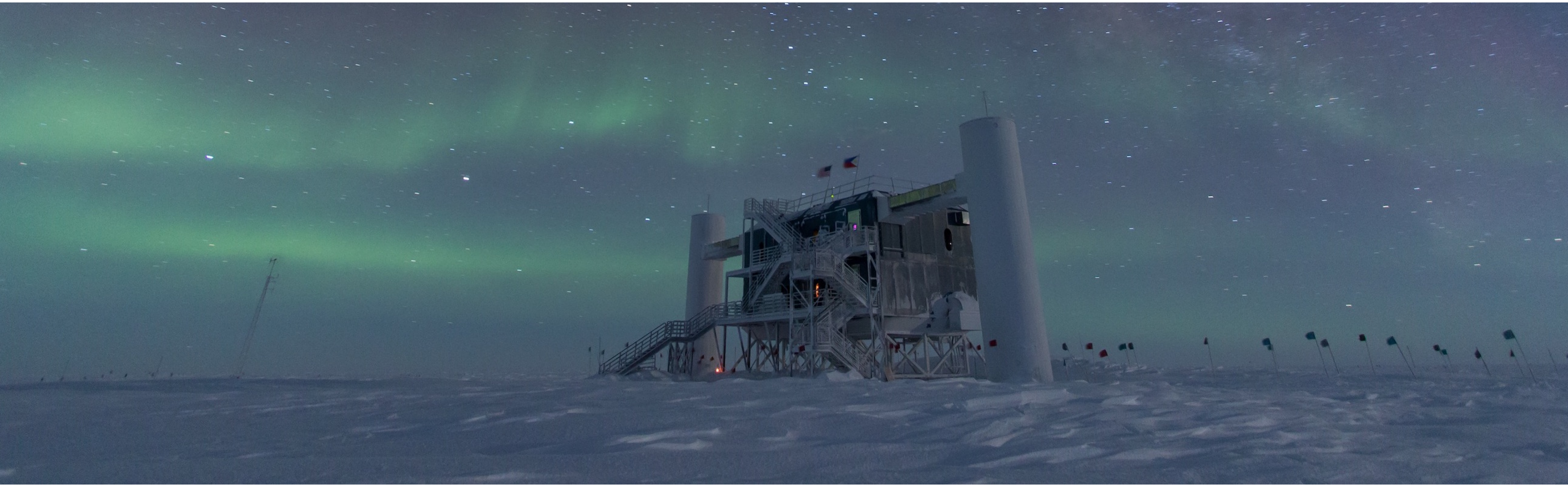

Counting Analysis Walkthrough

Maryland LLH Meetings

Anatomy of an Analysis

- Define your signal and background expectations. How should your analysis distinguish signal neutrinos from background?
- Define the analysis. What are you searching for? What information do you plan on using in your search?
- Write your likelihood/test statistic function. This will be a function to return a single value describing whether your events are more background-like or signal-like
- Test how your test statistic behaves on simulated background (and signal+background) datasets. If we get a test statistic of 4.2, what does that actually mean?
- Find your "sensitivity", the amount of signal that would be required for your analysis to successfully distinguish whether your events are due to background or signal+background
- Run your test statistic calculation over the data. Compare this to the background to determine the significance

Our dataset today

- We're going to be using the point-source (PS) sample for the next few sessions
  - Extremely track-pure sample (so NuMu events)
  - Events between 1 TeV to 100+ PeV
  - Good angular resolutions (less than 1 degree error)
  - Used pretty widely by the collaboration already
- The sample's been slightly modified
  - Only including 2012 data, simulation to avoid complications from combining years
  - Cut out detector downtime
  - Randomized the times and right ascensions to avoid issues with blindness
- Events are in a special numpy format
  - To access the declination of all events, run `data['dec']`
  - To access all information from event 4, run `data[4]`
  - To get the declination of event 4, run `data[4]['dec']` or `data['dec'][4]`

Coding today

- We'll be putting together analysis code in the counting_analysis python notebook

- Open the notebook and run the first three boxes. These will import the libraries, load the data and simulation, and load something called the "GoodRunList"
  - GoodRunList: A file containing information about when the detector was taking good-quality data. Times when something broke (a string turned off, DOMs behaved badly) or when we were calibrating the detector (eg, by flashing LEDs in the ice) are difficult to use, so we exclude those times from our data using the GRL
  - We will not be using this much today, but it's important for time-dependent analyses.

- Find the total number of events and livetime of the data to calculate the average rate

Define your expectations

- Define your signal and background expectations. How should your analysis distinguish signal neutrinos from background?
  - How will we be able to tell if we have signal events?
  - Do signal events appear at a specific time?
  - Should they be coming from a particular direction?
  - Are your signal events typically higher energy than the background events?
  - Do the signal events follow a different energy spectrum?

- No coding, but try to think about what we should be expecting. The choices here determine what we need to put into our likelihood calculations later
  - Next time, we'll use this step to define functions to describe our signal, background weights

Define your search methods

- Define the analysis. What are you searching for? What information do you plan on using in your search?
  - We want to know whether there is a significant excess of events within 500 seconds of a time T
  - We will be using timing information, but will not include spatial or energy information yet

- How can you pick out a window around T=123.4 days from your data?
  - Note that the dataset we're working with uses times in the MJD format, meaning that T=1.0 means day 1

Write a likelihood

- Write your likelihood/test statistic function. This will be a function to return a single value describing whether your events are more background-like or signal-like

- Find the get_test_statistic() function.
  - We're just counting events today
  - What form should this take for our search?
  - Write this function. It should take a sample of events from data and return a test statistic (LLH) value

Test your TS function with signal, background

- Test how your test statistic behaves on simulated background (and signal+background) datasets. If we get a test statistic of 4.2, what does that actually mean?

- We want to know how this behaves **before applying it to our data**.
  - We can use simulation to produce a "trial", a test dataset that we can run our function on
  - We want to know how it performs for both signal and background events

- Write a function to produce a trial dataset
  - Function should take any parameters required to predict the number of signal events from your simulation (here, we just need mean expected number of signal events)
  - Take a look at np.random.poisson and np.random.choice for two usable options for this analysis. Later work will need to use np.random.choice
  - Run the trial dataset through your TS function

ICECUBE
South Pole Neutrino Observatory

Test your TS function with signal, background

- Test how your test statistic behaves on simulated background (and signal+background) datasets. If we get a test statistic of 4.2, what does that actually mean?

- How do the distributions look if you include 5 signal events?
  - Run your code for 10000 background trials and 1000 signal+background trials
  - Create a plot to show how the two distributions compare

Calculating an analysis sensitivity

- Find your "sensitivity", the amount of signal that would be required for your analysis to successfully distinguish whether your events are due to background or signal+background

- The traditional definition used in IceCube is "What signal parameters are necessary so that 90% of your signal+background distribution is above the median of your background distribution?"

- How many signal events are requires for our analysis?
  - Hint: Take a look at np.percentile. Note that it returns the value X so that q percent of your distribution is below X

- Extra: Find the 3σ "discovery potential"
  - What signal parameters are necessary so that 50% of the signal+background distribution is above 99.73% of the background distribution?

"Unblinding" your data

- Run your test statistic calculation over the data. Compare this to the background to determine the significance

- What fraction of the background is above your data's TS value?
    - This is the "pre-trial" probability. The lower the value, the more significant the observation

- If you look at enough times in the data, you will eventually see something significant by random chance
    - Handling this is called a "trials-correction"
    - Roughly speaking: the more times you look at the data, the less surprising (less significant, large probability) any deviation from background is.
    - Can be a little difficult to take into account. Not planning to talk about it today

Extra questions to answer (if we have time)

- How does the sensitivity of the analysis change with a time window of 1 day? What about 100 seconds?
    - Why does it change like that?
    - What other ways could we achieve the same effect?

- Go through the steps again, but only looking at events within 10 degrees of the horizon (dec = 0). How does this affect the sensitivity?