An Efficient Workflow for Collecting, Entering, and Proofing Field Data: Harnessing Voice Recording and Dictation Software

Rachel H. Toczydlowski
Department of Botany, University of Wisconsin-Madison, 430 Lincoln Drive, Madison, Wisconsin 53706 USA

Introduction

Anyone who has spent meaningful time in the field appreciates the amount of time and resources spent collecting, transcribing, and proofing data. Here, I describe a data collection, entry, and proofing workflow that only requires one person, is highly efficient, and uses freely available software. I was inspired to design this pipeline while collecting quantitative trait data regularly on thousands of field-grown *Impatiens capensis* Meerb. plants with limited time and personnel.

Traditional techniques

Collecting data solo is often slow due to the time spent continually switching between data collection and recording tasks, often thousands of times in a session. This is especially true when two hands are required to make measurements, and/or there is no convenient location to place a datasheet (e.g., when doing fieldwork standing upright, in the rain, or in a tree). To eliminate the time (and mental fatigue) required to switch tasks, two people commonly work as a team. One person takes the measurements and calls them aloud to a partner, who writes the data on a printed datasheet.

After collection, data must be converted into a manipulatable format. This usually involves manually entering data from the hardcopy into a spreadsheet or other digital form. Recording data directly into a computer eliminates the time required to transcribe the handwritten datasheets and to proof this transcription, but mistakes are more likely to be made typing than writing, and environments may be too extreme for laptop or tablet computers (Johnson et al. 2009). For these reasons, most field data are still recorded on paper first and then converted to a manipulatable digital format.

The final step is data proofing, which like collection, requires two people to be efficient. Two common methods are read-aloud proofing and double-entry proofing (Cummings and Masten 1994, Kawado et al. 2003). Briefly, in read-aloud proofing, one person reads the data aloud from the digital copy, and the other visually validates these data against the original paper datasheet. In double-entry proofing, two people independently enter the data, and the two digital copies are compared to each other to catch entry inconsistencies.

The workflow

Overview

The following pipeline eliminates the need for a partner at the data collection and proofing steps, and fully automates the data entry step. This means data can be collected as efficiently with one person as with a team of two (or, equivalently, twice as much data can be collected in the same period of time).
This pipeline relies on a digital audio recording device and free downloadable software. The overall schema is as follows (Fig. 1):

1. A single person collects data and records it in a digital audio recording.
2. The audio file is played on a computer and the data are automatically transcribed into a spreadsheet using dictation software.
3. A single person listens to the original audio file while looking at the transcribed data, correcting any mistranscriptions.

This pipeline will likely work best for recording one type of data at a time on many individuals—each with a specific ID and arranged in a known pattern, or occurring within a defined plot where re-identifying specific individuals within the plot is not necessary, for example, measuring the height of each plant in a common garden grid, recording all of the species present within a plot, or recording the diameters of all saplings present within a plot.

Fig. 1. Schematic comparison of collecting, entering, and proofing field data using a traditional method (Panel a) versus a voice recording method (Panel b). Traditionally, efficient field data collection required two people to collect the data, one person to manually enter the data into a computer, and two people to proof the data entry. In the voice recording method, one person collects and states the data aloud, and a smartphone or other digital recording device records the data in an audio file. Free transcription software then enters the data into a computer and plays the original audio recording back for one person to proof the automatic data entry. Note the fewer number of human hours needed to collect, enter, and proof the same data in Panel b, compared to Panel a. Original data are shown in orange and electronic copies (e.g., .csv file) in blue.
Collecting and recording data

Start a digital audio recording, state important information in the beginning of each recording like plot ID and the ID of the first individual or sample measured in that recording, and then call data aloud, recording it into the audio file. Save and label audio files on the recording device and in backup locations (e.g., cloud storage or in an email) when service allows.

A smartphone is ideal for recording data for this workflow. Most smartphones have built-in audio recording software and high-quality microphones, and recorded files can be quickly backed up and emailed. I fashioned an elastic band to wear around my chest that held the phone securely, positioned the microphone close to my mouth, and allowed me free use of both hands. Smartphone armbands would also likely work well, especially in environments with minimal background noise. A judgment call must be made about how often to state only the data versus including the sample ID, and how many data points to include in each recording. Stating the sample ID often before the measurement adds robustness to the dataset, but those statements will be transcribed along with the rest of the data, and will likely require additional downstream work to remove from the dataset. Stating the sample ID each time also takes time. I was measuring plants in common garden arrays where each plant had a specific ID. I stated the ID of the first plant measured, the ID of about every 20th plant that I measured, and the ID of the last plant that I measured in a recording. Splitting the data into many recordings limits the chance of losing lots of data if a recording does not save or becomes corrupted, but also takes time. These judgment calls will ultimately reflect the feasibility of recollecting data and the users’ confidence in themselves and their recording equipment.

Entering data via automated transcription

Split a computer’s sound card into two channels (using a freely available audio routing software). Play an audio recording of data as the output from an audio player and the input to a dictation program while a document (e.g., spreadsheet or text file) is open. The audio file plays internally in the computer, and the dictation program automatically transcribes the audio into type and enters the data into cells in a spreadsheet or lines of a text file.

Soundflower (Ingalls 2004; for mac) and PulseAudio (Poettering et al. 2014; for Windows and Linux) are two open source audio routing programs that are available for free download. This paper is not meant to provide an exhaustive list of software options or instructions for using each. Multiple programs exist that can accomplish the tasks laid out in this section. Here, I outline the software that I used to run this pipeline on a mac running macOS Sierra (V 10.12.4) to give one complete example. The website “www.alternativeto.net” is an especially useful search engine to find related software and alternatives to a given program if these programs become unsupported or do not work on your operating system (The AlternativeTo Team 2014).

I use Soundflower (V 2.0b2; Ingalls 2004), Dictation (from macOS Sierra 10.12.1; The Apple Team 2014), and Audacity® (V 2.1.0; The Audacity Team 2014) in concert to automatically transcribe data from the audio recording into a spreadsheet.

Software summary.—Dictation comes automatically installed on mac operating systems and converts speech into text. Usually, a user turns on Dictation and speaks into their computer’s microphone to
convert their spoken words into text in real time in an open document. Instead of speaking into the computer’s microphone, we will pipe our audio recording into Dictation, which will automatically transcribe our data into a spreadsheet. I use Audacity to remove loud-consistent background noise from an audio recording, to stitch multiple audio recordings together into one continuous recording, and to play an audio file of recorded data. Audacity is free, open source, software that runs on Mac, Windows, and Linux platforms (http://www.audacityteam.org/). Finally, I use Soundflower, free, open source audio routing software, to pipe the audio from Audacity to Dictation (https://github.com/mattingalls/Soundflower).

**Use Soundflower to set up audio routing.**—After downloading and installing Soundflower, open Audio MIDI Setup.app (Applications folder → Utilities folder). Soundflower (2ch) should now appear in the left-hand list of audio devices. Create a multi-output device (plus symbol in lower left), and select Built-in Output and Soundflower (2ch) as outputs (place a check next to each under the “use” column). Quit Audio MIDI Setup.

Open Dictation (System Preferences → Keyboard → Dictation), and select Soundflower (2ch) from the drop-down menu below the microphone symbol. Dictation should be “On” and “Use Enhanced Dictation” should be checked. Note how to open Dictation later under “Shortcut.” You can also set custom Dictation commands, so that when you say a specific word, Dictation performs a specific action. For example, if you are recording multiple measurements of conductivity in an experimental tank, you could make a Dictation command that moves to the next line every time you say the word “tank.” Dictation commands are listed under System Preferences → Accessibility → Dictation (from the categories on the left) → Dictation commands… Use the plus symbol to add custom commands. Quit System Preferences.

Open Audacity, and change “Playback Device” to Soundflower (2ch). Playback Device is the last of four drop-down menus within the Audacity window and is usually initially set to “Built-in Output.”

**Use Audacity to stitch multiple audio files together.**—Multiple audio files can be stitched together into one to eliminate having to open and dictate many shorter recordings. With a blank slate open in Audacity, select all audio files you wish to combine (in Finder) and drag them into the Audacity window. If you do not open them in this exact way, each file will open in a separate window of Audacity and they cannot be easily combined. If opened correctly, all of the files should be open within the same Audacity window and stacked on top of one another. Select the double-headed arrow tool (upper left menu area). Click and hold on the beginning of each audio recording and drag them so that the clips are staggered one after another within their own boxes, and none of the audio files overlap with one another. To stitch all of the recordings together and export this as one audio file: File → Export Audio → click OK to all prompts, and name your file when prompted. The default format of WAV works well.

**Use Dictation to transcribe data into a spreadsheet.**—Open the saved audio file you wish to dictate from (likely the combined file created in the previous section) in Audacity. Unlike in the previous section, if you are using a combined WAV file, you cannot drag it into the Audacity window to open the file. The best way to open the file is to “right click open” it through Finder. You will be prompted if you want to open the original or a copy. The original is faster but any edits you make are final and automatically applied to the file itself. I always use the faster original option, as the combined file is not the actual original copy of my voice data. It is helpful to add some silence to the beginning of the track before dictating so you have time to get everything running before actual data are played. Use the
double-headed arrow tool like in the previous section to drag the audio to the right, adding silence to
the start of the track (usually 10–30 second is sufficient). Open and start the Dictation software using
the shortcut you noted earlier when working in System Preferences. A small microphone should pop
up in the lower right-hand corner of the computer screen.

Go to the Audacity window and start playing the audio file to be dictated (green play arrow). If
needed, use the pause button to stop and start (the play button takes you back to the very beginning of
the recording). After starting the audio recording, quickly open a spreadsheet and select the blank cell
you would like to dictate the data into. It is helpful to position the Audacity window and spreadsheet
windows so that you can see the audio recording in Audacity while the spreadsheet is open, to track
progress. Do not click anywhere else now within the spreadsheet or select another window. Once the
playback plays through the initial silence that you added, you should start to see your data being trans-
scribed into the cell. There may be a few seconds delay to start, but if after 5–10 seconds you do not see
any data, something is wrong. Sometimes going back to Audacity and Dictation and reselecting “Sound-
flower (2ch),” or just stopping and starting the recording can fix this. You should not be able to hear any
of the audio, as it is being played internally from Audacity into Dictation via Soundflower. Do not click
off of the cell in Excel and/or into another program, as this will confuse and stop the Dictation software;
the file you are dictating into must be live/active and on top. Unless you have set up a special custom
Dictation command, all of the data will be transcribed, in order, into the one cell you selected. Hitting
“enter” once and awhile in the spreadsheet during pauses between the data will split the data up into
multiple cells. I am unsure the maximum number of characters that fit into a single spreadsheet cell, but
I have found that Dictation can transcribe 500–900 data points into one single cell in Microsoft Excel.

When the recording is finished playing, stop dictation (click “Done” below the dictation microphone),
and rearrange your data so it is in a vertical format with one data point per cell. To do this, select the cell
with all of the transcribed data, under the Data tab, click “Text to Columns,” then click “Next,” select
“Space” as the delimiter, and then click “Finish.” This should split the data into individual cells, one data
entry per cell, arranged horizontally. To transform the cells into a vertical alignment, select the entire
row with the newly delimited data, copy, paste special, and select transpose. Paste the data into a ver-
tical format in the final desired location. Repeat these data reformatting steps for each cell that contains a
string of transcribed data (if you split the transcription into multiple cells as it was running by occasion-
ally hitting “enter”). Be sure to maintain the original transcription order when copying and pasting data
in this step. Clean transcribed sample ID information (e.g., “plant 501”) from the data.

Proofing the transcribed data

Remove excess silence between data in the audio recording using Audacity. Play the truncated audio
recording of the original data while visually comparing to the final digital copy of the transcribed data.
Correct any transcription errors that Dictation made.

Open the audio file that the data were transcribed from (likely a larger file of stitched together record-
grings) in Audacity. Removing long pauses in the recording between data makes the data proofing process
considerably faster, as data are played in quick succession instead of the real time in which they were
collected. To remove excess silence in Audacity, select Effect menu → Truncate silence → Truncate
detected silence → Level: −25, Duration: 1 second, Truncate to: 0.6 second → OK. To choose the
decibel level that constitutes silence, play a portion of an audio recording of data and look at the decibel meter in Audacity during the portions that data are not being said. Play back the shortened audio file using Audacity and visually proof data entered by Dictation while listening to the recording. Check that sample ID numbers, when stated in the audio recording, match the sample ID you assigned in the final transcribed digital spreadsheet.

Conclusions

Data collection, entry, and proofing are all time-consuming tasks and often require multiple people to complete efficiently. Here, I have presented a data collection, entry, and proofing workflow that only requires one person and is actually faster than traditional two-person methods. The workflow relies on recording data into audio files, automatically transcribing them into a digital form using dictation software, and proofing the transcription visually while listening to the original audio recording. I used a smartphone and pre-installed or free open source software on a mac platform. Equivalent software for additional platforms can be found using software search engines like http://alternativeto.net/.

In addition to saving time and resources, the overall speed of the entire process can result in a cleaner final dataset. When collecting data during intensive field seasons or laboratory experiments, often a pile of datasheets are accumulated, and data are only entered and proofed after the season or experiment has ended. Because this pipeline makes it feasible to enter data quickly after it is collected, errors and holes in the data can be identified and often corrected. For example, one could compare flowering stages of plants in a common garden to flowering stages collected two weeks previously. When possible errors and inconsistencies are identified (e.g., a plant marked “flowers senesced” the first time and “flowers in bud” the second time), a list of potential errors can often be compared to the actual “truth on the ground” just hours or days after data collection, potentially preventing gaps or uncertainty in data.

This method is not without drawbacks. For one, this method relies on electronics and technology to store data. There is no paper datasheet, which many researchers value. I advise using this method when confidence in equipment is high and it is feasible to recollect data if necessary. I also advise researchers to bring a paper datasheet into the field with them still—for noting additional comments next to specific samples or in case recording equipment fails. Dictation software may also initially make more mistakes than a human entering the data, but if the recording is clear and loud, I had minimal issues—about one mistranscription every 500 data points. The time saved in automatically transcribing the data and only needing one person to proof the entry was still considerable compared to time spent correcting errors made by Dictation. Finally, two-person data collection systems can provide an important safety net in the field and laboratory. This method does allow efficient collection with one person, but if safety is a concern, two people can still work in the vicinity, recording data independently via two audio devices, and accomplish twice the work.

This paper is meant to provide an example workflow and inspire efficiency in data collection. I have provided one specific set of tools and the basic idea. I hope that researchers will use this as a launch point to develop similar pipelines for their specific data tasks. In many cases, the specific tools and workflow that I have provided can be directly implemented. As mentioned earlier, this specific pipeline works best for collecting one type of data on many individuals when switching between collecting data and writing data is inefficient and/or must be done hundreds or thousands of times. Time saved in handling data is time to spend analyzing data and writing up results.
Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. (DGE-1256259) awarded to R. Toczydlowski. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Support was also provided by the Graduate School and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison with funding from the Wisconsin Alumni Research Foundation. A Flora Aeterna Research Award through the University of Wisconsin-Madison Botany Department funded the direct research that enabled the development of these methods. I thank the University of Wisconsin-Madison Arboretum for providing the field site where this method was developed, and for assistance in the field. I thank D. M. Waller for helpful edits and S. Friedrich (UW-Madison, The Botany Media Studio) for figure support. Finally, I thank B. Waldorf, A. de la Cruz, and others for their contributions in the field. The research that led to the development of this method was conducted under University of Wisconsin-Madison Arboretum Research Permit 2015-15.

Literature Cited

The AlternativeTo Team: crowdsourced software recommendations. 2014. http://alternativeto.net/
The Apple Team. 2014. Dictation, part of macOS Sierra. Apple Inc., Cupertino, California, USA. https://www.apple.com/macos/sierra/
Audacity Team. 2014. Audacity(R): free Audio Editor and Recorder. http://audacity.sourceforge.net/
Cummings, J., and J. Masten. 1994. Customized dual data entry for computerized analysis. Quality Assurance 3:300–303.
Ingalls, M. 2004. Soundflower. Cycling ‘74. https://github.com/mattingalls/Soundflower
Johnson, C. L., G. M. Temple, T. N. Pearsons, and T. D. Webster. 2009. An evaluation of data entry error and proofing methods for fisheries data. Transactions of the American Fisheries Society 138:593–601.
Kawado, M., S. Hinotsu, Y. Matsuyama, T. Yamaguchi, S. Hashimoto, and Y. Ohashi. 2003. A comparison of error detection rates between the reading aloud method and the double data entry method. Controlled Clinical Trials 24:560–569.
Poettering, L., P. Ossman, S. E. King, T. Kaskinen, C. Gutherie, A. Raghavan, and D. Henningsson. 2014. PulseAudio. https://github.com/pulseaudio/pulseaudio