**Project Title:** Two Channel Emission Analysis

**Overview:** This code repository will contain all the scripts necessary to prepare raw movie files of two channel images and obtain intensities in two channels. The general workflow is based upon ideas from Curly Zuo, but significantly changed to be readable/usable by another human and run more efficiently. Also the registration algorithm is an original implementation.

Original Author: Zechariah Pfaffenberger

Original working version date: 01-08-23

Updating Author:

Update Date:

**List of Scripts that are in this repository:**

1. selectNanoparticlesForOverlay.m
2. cropAndOverlay.m
3. appendNPs.m
4. finNP\_mol\_off\_win.m
5. subtractChannelFromFits.m
6. twoChannelWrapper.m

**Additional Dependency Scripts**

1. SMALL-LABS: I assume you have the full SMALL-LABS repository on your MATLAB path
2. findNanoparticle: script from that will allow user to locate nanoparticle positions within a movie
3. groupPositions: script that will organize all the fit positions from SMALL-LABS into “on particle” and “off particle sets”

**Protocol and Explanation for Analysis**

Note: Additionally, before going into these scripts, make sure that you have converted your movies to matlab format using movie2mat script (found in SMALL-LABS repository)

1. Overlay movies- The first step is to crop out the two channels from the original full movie and then register the two images (line up the two image channels so they have the same coordinates)
   1. Associated scripts: selectNanoparticlesForOverlay.m, cropAndOverlay.m
   2. selectNanoparticlesForOverlay.m: This script is basically a wrapper for cropAndOverlay.m that allows the user to select a nanoparticle or other reference point from each channel (center and offcenter). Make sure you are selecting the same reference point in each channel and that the reference signal is approximately a gaussian diffraction limited spot. Note that the user is defining center and offcenter in this script *so make sure you know which channel is which*. The script is pretty explicit about which it is asking for. You select the particle in each channel sequentially, first the center channel is selected (pink border), second, the offcenter channel is selected (dark blue border). The first movie you run this on, you will have to select particle manually. However, in subsequent movies, you set loadGuessPos to 1 and then load in previous guess positions from a separate matlab file.
      1. *Important Settings*
         1. saveGuessPos- Boolean, toggles whether you will save the guess positions to a separate matlab file for loading later so that you don’t have to select nanoparticles in every video
         2. loadGuessPos- Boolean, set to 1 if you want to load the saved guess positions, set to 0 if you want to manually select guess positions. Will throw an error if you don’t have a file with the saved positions in the same directory as the movie.
         3. checkGuesses- toggles whether you want to view the guess boxes overlaid on a sum movie before moving into a fitting attempt
         4. pxsz - size of the pixels in the movie from Ronchi ruling calibration
         5. updateSaveGuesses- Boolean, toggle whether the fit position from a movie gets saved in the saved guess positions to be used in subsequent movies as the starting guess.
   3. cropAndOverlay.m: This script does the major work for overlaying the center and off-center channels. It takes in the total summed movie image as well as the coordinates fit to the particle positions from “selectNanoparticlesForOverlay.m” as well as the original movie and some parameters. First it will create two boxes that surround the pixels in each channel on the camera based upon the fit particles positions. Then it will crop out each of these boxes and attempt to overlay them using a registration algorithm of the user’s choice. There are three algorithms currently implemented. [Optimizer Affine](https://www.mathworks.com/help/images/ref/imregister.html), [Phase Correlation](https://www.mathworks.com/help/images/use-phase-correlation-as-preprocessing-step-in-registration.html), and [Control Points](https://www.mathworks.com/help/images/control-point-registration.html). Control points is the one used most often for the nanoparticle arrays as you have very specific points you know should line up in the movie (the nanoparticle centers).
      1. *Important settings*
         1. checkROIs: Boolean, toggle for whether the code will display an overlay of the two crop boxes on top of a summed movie image so user can visualize what areas will be cropped out
         2. ROIsize: integer, size in pixels for each square crop region, this will determine the dimensions of the final overlay movie. (e.g. if you set it to 150, the final overlay movie will a 150 x 150 pixel array). This size will basically be around the particles you selected in selectNanoparticlesForOverlay and how that works will be determined by cropType (next input parameter)
         3. cropType: this is a string that tells the script which particle you select in the array (e.g. if it was the top left particle you set it to ‘topleft’) The comments on lines 44-54 show what options you have and do a decent job describing what that option will actually do. Basically, you might want to change this because not every video will have a particle in the same position with high signal. For example, the bottom right might be best for some movies while top left is better for others. Additionally, if the drift between two subsequent movies is large or if you move the array for some reason, the particle you want can drift out of the box. You can always add in additional options between lines 82 and 109.
         4. registrationType: string, this tells the script what type of registration algorithm you want to use. For the nanoparticles, mostly we use the Control Points method.
   4. Notes on step 1:
      1. It’s important to have the right pixel size. Make sure to calibrate and set this
      2. If you have a drastic difference in the intensity of particles in off and on center channels, you need to adjust the contrast limits for displaying the zstack when you select the nanoparticles. This can be done in line 156-157 of selectNanoparticlesForOverlay.
   5. Outputs from Step 1:
      1. Three matlab files that contain the mov variable. Their filenames are the original movie plus “\_overlay”,”\_center”, and “\_offcenter”. Also a matlab file with the coordinates for the cropping, and transformation matrix info for the registration are saved.
2. Run SMALL-LABS as normal on the overlay movie(s)
3. Find the nanoparticles and determine on and off localizations. (findNanoparticle and groupPositions in NP\_localization scripts)
4. Append the nanoparticle positions to the guess list from SMALL-LABS and find the off frames for these nanoparticles.
   1. Associated scripts: appendNPs and findNP\_mol\_off\_win
   2. appendNPs: This script is a wrapper for findNP\_mol\_off\_win and appends all *particle* guesses to the list of all *molecule* guesses. You might want to do this if you want information on particle intensities in addition to molecule intensities.
      * 1. *Important settings:*
           1. Bgsub: **important.** Boolean, toggle to tell the script whether this was from a background subtracted movie or not. Essentially, this will determine filenames it loads in guesses and fits from because they are different depending on if you use background subtraction in small-labs or not. If you haven’t set it correctly, the script will throw a “file not found” error
           2. findNP\_off\_win: Boolean, toggle for whether to run the findNP\_mol\_off\_win script. This additional script will additionally determine the off frames for a *particle* so that you can get a more accurate intensity measurement in each channel not affected by molecules sticking down.
           3. moloffwin\_np: integer, even number. frame number of the off window for the particle. This is exactly the same sort of value as mol\_off\_win in the main SMALL-LABS script. You just sometimes want a different length the particle vs that use for a molecule (typically longer if you are getting a lot of molecule fits), Default is 2000.
           4. npBoxSz: integer, size of box for fitting nanoparticle, typically set to the same size as the diffraction limit spot size you give SMALL-LABS
      1. findNP\_mol\_off\_win: This is a script that finds the off windows for all *particles* selected in the movie. It uses the same algorithm as SMALL-LABS does for local background subtraction.
         1. *Important Settings*
            1. Really no settings in this individual script, appendNPs sets up everything for this function
5. Subtract the off frames using the off and on center channel movies to find the true intensities of the localizations and the nanoparticles in each channel
   1. Associated scripts: subtractChannelFromFits.m
   2. subtractChannelFromFits: this script loops through all the molecule guesses and appended particle guesses, finds their local background and subtracts the background away to get the true intensity of each molecule and particle in each channel. It does this by looping through all the guesses twice, once for each channel.
      1. Important Settings
         1. Bgsub: Boolean, toggle to tell script if this was from a bgsub movie or not
         2. Do\_avgsub: Boolean, mean or median subtraction to calculate the local background
         3. includeNPs: Boolean, toggle to tell script if you want to also calculate intensities for the particle localizations
         4. NP\_biggerbox: integer, size of box in pixels to do local background subtraction on the particles.
      2. Note about subtraction for particles
         1. See the comment on lines 211-223 for methodology regarding how the background for a particle is calculated
      3. Inputs:
         1. Filepaths to the overlay movies. Make sure the center and offcenter channel movies are saved in the same directory for each one.
      4. Outputs
         1. In each fits file, it will add two variables to the fits structure
            1. Mol\_intensity\_center: Nx1 double which contains the sum intensity of each localization from the center channel
            2. Mol\_intensity\_offcenter: Nx1 double which contains the sum intensity of each localization from the off-center channel
            3. N is the number of guessed molecules PLUS the number of nanoparticles if you ran appendNPs