of gravity for /k/ (left) and /x/ (right) at the V₁C boundary.
Fig. X.25: EPG-COG data over the extent of /k/ closure (left) and /x/ frication (right) shown by vowel category and synchronised at the consonants’ acoustic temporal midpoints.
Fig. X.26: AI (black) and DI (gray) for one of the segments from the *engassim* database fragment.
Fig. X.27: Three palatograms at the temporal midpoint of [ɛ].
Fig. X.28: Palatograms over the /dk/ interval of 'said Coutts' spoken at a slow rate.