

Synchrony in Psychotherapy, Data management, example with F1044 patient data

Aim and Hypothesis

The aim of this project is to evaluate if it is possible to detect automatic signals of synchrony that could predict the outcomes of a familial psychotherapy.

Here is the analysis of the F1044 subject, his family (father and mother) and the therapist.

Is there synchrony signals computed by [SyncPy module](#) between himself, his parents and the therapist ? Could this synchrony signal predict outcomes of the psychotherapy ? See the full [pre-registration](#).

We used this data from a large european psychotherapy study. This [INCANT](#) study. study aimed to evaluate the efficacy of the [Multi Dimensional Family Therapy](#) (MDFT) for cannabis use disorders in adolescents.

The main outcome was be cannabis consumption evaluated with the Timeline Follow-Back (TFLB) [questionnaire](#).

Data structure

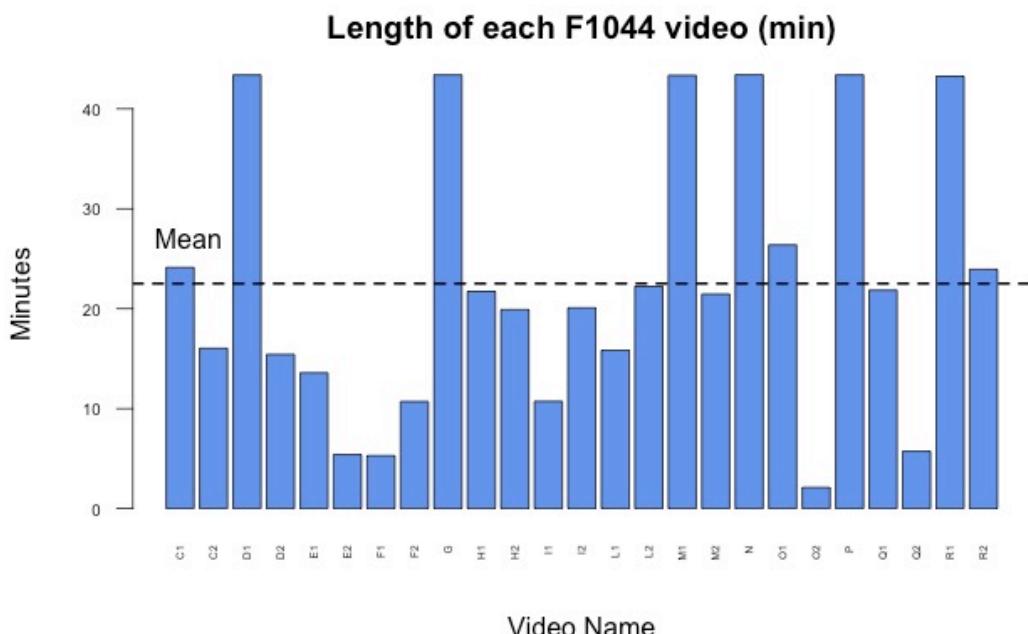
Data consist of videos and psychometric data.

Videos

- A 252.44GB database of 277 Videos, with a rate of 25 frames by second.
- They are encoded in [VOB](#) format
- There are only accessible on ISIR local network and to registered persons on [ISIR NAS](#).

Example with F1044 subject

F1044 subject had a lot of videos so we decided to make a pilot study with him.



Psychometric data

- It consists of 3 excel files with :
 - the Timeline Follow-Back ([TLFB](#))
 - the Youth Self-Report syndrome structure ([YSR](#))
 - the Child Behavior Checklist syndrome constructs ([CBCL](#)).

TODO

- NA is -99 ? to check
- Define all the variables
- get the questionnaire

Nomenclature

F1044 is the name of the subject studied (called *patient*). When referring to several persons, variables take the name of the shorter form *pa* for *patient*.

He has a *mother* (*mo*) a *father* (*fa*) helped by a *therapist* (*th*). When a variable is referring to several participants, it is organised in alphabetical order separated by underscores, eg. *SSI_fa_mo* refers to the synchrony index (*SSI*) between the *father* and the *mother*.
SSI_mo_fa doesn't exist.

This family had several consultations with the psychotherapist (*therapist*). Some of them were video recorded. These videos are names with the name of the subject + an index letter. They can be subdivided after that with numbers (eg *F1044C*).

Softwares used

- Git
 - GitHub website
- Excel
- R
 - xlsx package
- Python
 - opencv
 - cv
 - panda
 - matplotlib
 - syncpy
 - [FFMPEG](#)
- C ++
- Paintbrush 2
- Quicktime
- VLC
- Elan
- Markdown

Steps

Description of the database

Psychometric data

Files were collected in [xls](#) format. They were imported via the [XLSToCSVConvertor.R](#) R script. It returns a dataINCANT.csv cvs file and a [dataCannabis.csv](#) file. The evolution of this scores are plotted [here](#).

Video Configurations

Spatial organisation

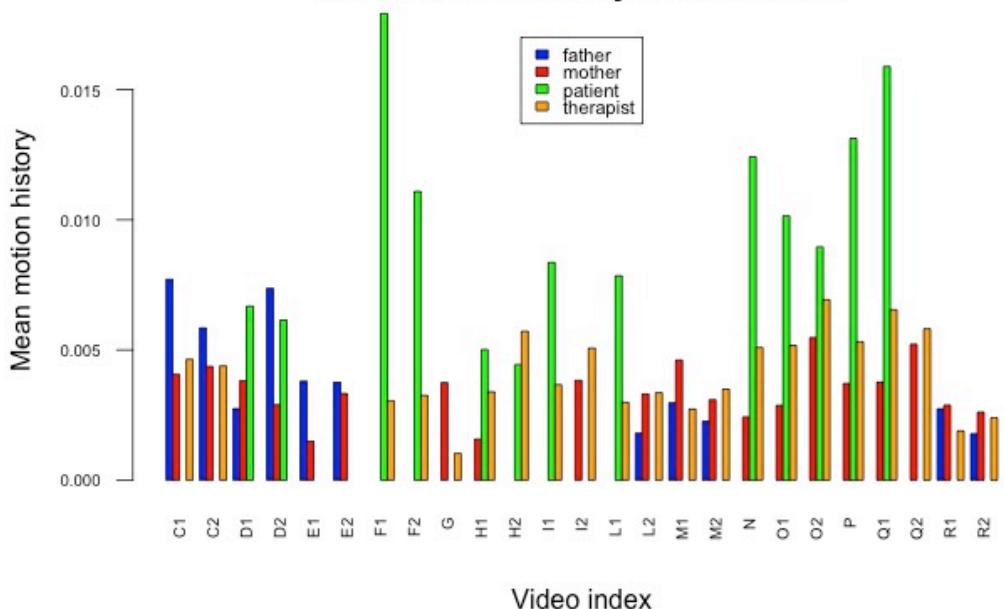
In the first center, psychotherapy sessions consisted of filmed familial psychotherapies with 2 to 5 peoples organized in a circle. Two participants of the psychotherapy were filmed from the front. Another subject is filmed from the other side and is embedded in a window. We can notice the date displayed on the video. Sometimes, there is an overlap of

different subject in the same place.

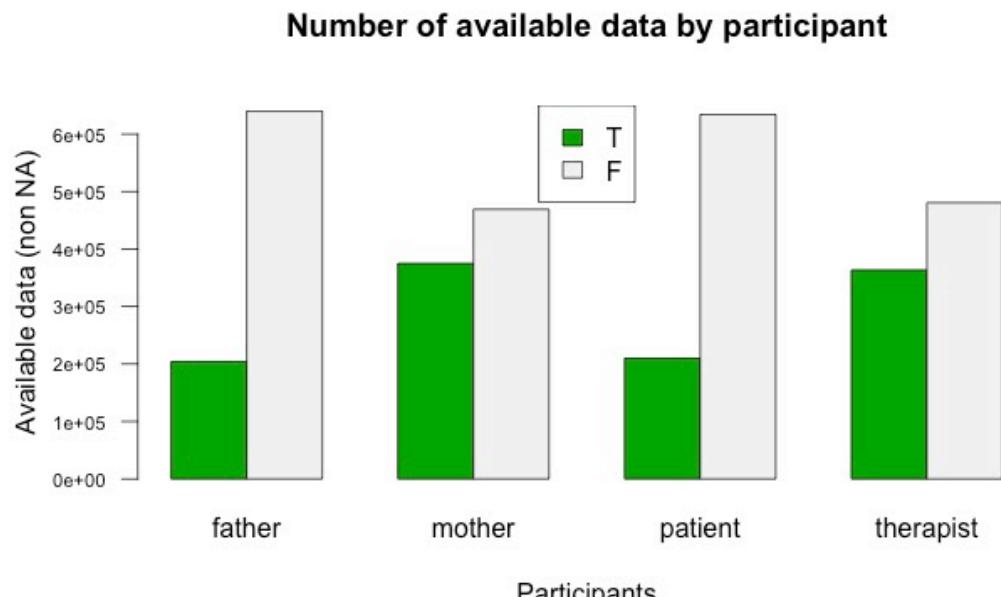


We can see that the configurations for the F1044 subject can be very different. The patient is not very present in the therapy (compliance problem). Sometimes, we don't have any information from the therapist because she isn't filmed.

Mean Motion history in each video



When we sum up the data we get, we can see that the mother and the therapist are the most present participants.

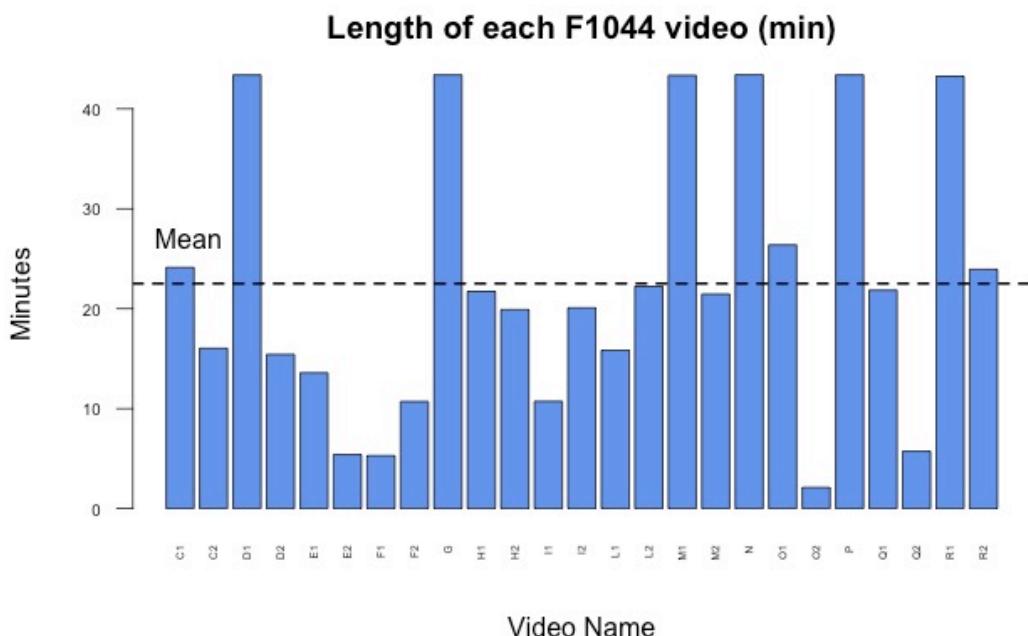


Time organization

Video length could be very different because it was necessary to reorganize them.

The configurations could change (eg patient-mother-therapist then mother-therapist) during the psychotherapy, a phone was used to communicate between the main psychotherapist and the supervisor. The participants made some pauses or the same video was used for different patients.

Hence, it was necessary to cut the videos with [Quicktime](#) software.



Raw data

The first step was to extract frames with the [Frames_extractor.py](#) python script made by Jonathan with [FFMPEG](#).

Idea	
	In a lot of other domains of fundamental algorithm development, computer labs compete on public databases to improve the detection rates of some features for instance. This work are presented during challenge sessions in congresses. However, there is a problem with confidentiality when we deal with psychotherapy videos.
	It would be better to make a specific database for this kind of automatic analyses to avoid changes of configuration of the room, the luminosity, the place of the different subjects, the number of subjects, the movement of the camera during recording.

A. Motion extraction

1. Extracting frame from the beginning of the video

Extract a frame with [Frames_extractor.py](#) with a [python](#) script from the beginning of the video to make a mask. This script was developed by Jonathan Aigrain with [Open CV](#).

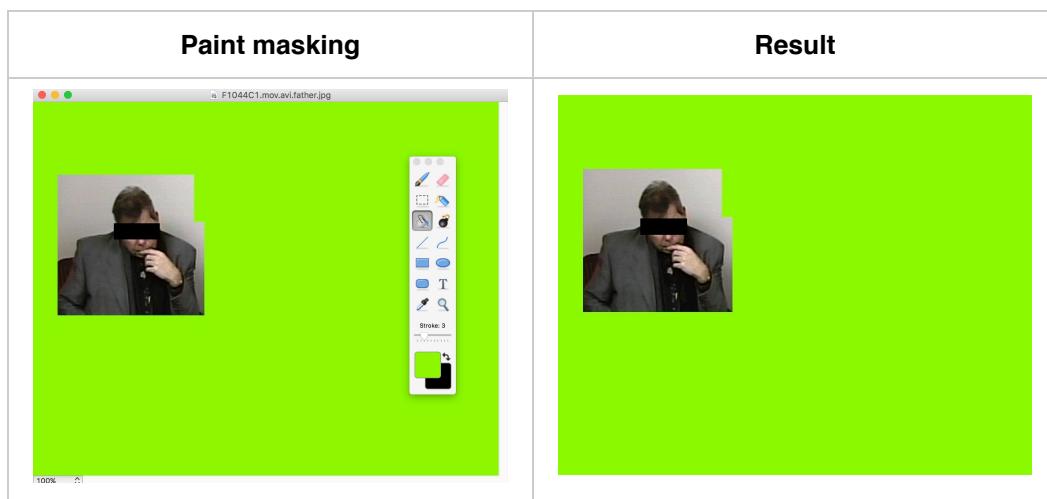
Here is an example of the father (eyes blurred to keep anonymity).



Idea	
	It was suggested to take a mean image of the video to be more precise instead of taking a video from the beginning.
Beware	
	Notice the date that was always masked after during the step 2.
	We can change the value of the median filter .

2. Make a mask for each relevant part of each video with Paintbrush

Select a part of the video with each participant (father, mother, patient, therapist) with [Paintbrush](#). Take only the upper part of the body to compare them since you can see only the upper part of the therapist.



This operation is repeated for each participant on each video.

Beware	
BEWARE	<p>It is necessary to always use the same background color, for instance RGB green. (0,255,0)</p>
	<p>Participants can move during the video (change their seats, leave the room). It would be preferable to anticipate it. Camera is moving sometimes too. Mean Motion history by minute can help us to detect big changes or disappearance of a participant.</p>
	<p>Participants are labelled in the mask name (eg <i>F1044C1.avi.father.jpg</i>)</p>
Idea	<p>Deep learning software or machine learning used for categorization/detection of people could make this process completely automatic.</p> 

3. Motion history extracted

The VOB videos were converted in [AVI](#) format in the form *F1044C1.avi*.

It is extracted with the video and the mask with a [C++](#) script. It returns [CSV](#) file of the form "*F1044C.VOB_res2.csv*". See files [here](#).

Headers :

- **frame** : number of the frame (original rate of 25 frames by second)
- **father** : motion history of the father frame by frame. It ranges from 0 (same frame than the previous frame, no movement) to 1 (every pixel change).
- **mother** : idem for mother
- **patient** : idem for patient
- **therapist** : idem for therapist
- **file** : name of the file in the form *F1044C.VOB*

- **NA** : (Non available) corresponding to absent subjects (NA column) or bad quality of frames at the beginning or end of the videos making impossible extraction of motion

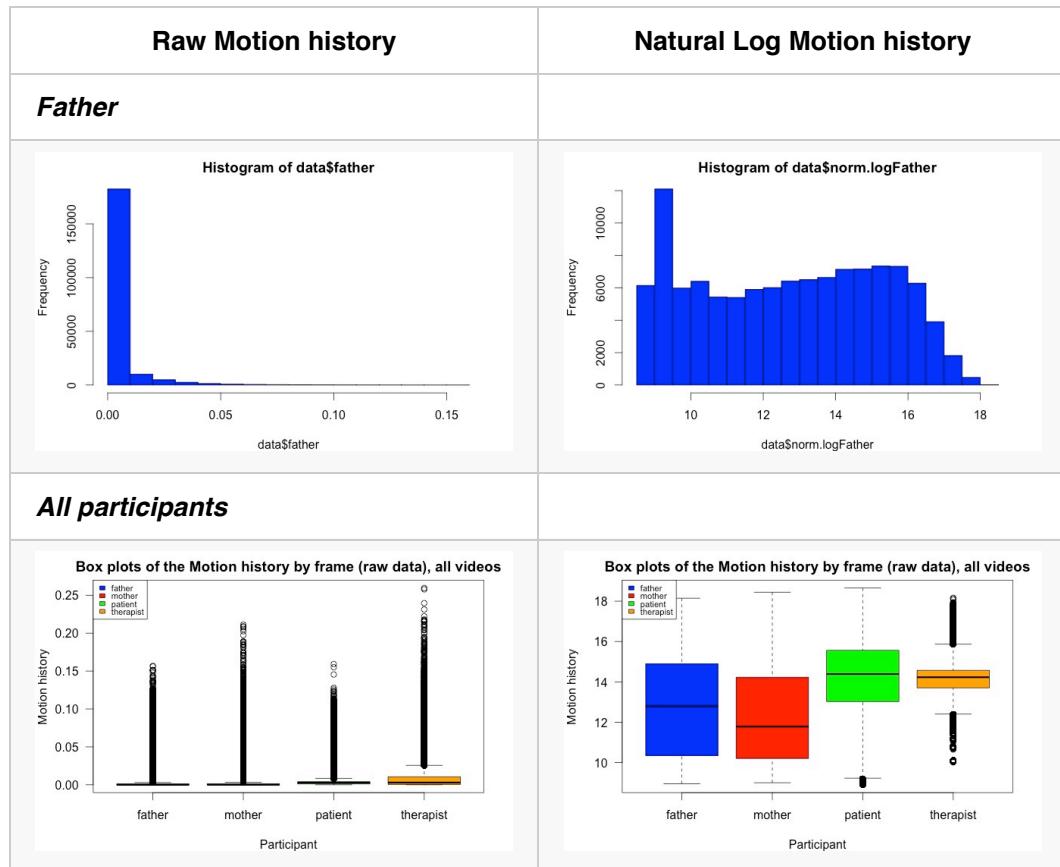
history for any participant (NA line). After a problem of encoding, from .VOB, to .mov (video format exported from quicktime when cut are made) and then to AVI, there is a problem of encoding and the length of the video is not well recognized and NA lines are generated that correspond to nothing.

If we plot (in histograms and box plots) this raw data (1st column), we notice that the distribution is not normal at all. Very small motion history values are over represented and bigger motion history are much more rare with a very long tail.

To normalize the distribution to compute synchrony scores on it, we made the natural logarithm. It produces negative numbers. SyncPy can't compute negatives scores, they are so shifted to positives values with an arbitrary value of 20 to avoid to keep extreme negative values.

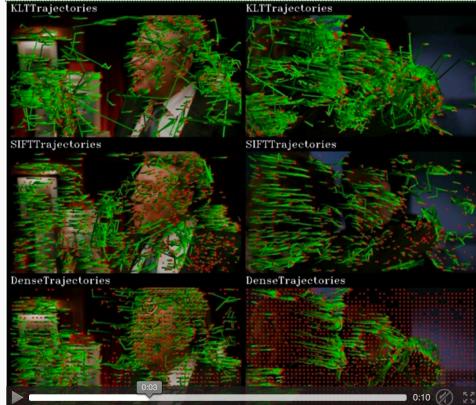
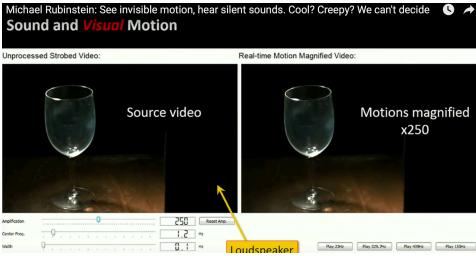
Values equal to 0 can't be loged. They generate a -Inf value. These values are set to NA. We lose the information of no movement at all. If we give a arbitrary value to this data (eg, the minimum value, they are over represented).

This normalized log data is shown on column 2.



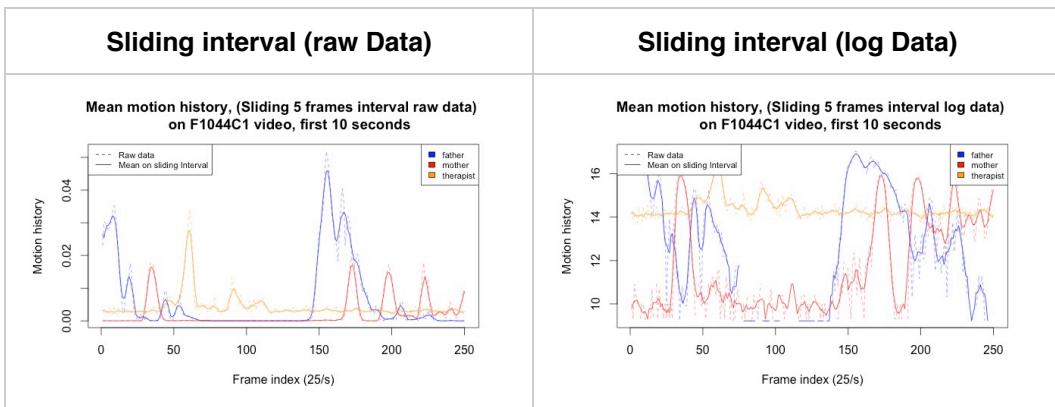
If we check all participants, we can see that the father and the mother motion history distribution is very similar. However, the therapist, which is always in a small window of

the video, as a very different distribution. We have less signal on it. In some videos the patient is in this window, it explains, it intermediates position.

Idea	
	<ul style="list-style-type: none"> - It could be possible to extract background to improve the quality of the motion history. See for instance backgroundExtractor.py with openCV BGS library.
	<ul style="list-style-type: none"> - It could be interesting to evaluate the trajectories of the people. If they are going toward the same direction or not. For that, the optic flow modules of open CV or the Dense Trajectories Video Description  <p>could help.</p>
	<p>Tiny changes can't be always obvious. Some softwares can amplify a lot very tiny changes in specific frequencies that can't be seen normally with naked eyes.</p> 

4. Motion history filtered with slidingInterval function

Motion history raw data is filters with a [R](#) function on [Rmarkdown script analysis](#). It generates CSV file of the form "F1044C.VOB.slidedata.csv". See files [here](#).



Headers :

- **video** : name of the video (eg F1044C.VOB)
- **frame_index** : number of the frame (original rate of 25 frames by second)
- **slidedFather** : filtered motion history with the sliding motion history function making mean on 5 frames (2 frames before and after the index frame). It ranges from 0 (same frame than the previous frame, no movement) to 1 (every pixel change).
- **slidedMother** : idem for mother
- **slidedTherapist** : idem for therapist
- **slidedPatient** : idem for patient
- **NA** : corresponding to absent subject

Idea	
	It is possible to change the size of interval to change the ration signal/noise
Beware	
	Bad quality of first frames at the beginning or end of the videos with NA were deleted to prepare the next script analysis.

4. Computing synchrony score with this filtered motion data

Amplitude (difference between min and max) and baseline levels (min) are not relevant to compute synchrony.

The script in [Python](#), [Call_S_Estimator.py](#) takes this filtered motion data and compute a synchrony score for each association possible of two or more subjects.

Download and install [SyncPy](#).

The script [Call_S_Estimator.py](#) must be installed in Syncpy installation in examples folder.
Create a folder *SynchronyCSV* in this folder.

You need to specify,

- the folder where you put your data (eg dataFolder =
'/Users/Ofix/Documents/Fac/internat/Recherche/projets/synchro/synchroData/CSV/filtered/')
- the list of videos you want to analyze, (eg VideoList = ["F1044C.VOB", "F1044D1.VOB", "F1044D2.VOB", "F1044E.VOB", "F1044F.VOB", "F1044G.VOB", "F1044H.VOB", "F1044I.VOB", "F1044L.VOB", "F1044M1.VOB", "F1044M2.VOB", "F1044N.VOB", "F1044O.VOB", "F1044P.VOB", "F1044Q.VOB", "F1044R1.VOB", "F1044R2.VOB"])
- the number of frames used in an interval to compute synchrony, Nota bene, there are 25 frames by second. (eg. numberOfFramesByInterval = 25*10)

This process is relatively long and can take several minutes, especially with short intervals.

It returns synchrony CSV files that can be found [here](#).

Headers :

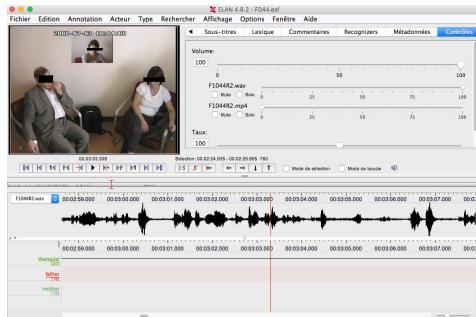
- **Interval**
- **Time_min**
- **video** : name of the video (eg F1044C.VOB)
- **SSI_fa_mo** : synchrony index between the filtered motion history of the *father* and *mother*. It ranges from 0 (no synchrony at all, not any relationship between the different participants of the score) to 1 (complete synchrony, the different elements move a lot at the same moment or don't move at all at the same moment).
- **SSI_fa_mo_th** : idem between *father*, *mother* and *therapist*
- **SSI_fa_th** : idem between *father* and *therapist*
- **SSI_mo_th** : idem between *mother* and *therapist*

There is not any NA in this file. Only the possible combinations are computed (eg there is not the *patient (pa)* in the video so, there isn't *SSI_fa_pa* signal neither any other combination with the patient).

B. Speech extraction

1. Manually

It is possible to use the software Elan to annotate a video. You need to extract first the sound (with [Free Video Converter](#) for instance) to get a waveform and import the video and the waveform.



Then you can annotate when someone is speaking. This process is very fastidious (1 week work for one hour video at the beginning and so was gave up).

2. Automatically

Idea					
	<p>Some softwares could be used to make this process easier. The FASST software get very good results to segment a music sample in very different instruments. However, different voices are too close to distinguish them easily automatically.</p> <p>Source separation examples</p> <table border="1"> <tbody> <tr> <td style="text-align: center;"> Linear instantaneous mixture (blind separation)  FASST Matlab script </td><td style="text-align: center;"> Professionally produced music recording (non-blind separation)  FASST Matlab script </td></tr> <tr> <td style="text-align: center;"> Separated sources Source 1  Source 2  Source 3  </td><td style="text-align: center;"> Separated sources Bass  Drums  Melody  Remaining sounds  </td></tr> </tbody> </table>	Linear instantaneous mixture (blind separation)  FASST Matlab script	Professionally produced music recording (non-blind separation)  FASST Matlab script	Separated sources Source 1  Source 2  Source 3 	Separated sources Bass  Drums  Melody  Remaining sounds 
Linear instantaneous mixture (blind separation)  FASST Matlab script	Professionally produced music recording (non-blind separation)  FASST Matlab script				
Separated sources Source 1  Source 2  Source 3 	Separated sources Bass  Drums  Melody  Remaining sounds 				