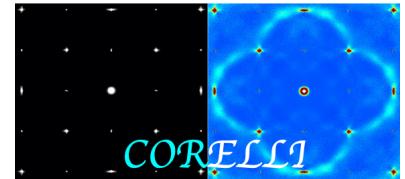


Instruction for reciprocal space coverage simulations for CORELLI

Yaohua Liu @ May 2019, liuyh@ornl.gov



This shows how to use a vanadium dataset to simulate the reciprocal space coverage of a single crystal sample in units of reduced reciprocal lattice constants. Vanadium is a nearly isotropic scatter, therefore, the simulated intensity also provides relative neutron flux information. At the same time, there is an option to predict the covered Bragg peaks: the peak position on the detector, as well as the Bragg scattering condition (incident neutron wavelength).

Input:

1. A UB matrix.
2. (optional) An updated instrument definition file.
3. (optional) A detector mask file.

User-defined parameters:

1. A list of rotation angles.
2. Three vectors for projection directions in the reciprocal space (HKL) in units of reciprocal lattice constants, and the calculation ranges.
3. (optional) The d-spacing range to calculate covered Bragg peaks information.

Output:

1. A MD file shows the reciprocal space coverage.
2. A peactable shows the covered Bragg peaks.

Python Scripts

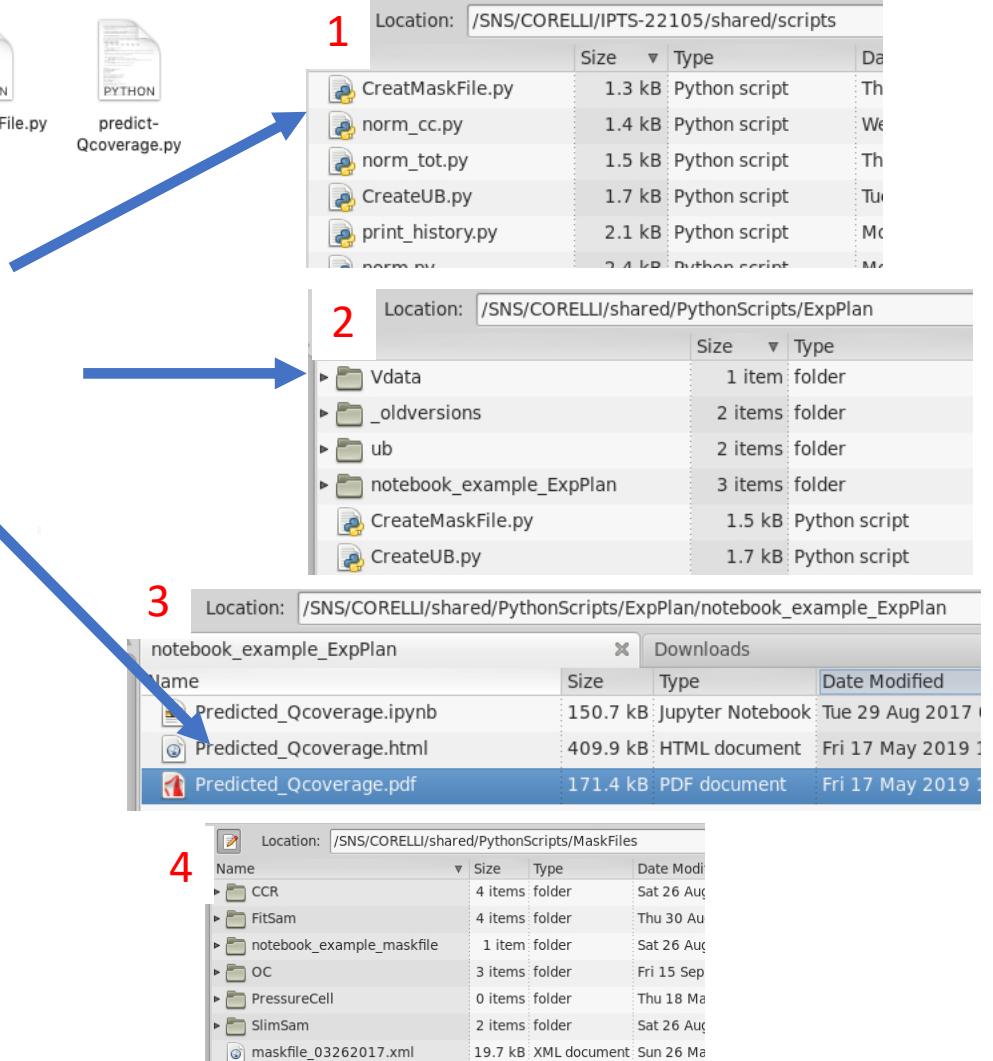


There are **three python scripts** that are useful for this purpose.

1. Normally you can find a copy in the folder
"/SNS/CORELLI/IPTS-xxxxx/shared/scripts/".
2. If not, there is a copy saved at
"/SNS/CORELLI/shared/PythonScripts/ExpPlan/".
3. There is a python notebook (as well as a html file and a PDF file with the same content) of an old version.

Other files:

- The Vanadium file is saved in the default folder.
- 4. There are default mask files saved in
"/SNS/CORELLI/shared/PythonScripts/MaskFiles/". You can just copy and paste to your working folder if you do not want to generate your own.
- Mantid has included a copy of instrument definition file that is sufficiently good for planning purpose.



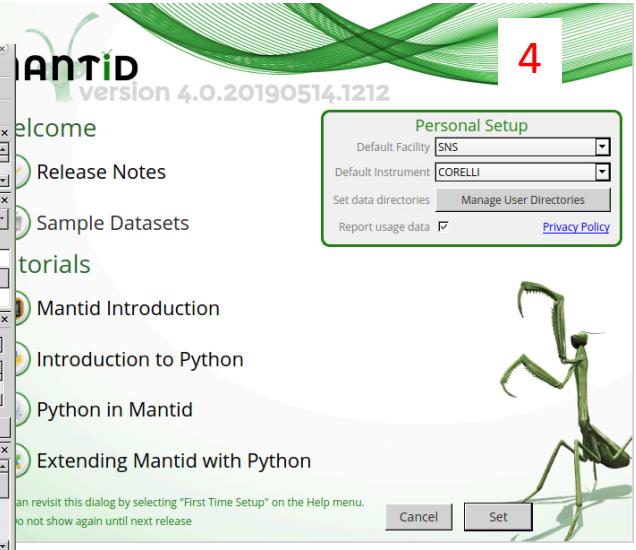
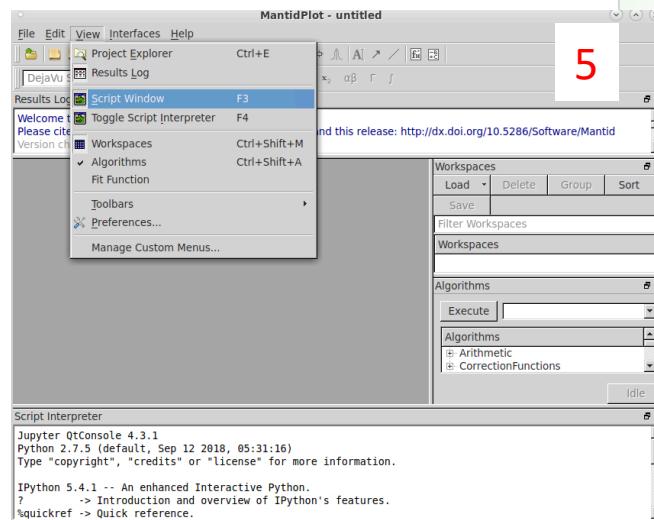
Get it started

1. Log into SNS analysis cluster using your UCAMS account.
2. Mantidplot is not required to run the script, but it is simpler if we use it.
3. You can launch “mantidplotnightly” on the SNS analysis cluster via Terminal.
4. You can select CORELLI as your default instrument, but it is not critical.
5. Launch the script window by clicking “F3” or selecting it from the menu.
6. Select the script that you want to run.



The image shows a 'MantidPlot: Python Window' with the number '6' in a red box. The window contains a script editor with the following code:

```
6 Execute Window Help
coverage.py x
An external IDF file becomes an option rather than required.
35
36
37
38 For more information, please contact one of the beamline staff.
39 ""
40
41 import sys,os
42 sys.path.append("/opt/mantidnightly/bin")
43 from mantid.simpleapi import *
44 from mantid import logger
45 import numpy as np
46 np.seterr("ignore")
47
48 iptsfd = '/SNS/CORELLI/IPTS-20712/'
49
50 # Provide necessary information on the data here.
51 outputdir = iptsfd + 'shared/'
52 UBfile = outputdir + 'optub.mat'
53 outputfile = outputdir + 'QCal.nxs'
54
55 # define the angles for virtual runs
56 Omegas = range(22, 59, 4)
57 totalrun = len(Omegas)
58 print "Total number of runs %d" %totalrun
59
60 #define the projected orientation in the HKL space
61 proj='1 1 A 1 A 1 A 1 A 1 1'
```



Predict-Qcoverage.py

Lines 41-46: No change needed.

```
41 import sys,os
42 sys.path.append("/opt/mantidnightly/bin")
43 from mantid.simpleapi import *
44 from mantid import logger
45 import numpy as np
46 np.seterr("ignore")
47
48
49 # Provide necessary information on the data here.
50 outputdir = '/SNS/CORELLI/shared/PythonScripts/ExpPlan/'
51 UBfile = outputdir + 'ub/test_ub.mat' ←
52 outputfile = outputdir + "Predicted_QSpace.nxs"
53
54 # define the angles for virtual runs
55 Omegas = range(0, 181, 10)
56 totalrun = len(Omegas)
57 print "Total number of runs %d" %totalrun
58
59 #define the projected orientation in the HKL space
60 proj=['1,-1,0', '0,0,1', '1,1,0']
61 Qlims = ['-10.1,-10.1,-10.1', '10.1,10.1,10.1']
62
63 #define IDF file. Ask instrument scientists for the updated information
64 IDFfile = '' # using default
65 #IDFfile = '/SNS/CORELLI/shared/PythonScripts/IDF/CORELLI_Definition_910.xml' # newly calibrated. April, 2017
66
67 #toggle the function to PredictPeaks. If turns on, need to update the Dmin and Dmax. This part is slow if dmin is smaller than 1.
68 predictpeaks = 0 # 1 to turn on Predictpeaks.
69 dRange = [1.5, 10.0]
70
71 #define the SE
72 SE = "" # "CCR", "OC", "SlimSam"
73 if SE in ["", "CCR"]:
74     mask = LoadMask(Instrument='CORELLI', InputFile='/SNS/CORELLI/shared/PythonScripts/MaskFiles/CCR/maskfile_03262017.xml')
75 elif SE == "OC":
76     mask = LoadMask(Instrument='CORELLI', InputFile='/SNS/CORELLI/shared/PythonScripts/MaskFile_20170221.xml')
77 elif SE == "SlimSam":
78     mask = LoadMask(Instrument='CORELLI', InputFile='/SNS/CORELLI/shared/PythonScripts/MaskFiles/SlimSam/SlimSam.xml')
79 MaskDetectors(Workspace=vdata, MaskedWorkspace=mask)
80
```

Lines 49-80:
User-defined parameters

It is shown later how to
generate the ub and the
mask files.

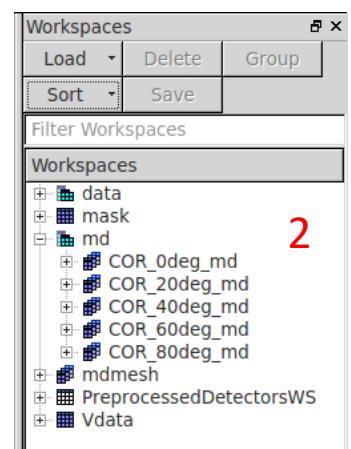
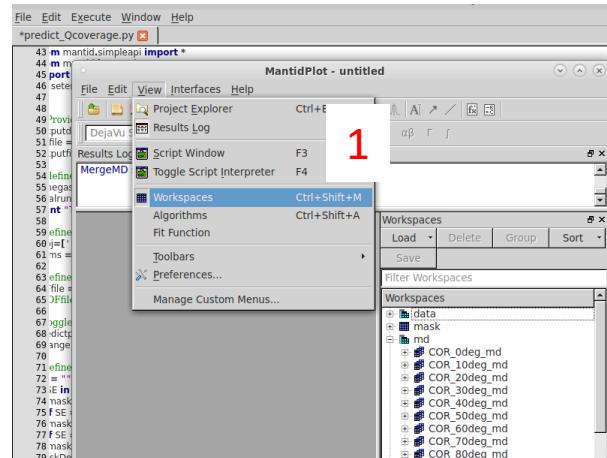
Lines 80-108 :
No change
needed.

```
79 MaskDetectorWorkspace=Vdata,MaskEventWorkspace=mdsR
80
81 # One Vandium Run is used to load the flux and detector information. No need to change.
82 filename='/SNS/CORELLI/shared/PythonScripts/ExpPlan/Vdata/COR_31663.nxs.h5'
83 Vdata = LoadEventNexus(Filename=filename, FilterByTimeStop=120)
84
85 # Load the data and convert to Q space for mesh plot and peak finding.
86 toMerge1=[]
87 for index, Omega in enumerate(Omegas):
88     print "Run %d of %d, Processing converting MD for run : %s" %(index+1, totalrun, Omega)
89     ows='COR_'+str(Omega)+'.deg'
90     toMerge1.append(ows)
91     print 'Omega = %5.2f deg.' %(Omega)
92
93 CloneWorkspace(InputWorkspace= Vdata, OutputWorkspace=ows)
94 if len(IDFile) > 0:
95     LoadInstrument(Workspace= ows, Filename=IDFile, RewriteSpectraMap=False)
96 SetGoniometer(ows,Axis0=str(Omega)+',0,1,0,1')
97
98
99 data = GroupWorkspaces(toMerge1)
100 LoadSawUB(InputWorkspace=data,Filename=UBfile)
101 ConvertToMD(InputWorkspace=data,OutputWorkspace='md',QDimensions='3D',dEAnalysisMode='Elastic', Q3DFrames='HKL',Uproj= proj[0],Vproj= proj[1],Wproj= proj[2],
102 QConversionScales='HKL',LorentzCorrection='0', MinValues=Qlims[0],MaxValues=Qlims[1])
103 mdmesh=MergeMD('md')
104
105 if predictpeaks == 1:
106     PredictPeaks(InputWorkspace=data, WavelengthMin=0.8, WavelengthMax=2.9, MinDSpacing=dRange[0], MaxDSpacing=dRange[1], OutputWorkspace='peaks')
107
108 SaveMD('mdmesh',Filename=outputfile)
109
110
```

It takes some time to save the md file. If you are using mantidplot, you can select not to save it by adding a "#" at the beginning of line 108.

You can view generated files in the window called "Workspaces".

1. The "Workspace" window can be activated by selecting it from the menu.
2. Some files are grouped into folders. You need to click it to view them individually.

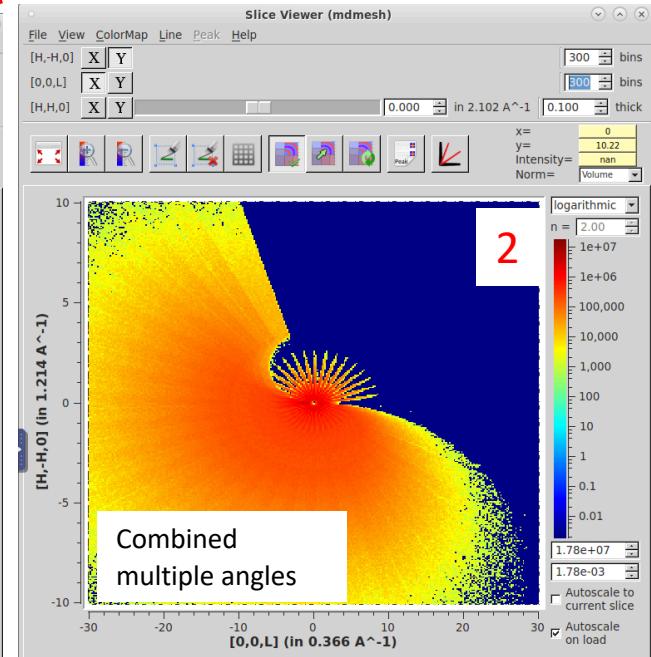
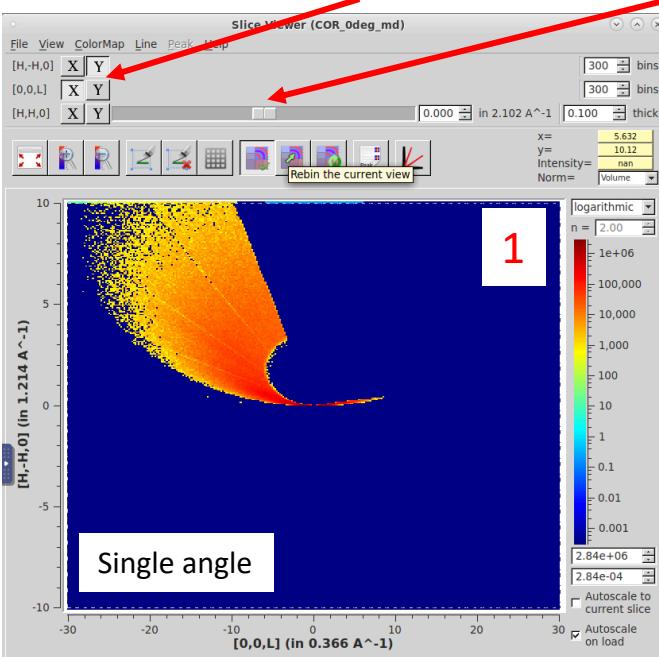
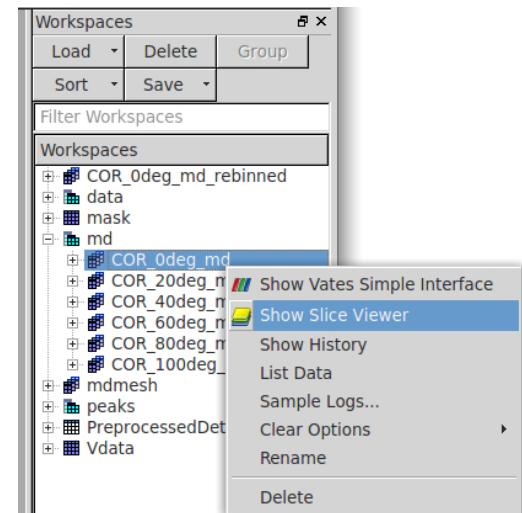


Q-coverage view

The generate Q-coverage is a 3D dataset. You can use “Slice Viewer” to look a selected 2D cross section.

1. Coverage of an individual angle is saved in the “md” folder.
2. The full coverage of the defined angle angel is saved as the “mdmesh” file.

You can select the projection direction and position by clicking these buttons and scrolling the bar.

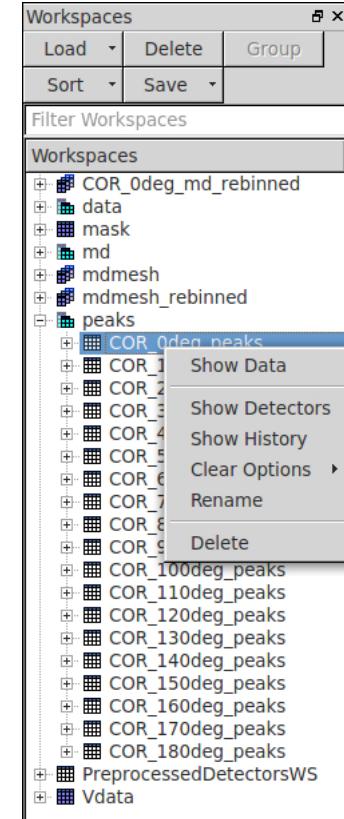


Peak prediction

```
67 #toggle the function to PredictPeaks. If turns on, need to update the Dmin and Dmax. This part is slow if dmin is smaller than 1.
68 predictpeaks = 1 # 1 to turn on Predictpeaks.
69 dRange = [1.5, 10.0]
70
```

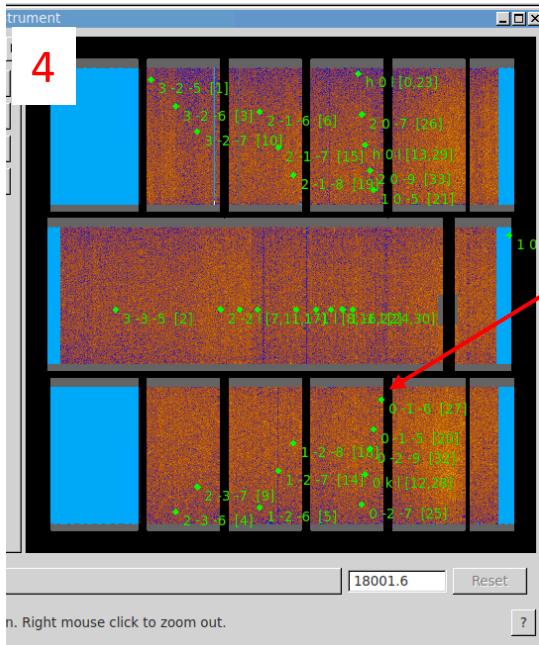
- If peak prediction is enable, a folder will be created including peaktables for each individual rotation angle.
 - The peaktables contains information on the HKL index, incident neutron wavelength (and correspondingly the energy and TOF), and the dspacing.
 - CORELLI has a neutron flux peaked around 1.3 Å. Please use this information to plan for order-parameter type experiment by selecting the sample rotation angle, if the scattering signal is weak.

	Å	meV	μs	Å					
unNumber[Y]	DetID[Y]	h[Y]	k[Y]	l[Y]	Wavelength[Y]	Energy[Y]	TOF[Y]	DSpacing[Y]	
1	31663	38728	2.00	-3.00	-7.00	2.2597941629787845	16.019090319512355	12916.127318937411	1.5303329459475785
2	31663	53541	1.00	-2.00	-6.00	2.4759293058459111	13.344402550054415	14180.905836871652	2.0667628732120531
3	31663	58979	1.00	-2.00	-7.00	2.0833947651062585	18.846576772124664	11891.005463008043	1.8993856041352954
4	31663	63378	1.00	-2.00	-8.00	1.7612139591825118	26.372504625340987	10030.066294907028	1.7453600797622837
5	31663	80990	0.00	-2.00	-8.00	1.0545581464235421	73.558786017836653	6020.6207891123713	1.6473518999294732
6	31663	70014	0.00	-2.00	-7.00	1.0545581464235421	73.558786017836653	6020.6207891123713	1.6473518999294732

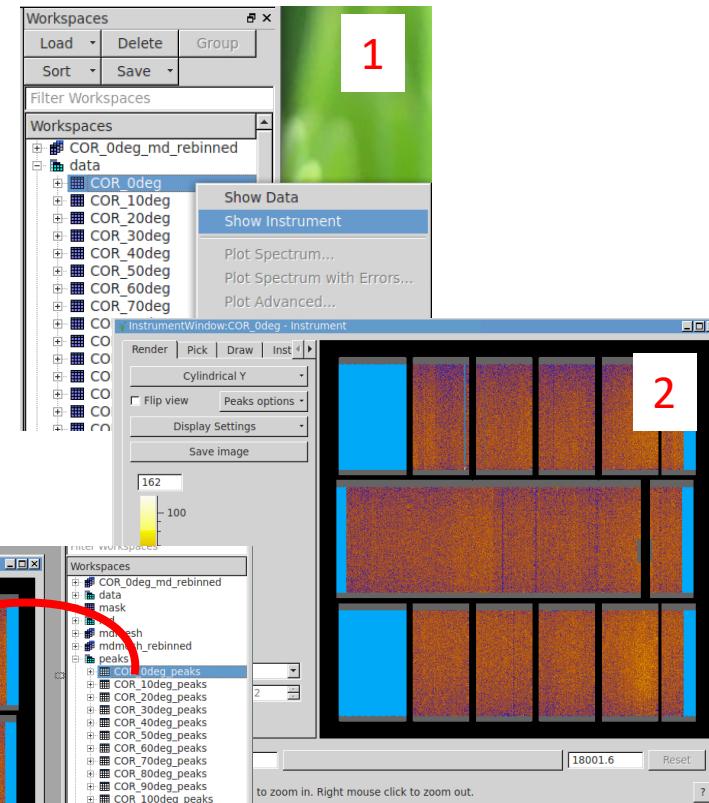
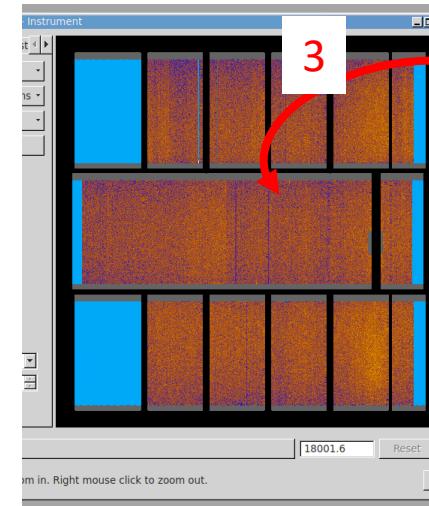


View peaks on detector

1. Select a data file and show the instrument view.
2. There will be no feature since it is vanadium data. However, you can see clearly the detector coverage. The DB and the edge of the detector have been masked.
3. Selecting a peaktable file and drag it in the instrument view window.
4. It shows the peak positions on the detector.
 - If you are working on a particular peak, check it on detector view to make sure it is not close to the detector edge. E.g. (0 -1 -6) peak is too close to the gap and it is better to rotate the sample by $\sim +0.5$ deg. if this peak is critical.



(0 -1 -6)
peak is too
close to the
gap.



to zoom in. Right mouse click to zoom out.

CreateUB.py

Lines 20-25: No change needed.

```
23 import sys,os
24 sys.path.append("/opt/mantidnightly/bin")
25 from mantid.simpleapi import *
26 from mantid import logger
27 import numpy as np
28 np.seterr("ignore")
29
30
31 # Provide necessary information
32 outputdir = '/SNS/CORELLI/shared/PythonScripts/ExpPlan/'
33 UBfile = outputdir + "ub/test_ub.mat"
34
35 # create a workspace (or you can load one)
36 ws=CreateSingleValuedWorkspace(5)
37
38 #set a UB matrix using the vector along k_i as 1,1,0, and the 0,0,1 vector in the horizontal plane, and the rotation axis is 1, -1, 0
39 # u is the vector along K_i when goniometer is at 0; and v is in-plane vector perpendicular to k_i, when goniometer is at 0
40 SetUB(ws,a=5.978,b=5.978,c=17.17,alpha=90, beta=90, gamma=120, u="1,1,0", v="0,0,1")
41 UB = ws.sample().getOrientedLattice().getUB()
42 SavelsawUB(InputWorkspace=ws, Filename=UBfile)
```

Lines 28-40:
User-defined
parameters

Sample specific information is required.

Typically, we can rotate the sample a lot in the horizontal plane, therefore, the vertical direction is most critical one when we plan CORELLI experiments, i.e., the direction perpendicular to the plane constructed by the vectors of u and v.

CreateMaskFile.py

There are default mask files can be found in
"/SNS/CORELLI/shared/PythonScripts/MaskFiles/".

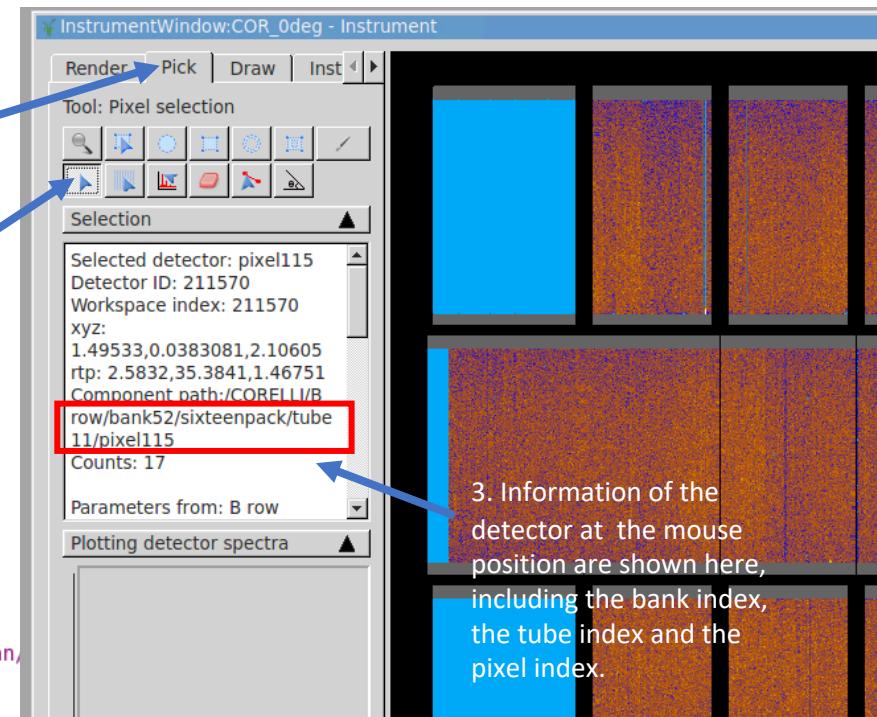
In case of that you want to update it yourself, you can use the
"CreateMaskFile.py". The detector information can be easily view
using the instrument view via Mantidplot.

Lines 20-25: No
change needed.

```
20 import sys,os
21 sys.path.append("/opt/mantidnightly/bin")
22 from mantid.simpleapi import *
23 from mantid import logger
24 import numpy as np
25 np.seterr("ignore")
26
27 # Please provide necessary information
28 outputdir = '/SNS/CORELLI/shared/PythonScripts/ExpPlan'
29 MaskFile = outputdir + "maskfile/SlimSam2.xml"
30
31 COR = LoadEmptyInstrument(InstrumentName='CORELLI')
32 # For SlimSam
33 MaskBTP(Workspace=COR, Bank="1-30,62-91") # out-of-range modules in the A and C rows
34 #MaskBTP(Workspace=COR, Pixel="1-16,195-256") # detector edges
35 MaskBTP(Workspace=COR, Pixel="1-10,200-256") # detector edges
36 # -- DB
37 MaskBTP(Workspace=COR, Bank="58", Tube="13-16", Pixel="80-130") #DB
38 MaskBTP(Workspace=COR, Bank="59", Tube="1-4", Pixel="80-130") #DB
39
40 SaveMask(COR, outputFile = MaskFile)
41
42
```

Lines 28-40:
User-defined parameters

Select these
options



You can add information of
detector regions to be
masked by using "MaskBTP"