Software for Sensing Ultrasound Localization Microscopy (sULM)

User Guide

Version 1.0 - January 23, 2023

Authors: Louise Denis, Sylvain Bodard, Vincent Hingot, Arthur Chavignon, Jacques Battaglia, Gilles Renault, Franck Lager, Abderrahmane Aissani, Olivier Hélénon, Jean-Michel Correas, Olivier Couture

Corresponding authors: Louise Denis, Jacques Battaglia, Olivier Couture

Academic reference to be cited: Details of the code in the article by Denis, Bodard, Hingot, Chavignon, Battaglia, Renault, Lager, Aissani, Hélénon, Correas, and Couture. *Sensing Ultrasound Localization Microscopy reveals glomeruli in living rats and humans*, In prep, 2023.

Inspired from: code and article by Heiles, Chavignon, Hingot, Lopez, Teston, and Couture. *Open Platform for Ultrasound Localization Microscopy: performance assessment of localization algorithms*, Nature Biomedical Engineering, 2022.

Contents

General information

Aim

`Akebia` is a software for sensing Ultrasound Localization Microscopy (sULM). With Ultrasound Localization Microscopy (ULM), individual and isolated microbubbles are localized with sub-pixel precision and tracked into long trajectories (see PALA). It enables microvascular imaging in various organs.
Using an approach of double-filtering and double-tracking (see details in article), we can image stationary microbubbles as well as moving microbubbles in larger arteries. Such precision opens the possibility to classify particular microbubble motion patterns corresponding to expected microscopic structures. The individual tracking of a micrometer-sized microbubble in arterioles and capillaries can be viewed as a probe of its immediate environment and the various forces impacting the agent: a form of sensing Ultrasound Localization Microscopy (sULM).

Minimum requirements

`Akebia` software integrates MATLAB Runtime R2021a (9.10) (MathWorks). We recommend using Microsoft Windows 10 (version 1803 or higher), with 8 GB RAM minimum, and 5 GB of free space. This space does not include the space required for data.
For the example of the rat's blocks 10 to 20 and 27 to 32, the RAM needed is 8GB. You need to have Windows C++ redistributable to make the `Akebia` software work properly. If it's not the case, click into the link to download it.

Installation instructions

1. Download `Akebia` from online repository
2. Unpack the archive in a preferred directory
3. Launch the installer `Akebia_v1_installer.exe`
4. Follow the instructions
5. Restart the computer
6. `Akebia` should be now available for use in the start-up menu

Complementary scripts

This graphical interface is accompanied by several scripts and functions that detail its operation. You can find all the information in the `GitHub` folder here.
We strongly recommend the user who is comfortable with programming to read the script `Akebia_example_script.m`, which can be considered as the main one.

Input data type

`Akebia` software accepts a list of `.mat` file with images organized in a *[space, space, time]* matrix. Images can be complex or real, without log compression. We recommend user to split their dataset if the data is too heavy. For a computation for the example of the rat's blocks 10 to 20 and 27 to 32, the RAM needed is 8GB.

For non-interpolated data (as for rats in the article), your matrix must be called *IQ* for linear data, inside each `.mat`. If it's not the case, you will get an error inside the `Akebia` software or in the `Akebia_example_script.m`.
For interpolated data (as for humans in the article), your matrix must be called *bubbles* inside each `.mat` or you will get similar error.

In-vivo rat 14 dataset is available in the `Zenodo` folder at here (same rat as in the article). Patients imaging data are not available due to ethical, medical and legislative considerations toward personal information.

Image visualization

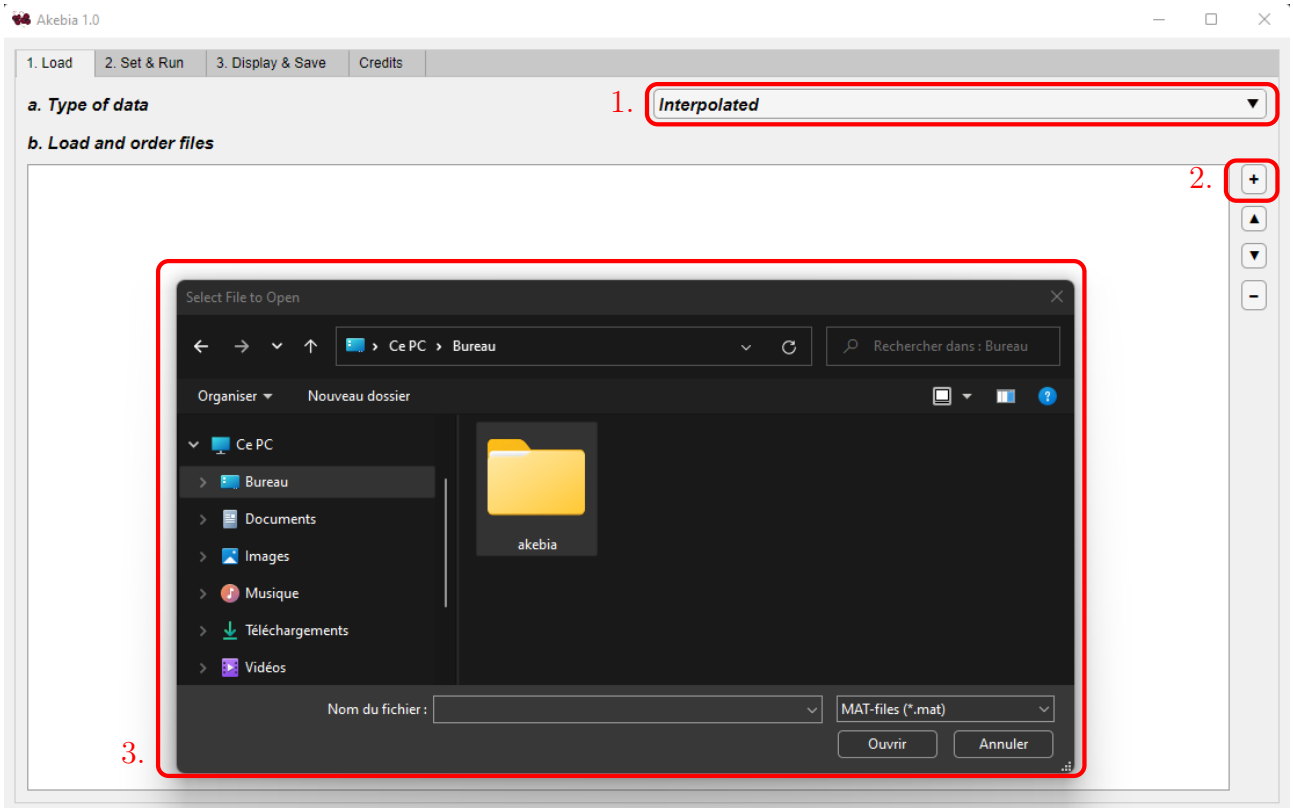
The `Akebia` software allow adjusting of several display parameters, such as normalization, saturation and gaussian fitting. If you want a better precision in adjustment, you can save images and adjust brightness, contrast and color balance with `ImageJ`.

Any questions should be addressed to the corresponding authors.

Disclaimer

1 Load data from the Akebia Software

Once you opened the **Akebia** software, the step **1.** is to select your type of data: Interpolated or Not Interpolated. Then in **2.**, you have to click on the + button to open the browser and **3.** to add the list of wanted `.mat` files.



If you want to reorganize the order of your `.mat` files, you can use the **4.** arrow buttons. It is important to precise that your data will be processed in the order fixed here. If you want to delete a `.mat` file, you can use the **5.** - button.

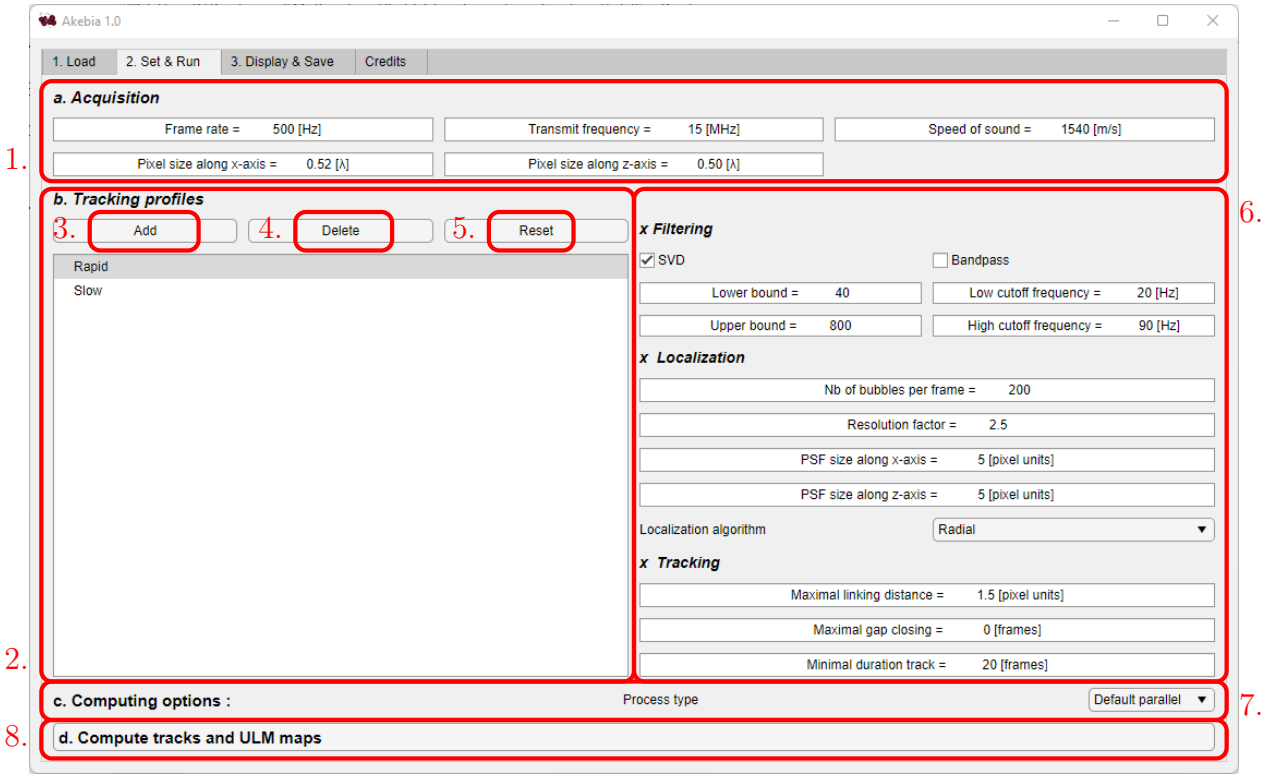
2 Set parameters and run the program

The second tab "Set & Run" available at the left corner is there to define **1.** parameters of your acquisition and **6.** parameters of sULM for filtering, localization and tracking. By default, parameters are fixed as in the article for non-interpolated data (rat 14) and interpolated data (patient 1). All parameters described in this section have their correspondents in the `Akebia_example_script.m`. A table that recapitulates all parameters is available later in this document.

Acquisition parameters are defined as followed:

- Frame rate: number of images per second [Hz].
- Pixel size along x-axis: width of pixel in [λ].
- Pixel size along z-axis: height of pixel in [λ].
- Transmit frequency: frequency of your acquisition probe [Hz].
- Speed of sound: speed of sound in your media [m/sec].

For this version of `Akebia`, we consider that pixel was isometric for interpolated data, i.e. pixel size along x-axis is equal to pixel size along z-axis.



Different tracking profiles:

The specificity of `Akebia` compared to existing platform `Lotus`, is that you will perform several times the sULM proceeding for the same dataset. That's what we called "Tracking profiles" in **2.**. By default, you have a "Slow" tracking profile, for stationary microbubbles, and a "Rapid" tracking profile, for microbubbles in larger vessels (see details in the article). You can add

3. or delete **4.** tracking profile (except for "Slow" and "Rapid" tracking profiles which are mandatory in this version). You can also **5.** reset default tracking profiles which corresponds to "Slow" and "Rapid" with default sULM parameters (as in the article).

For each tracking profile, you must define sULM parameters **6.** : filtering, localization and tracking parameters as described below. You can also rename each tracking profile by double clicking on it (except for "Slow" and "Rapid" tracking profiles which are fixed in this version).

sULM parameters are defined as followed:

- SVD filtering: adjust the minimum and maximum kept eigenvalues. This filter is available only for non-interpolated acquisition: spatio-temporal filter cannot be done on Contrast Enhanced Ultrasound Sequences (CEUS), i.e. interpolated data, because images are already filtered to enhance microbubbles signal.
- Bandpass filtering: adjust the range of the bandpass filter [Hz].
- Localization:
  - Nb of bubbles per frame: estimated number of microbubbles in a unique frame.
  - Resolution factor: the resolution factor determines by how much the original pixel size will be multiplied.
  - PSF size along x-axis: the estimated width of the Full Width at Half Maximum of the microbubbles in [pixel units].
  - PSF size along z-axis: the estimated height of the Full Width at Half Maximum of the microbubbles in [pixel units].
  - Localization algorithm: select a localization algorithm (as defined in `Pala`).
- Tracking:
  - Maximal linking distance: maximum distance between two microbubbles inside the same frame that can be paired [pixel units].
  - Minimal trajectory length: minimal duration of the track to be kept [frames] (see simpletracker on Github).
  - Maximal gap closing: maximum number of [frames] that can be jumped to link microbubbles together.

Parallel execution

Before running sULM with selected parameters, you need to choose **7.** your computing options, which corresponds to the selection of the number of workers that will work on parallel to accelerate the calculation. If you don't know how many workers your computer has, select "Sequential" option.

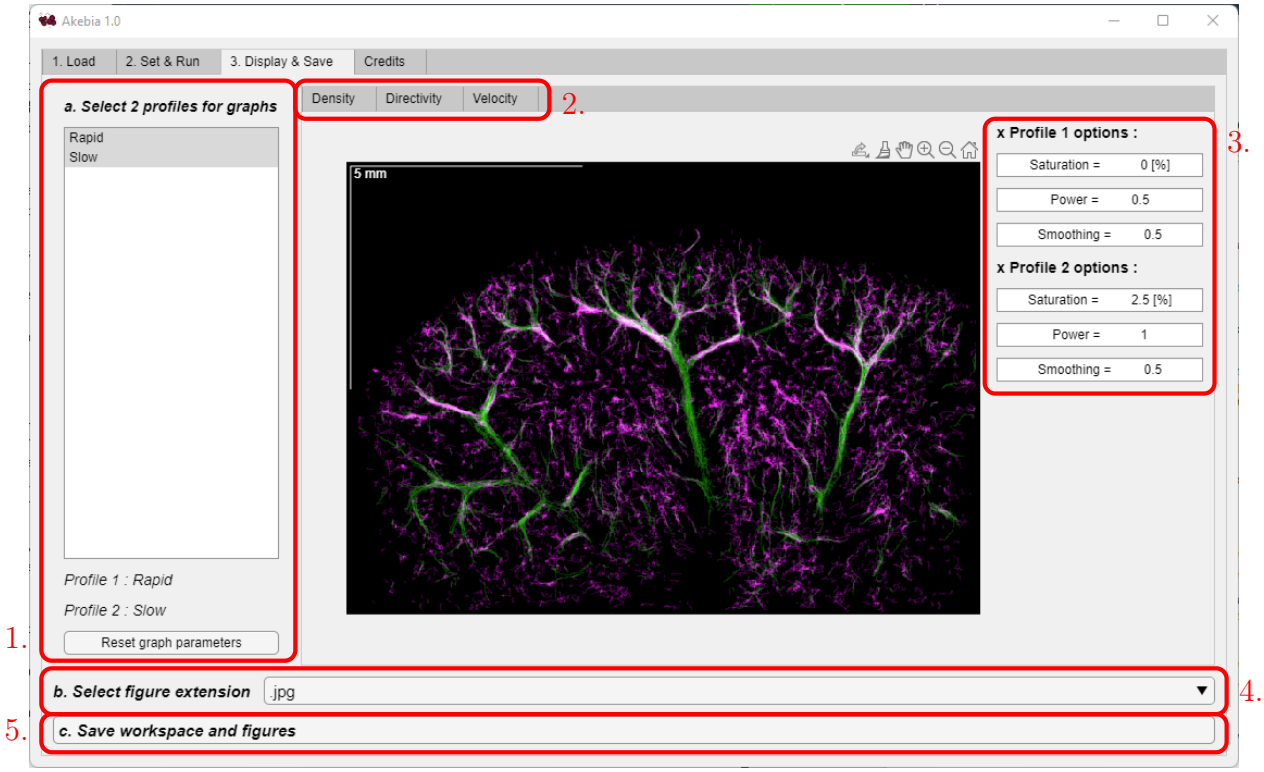
- Default parallel: let Matlab select the optimal number of workers for your calculation (this depends on the parameters defined on Matlab).
- Sequential: Matlab doesn't use parallel workers, it will run sequentially the program (it may be long).
- Custom parallel: if you know exactly how many workers are present on your computer, you can enter it on the `Akebia` software to optimize the calculation time.

Once you fixed all your tracking profiles and all your parameters, you can click on **8.** "Compute tracks and ULM maps". This may take several minutes.

Warning: The firewall may block the software from computing the tracks and sULM maps due to parallel pooling. This message is known and you have to permit the application execution.

3 Displays

When the calculation is complete, the third tab "Display & Save" opens and you can **1.** set the tracking profiles you want to display, **2.** choose the type of map you want to look at and **3.** set the display settings you want to adjust. Each type of mapping has specific display settings.



Different map types

As a reminder, the different maps **2.** are:

- Image density: based on microbubbles counts, pixel intensity codes the number of microbubbles crossing this pixel.
- Image directivity: axial color encoding, pixel intensity codes the number of microbubbles crossing upward/downward.
- Image velocity: velocity magnitude image, pixel intensity represents the average microbubbles velocity in mm/s.

By default, display parameters are fixed as in the article, and if you let `Akebia` runs with the blocks 10 to 20 and 27 to 32 present in the `Zenodo` folder of online repository, you will get the same picture for rat 14 as in the article.

Display parameters

The display parameters **3.** are different for each map type and will be described in this section.

- For density display you can adjust:
  - Saturation: higher it is, higher the image is saturated, between 0 and 100 [%].
  - Power: power of the image, between 0 and 1.

  - Smoothing: gaussian kernel to smooth the image, between 0.1 and 2.

- For directivity map you can adjust of each tracking profile and of the composite image (i.e. the superimposed "Rapid" and "Slow" image):
  - Saturation: higher it is, higher the image is saturated, between 0 and 100 [%].
  - Power: power of the image, between 0 and 1.
  - Smoothing: gaussian kernel to smooth the image, between 0.1 and 2.

- For velocity map you can adjust:
  - *vMaxDisp*: value of the maximum speed with which to normalize the image [mm/sec].
  - Saturation: higher it is, higher the image is saturated, between 0 and 100 [%].
  - Power: power of the image, between 0 and 1.
  - Smoothing: gaussian kernel to smooth the image, between 0.1 and 2.

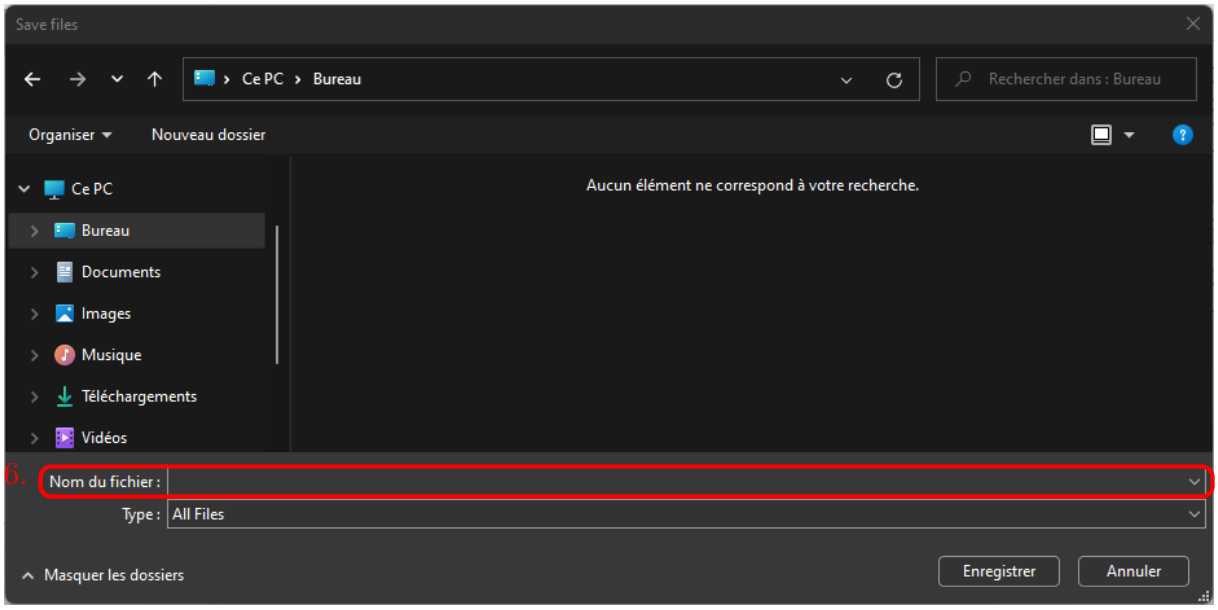
Results of each map type for blocks 10 to 20 and 27 to 32 of rat 14 with all default parameters (same as in the article) are presented in the `/Results` folder on `Zenodo` and on `GitHub` (available temporarily in online repository). In the same `/Results` folder, you can also find these 3 types of maps for the patient 1 (same as in the article).

Figures

For sULM maps described above, you have to select the **4.** extension (`.jpg`, `.pdf`, `.png` or `.tif`).

Workspace data

Then, you have to click on "Save" **5.** to select the location and the names **6.** for your figures and your workspace data.



Your `workspace.mat` will contain each tracking profile as a structure for which the fields are detailed below:

- *params* : parameters defined for computing the tracks

- *tracks* : each cell contains one track with variables in this order [*z*, *x*, *vz*, *vx*, *time*]
  - *z*, *x*: axial and lateral coordinated in [super pixel units].
  - *vz*, *vx*: axial and lateral speed in [λ] for non-interpolated data, and [pixels] for interpolated data.
  - *time*: interpolated time in [seconds].
- *density* : representing the density map
- *directivity* : representing the directivity map
- *velocity* : representing the velocity map

4 Corresponding parameters in `Akebia_example_script.m`

More details are available in the `README` inside the `/Example script` folder.

Acquisition parameters	*frameRate*	Number of frames per second [Hz]
	xPix2Lambda or *xPix2mm*	Width of a pixel [pixel units]
	zPix2Lambda, or *zPix2mm*	Height of a pixel [pixel units]
	transmitFreq	Frequency of your probe acquisition [Hz]
	speedOfSound	Speed of sound in your media [m/sec]
	lambda2mm	Value of λ in [mm]
sULM parameters	*name*	Name of your tracking profile
	dType	Type of IQ data, "interp" for human or "nointerp" for rats
	numberOfParticles	Estimated number of microbubbles to locate in a frame
	maxGapClosing	Number of frames jumped to pair 2 microbubbles
	minLength	Minimum duration of the track [frames]
	maxLinkingDistance	Distance between 2 microbubbles to be paired [pixel units]
	res	Resolution factor, factor that multiplies initial grid size [without unit]
	fwhm	Size in [pixel units] of the mask for localization along z and x-axis. (3x3 for pixel at λ, 5x5 at $\lambda/2$).
	locMethod	Localization method, see `Lotus`
	useBandpass, *bandpassBounds*	Use or not the Butterworth filter, Threshold for bandpass filter [Hz]
	useSVD, *svdBounds*	Use or not SVD, Threshold for SVD filter [eigen values]
	varNameInFile	"IQ" for non-interpolated data, and "bubbles" for interpolated data
	blockSize	Number of frames in a block [frames]
Display parameters	*denSig*	Smooth factor for density maps
	denExp	Power factor for density maps
	denSat	Saturation factor for density maps
	denSatForDirMap	Saturation factor for directivity maps
	denExpForDirMap	Power factor for directivity maps
	denSigForDirMap	Smooth factor for directivity maps
	cDirExp	Power factor for total directivity map
	cDirSig	Smooth factor for total directivity map
	vMaxDisp	Maximum velocity [mm/sec] to normalize velocity map
	cDenSat	Saturation factor for velocity map
	cVelSig	Smooth factor for velocity map
	cDenExp	Power factor for velocity map