**User guide to the AFM image nucleoprotein readout toolbox**

**1. System requirements**

The code was programmed and tested on Windows 10 Education Version 1803. However, it should in principle work on every Windows version that is capable of running Python and installing the required packages as described in “2. Installation guide”.

Besides Python no other software is required to run the automated image analysis toolbox. A guide on how to install Python and set everything up can be found in “2. Installation guide”.

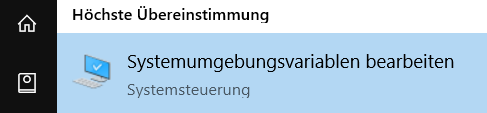
No special hardware is required.

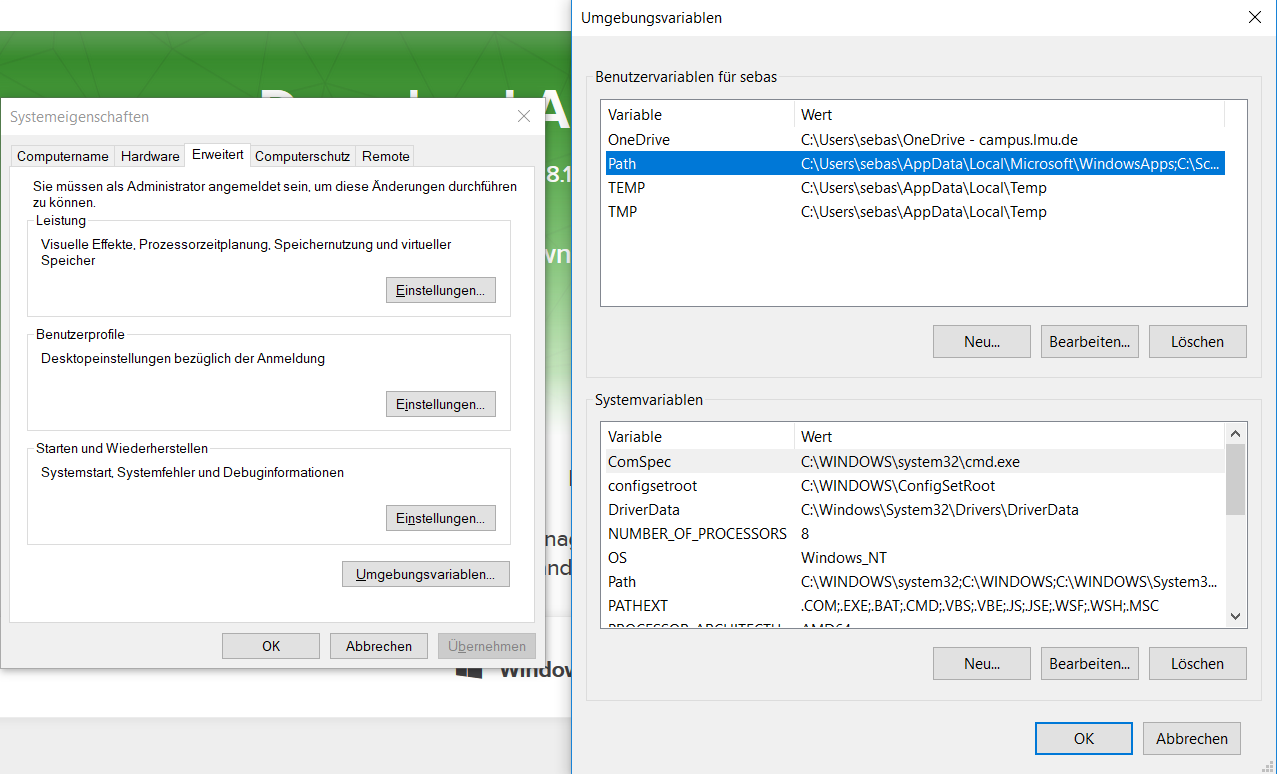
**2. Installation guide (Typical install time 15-30 minutes)**

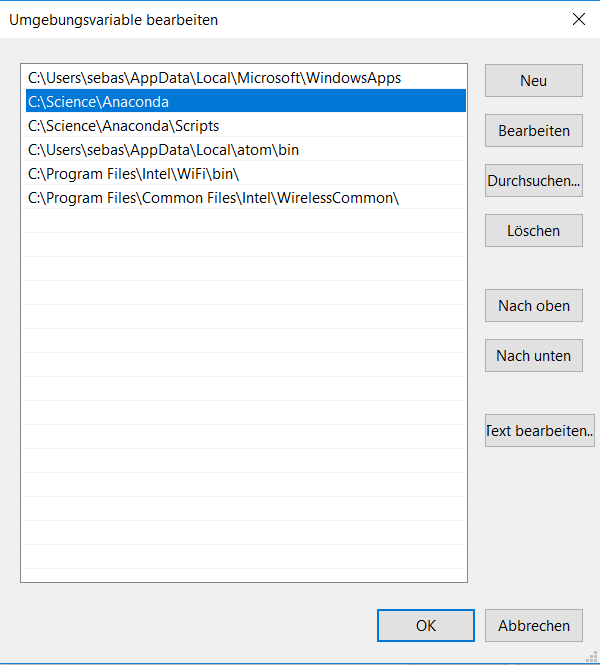
Required software: Python **3.x** (Python 3.7 used in this guide)

1. Easiest way to install Python 3 is by using Anaconda (<https://docs.anaconda.com/anaconda/install/windows/>)  
Download the installer for the Python 3.7 version and install it according to the installation guide in the link.

2. Add Anaconda to your system variables:

Type “environ” into the Windows search. Open System variables.

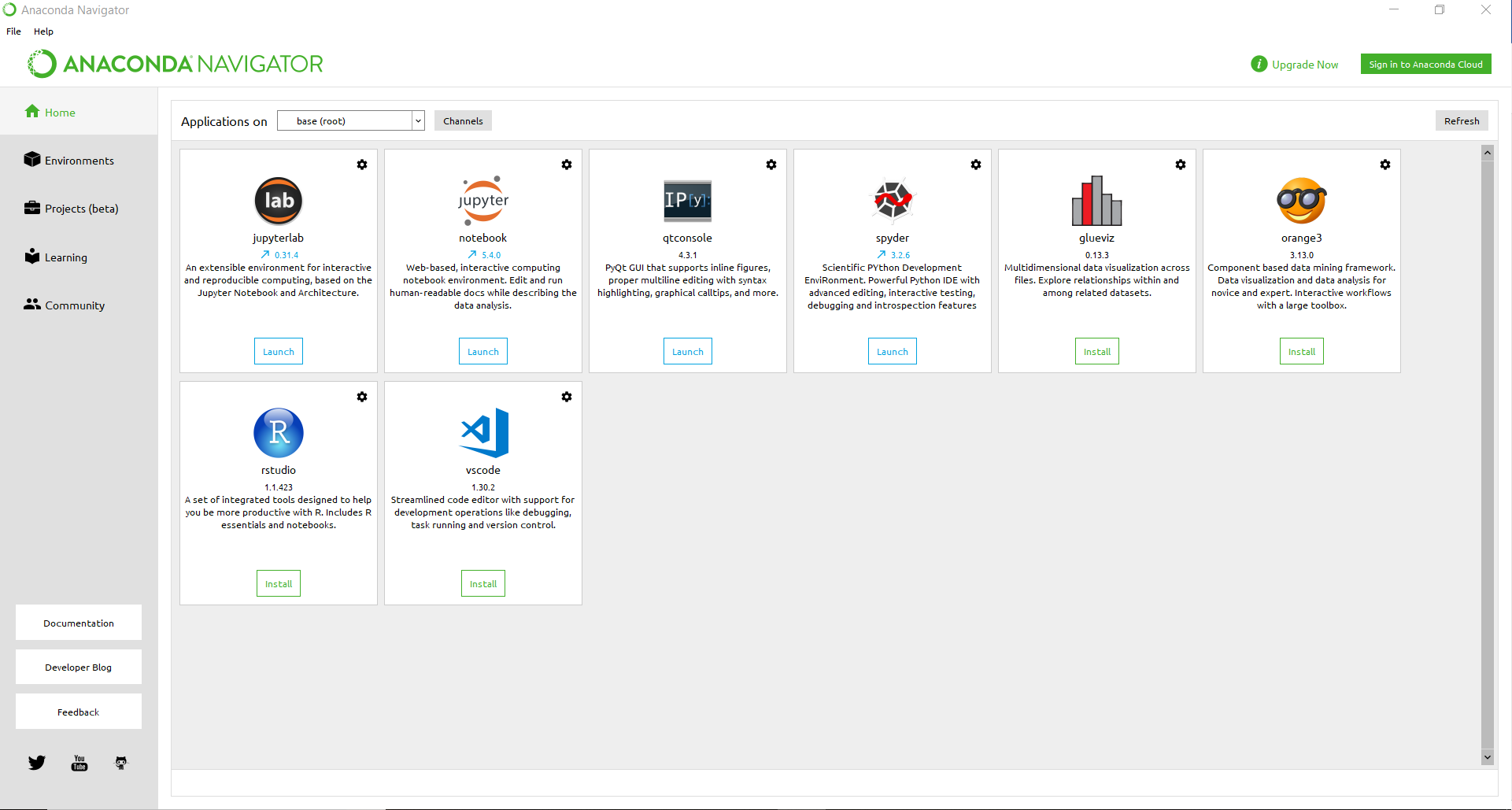
Press “Environment variables“ (or similar) on the bottom and the window on the right half will open. DoubleClick on Path.

  
Add the path of Anaconda (depending on where you installed it) to you system variables. In this case it is:  
“C:\Science\Anaconda”  
“C:\Science\Anaconda\Scripts”

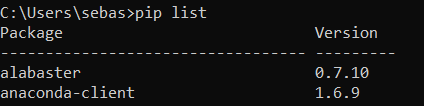
Press OK and close everything.

**RESTART YOUR COMPUTER**

3. Start the Anaconda Navigator

It should look similar to the following:

One of the Apps in this Window is “spyder”. If it is not installed yet, install it via pressing the “Install” button. Open to check that it is working properly. Close spyder again.

4. Installing Python packages

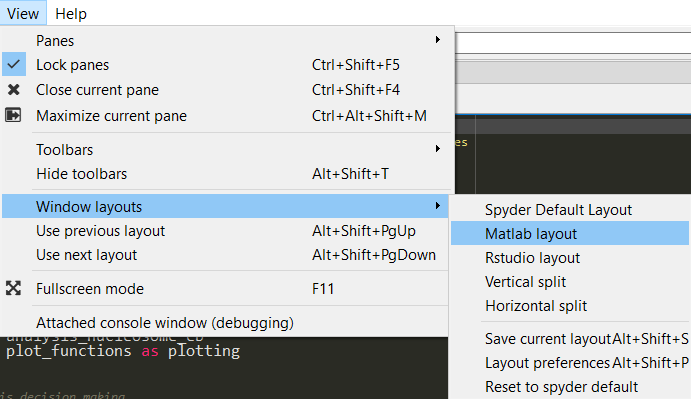
Open the command line via typing “cmd” into the command prompt. Check that you installed Python and Anaconda properly by typing “pip list” into the command prompt and “conda list” afterwards. In both cases, a list of different python packages should show up. If you get an error message similar to “pip was not found or was written wrongfully” you did not set up your system environment variables correctly.

Install the required packages one by one by subsequently typing the following commands into the prompt:

conda install -c conda-forge scikit-image

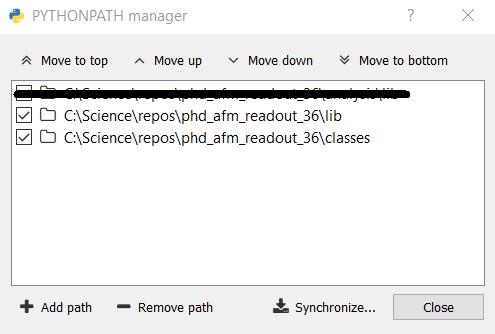
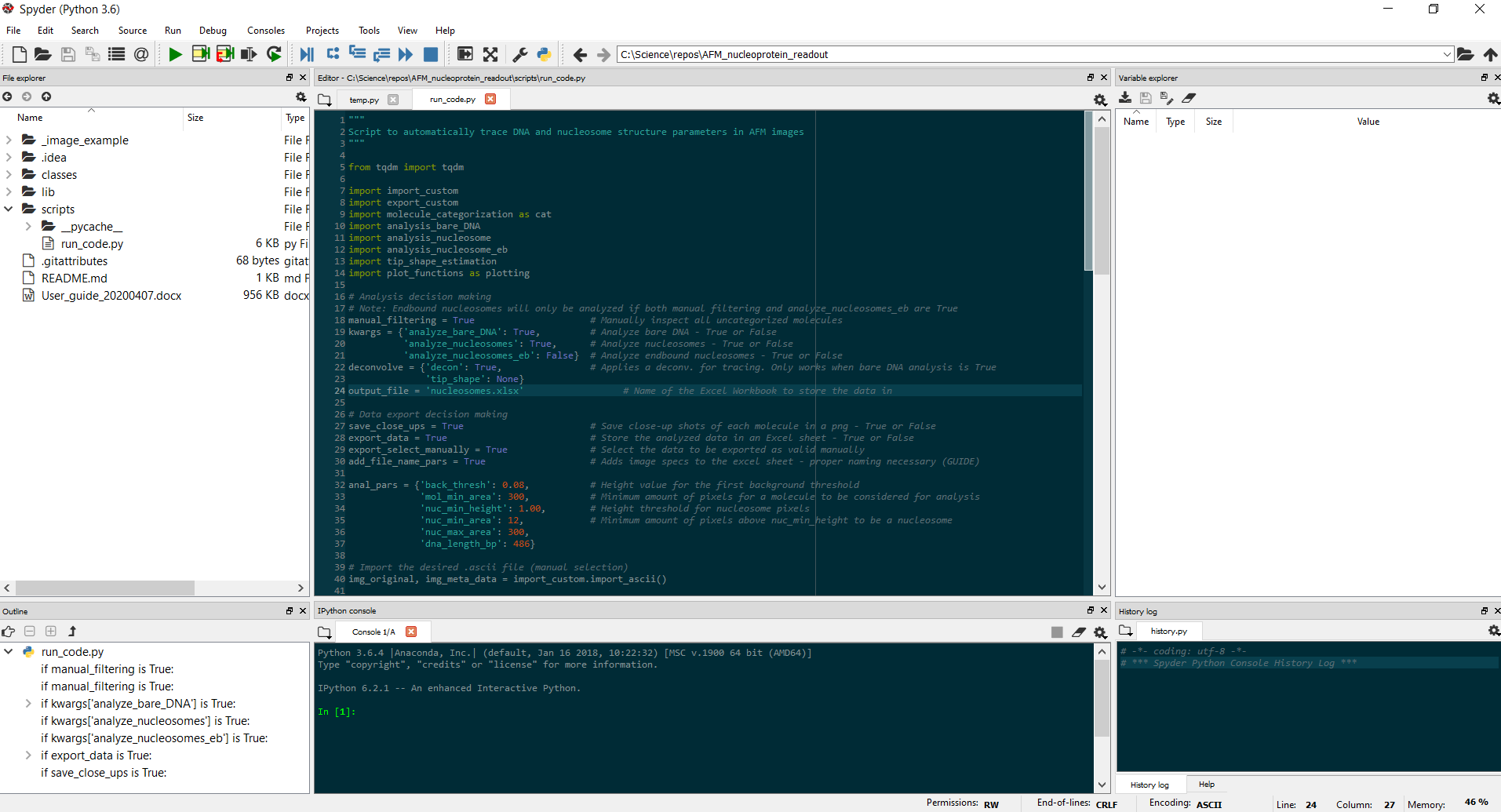
conda install -c conda-forge opencv

pip install pandas



5. Set up spyder

Open spyder in the Anaconda Navigator App window and choose the Spyder layout you prefer.

****  
In the file explorer window, navigate to where you stored the code on your computer. Additionally, by clicking on the open file icon on the top left of Spyder, open the “run\_code.py” which can be found in the scripts folder.

The final step is to integrate the code modules into Spyder: When clicking on Tools -> Pythonpath manager the window on the right will open. Use “Add path” to add the paths to the folders “lib” and “classes”. The file name before (in this case “C:\Science\repos\phd\_afm\_readout\_36” will vary based on where you store the code on your hard drive. **Close and restart Spyder!**

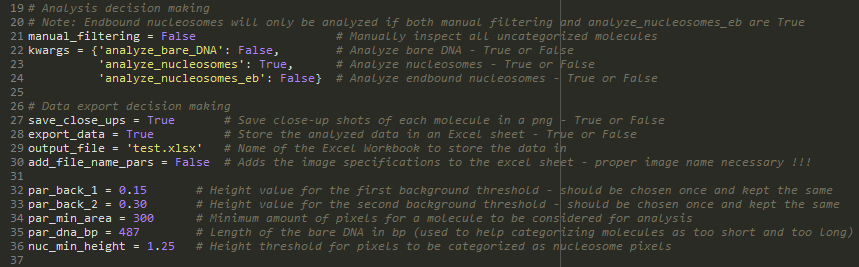
**In case you ever change the path of where the code modules are stored, you have to adjust the paths in this python path manager in Spyder.**

You should now be ready to run the AFM readout!

**3. Analyzing AFM images and instructions on reproducing the scientific findings**

Analysis time of the example image provided should be between 3-5 minutes on a standard computer.

1. **Set up analysis parameters**

Before starting analysis, choose the desired analysis parameters at first in the run\_code.py script (lines 19-36) as described in the following paragraphs.

**manual\_filtering: True or False**

Decide if you want to manually remove pixels and adjust the skeleton of molecules that could not be categorized properly by automatic detection. (more detail on manual filtering later and in Supplementary Figure 2!)

**kwargs: True or False**

For the different molecule types decide whether they should be analyzed or not. Important: Even if endbound nucleosomes are set to True, they will only be analyzed if manual\_filtering is also set to True.

**save\_close\_ups: True or False**

Store a close up .png of each detected molecule in a new subfolder,

**export\_data: True or False**

Save analysis results in an excel sheet. **Important: In case you want to export the data into an excel sheet and the excel sheet already exists, the sheet must be closed while the code is running otherwise the program will yield an error and not export anything.**

**output\_file: String with format ‘desired\_name.xlsx’**

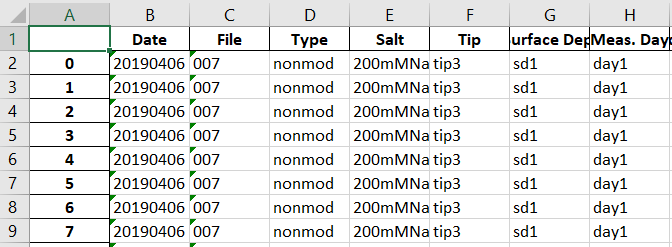
Choose the name of the excel sheet for data export. If it already exists, the data will just be appended on the bottom of the already existing data.

**add\_file\_name\_pars: True or False**

Add the measurement specifications for which the AFM image was recorded to the Excel sheet. If this is set to True it is important that the folder structure at which you store the .ascii AFM image files that you want to analyze are named correctly:

The images should be named the following way: date\_type\_salt\_tip. So in the case of the example image: 20190406\_nonmod\_200mMNaCl\_tip3. Additionally, the folder in which the images are stored must have the name in the following way: surfacedeposition\_daysincethesurfacedeposition. So in my example case: sd1\_day1.

The names of the parent folders are not important. **You will get an error if the folder structure is not correct and this parameter is still set to True!**

If it is set to True and your folder/image naming structure is correct the measurement specifications are automatically stored in the exported Excel sheet.

**Detection parameters (More detail on detection parameters and their effect further below)**

**par\_back\_1: float (e.g. 0.15)**

Height value in nm for removing background pixels in the first filtering step. Should be lower than par\_back\_2.

**par\_back\_2: float (e.g. 0.30)**

Height value in nm for removing background pixels in the second filtering step. Should be higher than par\_back\_1

**par\_min\_area: int (e.g. 300)**

Minimum number of pixels for a molecule not to be kicked out of the further analysis.

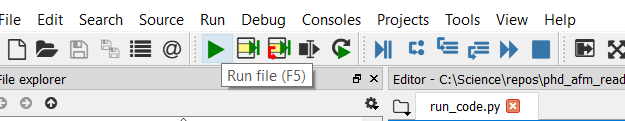
**par\_dna\_bp: int (e.g. 486)**

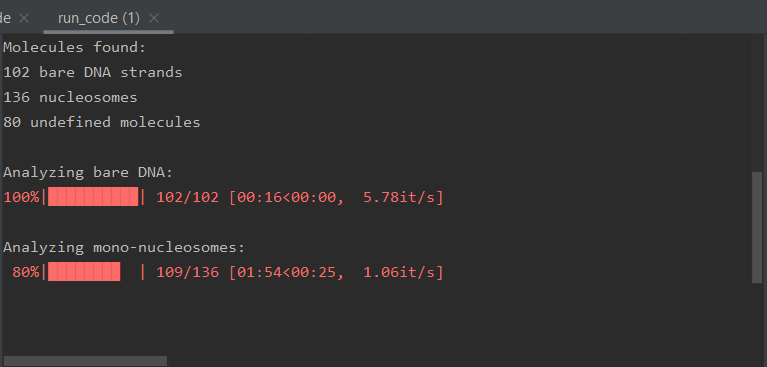
Number of base pairs of the DNA. This value helps in categorizing molecules. If it is not set properly, molecules might be categorized wrongfully.

**nuc\_min\_height: float (e.g. 1.25)**

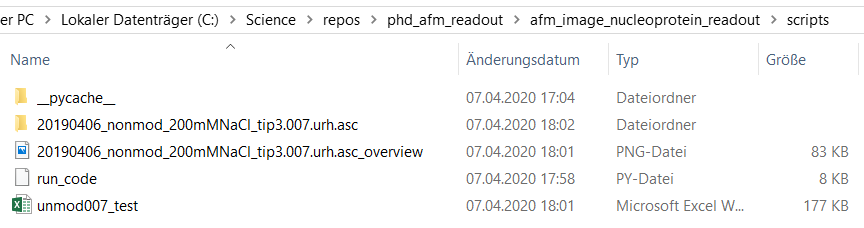
Height threshold in nm above which a pixel is thought of as a potential nucleosome core pixel.

1. **Starting the analysis**

Start the analysis by clicking on “Run file”. A window will open (may be in the background) where you can select the .ascii of the image you want to analyze. An example image is provided in the “\_image\_example” folder.

  
After selecting the image, analysis will start (**if manual\_filtering is set to False**). A number of molecules that were found is given and a progress bar for the individual analysis steps + remaining time is shown. After analysis has finished, a lot of windows will pop up in case you chose to store close ups of the individual molecules. Wait until everything is closed again.

**The analyzed data is stored in the “scripts” folder of the code.**

The close up shots are stored in a folder named according to the image name you just analyzed. The data values are stored in the Excel sheet.

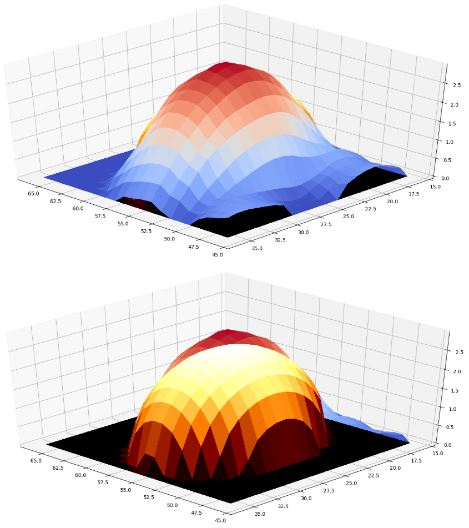
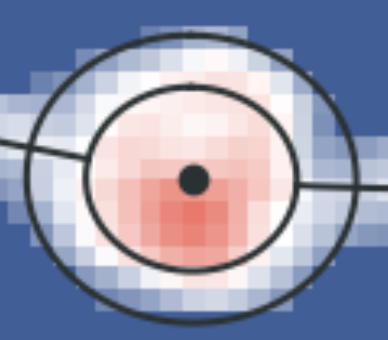
1. **Understanding the traced parameters**

The excel workbook is separated into three (based on the molecule types you chose to analyze) sheets named Bare DNA, Nucleosomes and Endbound Nucleosomes.

**Bare DNA parameters**

* **extension\_bot/right:** Extension of the molecule in nm from the left- to the rightmost and the top- to the bottommost pixel of the molecule
* **height\_avg/std:** Average height and standard deviation of the height values along the trace of the DNA strand
* **length\_avg:** Length of the traced DNA strand. The DNA strand is traced from both ends to the corresponding other end and the avg of both values is taken to increase length measurement accuracy.
* **position\_col/row:** Position of the molecule in the image
* **rightness:** Average of the orientation of the traced DNA segment. If value is 1 then the DNA is completely oriented horizontally, if value is 0 the DNA is completely oriented on the top/bot-axis
* **rog:** Radius of gyration
* **slope\_avg/std:** Average slope and standard deviation between adjacent points along the DNA trace.

**Nucleosome parameters**

* **angle\_arms:** Opening angle with which the two DNA arms move into the center of the nucleosome core
* **ellipsoid\_angle/height/width\_a/width\_b:** The nucleosome parameters are traced by fitting a rotating half ellipsoid to the actual height data (see image on the right). The value of the angle is not the real rotation in degrees but just a unitless parameter that is needed for proper fitting. Height is the maximum height of the ellipsoid. Width\_a and width\_b are the half widths of the ground ellipse in left/right and top/bottom direction respectively.
* **extension\_bot/right:** Same as for bare DNA just combined for both arms
* **height\_avg/std:** Same as for bare DNA just combined for both arms
* **length\_arm1/2\_40/50/60/70:** Length of arm 1 or arm 2 for different cut offs: The total length of an arm is calculated by tracing the Wiggin’s pixel along the DNA arm and finally adding a segment directed from the last traced arm pixel towards the center of the fitted ellipsoid. The length of this additional segment is cut off when the fitted ellipsoid reaches 40/50/60/70% of its maximum height. (See image). **60% was used for all length measurements presented in the manuscript.**

**Ground ellipse**

**60% ellipse**

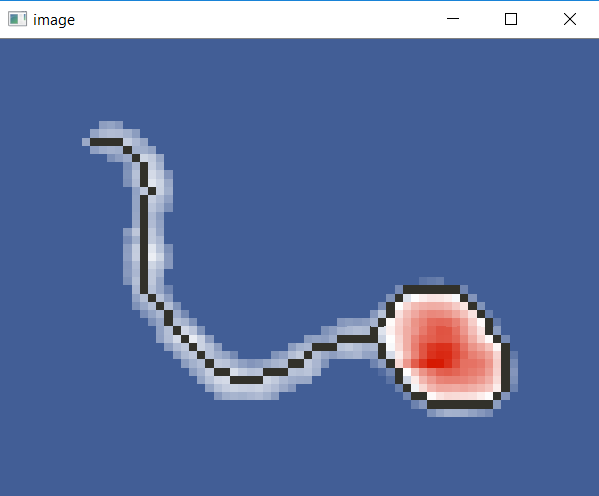
* **length\_etoe:** End to end distance in nm between the ends of the two DNA arms.
* **nucleosome\_volume/\_core:** Volume of the nucleosome in nm³. The volume is calculated by taking summing up the volumes of the pixels whose centers lie within the ground ellipse of the fitted ellipsoid. The volume\_core value is calculated the same way but just using all pixels that lie within the ellipse at 60 % of the maximum height of the fitted ellipsoid.
* **position\_col/row:** Position of the molecule in the image
* **rightness\_arm1/arm1:**  Same as for bare DNA just for each arm individually.
* **slope\_avg/std:** Average slope and standard deviation between adjacent points along the DNA trace.

**Endbound nucleosomes – parameters similar to the nucleosome parameters**

1. **More detail on manual\_filtering**

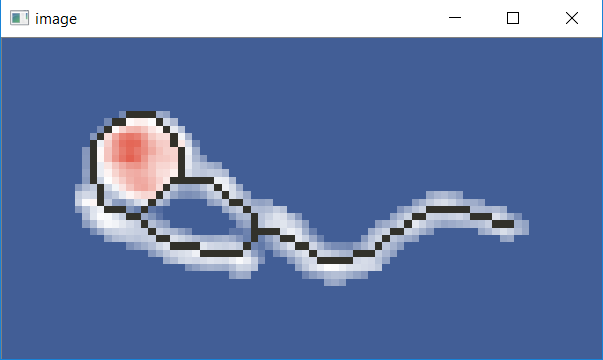
**Manual Filtering**

If the parameter manual\_filtering is set to True, after selecting an image for analysis, each molecule that was not able to be categorized will pop up in a window consecutively and it is possible to manually remove pixels to allow proper categorization. For example if for a nucleosome the arms overlap a little, it might be helpful to separate the arms manually. After removing the unwanted pixels, click ESC to continue to the next molecule.

**Examples**

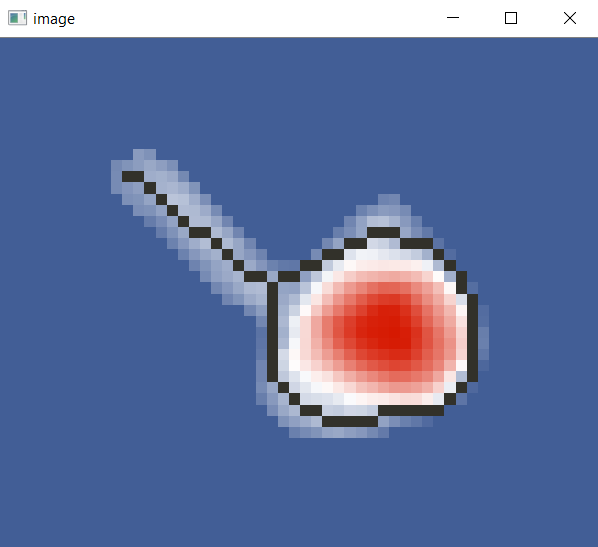
**Endbound nucleosomes**

Endbound nucleosomes will only be analyzed after manual inspection. All molecules that have exactly **1 endpoint and 1 branchpoint in the skeleton** (dark grey pixels) will be categorized as endbound nucleosomes. If you’re happy with the molecule the way it is just press ESC and it will be analyzed later.

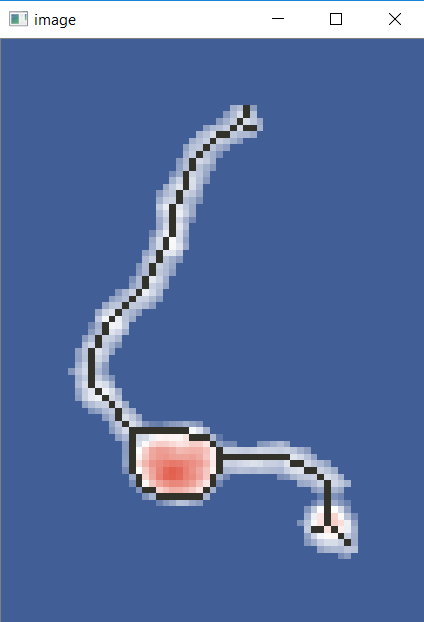
  
**Nucleosomes with overlapping arms**

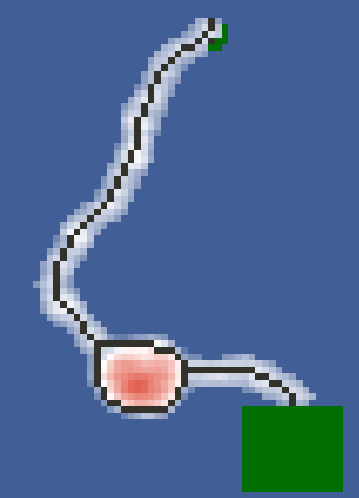
Nucleosomes are categorized by having exactly 2 endpoints and 2 branchpoints. In the example to the left, the end of the shorter arm touches the other arm a little and thus only one endpoint is existent in the skeleton of the molecule (dark grey pixels). By removing the overlapping pixels, both arms will be detected separately and the nucleosome can be analyzed properly. Removed pixels appear in green.

**Important: Be careful with the pixels you remove. If you remove pixels at the end of the arm, the length measurement will be influenced and you reduce the length of the arm.**

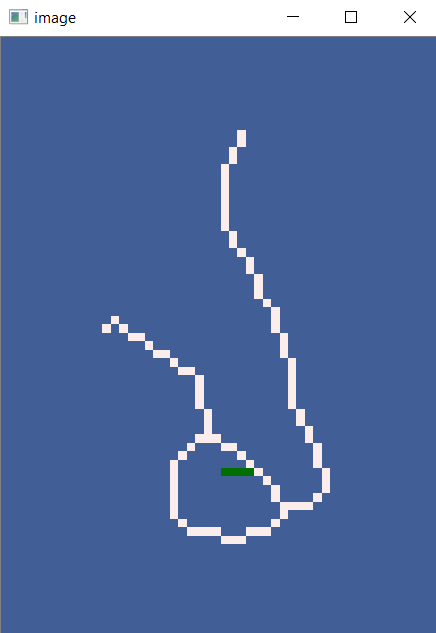
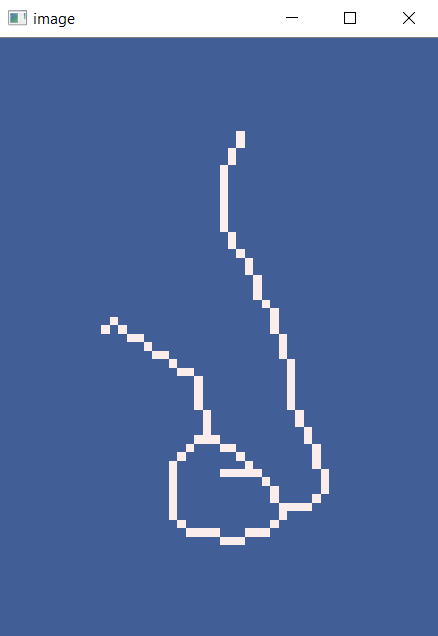
**Weird endbound nucleosomes**

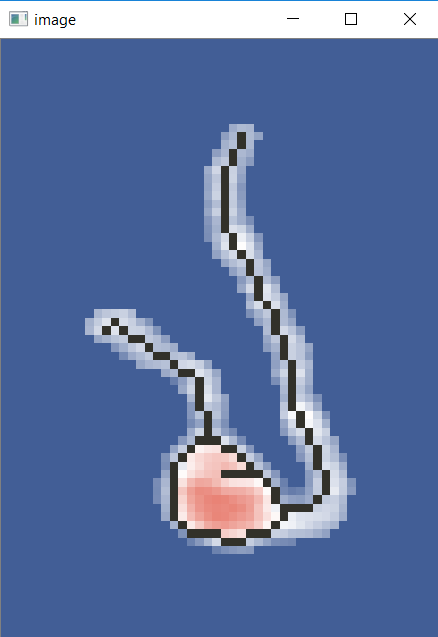
This molecule fulfills the requirements to be an endbound nucleosome (1 endpoint and 1 branchpoint) thus it will be analyzed as an endbound nucleosome if you press ESC. If you do not want to have a molecule analyzed, remove it via removing all pixels with the cursor and pressing ESC afterwards.

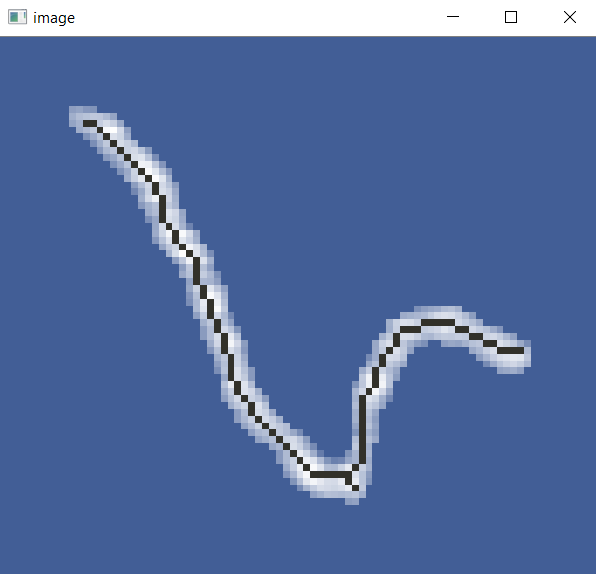
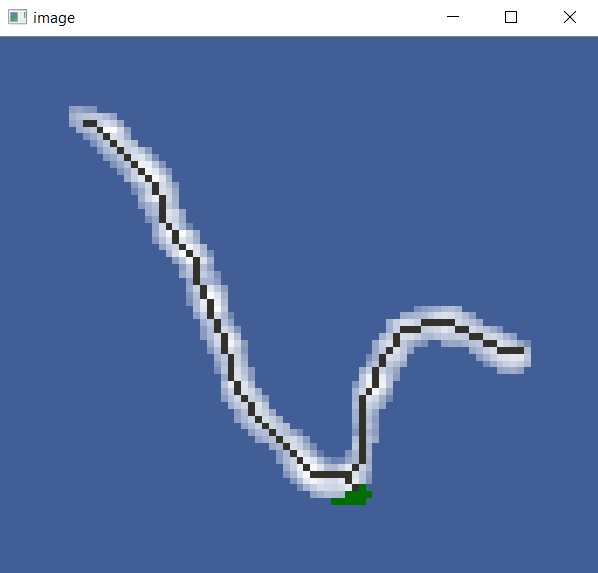
**Another nucleosome**

For the detection of this nucleosome, two things need to be done. In the bottom some small molecule fragment seems to have attached to the nucleosome. On the top end of the other arm a branchpoint appears. Removing these will make the nucleosome analyzable.

**Nucleosomes with three endpoints in the skeleton**

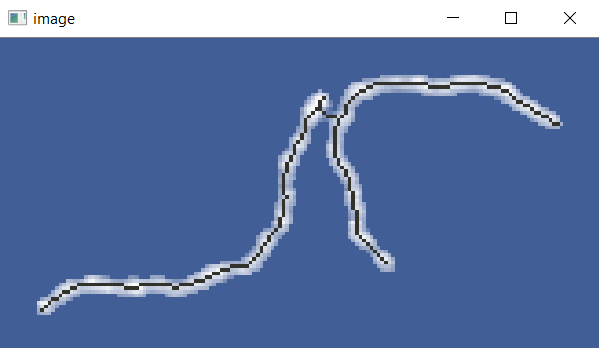
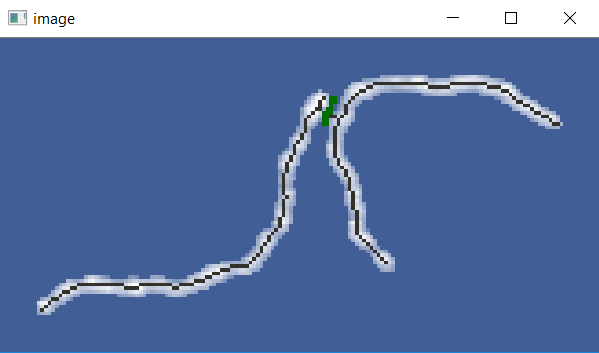
This nucleosomes is not recognized since it has 2 branchpoints but 3 instead of 2 endpoints. Make it analyzable by not doing anything and just pressing ESC. A new window showing only the skeleton will appear. Remove the third arm and press ESC. The nucleosome will now be analyzed properly. The skeleton only image will appear for all images that have exactly three endpoints and only two branchpoints. For the ones where it doesn’t make sense to remove anything, just press ESC and it will as well be discarded.



**Bare DNA**

Bare DNA is detected by having two endpoints and no branchpoints in the skeleton. In the case shown above, the skeleton has a branchpoint. By removing some pixels at this corner, the DNA will be categorized properly.

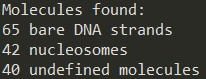
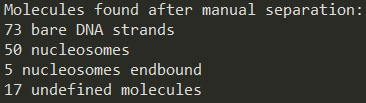
**Overlapping molecules**

If molecules have some overlapping edges, just remove the connecting pixels and the molecules are analyzed separately. This works for all molecules. So also if you have overlapping edges between mono-nucleosomes and bare DNA for example they will also be analyzed separately afterwards.

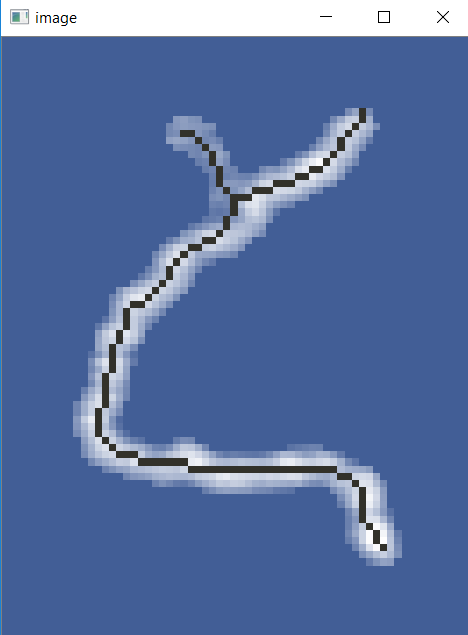
Once again, do not do this for molecules where you don’t know for sure if you wrongfully remove pixels at the end to not influence the length measurement. Similar to this, be careful when removing pixels at the nucleosome core since this might influence the ellipsoid fit for the molecule.

**Higher yield thanks to manual filtering**

Molecules in the same image with and without manual filtering



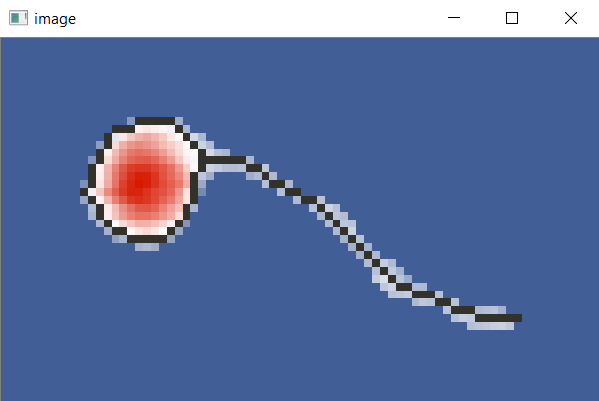
1. **More detail on the detection parameters**

**par\_back\_1/2:**

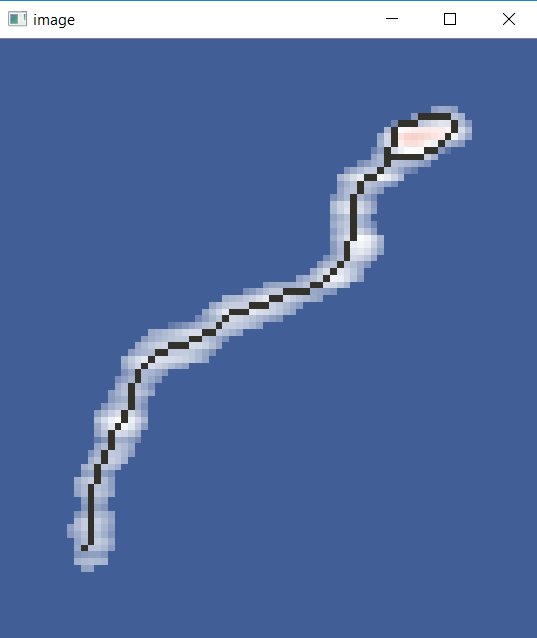
**Example for a background threshold set too low (0.20 nm)**

If the background threshold is too small, you will have a lot of molecules where additional arms appear since the background was not removed properly. (left)

**Example for a background threshold set too high (0.50 nm)**

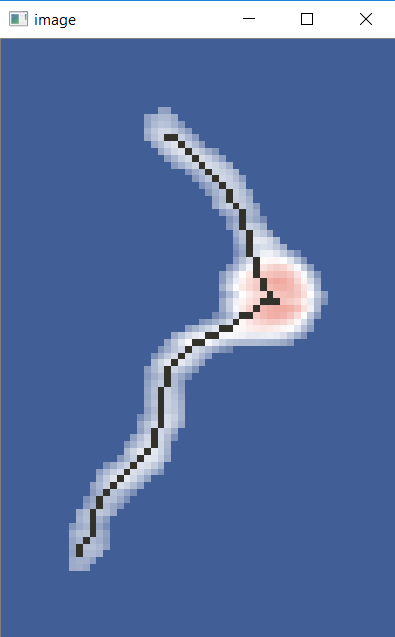
The DNA arms will be thinned out significantly which will give errors when tracing the molecule

**nuc\_min\_height:**

Together with par\_back\_1/2 the most important parameter to get a proper molecule detection. If the value is set too low, sometimes high pixels in a DNA strand will be thought of as nucleosome pixels wrongfully. It works the other way around if the value is set too high, then sometimes actual nucleosome pixels will not be counted as such.

**Example for the nuc\_min\_height chosen too small (1.00 nm)**

The end of the DNA strand is thought of as a nucleosome which will result in wrongful categorization.

**Example for nuc\_min\_height chosen too large (1.50 nm)**

If the value is too high, smaller nucleosome will not be detected properly. In general, the value should be kept constant for the same dataset but adjusted for different datasets taken at different imaging conditions and a new cantilever.

Sweet spot should be between 1.0 nm and 2.0 nm depending on the AFM machine. (sometimes also other values are good)