CDF/DOC/TOP/CDFR/7034
September 30, 2004– Version 1.0

Dilepton Analysis Tools in 5.3.3 TopFind

D. Glenzinski, P. Merkel, J. Thom, A. Yagil
Fermi National Accelerator Laboratory

C. Hill, J. Incandela, C. Mills
University of California Santa Barbara

N. Lockyer, P. Wittich
University of Pennsylvania

P. Savard,
University of Toronto

Abstract

In this document we describe changes made to the Top Group's TopNtuple and Top Event Module code in order to include the basic event selection tools used by the Lepton+Track group and expand the summary information available about leptons. This software is available as of the 5.3.3 version of the CDF offline code.

1 Introduction

In the interest of making the tools and selection used by the lepton+track group public, and updating the general dilepton summary information available, we have added a number of features to the Top group's analysis and ntuplizing code. The following are implemented in TopEventModule and are available in the TopNtuple:

- Phoenix electrons may be added to the tight lepton collection
- A collection of "track leptons", or "tl"s [1] as a set of offline tracks which pass certain cuts
- The "DJET" jet collection which is the basis of the tl fake rate calculation [2]
- Two corrections (one tl-based, one jet-based) to the $\not\!\!E_T$
- Information about the number of silicon layers through which an offline track is expected to pass ("SiExpected")
- Summary information detailing the classification of leptons in the event, and whether the event is a dilepton event

Some of the changes – the DJET collection, the SiExpected information, and some of the new information for the "offltrack" object – were implemented in a private version of the top code and were used in the analysis that was blessed in January 2004. This result is described in [3] and is currently part of a PRL which has been accepted for publication in October 2004. The rest of the changes are either completely new or were previously done at the analysis code level before and have been transferred to the TopFind/TopNtuple level in order to make the basic tools and selection available to the rest of CDF, as well as to make life easier for those of us who will be updating the measurement. In the following, we describe each of those features in turn, as well as how to access them in the 5.3.X version of TopFind/TopNtuple which will be used for the Summer 2004 analyses. After that, we briefly discuss the validation of the changes made to the top code which were made *after* the result was blessed. In an appendix, for reference, we explicitly list all of the tcl switches that must be set to use these features.

2 Tight Leptons

Phoenix electrons have been added to the official tight lepton list. The Phoenix electron selection itself is as implemented in the StdPEM class. All of the cuts are settable at the tcl file level if desired, including the number of silicon hits required on the track (`elePlugPHXMinSiHits`, which can't be less than 3 and defaults to 3) and the maximum Phoenix track $|z_0|$ allowed (`elePlugPHXZ0Cut`, which defaults to the standard 60 cm). All of the default cuts (those set if `leptonIdCutsParm` == 2), and the settable parameters describing them, are given in Table 1. These parameters are all in the top-level menu in the TopEventModule.

Variable	Parameter	Default cuts
E_T	`elePlugEtCut`	$\geq$ 20GeV
Pes2dEta	`elePlugEtaInnerBound`, `elePlugEtaOuterBound`	$1.2 \leq \|\eta\| \leq 2.0$
Had/Em	`elePlugHadEmConst`	≤ 0.05
Pem3x3Chisq	`elePlugPlugChi2Cut`	≤ 10
Pes5by9U	`elePlugU5by9Cut`	≥ 0.65
Pes5by9V	`elePlugV5by9Cut`	≥ 0.65
Isolation	`elePlugIsolCut`	≤ 0.1
ΔR (PES-PEM)	`elePlugDelRCut`	≤ 3.0 cm
PhxMatch	`elePlugPHXMatch`	`TRUE` – required by StdPEM
N_{hits}^{Si}	`elePlugPHXMinSiHits`	≥ 3
$\|z_0^{PHX}\|$	`elePlugPHXZ0Cut`	≤ 60 cm

Table 1: *PHX selection cuts. The baseline cuts (leftmost) are essentially the baseline ETF tight plug electron cuts with the additional restriction that* $|\eta| < 2.0$.

One necessary change generated by this addition is that the Z vertex finding algorithm TopAlgorithms::calculateBestVertexZ had to be changed to explicitly ignore Phoenix electrons. Currently, it chooses the $z0$ of the highest-p_T tight lepton as the best Z vertex for the event The z_0 from a Phoenix electron is not the best one to use for this purpose, since it is set before the tracking algorithm is run. The Phoenix tracking algorithm uses the best quality Z vertex from the `ZVertex` collection to construct the seed tracks fed into the pattern recognition [4]. The z_0 of the Phoenix electron is set to this and will not move much in the fit of the seed tracks to the silicon hits, and this z0 will sometimes not match the actual best vertex for the event as defined by the TopEventModule. To avoid this confusion, Phoenix electrons are omitted from TopAlgorithms::calculateBestVertexZ. So, that function will either take the z_0 of a second high-p_T lepton in the event, if there is one, or it will fall back on the best quality Z vertex form the `ZVertex` collection, which was used to seed the track for the Phoenix electron and is the answer that would have been given for the Phoenix electron anyways.

3 Track Leptons

Track leptons, which are high-p_T, well-measured, isolated tracks, are implemented as a view of offline tracks. There are also new variables added to the offltrack class in the TopNtuple which allow the user to cut on track isolation and to reproduce the dilepton event selection and/or the E_T/p_T $\not{E}_T$ correction used by the lepton+track group at the ntuple level.

3.1 Additions to TopEventModule

In the offline code, if `selectTl` in the TopEventModule menu is set to *true* (it defaults to *false*) in the tcl file, a new CdfTrackView is created and appended to the Event Record. That track view is called `MyTopTrackLeptonView`, if the `newTopCollectionPrefix` parameter is set to “MyTop”, the default. `MyTopTrackLeptonView` is the subset of the default track collection (defTracks) which pass a set of cuts determined by the user and described in Table 2. These cuts are set to the values given in the table if you set `leptonIdCutsParm` == 2, which corresponds to using the standard Run II cuts. Alternatively, the criteria listed there are all settable by parameters (in the new `tl` menu) in the tcl file. Otherwise they default so that all defTracks will pass the cuts.

Variable	Parameter	Default cuts
p_T	`tlPtCut`	$\geq 10\mathrm{GeV}/c$
Axial COT hits	`tlMinCotAxialHitsCut`	≥ 24
Stereo COT hits	`tlMinCotStereoHitsCut`	≥ 20
corrected d_0	`tlMaxD0Cut`	$\leq 250\mu\mathrm{m}$
$\|\Delta z_0\|$	`tlDeltaZVertexCut`	≤ 5 cm

Table 2: *The cuts defining the track lepton view. The “Default Cuts” given are those used in the lepton + track analysis, which are set if* `leptonIdCutsParm` *== 2 is set in the user tcl. Otherwise they can be set using the parameters listed, or they will default to values such that all defTracks will pass. The corrected d_0 is calculated by TopAlgorithms::d0Corr, and $|\Delta z_0|$ is between the track z_0 and TopAlgorithms::calculateBestVertexZ.*

However the user defines these cuts, they define the list of track lepton candidates which are used to correct the $\not\!\!E_T$. The values given in Table 2 correspond to those defining the set of tracks used to correct the $\not\!\!E_T$ in the lepton + track analysis. Further cuts – a higher p_T cut and a requirement on the track isolation – define the track leptons treated as lepton candidates in the lepton + track analysis.

The z_0 cut is implemented differently than in the analysis code for the lepton + track analysis. In the lepton + track analysis, the Δz_0 is between the lepton track and the tl track. Here it is between the BestVertexZ and the tl track. The motivation for doing this was to keep tl quantities independent of other leptons in the event so that they can be more generally useful. In the dilepton case, the BestVertexZ is the high-p_T lepton z_0, so this does not change the lepton + track acceptance, as one would expect (see Section 8).

3.2 Additions to the TopNtuple

New information has been added to the TopNtuple to facilitate the use of track leptons. The track lepton CdfTrackView will be written into the TopNtuple if `MyTopTrackLeptonView` is added to the `otrkList` in the tcl file. Also, new information has been added to the offltrack class, about tracking, calorimetry, and track isolation.

The number of COT hits in each layer is recorded in an array called NumCTHits. Also, the number of active silicon layers crossed by the track is available as SiExpectedLayers (see section 5.

A set of variables give the amount of energy in the 3x3 block of calorimeter pointed at by the track. CalEmTower and CalHadTower store the amount of energy deposited in the electromagnetic and hadronic calorimeter towers pointed at by the track, respectively. CalEm3x3Iso and CalHad3x3Iso store the amount of energy in the 3×3 block of towers around the tower pointed at by the track, excluding the one in the middle. This information is used in the $\not\!\!E_T$ correction, discussed later, and can also be used to distinguish among "electron-like", "muon-like", and "hadron-like" tracks. It is important to note that the calorimetry information is obtained from a specific PhysicsTowerData collection. Currently TopNtuple looks for the collection saved by the $\not\!\!E_T$ correction, under the name "CorrectedMetPTD_ + `newTopCollectionPrefix`".

Last, the track isolation of the track is calculated. The track isolation is the p_T of the seed track, divided by the total sum p_T of all of the tracks (itself included) in a cone of 0.4 around it. The $\sum p_T$ is calculated using TopAlgorithms::TrackIsolation, but a new flag (`tl_style`, which defaults to *false*) has been added, which puts the following requirements on tracks considered in the cone:

- $p_T > .5$ GeV/c
- $|\Delta z_0| < 5$ cm, where $|\Delta z_0|$ is between this track and the seed track
- ≥ 20 Axial COT hits
- ≥ 16 Axial COT hits

The cut on Δz_0 is implemented slightly differently than in the original analysis code. The original analysis code required that the z_0 of a track in the cone be within 5 cm of the z_0 of the tight lepton track in order for the track's p_T to be included in the track isolation calculation. Here, we calculate the track isolation independent of the existence of a tight lepton, for the tl's track isolation to be more generally useful. The track's z_0 is required to be within 5 cm of the z_0 of the *seed track* instead. In practice this has very little effect on the value of the track isolation, as shown in Figure 1, which shows the absolute difference between the track isolation as calculated according to these two definitions.

4 DJETS for Fake Calculation

The lepton+track analysis uses level 5 corrected JETCLU (cone of 0.4) jets, and calculates the fake rate for a track lepton in order to estimate the background contribution from events with a fake tl. The set of all jets which can fake a tl includes those jets for which the number of tracks has fluctuated down to 1, which are not included in the default jet collections. That complete set of jets forms the denominator of the fake rate calculation (see [2] for details). We construct that complete set using JETCLU jets and offline tracks, and store it as a "DJET" collection.

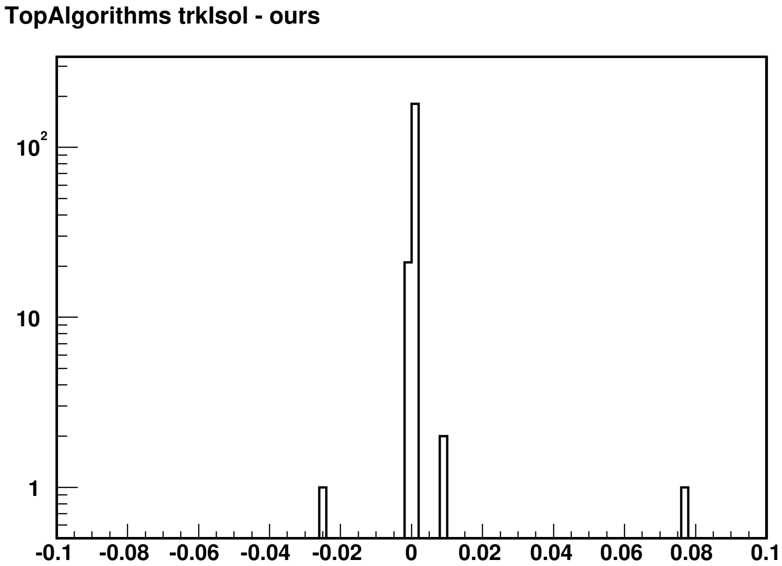

Figure 1: *Difference between the original definition of track isolation as used in the lepton+track analysis and the definition used to fill the track isolation variable in the TopNtuple. The difference between the two definitions is whether the Δz_0 cut is between the tracks in the cone and the tight lepton in the event (original lepton+track analysis) or between the tracks in the cone and the seed track for which the track isolation is being calculated (TopNtuple). This plot was made using 4.11.2 $t\bar{t}$ Monte Carlo.*

We have added the option to create a DJET collection in `TopAlgorithms::jetReclustering`. See the Appendix for details on the necessary tcl switches. The DJET collection is defined as follows. The code first runs the standard reclustering on JETCLU jets. Then it performs level 5 jet corrections on those jets. Tight electron towers are *not* removed from this jet collection. This is expected to be done at the analysis code level. Then, it loops over the DefTracks track collection. Tracks with the following properties are selected:

- $p_T > 10$ GeV
- ≥ 24 axial COT hits
- ≥ 20 stereo COT hits
- `trk.getHelixFit().getNumDegreesOfFreedom()` > 0
- $|z_0| < 60$ cm

In the data, two additional requirements are imposed:

- ≥ 3 φ-layer silicon hits if ≥ 3 active silicon layers are crossed
- corrected $d_0 \leq 250\mu$m if ≥ 3 active silicon layers are crossed,$d_0 \leq 2500\mu$m otherwise

These are essentially the tl candidate requirements. If the tl-track is found inside of a jet (within $\Delta R = \sqrt{(\Delta\varphi)^2 + (\Delta\eta)^2} < 0.4$), and the p_T of the track is higher than that of the jet, the jet's four-momentum is set to that of the track, and a new centroid for the jet is filled using the η, φ, and p_T of the track. If any tracks with the above properties are not already inside the cone of a jet, a new jet object is created with a four-momentum and centroid from that track. The jet is marked with "user flag" 1. Another way to tell that a jet in the DJET collection was filled from a tl candidate is that it will have an `EmFraction` and `EtaDetector` of exactly zero.

5 SiExpected

The number of active phi-side SVX layers is stored in the offltrack object, in the integer member `SiExpectedLayers`. This is filled in the new method TopNtuple::fillSiExpected, which uses the SiExpected tools defined in the TrackingSI package. "Active" means that the ladder was biased and read out during this run according to the database. This can be used to make the tl d_0 cut depend on the number of active layers of silicon crossed, though it was not used this way in the lepton + track analysis.

6 $\not\!\!E_T$ Correction

The lepton+track group applies two corrections to the $\not\!\!E_T$in addition to the standard correction for the z position of the interaction vertex. The first is a correction for the E_T/p_T of any tracks with $E_T/p_T < 0.7$. The second is for the level 5 E_T correction on jets.

The tl E_T/p_T correction can be switched on and off inside the `met` menu in the tcl file, using the switch `correctMetForTrackLeptons`. This switch defaults to *true* if not explicitly set. Tracks considered for the $\not\!\!E_T$correction satisfy exactly the criteria specified in Section 3.1, and are read in from the tl view stored in the event record. For each of the tracks with $E_T/p_T < 0.7$, the track's E_T is subtracted from the $\not\!\!E_T$and its p_T added back in. The E_T used is the calorimeter energy information from the PhysicsTowerData described earlier. The resulting corrected $\not\!\!E_T$ is then stored in two additional arguments given to `TopAlgorithms::applyCorrectionsToMet`, `tlMetMag`, and `tlMetPhi`. Those are then written to the TopSummary object (available through accessors `tlCorrMissingEt()` and `tlCorrMissingEtPhi()`), and into the TopNtuple as `ftlMet` and `ftlMetPhi`.

In the original analysis code, if the tight lepton is a muon, the muon's p_T is automatically subtracted from the $\not\!\!E_T$, and in the general E_T/p_T correction, the track from the tight lepton (muon or not) is skipped if it is found in the tl list. This has two minor difficulties. One is that the correction for tight muons is for p_T only and ignores any energy in the calorimeter, which is a small effect occurring in all events. The other is that there may be a tight muon in the event which is not chosen as the tight lepton for the tight lepton-tl pair in the event, which can happen if there is more than one tight lepton in the event. If that muon also fails the tl selection because it does not pass the d_0 cut[1], the $\not\!\!E_T$ will not ever be corrected for that tight muon. The latter is a big effect occurring in very few events, but it makes no sense to ignore a tight muon. This second difficulty is also the reason why the correction can't be simplified by just correcting for the whole tl collection and ignoring any tight leptons — some obvious muons are missed. The version of the tl $\not\!\!E_T$ correction now in TopAlgorithms attempts to keep the generality of the original tl collection while avoiding the two potential problems listed above.

The $\not\!\!E_T$correction, as currently implemented, proceeds as follows. First, the $\not\!\!E_T$is corrected for tight CMUP or CMX muons, for both the p_T of the muon and the small amount of energy it deposits in the calorimeter. This stage of the correction is identical to the usual correction done and stored as `fmuoMet` in the TopNtuple summary object, except that it is explicitly checked that the muons are CMUP or CMX only. Then the E_T/p_T correction for the tls is done, and tight CMUP and CMX muons in the tl list, identified by matching η and φ, are skipped. This is a minor improvement over what was done at the analysis code level (see Section 8, Validation), since CMUP and CMX muons are always corrected, regardless of whether there are other tight leptons in the event, or more than one muon. Also, in the analysis code, the muon p_T is just added to the $\not\!\!E_T$, whereas in the TopEventModule implementation the p_T is added in *and* the energy in the calorimeter towers pointed at is subtracted, similar to what is done with the tls. Because of this improvement, there is a difference between the $\not\!\!E_T$corrected by TopMods and that corrected by our analysis code. See the validation section for more details.

The correction for level 5 corrected jet energies can be switched on and off inside the `met`

[1]This is true of ≈1% of tight CMUP or CMX muons in the Monte Carlo. Specifically, in the first 300k of the 5.3.2 Pythia $Z \to \mu\mu$ sample `ztop1i`, for tight CMUP or CMX muons forming one leg of a tight CMUP-CMUP, CMUP-CMX, or CMX-CMX Z candidate as defined in CDF note 6788, 1.1±.1% of CMUP muons and 1.0±.1% of CMX muons failed the d_0 cut.

menu in the tcl file, using the switch `correctMetForCorrJetsTl`. This switch defaults to *false* if not explicitly set. If the user has requested to correct the $\not\!\!E_T$for track leptons, this is used preferentially as the starting point for this correction. Otherwise, if the user has requested to correct the $\not\!\!E_T$for muons, that is used as the starting point. Finally, if this correction is requested but neither of the above two is, the vertex-corrected $\not\!\!E_T$is used as the starting point. Jets considered for the correction come from the same jet collection which was used to generate the tight jets list (at the moment, `JETCLU` with a cone of 0.4). Level 5 corrections are applied to the jets. In order to be used for the correction, jets are required to have a corrected $E_T > 10$ GeV, an `emFraction`< 0.9 (rejects electrons), and have $\eta_{\mathrm{detector}} < 2.0$. Also, jets which match a muon in the tight muon collection in η and φ are skipped. The result is then stored in two additional arguments given to `TopAlgorithms::applyCorrectionsToMet`, `tlJetMetMag`, and `tlJetMetPhi`. Those are then written to the TopSummary object (available through accessors `tlJetCorrMissingEt()` and `tlJetCorrMissingEtPhi()`), and into the TopNtuple as `ftlJetMet` and `ftlJetMetPhi`.

7 Event Information

Some new information about the number and type of leptons in the event has been added to TopSummary.

Several loose lepton counters are available. Those containing the word "Unique" count unique loose leptons, that is, those not duplicated by a tight lepton. Here, duplicated means nearby in $\eta - \varphi$ space: $\Delta R = \sqrt{(\Delta\varphi)^2 + (\Delta\eta)^2} < 0.1$.

nTrackLepton The number of track lepton candidates in the MyTopTrackLeptonView as described above. Available at the TopNtuple level as summary::fnTrackLepton.

nUniqueTrackLepton The number of track leptons not duplicated as a tight lepton. Available at the TopNtuple level as summary::fnUniqueTrackLepton.

nUniqueDilLepton The number of loose e or μ - type leptons not duplicated as a tight lepton. Available at the TopNtuple level as summary::fnUniqueDilLepton

nUniqueLooseLepton The number of loose leptons, any kind (e, μ, or tl), not duplicated as a tight lepton. Available at the TopNtuple level as summary::fnUniqueLooseLepton

The event can be marked "dilepton" in three ways. These dilepton event flags only count leptons and do not look at the jet information or the $\not\!\!E_T$in the event.

isTlDilepton Marked true if there is at least one tight lepton in the event and at least one unique tl candidate which passes the "analysis" p_T and track isolation cuts as defined by the tcl parameters `tlDileptonEventTlPtCut` and `tlDileptonEventTrkIsolCut`, respectively. If not specified by the user these default to the values used in the lepton + track analysis, 20 GeV/c and 0.9.

isDilDilepton Marked true if there is are two or more tight leptons in the event, or if there is at least one tight lepton in the event and at least one unique loose lepton in the event in addition.

isDilepton Marked true if either of the above two is true.

8 Validation

The bulk of the validation of this code was done in 4.11.2, by comparing the output of TopFind and TopNtuple to the output of the original analysis code, before the code was spliced into the 5.3.1 version of the top code. First we checked the acceptance in detail for a large Monte Carlo sample, and then, for quantities (such as track isolation) whose definitions had changed, we plotted the residuals between the old and new definitions.

Some of the machinery (In particular, DJETS, SiExpected, and some of the extra track information) was included already in the version of the offline code that generated the ntuples we used in the lepton + track analysis. We did no additional validation for these, since they were tested extensively as part of the analysis.

8.1 Acceptance Check

We checked the top acceptance in detail with 222,000 events of Pythia $t\bar{t}$ Monte Carlo.[2] The classification of events by tight lepton type and number of jets was unchanged by the code, and the run and event numbers selected as top candidates were unchanged, with one event as an exception.

In that event, the tl failed the track isolation cut the new code but not the original code. The reason is that a track was counted in the cone for the new version, which would have been excluded in the old version since it matched the tight lepton track ID. The check for the match to the tight lepton track ID is removed in the new version since it is intended that tl quantities like the track isolation be independent of the determination of a tight lepton for the event.

8.2 $\not\!\!E_T$ studies

The $\not\!\!E_T$ correction, as described above, is different from both the standard muon-only correction in TopEventModule and the tl correction as implemented in the lepton + track analysis. It is therefore important to check how the $\not\!\!E_T$ resolution after this new correction compares to the resolution after the other two corrections. It seems that this correction is indeed a modest improvement over the other two. This is illustrated in Figures 2 and 3. Figure 2 shows that the $\not\!\!E_T$ resolution as implemented in TopAlgorithms is improved in $Z \to \mu\mu$ events, where the $\not\!\!E_T$ is expected to be zero. Figure 3 shows the improvement in the $\not\!\!E_T$ resolution for $t\bar{t}$ dilepton events for the new TopAlgorithms code relative to the lepton + track analysis code, and shows that the improvement is primarily from events containing muons.

[2] This is the `ttopei` sample which was used to calculate the acceptance for the lepton + track analysis

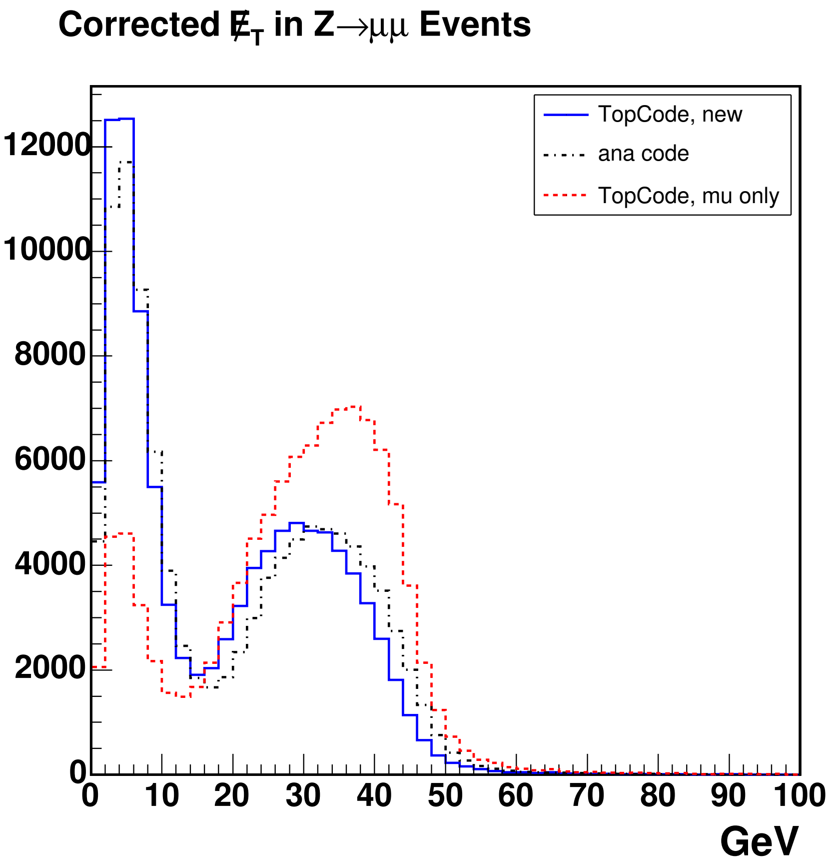

Figure 2: *Three versions of the corrected $\not{E}_T$ in $Z \to \mu\mu$. The lepton + track analysis code (in black) distribution is closer to zero than the distribution from the original TopAlgorithms tight-muon-only correction (in red), and the new TopAlgorithms correction (in blue) is a slight improvement over either. For this plot we used 300k events in the 4.11.2 $Z \to \mu\mu$ sample ztop0m, choosing events which have at least one tight CMUP or CMX muon. The bump at about 35 comes from muons which have been missed by the correction.*

We have also checked the effect of the new $\not{E}_T$ correction on the $t\bar{t}$ acceptance. It causes a slight (1%) relative decrease in the acceptance due to sloshing in the magnitude and direction of the $\not{E}_T$, which is well within the statistical error on the acceptance.

9 Summary

We have added a number of new features to the Top Group's source code. They have been validated and checked into the offline code, and are being refined as they start to be used in analyses.

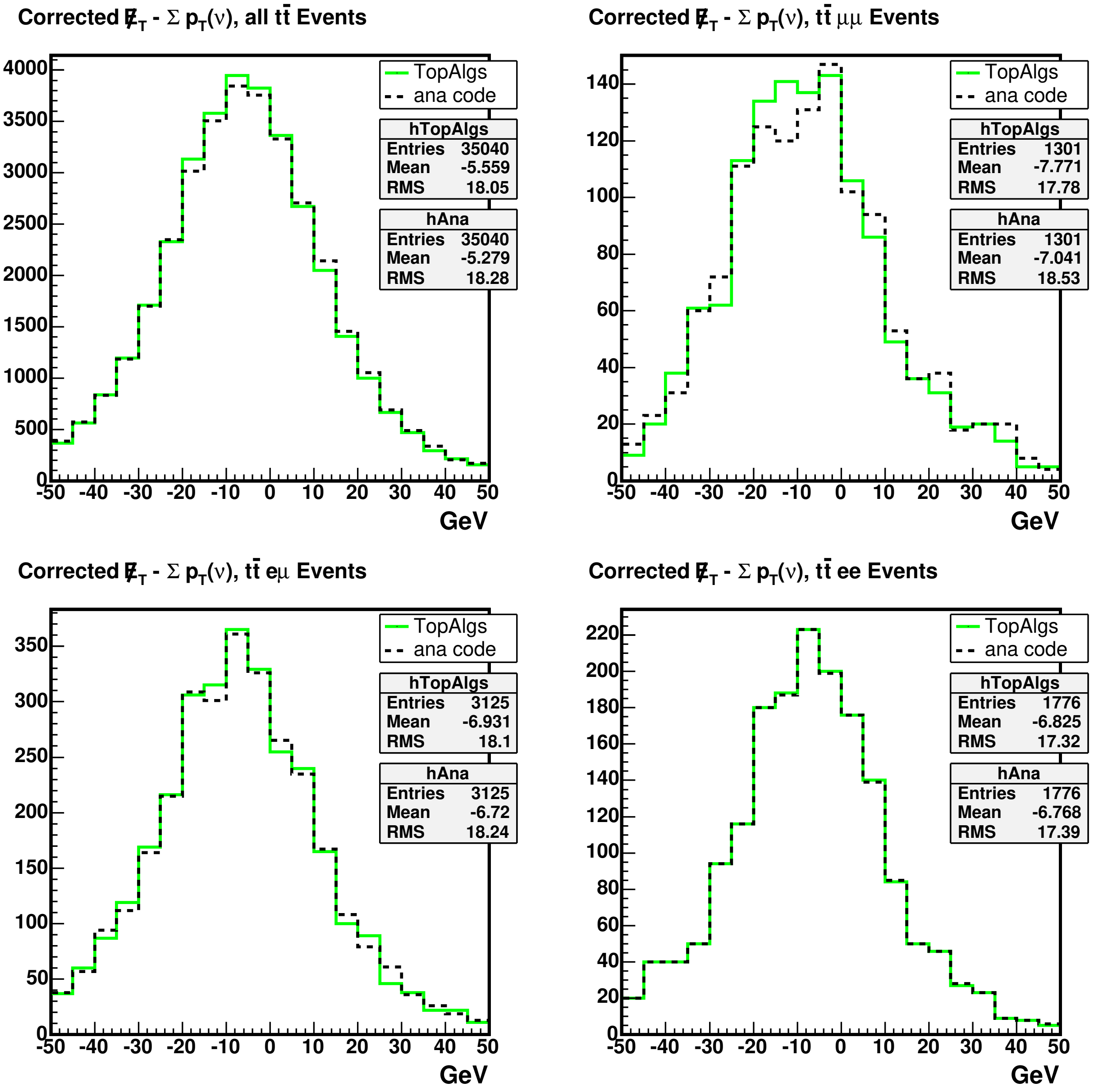

Figure 3: *The difference between the corrected* $\not{E}_T$ *and the sum* p_T *of the two neutrinos from the* W *decay is plotted for two versions of the* $\not{E}_T$ *correction: the new TopAlgorithms tl-corrected* $\not{E}_T$ *and the tl* $\not{E}_T$ *correction from the lepton + track analysis. Ideally, the corrected* $\not{E}_T$ *and the sum* p_T *of the two neutrinos should be the same, so the more centered the distribution at zero, the better the resolution of the correction. Plots are made from 222k of* $t\bar{t}$ *events (ttopei), for events where all of the corrected* $\not{E}_T$*s are defined (basically, events with one or more tight leptons and a nonzero tl collection according to the above definition). The residuals are shown for all events (upper left),* $\mu\mu$ *events (upper right),* $e\mu$ *events (lower left) and ee events (lower right). The residuals are a bit narrower for the new correction, and from the plots split up by lepton flavor combinations, it evident that the improvement is from events containing muons.*

A tcl Switches

Here is a list of the relevant tcl switches and their appropriate settings to use the above features. If you have changed the `newTopCollectionPrefix` to something other than MyTop, remember to substitute your new prefix for "MyTop" everywhere here!

A.1 `module talk TopEventModule`

`selectTl set true :` Main menu, defaults to *false*. This is the main switch for all of the tl event selection. If you don't set this to true, the tl collection will not be selected, the tl $\not\!\!E_T$ correction will not happen, and the tl dilepton event flags will not be set.

`leptonIdCuts set 2 :` Main menu, defaults to 3. If you set this reasonable values will be set for the tl selection (same as in lepton + track analysis and what's listed above). Otherwise the selection defaults to values such that all defTracks will pass, or you can set the values yourself.

`leptonTypes add TPE :` Main menu. This adds Phoenix electrons to the total lepton list in A. Kovalev's new lepton selection framework.

`leptonTypesTight add TPE :` Main menu. This adds Phoenix electrons to the total lepton list in A. Kovalev's new lepton selection framework.

`correctMetForTrackLeptons set true : met` menu, defaults to *true*. Turns on and off the tl $\not\!\!E_T$ correction. This will just fail if selectTl is false.

`correctMetForCorrJetsTl set true : met` menu, defaults to *false*. Turns on and off the L5 jet correction. This will just fail if selectTl is false.

`reclusterJetAlgorithmList add DJETS`

`reclusterJetConeRadiusList add 0.4:` These two are both in the `jets` menu. They add the DJETS to the list of reclustered jet collections to be created.

`tlDileptonEventTlPtCut set <value>: tl` menu, defaults to 20. Sets the tl p_T cut to be used in TopAlgorithms::isTlDileptonEvent.

`tlDileptonEventTrkIsolCut set <value>: tl` menu, defaults to 0.9. Sets the tl track isolation cut to be used in TopAlgorithms::isTlDileptonEvent.

A.2 `module talk TopNtuple`

`newTopCollectionPrefix set MyTop :` Main menu, defaults to "TopVal". If this switch isn't set correctly you won't pick up any of the tl information in your ntuple.

`jetList add MyTop_DJETS0.4_ReclusterJetColl :` Main menu. Add this line to pick up DJETS.

`otrkList add MyTopTrackLeptonView` **:** Main menu. Add this line to pick up track lepton collection.

References

[1] D. Glenzinski et al, *Signal and Background Acceptance for the Tight Lepton and Isolated Track Sample*, CDF Note 6532.

[2] D. Glenzinski et al, *The Fake Rate for the Tight Lepton and Isolated Track Sample*, CDF Note 6524.

[3] D. Glenzinski et al, *Results from the Top Dilepton Analysis of the Tight Lepton and Isolated Track Sample*, CDF Note 6562.

[4] T. Nelson et al, *Forward Electron Tracking with the PhoenixMods Package*, CDF Note 6278.