

Beyond Sound

A Gentle Introduction
to The Emu Speech Database System
CoEDL Summer School

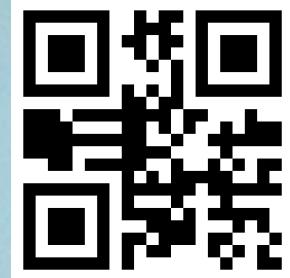
Hywel Stoakes

The University of Auckland
The University of Melbourne

4th December 2019

Before we start...

Links to Rstudio cloud



- [https://rstudio.cloud/spaces/42005/join?
access_code=%2Fp3fDUIHqM3hwVBL9npun0TejimJ7g98DW](https://rstudio.cloud/spaces/42005/join?access_code=%2Fp3fDUIHqM3hwVBL9npun0TejimJ7g98DW)
- You will need an Rstudio cloud account if you
haven't signed up already?

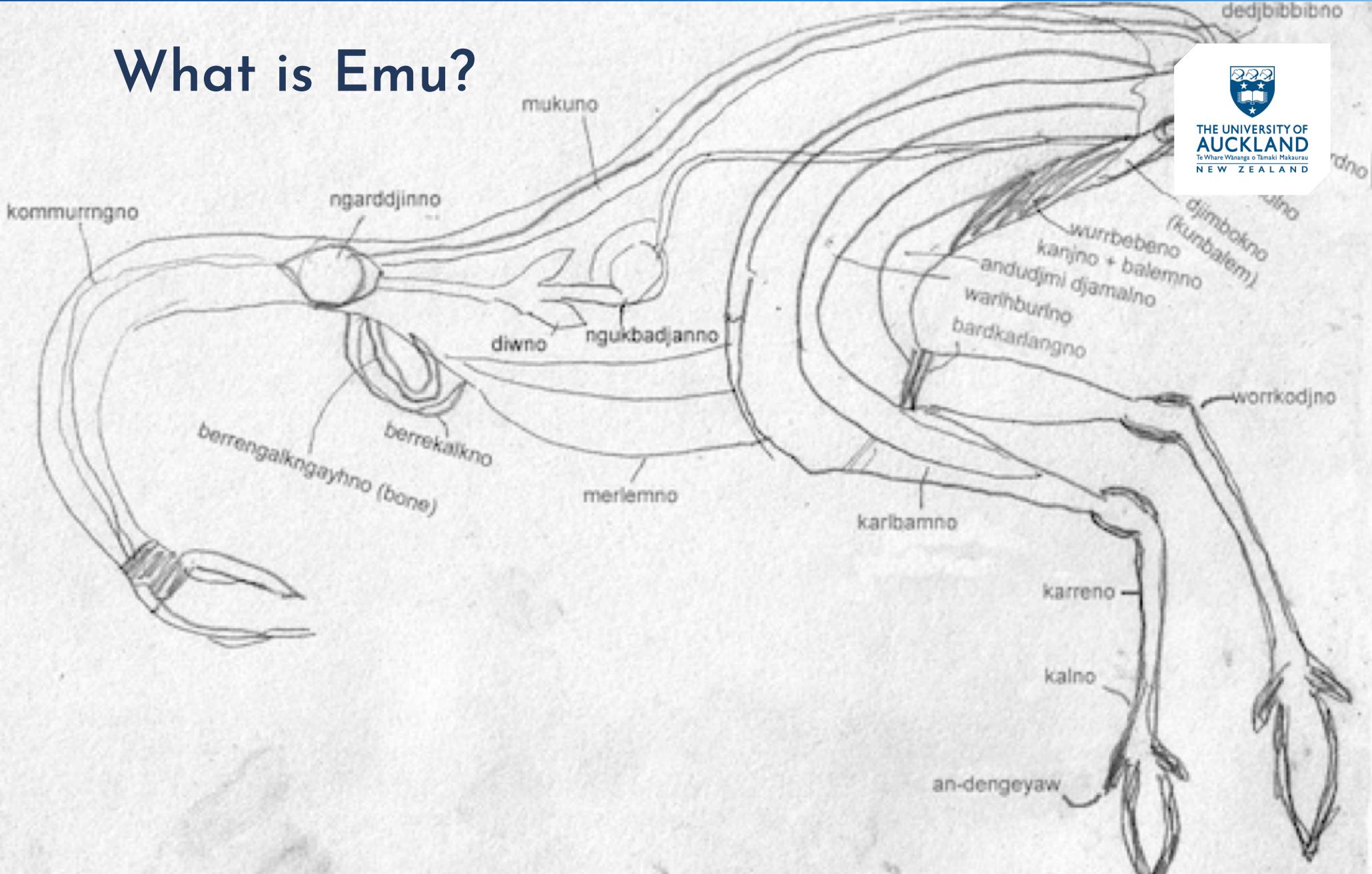


After this course we:

- will have an understanding of the structure and application of the Emu Speech Databases to linguistic problems.
- will have an understanding of how to structure large speech corpora for new projects.
- will be able to convert existing databases and corpora into Emu Speech Databases.
 - e.g. Praat textgrids, ELAN XML files, plaintext

What is Emu?

What is Emu?



THE UNIVERSITY OF
AUCKLAND
Te Whare Wananga o Tamaki Makaurau
NEW ZEALAND

What is Emu?



- A flightless bird native to the Australian Continent (Sahul).
Related to the Northern Australian Cassowary and the New Zealand Moa.
- **A tool to efficiently manage speech files with time aligned annotations and access advanced statistical analyses**

More Specifically?



- Emu is a speech database management system:
- Developed by Raphael Winkelmann at IPS, Munich (Winkelmann et al. 2017, 2019)
- Evolved from The Emu Speech Database (mu+) developed at Macquarie Uni and extended by Lasse Bombien and Tina John (Bombien et al. 2006).
 - *Bombien, L., S. Cassidy, J. Harrington, et al. (2006). "Recent developments in the Emu Speech Database System". In: Proceedings of the 11th Speech Science and Technology Conference Auckland. , pp. 313–316.*
 - *Winkelmann, R, J. Harrington, and K. Jansch (2017). "EMU-SDMS: Advanced speech database management and analysis in R". In: Computer Speech & Language. ISSN: 0885-2308. DOI: <http://doi.org/10.1016/j.csl.2017.01.002>. URL: <http://www.sciencedirect.com/science/article/pii/S0885230816302601>.*



THE UNIVERSITY OF
AUCKLAND
Te Whare Wananga o Tamaki Makaurau
NEW ZEALAND

Linguistic Annotation

- An important concept in linguistic annotation is that a speech signal has metadata and linguistic annotation attached to it
- It is important that the *Text* and *Signal* stay grouped together somehow.
- Researcher often achieve this by using the same filename for the *text* and *signal* digital files.
- Emu uses a bundle concept to keep these things together (we will revisit this later).

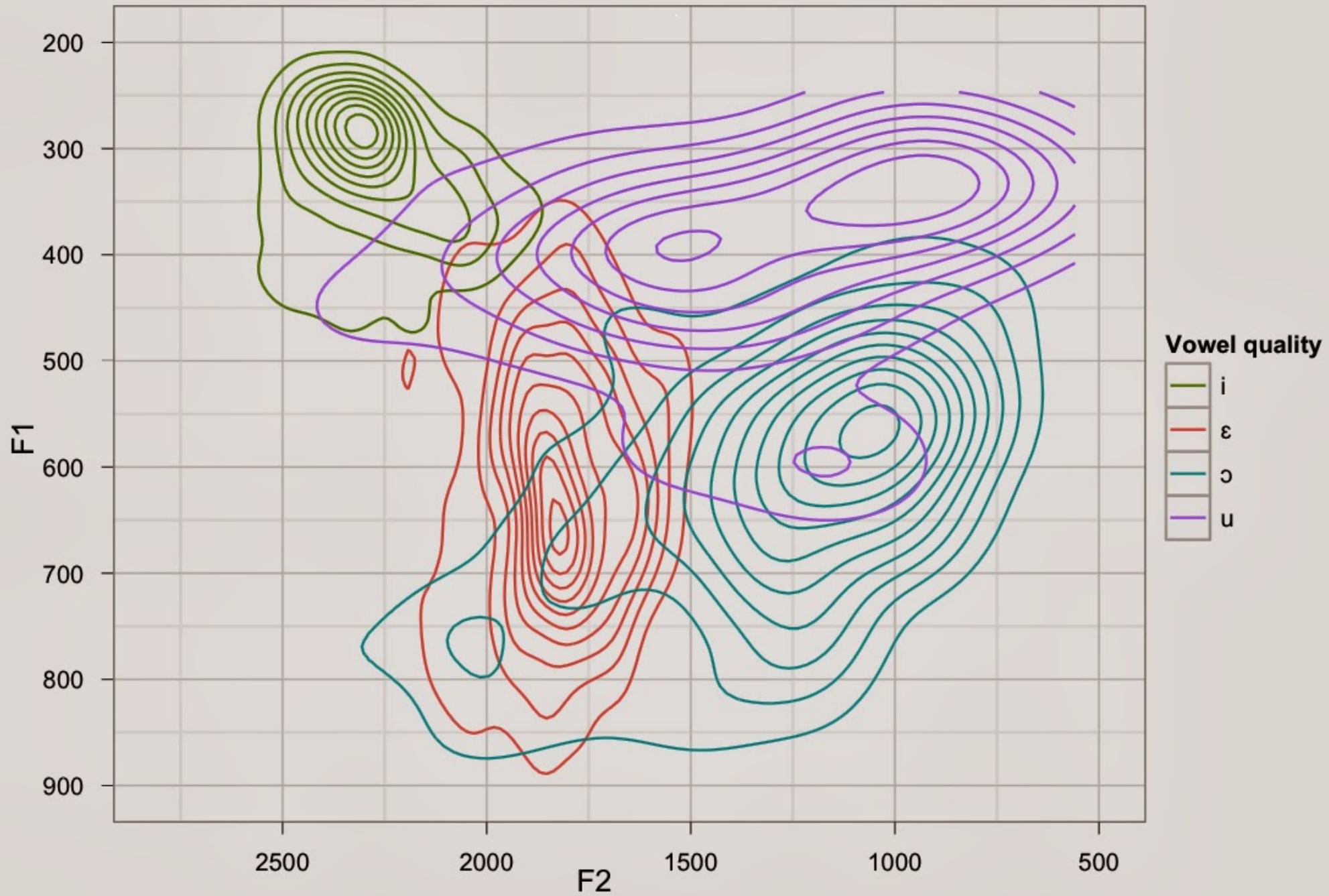
Why use Emu?



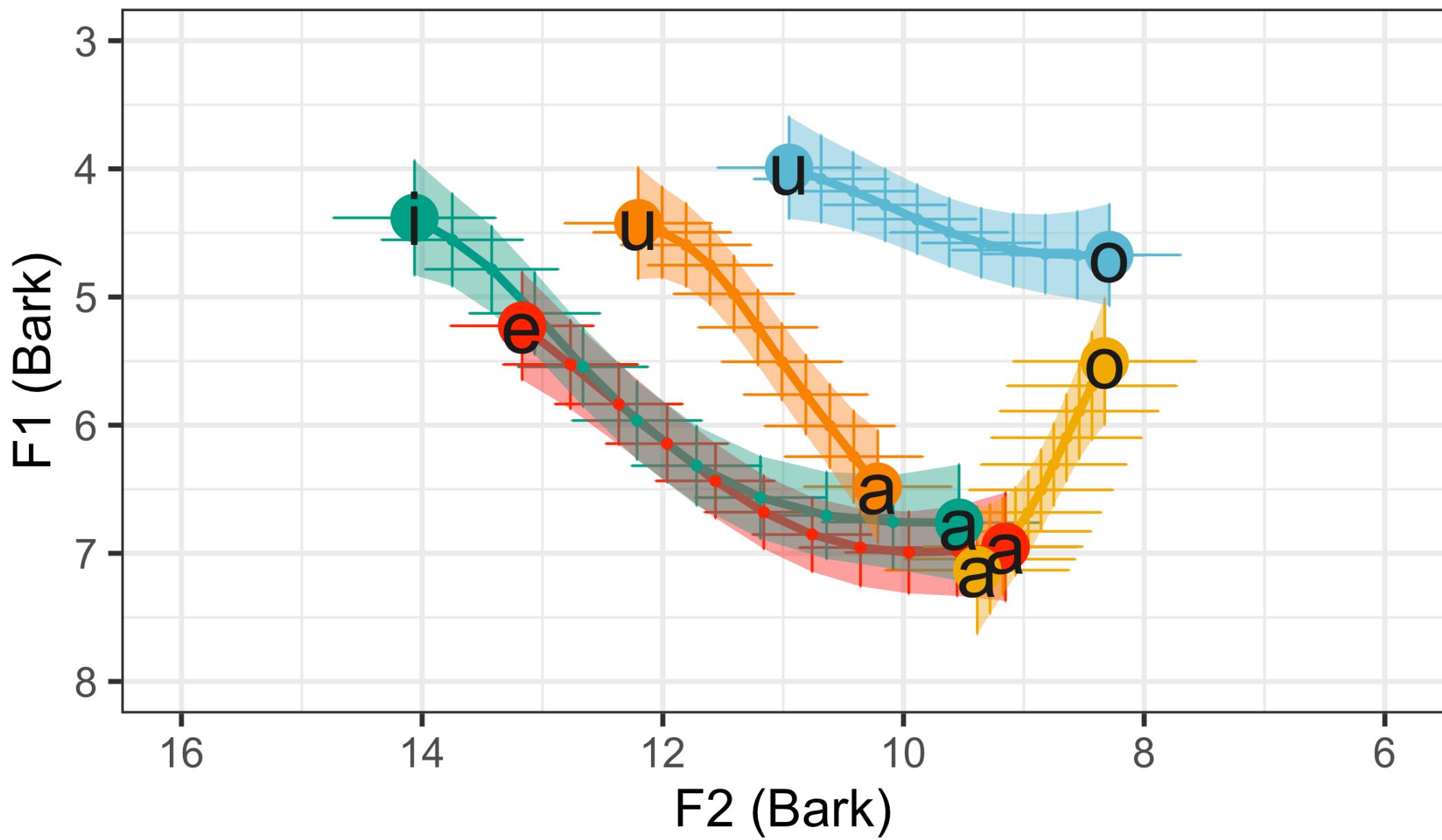
- Fast and powerful way to process and analyse speech files
 - particularly if they are part of large corpora
- Based in a popular language environment, R with a growing userbase
- Uses an established Database format (SQL) with a querying language (EQL)
- Linear and Hierarchical analyses of speech annotation
- Direct access to powerful statistical tools in R
- Access to powerful signal processing tools (wrassp) and praat

"but you promised us pretty pictures..."





Younger Female Speakers



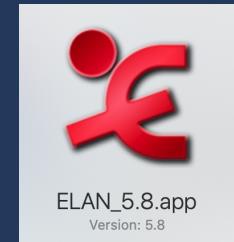
What we will learn in this workshop.

- How to visualise speech files in Emu The Emu Webapp
- A quick introduction to using the visualisation and signal processing tools within RStudio
- How to import existing annotations
- How to setup a database in **EmuR**
- Best practise for naming files



Other tools that occupy the same space

- ELAN
 - Hierarchical - no internal phonetic analysis tools

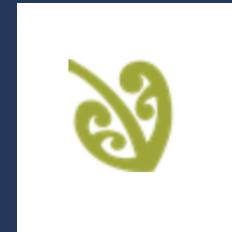


ELAN_5.8.app
Version: 5.8



Other tools that occupy the same space

- LABBCAT
 - Database software - no internal phonetic analysis tools



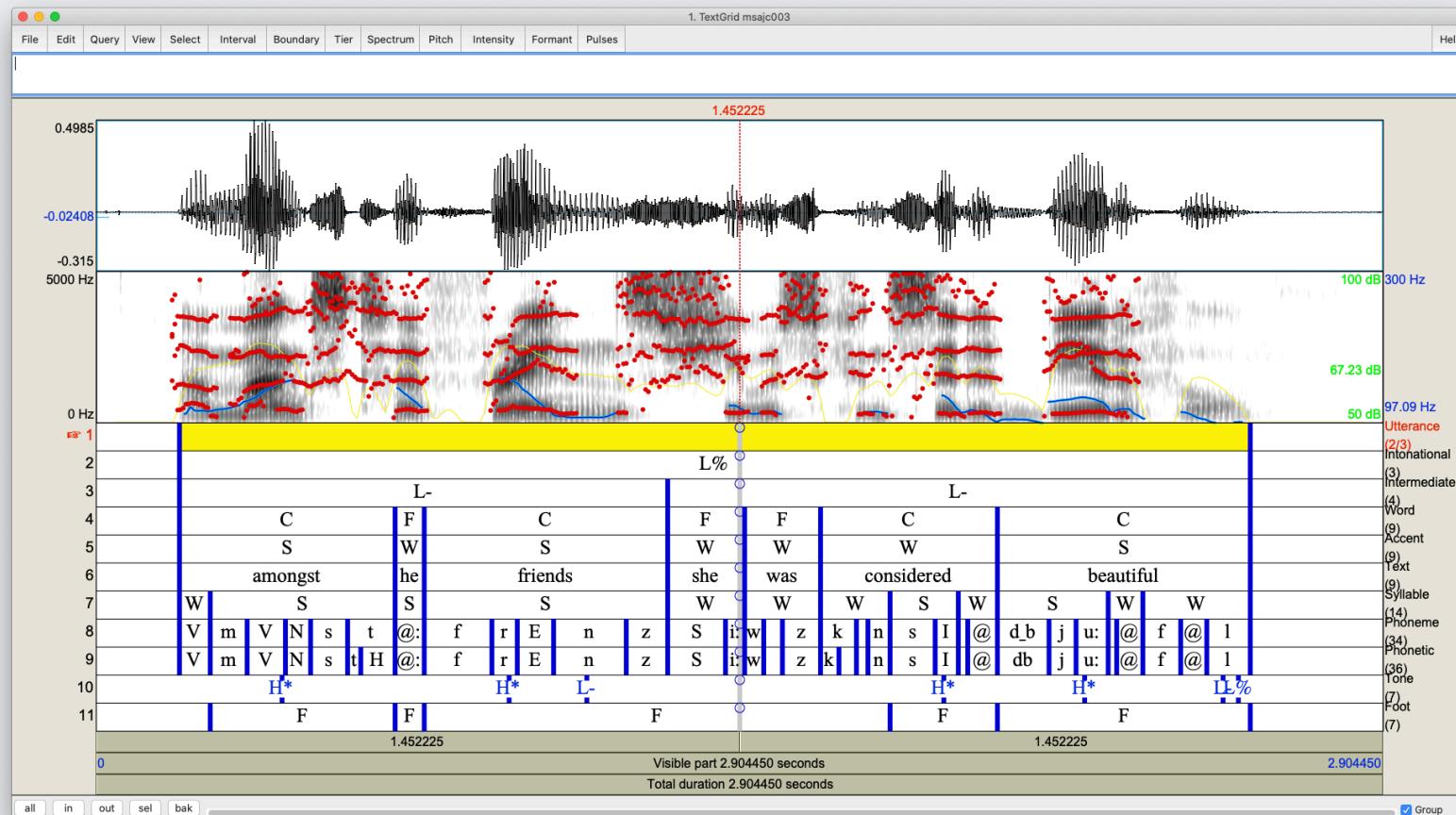


Other tools that occupy the same space

- Praat
 - widely used annotation format
 - difficult to organise large amounts of data



- Many of you will have used Praat www.praat.org.



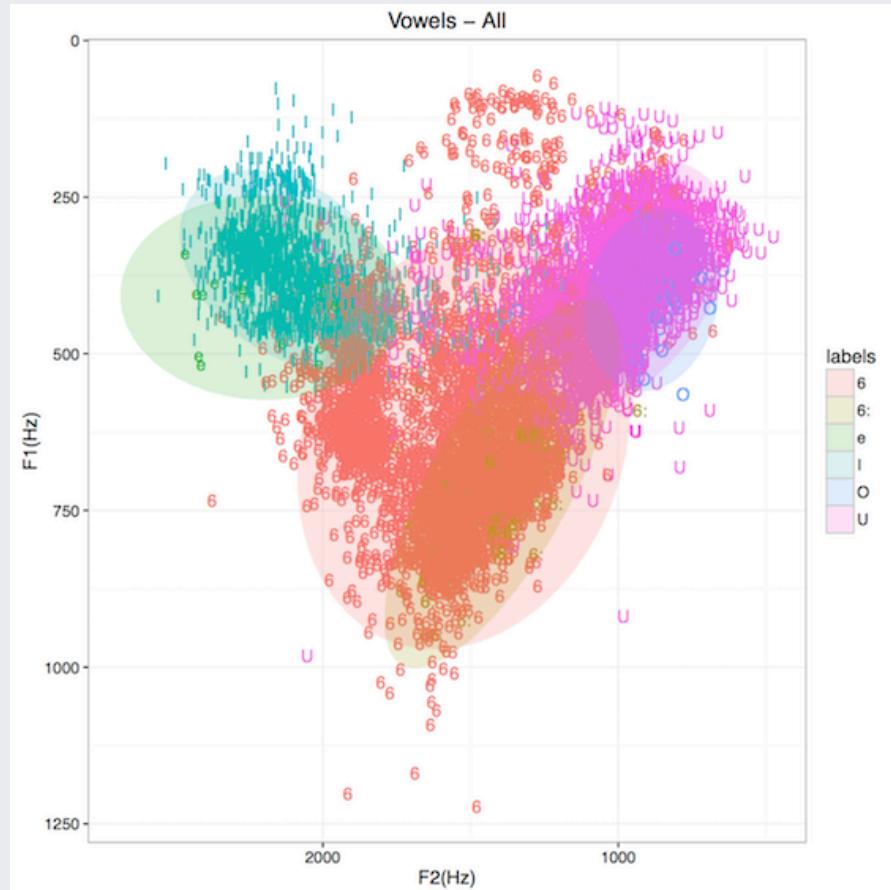
Comparisons between Praat and Emu:



- What does Emu do, that Praat does not?
- *Unlike Praat*: Emu has a database structure at its heart
 - Enables hierarchical querying
- *Like Praat*: Emu is Platform Independent
- **Emu** is browser Based (works best with **Chrome** or **Firefox**) or directly inside Rstudio
- *Unlike Praat*: Emu interfaces easily with **R** via Rstudio

What are the further advantages of Emu?

1. Visualisation



2. Hierarchical structure

- important for prosodic research

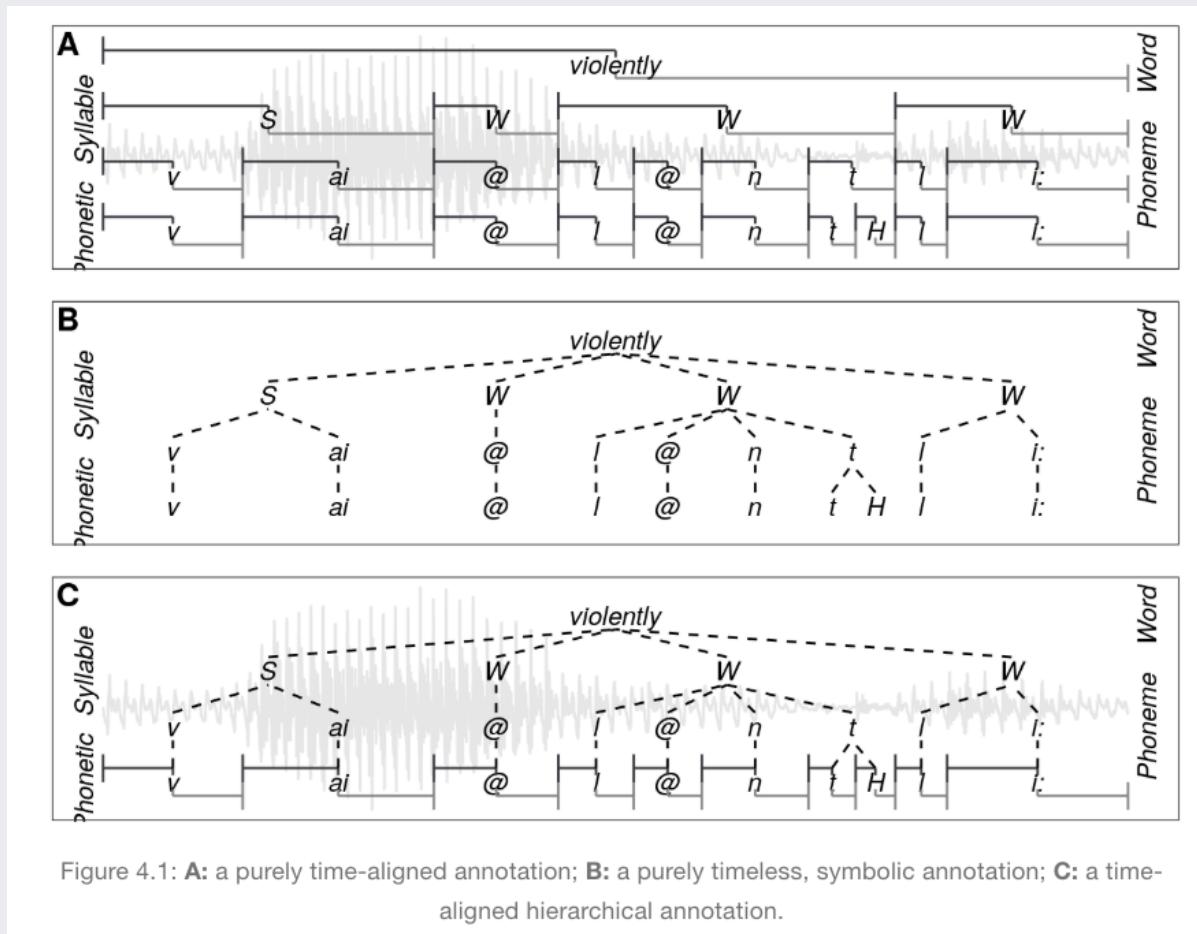


Figure 4.1: **A:** a purely time-aligned annotation; **B:** a purely timeless, symbolic annotation; **C:** a time-aligned hierarchical annotation.

3. Realtime annotation

- If you change something within the labeller this will reflect in your analysis instantly
 - This is particularly useful for refining a data set in readiness for training a speech recognition model
 - e.g. Kaldi using Elpis

4. Other advantages

- *Text* Can import from other annotation formats such as Praat
- *Signal* Due to the power of R it is possible to pipeline data to other tools such as opensmile or even praat or any Python library.

Time to try something
interactive!

Simple Importing from Praat TextGrids

- Navigate to: <http://ips-lmu.github.io/EMU-webApp/>
- Drag and Drop a pair of files (wav and textgrid) examples/
- These can be up to an hour long depending on quality of audio
 - (there is a limit of 2Gb per tab in Chrome)

Emu Web App: annotation

- Open a web-browser (ideally Chrome-based) (<https://www.google.com/intl/en/chrome/>)
- Navigate to: <http://ips-lmu.github.io/EMU-webApp/>
- See help in the Emu Web App
- Full set of Keyboard shortcuts
- A new version is being developed but feature requests are welcomed 😊
 - may only find a way into the next version 😔

Options

- Spectrogram options



Options

- Spectrogram options



The screenshot shows the 'OSCI/SPEC Settings' page of the EMU-webApp. On the left, there's a sidebar with project names: 'msajc003' and 'msajc010'. The main area has a blue header 'OSCI/SPEC Settings'. Below it, under 'OSCI Settings', is a section for 'Current channel' with a dropdown menu set to '0'. Under 'SPEC Settings', there's a 'View Range (Hz)' section with input fields 'From: 0' and 'To: 5000'. Below that is a 'Window Size (seconds)' section with a note: 'resulting number of samples 200 zero-padded to 512 (min. = 512) 0.01'. At the bottom, there's a 'Dynamic Range for Maximum (dB)' section with a dropdown menu set to '70'. At the very bottom of the page, there are 'Cancel' and 'Save' buttons.



Limitations of the Webapp

- No access to speech analysis tools
 - can't see formats without some additional setup
 - can't see F0 or intensity
- No direct access to statistical and visualisation tools within R

Break Time!

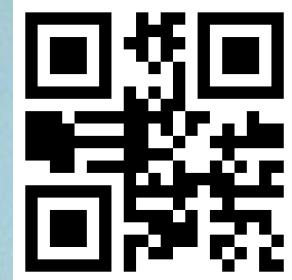
Part Two...



Advanced: Installing R, Rstudio

- To install R - go to: <https://cran.ms.unimelb.edu.au>
 - choose your platform (Windows, OSX, Linux)
- To install Rstudio go to: Download Rstudio
 - choose your platform (Windows, OSX, Linux)

Links to Rstudio Cloud Project



- <https://rstudio.cloud/project/233426>

Installing: EmuR

Once R and Rstudio are installed:

- Download the Summer School Files from:

<https://github.com/Hywel-Stoakes/EmuRSummerSchool>



Installing: EmuR

Once R and Rstudio are installed:

- Navigate to downloads folder and open the EmuRSUMMERSchool.Rproj



Installing: EmuR

Once R and Rstudio are installed:

```
install.packages("emuR")
```

To load the library, use:

```
library("emuR")
```



Quickstart using Emu in R:

- Create a demo emuDB



Either

Local

```
create_emuRdemoData(dir = file.path("DATA")) #local
```

on RStudio Cloud:

```
create_emuRdemoData(dir = file.path("/", "cloud", "project", "DATA
```

Serve a database



- construct path to demo emuDB

```
path2ae = file.path("DATA", "emuR_demoData", "ae_emuDB") #local  
path2ae = file.path("/cloud/project/DATA", "emuR_demoData", "ae
```

- serve the database

```
ae = load_emuDB(path2ae)  
serve(ae)
```

- Note: This command doesn't work RStudio cloud

Querying in R



- Query the emuDB for all /n/ segments of the "Phonetic" annotation level

```
sl = query(ae, "Phonetic = n")
```

An example to plot vowels in two-dimension

```
require(tidyverse)
# query A and V (front and back open vowels),
# i:and u: (front and back closed vowels), and
# E and o: (front and back mid vowels)
# Labels are in machine readable ASCII (could be IPA or any oth
ae_vowels = query(emuDBhandle = ae,
                  query = "[Phonetic = V | A | i: | u: | o: |
```



Querying in R

- Take a look at `ae_vowels` by typing the object name into the console or clicking on the object in the Environment.

Calculating formants using wrassp



```
ae_formants = get_trackdata(ae,  
                           seglist = ae_vowels,  
                           ssffTrackName = "fm"  
)
```

Calculating formants using wrassp



```
# time normalize the formant values  
ae_formants_norm = normalize_length(ae_formants)
```

- extract the temporal mid-points
- the %>% operator comes from the magrittr package “Ceci n’est pas une pipe”

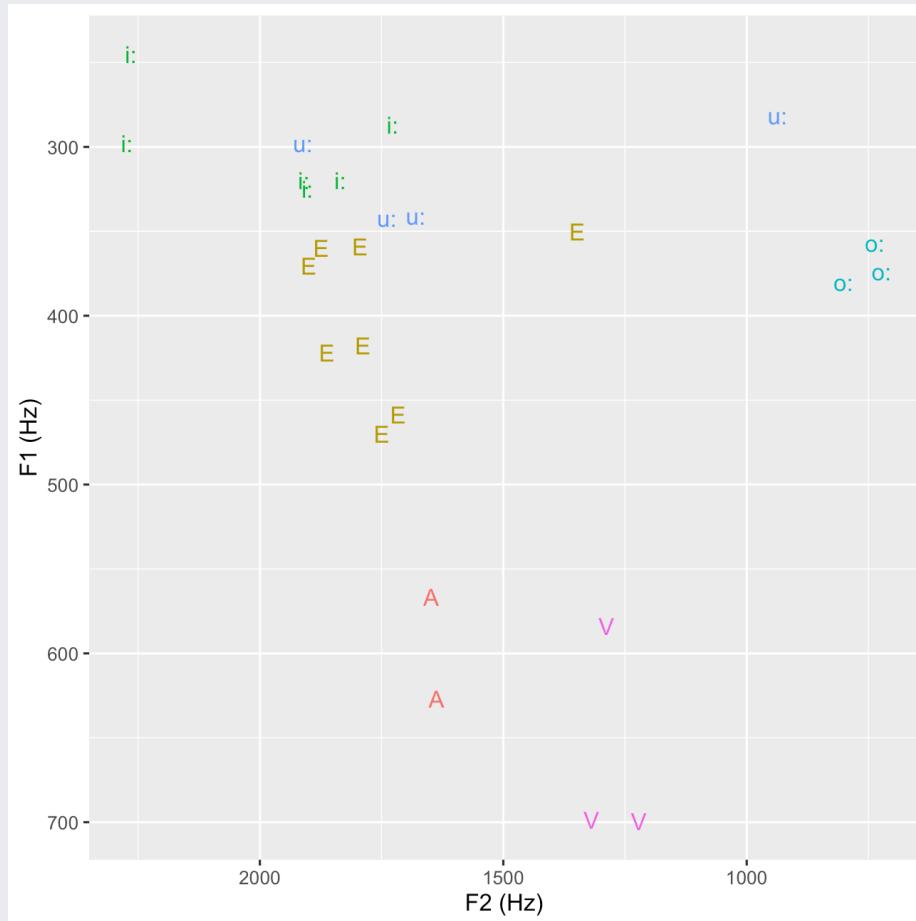
```
ae_midpoints = ae_formants_norm %>%  
  filter(times_norm == 0.5)
```

Plotting Formant Values



```
# plot F1 & F2 values (= eplot( ... , dopoints = T, doellipse =  
ggplot(ae_midpoints) +  
  aes(x = T2, y = T1, label = labels, col = labels) +  
  geom_text() +  
  scale_y_reverse() + scale_x_reverse() +  
  labs(x = "F2 (Hz)", y = "F1 (Hz)") +  
  theme(legend.position = "none")
```

Plotting Formant Values



Data Preparation

Have all your data in the same format

This holds for any collection of audio data:



- Audio files
 - WAV files 44.1 kHz 16bit are most portable
 - Use of mp3 files is discouraged for acoustic analysis
- Annotation files
 - plain text files (TextGrid or JSON)
 - Tools to convert between files are more available but always check!

Working Files



- Your master audio files may be recorded at archival quality (96 kHz, 24bit)
 - these can be downsampled and split into smaller stretches of audio
- Any edits/splits should be recoverable (keep the master files separate and notes of all extractions)
 - The chronological sequence can be put into the names of the files
 - Could use timecodes as well (less informative)
- Use version control such as git

Importing Existing Files



- These slides are based on material from Winkelmann's (2019) Emu SDMS Manual <https://ips-lmu.github.io/The-EMU-SDMS-Manual/chap-tutorial.html#converting-the-textgrid-collection>.
- Data can be imported from a variety of formats

Importing Textgrids



The current import routines provided by the emuR package are:

- `convert_TextGridCollection()` - Convert TextGrid collections (.wav and .TextGrid files) to the emuDB format,
- `convert_BPFCollection()` - Convert Bas Partitur Format (BPF) collections (.wav and .par files) to the emuDB format,
- `convert_txtCollection()` - Convert plain text file collections format (.wav and .txt files) to the emuDB format,
- `convert_legacyEmuDB()` - Convert the legacy EMU database format to the emuDB format and
- `create_emuDB()` followed by

Importing Textgrids

This can be used to import files from ELAN or TextGrid collections from Praat.



```
# convert TextGrid collection to the emuDB format
convert_TextGridCollection(dir = tg_col_dir,
                            dbName = "my-first",
                            targetDir = tempdir(),
                            tierNames = c("Word", "Syllable",
                                         "Phoneme", "Phonetic"))
```

Naming files

Based on Slides by Jenny Bryan (The Reproducible Science Workshop)



File names



BAD

- myabstract.docx
- Joe's Filename Using Spaces and Punctuation.xlsx
- figure 1.png
- fig 2.png

GOOD

- 2014-06-08_abstract-for-sla.docx
- joes-filenames-are-getting-better.xlsx
- fig01_scatterplot-talk-length-vs-interest.png
- fig02_histogram-talk-attendance.png

File names



Three principles for naming files:

- Machine Readable
- Human Readable
- Works well using the ordering rules in the operating system

Naming Files for use in Speech Research



- Information including:
 - speaker
 - repetition
 - word
 - date/time
 - location
- Are all candidates for inclusion in a file name
- Be sure they are not too long though as some online syncing apps such as One-Drive have a limit to the overall file path!

Metadata in an Emu Database



- Much of this **Metadata** information can actually be located within an Emu Database
 - these fields are called *attribute definitions* and can be placed as a sub-category of the highest level in the database (usually the utterance)

section: Database construction

subsection: Converting Existing Databases to Emu

Annotation data can be converted from various formats:

- Praat TextGrid

Thanks!