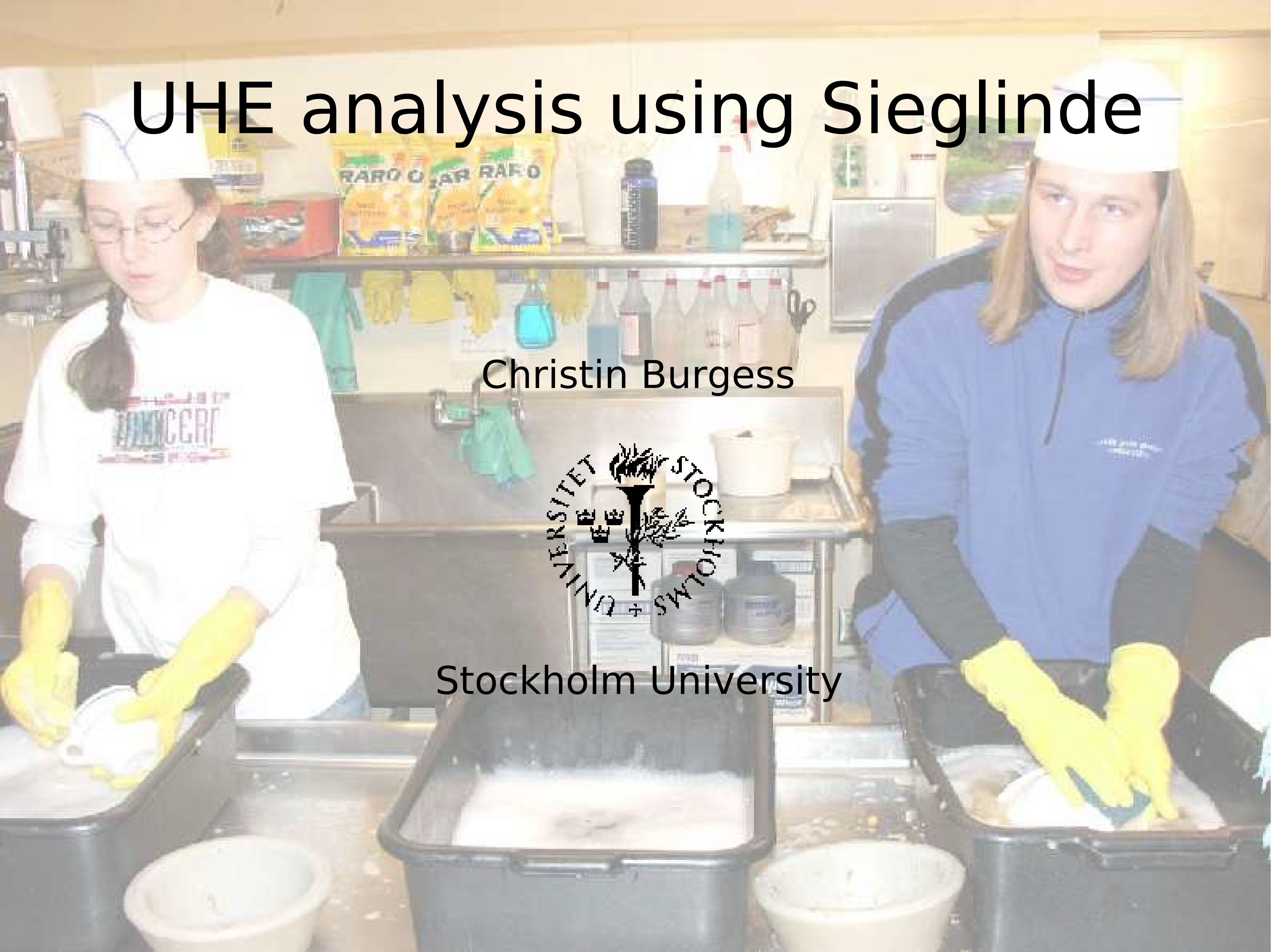
UHE analysis using Sieglinde



Christin Burgess

Stockholm University

Purpose of talk

Present my experience of using the presently available tools to perform an UHE analysis

"Proof of principle": my aim has been to lay the groundwork of an UHE analysis that does not use Siegmund and PAW

The main effort has been put into arriving at a working method, and now I am ready to focus on the actual analysis

I only use Sieglinde, ROOT and IcePack

I only write and read ROOT files (expect the original experimental data files)

Why Sieglinde?

Why Sieglinde?

Want to use the TWR information, can't use Siegmund

Want the advantages of using ROOT

Expected Sieglinde to replace Siegmund, and support to be shifted from Siegmund to Sieglinde

Why not IceTray?

Need something that is working now

Sieglinde and ROOT

Not enough information in ROOT output for analysis in ROOT →

Classic Sieglinde:

- Read and write F2K
- Write limited ROOT output
- Tested
- Documented

↓

All classical modules re-implemented, new modules added

↓

SLART (SiegLinde Analysis ROOT Tree):

- Read and write F2K
- Read and write ROOT files
- Not thoroughly tested
- Limited documentation

Sieglinde and ROOT: pros

- + Sieglinde is fast
- + Sieglinde reads:
  - raw data (TWR or muon-DAQ)
  - f2k files (including compressed files)
  - ROOT files
- + Sieglinde writes:
  - f2k files
  - ROOT files
- + ROOT binary files are smaller than F2K files
- + One main program, no piping
- + ROOT is supported
- + Easily handles large files (GB)

Sieglinde and ROOT: cons

- Analysis using complete Sieglinde and ROOT has never been done before
- Variable names and storage structure are very different
- Insufficient documentation
- Some features still not implemented

Fixed problems

- √ All variables not available in Sieglinde ROOT output (n_{hits}, n_{ch}, f_1, etc were implemented by Thomas Burgess)
- √ No info about run number, event number or day in year
- √ Sieglinde could not properly read it's own ROOT files
- √ Selection when writing ROOT output did not give same results as classic Sieglinde

Data sets

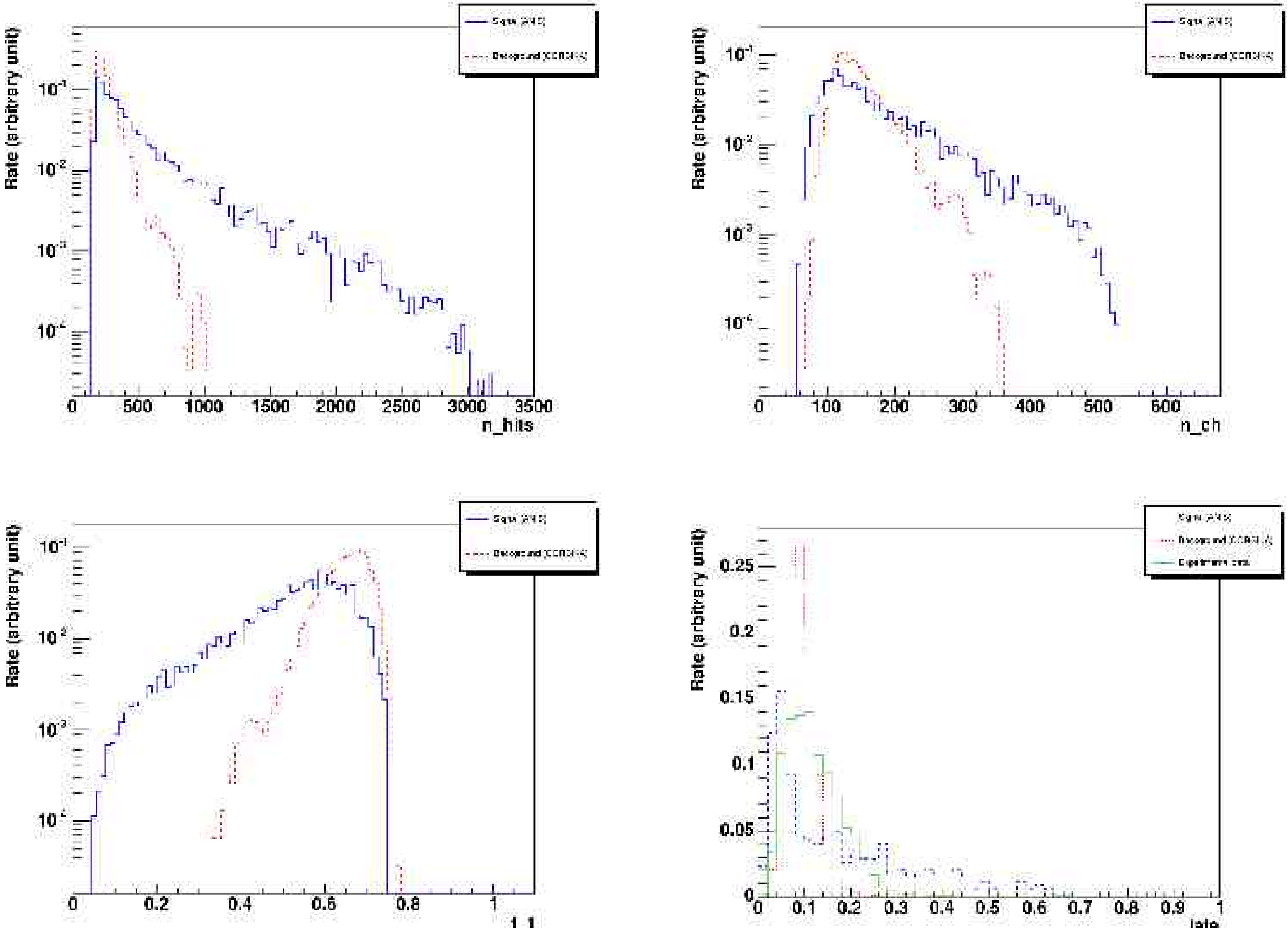
Experimental data: $9.0 \cdot 10^7$ events from 2003, refiltered HE stream (n_{hits}>140, f_1<0.72)

Simulated background: $2.4 \cdot 10^6$ atmospheric muons generated with dCORSIKA

Simulated signal: $1.4 \cdot 10^4$ muon neutrinos following an E^{-1} spetrum generated with ANIS, reweighted to E^{-2} spectrum

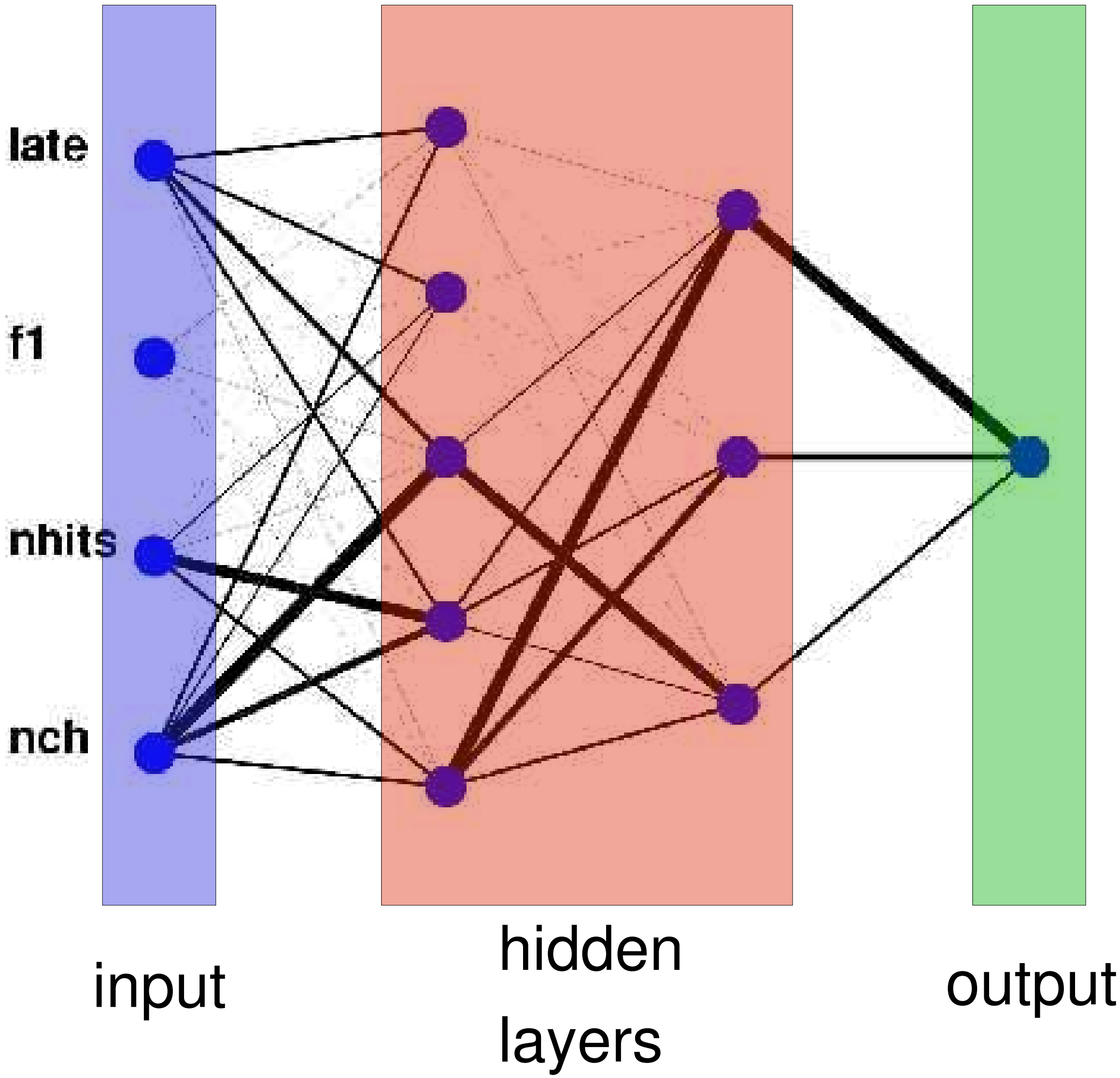
↓

simulated on SweGrid using Simuperl



Rate (arbitrary unit)
n_hits
n_ch
late

Neural net



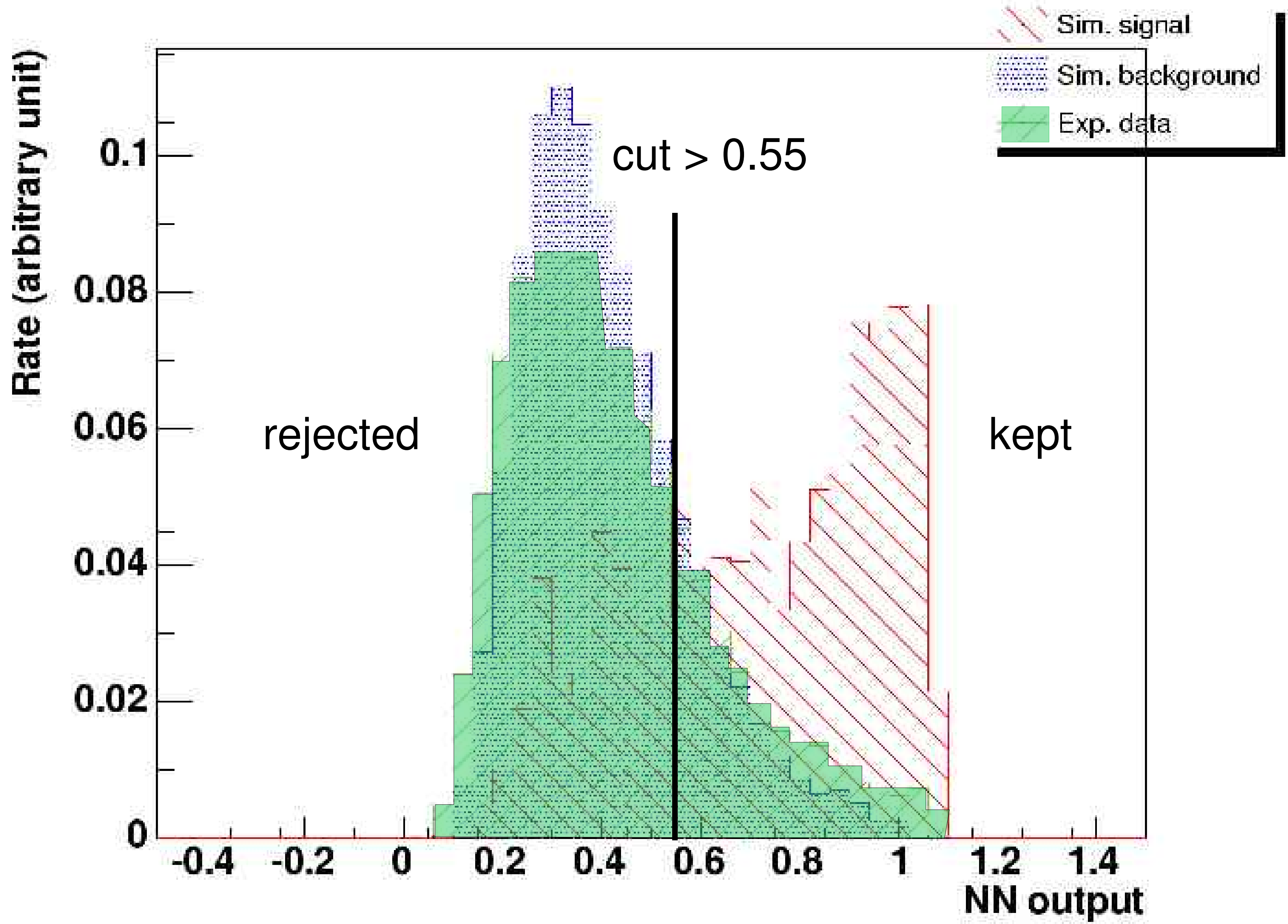
4-5-3-1 layout

100 epochs

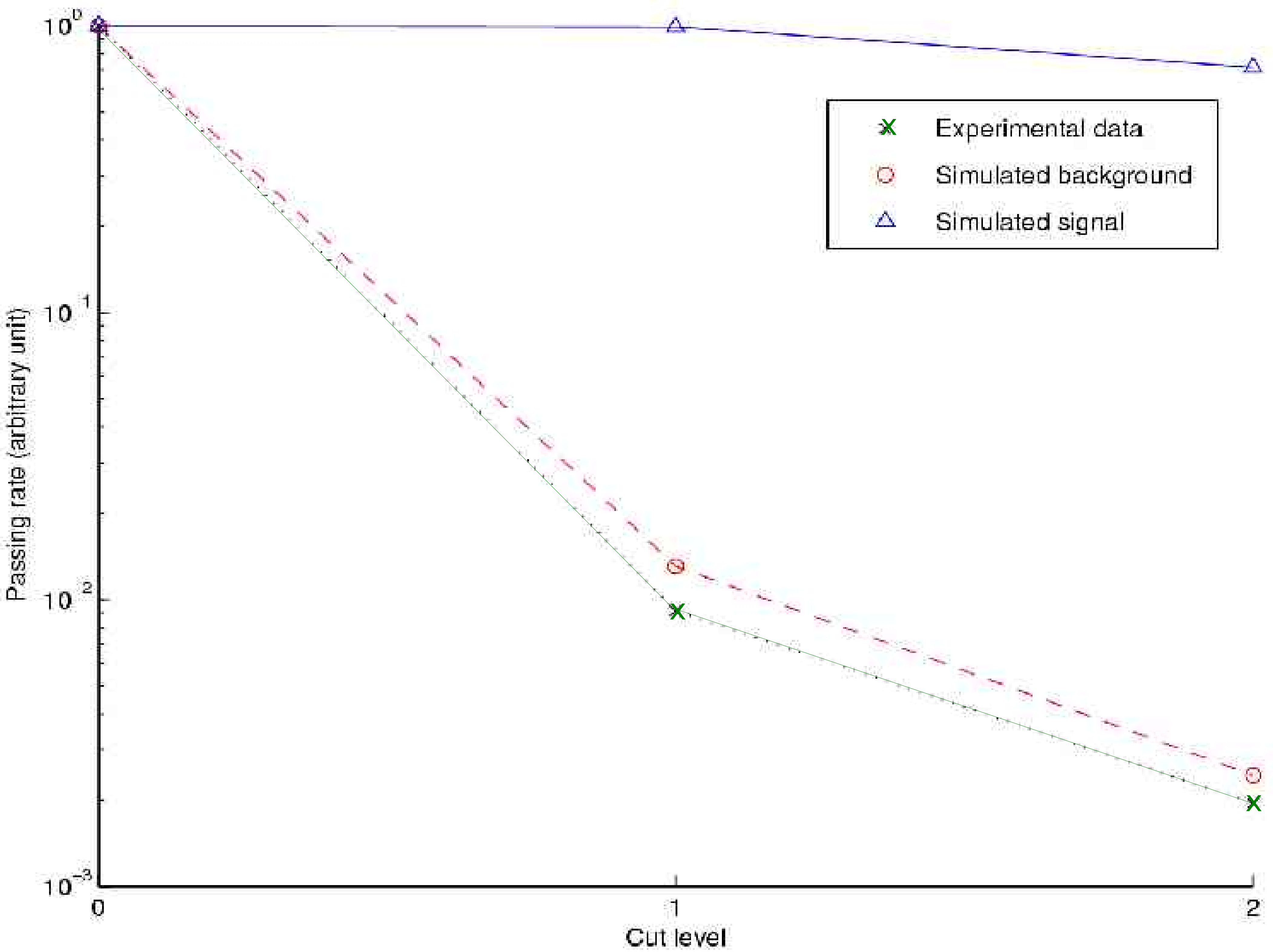
Training samples:

9809 experimental data events (background)

5573 simulated signal events



Sim. signal
Sim. background
Exp. data
Rate (arbitrary unit)
0.1
0.08
0.06
0.04
0.02
0
cut > 0.55
rejected
kept
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
NN output



Experimental data
Simulated background
Simulated signal
Passing rate (arbitrary unit)
Cut level
10^0
10^{-1}
10^{-2}
10^{-3}
0
1
2

Using IcePack to tune MC

Principle:

1. Use IcePack to determine gain for all channels
2. Put gain values in AMASIM input and simulate atmospheric muons
3. Compare noise rate in exp. data and MC
4. Adjust noise level in MC to agree with data