Reflectance Analysis Code Deep-Dive

Mackenzie Devilbiss
Scintillator R&D Meeting
3/7/22

Summary

- Last meeting: just finished taking data on all coupons
- Now: developing analysis code
  - First, get reflectance data from Excel into ROOT (Macro #1; reflectance_data.C)
  - Then, do more sophisticated analysis and manipulate data to make new plots (Macro #2; reflectance_analysis.C)
  - Trying my best to make smart choices so that others will be able to use this in the future!

reflectance_data.C

Data from Excel to ROOT

- Data is copied from Excel files into ROOT macro using the syntax of a **map** of **strings** to **vectors**
- The **map** format allows us to automatically order our data alphabetically using **names**, **vectors** allow for variable size data to be compared
- Copying the data in is fairly quick and easy using a string replace for the Excel cell delimiter spaces
- We will be able to add to the **map** in the future without changing the rest of the macro to make the data file!
- Data stored in histograms to do bin-wise arithmetic operations

```
// this file stores the raw data arrays collected using the spectrophotometer
// and will turn these arrays into .root files to use for manipulation

void reflectance_data(){

  //Create output file. If it already exists, recreate it.
  TFile *ofile = new TFile("output_TiO2_data_hists.root","RECREATE");

  // reflectance data by sample_scientist
  // this is stored in a map!! this allows us to have nice, fancy names attached to each
  // dataset, while being able to simultaneously loop through the dataset names and
  // data values in each dataset. The vector aspect means that datasets can easily be
  // different sizes. We can always add to the map

  // when entering new datasets, please be sure to use the correct map syntax

    std::map < string, std::vector<Float_t> > map =
      {
        // USPN085939
        {"USPN085939_mackenzie_thin_0", {10.42,12.26,16.12,27.73,48.24,74.36,89.73,93.7,94.07,94.6,95.01,
95.21,95.45,95.61,95.76,95.85,95.98,96.12,96.19,96.16,96.16,96.18,96.21,96.2,96.25,96.31,96.4,96.54,96.65
,96.7,96.77,96.8,96.9,97.03,97.07,97.04,96.95,97.01,96.95,97.04,97.02,97.05,96.65} },
        {"USPN085939_mackenzie_thin_1", {10.12,12.19,16.22,27.77,48.39,74.43,89.6,93.52,93.84,94.44,94.88
,95.1,95.32,95.45,95.56,95.65,95.79,95.94,96.01,95.97,95.96,95.97,96,96.01,96.04,96.1,96.19,96.31,96.4,96
.46,96.5,96.51,96.67,96.73,96.76,96.73,96.61,96.62,96.57,96.64,96.63,96.68,96.35} },
        {"USPN085939_mackenzie_thin_2", {10.05,11.97,16.25,27.91,48.55,74.49,89.3,92.87,93.1,93.65,94.06,
94.24,94.38,94.49,94.56,94.57,94.63,94.75,94.73,94.65,94.57,94.53,94.52,94.48,94.53,94.53,94.67,94.72,94.
75,94.76,94.73,94.71,94.83,94.84,94.78,94.61,94.44,94.43,94.36,94.29,94.2,94.31,94.02} },
        {"USPN085939_mackenzie_thin_3", {9.98,11.98,16.29,27.83,48.45,74.36,89.49,93.3,93.55,94.17,94.54,
94.78,95,95.1,95.21,95.27,95.42,95.57,95.6,95.55,95.53,95.5,95.47,95.42,95.45,95.46,95.57,95.68,95.71,95.
78,95.79,95.79,95.91,95.96,95.93,95.85,95.7,95.68,95.68,95.68,95.7,95.6,95.29} },
U:---  reflectance_data.C   Top L23    (C++/l Abbrev)
```

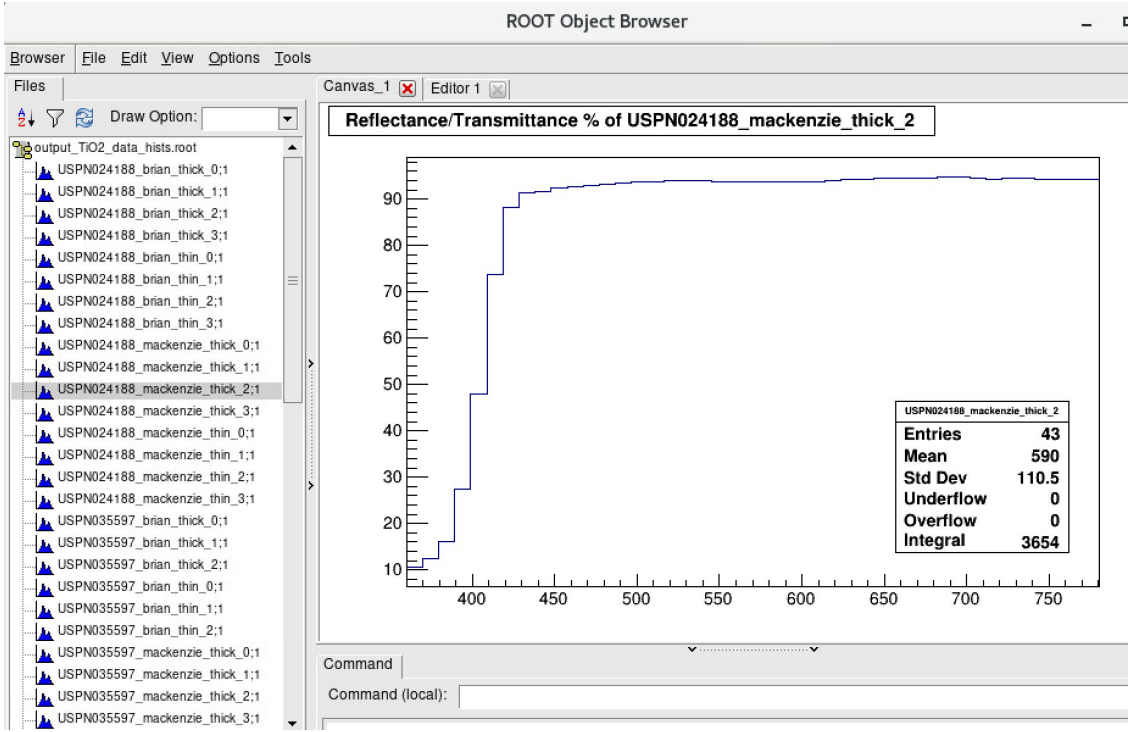
```
// set up loops to handle array titles and appropriately name histograms
std::vector<TH1F*> ref_hists;
string histname;
string label;

// initialize histograms and create vector of empty histograms
for(const auto& pair : map){
  histname = pair.first;
  std::cout << "histname = " << histname << std::endl;
  label    = "Reflectance/Transmittance % of " + histname;
  ref_hists.push_back(new TH1F(histname.c_str(), label.c_str(),43,360.,780.));
}

// fill histograms, first loop is same as above
int nhist = ref_hists.size();
int index = 0;
for(const auto& pair : map){
  int bin_index = 1;
  for(Float_t f : pair.second){
    ref_hists[index]->SetBinContent(bin_index,f);
    ++bin_index;
  }
  ++index;
}
```

Screenshot of Data Histogram File

- Output from reflectance_data.C is this .root file that just contains spectra that would be identical to spec. GUI
- Naming convention:
  - Lot ID, scientist, thickness, coupon #
  - Descending order of importance for grouping



reflectance_analysis.C

Doing Some Fancier Analysis

- The second ROOT macro is where I am planning to do some more sophisticated work, like...
  - Take an average over all of the coupons with a given ID
  - Take differences between the averages of data from me and from Brian
    - We measured some of the same coupons 8-14 months apart, have they aged?
  - Extract a point at an interesting wavelength for each ID and plot against production date
    - Are there any trends here?
- Input: reflectance data histogram .root file

```
// get list of keys in old file
TIter keyList(file->GetListOfKeys());
TKey* key;
std::vector<TString> keyNames;
std::vector<string> key_strs;
std::vector<TH1F*> input_hists;
while((key = (TKey*)keyList())){
  // std::cout << key->GetName() << " " << key->GetClassName() << std::endl;
  TString key_name = key->GetName();
  keyNames.push_back(key_name);
  input_hists.push_back(new TH1F(key_name, key_name,43,360.,780.));
}

// get histograms from input file
int nkeys = keyNames.size();
for(int i = 0;i < nkeys;i++){
  //std::cout << keyNames[i] << std::endl;
  input_hists[i] = (TH1F*)file->Get(keyNames[i]);
  std::string key_string(keyNames[i].Data());
  key_strs.push_back(key_string);
}
```

Grouping

- I am putting work into also making this code automated
  - Currently, it reads in the input histograms, so they will be in the same order as the map
  - I can then search for strings in the histogram names (also in same order) to make groups of interesting indices to histograms
  - Then, I do bin-wise arithmetic on histograms in the group of interest
  - Write out to a new analysis histogram .root file
- Breaking down tasks that I want to repeat, make them generalized functions to call given a set of interesting indices
  - Average histogram function
  - Difference of histogram function

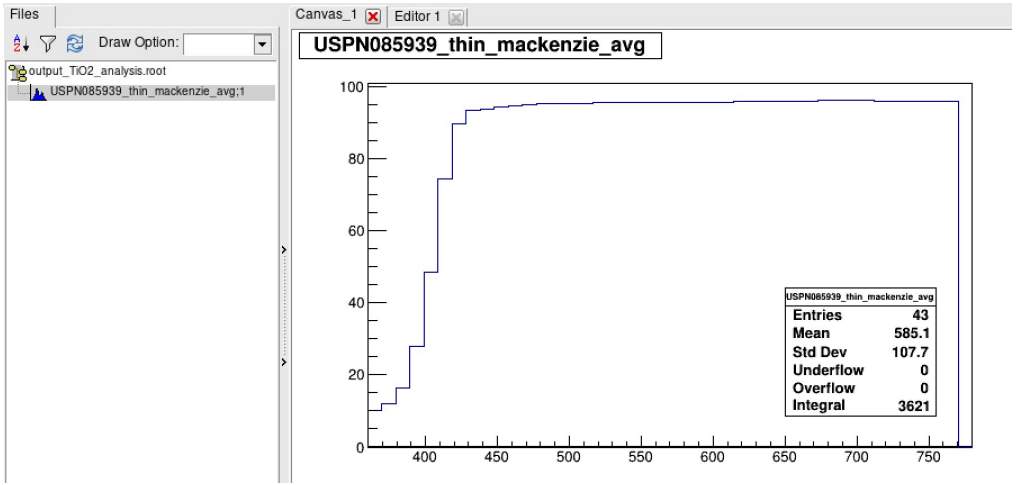
```
// determine groups of files that are interesting...
// here, you want to collect vectors of indices into the key vectors to do operations on a particular group of hists
std::vector<int> thn_avg_b_085939;
std::vector<int> thk_avg_b_085939;
std::vector<int> thn_avg_m_085939;
std::vector<int> thk_avg_m_085939;
std::vector<int> thn_avg_b_024188;
std::vector<int> thk_avg_b_024188;
std::vector<int> thn_avg_m_024188;
std::vector<int> thk_avg_m_024188;

// here is where we actually do the grouping and push indices into the above vectors
int index = 0;
for(std::string& s : key_strs){
  if(s.find("USPN085939_mackenzie_thin") != std::string::npos){
    //std::cout << s << std::endl;
    thn_avg_m_085939.push_back(index);
    //std::cout << "index = " << index << " val check: " << keyNames[index].Data() << std::endl;
  }
  if(s.find("USPN085939_mackenzie_thick") != std::string::npos){
    thk_avg_m_085939.push_back(index);
  }
  if(s.find("USPN085939_brian_thin") != std::string::npos){
    thn_avg_b_085939.push_back(index);
  }
```

In the list of names, there are numbers on the end of all these strings!!

First Average Histogram Using an Index Group!

- I searched through my list to make a group of interesting indices, then made an average of all of the ‘thin’ data samples from a group of coupons that I took
- Add entries in each bin for each histogram in group, then divide by number of hists in group
- It works! Just needs generalized, then it will be trivial to have N of these histograms



Summary

- I have been using ROOT for some time, but most of the macros that I have been using have been within Mu2e framework, where others have put in a lot of work to standardize data formats, naming conventions, etc.
- First time doing it from the ground up is taking some work
- Making progress and getting more confident
- Store on github?

Questions from Mu2e Collab Meeting

- Is there any older reflectance data, older than Brian's data? Perhaps from Anna or others at that time? Data from time of TiO_2 arrival to Fermi?
- Can we order new coupons? Is it reasonable to order coupons without an order in for TiO_2 pellets?
- Would it be interesting to look at a sample regularly over a long period of time? If aging occurs rapidly after production, this is where a new coupon could be interesting to continuously measure and monitor for change
- Are there any CRV electronics in Lab 6? CRV FEBs in particular?