SOFTWARE TOOL ARTICLE

Semi-automated analysis of dot blots using ImageJ/Fiji

[version 1; peer review: 2 approved with reservations]

Anna H. Klemm ^{1,2}

¹Core Facility Bioimaging, Walter Brendel Centre of Experimental Medicine, Biomedical Center (BMC), Ludwig-Maximilians-University, Planegg-Martinsried, 82152, Germany

²Department of Information Technology, Uppsala University, Division of Visual Information and Interaction, BioImage Informatics Facility, SciLifeLab, Uppsala, Sweden

V1 First published: 01 Dec 2020, 9:1385
<https://doi.org/10.12688/f1000research.27179.1>
 Latest published: 01 Dec 2020, 9:1385
<https://doi.org/10.12688/f1000research.27179.1>

Abstract

Commercially available dot blots provide a set of specific antibodies spotted on membranes in a given pattern. If the target analyte is present in the solution that the membrane is incubated with, the detection reaction will result in a chemiluminescence signal which is recorded by film or scanner. In order to know which analytes were detected, the analysis consists of measuring the intensity of the recorded signal on each spot. Manually measuring the entire array (typically ~200 spots) is unreliable and tedious. Fully automatic registration of the blot membrane to the template pattern often fails since there might be only very few positive spots on the membrane. This article presents an ImageJ/Fiji macro that requires minimal user input to perform a robust iterative registration of an adjustable template mask representing the spot pattern to the recorded blot. Once the template mask is matched to the dot blot, the spot intensity of each dot is measured and reported in a results table.

Keywords

Dot Blot, FIJI, ImageJ macro, template registration, intensity measurements

Open Peer Review

Approval Status ? ?

	1	2
version 1		
01 Dec 2020	view	view

1. **Christopher Schmied** , Leibniz-Forschungsinstitut für Molekulare Pharmakologie im Forschungsverbund Berlin e.V. (FMP), Berlin, Germany
2. **Bram van den Broek** , The Netherlands Cancer Institute, Amsterdam, The Netherlands

Any reports and responses or comments on the article can be found at the end of the article.

This article is included in the **NEUBIAS - the Bioimage Analysts Network** gateway.

Corresponding author: Anna H. Klemm (anna.klemm@it.uu.se)

Author roles: Klemm AH: Conceptualization, Data Curation, Investigation, Methodology, Project Administration, Resources, Software, Writing – Original Draft Preparation

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by CFBim, BMC, LMU Munich, Germany and the BioImage Informatics Facility, SciLifeLab, Sweden.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2020 Klemm AH. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Klemm AH. **Semi-automated analysis of dot blots using ImageJ/Fiji [version 1; peer review: 2 approved with reservations]** F1000Research 2020, 9:1385 <https://doi.org/10.12688/f1000research.27179.1>

First published: 01 Dec 2020, 9:1385 <https://doi.org/10.12688/f1000research.27179.1>

Introduction

A dot blot is a common technique in molecular biology to query the presence of an analyte in a solution. Commercially available dot blots contain a set of so-called capture antibodies spotted to a membrane in a given pattern, with each spot representing one specific antibody. Binding of the analyte to the capture antibody is detected by a detection antibody and chemiluminescence signals are recorded by film or scanner, similar as in classical western blots. Dot blots typically contain ~200 spots to be analyzed. For automatic quantification of the dot intensities a mask reflecting the spot pattern needs to be registered to the blot. Common registration algorithms are either intensity-based, as in e.g. StackReg¹, or rely on the automatic identification of prominent features present in both images (e.g. Linear Stack Alignment with SIFT²).^{1,2} A scanned dot blot can often not be aligned to the mask with either method, especially when only a few analytes in the dot blot give positive signals. For algorithms that depend simply on feature detection, the pattern of the mask is too dense and regular to be successfully registered to few sparse dots. The intensity-based algorithm fails for similar reasons. As a result, open source tools currently available for analyzing dot blots either strongly rely on user input³ or assume full regular patterns⁴, which is not always the case. The presented ImageJ Macro DotBlot_Analysis.ijm relies on a few (3–5) landmarks selected by the user to give a first estimate of the transformation between the scanned dot blot and a template pattern. It then automatically searches for more landmarks and performs the final registration and measurements. It runs within the free and open-source software Fiji⁵.

Methods

Running DotBlot_Analysis.ijm

The macro expects two images: the scanned dot blot and a mask image. The mask image is a representation of the spot pattern. After starting the macro, the user will be asked to select the names of the mask and dot blot images, as well as enter an expected spot size in pixel and a prominence value (discussed under Implementation). The user then needs to mark 3–5 corresponding dots (=landmarks) in the scanned dot blot and in the mask. DotBlot_Analysis.ijm uses the user-set landmarks to register the mask image to the scanned blot in an iterative process. After registration the macro creates selections for each spot. The user has the possibility to manually adjust the selections by activating a selection within the ROI manager, adjusting it and pressing “update” to save the new position. After this the macro performs a background subtraction and measures the intensities of each spot. The final output of the macro is a result table that lists intensity measurements for each spot. For quality control purposes, the macro also saves the selections within the ROI Manager as a zip-file and a control-image with all spot selections overlaid over the inverted background-subtracted dot blot. The zip-file can be re-opened in Fiji by drag&drop: all selections will be listed in the ROI Manager after re-opening.

Implementation

The mask is a schematic representation of the spot pattern and needs to be registered to the scanned blot by translation, rotation and scaling (affine transformation). The macro registers the mask in two steps. In the first step it uses the 3–5 landmarks set by the user. The macro runs the plug-in “Landmark Correspondences”³ by Stephan Saalfeld, which matches the two sets of dots to determine the transformation and applies the transformation to the mask. In a second registration step, the macro automatically searches for more landmarks, in order to achieve an even finer registration. The landmarks are determined with the following computational steps:

1. Gaussian-filtering of the scanned, inverted dot blot. Signals are thereby smoothed and single signals are transformed to a smooth intensity distribution with (ideally) one single intensity maximum in the center of the spot.
2. Activation of the outlines of the single spots, as detected after the first registration step.
3. Search for local maxima on the gaussian-blurred image within each single spot outline. Maxima are determined as those pixels which are brighter than the local background, by the prominence value initially defined by the user. This ensures detection of bright signals and avoids eventual noise.
4. Once a local maximum is found, its position is marked as a new landmark. The corresponding landmark on the mask image is determined as the centroid of the selection that was used for searching the maximum.

Following this procedure, a new pair of landmarks (one set for the blot, one set for the mask) is identified and the plug-in “Landmark Correspondences” is re-run, now with more landmarks.

Once the mask has been fully registered to the scanned blot, it can be used for creating the final spot selections: the centroids of each spot of the final transformed mask are extracted and used to draw a new selection of defined size. This ensures that each spot is measured with a selection of equal size. The size of selection is provided by the user at the beginning of the script. The user has the possibility to correct the position of the final selections.

Before doing the measurements, the background of the image is subtracted using the “Subtract Background” command⁴, based on the rolling ball algorithm. The radius of the rolling ball is set to three times the user-defined spot size, to ensure that only background but not signal is removed.

After background subtraction the intensities of each spot are measured.

¹ <http://bigwww.epfl.ch/thevenaz/stackreg/>

² https://imagej.net/Linear_Stack_Alignment_with_SIFT

³ https://imagej.net/Landmark_Correspondences

⁴ https://imagejdocu.tudor.lu/gui/process/subtract_background

The final output of the macro is a result table that lists the following measurements for each spot:

- Label of the spot
- Area (for reporting, since it is the same area for all spots)
- Mean intensity ("Mean")
- Minimum intensity ("Min")
- Maximum intensity ("Max")
- X- and Y-coordinate of the spot
- Intensity density: the product of area (scaled units, if set) and mean intensity
- RawIntDen: the sum of the pixel values within the spot

See the ImageJ documentation⁵ for more detailed information about the measurements.

Operation

The macro runs within the open-source and free software Fiji (<https://fiji.sc/>), which is supported on the following systems: Windows XP, Vista, 7, 8 and 10; Mac OS X 10.8 "Mountain Lion" or later; Linux on amd64 and x86 architectures.

⁵ <https://imagej.nih.gov/ij/docs/menus/analyze.html>

Use case

Use case data

The macro and the data for testing are available for download at <https://github.com/ahklemm/AnalyzeDotBlots>. Download the material from GitHub by clicking on the green "code" button and choose "download ZIP". Unzip the material and open DotBlot_Analysis.ijm by drag&drop to Fiji. For testing purposes, open blot.tif and mask.tif. Run the macro.

Use case

Figure 1a shows as use case the scanned blot, blot.tif, and its mask, mask.tif. Four landmarks are marked in both the scanned blot and the mask (yellow arrows). Figure 1b shows the outcome after the first registration step. The blot has been inverted to create bright signals over dark background. While the spots already overlay the signals of the scanned dot blot quite well after this first registration, some selections (indicated with red arrow heads in Figure 1b) are offset with respect to the center of the actual signal. After the first registration the macro automatically identifies more landmarks, which are marked as yellow crosses in Figure 1c. The automatically identified landmarks are used for a second, finer registration step. The result of this second registration is visible in Figure 1d. Note that the center of the selections now overlays well with the center of the spots (red arrow heads, Figure 1d). Figure 1e shows the final result: the overlay of the final measurement selection over the background-corrected blot and the final results table.

Figure 1. Key steps of DotBlot_Analysis.ijm. a) User selected landmarks in the scanned dot blot and the mask. b) Overlay of the spots over the dot blot after the first registration step. Red arrowheads mark spots with major shifts in respect to the signal center. c) Yellow crosses show the automatically detected landmarks in the scanned blot image as well as on the mask. d) Overlay of the spots after the second registration step. The marked spots (red arrowheads) now match the signal center better. e) Final results of the workflow.

Adaptations needed for different dot patterns

The use case shows the scanned blot and mask of a very specific dot blot pattern/array. Unless the user wants to evaluate blots of exactly that array one needs to provide a new labelled mask. Within the labelled mask the spot on the upper left has an ID/gray-value = 1 and IDs/gray values of spots increase from the left to the right and from the top to the bottom. In mask.tif there are 218 spots, so the spot on the lower right has the gray value of 218. The ID/grayvalue is important in order to identify the spot even after the rotation/scaling/translation of the two affine registration steps.

MaskGenerator.ijm in the downloadable material helps to transform a typical vendor template⁶ into a labelled mask. Running MaskGenerator.ijm the user draws one selection around a spot of the template. The macro then identifies bright spots of the expected size and by roundness.

Summary

The presented Fiji macro DotBlot_Analysis.ijm supports users to analyze images of dot blots. It requires minimal input of few landmarks to then automatically measure the intensities of single spots in dot blots.

Limitations of the analysis

While scanning the dot blot one should avoid saturation of

the signal. Signals should not reach a value of 0 but should always contain a minimum value of intensity. Reduce exposure times when imaging the film if needed.

Support for problems when using the script is provided via [forum.image.sc](#) – please link to the username of the author: @aklemm

Data availability

Source code is available from: <https://github.com/ahklemm/AnalyzeDotBlots>

Archived source code as at time of publication: <https://doi.org/10.5281/zenodo.4134219>⁶

License: BSD-3-Clause

Acknowledgements

The image of the dot blot was provided by Katja Eubler, Mayerhofer-lab, BMC, LMU Munich, Germany.

The author thanks Giovanni Cardone, MPI of Biochemistry, Munich for comments on the manuscript and the macro script and Andreas Thomae, CFBim, BMC, LMU Munich for discussions about the imaging of dot blots and analysis.

This publication was supported by COST Action NEUBIAS (CA15124), funded by COST (European Cooperation in Science and Technology).

⁶ E.g. https://www.rndsystems.com/products/proteome-profiler-human-xl-cytokine-array-kit_ary022b

References

1. Thévenaz P, Ruttimann UE, Unser M: **A pyramid approach to subpixel registration based on intensity.** *IEEE Trans Image Process.* 1998; 7(1): 27–41. [PubMed Abstract](#) | [Publisher Full Text](#)
2. Lowe DG: **Distinctive Image Features from Scale-Invariant Keypoints.** *Int J Comput Vis.* 2004; 60(2): 91–110. [Publisher Full Text](#)
3. Ohgane K: **Quantification of Gel Bands by an Image J Macro, Band/Peak Quantification Tool.** 2019. [Publisher Full Text](#)
4. **Protein Array Analyzer for ImageJ - Gilles Carpentier Research Web Site: Computer Image Analysis.** (accessed Oct. 01, 2020). [Reference Source](#)
5. Schindelin J, Arganda-Carreras I, Frise Erwin, et al.: **Fiji: an open-source platform for biological-image analysis.** *Nat Methods.* 2012; 9(7): 676–682. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
6. ahklemm: **ahklemm/AnalyzeDotBlots: 1. release AnalyzeDotBlots (Version v1.0.0).** *Zenodo.* 2020. <http://www.doi.org/10.5281/zenodo.4134219>

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 22 December 2020

<https://doi.org/10.5256/f1000research.30023.r75773>

© 2020 van den Broek B. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bram van den Broek

BioImaging Facility & Dept. of Cell Biology, The Netherlands Cancer Institute, Amsterdam, The Netherlands

Summary

The author describes a tool to quantify dot blots or antibody arrays. These are membranes with antibodies absorbed to it in a certain pattern, used for (quantitative) detection of specific compounds in a solution.

The presented ImageJ1 macro utilizes a well-established registration tool to map a template spot pattern (mask) to the scanned image of the dot blot, in a clever iterative way. The user needs only to provide a few matching point selections (corresponding landmarks) on both images. The macro then automatically optimizes the alignment process by finding more landmarks, leading to a good match for every spot. When the user is satisfied with the registration, intensities and positions of the dots are reported, as well as an overlay image for inspection.

The paper starts off by describing why other open source registration tools and dot blot analysis tools fail in some or most cases. Fully automatic registration is a utopia, because of the sparsity of positive spots in the dot blot, compared to the periodic spot alignment in the template. The author also makes a valid point by stating that 'other tools require or assume full regular patterns, which is not always the case'. I cannot judge how often this actually happens, but the provided example blot is clearly not fully regular.

The paper is well written, and the workflow steps are clearly described. Additionally, the author provides ample information on how to get the software to run on your computer, and where to go for support.

I found the software tool easy to use, and rather robust. The clever procedure of iterative registration makes that the spots are correctly detected, even if the dot blot is somewhat distorted. The macro provides an output that is not overly complicated. Compared to another tool that is briefly mentioned, the Protein Array Analyzer by Gilles Carpentier, the functionality is in fact rather limited, but if you just need to quantify the intensities of the spots in a blot it will serve you well.

Things that could be improved

At the very end, the paper also briefly describes a second macro, on how to create a suitable template mask for a different dot blot pattern. For any user of the script, this part is of utmost importance, and I believe it needs a bit more attention.

I tried to reproduce the same template mask by downloading the vendor mask in reference 6, but could only find it as a picture embedded in the datasheet PDF. The template image consists of black open circles on a white background. This is important information missing in the manuscript and/or the MaskGenerator macro, because the macro only functions with such a template image. Still, I could only get it to create a functional mask (from a screen capture of the template) after looking into the code. (See also remarks below).

The panels in figure1 are rather small, and the resolution is not great. The yellow crosses in figure 1c, for instance, can hardly be discerned. (Here, using a color more contrasting with gray may also help.) The same holds for Figure 1d. The three registered masks on the bottom all look the same; the landmarks are almost invisible. A larger image, perhaps with the first and second iteration of the registration mask overlaid in different colors may clarify things.

Footnotes and references use the same numbering and typesetting. This is quite confusing and should be changed.

I have many other, more technical remarks and suggestions that are mostly meant as improvements for the macro:

- I wonder why the background is subtracted after the registration process. This could in theory lead to the detection of false maxima (landmarks). Perhaps I'm missing something, but I do not see a disadvantage of first subtracting the background.
- When the dot blot image has light spots on a dark background, the macro does not work (no landmarks found). I recommend to include a choice (drop-down menu) in the script parameters, and then invert the image only if required.
- Although described in the text, 'prominence' is not very intuitive for users, nor is there feedback on whether the chosen prominence has been a good choice. It would be helpful to provide information on the number of landmarks found in e.g. the log window, and also the possibility to pause the macro at this step (again with a checkbox in the parameters dialog) to inspect the secondary landmarks in case the registration fails.
- The Gaussian blur in line 75 should be a certain fraction of the spot diameter.
- Things go wrong if there are more than 255 spots in the mask and/or blot image. (Transformation2_Thres does not look as it should for spot number 256 and up, leading to the error message 'incorrect number of ROIs detected'.)
It can be fixed by allowing 16-bit spot template images, and changing the upper thresholds in line 65 and 146, from 255 to 65535.
- The message 'Incorrect number of ROIs detected, please check the ROI manager.' isn't very helpful. Please either explain what the user is supposed to do, if action is required.
- Show the labels of the ROIs in the manual modify step. This allows dragging the ROIs

without the need to update the ROI in the ROI manager. Also, more information on how to do this will be helpful to the user.

- Nice-to-have: a possibility to change the size of the spots in the manual modification step (e.g. with a (non-blocking) dialog instead of a `waitForUser` statement.)
- Show ROI labels in the flattened RGB overlay output image as well, so it is easy to check if the measured values make any sense. (Ideally, one would overlay the position in [column, row] format, but I realize that this would require a lot of work, and may be error-prone.)
- File handling and output, some suggestions:
 - Let the user choose a different output folder (or a subfolder) than the one containing the data.
 - Do not close/modify the original images, i.e. duplicate before processing.
 - Overlay the ROIs also on the original dot blot image, not only the processed inverted RGB.
- Some error catching:
 - A message when there are less than 2 images open.
 - A message/dialog when the prominence is set too high (now this leads to 'not enough landmarks selected to find a transformation model'), perhaps even with the possibility to change it without rerunning the macro.
 - The macro crashes if the image name does not contain a dot (e.g. if it is processed/duplicated with another name prior to running the macro).
- The `ID_Array()` function is overly complicated. Use `IJ.pad(n, length)` to introduce leading zeros. The whole function can be replaced by

```
ID_Array = newArray(nSpots);
for(i = 0; i < nSpots; i++) {
    ID_Array[i] = IJ.pad(i, 3);
}
```

Comments on `MaskGenerator.ijm`:

- The message 'Draw a circle around one of the Dots' is somewhat misleading, because it seems to imply that the user should draw a bit larger than a single spot, in which case the macro often fails to recognize any round structures. In other words: $0.8 \times \text{DotArea}$ is too strict.
- Very minor: Pre-select the circle selection tool before asking to draw a circle.
- When looking for test dot blots on the web, I found that many are in a fully regular pattern. In this case a mask could easily be generated without a vendor template. (It does not matter if in the real DotBlot Array some spots are missing, as long as the pattern is regular.) Below is the code I wrote for generating masks while testing several dot blots. It could be integrated in the macro:

```
#@ Integer (label="Number of columns" , value=12) cols
#@ Integer (label="Number of rows" , value=8) rows
```



```
#@ Integer (label="Spot size" , value=20) size

dist = 8;

newImage("Mask_temp", "16-bit black", cols*(size+dist) + dist, rows*(size+dist) + dist, 1);
setForegroundColor(255,255,255);
for (r = 0; r < rows; r++) {
    for (c = 0; c < cols; c++) {
        makeOval(dist + c*(size+dist), dist + r*(size+dist), size, size);
        fill();
    }
}
run("Select None");
setAutoThreshold("Default dark");
run("Convert to Mask");
run("Analyze Particles...", " show=[Count Masks]");
run("glasbey_on_dark");
rename("Mask");
close("Mask_temp");
```

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: BioImage Analysis, Fluorescence Microscopy, Cell Biophysics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 18 December 2020

<https://doi.org/10.5256/f1000research.30023.r75775>

© 2020 Schmied C. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Christopher Schmied

Leibniz-Forschungsinstitut für Molekulare Pharmakologie im Forschungsverbund Berlin e.V. (FMP), Berlin, Germany

The Author introduces a workflow for the semi-automated analysis of dot blots using Fiji macros. To measure the intensity of the individual dots, a template is registered to the input blot using a two stage landmark based registration. First a registration is applied using manual landmarks specified by the user. This achieves a coarse alignment. The second registration step uses automatic landmarks, generated via local maxima detection. Affine transformation is used to match the template with the dot blot. Based on the template mask, the macro then extracts a number of useful measurements and presents a result table. The results as well as files for quality control are saved into the folder that contain the input files.

I tested the workflow with the provided material. The download from the github repository provides the necessary scripts as well as test images for the blot and the mask. I was able to reproduce the main analysis on the first try. The controls of the macro are easy and intuitive, thus match with the described minimal user input. I like that the blot.tif test image provides already selections for the initial user input, so that I see how the author intended the macro to be used. I like that the necessary files for quality control are saved and easily accessible. I usually prefer to give the user an option to specify a different output directory, but this does not interfere with the usage of this macro. Two very small but important details of the article I like to highlight: the download of the material from github is well described, this is often a problem for people that never used github. I liked that the author referred to forum.image.sc and their user name there, as a way to find support for potential users.

The method is introduced and motivated well, although I have the feeling that references in the introduction are missing, when describing the method behind dot blots. The workflow logic is in general well described in its details and detailed components of the workflow are described sufficiently and also clearly. Unfortunately the structure of the article does not lend itself for an easy and quick understanding. Also details of how the input data needs to look like is not described clearly. Here are my points for revision:

Major: The article describes in detail how the macro works. However, I review this now also with the perspective of a user who wants to analyze my own blot and wants to get to the important bits fast. Here the article structure is confusing and not conducive for easy and quick adoption. Let me explain:

Methods:

1. **Running DotBlot_Analysis.ijm:** The author first introduces the general method and usage

of the workflow, no mention of where to find the script or an illustration of the workflow.

2. **Implementation:** The author explains the detailed methods and the individual components in the section Implementation, this is good. Still it lacks the way to action on this or references to an illustration to easily understand the workflow.

3. **Operation:** explains the macro uses Fiji and is OS independent.

Use case:

1. **Use case data:** introduces us finally to the location of the actual script and test files, something I would have liked to have at the beginning.

2. **Use Case:** shows us a good schematic of the workflow and really illustrates well how to operate the workflow. Something that could have helped to understand the "Methods" and also the "Implementation".

3. **Adaptions needed for different dot patterns:** we learn how to customize this workflow to a different dot blot. The author even provides a script for that which is great by the way. I just thought that a user, that presumably wants to analyze a different dot blot, would have needed that information earlier. I think this is also an important part of the workflow.

Summary: is fine in the end.

Limitations of the analysis: is an afterthought, although this is actually an important prerequisite.

I think this can be amended by changing the structure without changing the content a lot. Here are my suggestions for that:

Methods

1. **Prerequisite:** content of "Limitations of the analysis" but more (see other Major point)

2. **Download & Installation:** Discuss download of script and example data ("Use case data")

3. **Operation**

4. **Custom Mask:** "Adaptions needed for different dot patterns"

5. **Running DotBlot_Analysis.ijm** fused with **Use case**

Implementation

Could stay the same but could use the Figure better.

Use case

Maybe could describe how this particular example blot has been generated and scanned. Could then discuss the results of this example blot and how to interpret the results of the macro based on this blot briefly.

Summary

Limitations: could describe potential conditions where the macro fails to register properly.

Major: The Limitations of the analysis are very lightly touched upon and only at the end. Although they are, in the case of the discussed information, an important prerequisite for the analysis. I find them to be too brief and not clearly described for a user to understand immediately. Please explain clearer how a good blot should look like or describe briefly how one scans a good blot and/or refer to an article that does this in more detail.

Major: Figure 1b&d: Unfortunately, the details in the examples images shown are very small also in the online version of the article. Figure 1b and Figure 1d are crucial in illustrating the results of the registration and why there needs to be two distinct registration steps in the first place. However, the overlay with the yellow ROIs are very hard to see and the images are too small. One can enlarge parts of the image as insets and show specifically the effect of the registration. One can also enlarge the yellow ROI outline or change the color of the ROI to make the ROIs more visible.

Major: Figure 1c: from the description there should be automatically detected landmarks. But these are not visible in the image. Again cropping and enlarging can help.

Major: Using the same overlapping numbers for citations and footnotes does not work! Super confusing.

Minor: The Figure illustrates the workflow and the results of the individual processing steps for the DotBlotAnalysis.ijm. To reflect the entire complete workflow it would be good to set apart the input original blot and the generation of the original mask also visually.

Minor: The content of the upper row of images is very hard to see when printed.

Minor: it is great that the MaskGenerator.ijm is also included. Since I see this as a crucial part of the workflow. It would thus be great if an example input file for testing would also be provided for this script.

Minor: All the material is nicely provided on the github repository. Please also include a description of the usage in the readme there.

Minor: I think it is not obvious that the mask initially also has a different size than the blot and the adjustments are a result of the transformations applied during the registration. I think this needs to be noted in the usage or the figure legend.

Minor: the DotBlotAnalysis.ijm on the github repo is referred to as DotBlot_Analysis.ijm in the text.

Minor: I found this macro: https://www.optinav.info/MicroArray_Profile.htm a similar approach to Gilles Carpentier Protein Analyzer. Could be included in the references to regular patterned ROIs.

Minor: no citations for the methods used to generate and scan dot blots are provided.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioimage analysis, computer vision, data science

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research