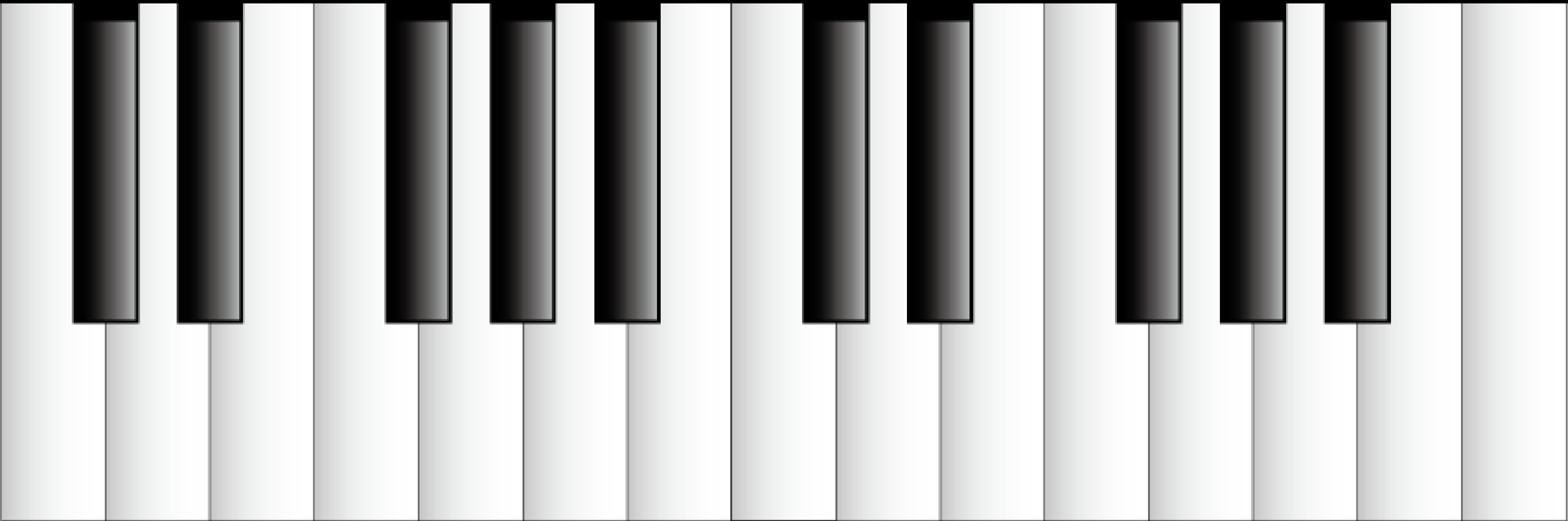
Algorithms for HEP data analysis

SCT Soft&Phys WP meeting
28.12.2018
Vitaly Vorobyev



Classic event loop (Belle)

```
class UserModule : public Module {
 public:
    b2ddsj(void);
    void init();
    void term();
    void begin_run();
    void event();
    void end_run();
};
```

High-level interface (Belle II) I

```
import basf2 as b2
import modularAnalysis as ma
path = b2.create_path()
ma.inputMdstList("MC10", [input_file], path=path)

from stdCharged import stdK, stdPi
stdK("higheff", path=path)
stdPi("higheff", path=path)

ma.reconstructDecay('D0:Kpi -> K-:higheff pi+:higheff', '1.7 < M < 1.9', path=path)
ma.fitVertex('D0:Kpi', 0.1, path=path)
ma.matchMCTruth('D0:Kpi', path=path)

ma.reconstructDecay('B- -> D0:Kpi pi-:higheff', '5.2 < Mbc < 5.3', path=path)
ma.fitVertex('B+', 0.1, path=path)
ma.matchMCTruth('B-', path=path)
```

High-level interface (Belle II) II

```
ma.variablesToNtuple('D0:Kpi', ['M', 'p', 'E', 'useCMSFrame(p)', 'useCMSFrame(E)',
             'daughter(0, kaonID)', daughter(1, pionID)', 'isSignal', 'mcErrors'],
             filename='D.root', path=path)

ma.variablesToNtuple('B-', ['Mbc', 'deltaE', 'isSignal', 'mcErrors', 'M'],
                     filename='B.root', path=path)
```

```
from ipython_tools import handler
calculation = handler.process(path, max_event = 100)
calculation.show_path()
calculation.start()
calculation.wait_for_end()
calculation.show_log()
```

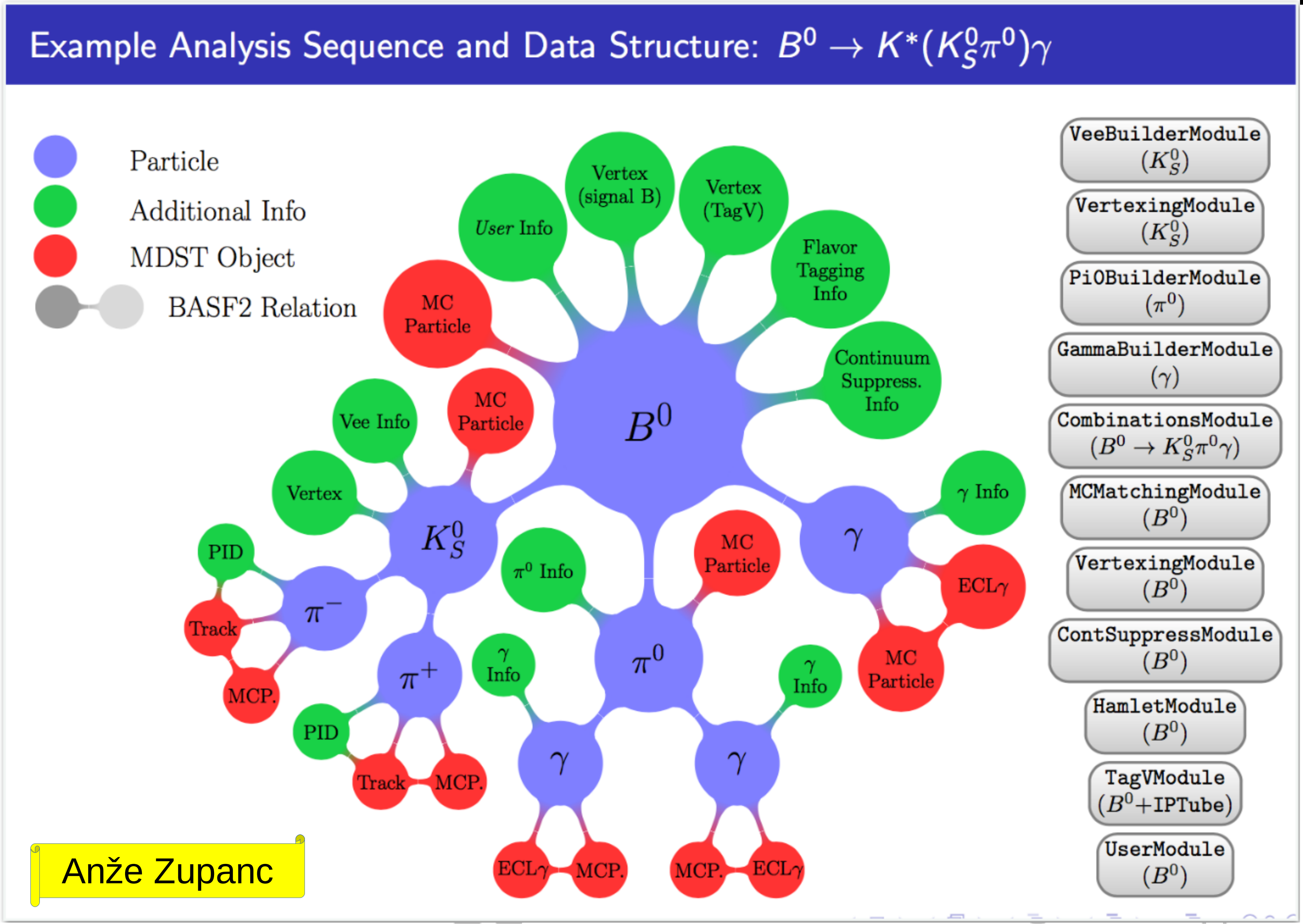
User’s actions are well-defined

- Particle
  - Particle, ParticleList
  - ParticleLoader,
  - FillParticleList
  - Standard particle lists
- Vertex
  - Vertex fitter functions
  - TreeFitter
- Variables
  - VariableManager
- Offline data
  - VariablesToNtuple
  - VariablesToHistogram
  - VariablesToEventBasedTree

- Truth
  - MC Decay Finder
  - Truth Matching
- Vetoing
  - Photon proximity veto
  - MVA pi0/eta veto
- Advanced
  - Full Event Interpretation
  - RestOfEvent
  - ContinuumSuppression
  - Global EventShape
  - EventKinematics
  - FlavorTagger
  - TagVertex
  - TMVA interface

Belle II algorithms

Belle II analysis example



Example Analysis Sequence and Data Structure: $B^0 \to K^*(K^0_S\pi^0)\gamma$

- 🟦 Particle
- 🟩 Additional Info
- 🟥 MDST Object
- ● ○ BASF2 Relation

```
VeeBuilderModule
($K^0_S$)
VertexingModule
($K^0_S$)
Pi0BuilderModule
($\pi^0$)
GammaBuilderModule
($\gamma$)
CombinationsModule
($B^0 \to K^0_S\pi^0\gamma$)
MCMatchingModule
($B^0$)
VertexingModule
($B^0$)
ContSuppressModule
($B^0$)
HamletModule
($B^0$)
TagVModule
($B^0$+IPTube)
UserModule
($B^0$)
```

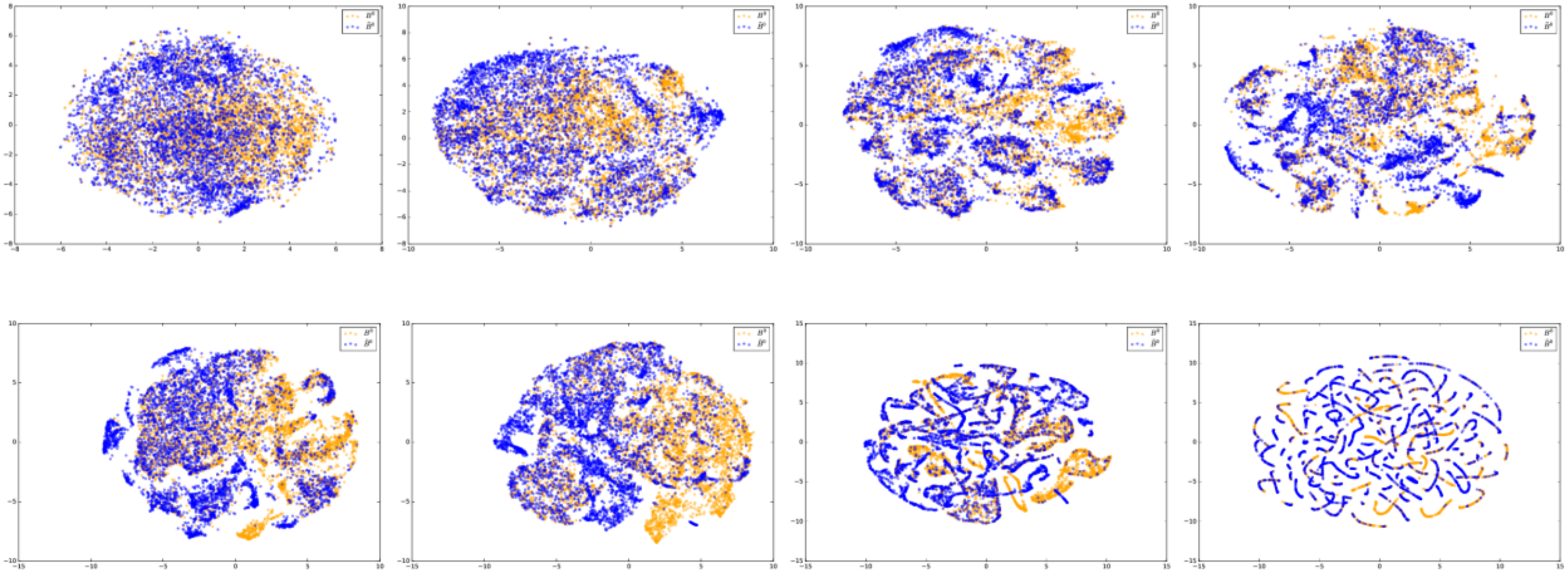
Anže Zupanc

What we want from a good analysis tools

- Well-documented and easy-to-learn
- Comprehensive
- Flexible
- Reproducible
- Short development time7

Deep-learning revolution

Deep Flavour Tagger by Jochen Gemmler @ KIT using Tensorflow



Neuron activity in different layers for B^0 and $\overline{B}^0$ using t-SNE projection

Data science tools

- Python
  - numpy
  - pandas
  - matplotlib
  - jupyter notebook
- Machine learning
  - scikit-learn
  - tensorflow
  - xgboost
  - catboost
  - ...