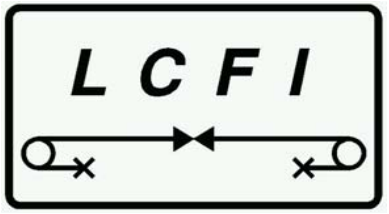
ILC workshop, Valencia, 6 – 10 November 2006

Overview of the LCFI Vertex Package

- Scope of the LCFI Vertex Package
- Overview of the current status
- Summary and Outlook

Sonja Hillert (Oxford)

on behalf of the LCFI collaboration



Introduction

- **The LCFI Vertex Package will provide:**
  - **vertex finder ZVTOP with branches ZVRES and ZVKIN (new in ILC environment)**
  - **flavour tagging based on neural net approach**
    - **includes full neural net package**
    - **default: Richard Hawkings' algorithm, cf. LC-PHSM-2000-021 ,**
      but flexible to allow change of inputs, network architecture etc
  - **quark charge determination, initally limited to jets containing a charged 'heavy flavour hadron'**

- **software will use LCIO for input and output and be interfaced to MarlinReco;**
  tests for running the code in the JAS environment planned in the US (Norman Graf)

- **moving forward at high speed – to be released a few weeks from now**

- **release will be followed by work on upgrades**
  for example improvements of flavour tagging / quark sign selection from using ZVKIN

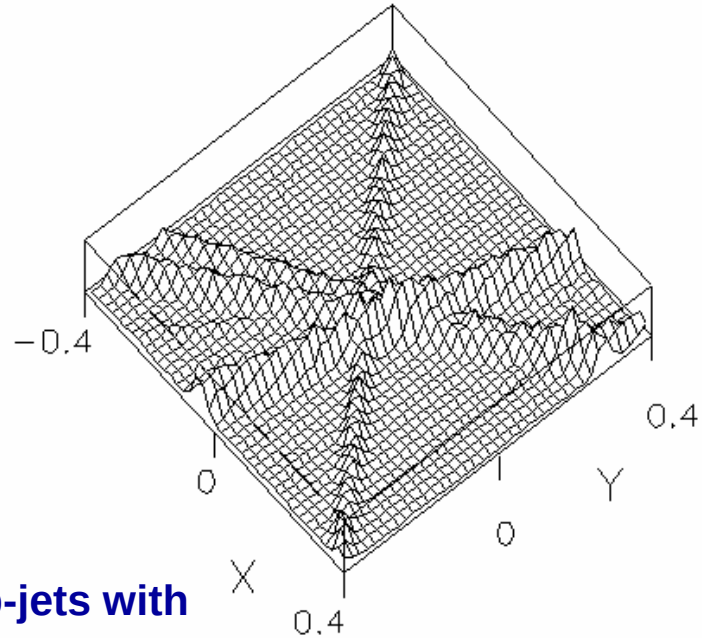
The ZVTOP vertex finder

D. Jackson,

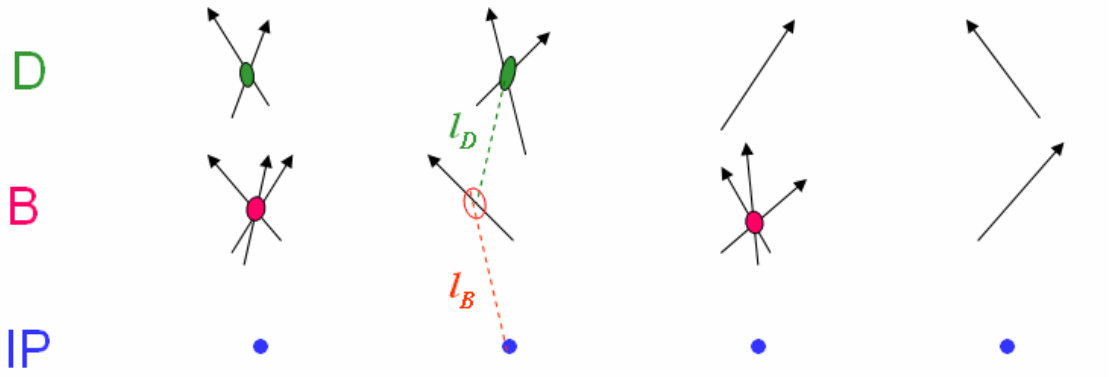
NIM A 388 (1997) 247

- **two branches: ZVRES and ZVKIN (also known as ghost track algorithm)**

- **The ZVRES algorithm: very general algorithm that can cope with arbitrary multi-prong decay topologies**
  - **'vertex function' calculated from Gaussian ´probability tubes´ representing tracks**
  - **iteratively search 3D-space for maxima of this function and minimise χ^2 of vertex fit**



- **ZVKIN: more specialised algorithm to extend coverage to b-jets with 1-pronged vertices and / or a short-lived B-hadron not resolved from the IP**

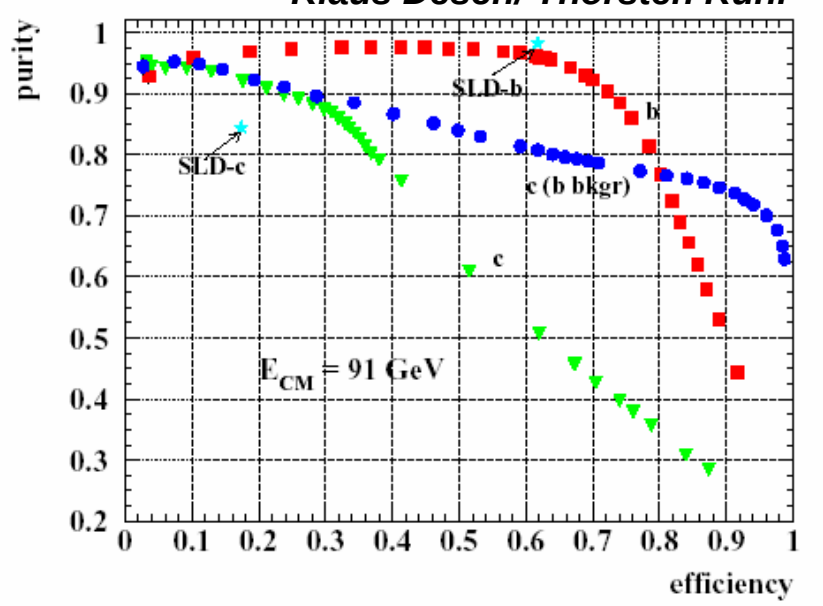


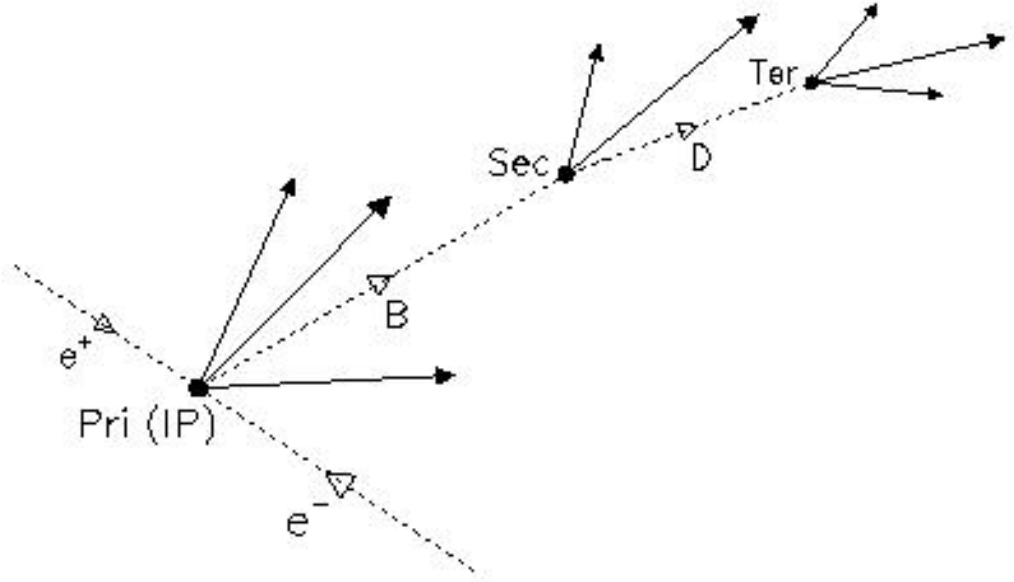
- **additional kinematic information (IP-, B-, D-decay vertex approximately lie on a straight line) used to find vertices**
- **should improve flavour tag efficiency and determination of vertex charge**

Flavour tag and quark charge sign selection

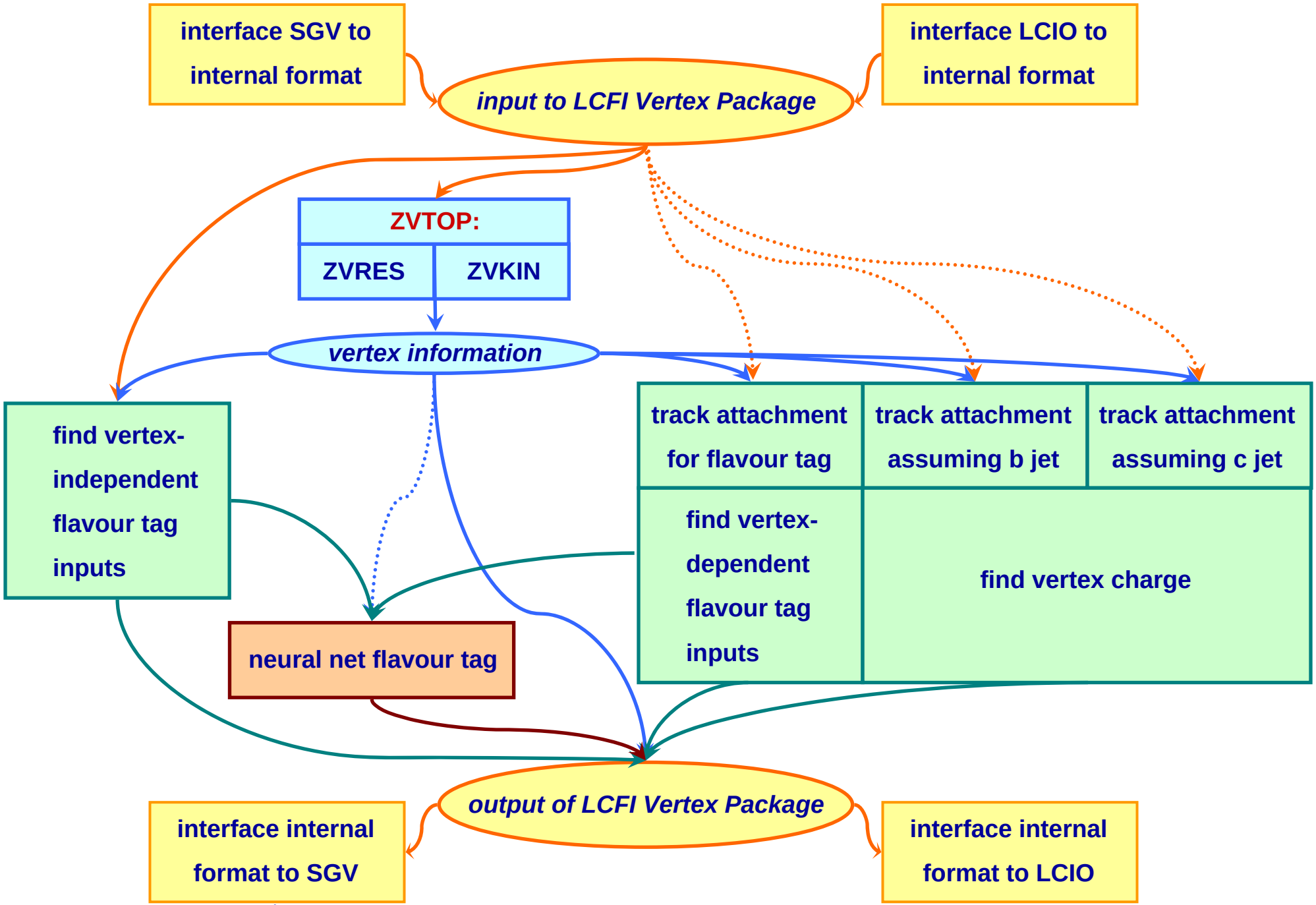
- aim of flavour tag: distinguish between b-jets, c- jets and light-quark / gluon jets
- heavy flavour jets contain secondary decays, generally observed as secondary vertices
- NN-approach to combine inputs; most sensitive: secondary vtx Pt-corrected mass & momentum

Klaus Desch/ Thorsten Kuhl



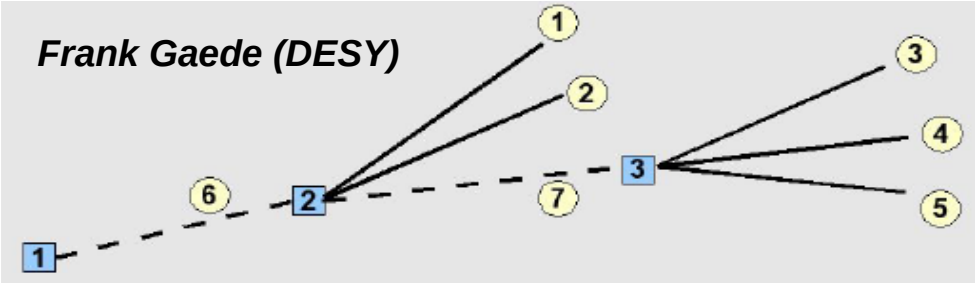


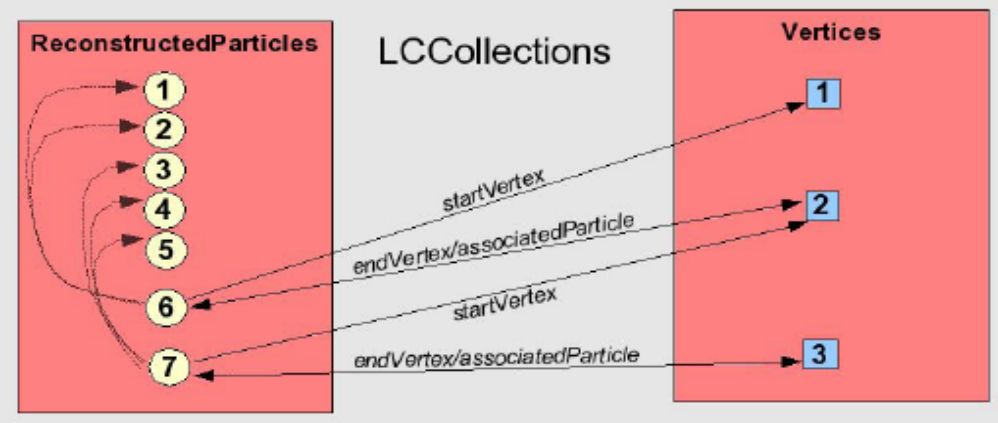
- for charged B-hadrons (40% of b-jets): quark sign can be determined from vertex charge: need to find all stable tracks from B-decay chain
- probability of mis-reconstructing vertex charge small for both charged and neutral cases
- neutral B-hadrons require 'charge dipole' procedure from SLD still to be developed for ILC


interface SGV to internal format
interface LCIO to internal format
input to LCFI Vertex Package
ZVTOP:
ZVRES
ZVKIN
vertex information
find vertex-independent flavour tag inputs
track attachment for flavour tag
track attachment assuming b jet
track attachment assuming c jet
find vertex-dependent flavour tag inputs
find vertex charge
neural net flavour tag
output of LCFI Vertex Package
interface internal format to SGV
interface internal format to LCIO

Interfacing the Vertex Package

- **LCIO persistency framework has been extended by dedicated vertex class to accommodate the output of our software:**





  - each ReconstructedParticle points to one vertex from which it originated & to decay vertex

- **will provide MARLIN processors (modules) giving example code for**
  - running ZVTOP (one processor for each of the two branches ZVRES, ZVKIN)
  - calculating neural net input variables from input to package & ZVTOP output
  - training neural nets for flavour tag, obtaining NN outputs, determine purity vs efficiency
  - vertex charge calculation
  - combined processor: ZVRES + Hawkings flavour tag + vertex charge calculation

Current status

- performance of ZVRES branch has been shown to be at least as good as FORTRAN in detailed tests of increasing complexity (Ben Jeffery, Mark Grimes)
- ZVKIN branch implemented, first tests successful (Ben Jeffery)
- calculation of flavour tag inputs coded (C++) and tested within SGV (Erik Devetak)
- designed & implemented a set of internal 'working classes' linking ZVTOP with the other parts of the package (Ben Jeffery)
- code ported into MARLIN framework;
  MARLIN processors providing examples how to use our code implemented,
  'full chain test' (ZVRES, tag, vertex charge) with SGV event reconstruction beginning,
  initial results promising (BJ, MG, ED, SH)
- work on LCIO interface ongoing;
  storage of output in LCIO implemented using the new Vertex class (Ben Jeffery)

Strategy for validating the code

➢ Tests using SGV event reconstruction

permits direct comparisons with results from FORTRAN version using identical input events

- standalone test of ZVRES, input / output directly from / to SGV common blocks ✓
- separate tests of Marlin processors for ZVRES, ZVKIN, flavour tag input calculation ✓

  FORTRAN-LCIO interface used to write out lcio file from SGV, read in by Marlin processor and used to feed values into internal working classes of our package

  results from those tests: Ben Jeffery's talk in this session
- full-chain test of ZVRES + flavour tag + vertex charge using same setup (✓)

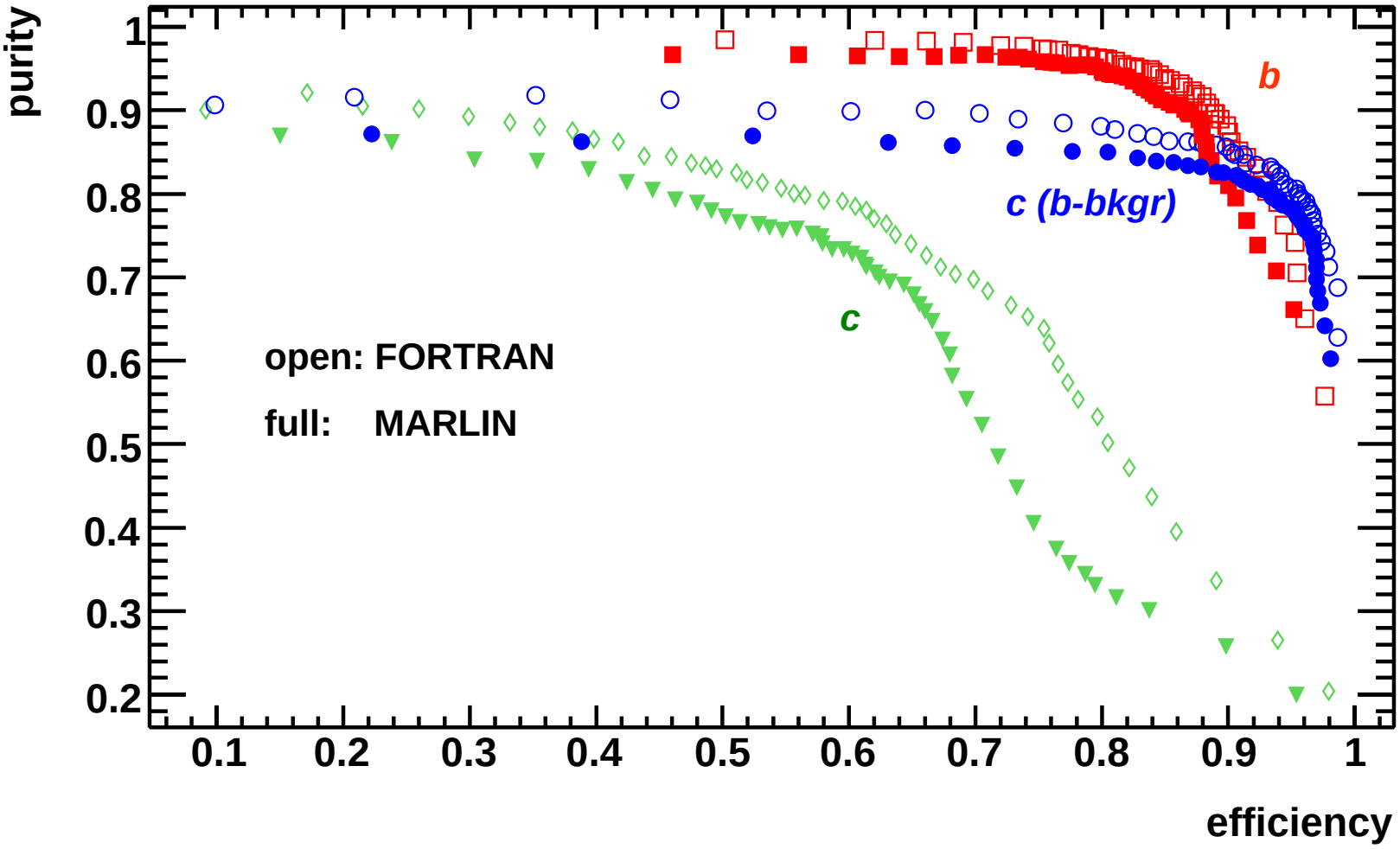
  convert Marlin output to root & use analysis software previously developed for FORTRAN setup

➢ Tests using MarlinReco event reconstruction

- once interface from MarlinReco to our working classes is in place, will repeat full chain test

Test of Marlin-ZVRES + Marlin flavour tag

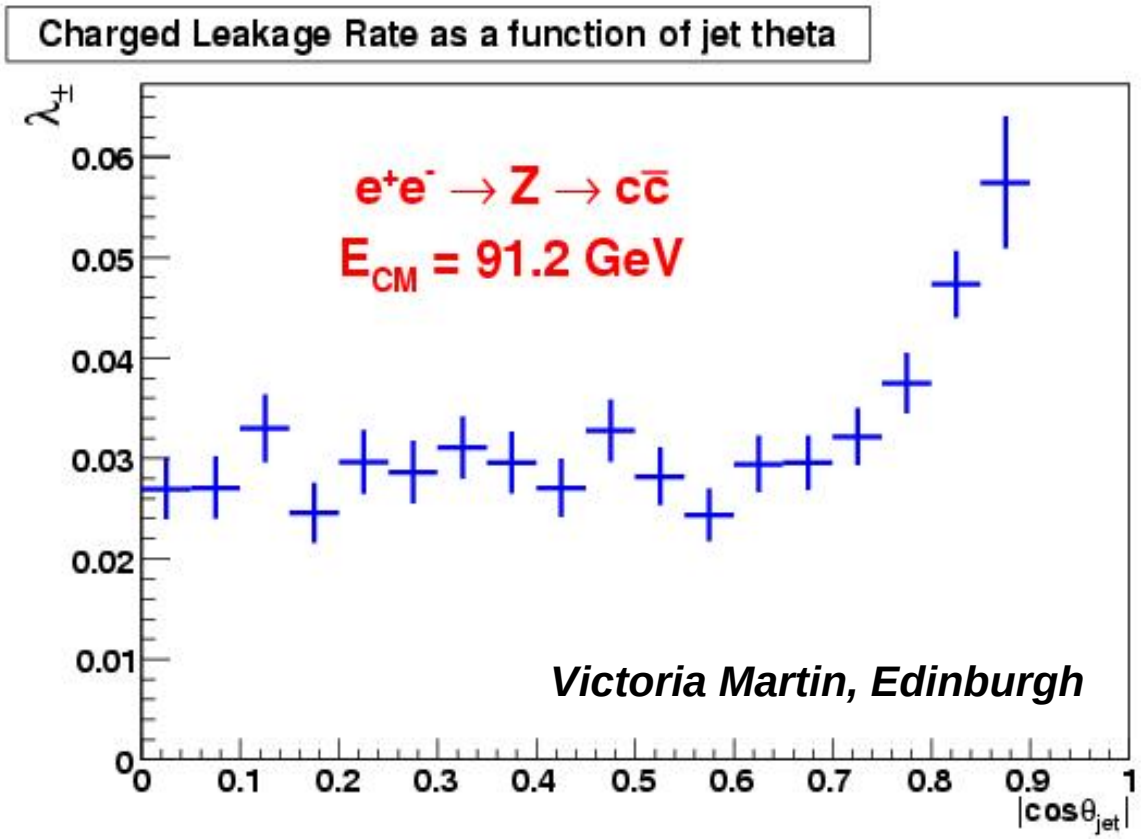
- Comparison of MARLIN and FORTRAN at the Z-peak, identical input events



- good result for a first attempt, differences to be looked into in more detail

Vertex charge reconstruction

- reconstruction method for vertex charge of charged c-jets developed and performance evaluated using FORTRAN setup (to be compared to Marlin result)



- look at leakage rate (probability of reconstructing neutral hadron as charged) as function of polar angle

- for pure c-jet sample find excellent performance –
  easier compared to b-jets, due to complexity of B-hadron decay chain

Areas needing further work

- **interfacing to event-input from MarlinReco-based event reconstruction**

  (for initial tests will only use track cheaters)

- **make code more robust by including handling of bad user input and other errors**

- **system test of full chain (ZVRES + flavour tag + vertex charge)**
  - **run using SGV input needs to be understood**
  - **repeat tests using input from MarlinReco-based event reconstruction**

- **general usage documentation (independent class documentation mainly complete)**

Summary and outlook

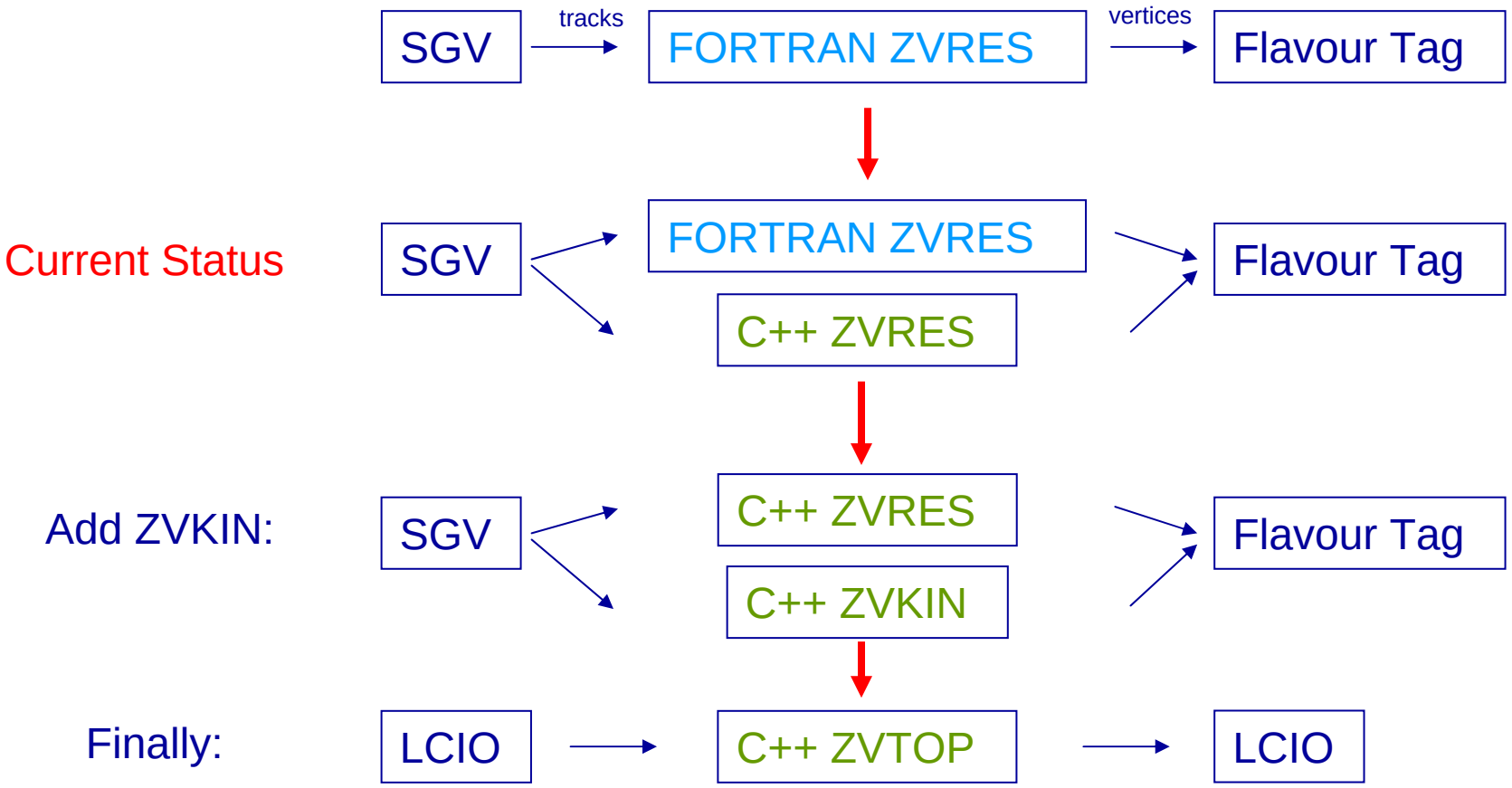
- Development and validation of the LCFI Vertex Package are far advanced.
- A new Vertex class has been introduced into LCIO. Integration of our package into MarlinReco is in progress. Running code from JAS environment to be investigated to ensure interoperability of the reconstruction frameworks in this area (N Graf).
- Interfacing to event-input from MarlinReco event reconstruction needs further work.
- First results from a full-chain run with SGV input are promising, but need to be understood further. A full-chain test with MarlinReco reconstruction will follow.
- The first release of the code is planned in a few weeks.
- Detailed comparisons with MarlinReco input and quantitative exploration of improvements from the ghost track algorithm will be the next steps after the release.

Additional Material

ZVTOP - Progress

Initial aim: replace FORTRAN ZVRES in SGV for testing

- allows comparison of intermediate algorithm states when working on identical tracks
- new version can be verified to be at least as good as FORTRAN



The ZVTOP vertex finder

D. Jackson,

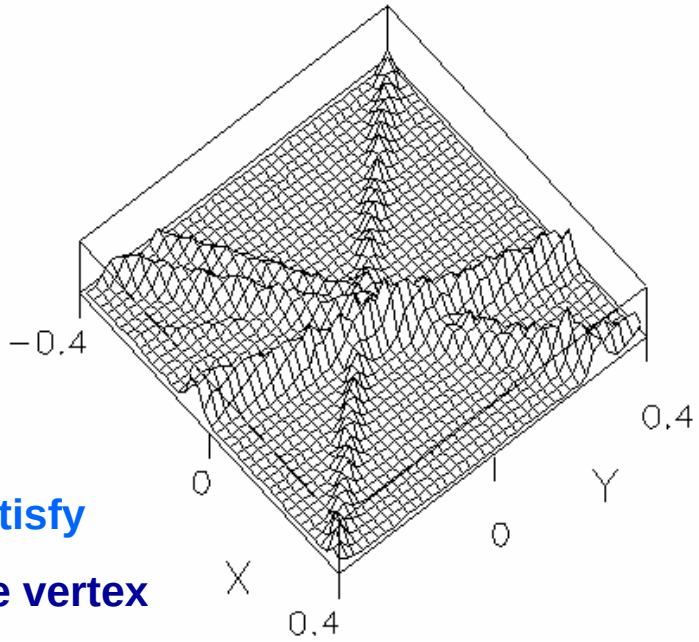
NIM A 388 (1997) 247

two branches: ZVRES and ZVKIN (also known as ghost track algorithm)

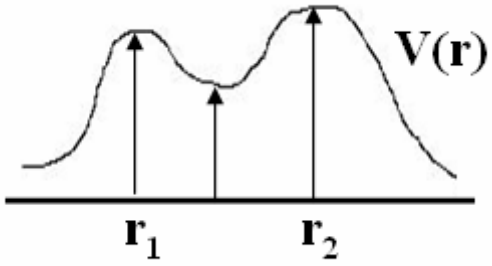
The ZVRES algorithm:

- **tracks approximated as Gaussian ´probability tubes´**
- **from these, a ´vertex function´ is obtained:**

$$V(\mathbf{r}) = \sum_{i=0}^{N} f_i(\mathbf{r}) - \frac{\sum_{i=0}^{N} f_i^2(\mathbf{r})}{\sum_{i=0}^{N} f_i(\mathbf{r})}$$



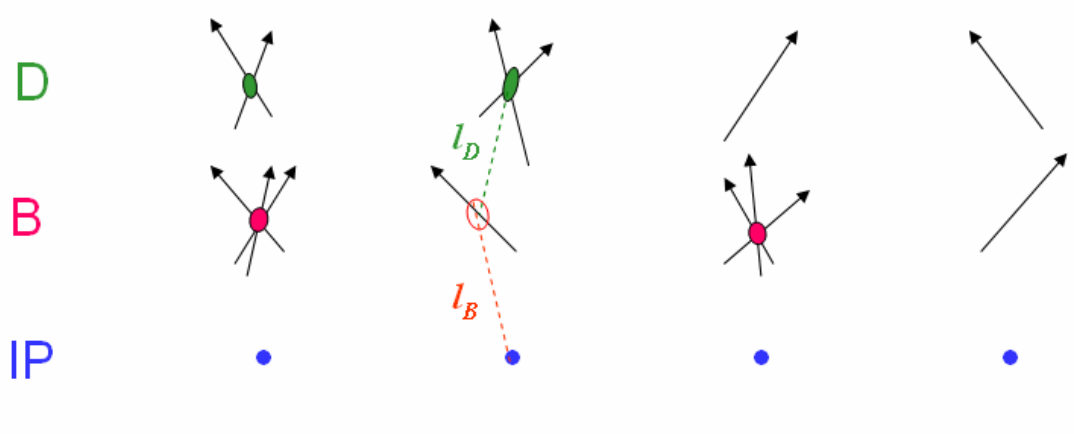
- **3D-space searched for maxima in the vertex function that satisfy resolubility criterion; track can be contained in > 1 candidate vertex**
- **iterative cuts on χ^2 of vertex fit and maximisation of vertex function results in unambiguous assignment of tracks to vertices**

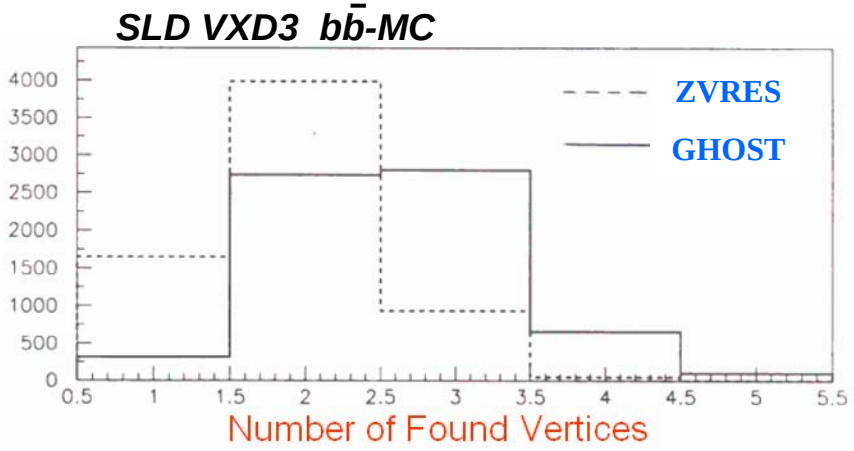


- **has been shown to work in various environments differing in energy range, detectors used and physics extracted**
- **very general algorithm that can cope with arbitrary multi-prong decay topologies**

The ZVKIN (ghost track) algorithm

- more specialised algorithm to extend coverage to b-jets in which one or both secondary and tertiary vertex are 1-pronged and / or in which the B is very short-lived;
- algorithm relies on the fact that IP, B- and D-decay vertex lie on an approximately straight line due to the boost of the B hadron





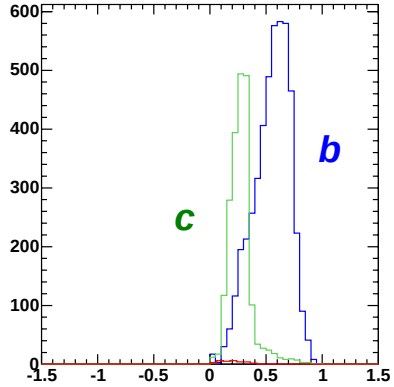
- should improve flavour tagging capabilities

Flavour tag

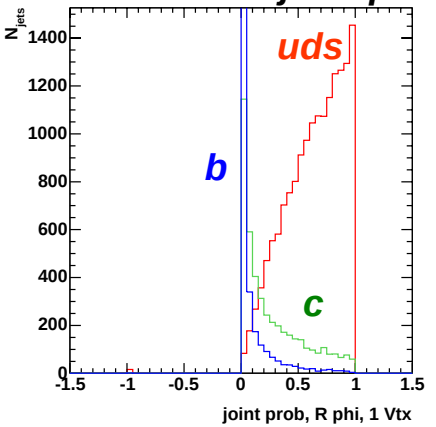
- **Vertex package will provide flavour tag procedure developed by R. Hawkings et al (LC-PHSM-2000-021) and recently used by K. Desch / Th. Kuhl as default**

- **NN-input variables used:**
  - **if secondary vertex found: M_{Pt} , momentum of secondary vertex, and its decay length and decay length significance**
  - **if only primary vertex found: momentum and impact parameter significance in R-ϕ and z for the two most-significant tracks in the jet**
  - **in both cases: joint probability in R-ϕ and z (estimator of probability for all tracks to originate from primary vertex)**

tanh (M_{Pt} / 5 GeV)

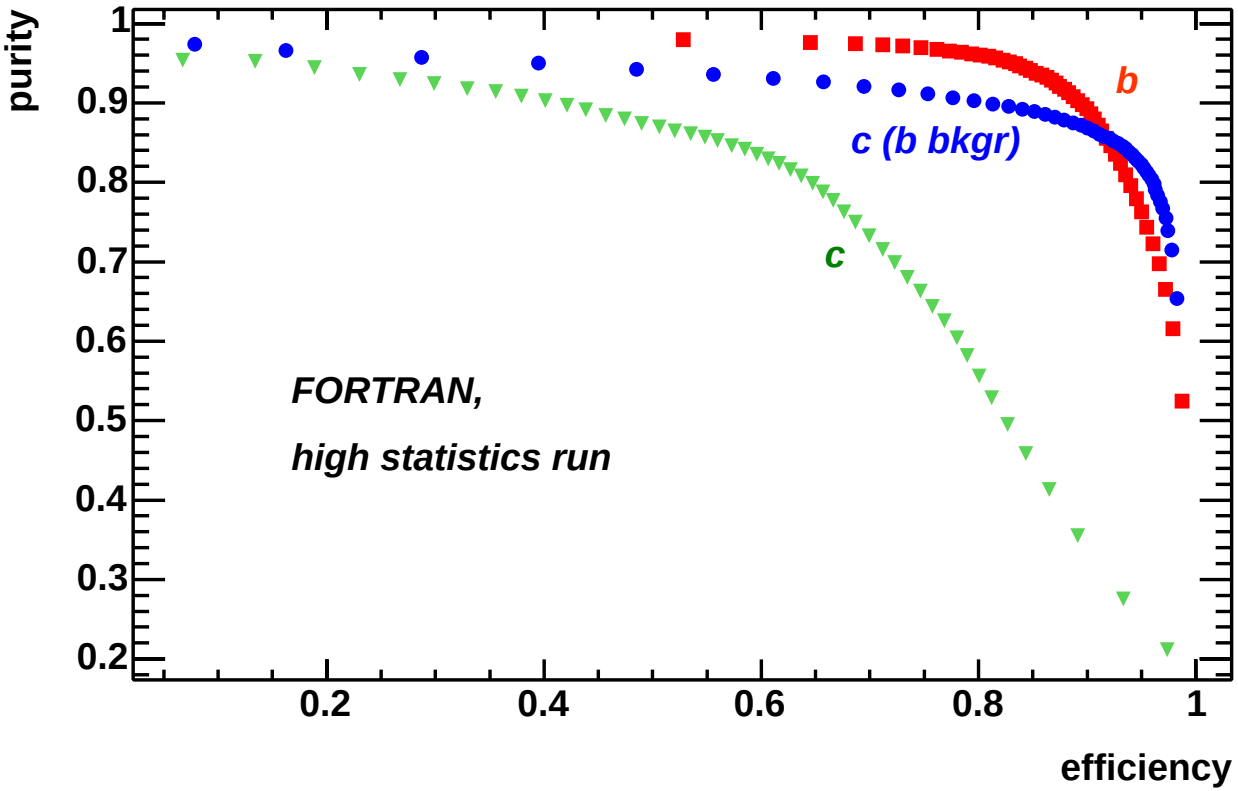


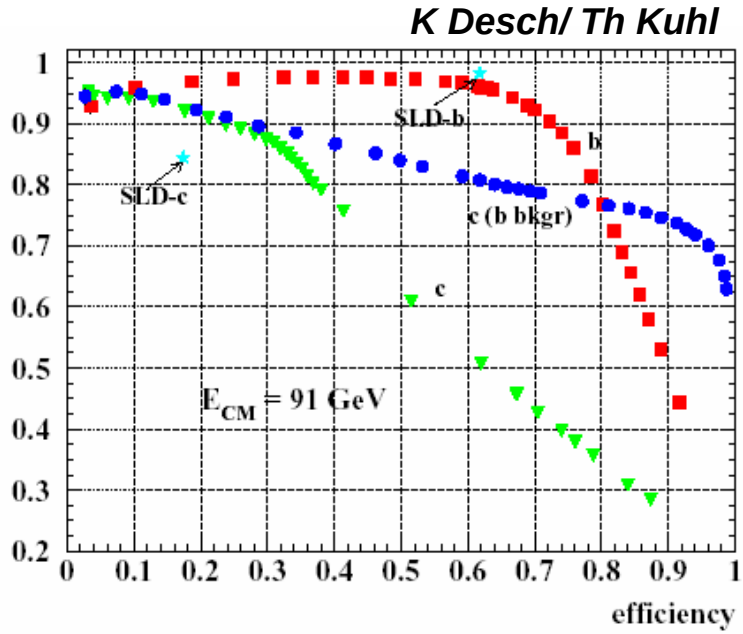
joint probability

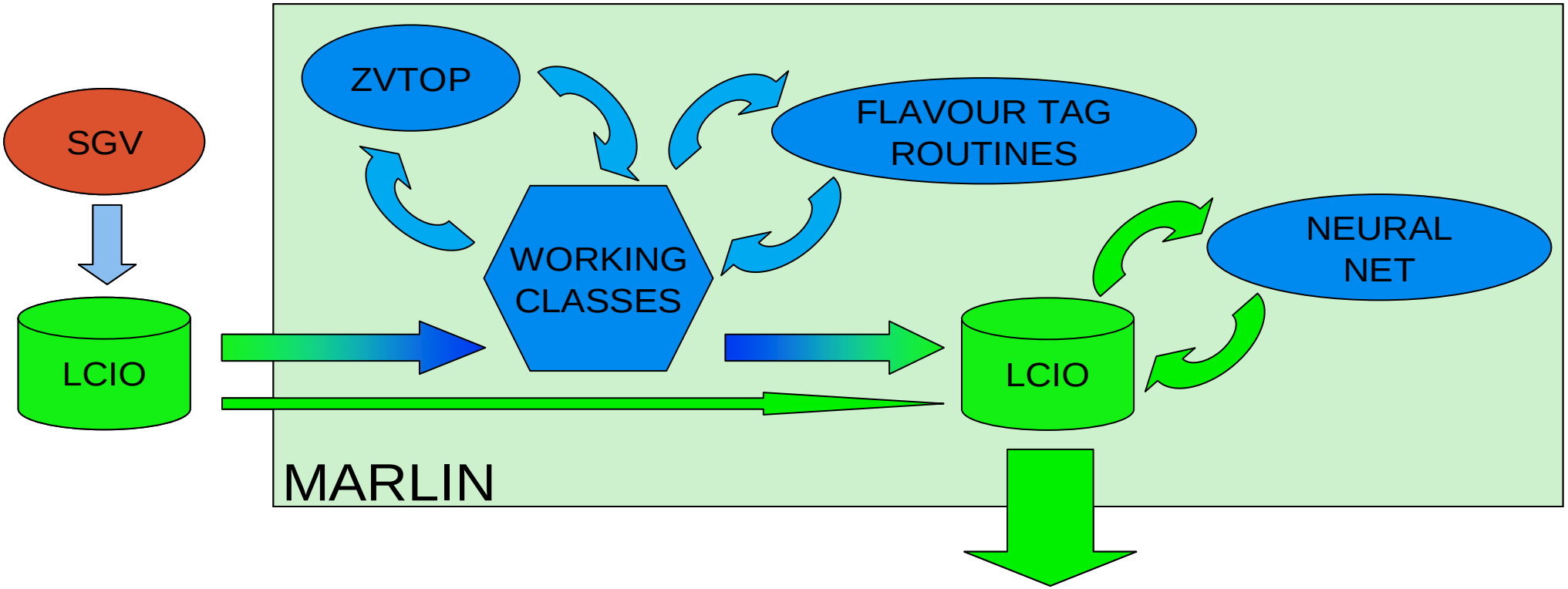


- **will be flexible enough to permit user further tuning of the input variables for the neural net, and of the NN-architecture (number and type of nodes) and training algorithm**

Flavour tag purity vs efficiency at the Z-peak






SGV
LCIO
ZVTOP
WORKING CLASSES
FLAVOUR TAG ROUTINES
NEURAL NET
LCIO
MARLIN