Prerequisites:

- Batchalign is a command line program, meaning you will be interacting with the shell of your machine to execute commands. The shell is the functionality in your "terminal" window. We will write down shell commands for you to execute. To execute them, you must first locate your platform's terminal.
    - On MacOS: https://tinyurl.com/OSXterminal
    - On Windows: https://tinyurl.com/PCterminal
    - Follow your Distribution's Instructions for Linux
- To use Batchalign, you first have to install a version of Python great that 3.9. You can install the current version from https://python.org/downloads. Some external dependencies will be **installed automatically** when you install Batchalign. This means that, you should not install any dependencies (like Whisper or Nvidia Nemo) manually. If a special function requires a manual dependency, the Batchalign program will prompt you with instructions.
    - To install Batchalign, follow the instructions at https://github.com/talkbank/batchalign2 for Quick Start
    - Remember that the commands we provide are meant to be executed in the terminal which you located in the step above.

Updating: Batchalign's update and install instructions are almost identical, although you don't need to install Python again. To update simply follow the instructions in the section at github.com/talkbank entitled "Install and Update the Package".

Usage: Once installed, the usage of Batchalign follows these steps. These instructions supersede the earlier descriptions in our 2023 article in JSLHR.

1. At your user root level of ~/ you should create a folder using: *mkdir ba_data* (or whatever name you prefer). Then use *cd ba_data* to go inside that folder and create subfolders using: *mkdir input* and *mkdir output*.
2. Next, you will need to prepare your audio or video file and put it inside the ~/ba_data/input folder. Batchalign only works on .wav and .mp3 files. If you have another format, you could either use a third-party converter such as Amadeus Pro, Audacity, or Video Converter to create .wav or you could install FFmpeg to your computer and Batchalign will convert automatically. For information on installing FFmpeg, see below.
3. You can put as many files as you wish into your input folder, and they will be processed in sequence. If your machine has enough memory and multiple processor cores, as with the M2 Apple, you can even create multiple input and output folders to run multiple jobs in parallel.
4. Batchalign supports different processes with different verbs. The three most used are marked with an asterisk:
    - ***align** produces utterance- and word-level alignment of a text when you place both the media and transcript files into /input. If utterance bullets are present, Batchalign will use them even if they are wrong, potentially worsening the alignment of the whole file; hence, it is best to first remove current bullets using

this CLAN command: _chstring -cbullets.cut *.cha +_1 unless you are sure the bullets are absolutely correct. It uses the @Languages line in the transcript to detect what language model to use.

- ***morphotag** uses Stanza, following Universal Dependencies, to add %mor and %gra lines to a transcript. This function does not require a media file. It uses the @Languages line in the transcript to detect what language model to use.
- ***transcribe** provides transcription directly from audio or video. This only requires raw media files (audio or video) in /input.
- **clean** empties the input and output folders.
- **version** lists the version of batchalign.
- **benchmark** compares ASR output with human transcription in /input

5. Each of the command verbs has some additional switches that modify usage. To see the complete list of switches for a given command, use the --help flag following the verb. For instance, for the transcribe verb, you would type the command in this format: batchalign transcribe --help. Understanding these additional switches is particularly important for the transcribe verb.
6. You can use either Whisper or Rev-AI for transcription. The default mode for English uses Rev-AI. For this, you will need to open a rev.ai account. Rev-AI provides you with 6 free hours for your new account. Charges are $.02/minute of audio for this service. Go to rev.ai, sign up, and on the left side of your dashboard, you will find a tab called Access Token. Click generate to generate a new token, copy and paste the key to somewhere you can find later. If you want to comply with IRB rules against sending data to third parties, you can configure your Rev-AI service to auto-delete your data after processing, as described in this screenshot.
7. In addition to Rev.AI, we have adjusted Whisper, a local ASR model, to perform nearly as well as Rev-AI. Whisper has a wider potential coverage of languages than Rev-AI. For example, Cantonese only works on Whisper. Alsok Whisper seems to be better than Rev-AI for Spanish. However, for all of the other languages that it supports, Rev-AI does better than Whisper. Moreover, Rev-AI can create reasonably useful speaker diarization and Whisper does not do this at all. Although Whisper runs much more slowly than Rev-AI, some projects may prefer Whisper's local mode of operation.
8. Whichever ASR engine you choose, basic Batchalign command for transcribing is:
    - batchalign transcribe --lang=[3 letter ISO language code] ~/ba_data/input ~/ba_data/output
    - For example, to transcribe with Rev.AI: batchalign transcribe --lang=eng ~/ba_data/input ~/ba_data/output
    - To use Whisper instead: batchalign transcribe --lang=eng --whisper ~/ba_data/input ~/ba_data/output
9. The first time you run Batchalign, the program will take about 5 minutes to download the material that will go into various cache folders on your system (this manifests as the program appearing to hang). After that, the system will ask you for your Rev.AI key from step 6 above, which you will need to paste into the program when asked if you wish to use Rev.AI. Cut and paste that from the place where you saved it earlier.

10. The program will provide output as it processes each input file, and you will soon see transcribed or coded CHAT (*.cha) files in your output folder(s)!

*Only the **transcribe** function requires the --lang flag. All other functions will read language information from the input CHAT file.*

FFmpeg installation for MacOS: We recommend that input audio files be in .wav or .mp3 format. If your files are in another format, the FFmpeg program can convert them. However, you must have FFmpeg installed. For MacOS this can be done by using Homebrew. To install Homebrew, go to https://brew.sh and copy the long command from the box to your terminal. Once installed, you can add ffmpeg using this command: brew install ffmpeg

FFmpeg for Windows: For information on how to install FFmpeg on windows, please go to https://www.wikihow.com/Install-FFmpeg-on-Windows. This process is rather tricky.

.**m4a conversion**: If you record with iPhone, the format is m4a. Since batchalign only accepts mp3, mp4, and wav, you will need to convert .m4a to .wav. You can do this using a program such as Audacity or Amadeus Pro or an online converter site such as this one: https://cloudconvert.com/m4a-to-wav

Support Information:
Please feel free to reach out if you have questions! You can send email to macw@cmu.edu or to houjun@cmu.edu. Should you reach out for help, please run "batchalign version" to tell us which version you are using.

"Verbose" Output:
There is a -vvv flag which allows Batchalign to run a file in "diagnostic mode." For instance, if your original command was:

```
batchalign align ~/ba_data/input ~/ba_data/output
```

To get diagnostic information, you would write:

```
Batchalign -vvv align ~/ba_data/input ~/ba_data/output
```