# KICA User Guide

Version: 1.0

## Introduction

KICA (Kinetic Image Cytometry Analyser) is a MATLAB-based software package for processing and analysing high-throughput time-series cytometry data. KICA provides functionalities for signal denoising, baseline drift removal, pulse detection, and calculation of cardiac-specific metrics like action potential duration (APD) and signal rise/fall times. This user guide will help you install, configure, and use KICA effectively.

## Table of Contents

1. System Requirements
2. Installation Instructions
3. Input File Format
   1. Generic CSV
   2. CyteSeer CSV
4. Software Workflow
5. Getting Started
   1. Setting up analysis parameters
   2. Running KICA
   3. Analysis results
6. Software Scaling Performance
7. Contact and Support

## 1. System Requirements

1. Operating System: Microsoft Windows 10 or higher (ActiveX support required for Excel reporting)
2. MATLAB Version: R2020B or above with the following toolboxes installed:
   1. Parallel Computing Toolbox
   2. Audio Toolbox
   3. Bioinformatics Toolbox
   4. Signal Processing Toolbox
   5. Wavelet Toolbox
3. Hardware: Intel Core i7 processor (or equivalent), Minimum 8 GB RAM

## 2. Installation Instructions

1. Download the KICA software package from the GitHub repository: <https://github.com/VCCRI/KICA>
2. Extract the downloaded ZIP file to a directory of your choice.
3. Open MATLAB and set the extracted directory as your working folder:  
   >> cd /root\_folder/KICA

## 3. Input File Format

KICA accepts input files in the following formats:

1. Generic CSV

Each file must contain time-series data with rows representing time points in milliseconds and columns representing fluorescent intensities of individual cells. For example, see KICA/Examples/input\_data\_formats/generic.csv. A snapshot of the file is provided below:

A table with numbers and lines

Description automatically generated

1. CyteSeer CSV  
   KICA supports CSV files generated by CyteSeer version 3.0.0.1 and older (KICA/Examples/input\_data\_formats/cyteseer\_3.0.0.1\_voltage.csv), and version 3.0.1.0 and newer (KICA/Examples/input\_data\_formats/cyteseer\_3.0.1.0\_voltage.csv)

## 4. Software Workflow

1. Figure below shows the overall KICA software workflow:

A diagram of a process

Description automatically generated with medium confidence

1. KICA Data Import: Reads input files and extracts metadata.
2. Cell Analysis: Denoises signals, removes drift, and calculates signal-to-noise ratio (SNR).
3. Pulse Analysis: Detects pulses, peaks, and calculates action potential metrics.
4. Quality Control: Flags invalid signals based on predefined thresholds.
5. Reporting: Generates Excel reports and visual outputs.

## 5. Getting Started

1. Setting up analysis parameters.

A screenshot of a computer

Description automatically generatedOpen Start.m file in the MATLAB editor to change the following parameters according to your requirements:

* 1. Signal type
     1. pulseDetection\_signalType = SignalType.Calcium; or
     2. pulseDetection\_signalType = SignalType.Voltage;
  2. Pulse stimulation parameters
     1. stimuliPeriod\_ms = 1000;
     2. stimuliNumber = 10;
     3. stimulationStart\_ms = 5000;
  3. Pulse detection parameters
     1. pulseDetection\_thresholdPercentage = 20;
     2. pulseDetection\_numberOfSecondsAtStartToIgnore = 0;
  4. Pulse analysis parameters
     1. pulseAnalysis\_apDurations = [30 50 75 90 95];
     2. pulseAnalysis\_pulseStartOnStimulusDetectionDelta\_ms = 30;
     3. pulseAnalysis\_pulseStartPointType = PulseStartPointType.ActivationPoint; or
     4. pulseAnalysis\_pulseStartPointType = PulseStartPointType.UpstrokeStart; or
     5. pulseAnalysis\_pulseStartPointType = PulseStartPointType.UpstrokeEnd;
  5. Visualisation Parameters
     1. visualize\_showVisualizedCells = false;
     2. visualize\_saveVisualizedCellsToFiles = true;
  6. Quality Control Parameters
     1. qc\_isQC\_Required = true;
     2. qc\_SNR\_Threshold = 10;
     3. qc\_writeFiguresToQCReportFile = true;
     4. qc\_checkForSNR = true;
     5. qc\_checkForNoPulsesDetected = true;

1. Running KICA
   1. To start KICA, execute the main script in MATLAB by running the Start.m module:  
      >> Start
   2. A dialog window will appear to select the input data files with the three options as shown below. The format of the csv files can be any of the ones listed in Section 3.

A white rectangular object with a white border

Description automatically generated

* + 1. Parallel analyse csv files: parallel batch analyse all csv data files in a selected folder (uses MATLAB Parallel Computing Toolbox). If this option is selected, a folder selection dialog box will appear, and you can select the folder that contains the csv files that you want to analyse:

A screenshot of a computer

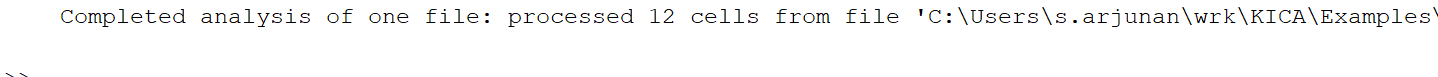
Description automatically generated

* + 1. Sequential analyse csv files: sequentially batch analyse all csv data files in a selected folder. If this option is selected, a folder selection dialog box will appear and you can select the folder that contains the csv files that you want to analyse:

A screenshot of a computer

Description automatically generated

* + 1. Analyse a csv file: analyse a single selected csv file. If this option is selected, a dialog box will appear, and you can select the input data csv file that you want to analyse:  
       A screenshot of a computer

       Description automatically generated
  1. KICA will perform the analysis on each csv file that was selected and will output the progress of the analysis in the MATLAB window. Once the analysis is completed, it will display the following

1. Analysis results
   1. The results of the analysis will be stored in a new folder called filename\_Analysis\_and\_QC located within the same folder of the input file, where filename.csv is the name of the input file. Inside filename\_Analysis\_and\_QC, there will be an Excel report file called filename\_Analysis\_and\_QC.xlsx and a folder called Figures.
   2. The Figures folder will contain cell analysis figures in .png and .fig formats. You can view the .fig file with MATLAB to further refine the figure according to your requirements.
   3. The Excel report file contains the following five tabs:
      1. *QC Passed Pulses*: listing pulse detection and analysis results of all cells that have passed QC.
      2. *QC Passed Stats (All)*: listing N, average, standard deviation (STDEV) and standard error of the mean (SEM) of all QC passed cells.
      3. *QC Passed Stats (Paced)*: listing the same statistics above but only for cells with pulse(s) that start on stimulus.
      4. *QC Passed Data*: listing raw baseline corrected data for all QC passed cells.
      5. *QC Failed Pulses*: listing pulse detection and analysis results of all cells that have failed QC.
   4. Below is a snapshot of QC Passed Pulses tab of the example file KICA/Examples/voltage/voltage\_example.csv with the following parameters:
      1. Signal type
         1. pulseDetection\_signalType = SignalType.Voltage;
      2. Pulse stimulation parameters
         1. stimuliPeriod\_ms = 1000;
         2. stimuliNumber = 10;
         3. stimulationStart\_ms = 5000;

A screenshot of a graph

Description automatically generated

* 1. Below is a detailed pulse detection figure with a legend describing each plotted line (generated from .fig file):

A graph of a signal

Description automatically generated with medium confidence

* + 1. The pulse detection involves the following steps:
       1. Denoise signal
       2. Remove baseline drift
       3. Detect pulses
       4. Calculate SNR
  1. Below is a detailed pulse analysis figure with a legend describing each plotted line (generated from .fig file):

A graph of a graph

Description automatically generated with medium confidence

* + 1. Information extracted from the detected pulses:
       1. Pulse duration
       2. Activation time
       3. APD 30/50/75/90/95
       4. 10% to 90% rise time
       5. 90% to 10% fall time
  1. Below is another snapshot of QC Passed Pulses tab of the calcium example file KICA/Examples/calcium/calcium\_example.csv with the following parameters:
     1. Signal type
        1. pulseDetection\_signalType = SignalType.Calcium;
     2. Pulse stimulation parameters
        1. stimuliPeriod\_ms = 1670;
        2. stimuliNumber = 9;
        3. stimulationStart\_ms = 5000;

A screenshot of a graph

Description automatically generated

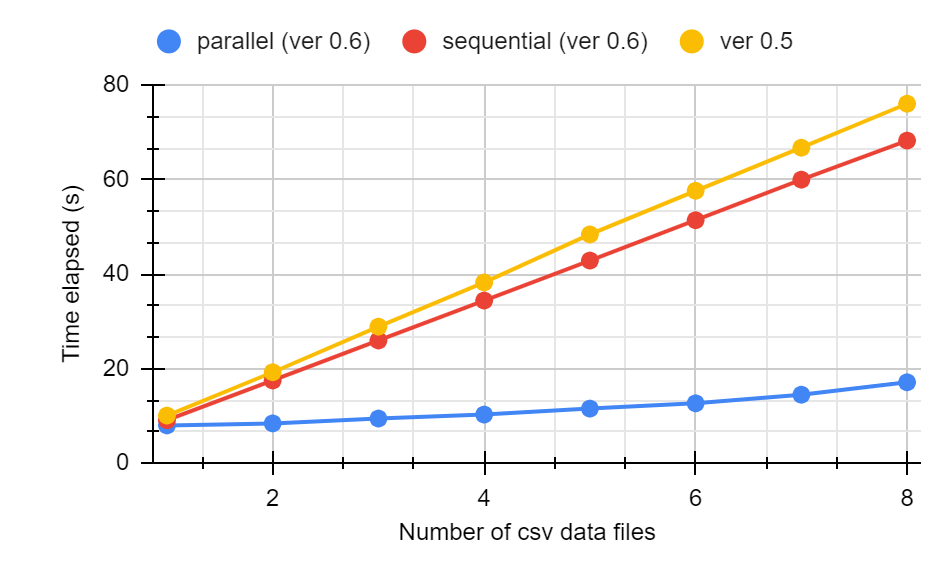
* 1. Below is a snapshot of QC Failed Pulses tab of the example file KICA/Examples/calcium/calcium\_example. Cell 27 below failed QC because the signal-to-noise (SNR) is below the 10 dB threshold set in the Start.m file.

A screenshot of a computer

Description automatically generated

## 6. Software Scaling Performance

1. A major challenge of KICA is the significant computation time required to analyse a large number of datasets.
2. Figure below shows the scaling performance of KICA. The parallelised implementation of KICA achieved 4.45x shorter execution time with 8 CPU cores.



1. The above results indicate that KICA scales well with the addition of CPU cores.

## 7. Contact and Support

For technical support or to report bugs, please contact:  
- Email: [s.arjunan@victorchang.edu.au](mailto:s.arjunan@victorchang.edu.au), [a.hill@victorchang.edu.au](mailto:a.hill@victorchang.edu.au)   
- GitHub Issues: <https://github.com/VCCRI/KICA/issues>