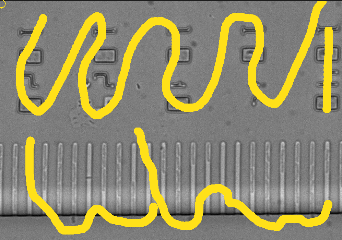
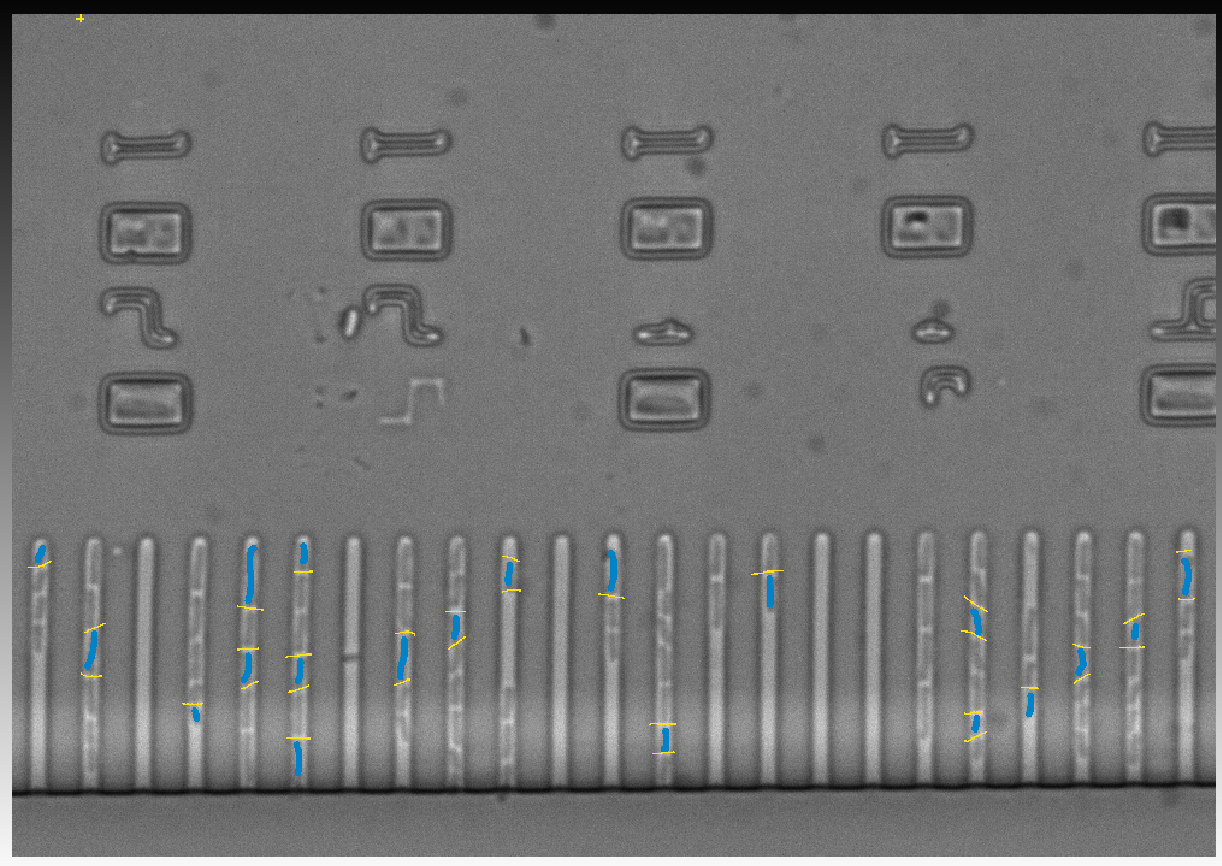
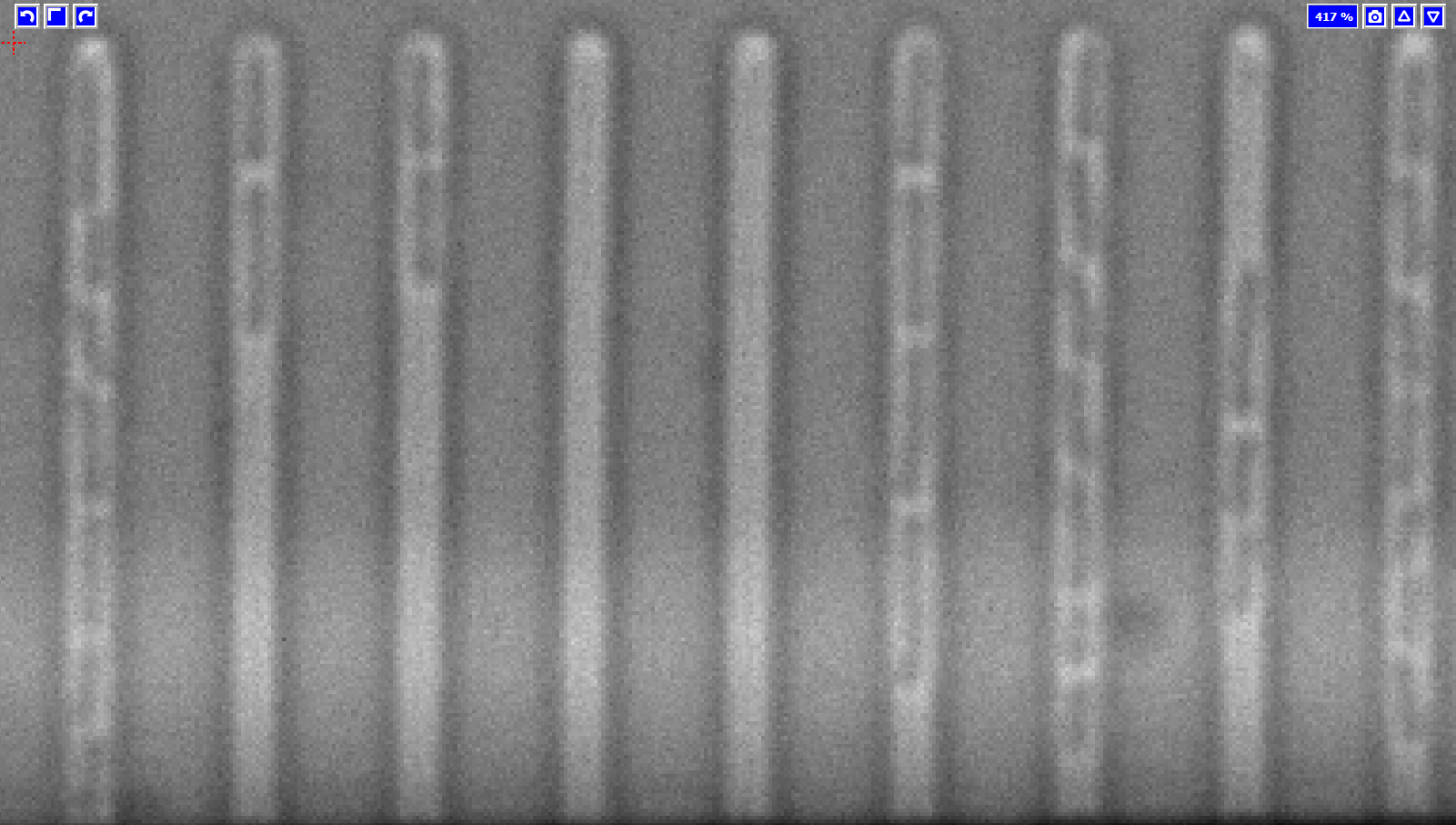
**Preparation:**

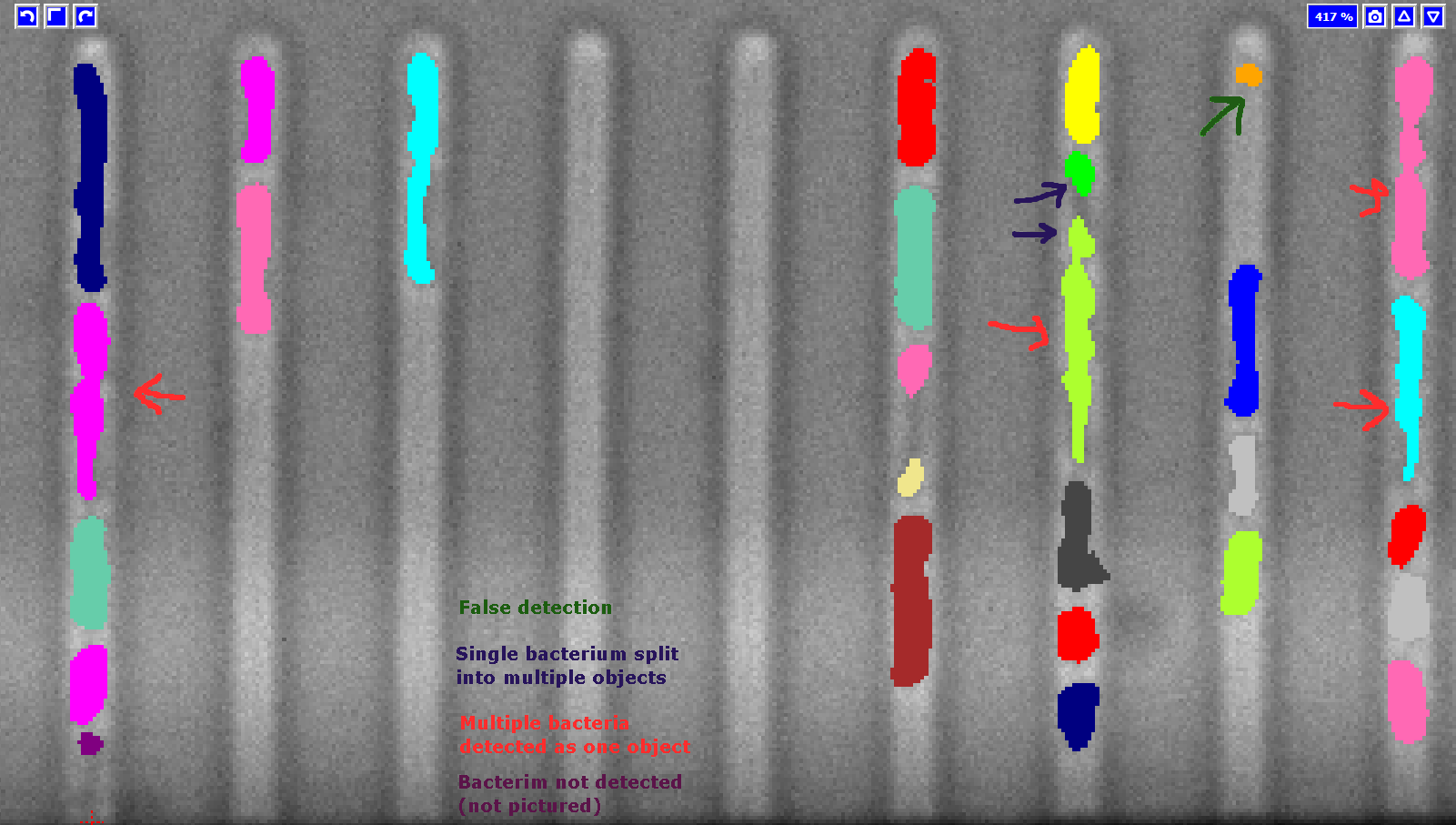
1. Delete all time lapse images that aren’t in a brightfield-fluorescence pair (eg any initial brightfield-only images taken during calibration).
2. Separate the brightfield and fluorescence images into different folders (thanks to step 1, both folders should have the same number of images). Ensure that they are named such that they show up in the same order (1st brightfield image is paired with 1st fluorescence image, 2nd with 2nd etc.) .
3. Make sure that you have downloaded ilastik <https://www.ilastik.org/> and a python interpreter of your choice.

**ilastik – Pixel classification:**

1. Start a new pixel classification project
2. Import images:
   1. “Add a single volume from a sequence”
   2. “Select files”
   3. Select every brightfield image *(ie each individual TIFF file)*
3. In “Feature Selection”, select every feature.
4. In “Training”, go to a random picture
5. If this is your first frame, liberally spread the “background” colour *(in this case, yellow)* around the picture, making to hit all of the points of interest *(eg numbers, empty columns, the central channel).*
6. In a different colour, carefully draw over some of your bacteria with the “bacteria” colour *(in this case, blue) [suggestion: size 5].*
7. Draw a thin line separating the bacterium and its neighbours with your “background” colour *(this helps the program with fine separation) [suggestion: size 1]*.
8. Repeat steps 7b/c until the desired number of bacteria have been labelled. As a suggestion, try to get at least 100 total bacteria *(suggestion = about 5 frames with 20 bacteria labelled on each on each, spread throughout the time-lapse)*. An example of this is shown below:
9. *Note not to worry about discriminating between bacteria that you want to and don’t want to measure, unless they have very different shapes/sizes. At this point we simply want to train the model to best detect any bacteria.*
10. You should end up with a couple of frames that look something like this:
11. Save the project! This is important as the next step can crash if you’ve drawn too much.
12. Press “live update” to see the program’s current segmentation model.
    1. Be patient, but if you’ve left the computer alone for a while and the program has been stuck at 80% for a long time, you may need to close and reopen ilastik and use the eraser tool to remove a bit (or remove some of the options in feature selection).
    2. Scroll through the frames and check that the prediction quality seems generally reasonable *(doesn’t need to be perfect)*. If not, draw over any large parts of the image that are wrongly labelled.
    3. If some of your images are very different from the rest for whatever reason (changed contrast, momentary obstruction of the microscope…), that image should probably be given special care. You may need to label more bacteria on that frame specifically (although this could muddy the model as a whole so be careful). Often unfortunately the best solution is to cut it out of the dataset.
13. In “Prediction Export”, “choose export image settings”. There, change the export format to “multipage tiff”, set the output file name and location *(leave it on the default “probabilities” export option)*, then return to the main ilastik screen and “Export”.

ilastik – Tracking with Learning:

1. Start a new “Tracking ” or “Tracking with learning” project (with pixel prediction map). We recommend “Tracking”, as the other option requires more manual work. However feel free to try both versions for yourself, or switch between them if the tracking results aren’t good enough. The next steps areinstructions for the “Tracking” module. *(Unfortunately, we ran out of time to quantitatively determine the difference between the two, but they both seem to work decently well.)*
2. Import images as in step 5.
3. Import the pixel classification export from the previous project (In “input data”, switch tabs to “prediction maps” and “add new separate image(s)”).
4. In “Threshold and Size Filter”, change the thresholds so that the bacteria look as well defined as possible:
5. Input: change to the colour of the bacteria’s label.
6. Smooth: changes how much artificial smoothing is applied to the bacteria [suggestion: 1; 1.5]
7. Threshold: How certain the model must be before labelling a node as part of the bacterium [suggestion: 0.5. This means that only pixels that the model is 50%+ sure belong to bacteria will be labelled as such. Higher numbers lead to smaller bacteria and sometimes causes them to split into two, but also leads to better separation between those that are almost touching. Lower will do the opposite.]
8. Size filter: The maximum and minimum size of what the model will label a bacterium [suggestion: min 30 max 2000. Only bacteria that cover more than 30 pixels or less than 2000 will be counted]
9. Adjusting these will affect the model in real time, so manually play around with these numbers to get the most accurate model (clicking the eye next to “final output” will show the underlying images).
   1. Here are some problems to look out for:



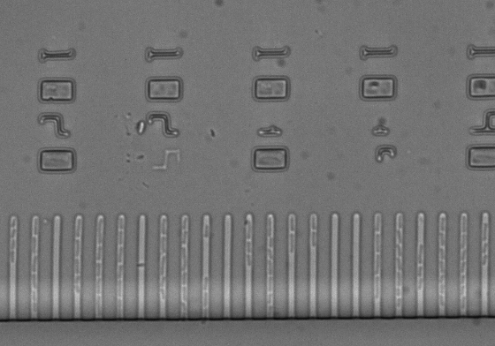
1. If your cells actively divide at any point in the experiment, even just the first frame or two, go through the process of division detection on a frame or two. *(As far as I understand, only mark bacteria as dividing if they show up as two distinct objects next frame. The purpose of this step is to help the model distinguish between bacteria moving around [in which case they need to be tracked] and bacteria that are dividing [in which case the algorithm needs to make new “child” objects from the original “parent]. This may all be wrong, but it’s my best guess as to how this all works)*
2. Use “live update” to check across other frames that the prediction seems mostly accurate. Add more manual labels to problem areas if needed to give the model additional training.
3. In “Object count classification”, go through a couple of frames and label objects that ilastik has detected. Pressing “add label” will create new colours for you to label outlines that contain two bacteria or more. *(As far as I understand, this is for telling the algorithm “this blob is good, it contains one bacterium” [labelled as 1 object] or “this blob is too big, it should be two distinct bacteria” [labelled as 2 objects] or “this blob is bad, it doesn’t correspond to a bacterium at all” [labelled as false detection] etc.)*
4. Use “live update” to check across other frames that the prediction seems mostly accurate. Add more manual labels to problem areas if needed to give the model additional training.
5. In “Tracking”, you can leave everything as default and press the big “Track!” button *(if you’ve forgotten a step, you’ll be prompted here).*
6. In “Tracking Result Export”, change the export format to “multipage tiff”, set the output file name and location, and export.

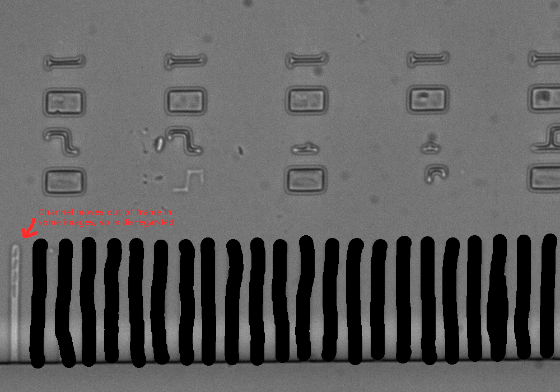
**Collating the images with ImageJ/Fiji:**

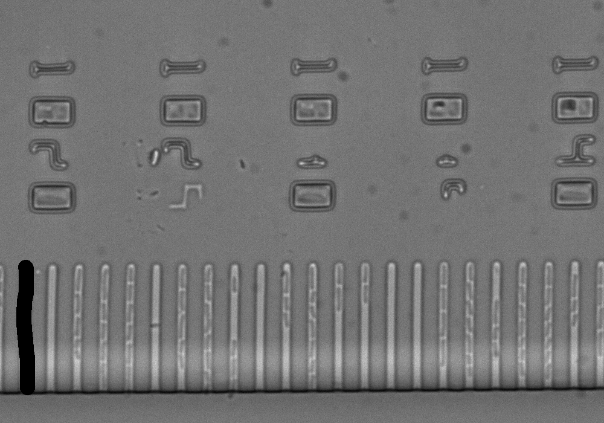
1. File → Import → Image sequence. In the popup window, select one picture in the brightfield images folder, and the program will open up a dialogue box which will import the others.
2. At this step, look through your images and make a note of which channels get cut off/fall of the side of some images, in the next step, you’ll only want to label the channels that are consistently present in the whole time lapse
3. Save as → Tiff.
4. Repeat for fluorescence images.

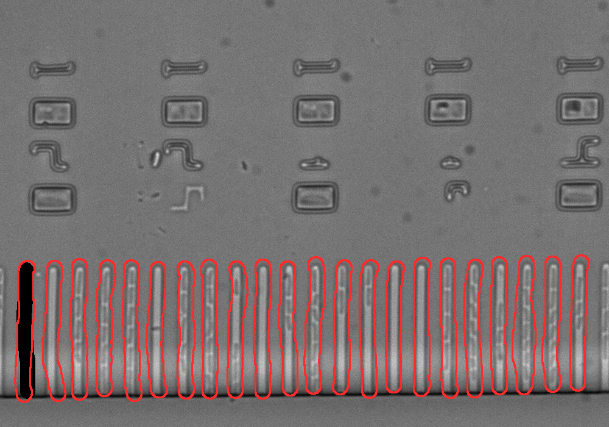
**Making the channel masks:**

*(Use an imaging software of your choice to make the masks. Here I will detail the process for using the free software Gimp:)*

1. Open the stack of brightfield images created above (Open → select image stack → import).
2. Click on the first layer (probably automatically labelled “background”) and a new layer (the defaults options are fine).
3. Repeat this step until each layer has a corresponding new empty transparent layer, making sure to name them alphabetically (Ctrl-Shit-N makes a new layer).
4. Pick a layer and start to draw over the channels *(start at the top layer (last alphabecically), and work your way down)*. Now is the time to discriminate between desired and undesired bacteria: Make sure to only label channels that are present throughout the entire experiment and ignore any that touch the edge / get cut off in a later image / are undesirable for any other reasons [but do leave in empty channels unless they are undesirable for other reasons]. The code will only analyse bacteria that are contained within the channels that are labelled here.
   1. Select the pencil tool and set the thickness to something a bit thicker than your channels.
   2. Draw quick strokes down each channel.
   3. Note that in this step, we only want a rough sketch, so don’t worry about being too precise. However, make sure the drawn channels don’t intersect, and try to cover the whole channel (small slips are fine).





1. Move onto the next transparent empty layer:
   1. Click on the eye next to the two previous layers, this will make them disappear.
   2. Use the mouse or arrow keys to select the next empty transparent layer.
      1. Either create full masks for each layer as in step 4
      2. Or only draw the leftmost channel for each frame. The code will duplicate the first frame’s channel mask onto other layers, using that one drawn-in channel as a guide (an example of this is shown here in red).
2. Once each frame’s channels have been labelled, export each of the transparent layers. I suggest using this extension to make the process easier: <https://github.com/khalim19/gimp-plugin-export-layers>
3. Import into ImageJ and save as Tiffs as done in the previous step. Make sure you tick the button to convert the images to greyscale.

**Running the code:**

1. Make sure these files are all in the same folder, and rename them as such:

* Stack of brightfield images: [something]\_Brightf
* Stack of corresponding fluorescence images: [something]\_Fluo
* Stack of manually drawn channels: [something]\_Channels
* Stack of ilastik-created masks: [something]\_ilastik *(Note the lowercase i)*

1. At this point you should be able to run the code. It’ll pull up an options window in which will allow you to change a couple of settings, as well as set the location of the folder in which the files discussed above are stored.

**Note on data storage:**

The code stores the data in terms of a matrix:

[bacterium number, type of data (e.g. brightness…), frame]

If the user requests the data to be saved, it must be converted to 2D in order to save it as a csv. This has been done by combining the data type and frame:

If each bacterium’s data was once saved like this:

[[[bac1data1 frame1, bac1data1 frame2, bac1data1frame3],

[bac1data2 frame1, bac1data2 frame2, bac1data2frame3]],

[[bac2data1 frame1, bac2data1 frame2, bac2data1frame3],

[bac2data2 frame1, bac2data2 frame2, bac2data2frame3]]]

It is now instead saved like this:

[[bac1data1frame1, bac1data2frame1, bac1data1frame2, bac1data2frame2, bac1data1frame3, bac1data2frame3],

[bac2data1frame1, bac2data2frame1, bac2data1frame2, bac2data2frame2, bac2data1frame3, bac2data2frame3]]