

## **Phase 1**

### **Spatial analysis based on dysfunctional pixel locations Can be used for both dead pixels and other categories of dysfunctional pixels**

#### **phase1\_PixelReading.R:**

Reads raw data about damaged pixels from a variety of formats to define mask of dead pixels (Tbin) and coordinates of dead pixels (coo.dead) and check if they are consistent with the given Layout (i.e. inside the detector and inside the modules)

WORKFLOW: Assumes already run  
- phase0\_Layout.R

INPUT: Layout (object), raw data (currently works for Pilatus (dead pixel mask as tif), To Do: other formats)

OUTPUT: coo.dead (matrix)

GLOBAL FUNCTIONS: inconsistency.dead.layout()

#### **phase1\_LogisticRegression.R**

Fits logistic regression models with a variety of factors.

WORKFLOW: Assumes already run  
- phase0\_LayoutPixel.R  
- phase1\_PixelReading.R, including consistency checks there

INPUT:  
- Tbin (dead pixel mask built in phase1\_ReadingPixels),  
- LayoutPixel (object built in Phase0 for suitable detector type)

OUTPUT:  
- text file "modelResults\_....txt" with summary(model) for a variety of models

#### **phase1\_CountsAngles.R**

Descriptive analysis of damaged pixels including plots of layout with damaged pixels added, counts of damaged pixels.

WORKFLOW: Assumes already run (including auxiliary functions)  
- phase0\_LayoutExamples.R (for matching Layout example)  
- phase0\_LayoutVisualisation.R  
- phase0\_LayoutPixel.R (do not run the plots again (to save time))

- phase1\_ReadingPixels.R, including consistency checks there

#### INPUT:

- coo.dead (dead pixel coordinates built phase1\_ReadingPixels)
- ppp.edges.col, ppp.edges.row
- Layout (object built in Phase0 for suitable detector type)
- auxiliary variables for layout (obtained during workflow given above)

#### OUTPUT:

- Plot of damaged pixel locations with underlying detector layout
- Plots of layout with damaged pixel counts per module
- Rose plots of NN (nearest neighbour) oriented angles for full detector and for individual modules
- Damaged pixel locations with NN oriented arrows
- Plots comparing DIY and spatstat package function of the previous two (for debugging, see below)
- Text file "counts\_....txt" with counts in all modules, Chi-square test results, Layout parameters etc.

### **phase1\_KFG.R**

#### Spatial

WORKFLOW: Assumes already run (including auxiliary functions)

- phase0\_LayoutExamples.R (for matching Layout example)
- phase0\_LayoutVisualisation.R
- phase0\_LayoutPixel.R (do not run the plots again, though)
- phase1\_ReadingPixels.R, including consistency checks there

#### INPUT:

- coo.dead.module (dead pixel coordinates with module indices built phase1\_ReadingPixels)
- ppp.edges.col, ppp.edges.row
- Layout (object built in Phase0 for suitable detector type)
- auxiliary variables for layout

#### OUTPUT:

- individual plots of K, F, G functions
- plot of all the above on one page along with dead pixels shown on detector layout (for reference)
- inhomogeneous versions of the above