

Multiplex Immunofluorescence Titration Optimization (mIFTO) R package

v. 2.00.0017

Created by: Benjamin Green & Charles Roberts

Tumor Microenvironment Technology Development Center

The Johns Hopkins University Bloomberg~Kimmel Institute for Cancer Immunotherapy

Correspondence to: bgreen42@jhu.edu

Section 1: Summary

This R package was developed to help organize and quantify the inForm Cell Analysis output for pixel-by-pixel, cell-by-cell, and tissue segmentation data. The primary goal of this package is to aid in the assessment and determination of optimum staining conditions for multiplex immunofluorescence titrations on Akoya's scanning and staining platform. There are two primary functions of mIFTO; *FOP()* and *mIFTOapp()*. *FOP* was designed to measure the 'fractions of positivity' for individual conditions. Then, with a unique identifier, the group conditions together into a single csv output file. The function uses the inForm output tables (IF or IHC) for cell segmented, colocalization data, or tissue segmented data as input. The other primary function, *mIFTOapp()*, was developed to aid in determining an optimum condition for a series of reagent titrations. The output includes t-statistics, signal-to-noise ratios, fractions of positivity, and boxplots of expression profiles.

Section 2: Getting Started

Section 2.1: Initial Package Install

Open an Rstudio session to get started. Next, install the package from github using the following commands:

```
install.packages('devtools')
```

```
library(devtools)
```

```
install_github("beng1290/mIFTO")
```

When the package begins to install a number of messages will appear in the console indicating the status of different events, especially during the initial install or just after R has been updated, this is normal. Some of the messages may be in red, this does not indicate an error. If there is an error during installation, usually the output message to the console will start with "Error:" or "Warning:", then describe the corresponding error.

The installation of 'devtools' is usually the most interactive and takes the longest. This is a provided R package which is used for external programmers to develop R packages. During the 'devtools' installation command (command 1 above), R may require some user input examples of this include

1. R may indicate that some packages should be updated. The user can either indicate to update all, some or none of the packages. It is usually best to select the update all option.
2. A dialog box may also open asking the user if the package should 'be compiled from source', it is usually best to answer 'yes' to this

3. A dialog box may open telling the user that a package is in use and that the R session must be restarted to continue installation of a package. It is usually best to allow R to do so. Note that it may take a few moments for R to reinitialize after it restarts, this is normal.

Errors are common in this step, usually when updating packages. If a package fails to update properly and crashes the installation process an error message is usually output beginning with: "Error: package 'failed_package_name' failed to install". Installing the package separately may be necessary. Enter `install.packages('failed_package_name')`; replacing 'failed_package_name' with the name of the failed package.

For more advanced users of R, it may be easier to specify additional options during a package installation. For example, when installing 'devtools' the following options can be specified:

`install.packages('devtools', ask = FALSE, quiet = TRUE, verbose = FALSE)`

This command will install 'devtools' without asking the user if it is okay to update or compile a package and will carry out the procedures automatically. The command will also significantly reduce the output to the console.

When the `install_github('beng1290/mIFTO')` command is used, R will output the following messages, followed by a number of other messages.

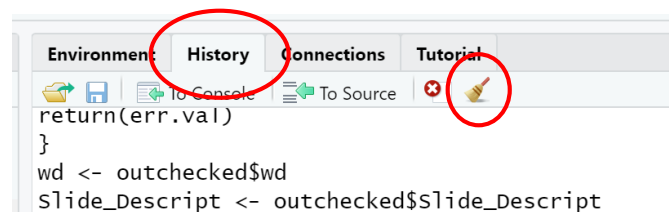
```
Downloading GitHub repo beng1290/mIFTO@master
✓ checking for file 'C:\Users\bgreen42\AppData\Local\Temp\26\RtmpQfDpMO\remotes387c25c35c49\beng1290-mIFTO-8d5df6c/DESCRIPTION' (532ms)
- preparing 'mIFTO': (362ms)
✓ checking DESCRIPTION meta-information ...
- checking for LF line-endings in source and make files and shell scripts
- checking for empty or unneeded directories
- building 'mIFTO_2.00.016.tar.gz'
```

It is advisable to check the circled region above to make sure that the correct version of the package is installed.

Section 2.2: Updating the Package

When an update for the package is available on github, it is advisable that, if the user already had an R session open with an old version running, the user clear their R history, completely close their current R session or window, and open a new session.

To clear the R history, go to the 'history' tab on the top right of the R user interface and select the broom icon, as shown below.



After the new R session is open, the advisable steps for updating the mIFTO R package would be the same as a new install:

`install.packages('devtools')`

`library(devtools)`

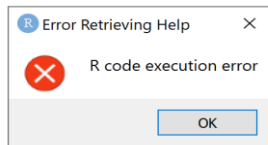
`install_github('beng1290/mIFTO')`

For additional details see section 2.1. If the most up to date package is already installed under the current user profile the following message may appear instead:

```
> devtools::install_github('beng1290/mIFTO')
Skipping install of 'mIFTO' from a github remote, the SHA1 (cacb5547) has not changed since last install.
Use 'force = TRUE' to force installation
```

If this is a potential mistake it is then possible to use `install_github('beng1290/mIFTO', force = T)` in order to force an updated installation of the R package.

A second installation may corrupt the package in this case the following message will appear:



Once 'Ok' is clicked the following lines will be printed into the console (with a local R library path in place instead of bgreen42):

```
^ DONE (mIFTO)
Error in fetch(key) :
  lazy-load database 'C:/Users/bgreen42/Documents/R/win-library/4.0/mIFTO/help/mIFTO.rdb' is corrupt
```

To fix this error, close the current R sessions completely, open a new session, and reinstall the package.

Section 2.3: Running Package

The installation calls and devtool library call will only be needed the first time the mIFTO package is used, unless additional updates to the package have been posted (see section 2.2).

Once installed once for a user profile, the user can run the apps one of two ways:

1. load a library into the current 'workspace' by using `library(current_package)` replacing 'current_package' for whichever library is needed. To load the mIFTO R package type: `library(mIFTO)`. Afterward the call the apps using `FOP()` and `mIFTOapp()`.
2. Explicitly call the package and the functions simultaneously without loading the package into memory. In this case use the syntax `package_name::function_name`. For the main apps described in this document use `mIFTO::FOP()` and `mIFTO::mIFTOapp()`.

Code performance is reliant on the local systems connection to the images as well as the computational resources of the local system. Since the image data for these images can be very large, it is advisable that either the images are kept local to the machine the software is run on or that they can be accessed by a high speed network connection.

Section 3: FOP

Section 3.1: Description

This function is used to measure the fraction of positivity across a subset of images for uniquely named slide. The function is designed to count positivity for cell-by-cell, pixel-by-pixel, or tissue segmented data exported from inForm.

Section 3.2: Setting up Slides Names

When naming slides for this protocol, be sure to name slides using unique slide ids, followed by unique condition ids. An example of an acceptable name would be M1_Multiplex1 (M1 being the slide id and Multiplex the condition id).

Section 3.3: Extracting and Exporting Data from inForm

- a. For **cell-by-cell data**:
 - use the phenotype module and export the cell segmented inForm, name the positive phenotype with the same case as the input for 'Primary Antibody' in the GUI (see below).
- b. For **pixel-by-pixel**:
 - use the colocalization module of inForm and export the colocalization data. Do not name the opals in the 'prepare' tab of inForm.
- c. For **tissue segmentation**:
 - use the tissue segmented module in inForm and export the tissue segmentation data. Name the categories 'Tumor' and 'Non Tumor' in inForm.

For all exports, do not name the opals in the 'prepare' tab of inForm, the code will not be able to find the correct columns. Note that the code will generate statistics for all slides in the selected folder (see below for more details in step 5).

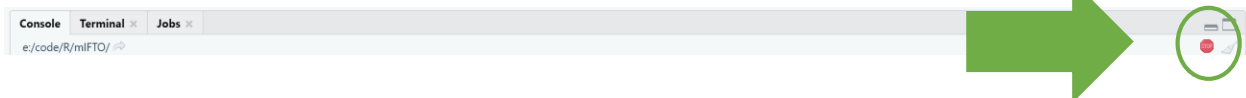
Section 3.4: Running

1. Open R Studio and install the package (see '[Section 2: Getting Started](#)' for installation instructions)
2. Type `mIFTO::FOP()` & ENTER
3. The app will open in a web browser with the following inputs
 - a. Slide Descriptor:
 - i. This is where the specimen numbers or other slide distinctions are located
 - ii. Do not add '-' or spaces in names
 - iii. Place a comma between names
 - iv. E.g.: Melanoma1,Melanoma2
 - b. Primary Antibody:
 - i. This is the antibody the analysis is running for
 - ii. E.g.: CD8
 - c. Other condition delineation:
 - i. Anything else in the name that could separate different conditions in excel spreadsheet output
 - ii. E.g.: (V1, IHC, IF, 1to100, 1to200)
 - iii. E.g., for GranB put IF in the 'other condition delineation' field for the IF and IHC in the 'other condition delineation' field for the IHC you will would get the following:

Concentration	CL22	CL26	CL31	CL32	CL35	Antibody
IF	0.015073	0.001671	0.007988	0.020456	0.000515	GranB
IHC	0.010049	0.004641	0.010594	0.01801	0.001286	GranB

- d. Primary Opal
 - i. This is the Opal the primary AB is in
 - ii. E.g.: 540

- e. Is this IHC?
 - i. Blank = no
 - ii. Check = yes
 - f. What kind of positivity measure?
 - i. Drop down box with the following options
 1. PPC pixels
 2. Cells
 3. Tissue
 4. Click 'Run FOP'
 5. A windows explorer will ask you to point it to the directory with the data inside
 - a. The code will generate statistics for all slides in the selected folder.
 6. The program will run for this condition then ask if there is another condition
 - a. Only say yes if the Slide Descriptors are the exact same
 7. Yes:
 - a. A new GUI will open asking for the new:
 - i. Primary Antibody
 - ii. Other condition delineation
 - iii. Primary Opal
 - iv. IHC?
 - b. Click Run
 - c. A windows explorer will ask you to point it to the directory with the data inside
 - d. The program will run and ask if there is more data
 8. No:
 - a. A window explorer will open for the output directory
 9. A file called +pixels will be located in the output directory
- Once finished running the app will idle in the background and continue to wait for input. To stop the UI click the red 'stop' sign button just above the console window in RStudio and close the web browser.



Section 3.5: Common things to check after running into errors

1. Not putting conditions into separate folders
2. Misspelling input to charts
3. Not labeling Opals – AB pairs in inform output
4. Leaving out Slide Descriptors for slides in the data folder when writing into input box
5. Typing Slide Descriptor names in wrong
6. Sometimes the folder browsing window opens behind the Rstudio session.
7. The UI will not respond well to multiple clicks
8. You may need to adjust the browser window sizing on your machine to get everything to line up well
 - In output, if some columns do not average
 - a. See 4 or 5

Section 4: miFTOapp

Section 4.1: Description

This function is used to select optimum dilutions from a titration series. For a titration series, at least 3 cases with 10HPFs each should be used. For best results stain at least 5 dilutions, one above and 4 below the manufactures recommended concentration.

Section 4.2: Setting up Slides Names

When scanning, so that the slides are compatible with the code, name the slides with the following naming convention switching out only the bracketed and italic expressions being sure to use underscores between words:

[SlideID]_[PrimaryAntibody]_1to[CocentrationofPrimaryAntibody]_[PolymerUsed]_Opal[OpalUsed]_1to[OpalConcentrationUsed]. An example of a properly named slide is

T6_PD1_1to150_PV50_Opal650_1to50, this is from slide 'T6', stained with 'PD1' at '1to150', using a power vision polymer at 50 percent (PV50), in 'Opal650' with an opal concentration of '1to50'.

Section 4.3: Extracting and Exporting Data from inForm

Once the images have been stained and scanned, establish thresholds for pixel by pixel data or phenotype the data before exporting cell segmented data for optimum accuracy in the respective modalities. The pixel analysis and cell analysis will be performed separately in the code usually one modality or the other is used to determine the optimal dilution.

For cell-by-cell analysis export the cell segmented for each HPF. For pixel-by-pixel analysis export the component data images for each HPF. Dilutions\ conditions can either be exported into separate or a single folder. Be sure to follow naming conventions if the data is in separate folders for dilutions using the above naming convention for the folders removing slide designations as well as the first underscore (for the example this would be 'PD1_1to150_PV50_Opal650_1to50'). Once the preprocessing has been finished move on to running the code.

Notes on naming convention: It is important that the slides are named in this way at scanning as the slide names are propagated to the image HPFs and the image data. In order to differentiate between conditions/ slides and collect the correct data, the functions look for specific keys in the names. For example it looks for the slide name, in this example 'T6', followed by an underscore to determine that the slide is not actually 'T60' or 'T600'. This is most important in the dilution series where one might titrate 1to100 and 1to1000 in the same series, without this naming convention the code might mistake all of these slides as 1to100. Additionally, since the slides can have different primary and opal concentrations, it is useful to always designate both of these as a standard so that the code searches for the concentration that directly follows the primary antibody or opal.

Section 4.4: Running

1. Open R Studio and install the package (see '[Section 2: Getting Started](#)' for installation instructions)
2. Type `mIFTO::mIFTOapp()` & ENTER
3. The app will open in a web browser with the following inputs: (Note: it may be necessary to resize the window to allow all fields to line up appropriately)
 - a. Slide Descriptor:
 - i. This is where the specimen numbers or other slide distinction are located
 - ii. Do not add – or spaces in names
 - iii. Place a comma between names
 - iv. E.g. Melanoma1,Melanoma2
 - b. Primary Antibody:

- i. This is the antibody the analysis is running for
 - ii. E.g. CD8
 - c. Concentration:
 - i. Concentrations that have been completed
 - ii. 100,200,400
 - d. Primary Opal
 - i. This is the Opal the primary AB is in
 - ii. E.g. 540
 - e. Was more than one dilution used in the name? (Was naming convention followed?)
 - i. Check Yes if the files are named properly
 - f. What is being titrated
 - i. Primary
 - ii. TSA
- For Cell segmented data – NOT YET FUNCTIONING
 - g. Was the data phenotyped
 - i. Checked = yes
 - ii. Blank – no
 - h. Cell Compartment for Analysis:
 - i. Membrane
 - ii. Nucleus
 - iii. Cytoplasm
 - iv. EntireCell
 - i. What was the name used for positive phenotype
 - i. The name used in inform
 - j. Is the data in separate folders according to dilution
 - i. Check = yes
 - ii. Blank = no
 - k. Were antibodies named in inform
 - i. Where the opal – ab pairs give for inform
 - l. Is the antibody sparse
 - i. Will do decile data out of 100 instead of 10
 - ii. Check = yes
 - iii. Blank = no
 - m. If so list AB in order...:
 - i. If Abs – Opals were named in inform list the Abs
 - ii. E.g.. PDL1,CD8,FoxP3,Tumor,PD1,CD163
- For Pixel data
 - g. Is the data in separate folders according to dilution
 - i. Check = yes
 - ii. Blank = no
 - h. Was an IHC thresholded with this dilution -DO NOT USE NOT YET FUNCTIONAL
 - i. Check = yes
 - ii. Blank = no
 - i. Were thresholds different for each case -DO NOT USE NOT YET FUNCTIONAL
 - i. Check = yes
 - ii. Blank = no
 - j. List the thresholds in order of increasing dilution separated by a comma
 - i. E.g. 12.5,13.6,14.7

- n. Must be same length as number of concentrations
8. List the thresholds in order of increasing dilution separated by a comma
 - a. E.g. 12,13,14
 - b. Number of pixels above the threshold which must be connected for the values to be included in the positive signal, (setting can be set in inForm colocalization module and should be translated here)
 - c. Must be same length as number of concentrations and should be whole numbers
9. Click 'Run Pixel-by-Pixel'
10. A windows explorer box will be generated to find data folder
 - a. If data is in separate folders select the parent folder
 - b. The windows explorer and progress bar may open behind the Rstudio or web browser window
11. The program will produce results in a Results or Results.pixels folder for cell or pixel data respectively
12. There will be a 'Finished' Dialog when the software finished
 - a. for most errors a dialog should open with a clear message of what went wrong

Section 4.5: Common things to check after running into errors

1. if the error causes the UI windows to grey out simply close the web browser window, select the red 'stop sign' in the Rstudio console and rerun the UI using 'mIFTO::mIFTOapp()'



2. Sometimes it may be necessary to resize the window manually to get the UI to line up appropriately.

Section 4.6: Results for mIFTO::mIFTOapp() – What they are

1. Stats
 - a. Use these graphs and stats to determine the best dilution for the titration series
 - b. **The test statistics are invalid after a significant amount of signal is lost** into the noise populations; for this reason be sure to check the histograms for the intensity profiles\ thresholds and the fraction of positivity to determine if the statistics are valid
 - c. Boxplots will aid in the description of the intensity profile; SN Ratio graphs and t test graphs will aid in determining the ability to determine signal from noise;
 - i. Note: the t test measure is more robust than the SN Ratio graphs in the 'true population range' (before signal loss and after signal saturation has decreased)
 - d. boxPlots/
 - i. Decile and Positive/Negative (either threshold or phenotype) box plots for signal only and all data
 - ii. Pixel data will produce violin plots of the data here
 - iii. Boxplots for phenotype data are made using R boxplot function
 - iv. Boxplots for pixel data were created using calculated values from the data
 1. The middle line is the median of the data
 2. The top and bottom of the boxes are the $\frac{1}{4}$ and $\frac{3}{4}$ of the data respectively

3. The outer bounds are the top and bottom inner fences calculated as $Q1 - (1.5 * IQR)$ and $Q3 + (1.5 * IQR)$ respectively
 - e. graphs/
 - i. Welch t-tests and signal to noise ratio graphs for the data
 - ii. Signal to noise ratio graphs show separation between estimated signal and noise
 - iii. Welch t-test
 1. Tests how dissimilar the Positive and Negative populations are; **higher value = higher separation between signal and noise**
 2. Assumes normal distribution but not equal variance between the two populations
 3.
$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$
 - f. Fractions of positive cells or pixels
2. Histograms
 - a. Use these to understand the intensity profiles
 - b. For histograms:
 - i. x- axis is the log2 of the intensity
 - ii. y -axis is the density of values
 - iii. bin number is calculated using Freedman–Diaconis' rule
 1. $2 * \frac{IQR(x)}{n^{1/3}}$
 2. If IQR = 0 or another error occurs in resolving bins; then bins resolve to 200 for cell data or 1000 for pixel data
 - c. Plus 1 graphs are graphs with an epsilon of 1 added before the log and Plus .001 are with an epsilon of .001
 - i. Adding 1 may distort the noise population and some of the lower signal, therefore if the antibody is weakly stain or is sparse it may be better to use .001. If the population is distinct and robust it may be easier to see the separation with an epsilon of 1.
 - d. Overlaid histograms:
 - i. Overlays the histograms of different dilutions to see how the intensity profiles change over a dilution series
 - e. Individual histograms
 - i. Shows the individual intensity profiles
 - ii. The line is the threshold
 - f. Data/
 - i. Data for histograms
 1. This is the data to recreate histograms for later use
 2. Columns
 - a. Mids - midpoints of bins
 - b. Density – density of data
 - c. Counts – number of data in that bin
 - d. Concentration/Slide

Section 4.7: Results for mIFTOapp() – How to interpret

It is best to start with the histograms for the given cases. Opening the histograms for individual images or slides helps understand the distribution of signal across cases and images. The histograms are plotted as the fraction of positivity vs. signal intensity. The ideal histograms for this type of data should appear to be made up of two Gaussian distributions. One of the distributions will appear to have a mode or peak far to the left at low intensity, the noise, and the other will be closer to the right, the signal, at higher intensity. As is apparent when looking at the images, these distributions separate out nicely for some antibodies more than others. They are also more separated and have higher intensity at concentrated dilutions, tending to merge together and decrease in intensity at lower concentrations. Creating separate exposure times for the different dilutions will lead to seeing more separation or levels at lower concentrations, but because the exposure times are normalized in inForm's unmixing the intensities can still be fairly compared across dilutions. Looking at the distributions alone one may be able to, very early on, toss out some of the dilution options if the separation or signal visibly is too low. This is also a great place to check the thresholds for pixel based analysis, the thresholds should generally be consistent across dilutions falling in the same place relative to the inflection between the distributions. The threshold should have optimally been placed slightly toward the right of the inflection point, from experience this placement properly accounts for off target staining, artifact, and background. For the best results, placing the threshold optimally is less important than placing the threshold consistently.

Once it is determined that the thresholds are fair, we move on to the fraction of positivity output csv files. Compare these results with positivity on an adjacent slide stained with an optimized\ validated IHC stain. Note that when comparing between IHC and IF, one must be aware of adjacent pixel leakage. If the signal on a cell is very high in IF the fluorescence of intensity will 'leak' into adjacent pixels, this can inflate the numbers when taking IF measurements. When comparing IHC to IF, look to where the IF signal of any of the test cases drops below the IHC signal. It might be worthwhile do a statistical test to determine loss of signal.

Once this 'loss of signal' is established, in even one case, only conditions above this signal should be considered. For primary antibody dilution series, open the t-tests and the signal-to-noise ratio graphs; from the remaining concentrations select the highest 'average' peak in both tests. Understand that the t-test measures the difference between the populations taking into account their respective signal distributions. A higher t-test value indicates 'tighter' distributions with more separation between modes. This can sometimes become erratic if the distributions are not well defined, the marker is sparse, or the signal is too saturated. The signal-to-noise ratio is simply a ratio defined as the positive signal (above the threshold) / noise (below the threshold). This metric is much more simple and straightforward.

To select the TSA dilution from the remaining concentrations, look at the boxplots to see how low the average signal drops. Balancing the intensity counts across opals in a panel helps reduce bleed-through or crosstalk between channels. Select opal concentrations that balance the counts.