

Look@NanoSIMS*

Lubos Polerecky[†]

Max-Planck Institute for Marine Microbiology, Bremen, Germany (2002–2013)

Department of Earth Sciences, Utrecht University, Utrecht, The Netherlands (2013–2025)

User's manual, version 28-06-2025

<https://github.com/lpolerecky/LANS>

Summary

Look@NanoSIMS, or shortly **LANS**, is a program for the analysis and processing of nanoSIMS data. Its primary aim is to be *useful*. The original development of the program started back in 2008, so in many aspects the program may be considered as an ‘old school’. However, based on input and requests from users, the program has been improved in many ways, and an active development continues until today. Over the years, the program has matured well and can now be used for many types of analysis of nanoSIMS data. This document describes how such analyses can be done at basic, intermediate and very advanced levels.

Contents

1	Installation instructions	2
1.1	Install Matlab	2
1.2	Install L ^A T _E X	2
1.3	Install software for compressing/decompressing files	3
1.4	Install Look@NanoSIMS	3
1.5	Starting Look@NanoSIMS for the first time	3
1.6	Starting Look@NanoSIMS on a regular basis	4
1.7	Updating Look@NanoSIMS	4
2	Organization of the input and output data	5

***Citation:** L. Polerecky et al. (2012). Look@NanoSIMS — a tool for the analysis of nanoSIMS data in environmental microbiology. *Environmental Microbiology* 14 (4): 1009–1023. DOI: 10.1111/j.1462-2920.2011.02681.x

[†]Feedback and questions can be sent to: l.polerecky@uu.nl

1 Installation instructions

LANS is a Matlab-based program. Thus, a core **Matlab** installation, along with image processing and statistical toolboxes, is required to run LANS and perform basic functions. This makes it possible to run LANS on a variety of operating systems, including Linux, Microsoft Windows and MacOS. However, this also limits LANS to users with access to a Matlab license. Additionally functionality of LANS is achieved by integrating the program with **L^AT_EX** (for exporting results in a nicely formatted PDF output) and data compression programs such as **zip** (for decompressing input files and compressing output generated by LANS), both of which are available for free.

1.1 Install Matlab

1. Matlab is available from www.mathworks.com and requires a license. It is useful to check whether your institution has a site-license (e.g., your university may have one for all students and academic staff).
2. Version 2019b of Matlab is most recommended to ensure that all features of LANS work as designed. LANS works with newer Matlab versions as well, however, there is a risk that some functionality will issue errors due to less than perfect back-compatibility of Matlab.
3. When installing Matlab, you will need the *core Matlab* and *two toolboxes*: image processing and statistics and machine learning.

Note

Some output generated by LANS, e.g., information generated during the alignment of planes and stored in the file `xyalign.mat`, may not be read correctly by the Matlab version 2019b when it was generated by a newer Matlab version. Thus, if you plan to use LANS in collaboration with other people, e.g., by sharing the files generated by LANS among each other, it is recommended that everyone in the team uses the same Matlab version.

1.2 Install L^AT_EX

This software is required to enable export of graphical output as tagged PDF documents.

1. To install L^AT_EX, use one of the well-established L^AT_EX distributions for your operating system, as described on the [L^AT_EX project](http://www.latex-project.org) website (e.g., **texlive** for Linux, **MikTeX** for Windows, **MacTeX** for MacOS). Note that the on-line LaTeX service, such as Overleaf, is insufficient; you really need a locally installed L^AT_EX distribution.
2. To correctly integrate L^AT_EX with LANS, you will need the following executables and packages installed and working:
 - executables: `epstopdf`, `pdflatex`
 - packages: `graphicx`, `geometry`, `hyperref`

Note

If you have never used L^AT_EX on your computer, it may be that some L^AT_EX packages, particularly `geometry` and `hyperref`, are not installed during the 'standard' installation procedure.

As a result, the execution of the LANS functions **Export LaTeX and PDF output** (main LANS) or **Export images for each variable as PDF** (Process metafile) may get stuck if the packages are missing. If this happens, you can fix this problem by compiling the `tex` file using the native L^AT_EX environment (e.g., open the `tex` file in the default editor of your L^AT_EX distribution and then compile it into a `pdf` output from there). When doing so, the missing L^AT_EX packages should automatically be installed and the `tex` file should compile into a correct `pdf` output. Once this is done, the automatic L^AT_EX compilation from within Matlab will also work.

1.3 Install software for compressing/decompressing files

This software is required for two reasons.

1. It enables you to load compressed nanoSIMS datasets (`im.zip` files) by LANS. This is a useful feature because `im.zip` files have roughly a 10-fold smaller size than the original `im` files generated by the Cameca's nanoSIMS measurement software. It is recommended to store and distribute the raw data files by first compressing them with the `zip` program (extension `im.zip`).
2. It allows you to compress the processed data generated by LANS. This is convenient for making data backups, since it is much more efficient to upload and download few compressed folders than hundreds of smaller files present in those folders.
3. 7-Zip (freeware) is recommended for Microsoft Windows. `zip` and `unzip` are available by default on Linux and MacOS systems.

1.4 Install Look@NanoSIMS

This is done by copying the source files to a folder on your computer.

1. For convenience, the compressed file containing the *latest version of LANS* is stored in this [Dropbox folder](#). Click on the `program` folder and then download the file `LANS-latest-src.zip`.
2. Unzip `LANS-latest-src.zip` to a folder of your choice.
3. Rename the `src` folder using a more reasonable name (e.g., `LANS-2025-05-26`, where the data will refer to the LANS version).

Note

In case the Dropbox link above does not work, because it became outdated or the official distribution location changed, visit the LANS' github repository or try to search the internet for more updated information. For users familiar with git and github, LANS can be downloaded by pulling the source code from the `src` folder in the LANS github repository: <https://github.com/lpolerecky/LANS>.

1.5 Starting Look@NanoSIMS for the first time

Before you run LANS for the first time, revise the content of the files `lookatnanosims.m` and `lans_paths.m`. These two files contain important settings you need to adjust to reflect your specific local installation of LANS. For example, you can specify there:

- locations of the compression/decompression software,
- location of the PDF viewer,
- default name of the file containing regions of interest (ROIs),
- default extension of the raw data files.

Note

If you browse through the LANS installation files, you will notice that the `*.fig` files, which define the graphical user interface (GUI), appear in two sub-folders: `figs` and `figs_win`. This is required because GUI defined on Unix-like and Windows platforms do not look the same. This is an issue due to — apparently — limited cross-platform compatibility of Matlab visual objects. It is not important for you as an end-user of LANS. You only need to be aware of it. Should you wish to modify any of the `*.fig` files, you will need to do it twice.

1.6 Starting Look@NanoSIMS on a regular basis

1. Start Matlab.
2. Set the current folder to the folder where you installed LANS. You can do this through the menu or, more easily, entering one of the following commands in the Matlab console (the precise syntax depends on whether you use Windows, Linux or MacOS, and on the path where you installed LANS):

```
>> cd c:/programs/LANS
```

```
>> cd /home/your_username/programs/LANS
```

3. Once you are in the correct folder, enter the following command in the Matlab console:

```
>> lookatnanosims
```

If everything is set up correctly, the main LANS graphical user interface (GUI) will open (Fig. 1). You can start from there, as described in the following sections of this document.

Note

During the data processing session, LANS provides quite a lot of useful information in the Matlab console. Thus, it is a good idea to *always* keep an eye on the output in the console. You can do this by arranging your desktop such that the LANS and Matlab console windows are *both visible at the same time*. This is also useful in case you encounter an error while working in LANS. These errors will be shown in the console, too.

1.7 Updating Look@NanoSIMS

1. LANS is updated quite regularly. You can update it easily by entering in the Matlab console:

```
>> lans_webupdate
```

You need to be in the folder where LANS is installed. You will be prompted to make a backup of your older LANS version, which is recommended to do, just in case.

2. If you are familiar with `git`, you can update LANS by pulling the latest sources from the LANS github repository <https://github.com/lpolerecky/LANS>.

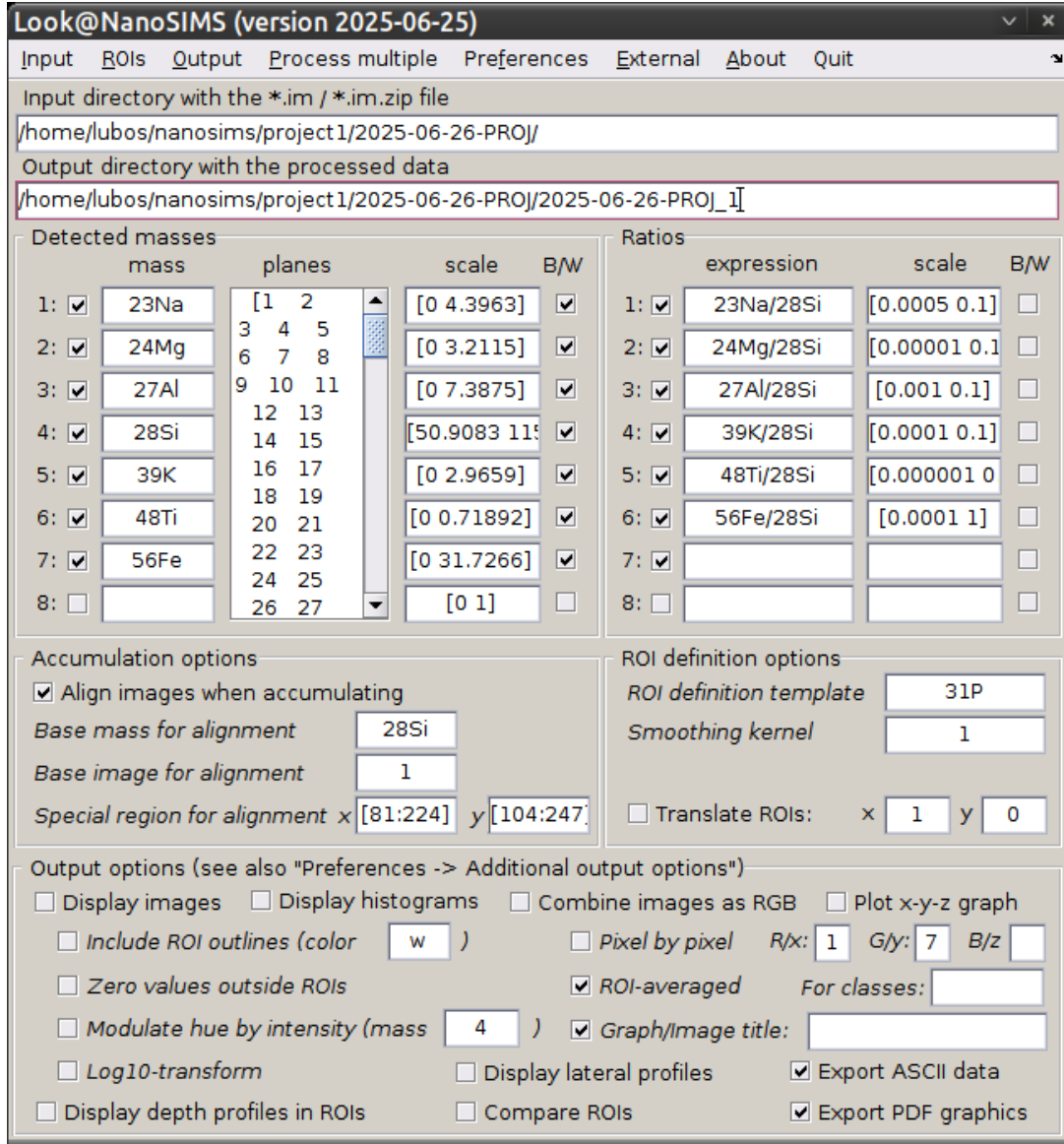


Figure 1: Main graphical user interface (GUI) of Look@NanoSIMS.

2 Organization of the input and output data

Working with LANS, and with nanoSIMS data in general, can be a book-keeping challenge. To start with, you will have many raw data files (im or im.zip files) acquired at different dates and from different types of samples (e.g., different treatments). Additionally, processing of those raw data files with LANS will create many output files, including ASCII data, PDF images, matlab output, PDF output, and zipped backed-up folders. It is therefore a good idea to develop and maintain a certain structure of those folders and output files, to *keep everything organized*.

In this document, we assume that the raw and processed nanoSIMS data are organized hierarchically as shown in Table 1. We have used this data organization at Utrecht University for many years, and it works pretty well. We therefore highly encourage users to adopt it as well. Its benefits will become more apparent later on, when we get to the point of explaining how to efficiently process and analyze *multiple* nanoSIMS datasets from a particular project.

Table 1: Hierarchical organization of the raw and processed nanoSIMS data implemented in Look@NanoSIMS. An example of such data organization along with a more detailed explanation is shown in Fig. 2.

root	→	project	→	measurment day 1	→	raw dataset 1.1	→	dataset folder 1.1				
						→	raw dataset 1.2	→	dataset folder 1.2			
						→	...	→	...			
				→	measurement day 2	→	raw dataset 2.1	→	dataset folder 2.1			
						→	raw dataset 2.2	→	dataset folder 2.2			
						→	...	→	...			
				→	...	→	...	→	...			
<hr/>												
dataset folder i	→	dat			→	text output (ion counts, ratios)						
	→	pdf			→	graphical output (images, graphs)						
	→	processing definition files (alignment, ROIs, preferences)										
	→	OutputG.pdf (output summary)										
<hr/>												

At the highest level of the data organization is a root folder that contains *all* nanoSIMS data. This folder contains ‘project folders’ with nanoSIMS data belonging to *specific projects*. Each ‘project folder’ contains ‘day folders’ with data acquired on *different measurement days*. Each ‘day folder’ contains the actual *raw datasets* (**im** or **im.zip** files). When a particular raw dataset is processed and analyzed, the corresponding data is stored in a ‘dataset folder’ with the *same name* as the dataset. Each ‘dataset folder’ contains sub-folders with the *results* of the analysis, including numerical values such as ROI-specific ion counts or ion count ratios (folder **dat**), and graphical output such as images or scatter plots (folder **pdf**). The ‘dataset folder’ also contains information defining the processing steps, such as alignment of frames, definition and classification of regions of interest (ROIs), and preferences. This information is useful if you want to go back to the analysis of the same dataset after you have analyzed a different one, e.g., to perform quality checks or more in-depth analyses. Finally, the ‘dataset folder’ also contains a PDF file with a comprehensive graphical summary of the analysis. This file is useful if you wish to share the results for a specific dataset with project collaborators.

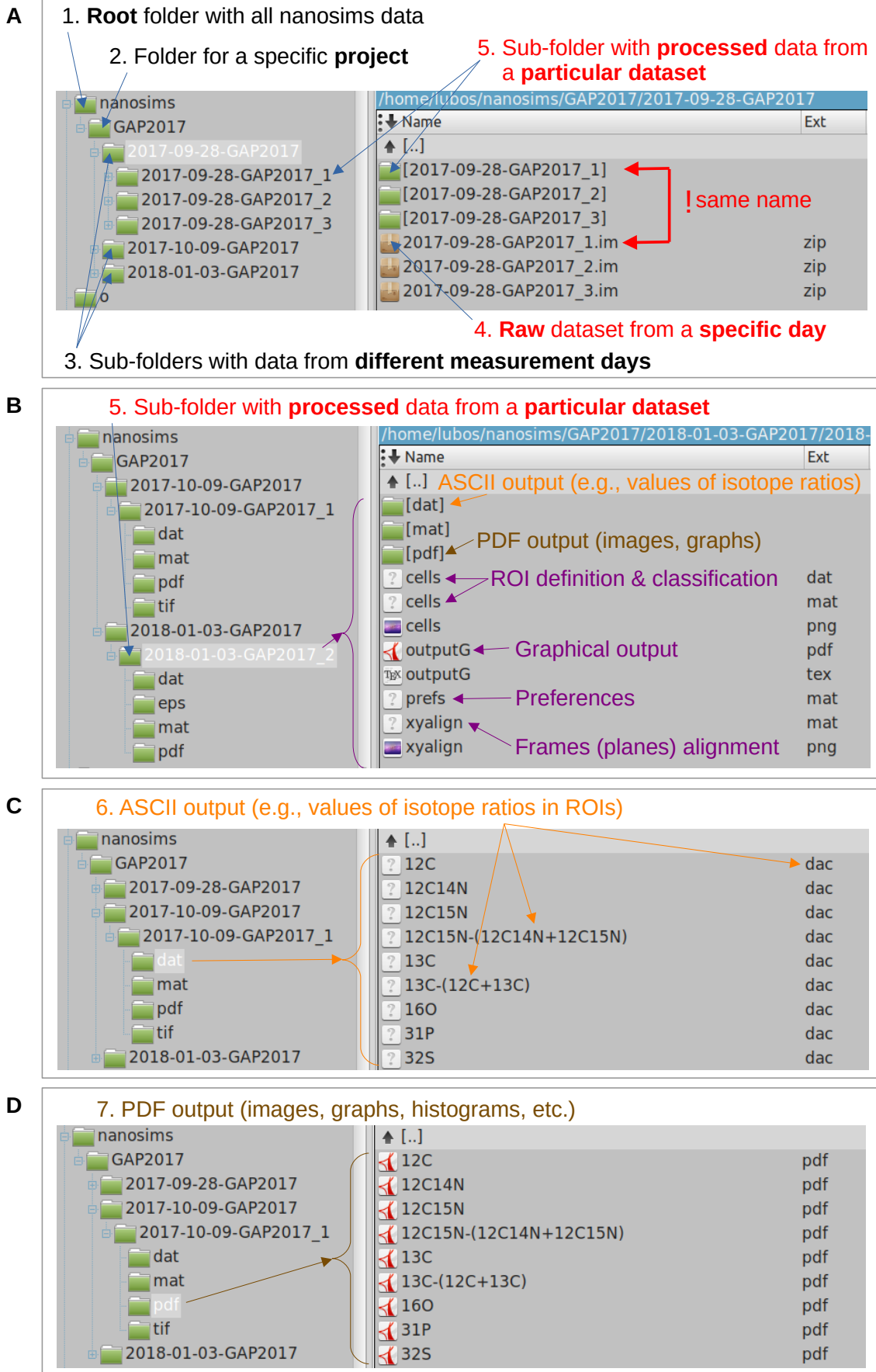


Figure 2: (see next page)

Figure 2 (see previous page): Example of a hierarchical organization of the raw and processed nanoSIMS data implemented in Look@NanoSIMS. **(A)** The root folder (**nanosims**) contains a project-specific sub-folder (**GAP2017**), which contains sub-folders with data measured on different days (e.g., **2017-09-28-GAP2017**, **2017-10-09-GAP2017**). The ‘day folder’ contains *multiple raw datasets* measured on that particular day (**im** or **im.zip** files). When a particular raw dataset is processed and analyzed, the data is stored in a ‘dataset folder’ with the *same name* as the dataset (see red ‘!’). **(B)** The ‘dataset folder’ contains files defining the processing steps, including alignment information (**xyalign**), definition and classification of regions of interest (**cells.mat** and **cells.dat**, respectively), preferences (**prefs.mat**), and a comprehensive summary of results exported in a PDF file (**OutputG.pdf**). The ‘dataset folder’ also contains sub-folders with ASCII (**dat**), PDF (**pdf**) and Matlab (**mat**) output. **(C)** The **dat** folder contains the ROI-specific ion counts and ion count ratios (**dac** files). **(D)** The **pdf** folder contains the exported images, graphs, histograms, etc.