2021 CERN Summer Student Report: Module Performance Evolution During Integration Steps

Tsz Hong Kwok[1]
thkwokae@connect.ust.hk
tsz.hong.kwok@cern.ch

Supervisors: Silke Möbius[2], Dr. Benedikt Vormwald[3]

[1] *The Hong Kong University of Science and Technology*
[2] *Georg-August-Universität Göttingen*
[3] *CERN*

August 26, 2021

Abstract

The High-Luminosity Large Hadron Collider (HL-LHC) is expected to be operated after 2027. It will have around 10 times higher luminosity than the current LHC. Therefore, a detector upgrade is necessary in order to deal with the increased particle, data rate, and backgrounds radiation. Especially the inner detector needs to be upgraded, which is the nearest detector component to the interaction point. The upgrade of the ATLAS detector foresees to completely replace it by an all-silicon inner tracker (ITk). During the construction, each pixel module will be evaluated at different assembly steps. This report introduces a systemic tool to compare each assembly stage in order to be able to spot potential degradation, by comparing various data from quality control (QC) tests at each stage.

1 Introduction

The High-Luminosity LHC (HL-LHC) is expected to be operated after 2027, which aims to achieve a peak luminosity of about 5×10^{34} cm^{-2}s^{-1}. Comparing to the statistics recorded in 2018, where the peak luminosity was around 2×10^{34} cm^{-2}s^{-1}, this is increased by around a factor of 2.5. The ultimate goal is to achieve a total luminosity of 4000 fb^{-1}, which is about one order of magnitude larger than what the current LHC is aiming at. However, the increased luminosity will increase the pile-up to about 200 on average. Besides, the higher luminosity will also increase the background radiation significantly, for instance to around 20 MGy total ionization dose. In other words, it is necessary to upgrade the detector to deal with the increased event density and improve the radiation hardness: especially the inner detector, that is closest to the interaction point and suffers the most from the large luminosity.

This report will focus on the ALTAS detector tracker upgrade. It is planned to be completely replaced by an all-silicon inner tracker (ITk), which consists of 5 pixel layers and 4 strip layers and thus features an increased coverage to $|\eta| = 4$. The pixel detector layers can be divided in different subsystems: the end caps, the inner barrel and the outer barrel. Each layer of the outer barrel is formed by two components, flat longerons in the center and inclined rings in the forward region of the outer barrel. They will be the mechanical supports of the modules. The module is formed by a sensor which is bump-bonded on four front-end (FE) chips. The RD53A FE chip is used in this report as a demonstrator, which consists of three types of analog FE technologies, namely Synchronous ('syn'), Linear ('lin') and Differential ('diff'). In the later steps of the ITk project, the final version will not be using the RD53A FE chip, but a FE chip that is only made of one of the three analog FE technologies. In the construction, there are three different assembly steps for the module before it ends on the support. At each step, the module should go through a series of quality control (QC) tests. The first QC test will be taken after the module is built. The second test will be done when the module is placed on a mechanical cell. The last test will be taken when the modules are placed into the local support, namely the flat longeron or the inclined ring.

It is important to keep track of the performance change from each assembly step to the next. This report introduces the tool for systemically analysing and spotting the potential degradation of the module at the three stages mentioned above. The tool automatically reads the corresponding data from the QC tests, and extracts meaningful quantities for comparing the evolution. The rest of the report will first discuss the data and the structure of the tool, and also the current features included. Then, some examples of the output will be shown and discussed. All information, including the tool, configuration files and plots can be found in the git repository [1].

2 Method

The pixel module will go through a series of QC tests, which characterise the performance of the module [2]. The QC test will be taken at each assembly step: from an individual module, to module on a mechanical cell, to the module on the local support. In order to

analyse the performance evolution, the tool will first automatically extract the corresponding QC test data sets, which are specified in the configuration file. Then, a series of resulting plots at different assembly stages and the comparison plots will be produced, and stored in the specified format. More specifically, the QC data used in this report is measured on the RD53A FE chips and the sensor of the modules [3]. However, not all the data at different assembly stages are available at the time of writing this report. Therefore, the plots in the following sections are using different datasets at one stage and duplicates, to serve as examples.

2.1 QC Data

The detailed set of QC tests can be found in [2]. However, in this report, only some of the suitable quantities for analysing the performance degradation will be used and discussed. They can be categorized into three types as listed below:

- 2D Histogram
  - Digital Occupancy Map: The map of the recorded number of hits at each pixel, after a test pulse injected after the discriminator. There is a known amount of injected test pulses to each pixel, the ideal and typically measured occupancy at each pixel would be the same as the injected amount.
  - Analog Occupancy Map: The map of the recorded number of hits at each pixel, after a test pulse injected before the discriminator. There is also a known amount of injections. However, there might be noisy or dead pixels, which will return a value larger or smaller than the known amount. The ideal case would also be the same amount as the injected number.
  - Noise Mask: The map of boolean value (0 or 1), which turns the pixel off if the pixel is noisy.
- 1D Histogram
  - Threshold Distribution: The distribution of the measured threshold of all the pixels in the chip. Each pixel's threshold will be tuned using internal DAC setting and then determined by measuring the turn-on behaviour for varied intensity of test pulses. Threshold is defined as the point which has 50% efficiency of detected hits.
  - Noise Distribution: The distribution of extracted noise of all the pixels in the chip. The noise value is determined by the widths of the turn-on behaviour. No noise would correspond to a step function.
- Graph
  - SLDO VI Curve: The voltage versus current curve of the Shunt-LDO. In the serial powering concept of the ITk pixel detector, the voltage required for the modules is generated in an Shunt-LDO current from a controlled/regulated current. The SLDO VI curve demonstrates the sanity of the power-providing part of the FE.

– Sensor IV Curve: The leakage current versus reverse bias voltage curve. In the ideal case, there is a sudden increase near 0 voltage, then the leakage current will increase as the bias voltage increases, with near 0 slope, until the break down.

Some of the above features will be plotted for direct comparison, while some of them will be used to extract meaningful values for comparing the evolution during different assembly steps. More details and examples will be shown and discussed in the later sections.

2.2 Configuration Files

Configuration files are used to make the tool more flexible and accessible. The complete version of the configuration files discussed in this section can be found in Appendix B, and also in the git repository [4]. The configuration is separated into two files, the General Configuration (`GenConfig.json`) in [5], and the Plot Configuration (`PlotConfig.json`) in [6]. The General Configuration file contains the module information, and the information of its four FE chips. Particularly, it includes first the module ID (`"ModuleID"`). Then there are three sub-definitions specified for the FE information, which contain the FE ID (`"ID"`), the FE types to be looked at (`"FE"`), and the data directories at different stages (`"FolderPath"`). Besides, the directory of the plotting configuration (`"PlotConfig"`) should also be specified, where all the plots and the parameters are stored. More details of the Plot Configuration will be shown in the later part of the report. Finally, the output format (`"OutputFormat"`) of the plots is defined. There are two options provided: `".png"` will store each plot into an individual png file, while `".pdf"` contains all the plots into one single pdf file as an overview. One of the general configuration file structures are shown below:

```
{
   "Module": {
      "ModuleID": "TBD",
      "FE1": {
         "ID": "20UPGFC0014902_1",
         "FE": [ "diff", "lin", "syn" ],
         "FolderPath": {
            "Stage1": "../Data/data-20UPGM20021168_1",
            "Stage2": "../Data/data-20UPGM20021168_2",
            "Stage3": "../Data/data-20UPGM20021168_3"
            }
      },
      "FE2": ...,
      "FE3": ...,
      "FE4": ...,
   },
  "PlotConfig": "./PlotConfig.json",
  "OutputFormat": [ ".png", ".pdf" ]
}
```

All the plotting parameters are stored in a separate Plot Configuration file. The plotting is categorized into 3 types as mentioned previously. First of all, the 2D histogram `"Hist2D"` example is shown below. For each type of a 2D histogram, the name (`"Name"`) of the plot needs to be specified, for reading the data file. Besides, there is an extra parameter, `"EvolutionPlot"`, which can produce the evolution plot throughout the three stages. It would be a plot with x-axis being the three stages, and y-axis being the variable that is kept track of. For instance, the default variable for 2D histogram is the number of dead pixels (i.e. pixel that has 0 occupancy). And there will be three distinct curves which represent the three different FE types ('syn', 'lin', 'diff'). Moreover, there are two options for the y-axis. They are `"Absolute"`, and `"Percentage"`, which represents the absolute number of pixels, and the percentage, respectively. Furthermore, for the analog occupancy map, there might be some noisy pixels which will widen the scale of the plot. Therefore, the extra `"Cut"` parameter is introduced to separate different cases so that we can see the details of the analog occupancy. In the example below, the cut is set to be 200. There will be 3 individual plot produced: Firstly, Occupancy $= 0$; Secondly, $0 <$ Occupancy < 200; And lastly, Occupancy ≥ 200. They correspondingly reflect the dead pixels, normal and inefficient pixels, and noisy pixels. The following shows an example of the configuration structure for plotting the 2D histograms:

```
"Hist2D": {
   "digitalscan": {
      "Name": "OccupancyMap",
      "EvolutionPlot": { "Y": "Absolute" }
    },
    "analogscan": {
      "Name": "OccupancyMap",
      "Cut": 200,
      "EvolutionPlot": { "Y": "Percentage" }
    },
    "noisescan": {
      "Name": "NoiseMask"
    }
}
```

Another type of plot is the 1D histogram `"Hist1D"`. First of all, similar to the 2D histogram, one of the parameters of each histogram type is its name, used to search for the data file. Since the 1D histograms at different stages can be plotted into the same figure for direct comparison, there is an extra parameter `"CompareType"` used for one-to-one comparison of two stages. There are two options, `"Ratio"` and `"Percentage"`, which represent the ratio x_n/x_{n-1} and percentage change $(x_n - x_{n-1})/x_{n-1}$ of the variables, respectively, where x_n is the value at stage n. By plotting the ratio or the percentage change, we can see the more detailed differences between the two distributions. However, calculating the ratio or percentage change is dependent on the bin width of the distributions. Therefore, the comparison plot will only be plotted when the bin widths are the same. Another extra parameter is the `"FitRange"`. It specifies the range for fitting a distribution curve of the

given 1D histogram. One of the reasons why we fit the 1D histogram is because it would be easier to compare a larger amount of distributions if we can extract some values from the distributions, and only compare those values. In this report, we fit the histogram by a Gaussian distribution and extract the mean value of the Gaussian. There are three options provided for determining the fitting range. Firstly, the default option would be using the full range, `"Full"`, in the raw histogram. Secondly, the automatic fitting range, `"Auto"`, which will be determined dynamically, with details listed in Appendix A. Lastly, since there might be unexpected distributions where the previous two methods fail to provide a satisfying fit. Therefore, the last option can be used to manually input the fitting range. Lastly, similar to the 2D histogram, there is also an optional parameter, `"EvolutionPlot"`, which will plot the evolution throughout the three stages. However, since there is no need to specify the y-axis, the expected input would be boolean. The example of the configuration file for the 1D histograms plotting are shown below:

```
"Hist1D": {
    "NoiseDist": {
      "Name": "NoiseDist",
      "CompareType": "Ratio",
      "FitRange": {
        "diff": {
          "Stage1": "Full",
          "Stage2": "Auto",
          "Stage3": [ 100, 150 ]
        },
        "lin": ...,
        "syn": ...
      },
      "EvolutionPlot": true
    },
    "ThresholdDist": {
      "Name": "ThresholdDist",
      "CompareType": "Percentage",
      "FitRange": ...,
      "EvolutionPlot": true
    }
```

The last type of plots are the graphs. The first type is the SLDO VI curve, which only requires the parameter `"Name"`. Another type of graph is the Sensor IV curve. Since the measurement of currents in a silicon sensor depends on the temperature, it is required to normalize different IV curves to the same reference temperature, T_{ref}, in order to be able to compare them properly. The reference temperature (in the unit of K) can be specified in the parameter `"Tref"`. Then, the measured current can be corrected by multiplying the

following factor:

$$\left(\frac{T_{\text{ref}}}{T}\right)^2 \exp\left(-\frac{E_{\text{eff}}}{2k_B}\left(\frac{1}{T_{\text{ref}}}-\frac{1}{T}\right)\right).$$

Similar to the 1D histogram, a meaningful value can be extracted from the IV curve, such as leakage current. To extract it, the extraction point (voltage) needs to be specified in the parameter `"FitPoint"` and also the fitting range, which is determined by `"FitStep"`. For instance, in the example below, the fitting range at stage 1 would be $(90-24, 90+24)$. Then a linear line will be fitted inside this range. The extracted leakage current would be the result when substituting the `"FitPoint"` into the fitted linear equation.

```
"Graph": {
    "VI": {
      "Name": "SLDO_VI"
    },
    "IV": {
      "Name": "SensorIV",
      "Tref": 298.15,
      "FitPoint": {
        "Stage1": 90,
        "Stage2": 100,
        "Stage3": 110
      },
      "FitStep": {
        "Stage1": 24,
        "Stage2": 25,
        "Stage3": 26
      }
    }
  }
```

To summarize the plotting configuration file, the overall structure is shown below. The first layer is the three categories of the plots. Inside these categories will be the types of the plot which contain their own specified parameters, as discussed above.

```
{
  "Hist2D": {
    "digitalscan": ...,
    "analogscan": ...,
    "noisescan": ...
  },
  "Hist1D": {
    "NoiseDist": ...,
    "ThresholdDist": ...
  },
```

```
  "Graph": {
    "VI": ...
    "IV": ...
  }
}
```

Furthermore, an additional tool is introduced to plot the overview of the modules in the last assembly stage where the modules are installed onto the local support. There is also a configuration file for the tool, Local Support Configuration (`LocalSupportConfig.json`), which can be found in [7], to organize the information and make it more accessible. The structure is similar to the `GenConfig` which includes the module and FE information. An example of the configuration file is shown below, where the structure of each module information is the same as in the `GenConfig.json`. But there is an extra parameter, `Layout`, which specifies the layout type of the local support. There are two major layouts, `"Longeron"` and `"Ring"`, which represent the Flat Longeron and the Inclined Ring parts, more details can be found in [8].

```
{
  "Layout": "Longeron",
  "Plot": "analogscan",
  "OutputFormat": [ ".png", ".pdf" ],
  "Module1": {
    "ModuleID": "TBD_Mod1",
    "FE1": {
      "ID": "20UPGFC0014902_1",
      "FE": [ "diff", "lin", "syn" ],
      "FolderPath": ...
    },
    "FE2": ...,
    "FE3": ...,
    "FE4": ...
  },
  "Module2": ...,
  "Module3": ...,
  ...
  }
}
```

3 Results

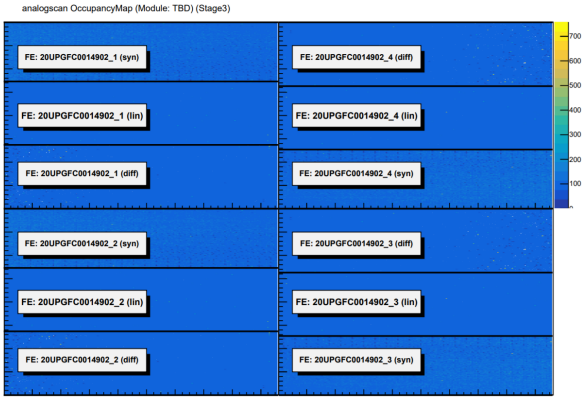
The resulting plots can be found in [9]. Some of the outputs will be shown and discussed in this section. The folder contains the file `"overview.pdf"` with all the desired plots. The

first plot on the front page is the Digital Occupancy Map at the last stage (if it is requested in the Plot Configuration file), as shown in Figure 1. This will serve as the overview of the module layout. First of all, the module ID is included in the title of the plot. Also the plot shows the quad layout of the module, which consists of four FEs, with their FE IDs printed on the plot. Each FE chip is made of three different readout technologies ('syn', 'lin', 'diff'). Their FE types are also printed on the corresponding location, and indicated by the border line.

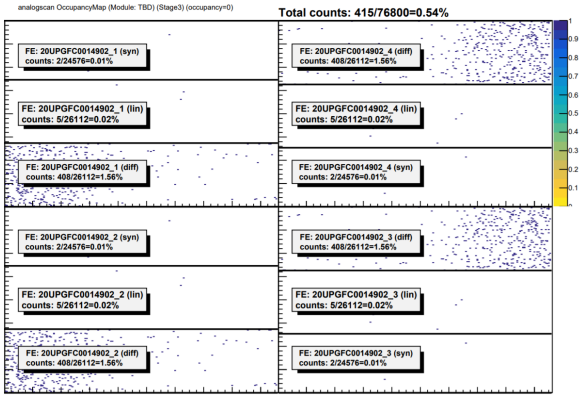
Figure 1: Example of the Digital Occupancy Map at stage 3.

The next example is the stage 3 Analog Occupancy Map shown in Figure 2. The original plot has the same layout as the Digital Occupancy Map, which contains the module and FE information, as shown in Figure 2a. However, due to the noisy pixels, the scaling will be pulled up to several hundreds, which makes it harder to see the details. Therefore, the `"Cut"` parameter was introduced to produce the other three plots in Figure 2. The maps of dead pixels, normal pixels, and noisy pixels are shown in Figure 2b, 2c, 2d, respectively. On top of the module and FE information, these plots also include the number of the pixels that have occupancy in the specified occupancy range (e.g. $0 <$ Occupancy < 200). The counts for each individual FE type in a chip, and also the total counts are included. By cutting the original plot into different ranges, it is easier and clearer to see more details on the chip, as shown in Figure 2c.

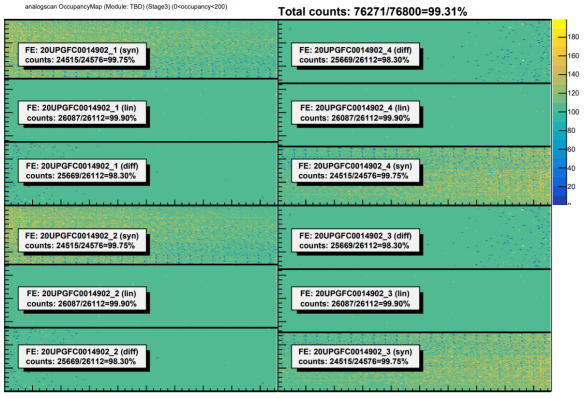
Furthermore, the pixel counts, of the specified occupancy range, of each FE type at different stages will be used to evaluate the pixel performance. An example of the noisy pixels (Occupancy ≥ 200) evolution plot is shown in Figure 3. There are four graphs, each representing the corresponding FE chip at the corresponding position (since this report is using the same dataset for testing, the graphs are identical). Each graph consists of different colors, which represent different FE types.



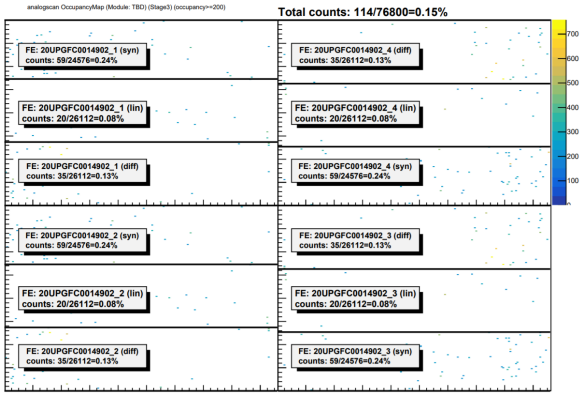
(a) The raw 2D histogram without any cut. The range of the occupancy is from 0 to around 800. This makes the details to be unclear.



(b) The 2D histogram of the dead pixels (i.e. Occupancy = 0). There are extra information printed in the plot, which include the counts for the whole module, and for each FE type in a chip.



(c) The 2D histogram of the normal and inefficient pixels (i.e. $0 <$ Occupancy < 200).



(d) The 2D histogram of the noisy pixels (i.e. Occupancy ≥ 200).

Figure 2: One FE chip example of Analog Occupancy Map at Stage 3.

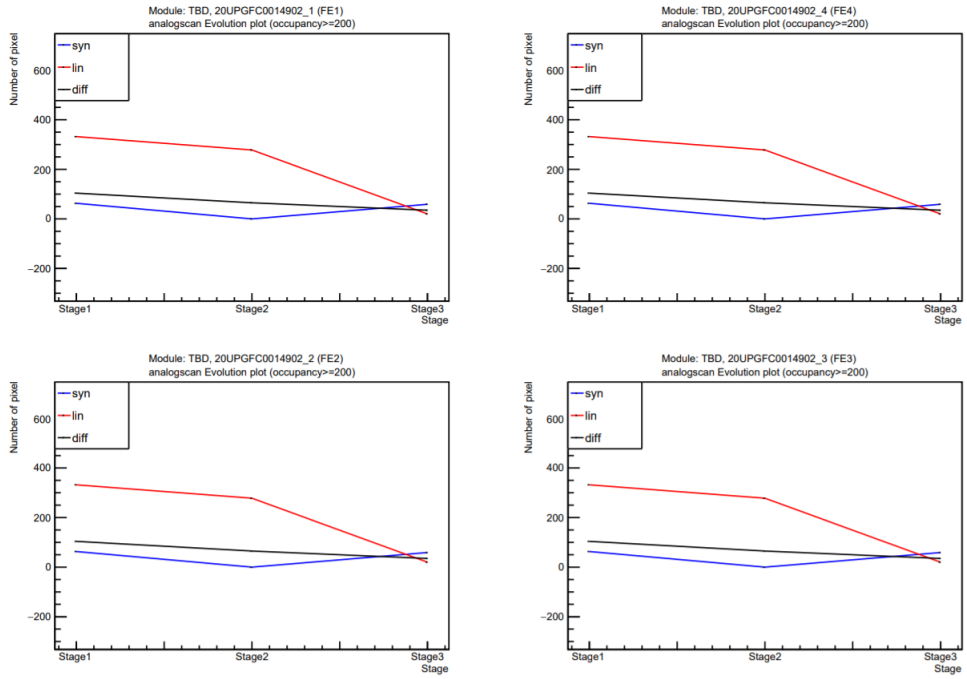


Figure 3: Evolution plot of the noisy pixel counts throughout the three stages. The four panels correspond to the different FE chips in the module. Different colors represent the corresponding type of FE. Since the dataset used for the four FE chips is the same, the four plots are identical.

The next category of plot is the 1D histogram. The first example shown in Figure 4 is the Threshold Distribution of the 'syn' FE. The upper panels are the individual plots at each stage. The red curve is the fitted Gaussian curve, with its fitting range and the fitted parameters printed on the plot, and also the χ^2/ndf to provide the goodness of the fit. The three fits also demonstrate the different fitting range methods introduced in the previous section. The stage 1 fit is using the default `"Full"` method, and the stage 2 fit is using the `"Auto"`, while the stage 3 is using the manually specified range `[800, 1000]`. The lower panels show the comparison plots. The first plot is the overall pure comparison of the three stages by overlapping them, while the second and the third plots are showing the one-to-one comparison of the two consecutive stages. The section of the plot below each one-to-one comparison is the percentage (or ratio) plot of the two stages. Similarly, Figure 5 shows the Noise Distribution plot. In this example, since the bin widths of different stages are different, there will be no percentage (or ratio) plot placed below.

Similar to the evolution plot in the 2D histogram case, the extracted mean value of the fitted Gaussian in the 1D histogram serves as the variable for the evolution plot, as shown in Figure 6, which is an example of the Threshold Distribution Evolution. Again, since the four FE chips are using the same dataset, the four plots are identical.

The last category is the graph, the first example is the SLDO VI curve, shown in Figure 7. The layout is similar to the 1D histogram plot so that the three panels represent the VI curves at the three different stages. Figure 8 shows the sensor IV curve. The upper panels are the individual plots of the IV curve at different stages. The red straight line plotted on the curve is the fitted linear line to extract the leakage current. Additionally, the details of the fitting

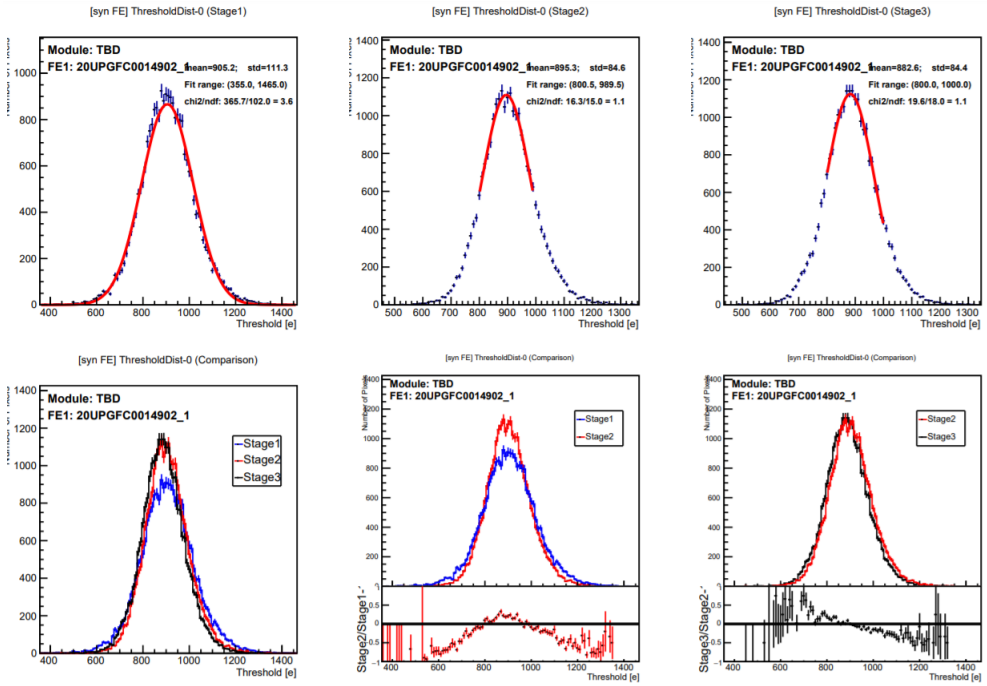


Figure 4: Example of the Threshold Distribution of the 'lin' FE in the first FE chip (FE1). The upper panels show the individual distribution and its Gaussian fit at the corresponding stage. The lower panels show the comparison of different stages. The last two plots include the percentage change plot below.

range, the fitted linear equation, the extracted leakage current, and the χ^2/ndf are printed on the plot. The plot below is the direct comparison of the three stages. The three colors represent the corresponding stages, indicated in the legend box.

Supplementary, the overview of the support structures are shown below. Figure 9 and 10 show the case of having 12 modules in a Flat Longeron layout and 11 modules in an Inclined Half Ring layout. The module IDs are printed in the corresponding location. Moreover, the first FE chip also indicates the locations of the three FE types. For both layouts, one overview with the whole structure is produced, as shown in Figure 9a and 10a. However, when there are many modules included in the structure, the details might become unclear. Therefore, the whole structure is broken into smaller parts. For instance, when the number of modules in the Longeron exceeds 6, it is separated into different pieces, with each of them having at most 6 modules, as shown in Figure 9b and 9c. Similarly for the Ring layout, when there are more than 4 modules, it will be separated into pieces which have at most 4 modules in it. The example can be found in Figure 10b, 10c and 10d.

4 Conclusion

To conclude, this report introduces a systemic tool for evaluating the performance comparison of the module during different assembly stages. In particular, different features are introduced to judge the performance of the module. These features are grouped into three main categories, the 2D Histograms, 1D histograms, and Graphs. Besides purely comparing these features at different stages, some representative quantities are extracted from the data

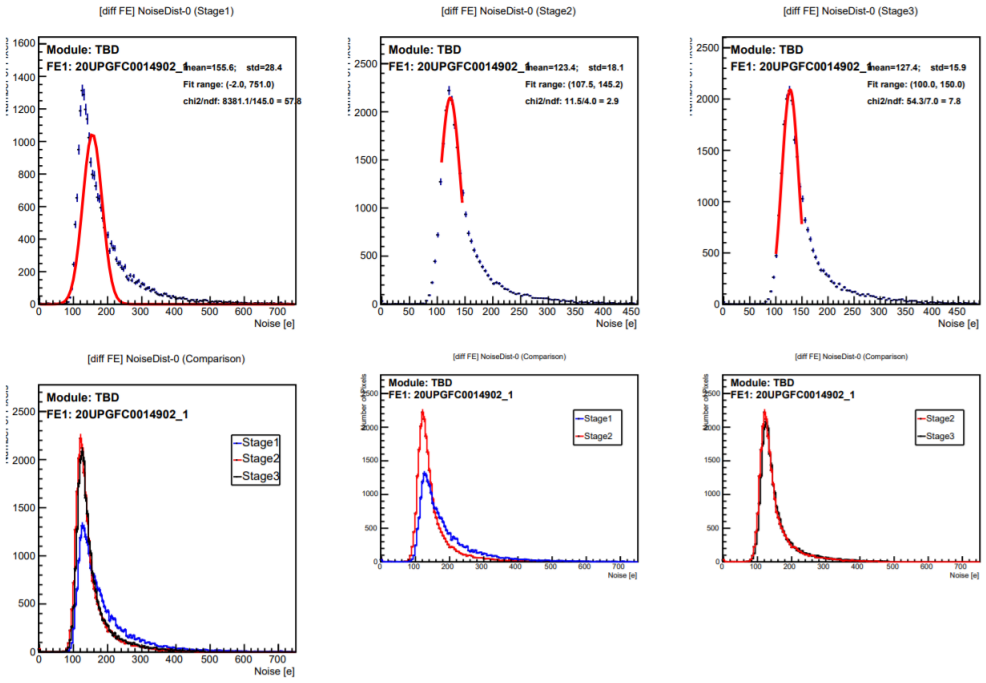


Figure 5: Example of the Noise Distribution of the 'diff' FE in the first FE chip (FE1). The upper panels also show the individual distribution and its Gaussian fit at the corresponding stage. The first panel at the bottom is the pure comparison of the three stages. The last two plot are the one-to-one comparisons of the consecutive stages. Since they have different bin widths, there are no ratio or percentage plotsz.

and used to produce the evolution plot throughout the three stages, such as the number of dead pixels, fitted Gaussian mean of the Threshold Distribution, etc. Furthermore, configuration files are introduced to improve the simplicity of the structure and the accessibility of the tool. More specifically, they include the general information of the module and the FEs, and also the parameters of the plots. Additionally, an overview plot is also introduced specifically for the last stage, where the modules are placed in the corresponding location on the large local support. The two layouts: Longeron and Half Ring are included, so that it provides an overview of the whole structure.

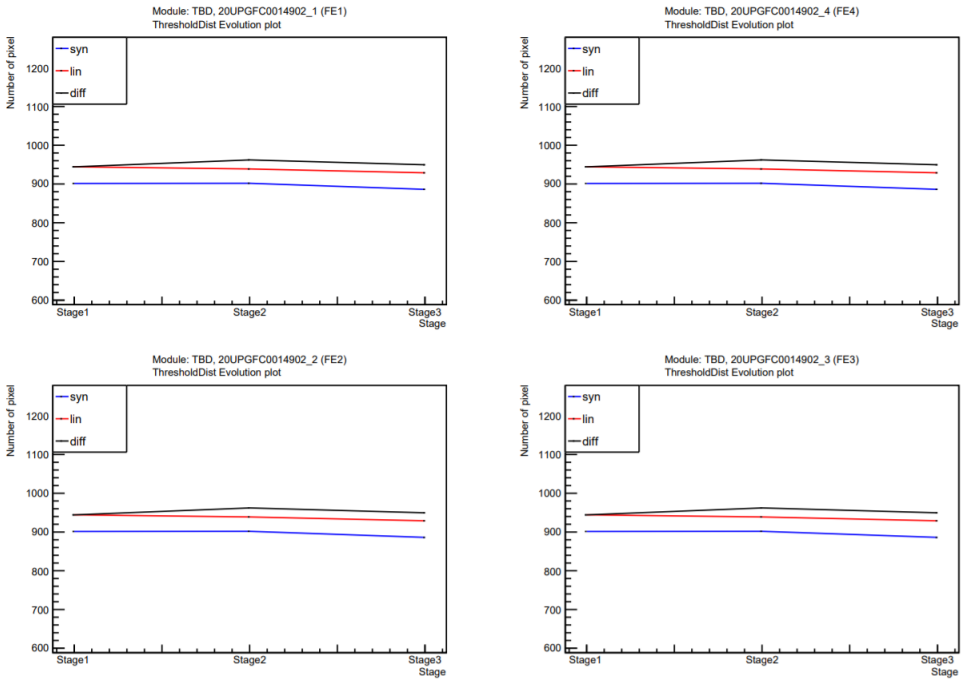


Figure 6: The evolution plot example of the Threshold Distribution of the module. Similar to the previous evolution plot the datasets of the 4 FE chips are the same, therefore the four plots are identical. The value in the y-axis represents the mean value extracted from the corresponding fitted Gaussian curve.

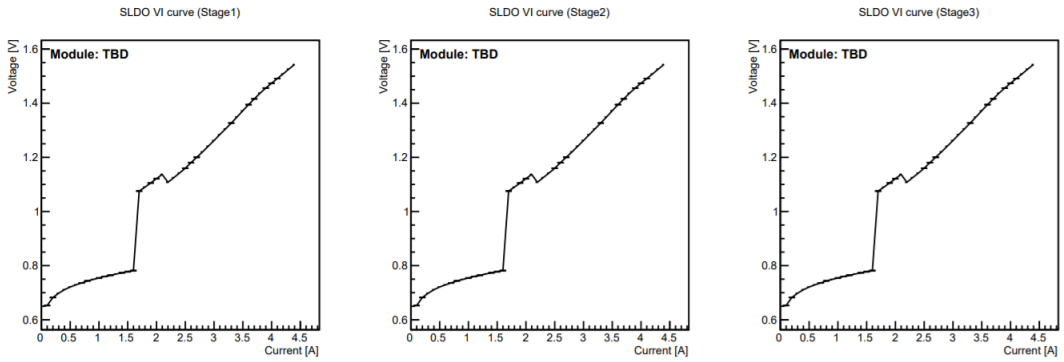


Figure 7: Example of the SLDO VI curve, similar to the previous plots, the three plots are the same, since they are from the same dataset.

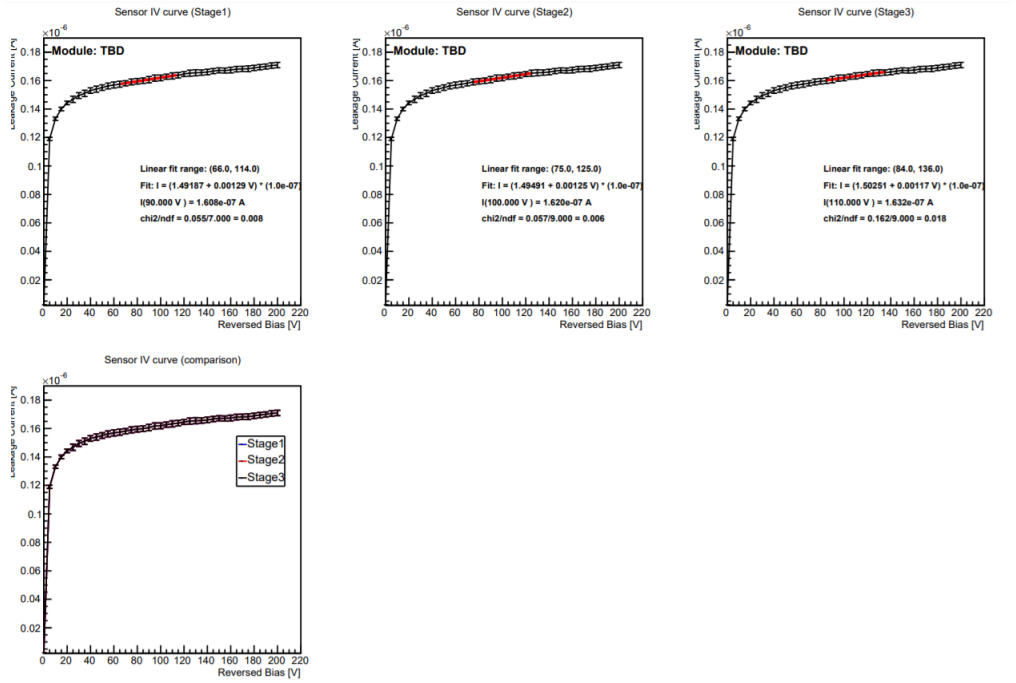
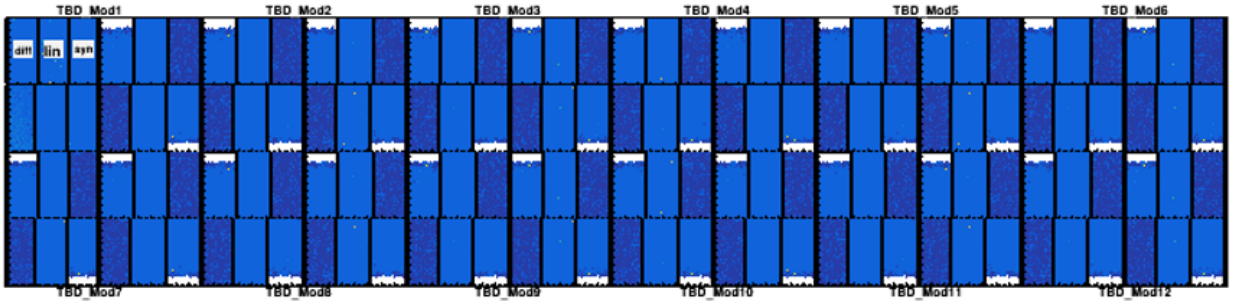
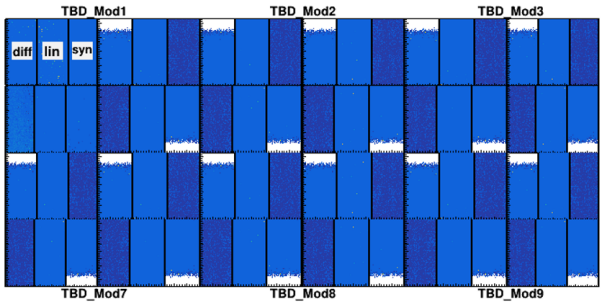


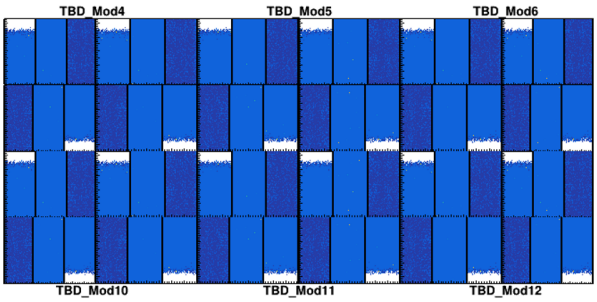
Figure 8: Example of the Sensor IV curve, where the red lines are printed on top of the data to indicate the fitted linear line. The fitting information and the extracted value are also printed on the plot. The bottom plot is the pure comparison of the three stages. Since the three datasets used for the stages are the same, therefore the comparison plot does not show any differences).



(a) The overview of the whole layout of the Longeron. The plot includes the module IDs in the corresponding locations (upper row: placed on top of the module; lower row: placed below the module). The first FE chip also includes the locations of the three FE types.



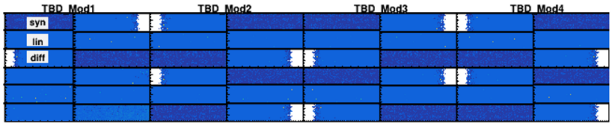
(b) The first piece of the detailed look on the local support. This only includes the modules in the first three columns. Namely, the 1st, 2nd, 3rd, 7th, 8th, and 9th module.



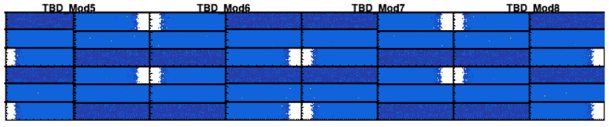
(c) The second piece of the detailed look on the local support. This only includes the modules in the 4th to 6th columns. Namely, the 4th, 5th, 6th, 10th, 11th, and 12th module.

Figure 9: Example of the Flat Longeron local support layout, which includes 12 modules in total. Since there are in total 12 modules, which exceeds 6. The plot will be separated into 2 pieces for the detailed plot, each piece consists of 6 modules.

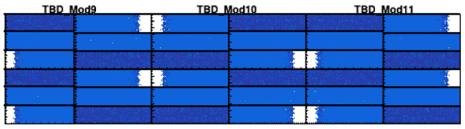
(a) The overview of the whole layout of the Half Ring. The plot includes the module IDs in the corresponding locations (placed on top of the module). The first FE chip also includes the locations of the three FE types.



(b) The first piece of the detailed look on the local support. This only includes the modules in the first four columns. Namely, the 1st, 2nd, 3rd, and 4th module.



(c) The second piece of the detailed look on the local support. This only includes the modules in the 5th to 8th columns. Namely, the 5th, 6th, 7th and 8th module.



(d) The third piece of the detailed look on the local support. This only includes the modules in the 9th to 11th columns. Namely, the 9th, 10th, and 11th module.

Figure 10: Example of the Inclined Half Ring local support layout, which includes 11 modules in total. Since there are in total 11 modules, which exceeds 4. The plot will be separated into 3 pieces for the detailed plot, they consists of 4, 4 and 3 modules, respectively.

Acknowledgement

I would like to first thank CERN for organizing the summer program even in this difficult time. And I would also like to express my deep gratitude to my supervisors at CERN, Dr. Benedikt Vormwald and Silke Mobius, for their guidance and patience in answering all of my questions and confusions. I would also like to thank Prof. Tao, Liu and Prof. Kirill Prokofiev, in my home university, for giving me this chance to participate in this summer program.

References

[1] Project git repository. `https://gitlab.cern.ch/itk-pixel-systemtest-ob/module-dossier`.

[2] Lingxin Meng. RD53A Module Testing Document. Technical report, April 2020.

[3] Maurice Garcia-Sciveres. The RD53A Integrated Circuit. Technical report, CERN, Geneva, Oct 2017.

[4] Folder for conguration files. `https://gitlab.cern.ch/itk-pixel-systemtest-ob/module-dossier/-/blob/master/Configuration`.

[5] General conguration file. `https://gitlab.cern.ch/itk-pixel-systemtest-ob/module-dossier/-/blob/master/Configuration/GenConfig.json`.

[6] Plot conguration file. `https://gitlab.cern.ch/itk-pixel-systemtest-ob/module-dossier/-/blob/master/Configuration/PlotConfig.json`.

[7] Local support configuration. `https://gitlab.cern.ch/itk-pixel-systemtest-ob/module-dossier/-/blob/master/Configuration/LocalSupportConfig.json`.

[8] Diego Alvarez Feito, Sergio Gonzalez Sevilla, and Nicolas Massol. ITk Pixel - Supporting documentation for PDR On-Detector Services – Outer Barrel. Technical report, March 2020.

[9] Folder for figures. `https://gitlab.cern.ch/itk-pixel-systemtest-ob/module-dossier/-/blob/master/Figure`.

A Details of Automatically Determined Fitting Range Option ("Auto")

The automatic and dynamic fitting range method aims at determining the suitable fitting range by the raw data from the histogram. This method is mainly used in the distribution which has a peak, and the fitting range aims to achieve a satisfying goodness of the Gaussian fit of this peak.

The first step is to approximate the rough location of the peak. We assume the highest 4 bins are concentrated around the peak, then we take the average of them to be the approximated peak location, say x. Secondly, the width of the peak, w, is first assumed to be the Root Mean Square (RMS) calculated from the raw histogram data. Therefore, the first fitting range would be $(x - w, x + w)$. If the first fit has $\chi^2/\text{ndf} \leq 10$ we will directly use the Gaussian fit result (to extract the mean value).

However, there might be some distributions that are asymmetrical. So the initial symmetrical fitting range by $x \pm w$ may not provide a satisfying fit. For instance in the Noise Distribution shown in Figure 5, there is a tail in the positive direction. To deal with this issue, we iterate the fitting range, by replacing the approximated peak location x, and the width w by the mean and the standard deviation from the previous fit. The iteration stops if the new fit has $\chi^2/\text{ndf} \leq 10$, or until the 10^{th} iteration.

B Complete Version of The Configuration Files

The complete version of the configuration files can be found in [4]. The following are some example of the complete version of the configuration files discussed above.

Firstly, the following shows an example of the complete version of the General Configuration file, which include the module and FEs information, directory of the plotting configuration and the output format:

```
{
  "Module": {
    "ModuleID": "TBD",
    "FE1": {
      "ID": "20UPGFC0014902_1",
      "FE": [ "diff", "lin", "syn" ],
      "FolderPath": {
        "Stage1": "../Data/data-20UPGM20021168_-15C",
        "Stage2": "../Data/data-20UPGM20021168_20C",
        "Stage3": "../Data/data-20UPGM20021168_30C"
      }
    },
    "FE2": {
      "ID": "20UPGFC0014902_2",
      "FE": [ "diff", "lin", "syn" ],
      "FolderPath": {
        "Stage1": "../Data/data-20UPGM20021168_-15C",
```

```
      "Stage2": "../Data/data-20UPGM20021168_20C",
      "Stage3": "../Data/data-20UPGM20021168_30C"
    }
  },
  "FE3": {
    "ID": "20UPGFC0014902_3",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGM20021168_-15C",
      "Stage2": "../Data/data-20UPGM20021168_20C",
      "Stage3": "../Data/data-20UPGM20021168_30C"
    }
  },
  "FE4": {
    "ID": "20UPGFC0014902_4",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGM20021168_-15C",
      "Stage2": "../Data/data-20UPGM20021168_20C",
      "Stage3": "../Data/data-20UPGM20021168_30C"
    }
  }
},
"PlotConfig": "./PlotConfig.json",
"OutputFormat": [ ".png", ".pdf" ]
}
```

Secondly, an example of the Plotting Configuration file is shown below. It consists of all the features, grouped in the three categories, mentioned above.

```
{
  "Hist2D": {
    "digitalscan": {
      "Name": "OccupancyMap",
      "EvolutionPlot": { "Y": "Absolute" }
    },
    "noisescan": {
      "Name": "NoiseMask"
    },
    "analogscan": {
      "Name": "OccupancyMap",
      "Cut": 200,
      "EvolutionPlot": { "Y": "Absolute" }
    }
  },
  "Graph": {
    "IV": {
```

```
    "Name": "SensorIV",
    "FitPoint": {
      "Stage1": 90,
      "Stage2": 100,
      "Stage3": 110
    },
    "FitStep": {
      "Stage1": 24,
      "Stage2": 25,
      "Stage3": 26
    },
    "Tref": 298.15
  },
  "VI": {
    "Name": "SLDO_VI"
  }
},
"Hist1D": {
  "NoiseDist": {
    "Name": "NoiseDist",
    "CompareType": "Ratio",
    "FitRange": {
      "diff": {
        "Stage1": "Full",
        "Stage2": "Auto",
        "Stage3": [ 100, 150 ]
      },
      "lin": {
        "Stage1": "Auto",
        "Stage2": "Full",
        "Stage3": [ 160, 220 ]
      },
      "syn": {
        "Stage1": "Auto",
        "Stage2": "Full",
        "Stage3": [ 300, 400 ]
      }
    },
    "EvolutionPlot": true
  },
  "ThresholdDist": {
    "Name": "ThresholdDist",
    "CompareType": "Percentage",
    "FitRange": {
      "diff": {
        "Stage1": "Auto",
        "Stage2": "Full",
```

```
        "Stage3": [ 800, 1000 ]
      },
      "lin": {
        "Stage1": "Auto",
        "Stage2": "Full",
        "Stage3": [ 800, 900 ]
      },
      "syn": {
        "Stage1": "Full",
        "Stage2": "Auto",
        "Stage3": [ 800, 1000 ]
      }
    },
    "EvolutionPlot": true
  }
 }
}
```

Lastly, the following is an example of the Local Support Configuration file, which is used in plotting the flat longeron with 6 modules:

```
{
  "Layout": "Longeron",
  "Plot": "analogscan",
  "OutputFormat": [ ".png", ".pdf" ],
  "Module1": {
    "ModuleID": "TBD_Mod1",
    "FE1": {
      "ID": "20UPGFC0014902",
      "FE": [ "diff", "lin", "syn" ],
      "FolderPath": {
        "Stage1": "../Data/data-20UPGR90021032",
        "Stage2": "../Data/data-20UPGR90021032",
        "Stage3": "../Data/data-20UPGR90021032"
      }
    },
    "FE2": {
      "ID": "20UPGFC0014902",
      "FE": [ "diff", "lin", "syn" ],
      "FolderPath": {
        "Stage1": "../Data/data-20UPGM20021168_20C",
        "Stage2": "../Data/data-20UPGM20021168_20C",
        "Stage3": "../Data/data-20UPGM20021168_20C"
      }
    },
    "FE3": {
      "ID": "20UPGFC0014902",
```

```
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGM20021168_30C",
      "Stage2": "../Data/data-20UPGM20021168_30C",
      "Stage3": "../Data/data-20UPGM20021168_30C"
    }
  },
  "FE4": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  }
},
"Module2": {
  "ModuleID": "TBD_Mod2",
  "FE1": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE2": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE3": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE4": {
```

```
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  }
},
"Module3": {
  "ModuleID": "TBD_Mod3",
  "FE1": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE2": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE3": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE4": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  }
```

```
},
"Module4": {
  "ModuleID": "TBD_Mod4",
  "FE1": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE2": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE3": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE4": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  }
},
"Module5": {
  "ModuleID": "TBD_Mod5",
  "FE1": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
```

```
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE2": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE3": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE4": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  }
},
"Module6": {
  "ModuleID": "TBD_Mod6",
  "FE1": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE2": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
```

```
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE3": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  },
  "FE4": {
    "ID": "20UPGFC0014902",
    "FE": [ "diff", "lin", "syn" ],
    "FolderPath": {
      "Stage1": "../Data/data-20UPGR90021032",
      "Stage2": "../Data/data-20UPGR90021032",
      "Stage3": "../Data/data-20UPGR90021032"
    }
  }
 }
}
```