**TRUVOT Example Instructions**

1. Connect to the schools VPN for access to IDL
2. Navigate to “truvot/example” directory
3. Launch IDL in your terminal by typing “idl” and hitting enter
4. Once IDL is opened, execute the first TRUVOT script by typing “idl\_script1” and hitting enter
5. Next launch ipython (in the same directory but in a separate terminal window) by typing “ipython --pylab”
6. Once Python is opened, execute the TRUVOT python script by typing “run uvotpy\_script.py” and hitting enter
7. Go back to the window where you are running IDL and run the second TRUVOT script by typing “idl\_script2” and hitting enter
8. Look for the file “truvot/example/00032094004/uvot/image/sw00032094004ugu\_1ord\_final.dat” and make sure it matches his “truvot/example/example\_final\_spectrum.dat”

**General TRUVOT Instructions (& Fixes)**

1. Download template data from Swift Archive website (<https://swift.gsfc.nasa.gov/archive/>)
   1. Click ‘Special Swift Interface’
   2. Find your Target ID and type it into the matching box and click ‘Start Search’
   3. Check the boxes on the left for the data that you want
   4. At the bottom of the screen, under ‘Data Products’ check ‘aux’ and ‘uvot’
   5. Click ‘Create Download Script’ and copy wget commands onto clipboard
   6. Open your terminal (and navigate into folder you would like the data to go in)
   7. Make a new text file with “touch \_\_SNName\_\_downloadcommands\_updates.txt”
   8. Edit file with “edit \_\_SNName\_\_downloadcommands\_updates.txt” and paste into text file (don’t forget to save!)
   9. Run text file with “source \_\_SNName\_\_downloadcommands\_updates.txt” and let template data downloadgu
2. If not on campus, connect to VPN
   1. Open Cisco app
   2. Click “Connect” button
   3. Log in with your NetID details and confirm DUO push on your phone
3. Navigate to your SN directory in 4 terminal windows (in one of the windows, unzip the files with “gunzip \*/uvot/image/\*.gz”)
4. **fkeyprint Window:** In this window, you will check the extension exposure times by typing in “fkeyprint \_pathtofile\_ exp” (can also check the roll angles match with pa instead of exp)
   1. Data: See how many extensions there are. ~~Take the top 3 with the longest exposure times.~~ (Can’t do this anymore bc of the python step). Take the first 3. If there is only 1, put 0 for the other table values
   2. Template: See how many extensions there are. If there is 1, use 1 as your number, if there is 2 or more, put 2 as your “te” value
5. **Ds9 Window:** In this window, you will get the x & y coordinates of the anchor and the background value for both data and template images
   1. To open a specific detector image (dt.img) type “ds9 \_pathtofile\_\[\_extension#\_\]” (if you want to open more than one image, just continue adding on file names with a space in between each)
   2. Once the detector image is open, pick an anchor (not the SN), note the x & y coordinates in ds9
   3. Pick a background region in the brighter part of the image and estimate what the average value is. That will be your background value
6. **IDL Window:** Open IDL with “idl” in another terminal window and run “idl\_script1\_generaluse,tobsid,oobsid,te,oe1,oe2,oe3,xcoor\_oe1,ycoor\_oe1,xcoor\_oe2,ycoor\_oe2,xcoor\_oe3,ycoor\_oe3,xcoor\_te1,ycoor\_te1,xcoor\_te2,ycoor\_te2,oback,tback” (info can be gathered, put into the excel file “SNInfo.xlsx”, and pasted into a .txt file named ‘TRUVOTcommands.txt’ located in ‘11SPRING2023/truvot-master’ folder)
   1. idl\_script1\_generaluse: I had to tailor this script to be able to run generally.
      1. Added more inputs after the codes file name: Template obsid, Data obsid, Template Extensions (1 or 2), 1st Data Extension, 2nd Data Extension (0 if there isn’t one), 3rd Data Extension (0 if there isn’t one), Data X Anchor Ext 1, Data Y Anchor Ext 1, Data X Anchor Ext 2 (0 if there isn’t one), Data Y Anchor Ext 2 (0 if there isn’t one), Data X Anchor Ext 3 (0 if there isn’t one), Data Y Anchor Ext 3 (0 if there isn’t one), Template X Anchor Ext 1, Template Y Anchor Ext 1, Template X Anchor Ext 2 (0 if there isn’t one), Template Y Anchor Ext 2 (0 if there isn’t one), Data Background Value, Template Background Value
      2. I had to also edit the code to accept that input as a string and then tailor the file name code portion to accept the generalized targetIDs.
      3. Had to comment out the “conmake” portion within “optimal\_shift\_time” because it was throwing the error “CONTOUR: Contour levels must be in increasing order.”
      4. Played around with deleting extensions with “fdelhdu ./00030390007/uvot/image/sw00030390007ugu\_rw.img\[1\]” for all dt, rw, and sk images (kept the extension with the longest exposure time). However, it wasn’t the problem, so I ended up redownloading that targetID
      5. Within optimal\_shift\_time.pro
         1. Added in “,exten\_no=1” on line 119 to fix the 1 ext template problem
         2. Changed it to “,exten\_no=0” on line 119 to fix the coadded template image problem
         3. Added in “addext” into its input as well as into the explanation AND changed the above stuff to “,exten\_no=addext” instead
      6. Added if statement based on number of extensions for data
7. **Python Window:** Open Python with “ipython --pylab” in another terminal window and run “import uvotpy\_script\_generaluse as usg” and “usg.truvot(targetid,ra,dec,exts,tw)”
   1. Edited the source file within UVOTPY “uvotgrism.py” to change the 3 tabs to spaces instead (9 spaces total) because that formatting changed from Python 2 to 3
   2. uvotpy\_script\_generaluse: I had to tailor this script to be able to run generally.
      1. Added more inputs after the codes file name: Data obsid, ra, dec, exts
      2. Had to edit code to get it to take ra and dec as floats
      3. Make sure that the target id has quotes around it to force it to read in as a string
      4. Had to put something in my zprofile to get the PYTHONPATH to look in the example directory to find the uvotpy\_script\_generaluse.py module
      5. Had to change all the hardcoded targetids to the generalized input targetid
      6. Had to include an if statement based on if it is g.pha or f.pha file at the very end
      7. Had to add exts and corresponding if statements to account for if there is only 1 extension or multiple
      8. Had to include if statements based on if it is a mix of g.pha or f.pha files with the existing if statements at the end (only did it at the end though, not the beginning, so I might have to do it at the start too)
      9. I ended up having to add the other if statements too lol. All fixed ☺
      10. Also added an input to allow for a change in trackwidth
      11. Switched from uvotgrism to uvotgetspec at Paul’s request because uvotgrism didn’t have an allowance to change the trackwidth
8. **Back to the IDL Window:** Go back to your IDL window and run “idl\_script2\_generaluse, oobsid, oe”
   1. uvotpy\_script\_generaluse: I had to tailor this script to be able to run generally.
      1. Added more inputs after the codes file name: Data obsid, # of Data Extensions
      2. I had to also edit the code to accept that input as a string and then tailor the file name code portion to accept the generalized targetIDs.
      3. Added if statements based on whether there are 1, 2, or 3 Data Extensions

**Definitions**

IDL tricks:

doc\_library, “\_codename\_”

.run \_codename\_

.compile \_codename\_