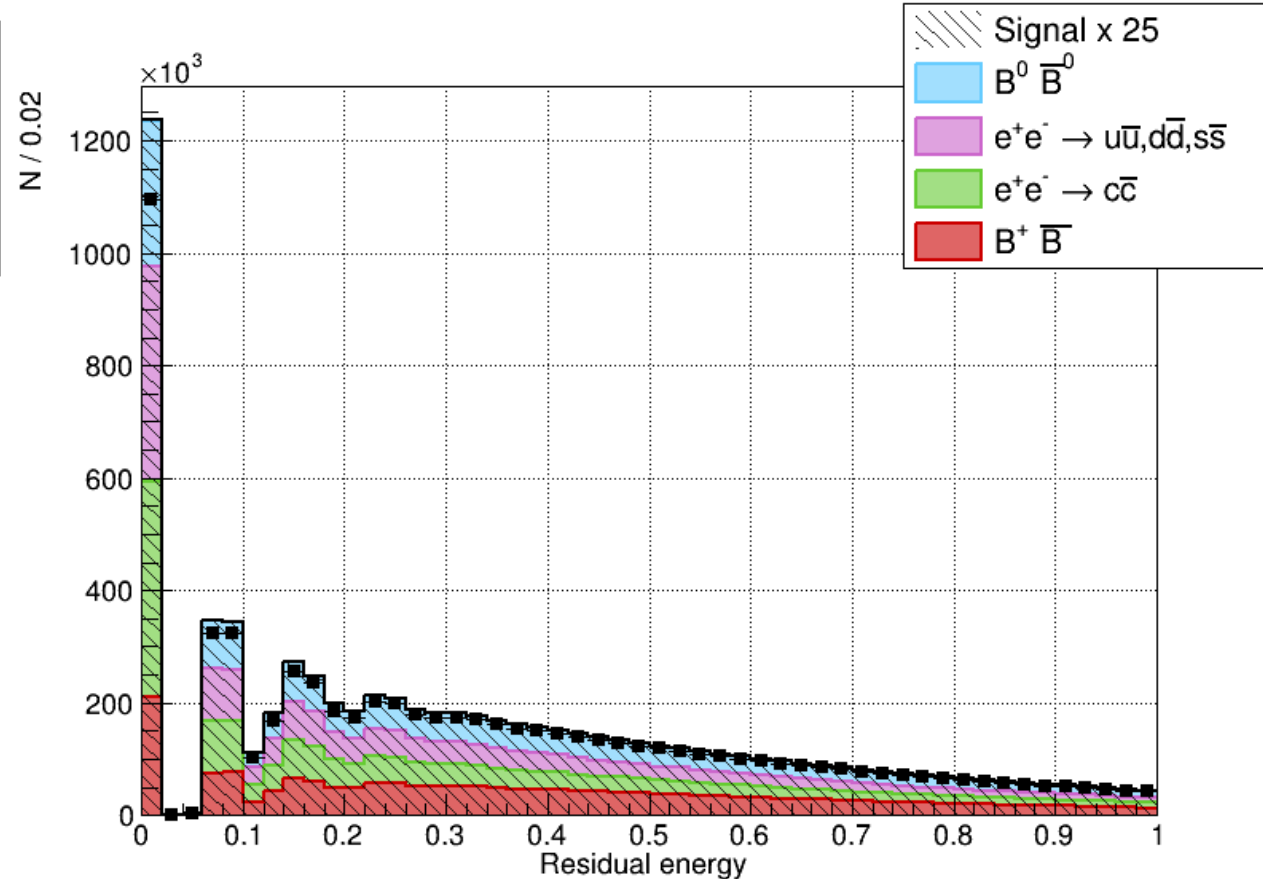
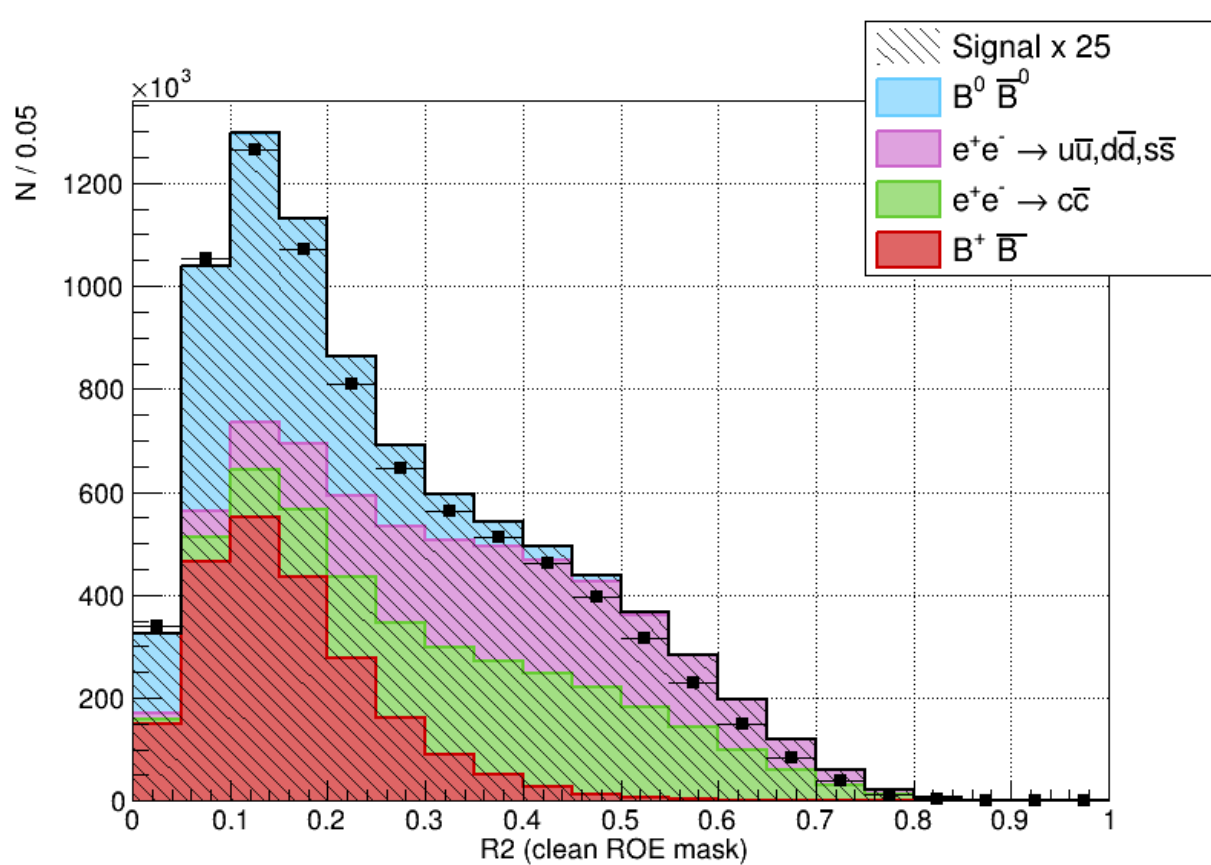


B→D* τ v Update

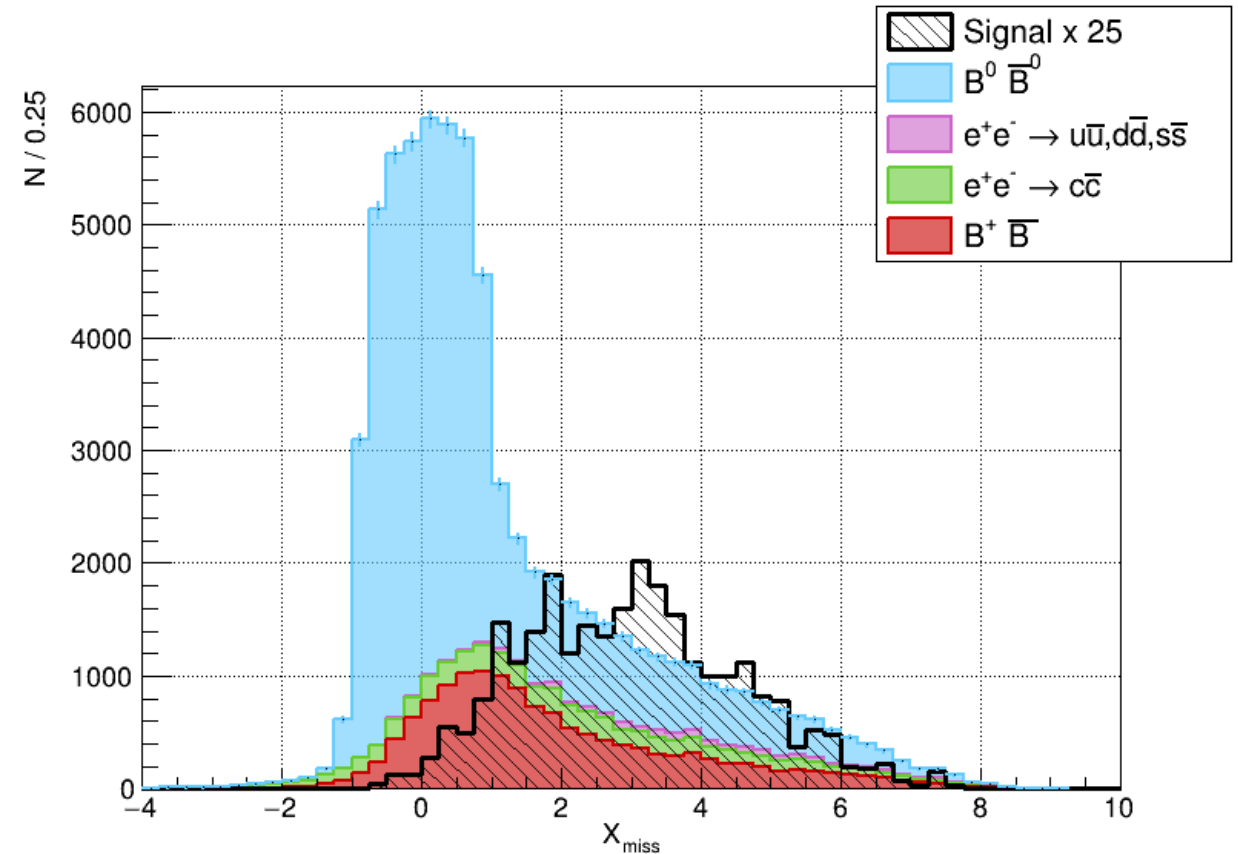
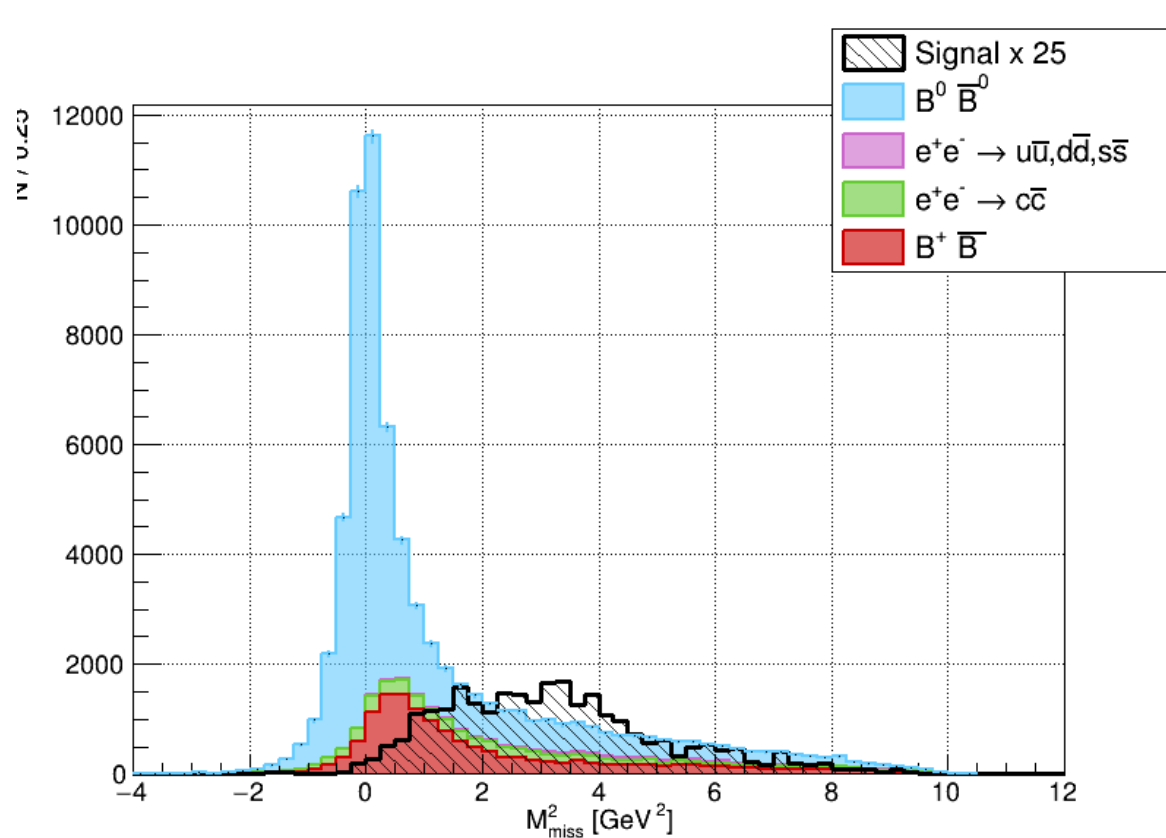
MATEUSZ KALETA

Additional variables: R_2 , E_{res}



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

Additional signal-side variables: M_{miss}^2 , X_{miss}



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

Background calibration: physical components

Physical background components:

(separate for B0/B+ channels)

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

Need to reprocess all ntuples including decay ID information

```
reconstruction > analysis_variables_belle.py > ...
128 'Btag_N_kaons' : 'nROE_ParticlesInList(K+:good, cleanROE)',
129 'Btag_N_protons' : 'nROE_ParticlesInList(p+:good, cleanROE)',
130 'Btag_N_pions' : 'nROE_ParticlesInList(pi+:good, cleanROE)',
131 'Btag_N_electrons' : 'nROE_ParticlesInList(e+:good, cleanROE)',
132 'Btag_N_muons' : 'nROE_ParticlesInList(mu+:good, cleanROE)',
133 'Btag_N_pi0' : 'nROE_ParticlesInList(pi0:good, cleanROE)',
134 'Btag_N_photons' : 'nROE_Photons(cleanROE)',
135
136 'roeP_lab' : 'roeP(cleanROE)',
137 'roePTheta_lab' : 'roeP(cleanROE)',
138
139 'roeP' : 'useCMSFrame(roeP(cleanROE))',
140 'roePTheta' : 'useCMSFrame(roeP(cleanROE))',
141
142 # decayModeID variables
143 'B_decayModeID' : 'extraInfo(decayModeID)', # signal B
144 'd0_decayModeID' : 'daughter(0, extraInfo(decayModeID))', # first Bsig daughter, ie. Dst/D
145 'd00_decayModeID' : 'daughter(0, daughter(0, extraInfo(decayModeID)))', # first D* daughter, i.e. D0
146 'd1_decayModeID' : 'daughter(1, extraInfo(decayModeID))', # second Bsig, i.e. tau
147
148 # mcPDG variables
149 'B_mcPDG' : 'mcPDG', # signal B
150 'd0_mcPDG' : 'daughter(0, mcPDG)', # first Bsig daughter, ie. Dst/D
151 'd1_mcPDG' : 'daughter(1, mcPDG)', # second Bsig daughter, ie. tau
152 'd10_mcPDG' : 'daughter(1, daughter(0,mcPDG))', # tau first daughter, ie. e/mu/pi
153
```

Fitting with TFractionFitter

- 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

```
analysis > control_plots > fitMCComponents.py > ...
1  import ROOT
2
3  # -----
4  # 1. Create or retrieve your histograms:
5
6  file = "histograms/histograms_preselection_B0_all_components.root"
7
8  data = file.Get("data_Xmiss_A")
9  mc0 = file.Get("data_Xmiss_A")
10 mc1 = file.Get("data_Xmiss_A")
11 mc2 = file.Get("data_Xmiss_A")
12
13 # -----
14 # 2. Put the MC histograms into a TObjArray
15 # -----
16 mc_array = ROOT.TObjArray(3)
17 mc_array.Add(mc0)
18 mc_array.Add(mc1)
19 mc_array.Add(mc2)
20
21 # -----
22 # 3. Create the TFractionFitter
23 # -----
24 fit = ROOT.TFractionFitter(data, mc_array)
25
26 # -----
27 # 4. Constrain fraction(s)
28 # Here, we constrain the fraction of mc1 (index=1)
29 # to be between 0 and 1
30 # -----
31 fit.Constrain(1, 0.0, 1.0)
32
33 # -----
34 # 5. Specify the bin range to use
35 # (e.g., use bins 1 through 15 for the fit)
36 # -----
37 fit.SetRangeX(1, 15)
38
39 # -----
40 # 6. Perform the fit
41 # -----
42 status = fit.Fit()
43 print("Fit status:", status)
44
45 # -----
```