# GLoBES

Patrick Huber

Center for Neutrino Physics, Virginia Tech

What is $\nu$?

Galileo Galilei Institute for Theoretical Physics, June 2012

# What?

## General Long Baseline Experiment Simulator

GLoBES is a software package designed for

- Simulation
- Analysis
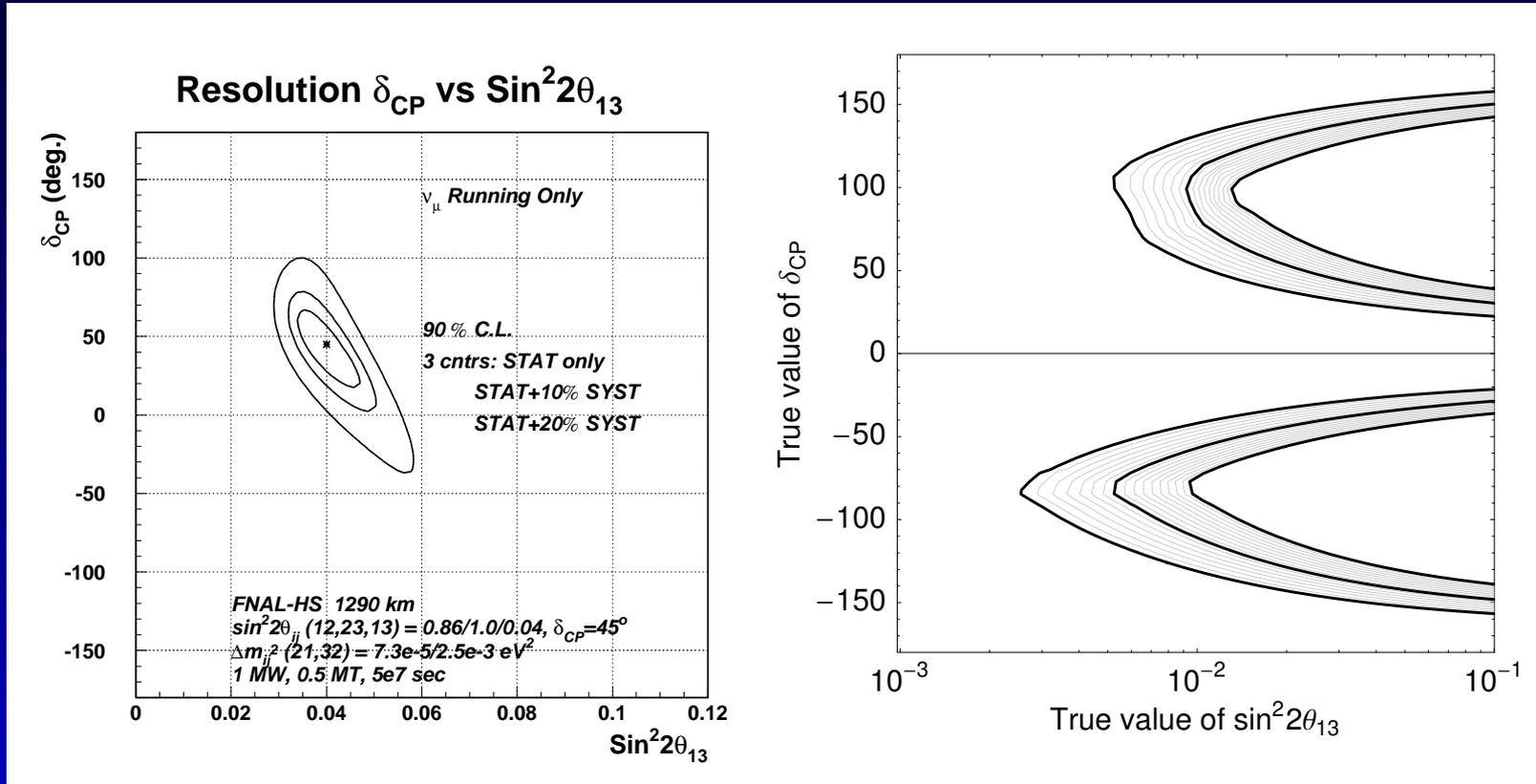- Comparison

of neutrino oscillation experiments

# Where?

It is developed and maintained by

- PH

- Joachim Kopp

- Manfred Lindner

- Walter Winter

URL – http://www.mpi-hd.mpg.de/lin/globes/
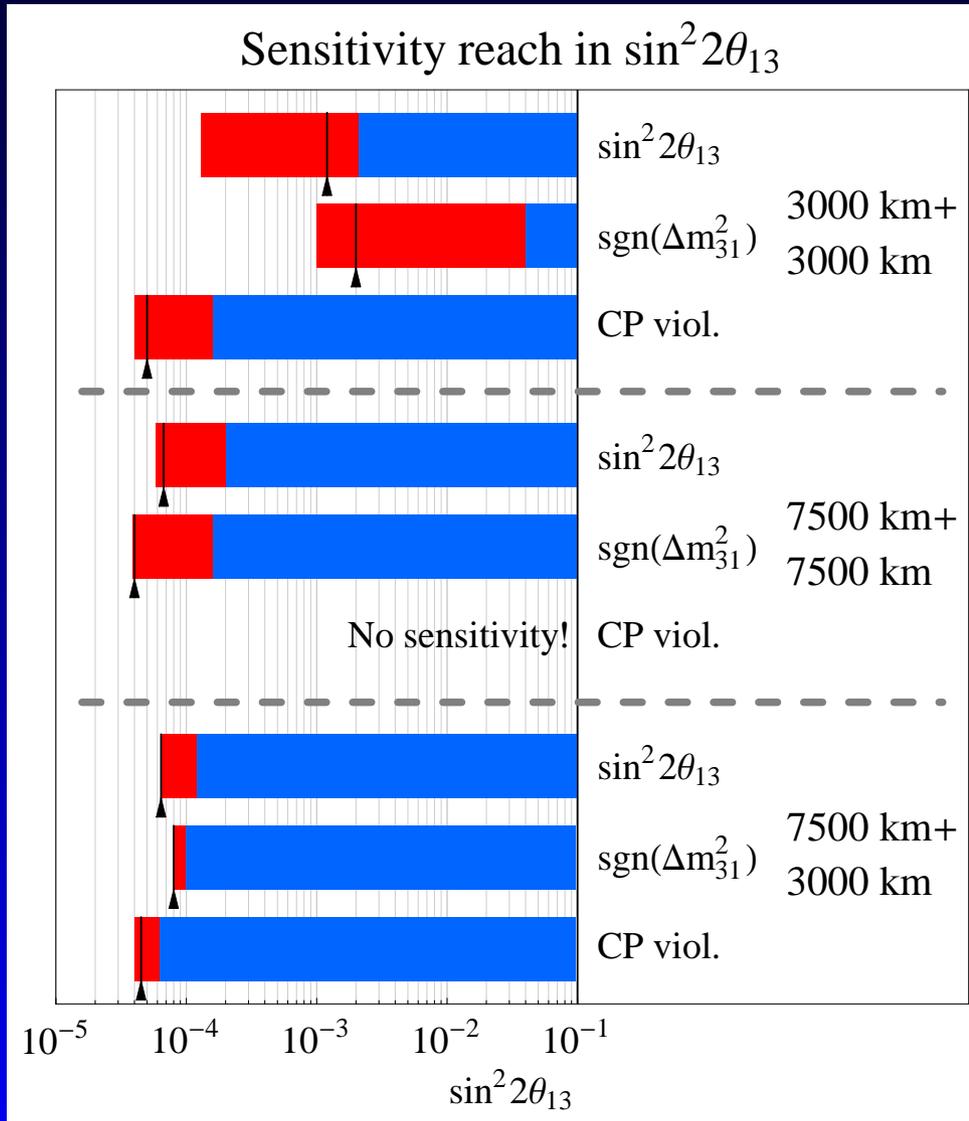email – globes@mpi-hd.mpg.de

# Why?

## Wide band beam

# Why?
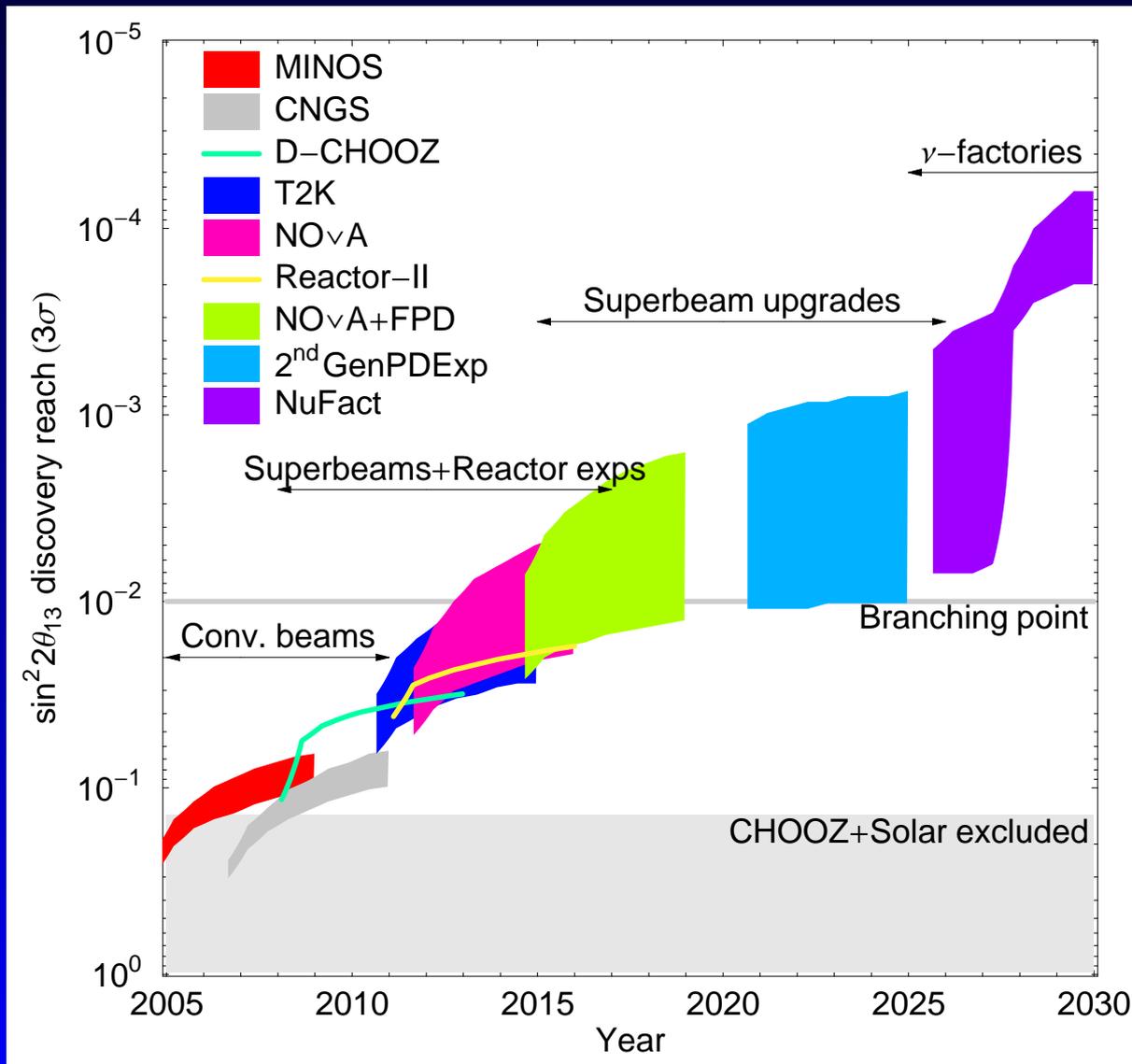
## APS study



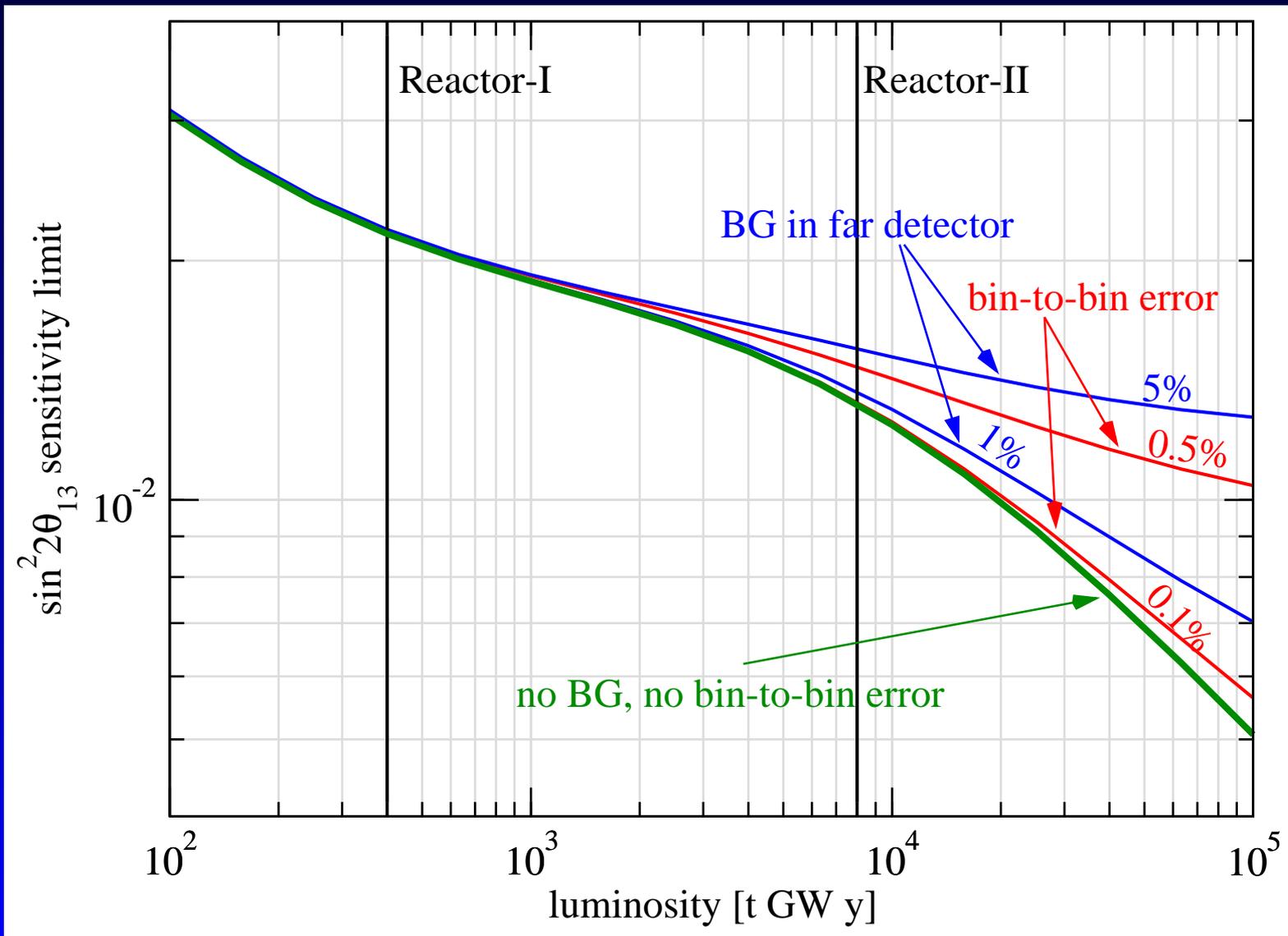Sensitivity reach in $\sin^2 2\theta_{13}$
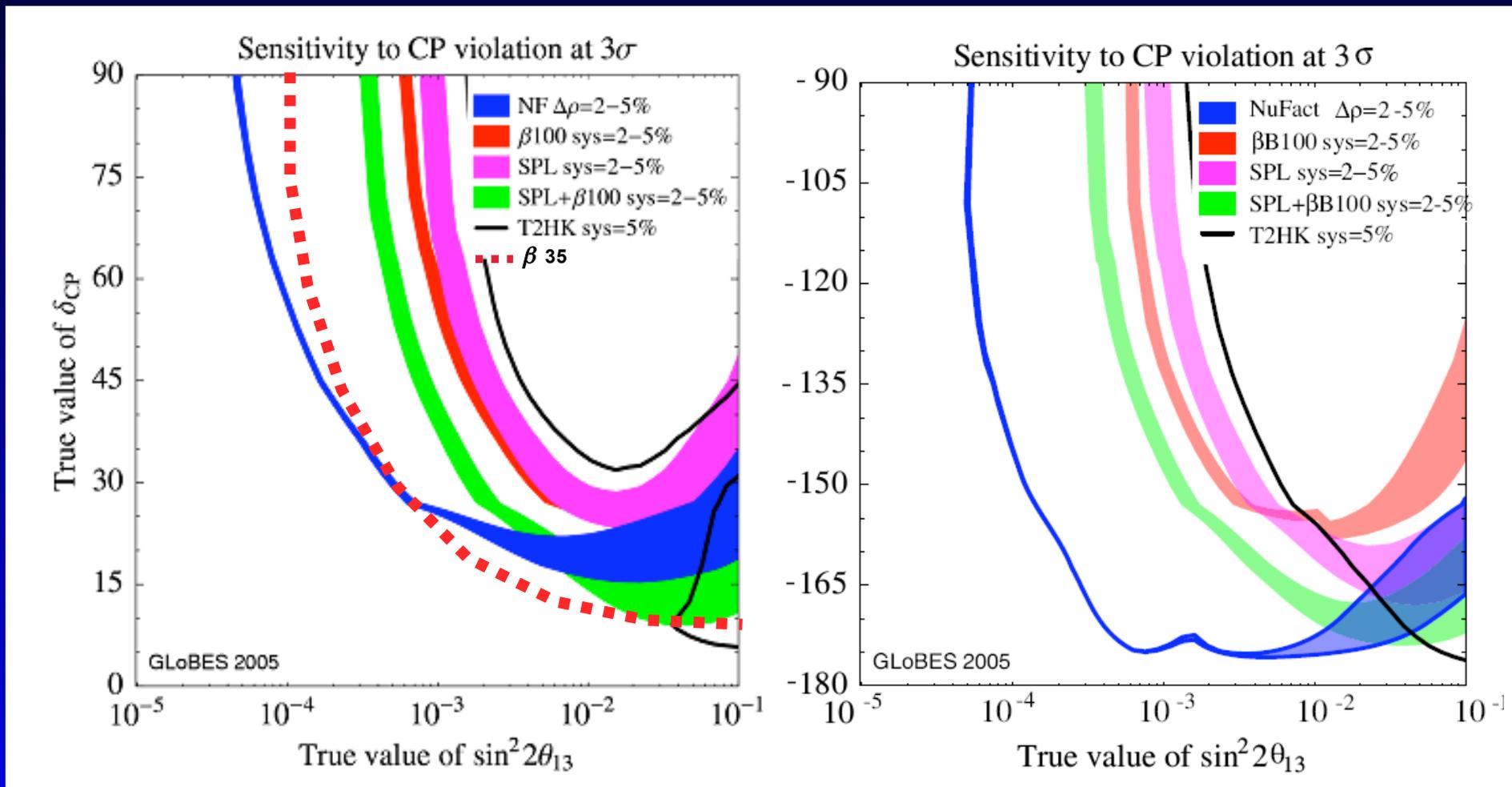
# Why?

## Fermilab's Proton driver report

# Why?

## White paper on reactor neutrinos

# Why?

## CERN strategy group

# Why?

## ISS



GLoBES 2006
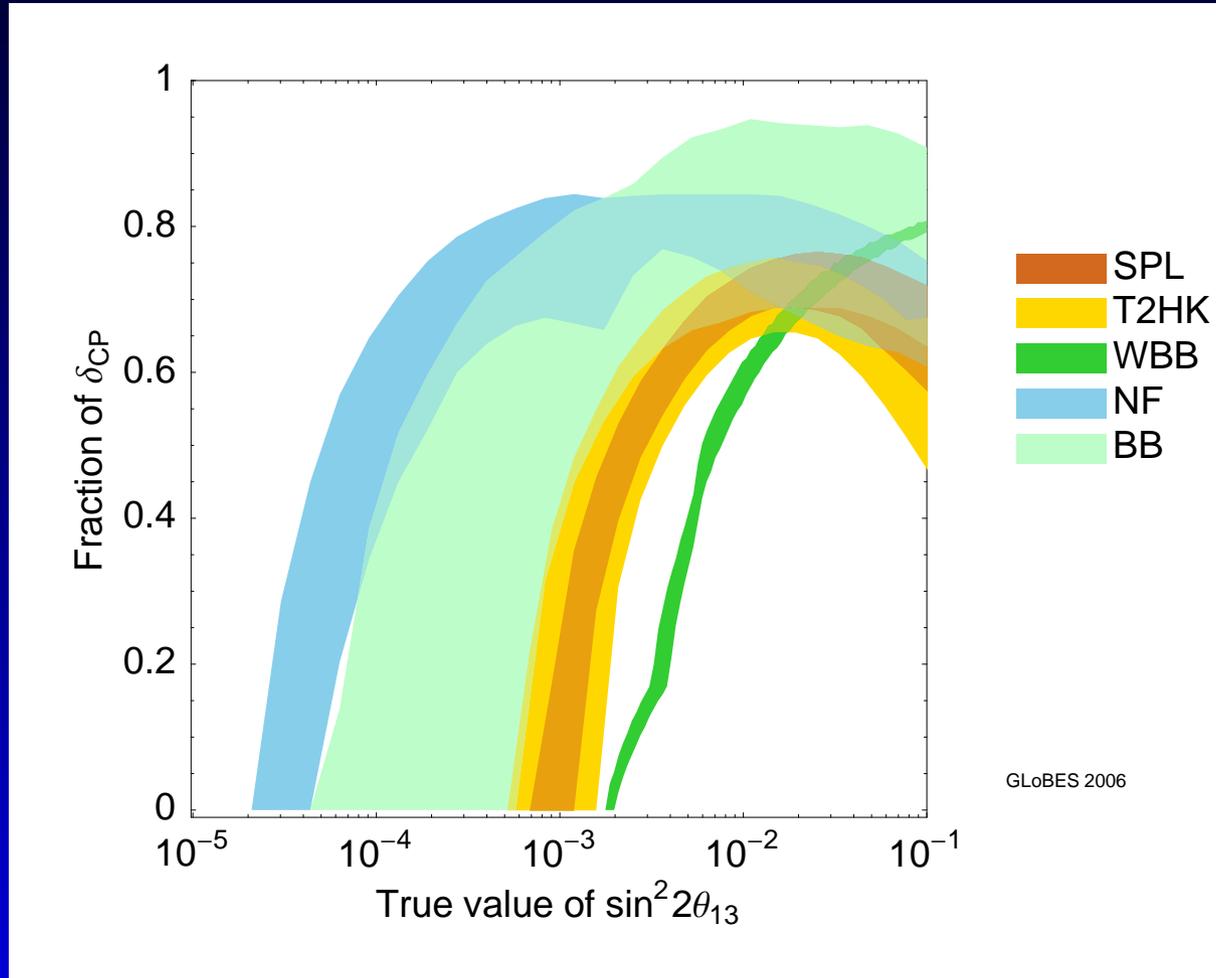
# Why?

## LBNE



Preliminary: LBNE Physics Working Group

5 years neutrino + 5 years antineutrino

# Why?

## MINOS+

# Reliability

- Re-use of code, the more a code has been used in real world applications the less likely are severe bugs.

- Extensive testing

- Good documentation
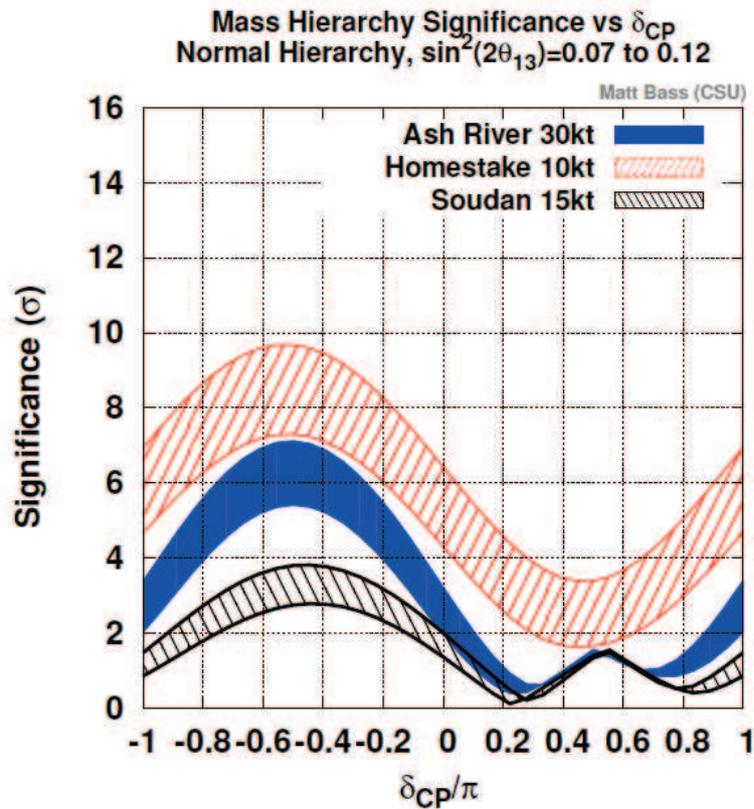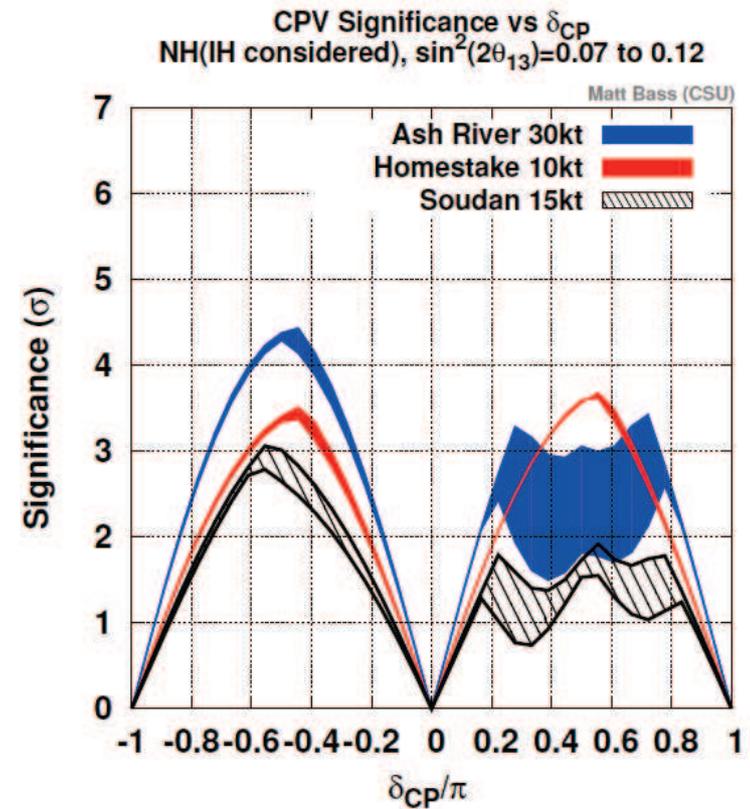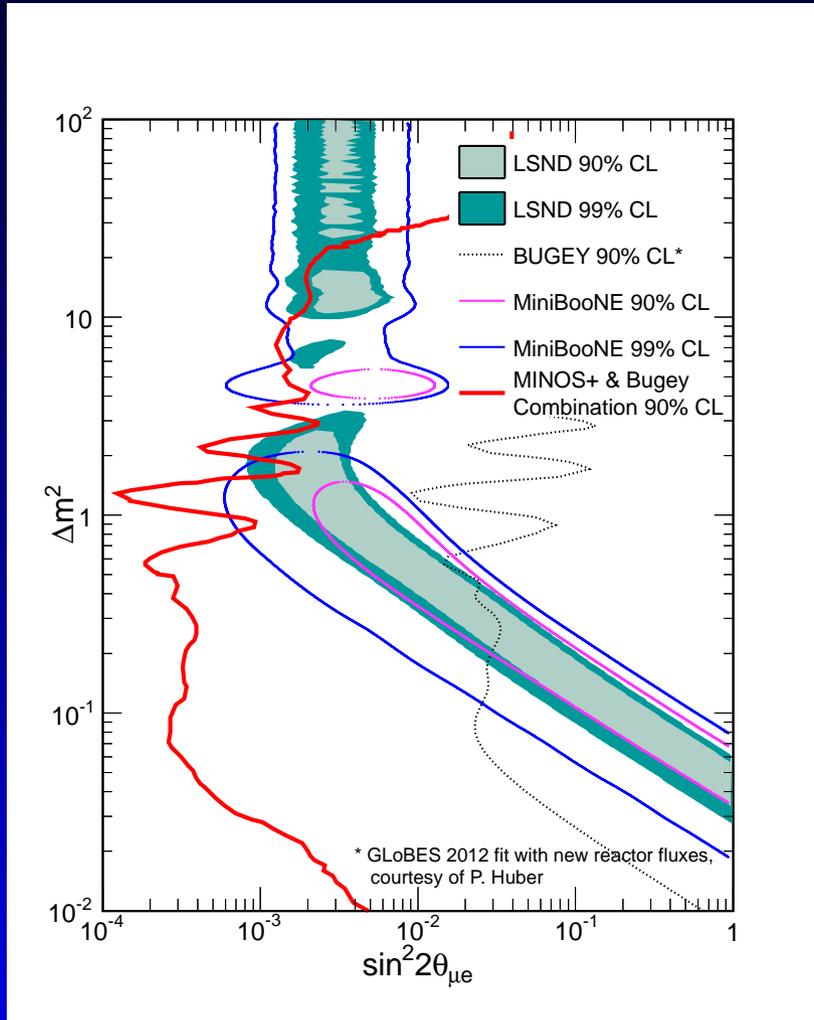
- Intuitive API with error checking

# Reproducibility

The information given in a publication or proposal is not sufficient to reproduce the sensitivity estimates.

- General data storage and exchange format for the inputs $\Leftrightarrow$ flexibility?

- All implicit assumptions and approximations have to be documented, that includes the actual algorithms $\Leftrightarrow$ accuracy of documentation?

- Version control and archiving

# Documentation

Without good documentation, the best software is useless or will be after very short time (=memory decay constant of typical physicist). This is a general problem with legacy code!

Document what you do – do what you document

and make sure that the average user understands what is going on. Also documentation needs testing and debugging.

# GLoBES history

- development started 2004 – PH, M. Lindner, W. Winter

- major effort went into documentation

- first release August 2004 – version 2.0.0

- major bug fix release March 2005 version 2.0.11

- J. Kopp joined in July 2005

- January 2007 – version 3.0, addition of major features

- >180 publications citing the GLoBES papers, creating a total of >3000 citations

- fall 2012 – GLoBES 4.0

# Design considerations

- GPL

- C-library – very portable, easy to interface, numerically efficient

- Unix style separation of functionality – freedom to design analysis and to use any graphics tools

- Experiments are defined using AEDL – relatively complicated parser, transparent experiment definition

- Pull approach for systematics – flexible and intuitive

- Local minimization instead of grids – much faster

# Features

- Accurate treatment of systematical errors
- Arbitrary matter profile & uncertainties
- Arbitrary energy resolution function
- Single and multiple experiment simulation
- Simple $\chi^2$ calculation
- Inclusion of external input
- Projection of $\chi^2$ (minimization)
- User-defined systematics, oscillation probability engine, priors
- Full support for lists in AEDL
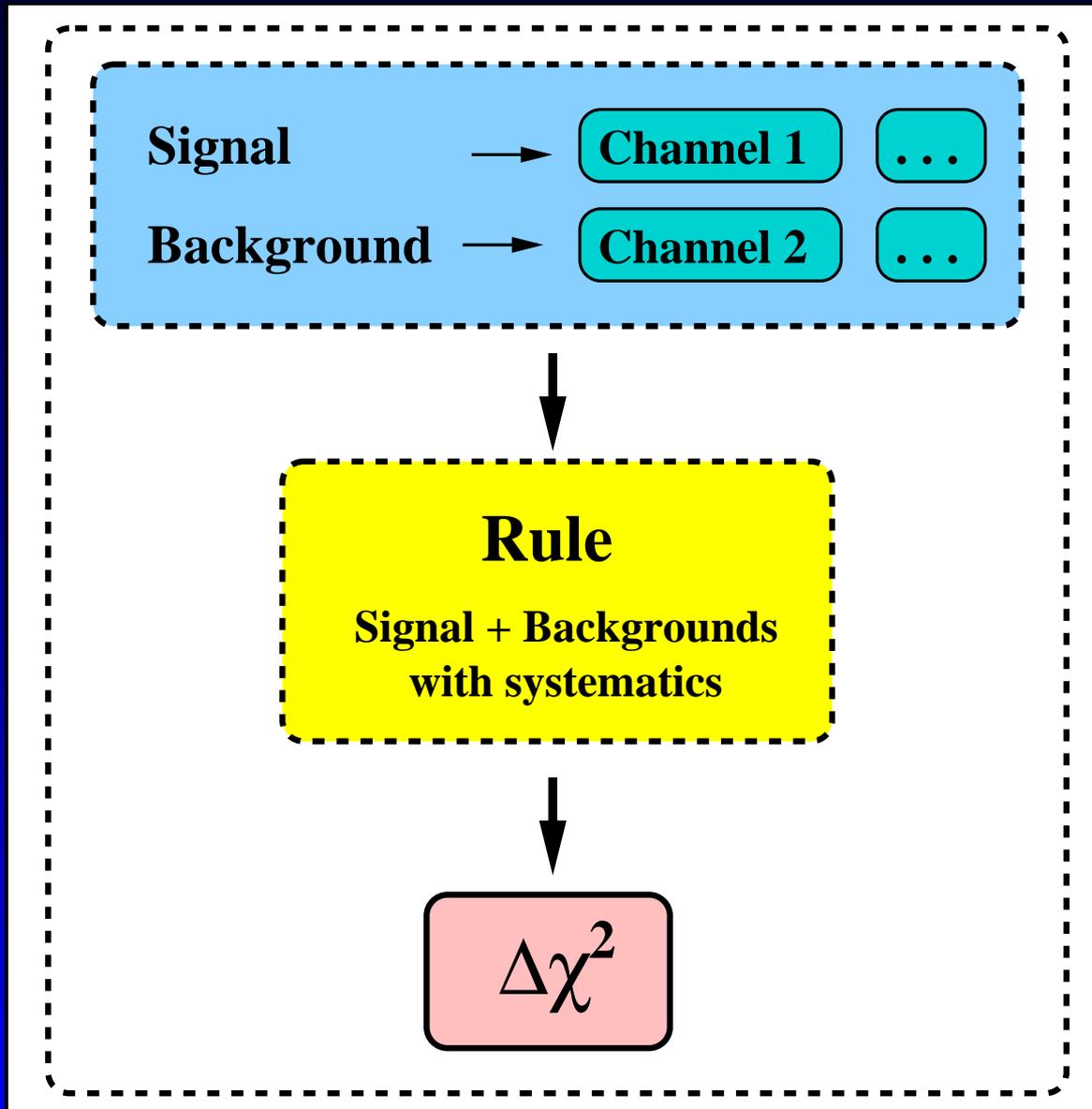- Interpolating functions in AEDL
- . . .
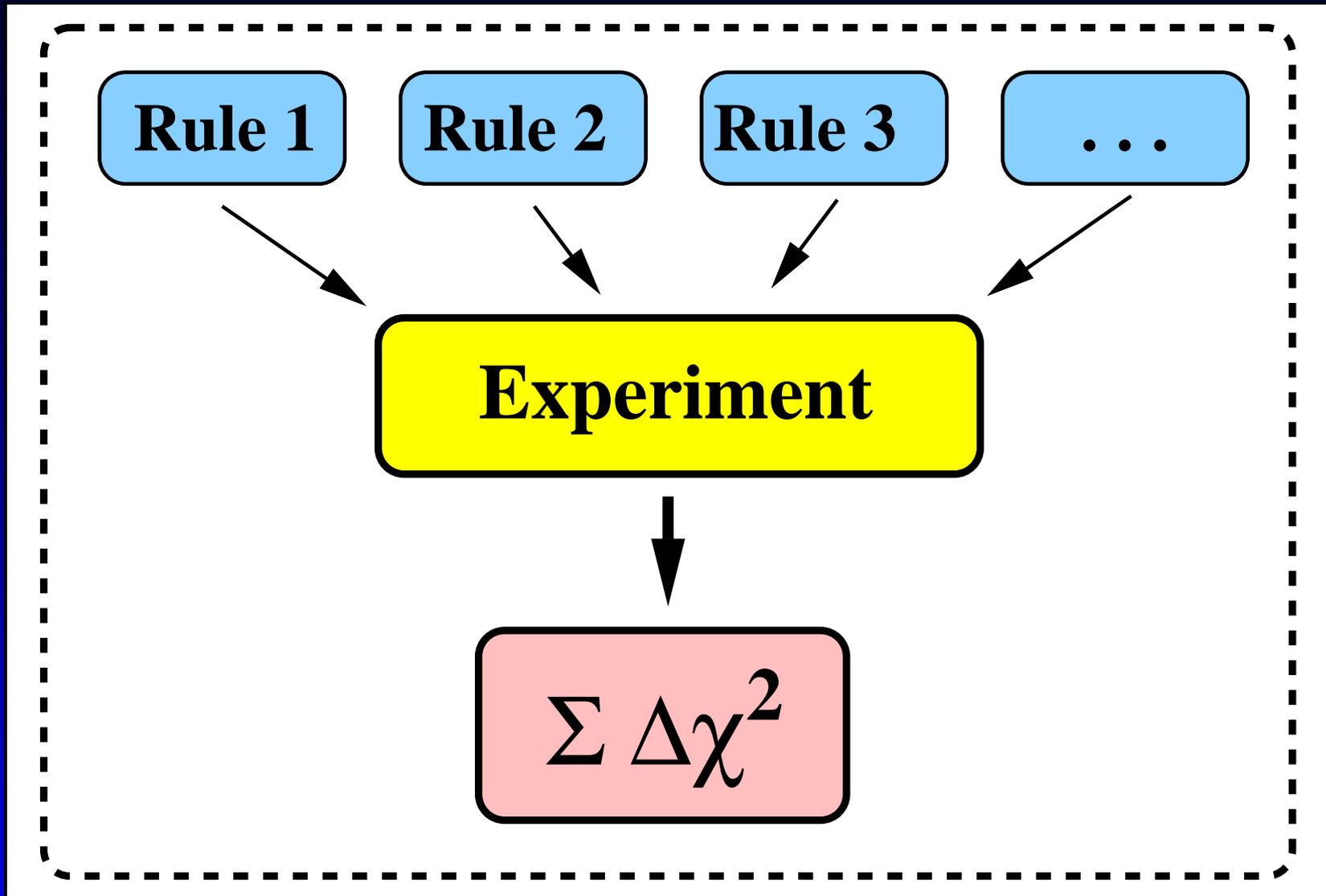
# GLoBES – overview
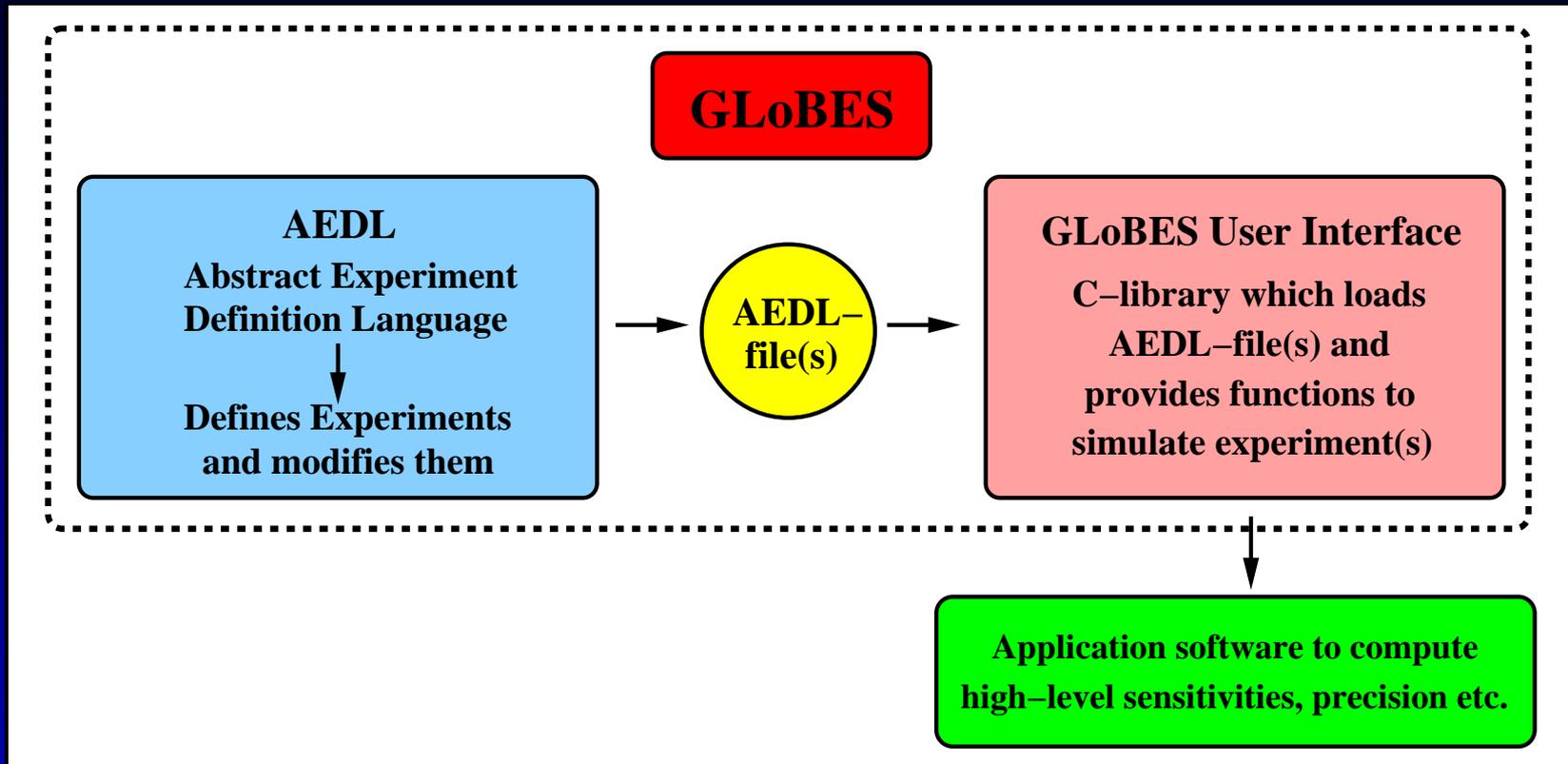
# GLoBES – AEDL
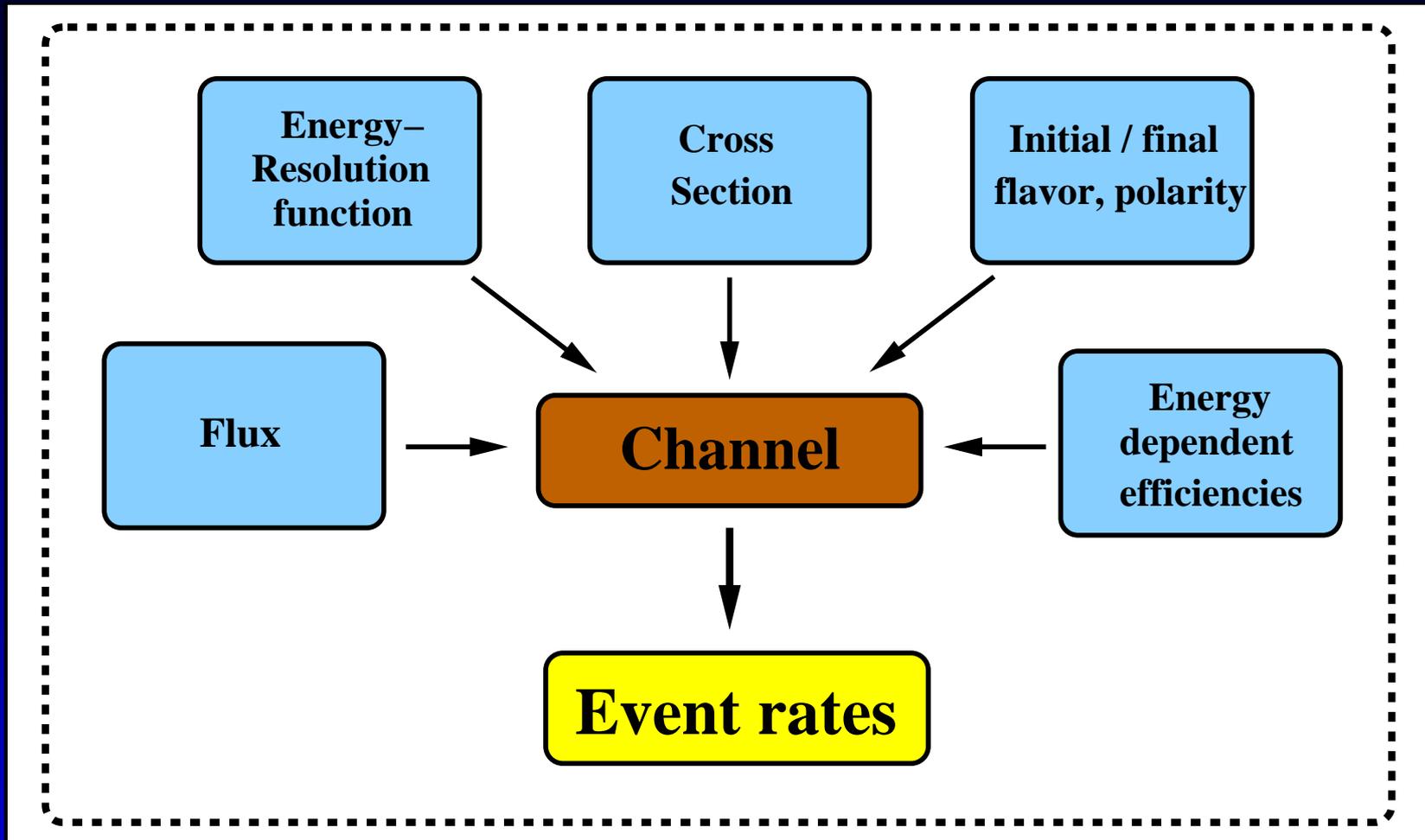
# GLoBES – AEDL

# GLoBES – AEDL

# Summary

GLoBES

- is the only open source software of its kind

- has withstood the test of time

- is at the core of most strategy documents

- completely in C

- flexibility to deal with complex many detector setups and non-standard physics

- v4.0 will greatly enhance the ability to deal with realistic systematics

# Installation

If you have Linux

- Install GSL if you don't have it already – ftp://ftp.gnu.org/gnu/gsl/

- Go to http://www.mpi-hd.mpg.de/personalhomes/globes/download/globes-3.1.11.tar.gz
– download GLoBES

- ./configure
make
sudo make install

# Installation

If you have a Mac

- Install GSL if you don't have it already – ftp://ftp.gnu.org/gnu/gsl/

- Download http://www.mpi-hd.mpg.de/personalhomes/globes/download/globes-3.1.11.tar.gz

- unpack it and change into the directory created by this

```
./configure --disable-rpath --enable-no-binary=yes
        make
        make install
```

# Installation – Mac issues

On some of them, there are problems with the architecture (32 vs. 64 bit) since Mac OS tries (and fails) to allow mixing 32 and 64 bit code. The solution to this is to compiile both the GSL and GLoBES with the `-m32` option enabled.

Finally, people sometimes have trouble compiling the examples, which is due to some yet unresolved problem in our autoconf script. The solution is to modify the Makefile in the examples directory in such a way that `globes-config --libs` is not used. Instead, use whatever options `globes-config --libs` proposes, minus any `--rpath` options.