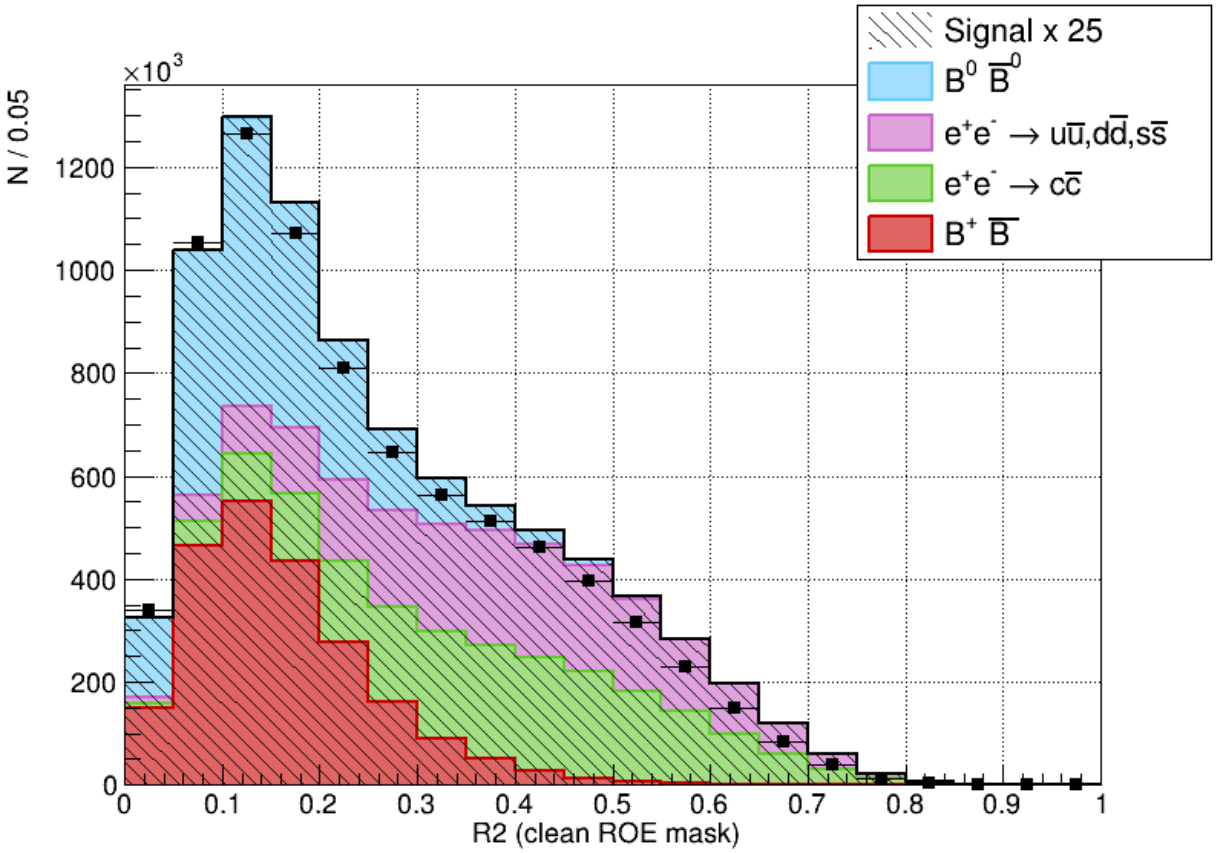
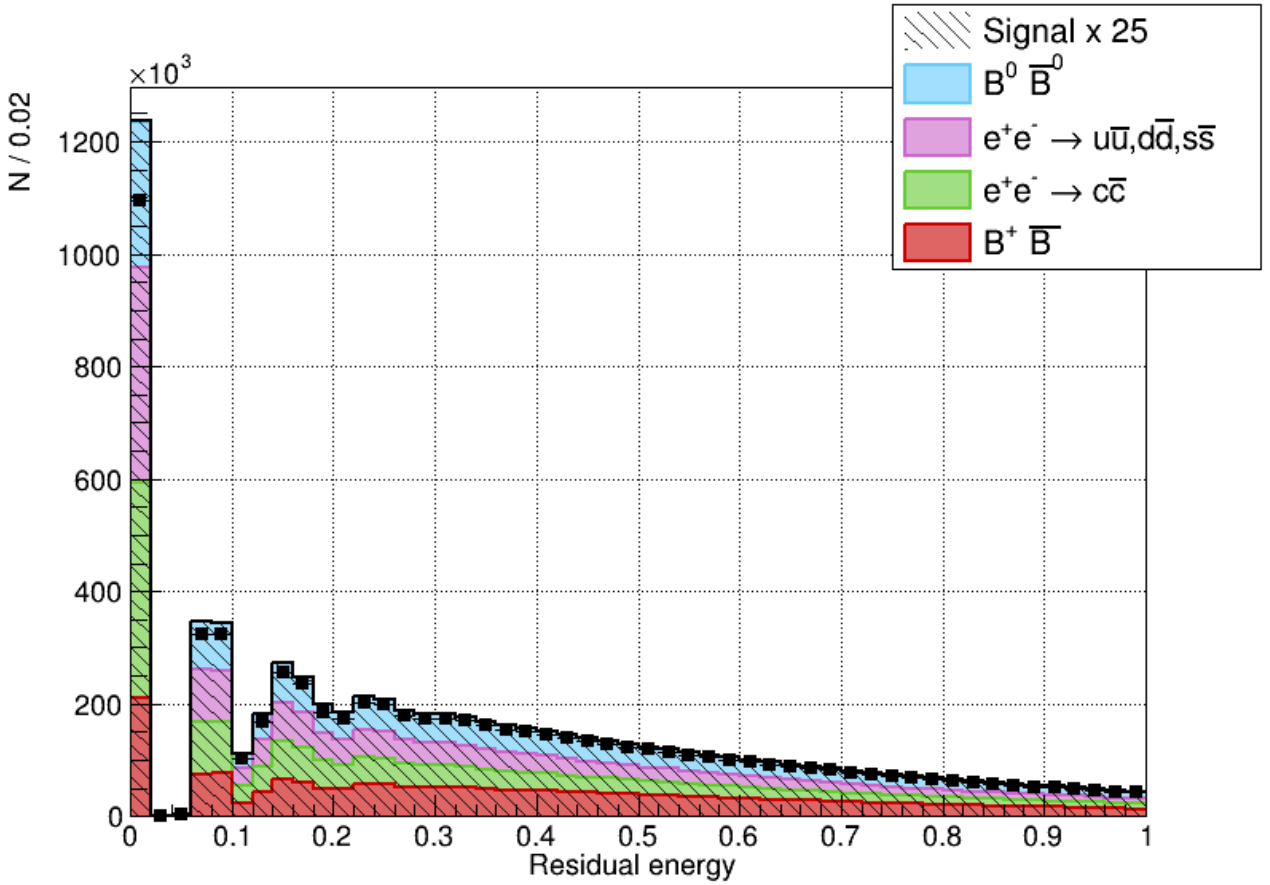
# $B \to D^* \tau \nu$ Update

MATEUSZ KALETA

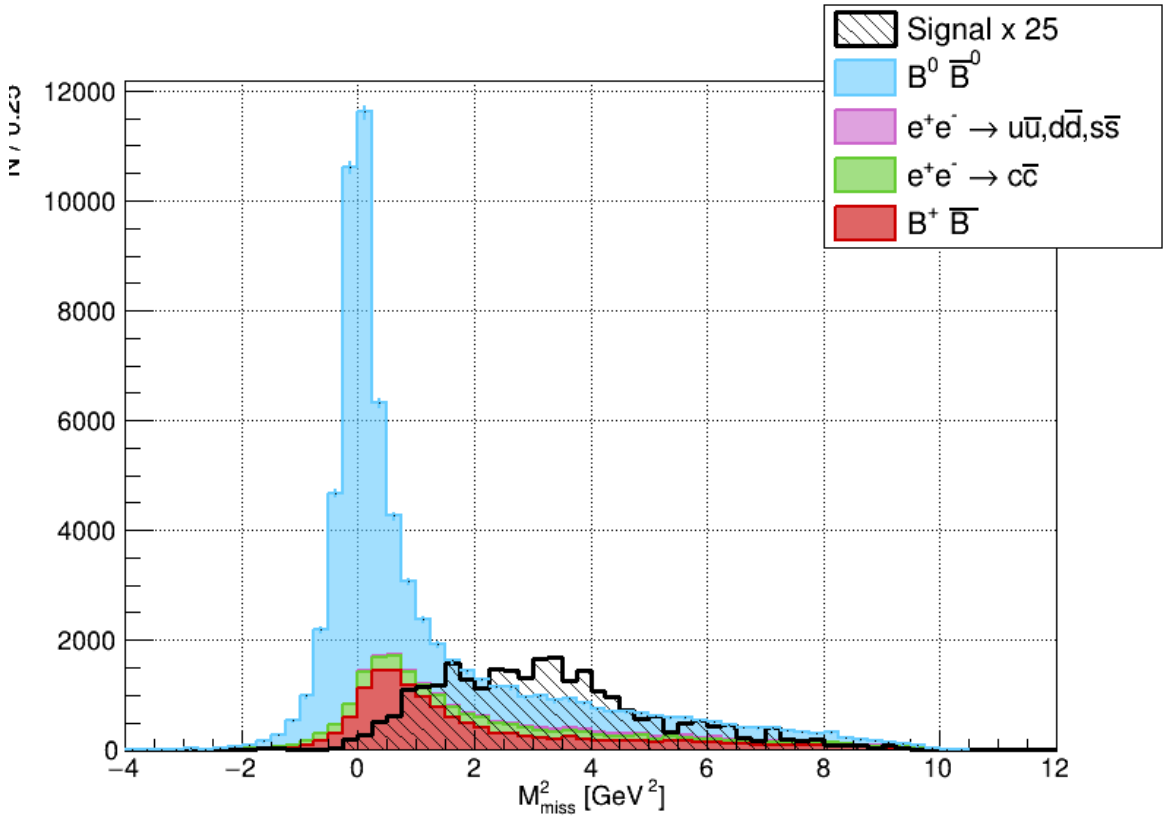
# Additional variables: R2, $E_{res}$

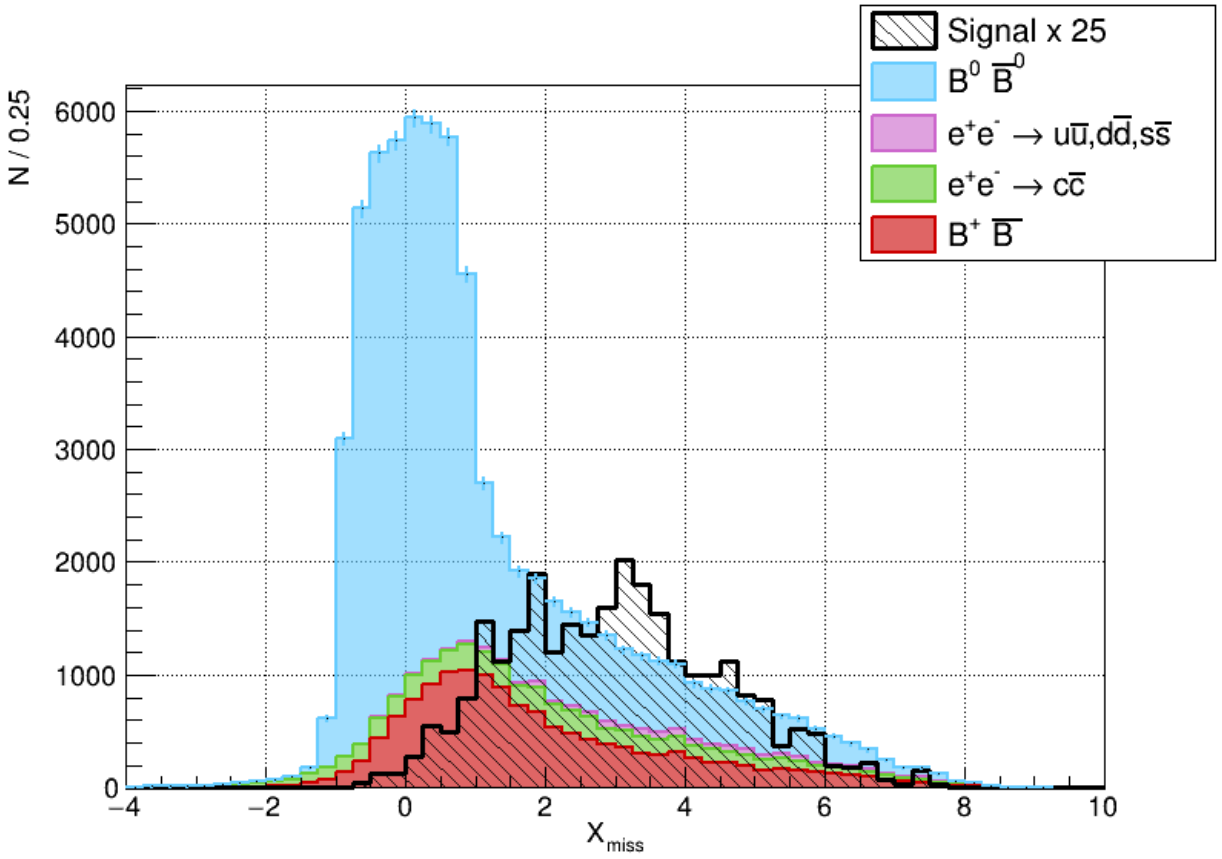



- Whole Belle dataset vs. one stream MC Generic
- Inside $D^0$ mass region: 1.83 GeV < M($D^0$ ) < 1.89 GeV
- Inside $M_{tag}$ mass region: $M_{tag}$ > 5.26 GeV

# Additional signal-side variables: $M^2_{miss}$, $X_{miss}$




- Whole one stream MC Generic
- Inside $D^0$ mass region: 1.83 GeV < M($D^0$ ) < 1.89 GeV
- Inside $M_{tag}$ mass region: $M_{tag}$ > 5.26 GeV

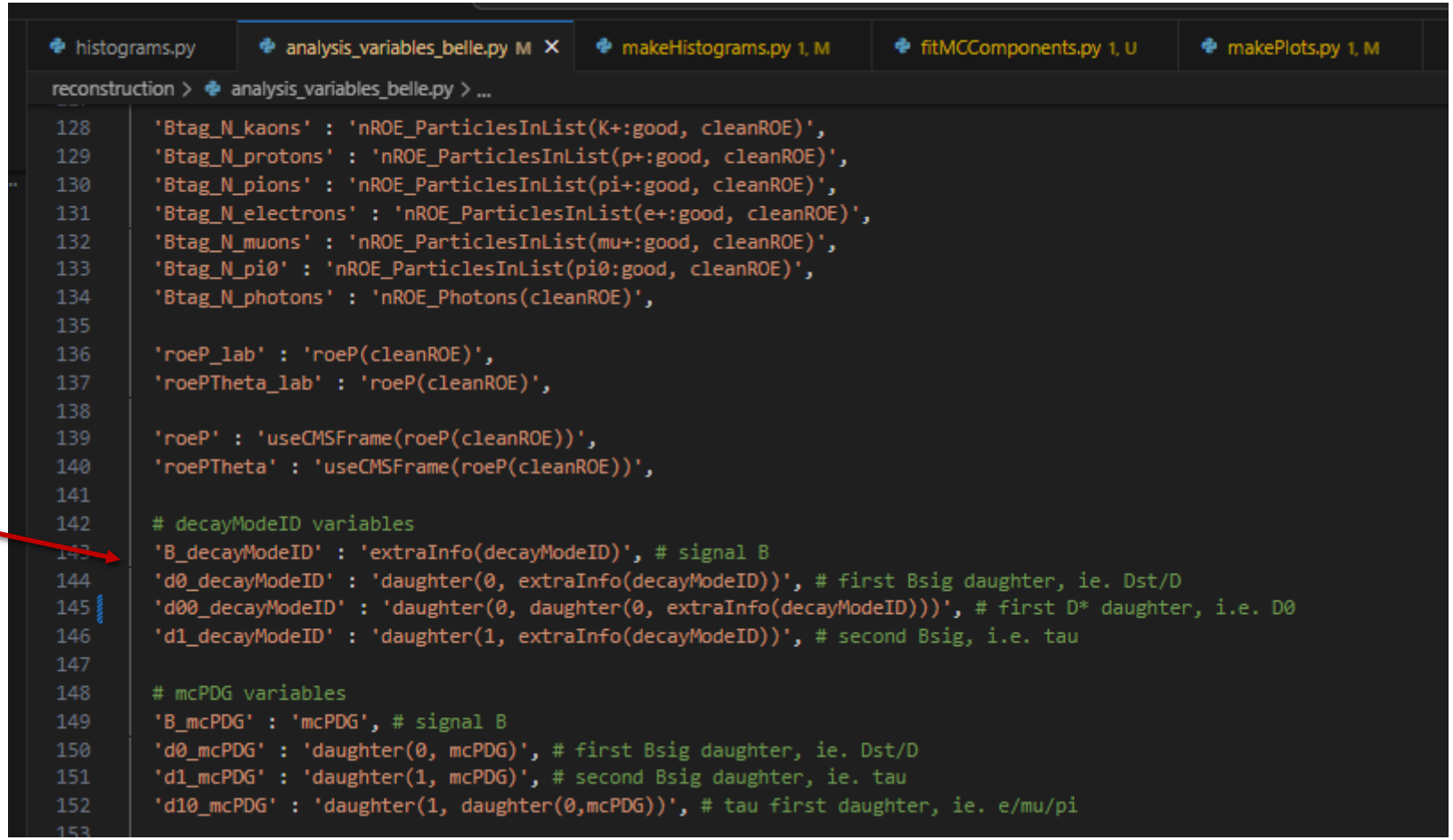
# Background calibration: physical components

Physical background components:

(separate for B0/B+ channels)

- B -> hadronic
- B -> D*lnu
- B -> D**

Need to reproces all ntuples including decay ID information



# Fitting with TFracionFitter

- Adopted TFractionFitter code to python and integrated in the analysis
- Problem: currently works for a single variable (e.g. X_miss)
- Need to add multidimensional fits in selected variables

histograms.py | analysis_variables_belle.py M | makeHistograms.py 1, M | fitMCComponents.py 1, U | makePlots.py 1, M

analysis > control_plots > fitMCComponents.py > ...

```python
import ROOT

# ------------------------------------------------------------
# 1. Create or retrieve your histograms:

file = "histograms/histograms_preselection_B0_all_components.root"

data = file.Get("data_Xmiss_A")
mc0  = file.Get("data_Xmiss_A")
mc1  = file.Get("data_Xmiss_A")
mc2  = file.Get("data_Xmiss_A")

# ------------------------------------------------------------
# 2. Put the MC histograms into a TObjArray
# ------------------------------------------------------------
mc_array = ROOT.TObjArray(3)
mc_array.Add(mc0)
mc_array.Add(mc1)
mc_array.Add(mc2)

# ------------------------------------------------------------
# 3. Create the TFractionFitter
# ------------------------------------------------------------
fit = ROOT.TFractionFitter(data, mc_array)

# ------------------------------------------------------------
# 4. Constrain fraction(s)
#    Here, we constrain the fraction of mc1 (index=1)
#    to be between 0 and 1
# ------------------------------------------------------------
fit.Constrain(1, 0.0, 1.0)

# ------------------------------------------------------------
# 5. Specify the bin range to use
#    (e.g., use bins 1 through 15 for the fit)
# ------------------------------------------------------------
fit.SetRangeX(1, 15)

# ------------------------------------------------------------
# 6. Perform the fitf
# ------------------------------------------------------------
status = fit.Fit()
print("Fit status:", status)

# ------------------------------------------------------------
```