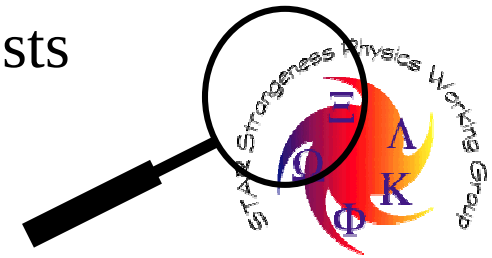
V0/XiFinder progress :

Outline :

- V0/XiFinder, what is it ?
- Writing a new code
- Status
- The V0Finder
- The xiFinder, results of the tests

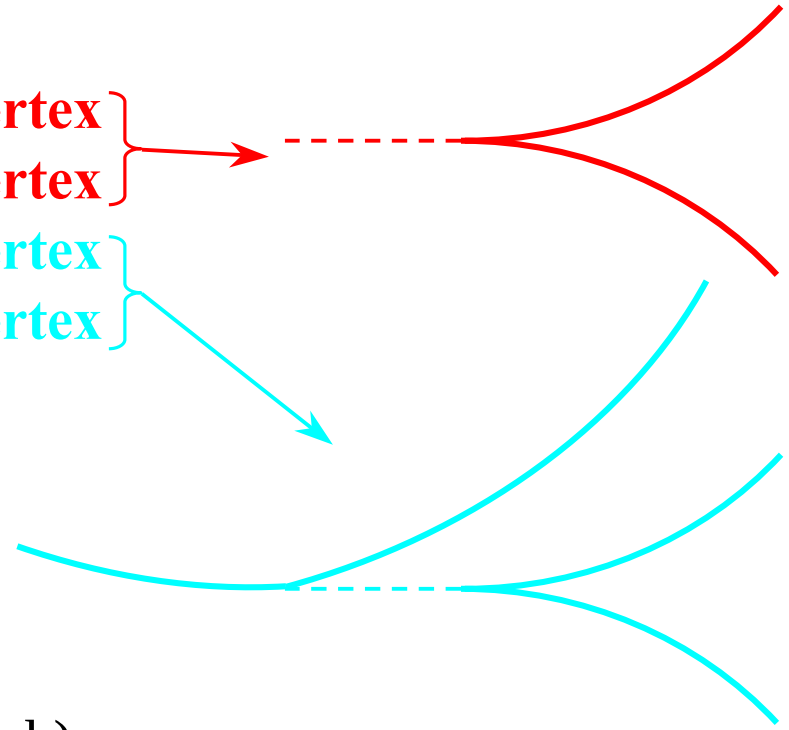
Strangeness group



ireS Institut de Recherches Subatomiques STRASBOURG

V0/XiFinder, what is it ?

- $K^0 \rightarrow \pi + \pi$: 2 tracks **V0 vertex**
  $\Lambda \rightarrow \pi + p$: 2 tracks **V0 vertex**
  $\Xi \rightarrow \Lambda + \pi$: 3 tracks **Xi Vertex**
  $\Omega \rightarrow \Lambda + K$: 3 tracks **Xi Vertex**

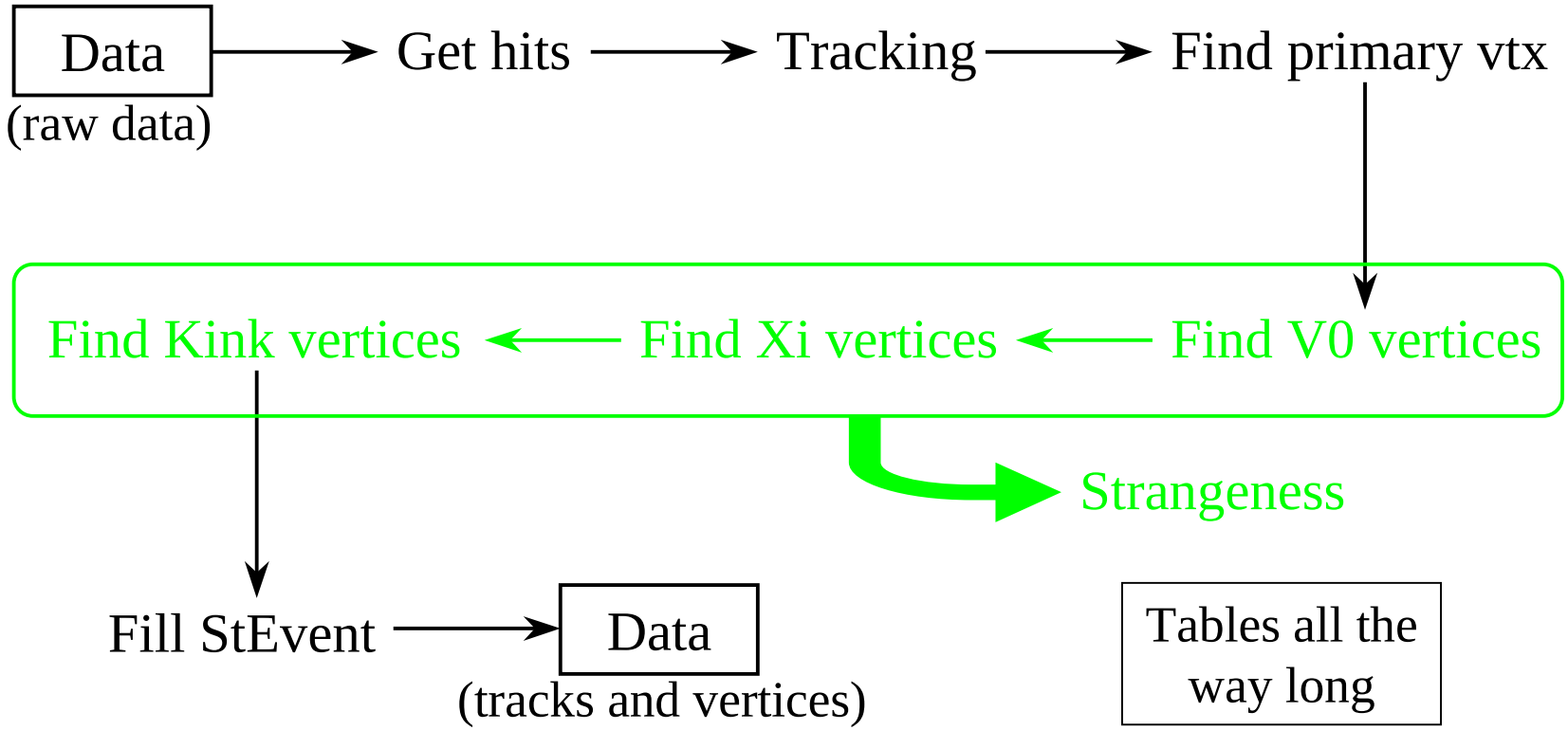


- **V0 vertex :**
  Loop over all pairs of tracks
  If close enough : **found a V0**

- **Xi Vertex :**
  Loop over all pairs of (V0, track)
  If close enough : **found a Xi**(-vertex)

V0/XiFinder, what is it ?

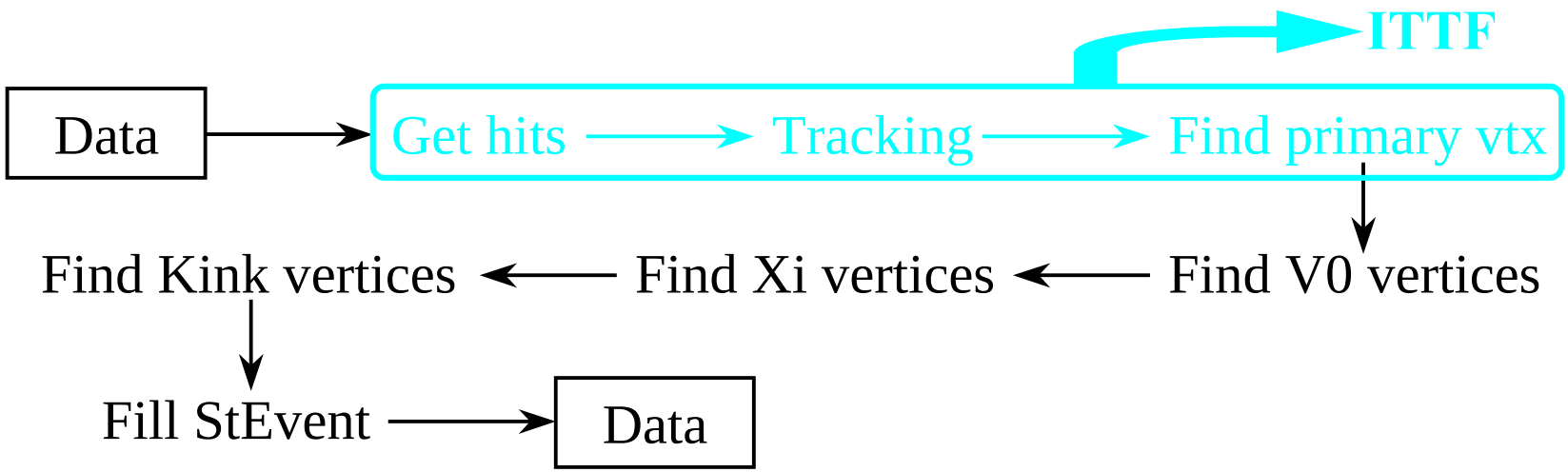
- My naive (basic user's **;-)**) view of the reco chain :



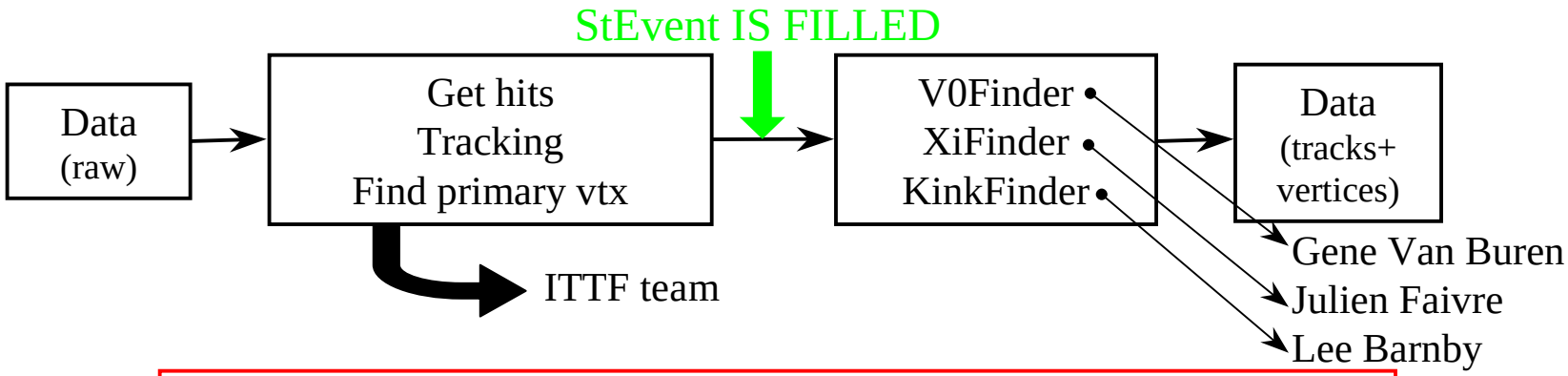
Writing a new code :

- Getting rid of **Fortran** inserted in C++ code
- **ITTF** coming : all **StEvent-based**, ⇒ getting rid of **tables**
- Easier to manage

Tables disappear ! Strangeness PAM don't work anymore !



Writing a new code :



Can read muDsts as well (much faster and more convenient)

Background studies

Status : tests :

- **1-** Current chain + Fortran PAM ⟷ Current chain + V0/XiFinder

  Test of the V0/XiFinder

- **2-** Current chain + V0/XiFinder ⟷ ITTF chain + V0/XiFinder

  Test of ITTF output

- **3-** Any chain + V0/XiFinder ⟷ Any chain + new V0/XiFinder

  Test of new algorithms
  for the V0/XiFinder

Both tests 1 and 2 are on their way

Test 1 : all the variables in the code must be identical.
Test 2 : the final results must be exactly the same.

Status :

- **Tester Fortran vs C++ V0/XiFinder :** Finished
  Tested (artificial errors introduced)
- **Tester current chain vs ITTF** : Not begun
- **KinkFinder :** To be begun soon
  Already C++, but still reads tables
- **V0Finder :** Finished
  Tests of the global behaviour OK
  Problem with tests value-by-value
  Being tested plugged on ITTF
  Speed : a little bit slowlier
- **XiFinder :** Nearly finished
  Tests of the coded part almost completed
  More or less tested inserted in the V0Finder (see next slide)
  Speed : faster so far

V0Finder :

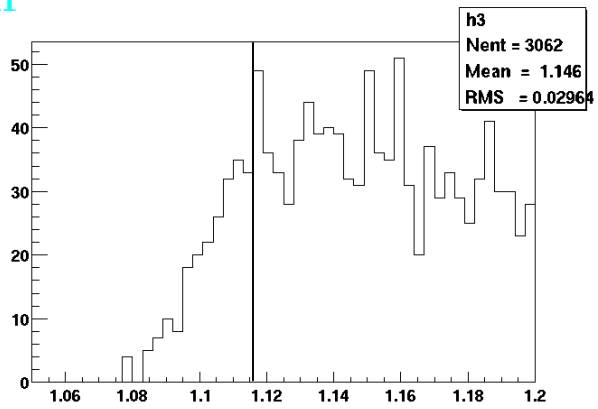
New structure : xiFinder is put ***inside*** the V0Finder

• Current way :

Loop over (track1,track2)
  if (cuts) keep V0
  Store V0
Loop over (kept_V0,track)
  if (cuts) keep Xi
  Store Xi
Loop over kept V0's
  if (not used in a Xi nor points to Pvx)
    remove it

• New way :

Loop over (track1,track2)
  if (cuts) keep V0
  Loop over tracks
    if (cuts) keep Xi
    Store Xi
  Store V0



XiFinder :

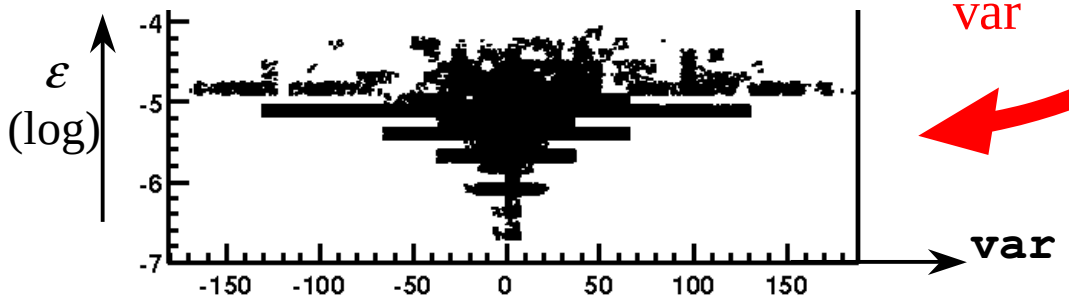
- Not the same parameters in tables and in StEvent
  ⇒ some algorithms change
  ⇒ some subroutines become unnecessary
  Tested… **OK !**

- Cheking the variables : e.g. **`var`** : comparing value $\mathbf{var}_{\mathbf{Fortran}}$ with $\mathbf{var}_{\mathbf{C++}}$. Distributions checked :

$$\varepsilon = \frac{\mathrm{var}_{C++} - \mathrm{var}_{Fortran}}{\mu}$$

μ = mean of the distribution of $\mathbf{var}_{\mathbf{Fortran}}$

- Expected max. error due to quantification : $\frac{\Delta \mathrm{var}}{\mathrm{var}} \leq 1.2 \cdot 10^{-7}$



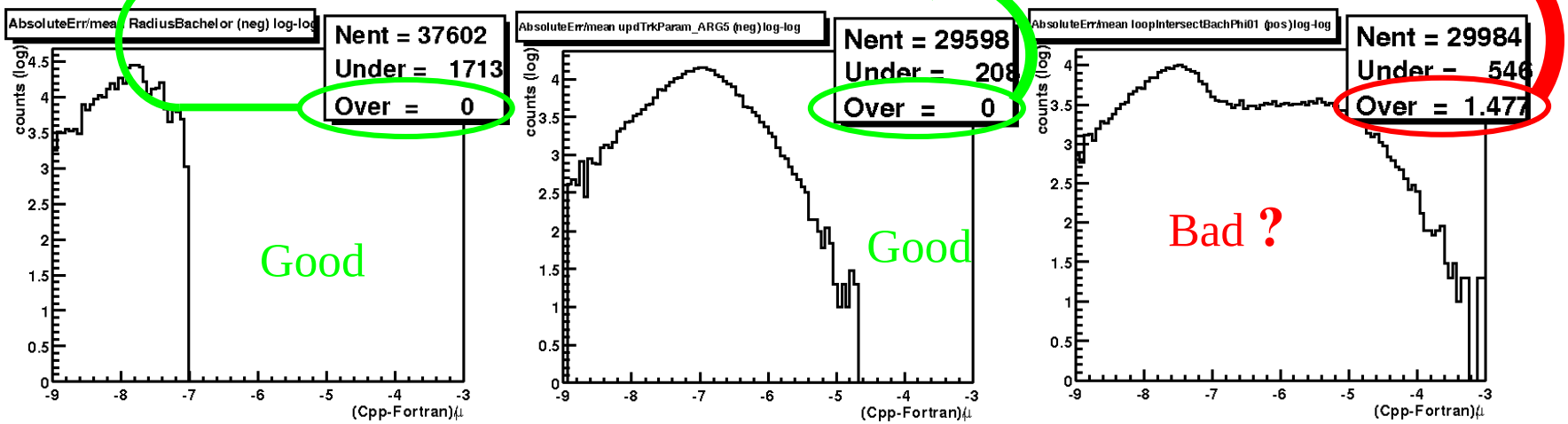
XiFinder :

- Check distributions of absoluteError/mean (log-log)
- Check that no overflows
- Check distributions of absolute error
- If problem : check if variables using the « bad » one are bad too
  check if next iterations are worse

Possibly ≈ 500 variables !

Always check, even if distrib. looks nice

3 candidates over



The future :

Doing like… Brazil !!!

- Finish the xiFinder (2 days)
- Test the xiFinder (> 1 week ?)
- Test the V0Finder (> 1 week)
- Insert the xiFinder inside the V0Finder (1 day)
- KinkFinder (> 1 week)
- Plug the code on ITTF
- Clean the code and improve it

That was in ‘98, not 2002… but you had probably noticed **;-)**