Sprout 

a software tool for your analysis

Malin Bohman

UPPSALA
UNIVERSITET

https://github.com/malle-b/Sprout

Introduction

- Sprout🌱: a little project that grew from ROOT
- ROOT is great, but:
  - Takes up many lines of code
  - Find myself writing a lot of similar code many times over
- Aims of Sprout🌱:
  - Automate the boilerplate
  - Make it easier to keep manageable and modular code

The Sprout Family

SproutTree: 2D vector to hold data with functions to convert to and from TTrees.

SproutPlot : To help you with ROOT histograms

SproutFit: Class dedicated to fitting with SproutPlots

SproutParam: Keep track of your analysis parameters

UPPSALA
UNIVERSITET

SproutTree

- 2D vector to hold data
- Uses indexes instead of names to access data
- Can loop over rows AND columns easily
- Branches can contain different number of events
- Can be read and written to .root files
- Can convert into TTree and vice versa

NOT suitable when data is larger than memory!

UPPSALA
UNIVERSITET

SproutTree - Storing data

```
SproutTree stree(4);

for(each event){
    stree.addToBranch(0,something);
    stree.addToBranch(1,something);
    stree.addToBranch(2,something);
    stree.addToBranch(3,something);
}

stree.write();
```

```
TTree tree("tree","tree");

float par1, par2, par3, par4;

TBranch* b1 = tree.Branch("branch1", &par1);
TBranch* b2 = tree.Branch("branch2", &par2);
TBranch* b3 = tree.Branch("branch3", &par3);
TBranch* b4 = tree.Branch("branch4", &par4);

for(each event){
    par1 = something;
    par2 = something;
    par3 = something;
    par4 = something;

    tree.Fill();
}

tree.write();
```

UPPSALA
UNIVERSITET

SproutTree - Retrieving data

```
SproutTree stree = get from p_cut_file

for(event i){
    do something with branch 0 event i
    do something with branch 1 event i
    do something with branch 2 event i
    do something with branch 3 event i
}
```

```
TTree tree = get from p_cut_file

float par1, par2, par3, par4;

tree.SetBranchAddress("branch1", &par1);
tree.SetBranchAddress("branch2", &par2);
tree.SetBranchAddress("branch3", &par3);
tree.SetBranchAddress("branch4", &par4);

for(each event){
    do something with par1
    do something with par2
    do something with par3
    do something with par4
}
```

UPPSALA
UNIVERSITET

SproutTree vs TTree

Speed comparison with 3 x 1e9 floats

	SproutTree	**TTree**
Adding	25 s	370 s
Writing	341 s	0.003 s
Reading	90 s	0.002 s
Retrieving	< 0.001 s	89 s
Total	456	459 s

UPPSALA
UNIVERSITET

SproutPlot

- A collection of histograms: keeps a list of all histograms created with it
- Can loop over histograms
- Write all histograms at the same time
- Can plot all histograms to the same canvas or individually
- Can easily set style of all histograms it contains and change the style for individual histograms
- Retrieve all histograms from .root file with a common prefix

UPPSALA
UNIVERSITET

SproutPlot

getTH1F() creates and returns a reference to a histogram - any change made to the reference will also affect the histogram stored internally

```
/**
* Returns an empty TH1F histogram with properties set to the specified parameters.
*
* @param name name of the histogram
* @param bins number of bins
* @param xmin lower edge of x-axis
* @param xmax upper edge of x-axis
* @param xlabel label displayed on x-axis. Set to "x" by default
* @param ylabel label displayed on y-axis. Set to "counts" by default
*
* @return empty TH1F histogram
*/
TH1F& getTH1F(TString name, int bins, double xmin, double xmax, TString xlabel="x", TString ylabel="Counts");
```

UPPSALA
UNIVERSITET

SproutPlot

- Creating some histograms, note the use of references!
- Filling them from a Gaussian random distribution
- The SproutPlot write functions writes all histograms it contains

```
SproutPlot splot;

TH1F& h1 = splot.getTH1F("test1", 20, -5, 5);
TH1F& h2 = splot.getTH1F("test2", 20, -5, 5);
TH1F& h3 = splot.getTH1F("test3", 20, -5, 5);
TH1F& h4 = splot.getTH1F("test4", 20, -5, 5);
TH1F& h5 = splot.getTH1F("test5", 20, -5, 5);
TH1F& h6 = splot.getTH1F("test6", 20, -5, 5);


for(int i=0; i<1000; i++){
    for(TH1F& h : splot){h.Fill(rnd.Gaus(0,1));} //Fills all
    h1.Fill(rnd.Gaus(0,1)); //Fills only h1
}

TFile file = TFile("testFile.root", "recreate");
file.cd();

// Writes to file or saves all histograms as a .png
splot.writeBasic();

// Writes to file or saves a canvas displaying all hisotgrams as a .png
splot.writeCanvas();

file.Close();
```

UPPSALA
UNIVERSITET

SproutPlot

Unsure of how many histograms you'll need at compile time?

- Initializing histograms in a loop and access them later through splot

```
SproutPlot splot;

for(int i=0; i<n_hist; i++){
    splot.getTH1F("hist_"+int2str(i), 100, 0, 1);
}
```

Easily read and sort histograms from file

- Reading histograms from file using a string identifier

```
// reads all histograms in "myFile.root" whose names
// contain "proton" into splot.
splot.add("myFile.root", "proton");
```

UPPSALA
UNIVERSITET

SproutFit

Based on Rafal Lalik's software package HelloFitty

- Can fit user-specified signal+background distributions to all histograms contained in a SproutPlot.
- Signal and background models and starting parameters (optional) is specified by the user in a .txt file.
- Automatically finds suitable start parameters if none are given
- Calculates the number of signal events with uncertainties

UPPSALA
UNIVERSITET

SproutFit

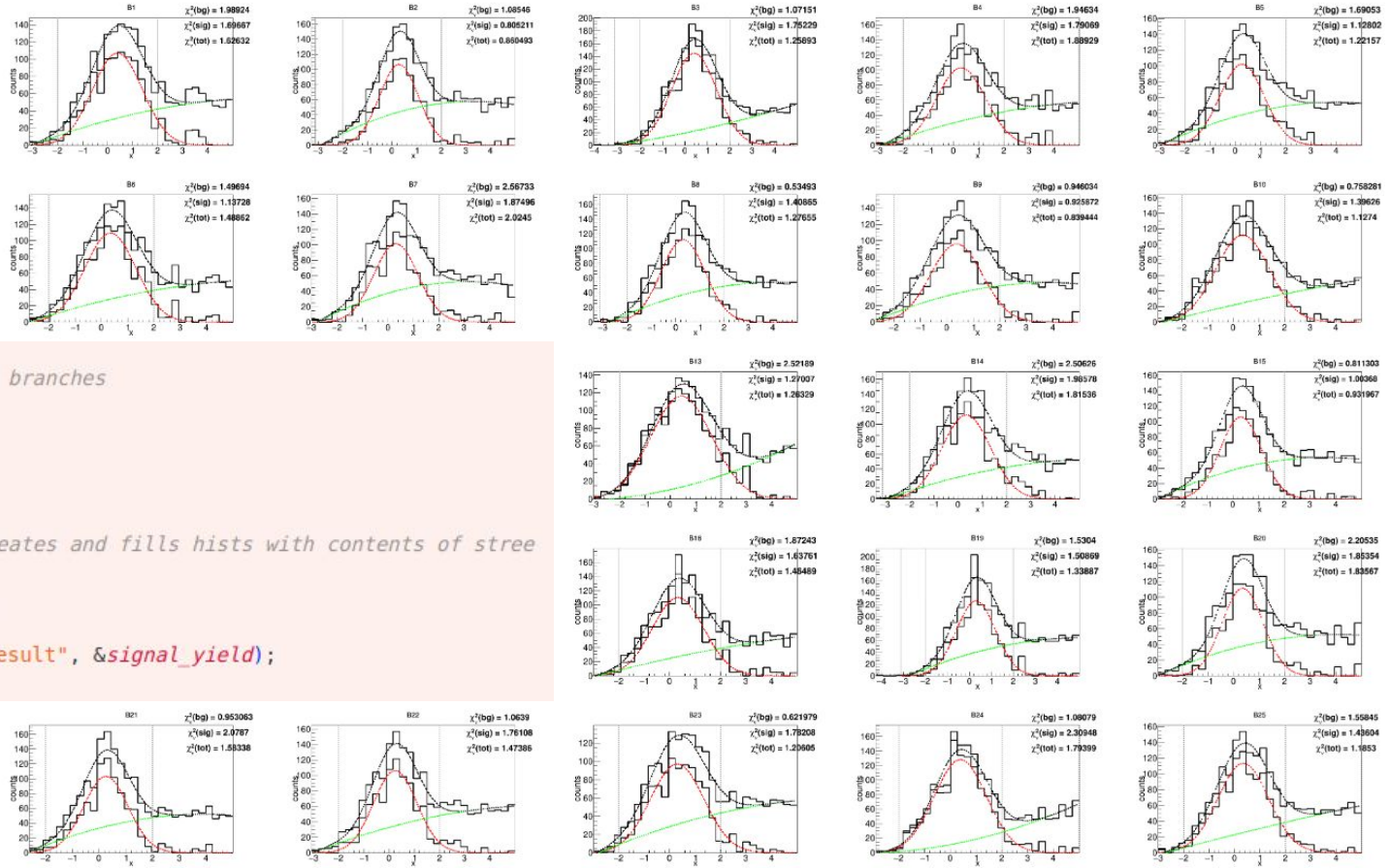
```
SproutTree stree(25); // 25 branches
SproutPlot splot;
SproutFit sfit;

fill stree

splot.getTH1F(stree); // Creates and fills hists with contents of stree

SproutTree signal_yield(4);

sfit.fit(splot, "Plots/FitResult", &signal_yield);
```



UPPSALA
UNIVERSITET

SproutParam

Its purpose

- Offers a simple way to store and access parameters over multiple files
- Stored parameters can be changed easily without needing to recompile
- Easily define and run different analysis cases with different sets of parameters for comparison
- Reduces the risk of human error

UPPSALA
UNIVERSITET

SproutParam

How it works:

- Parses a .txt file of key-value pairs and contains functions to retrieve the value of a given key.
- The key-value pairs are stored in a HashMap
- Search algorithm is O(1) on average
- won’t slow you down too much, regardless of how many parameters you have stored

UPPSALA
UNIVERSITET

SproutParam

Example of a parameter file

- Lines that start with “#” are ignored by the parser
- Lines parsed as *key* followed by *value*
- *key* expects a string
- *value* automatically stored as a string, int or float depending on its format

```
# paths and file names
#------------------------------
dst_list                /path/to/my_dst_list.txt
out_dir                 /path/to/output/folder
dst_loop_output         lambdaCands.root

# event selection
#------------------------------
trigger_pt1             0
trigger_pt2             0
trigger_pt3             1
proton_pion_sel         1

# track selection
#------------------------------
use_flag_Hadron         1
use_flag_kIsUsed        0
use_cut_pid             1
p_cut_file              Proton_Cut_file.root
pi_cut_file             Pion_Cut_file.root
use_hades_protons       1
use_fwd_protons         1

# corrections
#------------------------------
beam_correction         1
energy_correction       1

# background reduction
#------------------------------
invM_cut                1
```

SproutParam

- Read the parameter file `sp.read(std::string filename)`
- Access value with its key `sp.get<data type>(std::string key)`
- Enable / disable various steps of your analysis
- Store variables that might be tweaked multiple times

```
SproutParam sp;
sp.read("myParams.txt");

if(sp.get<int>("invM_cut")){ // enabled = 1, disabled = 0
    if(lambda.M() < sp.get<float>("lowM_cut") || lambda.M() > sp.get<float>("upM_cut")){
        //remove candidate
    }
}
```

UPPSALA
UNIVERSITET

How to use Sprout

- Relies only on ROOT
- Download the package: https://github.com/malle-b/Sprout.git

- Use in ROOT macro: Add this line to your rootlogon.C

  ```
  gSystem->Load("path/to/your/install/dir/lib/libSprout.so
  ```

- Use in executable: Add this line to your project's CMakeLists.txt

  ```
  include(your_sprout_install_dir/lib/cmake/Sprout/SproutTargets.cmake)
  ```

  and link it to your target along with ROOT::Core

UPPSALA
UNIVERSITET

Thank you!

UPPSALA
UNIVERSITET

SproutFit: Input parameter file example

Backup

```
pol2 gaus -3.11928 4.99871 -2 2 28.7394  7.78128  -0.558969  107.245  0.392814  0.991924
pol2 gaus -3.11928 4.99871 -2 2 41.237  9.95509  -1.49836  106.505  0.276774  0.791152
pol2 gaus -3.11928 4.99871 -2 2 19.3014  7.24632  0.374294  144.894  0.409082  1.00071
pol2 gaus -3.11928 4.99871 -2 2 29.7916  8.59777  -0.626348  102.846  0.298795  0.974751
pol2 gaus -3.11928 4.99871 -2 2 36.1238  8.90835  -1.10583  102.206  0.265028  0.878244
pol2 gaus -3.11928 4.99871 -2 2 72.5726  14.8853  -5.18355  -1.19141e+10  -1.28004  0.00565776
pol2 gaus -3.11928 4.99871 -2 2 37.4449  9.37459  -1.48758  101.723  0.310529  0.86604
pol2 gaus -3.11928 4.99871 -2 2 34.4013  9.57495  -1.14578  112.217  0.298796  0.909074
pol2 gaus -3.11928 4.99871 -2 2 75.3767  14.8835  -5.33855  112467  -0.878743  -0.0282641
pol2 gaus -3.11928 4.99871 -2 2 22.0282  7.48923  -0.195564  112.051  0.366174  -1.05047
pol2 gaus -3.11928 4.99871 -2 2 23.6093  7.68782  -0.12034  117.313  0.349493  0.943404
pol2 gaus -3.11928 4.99871 -2 2 24.4318  8.17412  -0.202554  105.343  0.227791  1.05504
pol2 gaus -3.11928 4.99871 -2 2 10.4908  6.32788  0.82252  116.25  0.438572  -1.20281
pol2 gaus -3.11928 4.99871 -2 2 29.1846  7.97549  -0.661164  112.269  0.336738  1.0057
pol2 gaus -3.11928 4.99871 -2 2 37.3783  9.18131  -1.28699  106.058  0.312023  0.798217
pol2 gaus -3.11928 4.99871 -2 2 22.4391  7.93826  0.120371  111.68  0.351523  1.07219
pol2 gaus -3.11928 4.99871 -2 2 78.2404  14.4209  -4.94196  -3.87011e+06  -1.3299  -0.024025
pol2 gaus -3.11928 4.99871 -2 2 24.6451  7.74814  -0.334181  110.305  0.332101  -1.01057
pol2 gaus -3.11928 4.99871 -2 2 36.5364  10.5195  -0.99113  125.986  0.294188  0.890542
pol2 gaus -3.11928 4.99871 -2 2 34.7326  8.65592  -1.05675  111.285  0.357855  0.801742
pol2 gaus -3.11928 4.99871 -2 2 33.2054  9.41413  -1.18427  103.127  0.256367  0.829187
pol2 gaus -3.11928 4.99871 -2 2 32.8913  8.84999  -0.673059  106.263  0.240476  0.858723
pol2 gaus -3.11928 4.99871 -2 2 28.615  8.67445  -0.593776  97.3454  0.299191  -1.04523
pol2 gaus -3.11928 4.99871 -2 2 11.157  6.63691  0.926309  128.247  0.388708  1.01046
pol2 gaus -3.11928 4.99871 -2 2 22.894  8.08729  -0.133195  113.377  0.343666  1.00274

```

UPPSALA
UNIVERSITET

SproutFit: Finding start parameters

Backup

Procedure from *Data Reduction and Error Analysis*, P. Bevington and D. Robinson

1. Simultaneously fit the background function above and below the signal region
2. Keep the bg parameters fixed and fit the total function (sig + bg) in the signal region
3. Use these parameters as starting values to fit the total function over the fitting whole region
4. Repeat step 1-3 with with the new start parameters.

UPPSALA
UNIVERSITET