

# AEDL 101

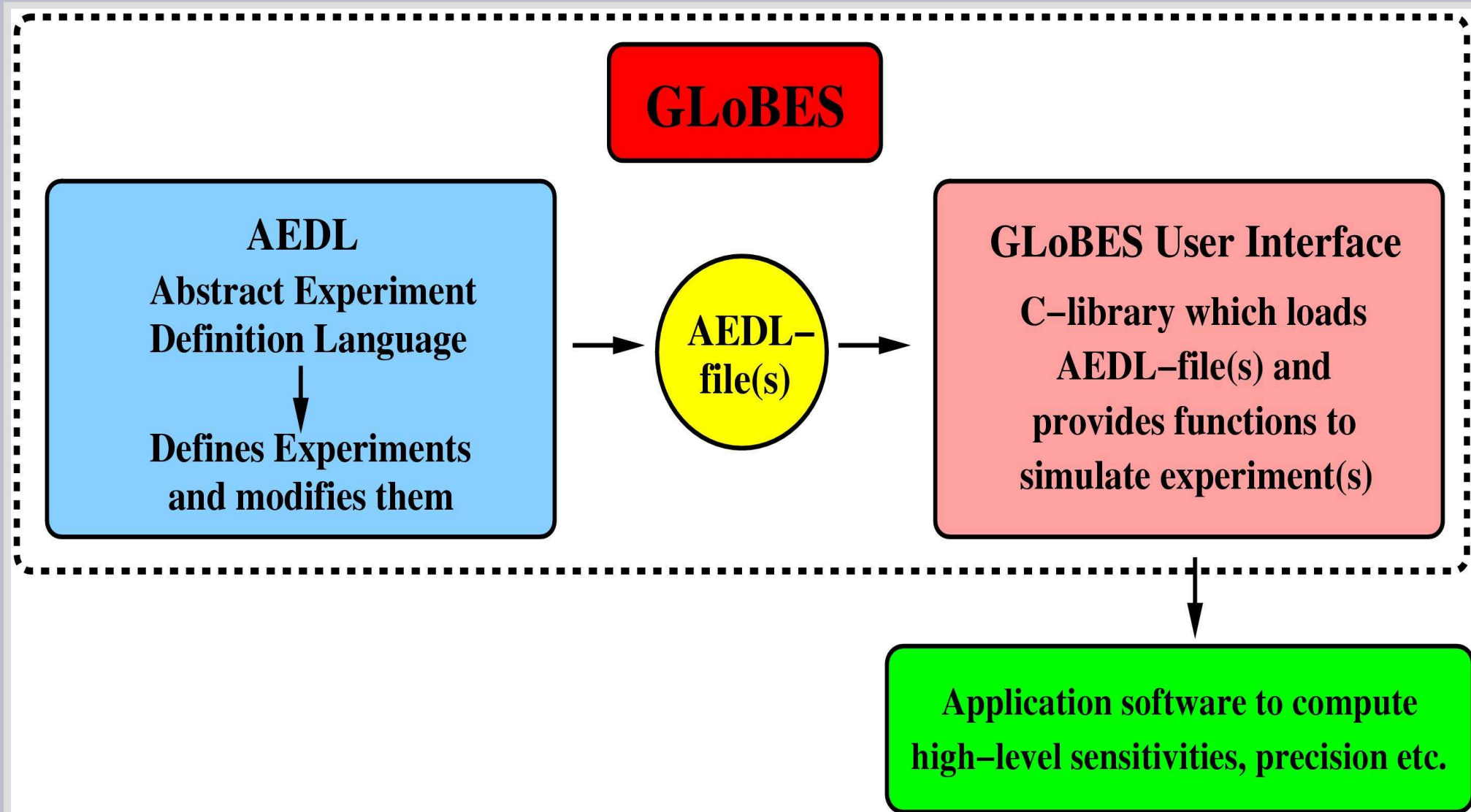
or how to put your experiment into GLoBES

P. Huber  
University of Wisconsin – Madison  
on behalf of  
The GLoBES Team

# Outline

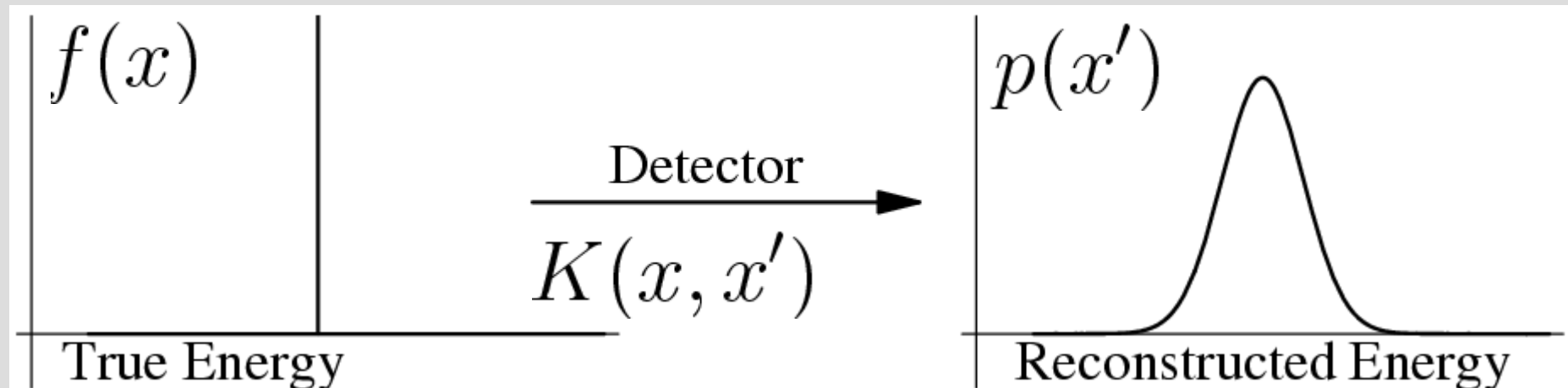
- Basic concept behind GLoBES
- Step by step example
- The Manual is your friend!
- Friends sometimes let you down
- In that case:  
`globes@ph.tum.de`

# Basic Concept



# What does a detector do?

- It maps some incident, true particle flux  $X$  into a detected, reconstructed flux  $Y$
- The particles in  $X$  and  $Y$  may be different
- There may be many different  $Y$ 's for one  $X$



$$p(x') = \int dx f(x) K(x, x')$$

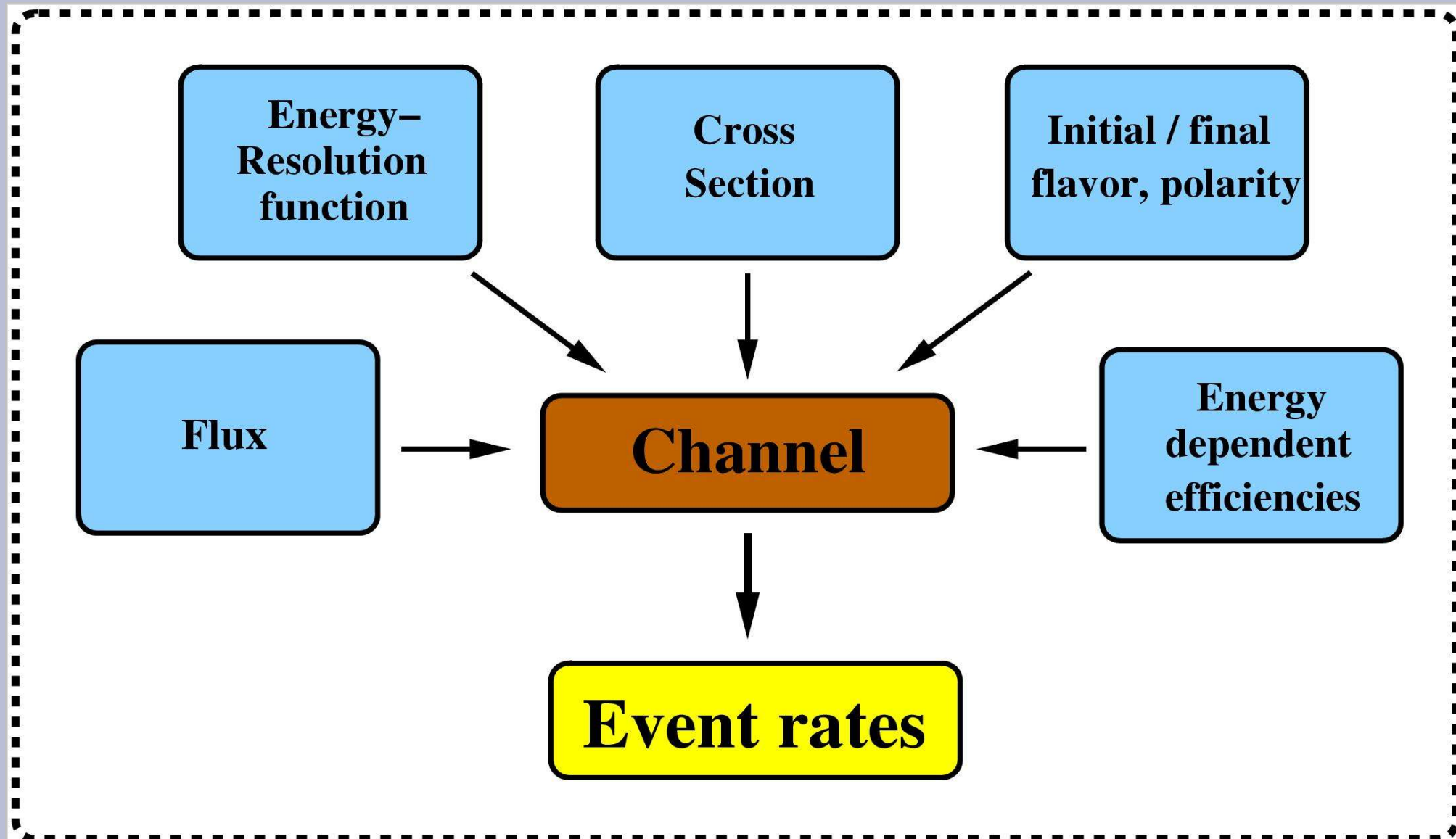
# Intermediate states?

- This mapping usually involves intermediate states, eg. hadron showers, muons asf.
- They are only important in as far as they determine the properties of the mapping function
- Goal: Fast detector “simulation”

# Detectors in GLoBES

$$\begin{aligned}
 \frac{dn_{\beta}^{\text{IT}}}{dE'} = & \quad N \int_0^{\infty} \int_0^{\infty} dE d\hat{E} \quad \underbrace{\Phi_{\alpha}(E)}_{\text{Production}} \times \\
 & \underbrace{\frac{1}{L^2} P_{(\alpha \rightarrow \beta)}(E, L, \rho; \theta_{23}, \theta_{12}, \theta_{13}, \Delta m_{31}^2, \Delta m_{21}^2, \delta_{\text{CP}})}_{\text{Propagation}} \times \\
 & \underbrace{\sigma_f^{\text{IT}}(E) k_f^{\text{IT}}(E - \hat{E})}_{\text{Interaction}} \times \\
 & \underbrace{T_f(\hat{E}) V_f(\hat{E} - E')}_{\text{Detection}},
 \end{aligned}$$

# AEDL / Channels



# AEDL Channels cont.

- Different intermediate states may yield a different mapping
- There may be some events cleaner than others, eg. QE
- Different sources for signal and backgrounds
- Different event types, eg. QE vs NC
- Currently up to 32 channels per experiment

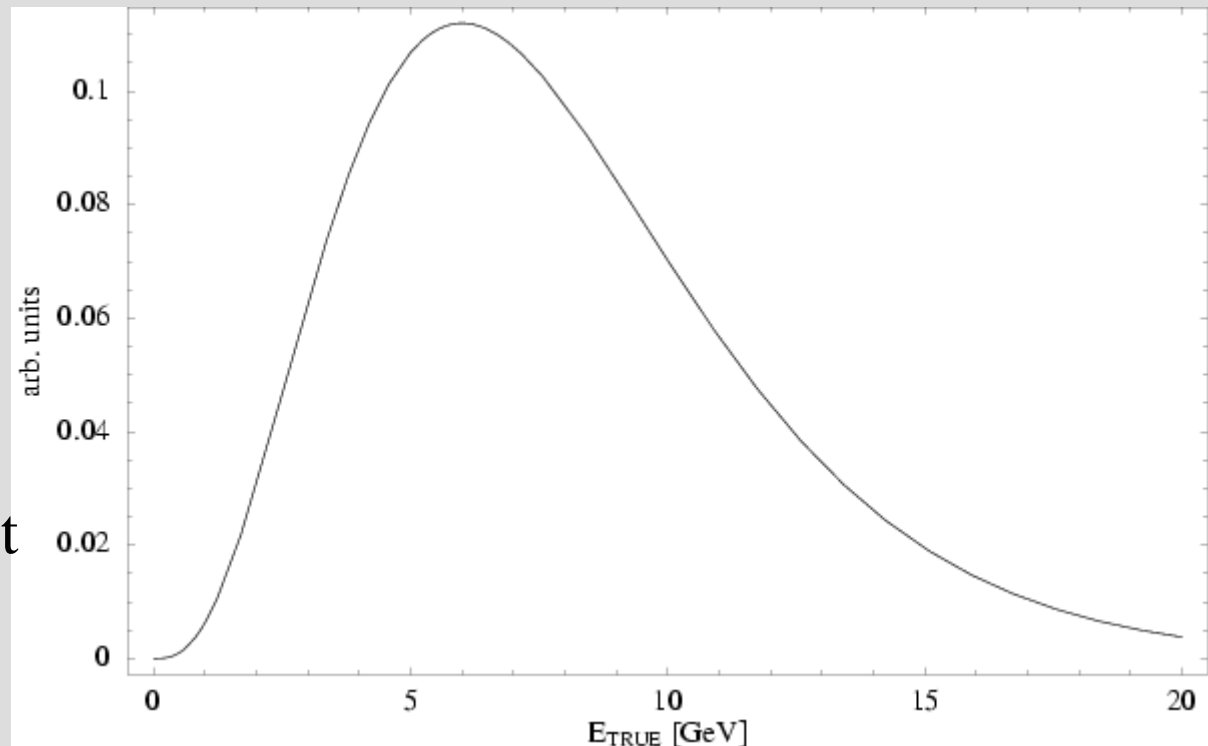


# Example

- From here on I will present a toy example
- The detector Monte Carlo is a black box
  - I made up a resolution function etc
  - I then “generated” events
  - and arbitrarily tagged them as 'signal' or 'garbage'
- How to put each of the parts into AEDL syntax, is better explained in the Manual

# Flux

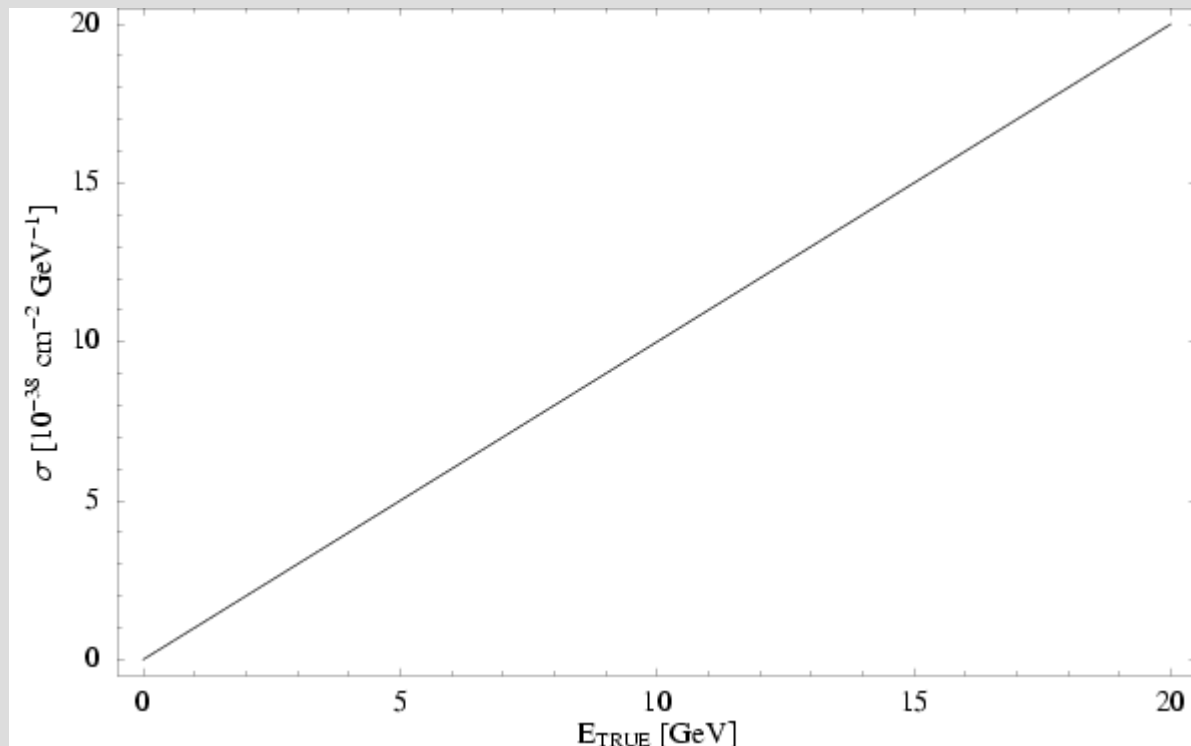
- Flux either given by formula (NF, Beta-Beam)
- or MonteCarlo (superbeams)



Warning!  
The normalization constant  
is not documented. There  
is a factor of 5.20... which  
has to be properly  
accounted for!

# Cross section

- From calculation (rarely)
- From fit to data
- Event generator, eg. NUANCE



# Mapping function

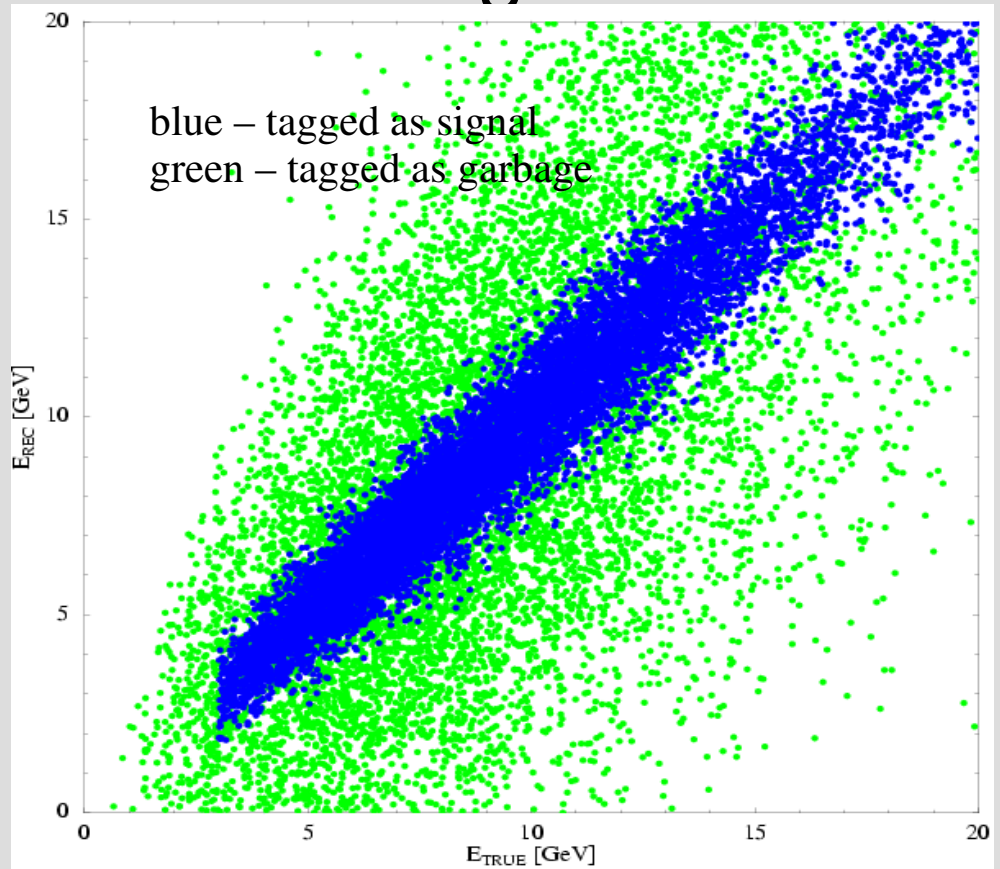
- Has to be derived from MC
- either parametrically, eg. Gaussian resolution
- or in form of migration or smearing matrices

Typical output from MC

Draw true energies from flux  
times x-section

Follow events through detector

Assign to each event the  
reconstructed energy and event  
type

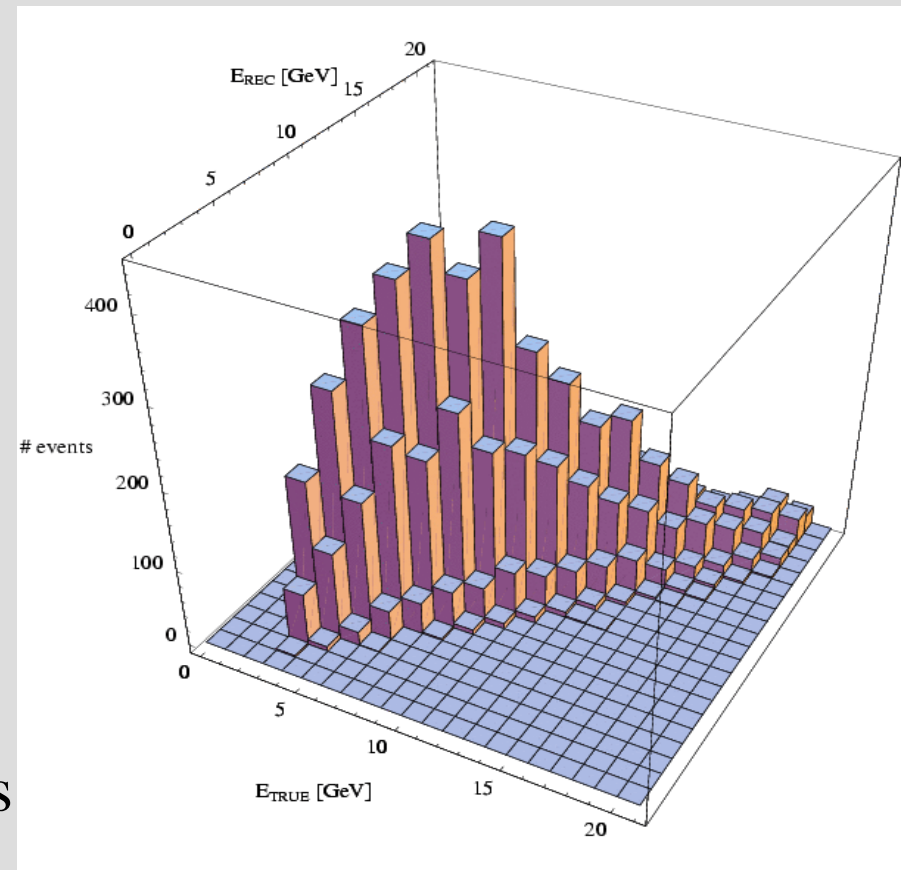


# Migration matrix for the signal

- Take the blue points
- Bin them in 2D bins
- Raw matrix

$$E_{\text{REC}} \begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 7 & 9 & 4 & 1 \\ 0 & 0 & 2 & 9 & 7 \end{pmatrix} E_{\text{TRUE}}$$

bins in true energy: sampling points  
bins in rec. energy: bins



# Reweighting

- Raw matrix has to be reweighted
  - flux used for MC will be modulated by oscillations
  - number of events in MC does not correspond to actual number, eg. I generated 20000 events, but the real experiment may have only 500
  - number of generated events usually is not the same than the number of events which are reconstructed as signal (efficiency)

# Reweighting cont.

Sum each column

$$n_i = \sum_{j=1}^{N_{\text{bins}}} m_{ij}$$

Divide each column by that sum

$$m_{ij} \rightarrow m_{ij} / n_i$$

Account for 'lost' events, efficiency

$$\epsilon = \frac{\sum_{i=1}^{N_{\text{sampling}}} n_i}{\text{number of generated events}}$$

# Computing events

- Event computation reduced to matrix multiplication

$$\Phi(E) = P(E) \times \sigma(E) \times \phi(E)$$

Can be regarded as vector and thus we can compute the events N

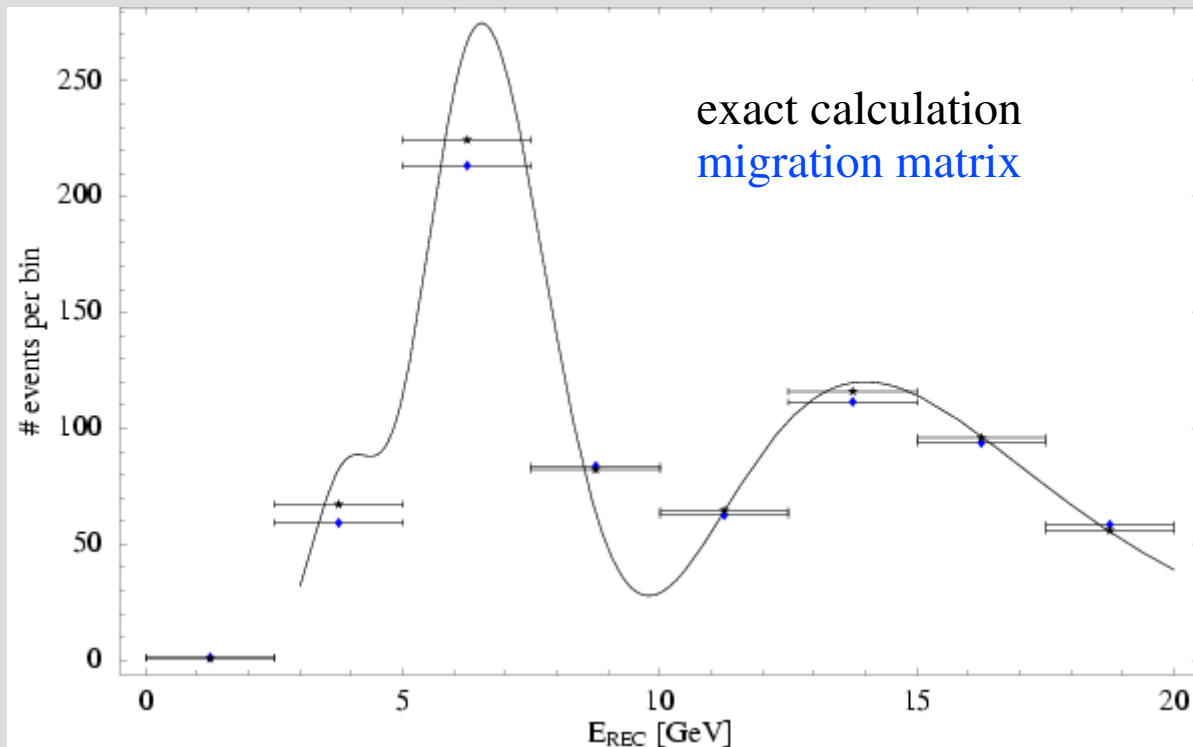
$$N_j = \sum_{i=1}^{N_{\text{sampling}}} m_{ij} \cdot \Phi_i$$



# Computing events

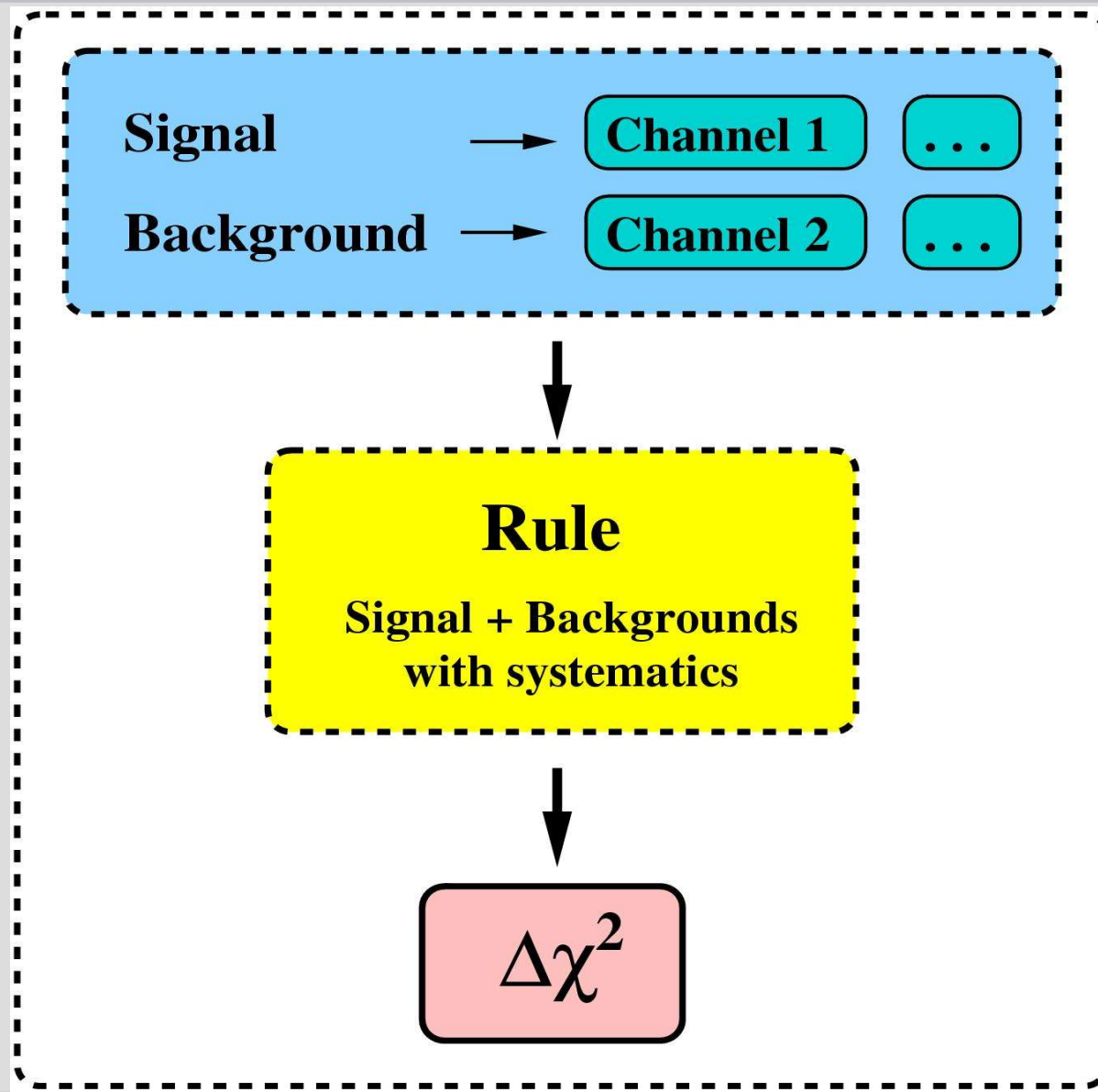
Toy oscillation probability

$$P(E) = \sin^2 \left( \frac{30.0}{E} \right)$$

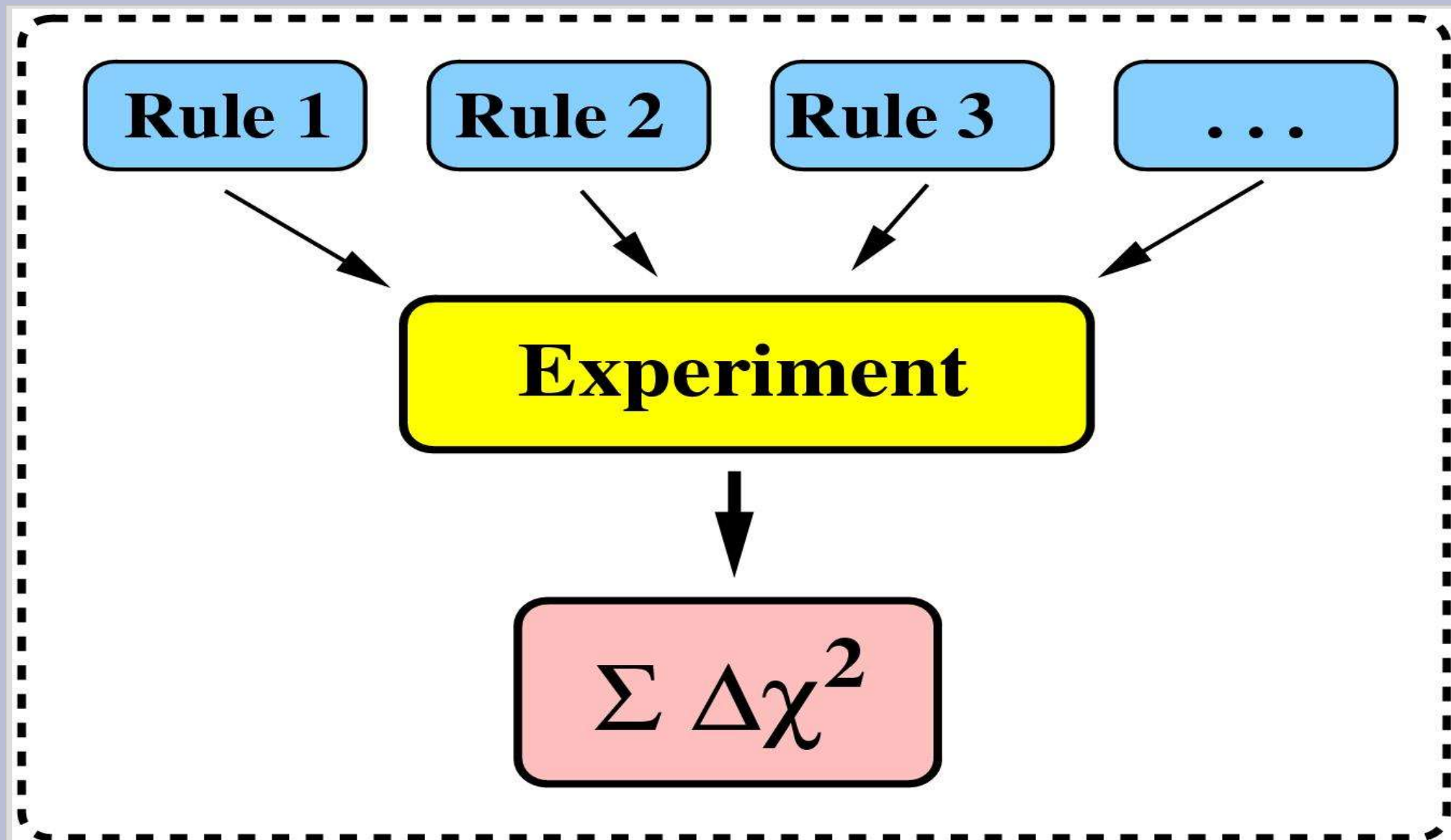


That was only 1 channel – next you would do the same for all the backgrounds

# AEDL / Rules



# AEDL / Experiment



# Debugging AEDL

```
> globes -s MINOS.glb
```

```
----- #rule0 -----
```

1.083	0.0603374
1.25	0.254606
1.417	0.663695
1.583	1.35012
1.75	2.31606
1.917	3.49245
2.083	4.76538
2.25	6.02335
2.417	7.18871
2.583	8.21697
2.75	9.08048
2.917	9.75821
3.083	9.44848
3.25	9.00641
3.417	8.46126
3.583	7.84135
3.75	7.17314
3.917	6.48066
4.083	5.7852
4.25	5.10508

4.417	4.45543
4.583	3.8479
4.75	3.29056
4.917	2.78792
5.083	2.3413
5.25	1.94942
5.417	1.60913
5.583	1.31619
5.75	1.0659
5.917	0.853586

```
-----  
Total: 135.989
```

```
.....  
.....  
....  
.  
.
```

Interactive call of globes

=> event rates

globes has many command  
line options

# Summary

- GLoBES tries to implement a **fast** simulation
- Generating migration matrices from MC output is straightforward, but requires care
- Channels offer quite some flexibility
- If all else fails, look into the manual
- There is an AEDL debugging tool, `globes`
- [globes@ph.tum.de](mailto:globes@ph.tum.de)