

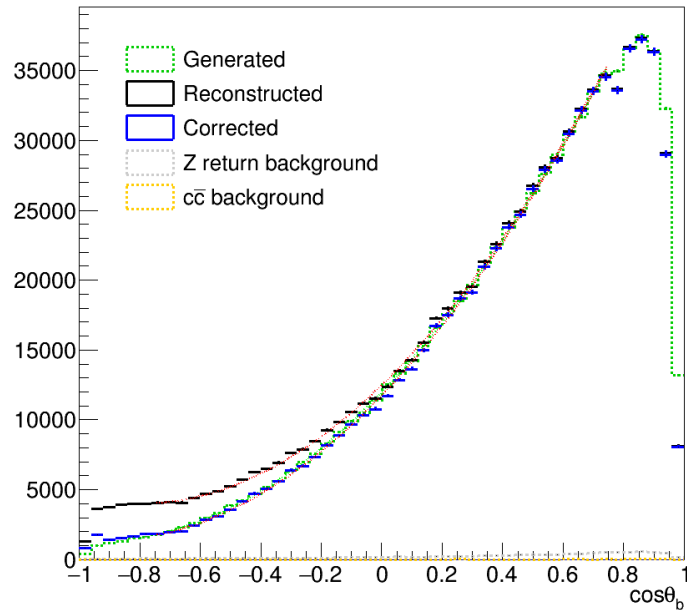


$e^+e^- \rightarrow q\bar{q}$  at the ILC



Simulation of a bb event at ILD (Bilokin)

A. Irlès, 1<sup>st</sup> June 2018



Nice agreement with what S. Bilokin has in the thesis

- I repeated S. Bilokin analysis of the  $bb$  asymmetry measurement ( $\cos\theta$ ) using  $e+e- \rightarrow b\bar{b}$ , 250 GeV,  $\sim 266 \text{ fb}^{-1}$  with left (and right) polarization simulated samples
  - `/ilc/prod/ilc/mc-dbd/ild/dst-merged/250-TDR_ws/2f-highM_Z_hadronic`
- **Analysis chain (processor chain):**
  - SatoruJetFinder (2 inclusive jets using Durham)
  - JetClusteringAndFlavorTag"
  - LcifiplusProcessor (jet clustering, cleaning, vertex refiner, and flavor Tag)
  - VertexRestorer (Bilokin)
  - TTBarProcessor,  $bb$  case (also Bilokin's code)
- **Submitted to DIRAC**
  - Is anyone using this regularly? If yes: I have a couple of questions for you...
  - ILCSoft release version: 011706\_sl6 (which is sl6-v01-17-06)

## VertexRestorer

### ● Version used

- No: ILCSoft/MarlinReco (outdated version with less functions and it does not compile)
- No: Bilokin's version in the github (it also looks a bit outdated)
- Yes: Version compiled by R. Poeschl (/sps/flc/poeschl/yasui-ttbar/yasui )

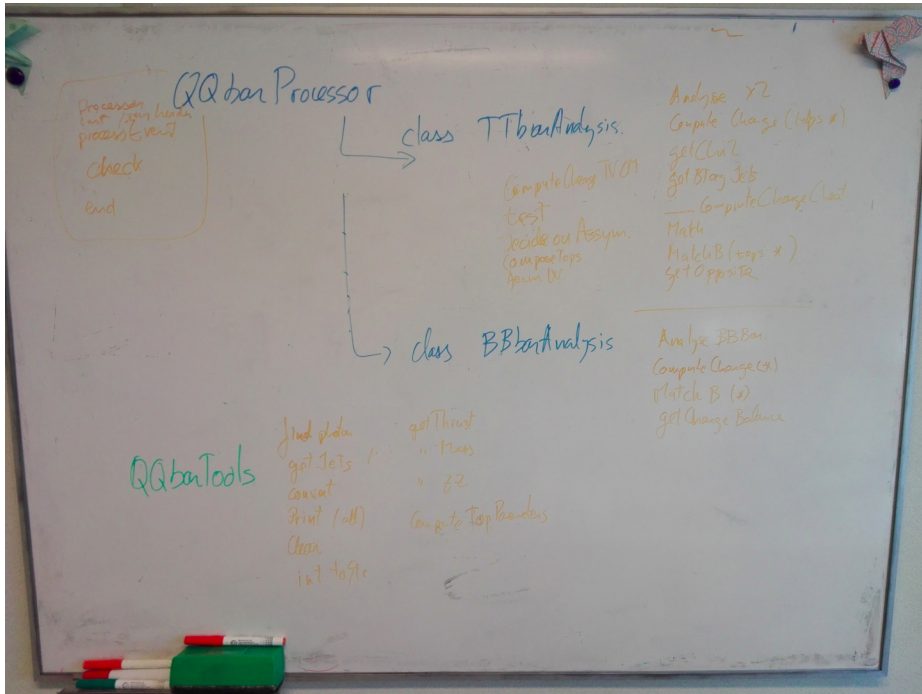
## TTBarAnalysis

### ● Version used:

- Similar situation than above: I use the private version with some minor changes

I focus, for the moment, in the understanding of the TTBarProcessor (next slide)

# TTBarAnalysis → QQbarAnalysis



- The TTBarAnalysis processor condensates a great amount of work from Sviatoslav.
- As usual, it has grown exponentially in time:
- It uses to ~ 35 functions (plus a lot of methods)
  - 5 are the processor base functions
  - ~11 are common for bbar and ttbar analysis
  - 4 are needed (so far) for the bbbar
  - The rest are for the ttbar analysis

● **All in a single file... makes things difficult!**

New structure ongoing:

- QQbarAnalysis is the processor
  - QQbarTools contains the common functions
  - BBbarAnalysis the bbar specific functions
  - TTbarAnalysis the ttbar specific functions.

# TTBarAnalysis → QQbarAnalysis

- All is in a recently created github community :  
<https://github.com/qqbaranalysis>
- VertexRestorer → updated
- QQbarAnalysis (bbbar is done, ttbar in progress)
- MarlinReco → forked only as reference (the VertexRestorer in it is outdated)
- TTBarAnalysis → being exported to the QQbarAnalysis
- ParticleTagger (non tested)
- Will be available soon...

The screenshot shows the GitHub interface for the 'qqbaranalysis' organization. At the top, there are navigation tabs for 'Repositories' (5), 'People' (1), 'Teams' (0), 'Projects' (0), and 'Settings'. Below this is a search bar and filters for 'Type: All' and 'Language: All'. The main content area lists several repositories:

- VertexRestorer**: Forked from Bilokin/VertexRestorer. Description: Vertex restorer. Language: C++. Updated 7 minutes ago.
- QQbarAnalysis**: Language: C++. Updated 13 minutes ago.
- MarlinReco**: Forked from ILCSOft/MarlinReco. Language: C++. License: GPL-3.0. Updated on 18 Apr.
- TTBarAnalysis**: Forked from Bilokin/TTBarAnalysis. Description: TTBarAnalysis processor. Language: C++. Updated on 24 Mar 2017.
- ParticleTagger**: Forked from Bilokin/ParticleTagger. Description: Creates PID on a basis of dE/dx separation. Language: C++. Updated on 11 Jul 2016.

On the right side, there are two sidebars: 'Top languages' showing C++ and 'People' showing a profile for 'airqui' with an 'Invite someone' button.

- At present day I am able to rerun the analysis with:
  - New samples (new models, different energies), different releases (maybe use the new conformal tracking algorithm when released)
- In the short term, in parallel to the previous bullet point:
  - Continue the cleaning, with care, of the software (lot of code that is commented out or not used)
  - Expand the truth level study of the bbar analysis which currently at born level, i.e. [QCD@LO](#)
- Medium term... get deeper in the VertexRestorer code from S. Bilokin.