

# Streamlining Lateral Flow Assay (LFA) Data Analysis: A MATLAB<sup>®</sup> Live Script for Integrated Batch Image Processing and Limit of Detection Determination

-Karan Saxena, Bhushan Toley

## AUTHOR ADDRESS

### Corresponding Author

Bhushan J. Toley – Department of Chemical Engineering, Indian Institute of Science, Bangalore 560012, India; [orcid.org/0000-0003-0119-2350](https://orcid.org/0000-0003-0119-2350); Email: [bhushan@iisc.ac.in](mailto:bhushan@iisc.ac.in)

### Author

Karan Saxena – Department of Chemical Engineering, Indian Institute of Science, Bangalore 560012, India; <https://orcid.org/0000-0002-0168-7444>

## Abstract

Lateral flow assays (LFAs) are widely utilized for rapid point-of-care diagnostics in various fields; however, their quantitative image analysis is frequently hindered by repetitive manual processes and inconsistencies, typically involving the use of combinations of image processing and data analysis software tools. This protocol introduces a MATLAB live-script-based code for the integrated batch processing of LFA images and standard curve development. This code simplifies quantitative LFA analysis by automating image processing, peak detection, statistical analysis, and data fitting to a 4-parameter logistic curve for accurate Limit of Detection (LoD) determination. This approach significantly minimizes manual intervention and the potential for human error, thereby enhancing reproducibility. This addresses the critical need for efficient and reliable LFA data analysis in research laboratories. By offering a standardized method for LFA image processing and data analysis, this protocol facilitates faster and more consistent research outcomes.

## Keywords

Lateral Flow Assays (LFAs); MATLAB Image Analysis; Image Analysis Automation; Point-of-care diagnostics; paper microfluidics; 4 parameter logistic (4-PL) curve

## Introduction

Lateral Flow Assays (LFAs) have emerged as valuable paper-microfluidic tools for point-of-care (POC) diagnostics, offering a quick, economical, and user-friendly method to detect the presence of analytes in complex samples<sup>1-8</sup>. Primarily known for their application in medical diagnostics, environmental monitoring, and food safety,

LFAs have gained significant attention for their potential in point-of-care testing, especially in resource-limited settings<sup>2,4,7,9,10</sup>. Accurate and reproducible quantification and interpretation of the assay results is crucial for reliable diagnosis. However, visual qualitative interpretation of LFAs, which is subjective, can lead to variability in the interpretation of results. This underscores the critical need for standardising and automating image analysis techniques that can ensure the precise and consistent quantification of LFA results, thus enhancing the reliability of the assay.

Various Image processing algorithms for the analysis of lateral flow assay images have been developed and reported in the literature<sup>11-21</sup>, most of which pertain to the development of smartphone applications for the accurate interpretation of the test results by the end-user. However, while the LFAs are being designed in the laboratory, the researchers need to 'run the standards,' i.e., perform the assay at several different analyte concentrations, in many replicates for each concentration to generate a standard calibration (signal-response) curve for the proposed assay. Only once the standard curve has been plotted can the limit of detection (LoD) of the assay be determined. This workflow typically involves the analysis of individual LFA images, one at a time, using standard image processing software like ImageJ<sup>22</sup>. The image data of individual LFA images retrieved from ImageJ is then transferred to and analysed in spreadsheets like MS Excel or statistical analysis packages like Origin, R, SPSS, etc. This typically requires a significant number of manual steps to be performed by the researcher separately for every image in the batch<sup>23</sup>, making the process inefficient and prone to manual errors. Therefore, it is necessary to integrate and automate these image processing and statistical data analysis steps into a single code to improve the reproducibility and accuracy of the analysis process.

In this article, we present a MATLAB<sup>®</sup> live script (.mlx) code to integrate and automate all the above-mentioned steps for the batch image analysis of sandwich LFAs into a single code. The user simply needs to enter the link to the directory containing the batch of LFA images to be analysed as the input, and the code automatically generates the standard curve for the assay, along with the limit of detection (LoD) as the final output.

## Protocol Overview

This manuscript introduces a comprehensive MATLAB-based protocol designed to automate batch processing of Lateral Flow Assay (LFA) images in research laboratories. The protocol aims to streamline the analysis of LFA strips, transforming qualitative visual data into quantitative information that is essential for the development and optimization of LFA-based diagnostics. Here, we provide an overview of the protocol's functionality, methodology, and the types of output it generates.

## Features

The protocol facilitates:

1. **Automated Batch Processing:** Enables the processing of multiple LFA image files in a single run, efficiently handling large datasets typical in LFA-related laboratory research.
2. **Standard Curve Development:** Automatically generates standard curves based on the Test Line Intensity to Control Line Intensity (T/C) ratios calculated from the assay images, crucial for assessing assay sensitivity.
3. **Limit of Detection (LoD) Calculation:** Employs a statistical approach to determine the LoD from the standard curve, providing a critical metric for assay performance.

## Methodology

The code integrates and automates a sequence of image processing and analytical steps:

1. **Image Pre-processing:** Images are loaded, and, if necessary, cropped to focus on the relevant assay area. This step ensures consistency in analysis across different batches.
2. **RGB Channel Segregation and Selection:** Each image is split into its RGB components, and the channel that provides the best contrast for detecting test and control lines is selected. This step is essential for enhancing the accuracy of subsequent analysis.
3. **Peak Detection and Quantification:** This code utilizes peak detection algorithms to identify and quantify the intensity of test and control lines. The protocol adjusts for baseline noise, ensuring that the intensities of the detected peaks corresponding to these lines are accurately quantified.
4. **Data Analysis and Standard Curve Fitting:** Aggregates the quantified data from all images, calculates the T/C ratios, and fits these data points to a 4-parameter logistic (4-PL) curve to create a standard curve.
5. **Statistical Processing:** Computes the mean, standard deviation, and coefficient of variation for T/C ratios at each concentration for statistical analysis of the assay performance.
6. **Limit of Detection Calculations:** Once a 4-PL curve has been fit to the data points, the limit of blank (LoB) and limit of detection (LoD) of the assay is calculated using the method described by *Ambruster and Pry*<sup>24</sup>

$$LoD = \mu_{Blank} + 1.645\sigma_{Blank} + 1.645\sigma_{Low\ Concentration\ Sample}$$

## Output

The protocol generates the following key outputs:

1. **Quantitative Metrics:**
  - a. For individual images, the code provides values for the test and control line intensities, along with the calculated T/C ratio.
  - b. The mean, standard deviation, and coefficient of variation of T/C ratios is reported for each tested concentration.

2. **Standard Curve:** A standard (or calibration or signal-response) curve is automatically generated, visualizing the relationship between analyte concentration and T/C ratio. The following curve-fitting data is displayed:
  - a. **Fitting parameters** for the four-parameter logistic (4-PL) curve with 95% confidence interval bounds
  - b. **Goodness-of-fit (gof) statistics:**
    - i. Sum of squares due to error: **sse**
    - ii. R-squared (coefficient of determination): **rsquare**
    - iii. Degrees of freedom in the error: **dfc**
    - iv. Degree-of-freedom adjusted coefficient of determination: **adjrsquare**.
    - v. Root mean squared error (standard error): **rmse**.
3. **LoD Determination:** The LoD is calculated based on the standard curve, providing a quantitative measure of the assay's sensitivity.
4. **Statistical Summary:** A comprehensive summary of statistical metrics for assay performance across different concentrations is produced.

## Requirements and Setup

### Prerequisites:

- **Software Requirements:** MATLAB R2018a or later (should support live scripts (.mlx) format)
- **Toolboxes:** MATLAB® Image Processing Toolbox, MATLAB® Curve Fitting Toolbox, MATLAB® Signal Processing Toolbox

### Setup Instructions:

1. Install MATLAB and required toolboxes using the MATLAB installer.
2. Ensure images are placed in a single, accessible directory.
3. Open MATLAB and set your current folder to the directory containing the code.

## Input Data Preparation

### Image Files:

- Format images as **.jpg** files. The code can also be modified to accept other image formats as input.
- Ensure images are clear, with uniform lighting and without obstructions.
- Crop images to include only the relevant assay area if necessary.
- The image should be oriented to have control line at the top and test line at the bottom of the strip (Figure 1).

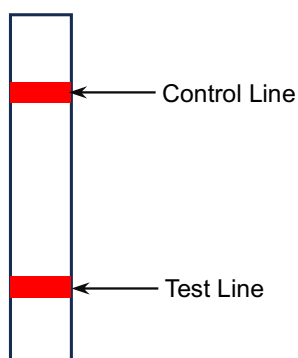


Figure 1 Correct orientation of input LFA Images

## Naming Convention:

- Name files in the format: **concentration\_ReplicateNumber\_Label.jpg** (e.g., **0.5\_II\_AuNP.jpg** represents an LFA image of the second replicate at 0.5 ng/mL concentration of the analyte employing AuNP as label).
- Ensure consistency in naming for automated processing.

## Procedure:

The users need to follow the following three steps:

1. **Load Images:** Enter the link to the directory containing the image files to be analysed. The code loads all **.jpg** images from the specified directory.
2. **Specify relevant parameters:** If required, change the values of the relevant peak detection, etc parameters from the default values. These parameters are discussed in detail in the next section.
3. **Run the live script.**

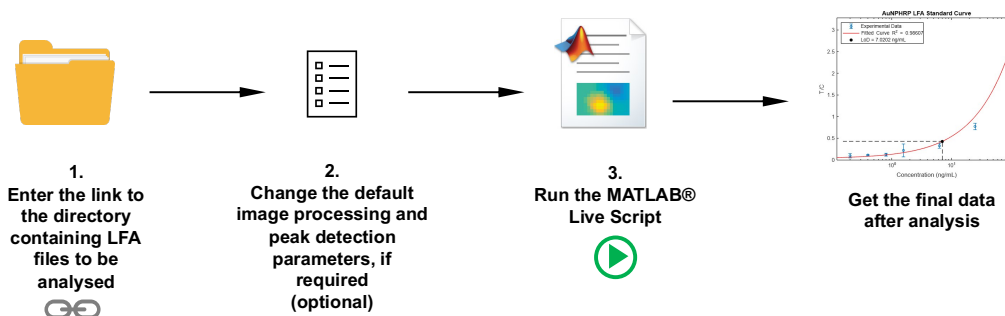


Figure 2 User Workflow for Using the Live Script

## Parameters and Customization

- **min\_peak\_prominence:** This refers to the minimum threshold height of the peaks above the baseline for the peak to be detected. Adjust this parameter to improve peak detection accuracy and sensitivity. A higher threshold will result in smaller peaks (faint lines) remaining undetected, while a lower threshold will

result in random fluctuations in signal being falsely labelled as peaks. The default value is 1.5.

- **min\_peak\_distance:** This refers to the minimum distance (in pixels) between the test and control lines. The default value is 200 pixels. This value can be modified as per requirements of the assay to ensure peak detection accuracy. Note that this value should be lower than the actual distance between the lines in the images in order to detect both the lines.
- **smooth\_the\_data:** Toggle to apply data smoothing using a moving average filter. Adjust the smoothing factor as required. This step is optional and may increase the computational load. This option should be selected to improve the peak detection accuracy if the data is very noisy. The default value is 'false.'
- **Image Cropping Parameters:** Modify based on the assay strip's position and dimensions.
- **Channel Selection:** The code evaluates the RGB channels of the first image to automatically select the one providing the best contrast. This same selected channel is used for the analysis of all the other images in the same batch. The user can also opt for selection of the 'best\_channel' based on the area under the curve of the three channels, to select the darkest among the R, G and B channels. Alternatively, the user can also manually select the channel for image analysis if required. For this, un-comment the relevant sections of the code under the heading "Determining the Best Channel for Analysis."

Customization for different types of LFA may involve adjusting the above parameters or the peak detection settings to accommodate variations in line intensity or spacing. However, it should be ensured that all the parameters specified remain the same for all the batches of images which have to be compared with each other.

## Troubleshooting Issues:

- **Suboptimal Curve Fitting:** Review data for outliers. Other sigmoidal curves can also be explored from MATLAB<sup>®</sup> Curve Fitting Toolbox.
- **Inconsistent Peak Detection:** Review the peak detection parameters: **min\_peak\_prominence** and **min\_peak\_distance** and image quality. Ensure consistent assay presentation across images.

## Conclusion

This MATLAB-based protocol is a useful tool for researchers engaged in LFA development. By automating and integrating the time-consuming process of LFA image analysis in the laboratory, it enables more efficient, accurate, and reproducible quantification of assay results. The protocol supports a wide range of LFA applications,

from disease diagnostics to environmental monitoring, making it a versatile asset for research laboratories focused on rapid assay development and optimization. This code is compatible with LFAs employing a variety of optical labels (generating lines of different colours) because of its ability to automatically select the best channel (the one with the highest contrast) among red, green and blue channels for image analysis.

Figure 3 outlines the **complete image processing and data analysis workflow** followed in the live script:

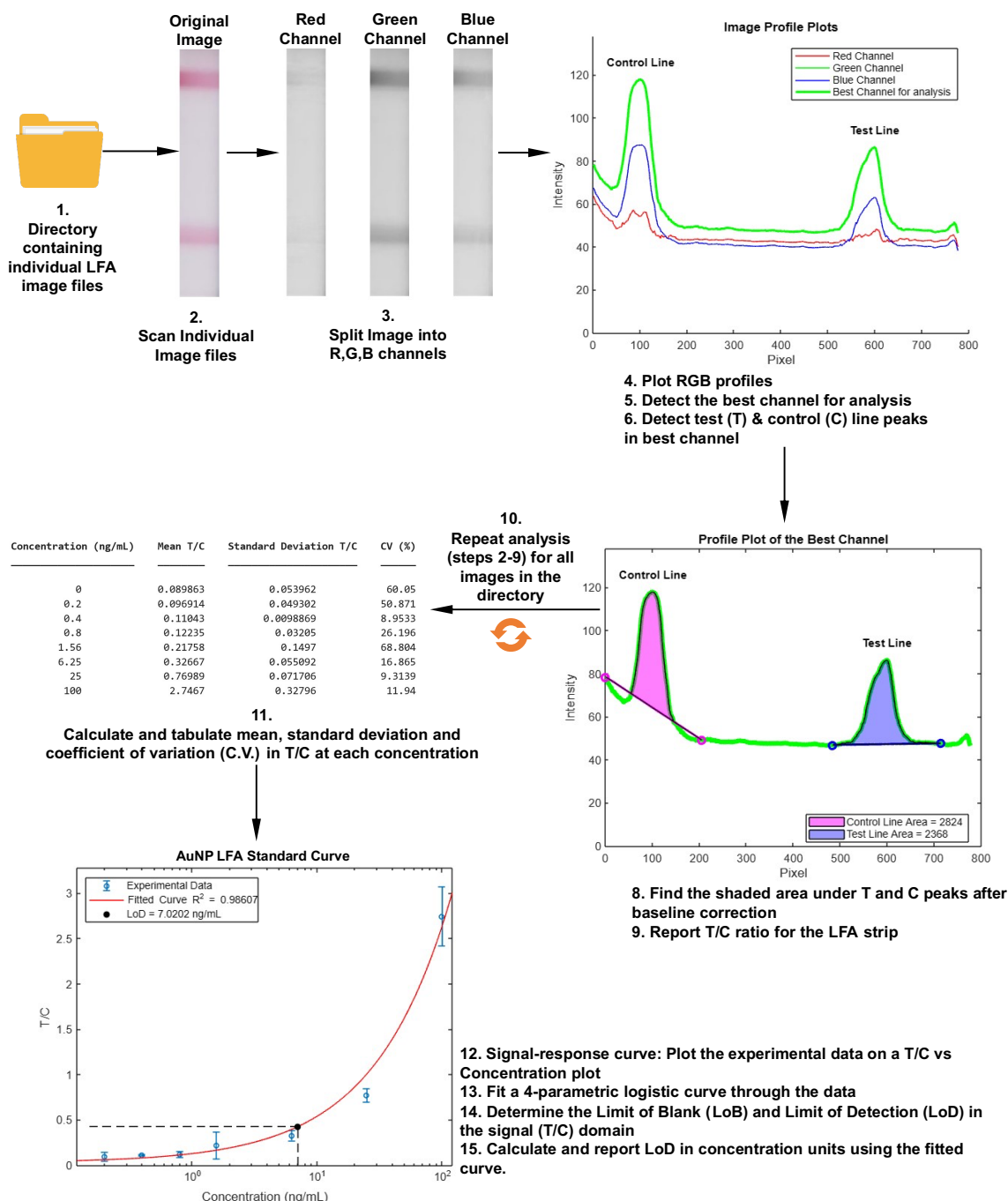


Figure 3 Image Processing and Data Analysis Workflow followed in the Live Script



## Code Availability

The MATLAB® Live Script code (.mlx) is available on MATLAB® File Exchange. Click on the following link:

[Integrated Batch LFA Image Processing and Data Analysis:](#)

Or copy/paste the following link in browser:

<https://in.mathworks.com/matlabcentral/fileexchange/160033-integrated-batch-lfa-image-processing-and-data-analysis>

## Acknowledgement

The authors would like to thank the Supercomputer Education Research Centre (SERC), Indian Institute of Science, Bangalore for providing access to MATLAB® through a MathWorks® student license.

## Conflicts of Interest

There are no conflicts of interest to declare.

## References

1. Quesada-González, D. & Merkoçi, A. Nanoparticle-based lateral flow biosensors. *Biosens Bioelectron* **73**, 47–63 (2015).
2. Sena-Torralba, A., Álvarez-Diduk, R., Parolo, C., Piper, A. & Merkoçi, A. Toward Next Generation Lateral Flow Assays: Integration of Nanomaterials. *Chem Rev* **122**, 14881–14910 (2022).
3. Parolo, C. & Merkoçi, A. Paper-based nanobiosensors for diagnostics. *Chem Soc Rev* **42**, 450–457 (2012).
4. Narayanaswamy, S. Paper microfluidic tools to improve the sensitivity and dynamic range of point-of-care immunoassays. (Indian Institute of Science, Bangalore, 2021).
5. Budd, J. *et al.* Lateral flow test engineering and lessons learned from COVID-19. *Nature Reviews Bioengineering* **2023 1:1** **1**, 13–31 (2023).
6. Liu, Y., Zhan, L., Qin, Z., Sackrison, J. & Bischof, J. C. Ultrasensitive and Highly Specific Lateral Flow Assays for Point-of-Care Diagnosis. *ACS Nano* **15**, 3593–3611 (2021).
7. Ahmed, S., Bui, M. P. N. & Abbas, A. Paper-based chemical and biological sensors: Engineering aspects. *Biosens Bioelectron* **77**, 249–263 (2016).



8. Martinez, A. W., Phillips, S. T., Whitesides, G. M. & Carrilho, E. Diagnostics for the developing world: Microfluidic paper-based analytical devices. *Anal Chem* **82**, 3–10 (2010).
9. Sathishkumar, N. & Toley, B. J. Paper-microfluidic signal-enhanced immunoassays. in *Progress in Molecular Biology and Translational Science* vol. 186 267–288 (Elsevier B.V., 2022).
10. Karan Saxena & Bhushan Toley. Novel Oxidative Coupling-Based Chromogenic Substrates for Horseradish Peroxidase-Enhanced Lateral Flow Immunoassays: A Highly Sensitive and Economical Alternative to Conventional Substrates. *ChemRxiv* (2023).
11. Dong, X. *et al.* A Handheld Reader with Optimized Image Analysis for HIV Lateral Flow Test. in *2023 16th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* 1–6 (IEEE, 2023). doi:10.1109/CISP-BMEI60920.2023.10373323.
12. Qin, Q. *et al.* Algorithms for immunochromatographic assay: review and impact on future application. *Analyst* **144**, 5659–5676 (2019).
13. Qin, Q. *et al.* Deep Learning on chromatographic data for Segmentation and Sensitive Analysis. *J Chromatogr A* **1634**, 461680 (2020).
14. Jing, M. *et al.* A Novel Method for Quantitative Analysis of C-Reactive Protein Lateral Flow Immunoassays Images via CMOS Sensor and Recurrent Neural Networks. *IEEE J Transl Eng Health Med* **9**, 1–15 (2021).
15. Wang, X., Chang, T.-W., Lin, G., Gartia, M. R. & Liu, G. L. Self-Referenced Smartphone-Based Nanoplasmonic Imaging Platform for Colorimetric Biochemical Sensing. *Anal Chem* **89**, 611–615 (2017).
16. Yan, W. *et al.* Machine Learning Approach to Enhance the Performance of MNP-Labeled Lateral Flow Immunoassay. *Nanomicro Lett* **11**, 7 (2019).
17. Schary, W. *et al.* Open-Source, Adaptable, All-in-One Smartphone-Based System for Quantitative Analysis of Point-of-Care Diagnostics. *Diagnostics* **12**, 589 (2022).
18. Park, J.-H., Park, E.-K., Cho, Y. K., Shin, I.-S. & Lee, H. Normalizing the Optical Signal Enables Robust Assays with Lateral Flow Biosensors. *ACS Omega* **7**, 17723–17731 (2022).
19. Colombo, M. *et al.* Real-time, smartphone-based processing of lateral flow assays for early failure detection and rapid testing workflows. *Sensors & Diagnostics* **2**, 100–110 (2023).

20. Bheemavarapu, L. P., Shah, M. I., Joseph, J. & Sivaprakasam, M. IQVision: An Image-Based Evaluation Tool for Quantitative Lateral Flow Immunoassay Kits. *Biosensors (Basel)* **11**, 211 (2021).
21. Wong, N. C. K. *et al.* Machine learning to support visual auditing of home-based lateral flow immunoassay self-test results for SARS-CoV-2 antibodies. *Communications Medicine* **2**, 78 (2022).
22. Parolo, C. *et al.* Tutorial: design and fabrication of nanoparticle-based lateral-flow immunoassays. *Nature Protocols* **2020 15:12** **15**, 3788–3816 (2020).
23. Kantor, A. G., Scherr, T. F., Wright, D. W. & Verberne-Sutton, S. D. Incorporating Lateral Flow Assays into Undergraduate Analytical Chemistry Lab Curricula for In-Person, Hybrid, and Remote Learning Formats. *J Chem Educ* **99**, 902–909 (2022).
24. Armbruster, D. A. & Pry, T. Limit of blank, limit of detection and limit of quantitation. *Clin Biochem Rev* **29 Suppl 1**, S49-52 (2008).