

EmissionTomographyBackground.java

```
1 package sample.control;
2
3 import com.mimvista.external.control.XMimEntryPoint;
4 import com.mimvista.external.control.XMimSession;
5 import com.mimvista.external.data.XMimMutableNDArray;
6 import com.mimvista.external.points.XMimNoxelPointI;
7 import com.mimvista.external.series.XMimImage;
8 import com.mimvista.external.series.XMimMutableImage;
9 import com.mimvista.external.series.XMimSeriesView;
10
11 import javax.swing.JOptionPane;
12
13 /**
14  * @author dezarnwa
15  *
16  * Main entry point for EmissionTomographyBackground MIM extension
17  */
18 public class EmissionTomographyBackground {
19
20     private static final String desc = "Subtract constant background
21     values from emission image.";
22
23     /*
24      * @XMimEntryPoint tells MIM that this function can be used to
25      * start a MIMextension
26      * Meta-data entered here will display in the extension
27      * launcher inside MIM
28      */
29     @XMimEntryPoint(
30         name="Emission Tomography Background",
31         author="William A Dezarn",
32         category="Utility",
33         description=desc,
34         institution="WFBH",
35         version="1.0",
36         outputTypes={XMimImage.class})
37     public static Object[] runOnSession(XMimSession session, XMimImage
38     image) {
39         // Run the code
40         XMimImage result = process(session, image);
41     }
42 }
```

EmissionTomographyBackground.java

```
38     //Once we finish processing the result, add it to the session so
it can be displayed
39     XMimSeriesView view = session.addImageAndReturnView(result,
"Bkgd Subtracted");
40
41     //Our @XMimEntryPoint says we return one XMimImage object, so do
that here
42     //This allows our MIMextension to integrate with workflows
43     return new XMimSeriesView[] {view};
44 }
45
46 public static XMimImage process(XMimSession session, XMimImage
image) {
47     if(image.getRawData().getDimensionCount() < 3) {
48         session.createLogger().error("Insufficient dimensions;
requires a volumetric image");
49         return null;
50     }
51
52     // MIMextensions don't allow a volume to be modified in place
53     // So, create a mutable copy so we can edit it
54     XMimMutableImage mutable = image.getMutableCopy();
55     // getScaledData() Provides the scaled data array of this image.
Want BQML for PET, so use scaled.
56     XMimMutableNDArray CountData = mutable.getScaledData();
57     int[] dimensions = CountData.getDims();
58
59
60     // Ask for background
61     String EntryString = JOptionPane.showInputDialog("Enter
background level.");
62     Double Bkgd = Double.parseDouble(EntryString);
63
64     // Following section subtracts background from image counts or
sets to zero if less than background
65     XMimNoxelPointI point = image.createNoxelPointI(); // voxel
coordinates used to get data
66     // Loop over entire image to subtract background and initialize
dose array
67     for(int z=0; z<dimensions[2]; z++) {
68         point.setCoord(2, z);
```

EmissionTomographyBackground.java

```
69         for(int y=0; y<dimensions[1]; y++) {
70             point.setCoord(1, y);
71             for(int x=0; x<dimensions[0]; x++) {
72                 point.setCoord(0, x);
73
74                 Double value = 1.0 *
75                 CountData.getDoubleValue(point); // get image counts
76                 if (value <= 1.0*Bkgd) { //
77                     subtract background as needed
78                     CountData.setValue(point, 0.0);
79                 } else {
80                     CountData.setValue(point, (value-1.0*Bkgd));
81                 }
82             }
83         }
84
85         //JOptionPane.showMessageDialog(null, "Activity map created.
86         TotCount = " + TotCount.toString());
87         return mutable;
88     }
89 }
90
```