Status of CORRFW

Renaud Vernet (INFN Catania)

- June 25th , 2009 -

renaud.vernet@cern.ch
ingrid.kraus@cern.ch

Outline

- *feedback from users*
- *new N-dimension unfolding*
- *new CORRFW design*

Most recent requests from users

- New requests mainly came from PWG3 people
  - di-muons — R. Arnaldi, L. Bianchi, X. Lopez
  - electrons — R.Bailhache, S. Masciocchi, F. Kramer…
- Other topics
  - p-pbar — M. Broz, M. Chojnacki, P. Christakoglou
  - resonances — A. Pulvirenti

More people should integrate the correction services in their train task!

New N-dim. unfolding

(already presented in PWG1 @ 22-04-2009)

- Need for a N-dimension unfolding procedure that
  - is not dependent on the assumed Monte Carlo used for efficiency calculation
  - is usable on any kind of spectrum to correct
  - has no limitation in the number of dimensions
  - must handle spectra with a very LARGE number of bins
- Such a code goes into AliRoot CORRFW directory
  - must be compatible with the objects used to handle efficiencies and spectra
    - basically AliCFContainer, AliCFEffGrid, AliCFDataGrid
  - common basis : ROOT's THnSparse class
    - efficient use of memory for very large matrices sparsely filled
- Already-existing ROOT-based unfolders can be found
  - but none (that I know…) supports more than 2 dimensions nor THnSparse
  - ⇒ need to write a new class

something like that should be

used to correct the data

=> try this one !

New N-dim. unfolding

- Idea proposed by d'Agostini (Nucl.Instrum.Meth.A362:487-498,1995)

Eq. 1 $P(T)=P(T/M)*P(M)$

P(T) : the true distribution you want to know
P(M) : the measured distribution
P(T/M) : you don't know

T=true
M=measured

but... Bayes saved us :

Eq. 2 $P(T/M)=\frac{P(T\cap M)}{P(M)}=\frac{P(M/T)*P(T)}{\sum P(M/T_k)*P(T_k)}$

P(M/T) : detector response you know via simulation
P(T) : that's what you want to know, but you can make an "a priori" hypothesis $P_0(T)$

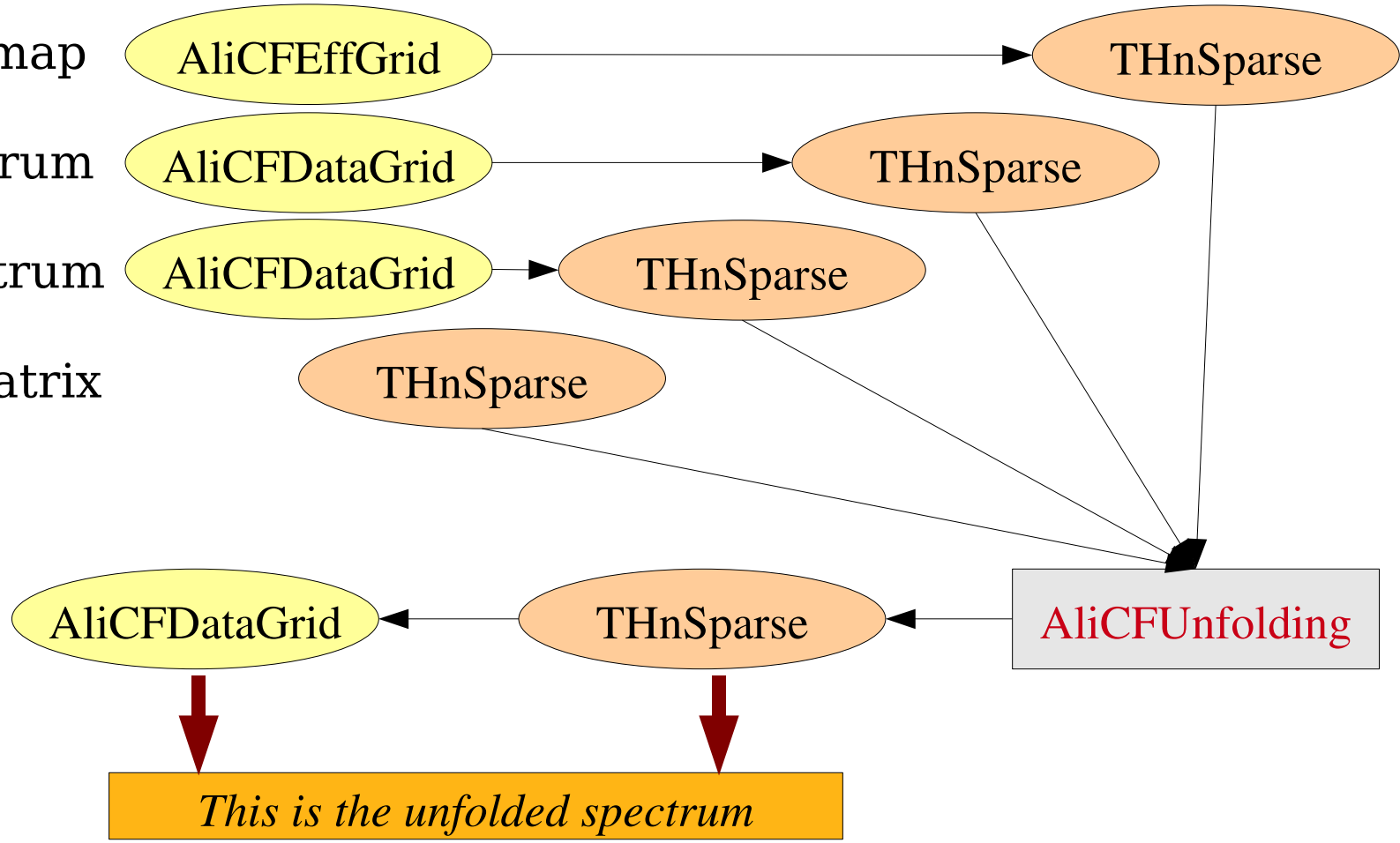
⇒ so you calculate $P_1(T)$ given $P_0(T)$ using Eq. 1&2
if $P_1(T) \sim P_0(T)$: you're done
if $P_1(T) \neq P_0(T)$: you must re-iterate using $P_1(T)$ as "new a priori" distribution etc.

- this method is general enough to be used in as many dimensions as desired
- it is already used in ALICE
  - PWG0/AliUnfolding (1D)
  - PWG4/JetTasks/AliJetSpectrumUnfolding (2D)

New N-dim. unfolding

user input :

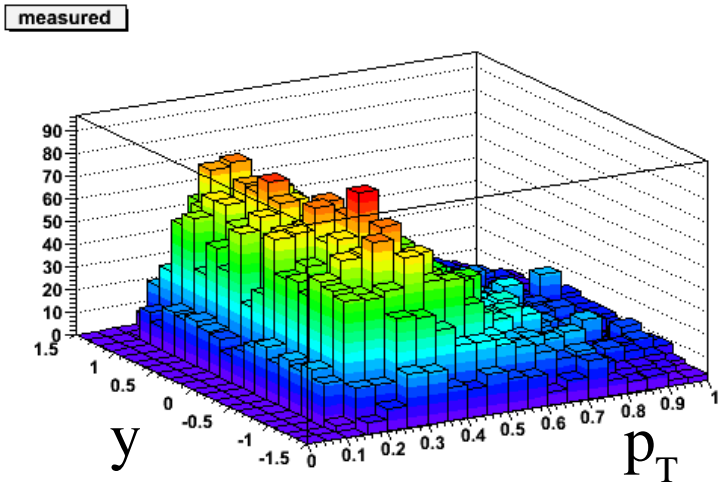
- efficiency map
- measured spectrum
- "guessed" spectrum
- response matrix

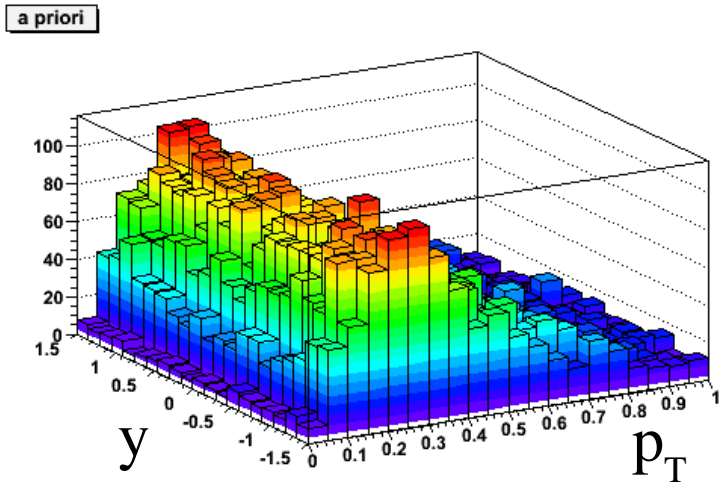


The actual *user input* are the THnSparse → the procedure is AliRoot-independent

Example: (p_T,y) spectra of π

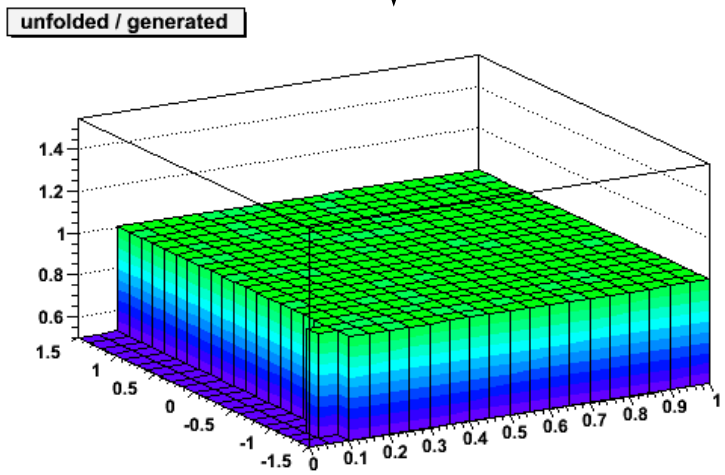
- 1st trivial example : use the *MC* as *a priori* spectrum (*yes, this is cheating...*)





AliCFUnfolding

ratio unfolded/generated :



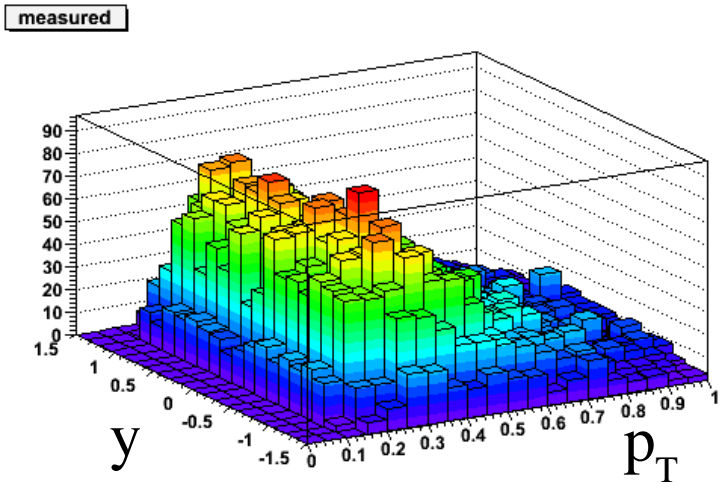
Ends at 1st iteration

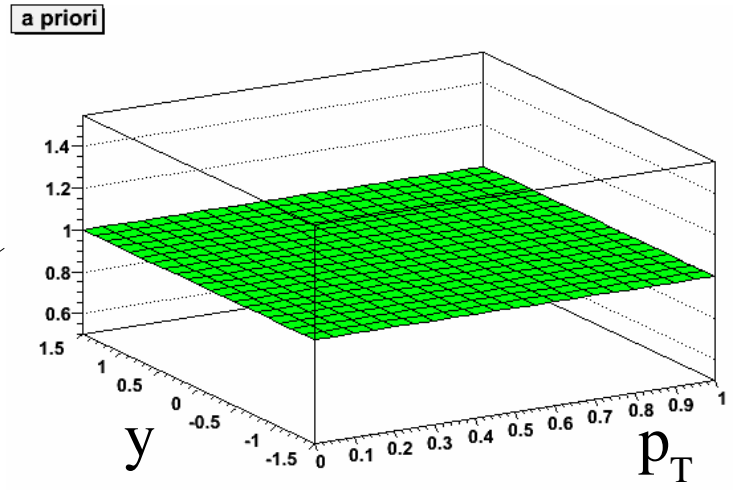
$\chi^2 \sim 10^{-13}$

cpu time ~ 20ms

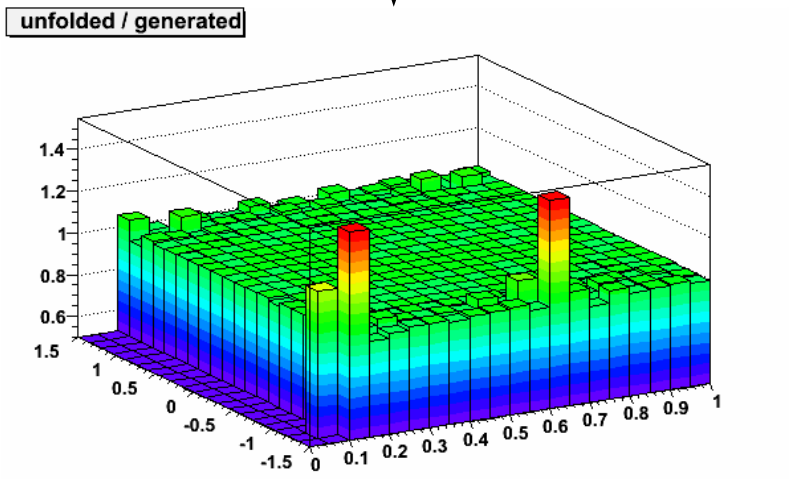
Example: (p_T,y) spectra of π

- 2nd example : use a flat *a priori* spectrum





AliCFUnfolding



ratio unfolded/generated :

Ends at 12th iteration

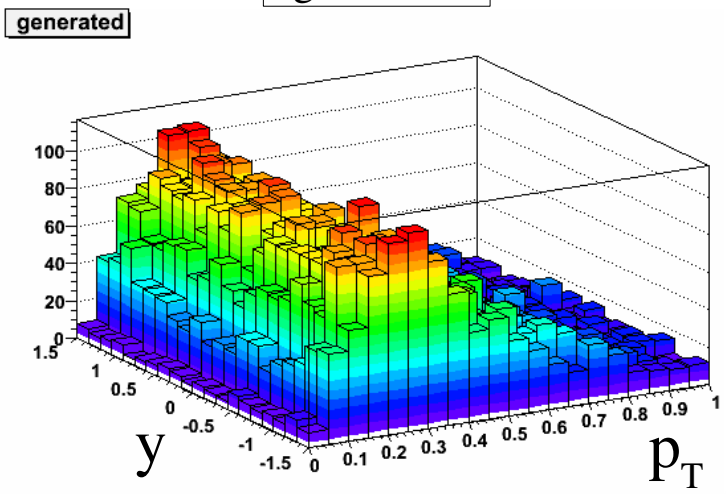
$\chi^2 \sim 10^{-5}$

cpu time ~ 300ms

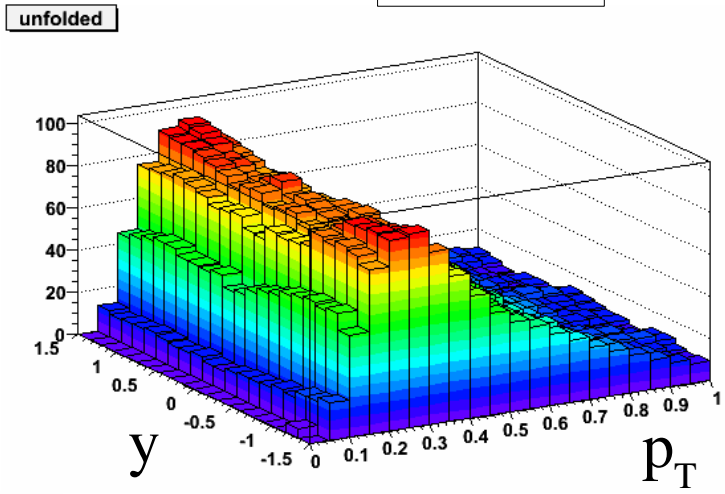
Example: (p_T,y) spectra of π

- 3rd example : use a flat *a priori* spectrum + smoothing with neighbours
  - same as previous slide, but with smoothing on :

generated



unfolded



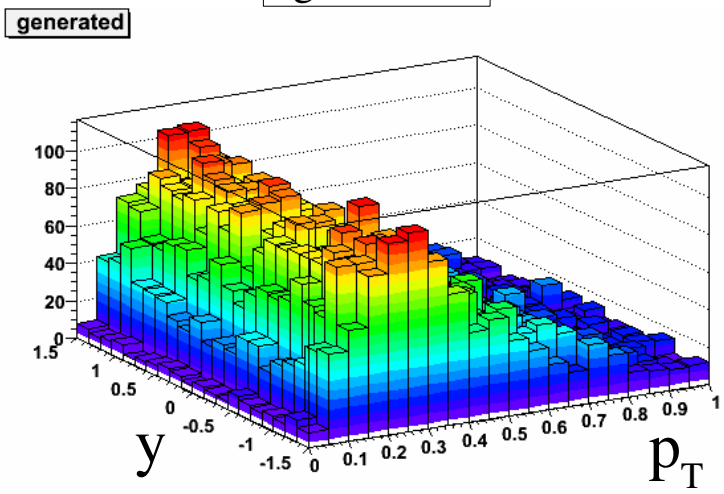
this example doesn't converge , $\chi^2 \sim 1$

- *generated spectrum not smooth here*
- *low pT part is very steep -> smoothing does no good*
- *user has to be careful !*

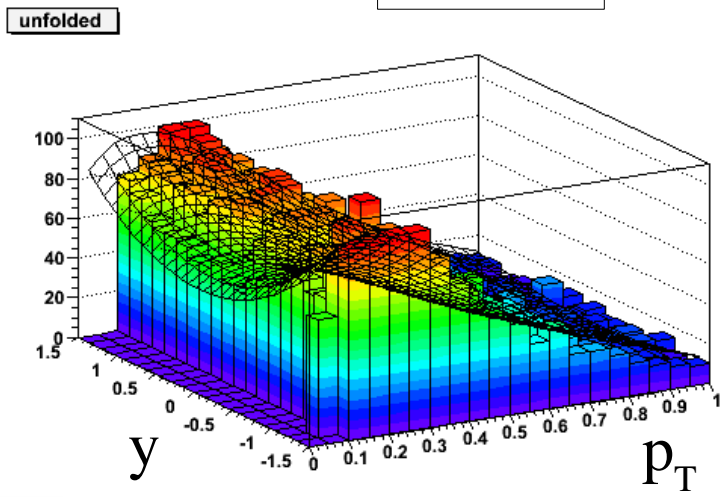
Example: (p_T,y) spectra of π

- 4th example : use a flat *a priori* spectrum + smoothing with TF2
  - same as previous slide, but with smoothing done with a fit function

generated



unfolded



Ends at 3rd iteration

$\chi^2 \sim 10^{-6}$

cpu time ~ 2s

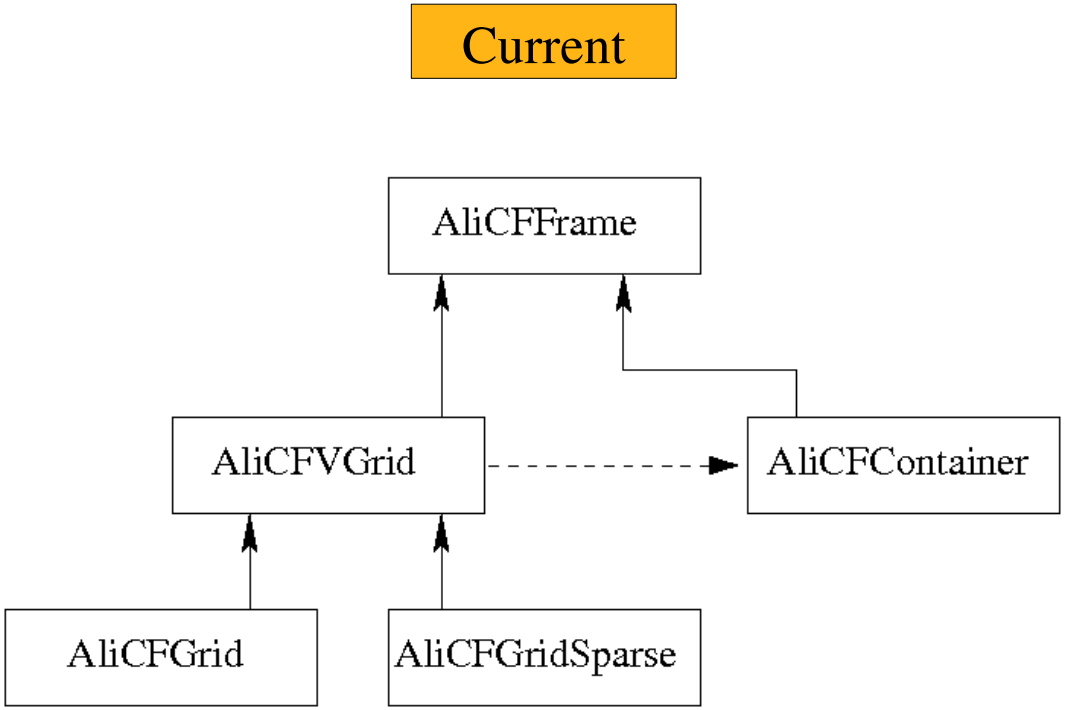
- this is limited to 3-dimension spectra
- you should have an idea of the fit function first, although you don't know the true distribution

New N-dim. unfolding

- Use
  - example task in CORRFW/test/AliCFTaskForUnfolding.C
    - prepares several objects
    - => the container used to calculate the efficiency map
    - => response matrix
  - then run unfolding macro example in CORRFW/test/testUnfolding.C
    - user input:
    - measured spectrum , a priori spectrum, efficiency map, response matrix
    - max number of iterations, max χ^2, smoothing…
- Feedback : none
  - please provide !!

New CORRFW design

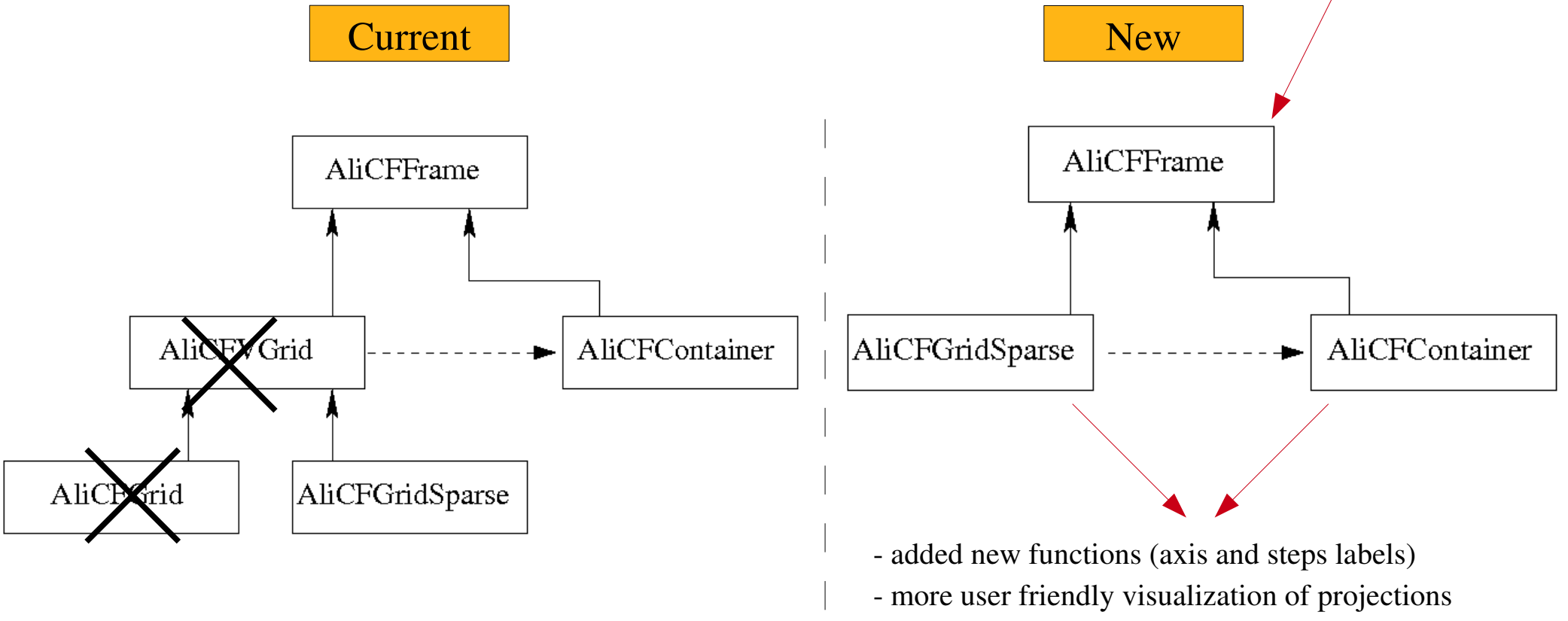
- Still in a validation step
- Idea : make everything simpler



New CORRFW design

- Still under validation
- Idea : make everything simpler

- simplified structure
- removed all data members
- pure virtual functions



- added new functions (axis and steps labels)
- more user friendly visualization of projections
- info on grid contents directly available from THnSparse

will be committed after tests (in a few days)

Summary & to-do list

- new N-dimension unfolding useable and working
  - people should use it to unfold correctly their data !!!
  - please try it out and give feedback
- new CORRFW design is under test and will be committed soon

- Further improve ergonomics
- Error calculation in AliCFUnfolding class
- Update online documentation!
- Give another tutorial (next offline week?)