

Digital Biomarker Pipeline for mental health interviews

Authors: Oliver Williams & Simon Hartmann

Table of Contents

1. Introduction	3
2. Digital Biomarker pipeline	4
2.1. Requirements	4
2.2. Installation	4
2.3. Interview files	5
2.4. Flowchart	6
2.5. Domains and dependencies	6
2.6. Output	10
3. How-to guide	13
3.1. How to run the program	13
3.2. How to read the results	14
3.3. If not using the executable	15
4. FAQs	16
5. Additional Resources	17

Version	Date	Author	Rationale
0.1	1 April 2024	Oliver Williams	First draft
1.0	18 April 2024	Oliver Williams	Reviewed version for software release 1.0
1.1	25 April 2024	Oliver Williams	Reviewed version for software release with GUI 1.1
2.0	7 April 2025	Simon Hartmann	Updated version for software release 1.2
3.0	14 April 2025	Simon Hartmann	Updated version for software release 1.3
4.0	16 April 2025	Simon Hartmann	Added Whisper timestamps

1. Introduction

Approximately 30% of all adults have experienced a common mental health disorder across their lifetime and on average one in five adults annually [1]. The assessment of a patient's mental health poses a complex challenge to clinicians. Traditional diagnostic tools mainly rely on subjective assessments such as patient self-report and clinical observation [2], [3]. Structured interviews or questionnaires capturing a patient's state are infrequently used in clinical practice [4] resulting in a lack of standardised or systematically recorded data in mental health care. Hence, objective biomarkers describing a biological characteristic can help identifying mental health disorders in a subject. Although the etiology of the majority of individuals with a certain mental health disorder cannot be described by one unifying biomarker [5], a set of biomarkers may provide crucial information about the precise pathological processes and can help with selecting the most relevant treatment method [6]. A multitude of potential biomarkers for mental health disorder such as schizophrenia or major depressive disorders have been studied but there is still a demand for more objective measures that can be useful to identify mental illnesses [7].

Due to the shift to online mental health assessment during Covid-19, there is increasing potential to facilitate care via extraction of video and speech features. Our aim is to implement automated facial and speech features extraction and processing to provide cross-sectional diagnostic and prognostic information. Such information will facilitate the longitudinal trending of symptoms for the measurement of treatment response and the monitoring for early signs of relapse.

The below described pipeline is a small-scale automated end-to-end solution that can be used to extract facial, audio, and linguistic features from recorded video interviews. It is designed to analyse video/audio automatically recorded during HIPAA zoom interviews as part of clinical trials or research studies but can also be used in other situations (see FAQs).

References:

- [1] Z. Steel et al., "The global prevalence of common mental disorders: a systematic review and meta-analysis 1980–2013," *Int. J. Epidemiol.*, vol. 43, no. 2, pp. 476–493, 2014.
- [2] A. P. Association, *Diagnostic and statistical manual of mental disorders*, 5th ed. Washington, D.C.: American Psychiatric Association, 2013.
- [3] W. Gaebel, J. Zielasek, and G. M. Reed, "Mental and behavioural disorders in the ICD-11: concepts, methodologies, and current status.," *Psychiatr. Pol.*, 2017.
- [4] A. Aboraya, "Use of structured interviews by psychiatrists in real clinical settings: results of an open-question survey," *Psychiatry (Edgmont)*, vol. 6, pp. 24–28, Jul. 2009.
- [5] K. S. Kendler, "From Many to One to Many—the Search for Causes of Psychiatric Illness," *JAMA Psychiatry*, vol. 76, no. 10, pp. 1085–1091, Oct. 2019, doi: 10.1001/jamapsychiatry.2019.1200.
- [6] A. Levchenko, T. Nurgaliev, A. Kanapin, A. Samsonova, and R. R. Gainetdinov, "Current challenges and possible future developments in personalized psychiatry with an emphasis on psychotic disorders," *Heliyon*, vol. 6, no. 5, p. e03990, 2020, doi: <https://doi.org/10.1016/j.heliyon.2020.e03990>.
- [7] T. R. Insel, "Digital phenotyping: a global tool for psychiatry," *World Psychiatry*, vol. 17, no. 3, pp. 276–277, Oct. 2018, doi: <https://doi.org/10.1002/wps.20550>.

2. Digital Biomarker pipeline

2.1. Requirements

The pipeline requires an installation of Windows 10 or Windows 11 on a 64-bit machine. Depending on the duration of the interviews, the number of interviews to be analysed, and which domains (video, audio, or both) need to be extracted, it may require a substantial amount of memory. Memory requirements need to be kept in mind when running the pipeline.

Important: Prior to running the pipeline, all files and folders which are not related to the analysis need to be moved or deleted from the path.

Installation requires following pre-requisites:

- OpenFace (tested with v2.2.0): Download [here](#) (OpenFace_2.2.0_win_x64.zip)
- Whisper (tested with r136): Download the standalone version [here](#) (Whisper-OpenAI_r136.7z)
- ffmpeg (tested with version 2024-03-11-git-3d1860ec8d-full_build-[www.gyan.dev](#)): Download [here](#) (ffmpeg-git-full.7z)

2.2. Installation

The pipeline can be installed by downloading the [GitHub repository](#) or the [Windows executable](#). It is recommended to use the Windows executable, but information on how to run the Python script will also be provided in this document. When downloading the Windows executable, extract the files to the preferred installation path.

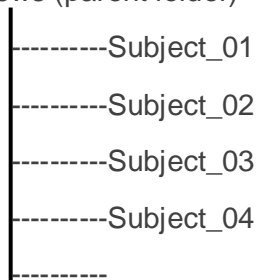
Afterwards, place the (extracted) OpenFace and Whisper folders into the '*biomarker_pipe*' folder. To check if the download was successful, check if '*whisper.exe*' exists in '*Whisper-OpenAI/Whisper-OpenAI*'.

ffmpeg can be installed on Windows by extracting the downloaded files, moving the files to the root of the C drive or the folder of your choice, and adding ffmpeg to the Windows PATH. The ffmpeg path (e.g. C:\ffmpeg\bin) can be added to the PATH in 'Advanced system settings' -> 'Environment Variables' -> 'System Variables' -> 'Path' -> Edit. Afterwards, the installation requires a Windows restart. If the installation of ffmpeg was successful can be checked by opening a Windows PowerShell and typing 'ffmpeg -version'. If the output confirms the version, the installation was successful. Further information on the installation of ffmpeg can be found [here](#).

2.3. Interview files

The pipeline expects one folder for each participant within one parent directory. The name of the folders can be e.g., the participants' IDs. The folder structure should look like the following

Interviews (parent folder)



or as shown in the snapshot below (where 0001 and 9090 are participant IDs).

> Interviews

Name	Date modified	Type
0001	30/04/2024 10:55 AM	File folder
9090	30/04/2024 10:55 AM	File folder

Each participant folder must contain the following files, downloaded from HIPAA Zoom or somewhere else.

- A file ending with '1.mp4' or '1.m4a' indicating the interviewer audio stream, e.g. when downloaded from HIPAA Zoom [date and time of the interview]_Recording_separate1.mp4
- A file ending with '2.mp4' or '2.m4a' indicating the participant audio stream, e.g. when downloaded from HIPAA Zoom [date and time of the interview]_Recording_separate2.mp4
- A file ending including 'gvo' and ending on '.mp4' indicating the recorded video, e.g. when downloaded from HIPAA Zoom [date and time of the interview]_Recording_gvo_1280x720.mp4

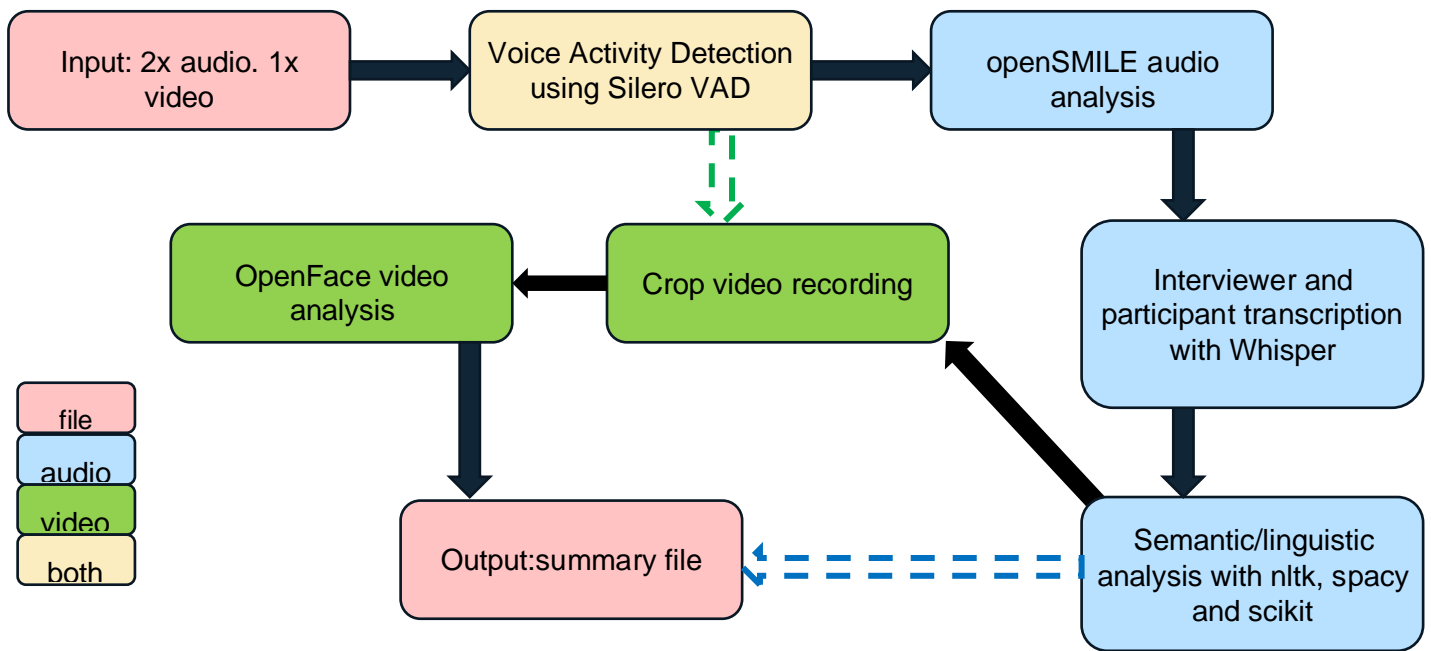
> Interviews > 0001

Name	Date modified	Type	Size
GMT20240412-012741_Recording_gvo_12...	12/04/2024 11:13 AM	MP4 Video	4,324 KB
GMT20240412-012741_Recording_separat...	12/04/2024 11:13 AM	MP4 Video	1,500 KB
GMT20240412-012741_Recording_separat...	12/04/2024 11:13 AM	MP4 Video	1,500 KB

Note: If there are no video files, the pipeline can be run in audio mode extracting only audio features.

WARNING: In overwrite mode, the pipeline will erase all files except for the aforementioned downloaded HIPAA Zoom files from the folder. PLEASE DO NOT STORE ANY OTHER DATA IN THE FOLDER!

2.4. Flowchart



The blue and green dashed arrows indicate an audio/video only run, respectively.

2.5. Domains and dependencies

The pipeline is comprised of five parts: Voice Activity Detection using Silero VAD (<https://github.com/snakers4/silero-vad/>), audio analysis (primarily using *openSMILE*, <https://audeering.github.io/opensmile-python/index.html>), transcription (using *Whisper*, <https://openai.com/index/whisper>), semantic analysis (using a combination of *spacy*, *nltk*, *scikit* and other packages) and facial movement analysis (using *OpenFace*, <https://github.com/TadasBaltusaitis/OpenFace>). Voice Activity Detection is executed every time when activated in the user interface whereas the next three parts account for the 'audio' domain whereas the last part comprises the 'video' domain. The 'audio' and 'video' domain can be run separately or together. It is recommended to provide audio files even if only the facial features are analysed as this way the pipeline can successfully execute the Voice Activity Detection to extract useful information on speaking and non-speaking periods for the participant and interviewer.

2.5.1. Voice Activity Detection

Voice Activity Detection is simply the detection of presence or absence of human speech in an audio stream. Here, we integrated Silero VAD (<https://github.com/snakers4/silero-vad/>) in our pipeline, a pre-trained enterprise-grade Voice Activity Detector. Silero VAD detects the onset and offset of speech in the participant and interviewer audio stream. We specified the minimum silence duration as one second and the minimum speech duration as 0.5 seconds.

Prior to Voice Activity Detection, the audio file needs to be converted to a '.wav' file. This is done using *ffmpeg*. In case the sampling rate of the input audio is not 8 kHz, 16 kHz, or a multiply of 16 kHz, the input audio is automatically resampled to 16 kHz.

2.5.2. Audio

Audio features can be divided into two major speech components: acoustic and linguistic features. Acoustic speech features refer to the physical properties of speech sounds, such as pitch, formants, and spectral envelope, whereas linguistic speech features refer to the linguistic content of speech, including words, grammar, and prosody.

Acoustic speech characteristics are influenced by the vocal apparatus and how it produces vocal sounds and have been a research topic in various fields for a long time. Large sets of acoustic features have been proposed that include parameters in the time domain, the frequency domain, the amplitude domain, and the spectral distribution domain. Table 1 lists an example of such a standard feature set, the Geneva minimalistic acoustic parameter set (GeMAPS), including the name and explanation for each feature.

Table 1: List of low-level descriptors in GeMAPS feature set, sorted by parameter groups

ACOUSTIC FEATURE	DESCRIPTION
FREQUENCY PARAMETERS	
Pitch	Logarithmic fundamental frequency (F_0) on a semitone scale, starting at 27.5 Hz
Jitter	Deviations in individual consecutive F_0 period lengths
Formant 1, 2, and 3 frequency	Centre frequency of first, second, and third formant
Formant 1, 2, and 3 bandwidth	Bandwidth of first, second, and third formant
ENERGY/AMPLITUDE PARAMETERS	
Shimmer	Difference of the peak amplitudes of consecutive F_0 periods
Loudness	Estimate of perceived signal intensity from an auditory spectrum
Harmonics-to-Noise Ratio (HNR)	Relation of energy in harmonic components to energy in noise-like components
SPECTRAL PARAMETERS	
Alpha Ratio	Ratio of the summed energy from 50–1000Hz and 1–5kHz
Hammarberg Index	Ratio of the strongest energy peak in the 0–2 kHz region to the strongest peak in the 2–5 kHz region
Spectral Slope 0-500 Hz and 500-1500 Hz	Linear regression slope of the logarithmic power spectrum within the two given bands
Formant 1, 2, and 3 relative energy	As well as the ratio of the energy of the spectral harmonic peak at the first, second, third formant's centre frequency to the energy of the spectral peak at F_0
Harmonic difference H1-H2	Ratio of energy of the first F_0 harmonic (H1) to the energy of the second F_0 harmonic (H2)
Harmonic difference H1-A3	Ratio of energy of the first F_0 harmonic (H1) to the energy of the highest harmonic in the third formant range (A3)
MFCC 1-4	Mel-Frequency Cepstral Coefficients 1-4
Spectral flux	Difference of the spectra of two consecutive frames

Acoustic features are extracted using *openSMILE*, a command-line interface that automatically extracts acoustic features from audio files. If Voice Activity Detection is set to 'VAD', *openSMILE* will provide low-

level and high-level acoustic features only for periods with participant's speech. If Voice Activity Detection is set to 'Both', *openSMILE* will provide low-level acoustic features for periods with participant's speech and high-level acoustic features for the whole audio. Here, *openSMILE* was configured to a default feature set containing the extended eGeMAPS feature set from the participant's audio stream. The pipeline saves the low-level descriptor information for every 10 ms in a spreadsheet in the participant's folder and writes the high-level summary information to the summary file in the parent directory.

Apart from the physical property of speech sounds, speech carries an abundance of information through the spoken word and the language that is used to convey a thought. This makes it a rich source that provides detailed insight into an individual's thought process.

Common linguistic markers are coherence, syntactic structure, complexity, and semantic and emotional focus. Linguistic markers can be extracted manually e.g., word counts or language abnormalities, as often done by clinicians in practice. Due to the laborious, time-consuming character of manual linguistic methods, automated natural language processing (NLP) methods were developed that show a high canonical correlation with manual linguistic methods. There are many NLP methods available that focus on extracting different aspects of speech. Some examples are Latent Semantic Analysis (LSA) that aims to examine semantic coherence in a text, or Part-of-Speech (POS) tagging which aims to analyse sentence structures and syntax.

Here is a short summary of the computational steps implemented in the pipeline to extract the linguistic features:

- Run *Whisper* on the silence removed audio to automatically transcribe the participant's and the interviewer's audio stream
- Split all sentences in the transcribed text, generate sentence embeddings (a vector of numbers that is used to objectively represent a word or sentence geometrically in space) using the Bidirectional Encoder Representations from Transformers (BERT) developed by Google which was pretrained on the entire English Wikipedia and Toronto BookCorpus datasets, and calculate the semantic coherence (similarity) using the cosine similarity function in *sklearn*.
- Calculate word count in the participant's text and in the interviewer's text and extract the ratio between participant and interviewer.
- Extract the positive, negative, neutral, and compound sentiment in each sentence using *NLTK's SentimentIntensityAnalyzer*.
- Remove punctuations and contractions using *NLTK*, perform part-of-speech tagging using *spacy*, calculate the frequency for all Universal, Penn Treebank, and dependency tags, and normalize according to the total number of sentences

2.5.3. Transcription timestamps

Although Whisper models were trained to predict approximate timestamps on speech segments, they cannot originally predict word timestamps and timestamps are not included in the original Whisper transcription. We incorporated a separate open-source solution (<https://github.com/linto-ai/whisper-timestamped/>) to predict word timestamps and provide a more accurate estimation of speech segments when transcribing with Whisper models. The approach is based on Dynamic Time Warping (DTW) applied to cross-attention weights. There are two different options for timestamps: 'segments' and 'words'. 'Segments' provides start and end timestamps for segments such as sentences or phrases whereas 'words' provides start and end timestamps for words. The timestamp information can be used in post-

processing to extract the acoustic and facial movement features during or after the speech of certain words or phrases.

2.5.4. Video

In 1976, Ekman and Friesen released the Facial Action Coding System (FACS), the first concerted effort of developing a system that can distinguish visually distinct facial movements. The FACS was the first comprehensive tool that focused on describing visually distinct facial behaviour that can be measured and described without any inference. The intention of the FACS was to provide unbiased information on facial behaviour that can be used later to make an inference that can be in turn tested by evidence. The FACS describes fundamental actions of individual or group muscles in the face as action units (AU) that can be rated on a 5-point scale (A = trace, B = slight, C = marked or pronounced, D = severe or extreme, E = maximum). Table 2 lists the number, name, and muscles of the AUs included in the FACS.

Table 2: Single action units included in the Facial Action Coding System (FACS).

AU NUMBER	NAME OF ACTION	MUSCLE(S) ACTIVATED
1	Inner brow raiser	Frontalis (pars medialis)
2	Outer brow raiser	Frontalis (pars lateralis)
4	Brow lowerer	Depressor glabellae, depressor supercilli, corrugator supercilli
5	Upper lid raiser	Levator palpebrae superioris, superior tarsal muscle
6	Cheek raiser	<i>Orbicularis oculi, pars orbitalis</i>
7	Lid tightener	<i>Orbicularis oculi, pars palpebralis</i>
8	Lips toward each other	Orbicularis oris
9	Nose wrinkler	<i>Levator labii superioris alaeque nasi</i>
10	Upper lid raiser	<i>Levator Labii Superioris, Caput infraorbitalis</i>
11	Nasolabial deepener	<i>Zygomatic Minor</i>
12	Lip corner puller	<i>Zygomatic Major</i>
13	Sharp lip puller	<i>Levator anguli oris (Caninus)</i>
14	Dimpler	<i>Buccinator</i>
15	Lip corner depressor	<i>Depressor anguli oris (Triangularis)</i>
16	Lower lip depressor	<i>Depressor labii inferioris</i>
17	Chin raiser	<i>Mentalis</i>
18	Lip pucker	Incisivii labii superioris and Incisivii labii inferioris
19	Tongue Show	
20	Lip stretcher	Risorius with platysma
21	Neck tightener	Platysma
22	Lip funneler	Orbicularis oris
23	Lip tightener	Orbicularis oris
24	Lip pressor	Orbicularis oris
25	Lips part	<i>Depressor Labii, Relaxation of Mentalis (AU17), Orbicularis Oris</i>
26	Jaw drop	<i>Maseter; Temporal and Internal Pterygoid relaxed</i>
27	Mouth stretch	<i>Pterygoids, Digastric</i>
28	Lip suck	Orbicularis oris
41	Lid droop	Relaxation of <i>Levator Palpebrae Superioris</i>
42	Slit	Orbicularis oris

43	Eyes closed	Relaxation of <i>Levator Palpebrae Superioris</i>
44	Squint	Orbicularis oculi, pars palpebralis
45	Blink	Relaxation of Levator Palpebrae and Contraction of Orbicularis Oculi, Pars Palpebralis.
46	Wink	Levator palpebrae superioris; Orbicularis oculi, pars palpebralis

The AUs in the updated version of the FACS describe only clearly visible facial movements, and do not account for subtle changes such as skin colour changes, muscle tonus, sweating, etc. Due to the tedious and time-consuming nature of manually applying the FACS, automated facial expression analysis (AFEA) software has been developed to facilitate the adoption of the FACS in human research.

Here, *OpenFace* was included in the pipeline to extract facial action units according to the FACS. *OpenFace* comprises an in-built high-performing processing pipeline consisting of facial landmark detection and tracking, head pose estimation, eye gaze estimation, and facial AU detection. *OpenFace* is called within pipeline as a command line tool and extracts a subset of the full FACS AU list. The detailed *OpenFace* output is stored in the spreadsheet in the participant's folder whereas a summary (number of activations a minute for each AU) is written to the summary file in the parent folder. Further, the facial movement output is calculated for speaking and non-speaking periods if audio files were provided and Voice Activity Detection was set to 'VAD' or 'Both'. Quality of the facial movement extraction is recorded in the confidence score which is written to the output summary file in the parent directory.

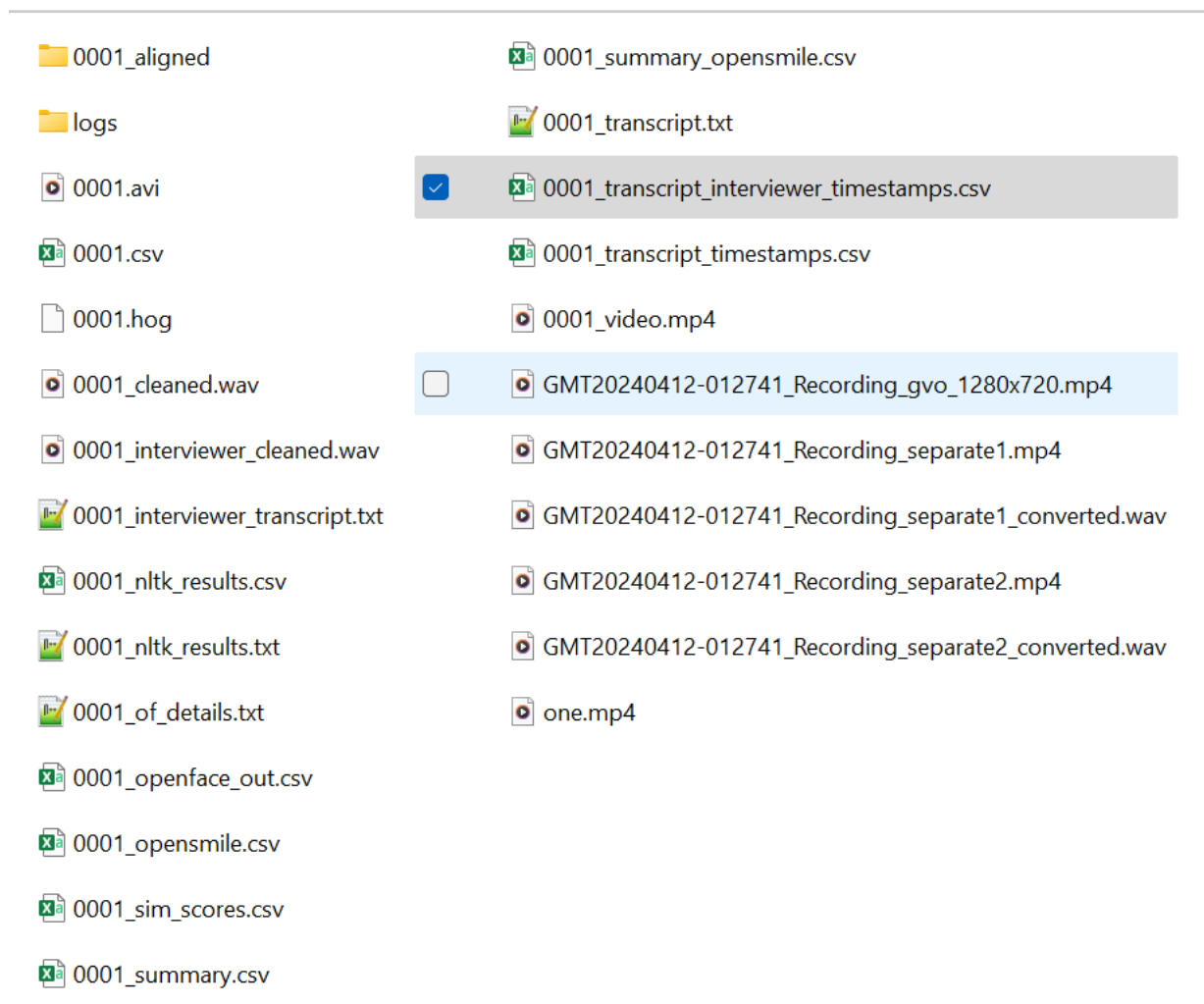
Prior to running *OpenFace*, the pipeline offers the option to crop the participant's screen including only the participant's face from the video. **Please make sure that the participant's screen is shown in the right half of the video.**

2.6. Output

- **logs:** Contains logs generated by *ffmpeg*, *Whisper* and *OpenFace* (depending on the 'Verbosity' that was selected). If there are any errors, the log files are a good starting point for troubleshooting.
- **[participant name]_aligned:** Generated by *OpenFace*. Contains every frame from the video recording. The size of the folder increases with the length of the video.
- **[participant name].mp4:** The processed *OpenFace* video showing gaze and posture.
- **[participant name].csv:** Generated by *OpenFace*. Contains data on gaze, posture and action units for each video frame.
- **[participant name].hog:** Generated by *OpenFace*.
- **[participant name]_openface_out.csv:** *OpenFace* summary file. Contains rate of binary action unit activation and rate for no-speaking vs. speaking.
- **[participant name]_of_details.txt:** Generated by *OpenFace*. General information on configuration and parameters.
- **[participant name]_nltk_results.txt:** Detailed information on tokenisation, part of speech and dependency tagging.
- **[participant name]_nltk_results.csv:** Summary file on semantic analysis step. Contains sentiment scores, POS and dependency tagging counts, average similarity score between neighbours and interviewer/participant speech ratio

- **[participant name]_sim_scores.csv**: Generated during semantic analysis. Full matrix showing similarity between each sentence.
- **[participant name]_opensmile.csv**: Generated by *openSMILE*. Contains low-level information on various acoustic markers every 10 milliseconds.
- **[participant name]_summary_opensmile**: Summary of *openSMILE* results. Shows average of every data point.
- **[participant name]_transcript.txt**: Transcript generated by *Whisper* for the participant.
- **[participant name]_interviewer_transcript.txt**: Transcript generated by *Whisper* for the interviewer.
- **[participant name]_transcript_timestamps.srt/csv**: Start and end timestamps for segments (csv) or words (srt) in participant speech
- **[participant name]_interviewer_transcript_timestamps.srt/csv**: Start and end timestamps for segments (csv) or words (srt) in interviewer speech
- **[participant name]_video.txt**: Video cropped to only show one person.
- **[participant name]_silences.txt**: Generated by *ffmpeg*. Contains on the detected periods of non-speaking in the participant's audio.
- **[participant name]_summary.csv**: Summary file per participant of all pipeline steps.
- **all_summary.csv**: Summary file for all participants. This file can be found in the parent folder.

An example of all output files in a participant's folder is shown below.



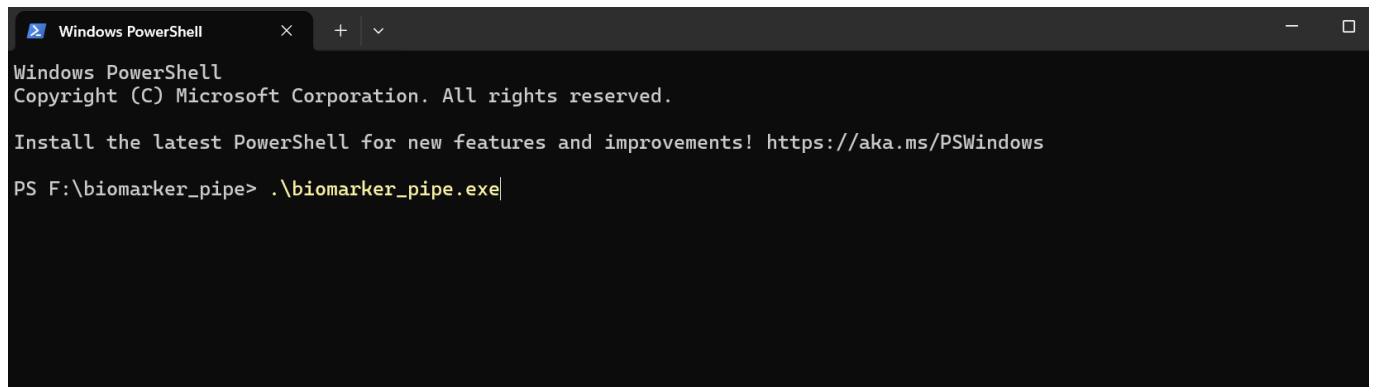
3. How-to guide

3.1. How to run the program

After the installation is finished and all folders are set up, the pipeline can be run. To start the pipeline, open a power shell window (type “*pow*” + ENTER in the search bar). Then type:

```
.\<path to the biomarker_pipe folder>\biomarker_pipe.exe
```

For example:



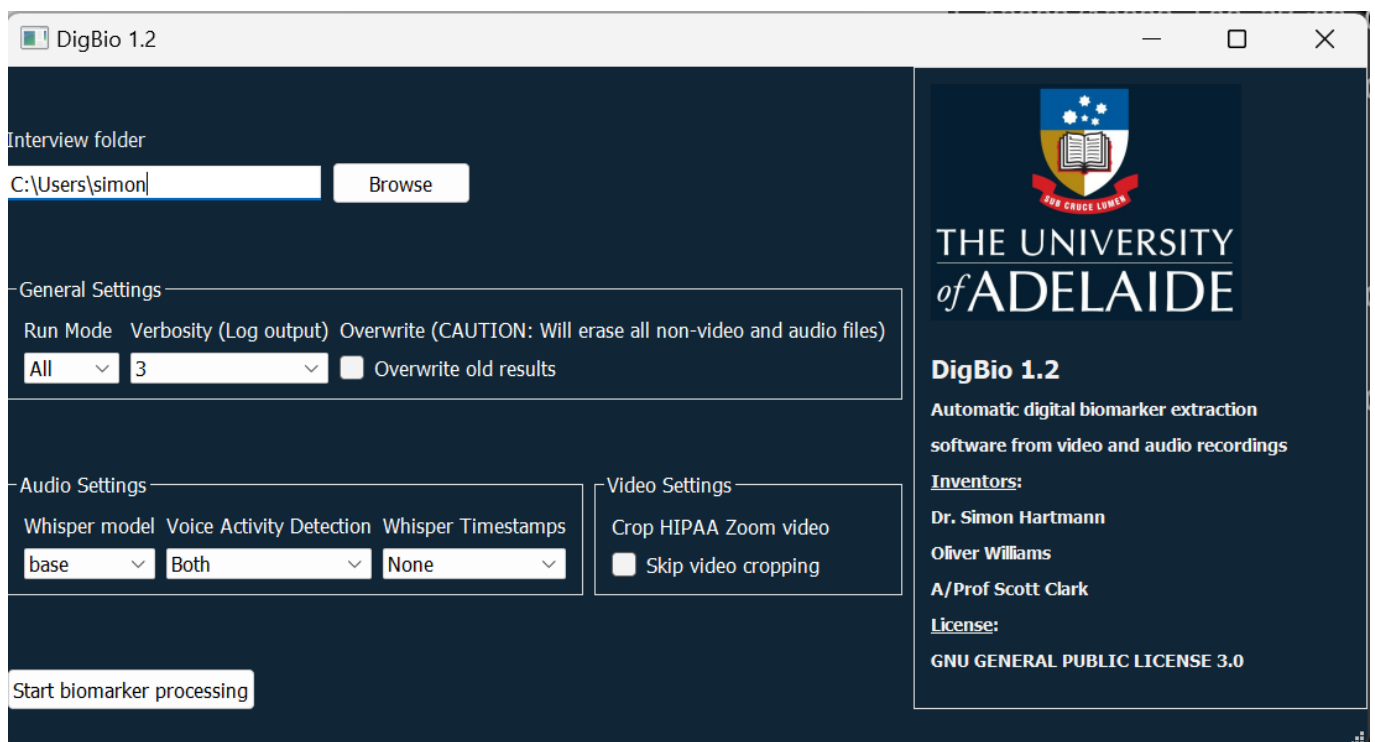
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS F:\biomarker_pipe> .\biomarker_pipe.exe|
```

This will open the popup window. Alternatively, double-click the file *biomarker_pipe.exe*. This will automatically start the pipeline. It may take a little while until the popup window appears, so be patient.

Note: If starting the pipeline using double-click, the user interface and shell window will close after the pipeline has completed. All information shown in the shell is stored in the log file, so if an error occurs it will be documented in there.



Once the window has opened, click 'Browse' and select the folders with participant data in it. You can select specific domains to run by selecting the tab under the 'Mode' field. Under verbosity you can change the level of logging output. 4 will give a more detailed output, 1 or 2 only minimal information.

Activating the 'Overwrite old results' box will run the pipeline from scratch for all participants. By default, this box is not activated, as it will, e.g. not rerun participants who already have results. **If activated it will delete all files other than the recordings, so use with caution.** If videos were not generated using HIPAA zoom, select the 'Skip video cropping' field. Note that videos must only contain one person (the participant), so videos may need to be cropped manually. Under 'Whisper model' different *Whisper* transcription models can be selected. This parameter can be changed to smaller models such as 'tiny' or 'small' when experiencing serious performance issues or crashes. 'Voice Activity Detection' defines whether speaking and non-speaking parts in the participant and interviewer audio stream will be identified and used for audio and facial movement analysis. 'VAD' means all audio analysis results are limited to speaking parts only whereas 'Both' means that detailed 10 ms audio information will be provided for the whole audio whereas the summary information only includes results for the speaking part. Selecting 'Segments' or 'Words' for Whisper timestamps will provide detailed start and end timestamps for segments such as sentences or phrases or words in the transcript of the participant and interviewer.

Finally, the 'Start biomarker processing' button will start running the program. The window will close automatically once the pipeline has finished.

Progress of the pipeline can be tracked in the PowerShell window. Log messages will appear there.

Once a participant has been fully processed following message will appear

INFO: Pipeline complete for participant <ID>.

Once all participants have been processed, following message will appear:

INFO: All participants have completed.

To cancel or kill the pipeline, either close the PowerShell window or use 'Ctrl + c' for keyboard interrupt.

3.2. How to read the results

The *all_summary.csv* file will contain a summary from all participants.

ID	max sente	avg senter	total num	neg sent	neu sent	pos sent	comp sent	max sim	sc min	sim sc	avg sim	sc var	sim sc	part_word	ADJ	ADP	ADV	AUX	CONJ	CCONJ	DE
9090	57	17.4	5	0.1296	0.6968	0.1734	0.42626	0.27954	0.086654	0.211892	0.007859	0.490385	1.8	1.4	1.6	1.2	0	1.4	0	0.	
1	11	7	3	0	0.549	0.451	0.4614	0.493996	0.493996	0.493996	0	0.111111	0.666667	0.666667	0.666667	0.333333	0	0	0	0.	

The following semantic/linguistic features are included in the summary file (in order from A to N):

Participant ID, max sentence length, total number of sentences, average sentence length for participant, negative/neutral/positive/complete sentiment score, the max, min, average, and variance of similarity score of neighbouring sentences, the proportion of words said by the participant (compared to all words said by interviewer and participant), POS (Universal) tag count, POS (Penn Treebank) tag count, and dependency tag count.

O	P	Q
F0semitor...		equivalen
23.40389	...	-30.5862
23.97303	...	-37.0117

The content from the *opensmile_summary.csv* file is also in the summary file. This includes information on the mean and standard deviations of acoustic characteristics such as volume, pitch, voice modulation etc.

R	S	T	U	V	W	X	Y	Z	
AU01_c	...	AU45_c	AU01_c_si...		AU45_c_si	AU01_c_si	...	AU45_c_sp	
0.148488	...	0.269978	0.094492	...	0.12149	0.053996	...	0.148488	
0.12443	...	0.228121	0.12443	...	0.165906	0	...	0.062215	

The summary of *OpenFace* provides information on facial movements, i.e. if certain muscle groups were activated or not. The information on the activation of facial muscles is called action units (see [here](#) for more information). Columns R to T (in the above snapshot) are the rate of the specific action unit, i.e. total number of activations divided by the total video length. Columns U to V are the action unit rates for periods while the participant was not speaking (“AUXX_c_sil”). The last columns are the proportion that happened while the participant was speaking (“AUXX_c_sp”).

3.3. If not using the executable

Note that it is **strongly recommended** to use the provided executable, as it contains all Python dependencies and packages!

3.3.1. How to install Python

Open a Windows PowerShell by typing ‘cmd’ into the Windows search bar. Hit enter and a PowerShell should pop up. Type ‘python --version’ and hit enter. If it displays the Python version, e.g. ‘Python3.12.3’, python is already installed, and no need to do anything further.

If not, Python needs to be installed. This can be done by downloading the latest release from <https://www.python.org/downloads/>. Launch the executable that was just downloaded and follow the steps. Alternatively, Python can be installed from the Microsoft Store. Just enter “Microsoft Store” in the Windows search bar, open the store, search for Python, and install it.

3.3.2. How to install Python packages

If Python is installed, it will have already come with an installation of *pip* (package installer for Python). You need *pip* to install packages for python.

The following packages are required for the pipeline to work:

argparse, logging, opensmile, csv, shutil, nltk, contractions, spacy, pandas, sklearn, sentence_transformers, silero_vad

They can be downloaded and installed using following command into a command shell window:

`pip install <package name>`

3.3.3. How to run the Python script

Running the python script works very similar to running the executable.

`python3 <path to digbio folder>\biomarker_pipe.py --interviews <path to interviews folder>`

For example:

```
PS C:\> python3 C:\Users\Oliver\Desktop\digbio_sourcecode\biomarker_pipe.py --interviews C:\Users\Simon\Desktop\Interviews\ --verbosity 4
```

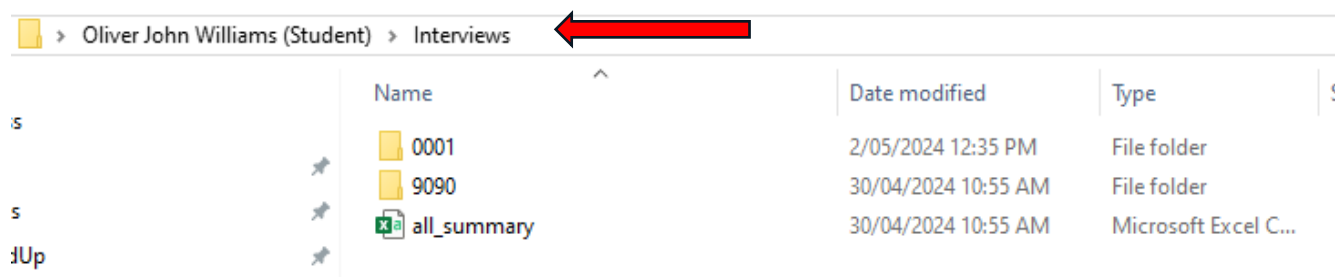

4. FAQs

I did not use HIPAA zoom to conduct the interview. Can I still use this tool?

Yes. You will need to rename your video and audio files to follow the normal convention. It is also important that you provide two audio files if not running just the video part. The audio must be separated into two files, so there should be one file per person. If your video shows more than one person, you need to trim your video manually to only show the participant. You can do this online. Make sure that when you run the pipeline you include the `--no_cut` option.

Why can the program not see my interviews folder?

Make sure you provide the full path to the file. A full path always starts with a 'C:' or 'U:', or something like that, depending on where it is located. To find the full path, open you're the folder that contains all the interviews.



If you click the line indicated by the arrow, the full path will appear and can be copied.

I get warning messages from Silero VAD saying that the sampling rate is manually casted to 16000. Do I have to worry about the messages or can I ignore them?

```
silero_vad\utils_vad.py:269: UserWarning: Sampling rate is a multiply of 16000, casting to 16000 manually!
warnings.warn('Sampling rate is a multiply of 16000, casting to 16000 manually!')
silero_vad\utils_vad.py:269: UserWarning: Sampling rate is a multiply of 16000, casting to 16000 manually!
warnings.warn('Sampling rate is a multiply of 16000, casting to 16000 manually!')
```

Silero VAD expects a sample rate of either 8 kHz or 16 kHz as input audio. If the input audio has a sample rate that is a multiply of 16 kHz, it will manually cast it to 16 kHz. The pipeline has an automatic check in-built that controls if the sampling rate of the input audio is 8 kHz, 16 kHz, or a multiply of 16 kHz. If not, the pipeline will automatically cast the input audio to 16 kHz. Hence, above warning messages can be ignored as they only indicate that Silero VAD will automatically cast the sampling rate of input audio to 16 kHz. This has no effect on the quality of the analysis output.

The program is taking a long time to run. How do I know that it is working correctly?

The pipeline uses some complex models. It is normal for this to run for a long time. How quick it finishes depends on a few things, like what parts are being run, how fast the computer is, how long the interviews are, etc.

It is a reasonable estimate that one participant with a ca. 40-minute interview will take at least 60 minutes to finish.

There will be some output to the command line telling you what is currently going on. Note that the order in which participants' folders finish being processed is random.

An error occurred: [WinError 32] The process cannot access the file because it is being used by another process

This means you have a file open that the program needs to access. Before you (re)run the program close all files that the program needs to access. That includes audio files, video files, text files, csv files or log files.

Openface/Whisper/ffmpeg returned exit code x. See log file for detailed error message.

In each participant's folder there is a folder called 'logs'. Open the file of whatever is causing the error and read it. In a lot of cases, the issue is either not putting the correct installation in the correct place (see 'Dependencies' on where to put what) or an issue with the audio/video files. Make sure they are not corrupted in some way, have the correct name, are in the right place and are mp4 files. Also check that you provided the correct path to your interviews folder and that it

After I started the pipeline, the popup window stopped responding. What does this mean?

This is normal. Once the program has started all important output will happen in the PowerShell window that appeared. If the window has closed, you can find the detailed log in the file DigBio.log which is in the same folder as the biomarker_pipe.exe file.

5. Additional Resources

See [here](#) for the biomarker download and some more documentation.