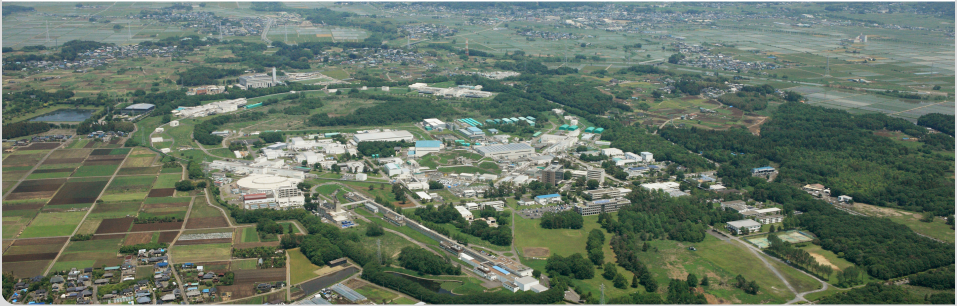
Karlsruhe Institute of Technology

Belle II

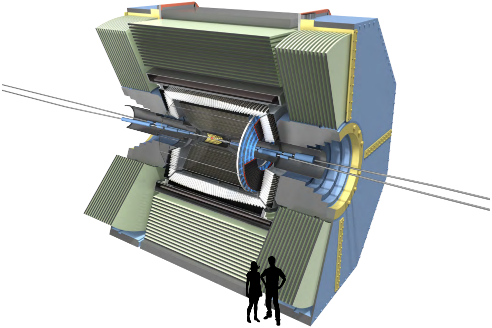
An automated framework for hierarchical reconstruction of B mesons at the Belle II experiment

Christian Pulvermacher, Thomas Keck, Michael Feindt, Martin Heck, Thomas Kuhr | 2 September 2014

INSTITUT FÜR EXPERIMENTELLE KERNPHYSIK (IEKP)



The Belle II Experiment



- Experiment at the SuperKEKB B factory being built in Tsukuba, Japan
- Successor of Belle (1999–2010), with 40$\times$ higher luminosity and upgraded detector
- Planned start in 2016

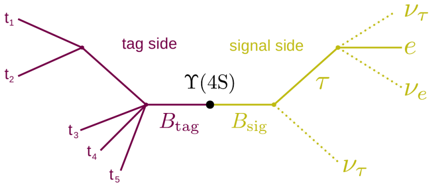
$\Upsilon(4S) \to B\bar{B}$

B factories produce large amounts of $\Upsilon(4S)$ mesons, which decay to almost 100 % into B^+B^- or $B^0\bar{B}^0$.

Tag Side Reconstruction

Reconstructing one of the two B mesons allows for improved signal-side reconstruction:

- B_{sig} momentum known without any B_{sig} reconstruction
- $B \to \tau\nu$ – Require no remaining tracks $\Rightarrow$ increases purity
- $B^+ \to l^+ \nu \gamma$ (with $l^+ = e^+/\mu^+$)
  Use four-momentum to calculate m_{miss}, expect correctly reconstructed events at $m_{\text{miss}} = 0$.



In other words: reconstruct $\Upsilon(4S)$ to improve B_{sig}.

Hierarchical Reconstruction

To avoid the computational overhead of constructing each decay exclusively, reconstruction is separated into stages:

- **Final state**: Multivariate classifier can improve PID
- **Intermediate**: Combine final state particles (or intermediates) in different decay channels with own classifier – also incorporates soft cuts to reduce computation time
- **B^0/B^+**: Reconstruct B mesons based on the information from preceding stages, provide classifier output.

$\Rightarrow \mathcal{O}(100)$ classifiers handling $\mathcal{O}(1000)$ exclusive B decay modes.

Hierarchical Reconstruction

To avoid the computational overhead of constructing each decay exclusively, reconstruction is separated into stages:

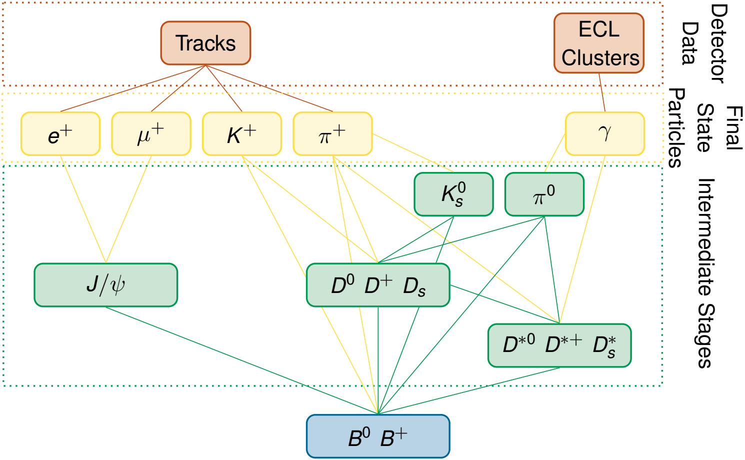
- **Final state** Multivariate classifier can improve PID
- **Intermediate** Combine final state particles (or intermediates) in different decay channels with own classifier – also incorporates soft cuts to reduce computation time
- **B^0/B^+** Reconstruct B mesons based on the information from preceding stages, provide classifier output.

$\Rightarrow \mathcal{O}(100)$ classifiers handling $\mathcal{O}(1000)$ exclusive B decay modes.

Efficiency

At Belle, a similar approach yielded $\approx 0.3\,\%$ efficiency, corresponding to $\approx 2.3 \cdot 10^6$ correctly reconstructed B_{tag}.

Hierarchical Reconstruction



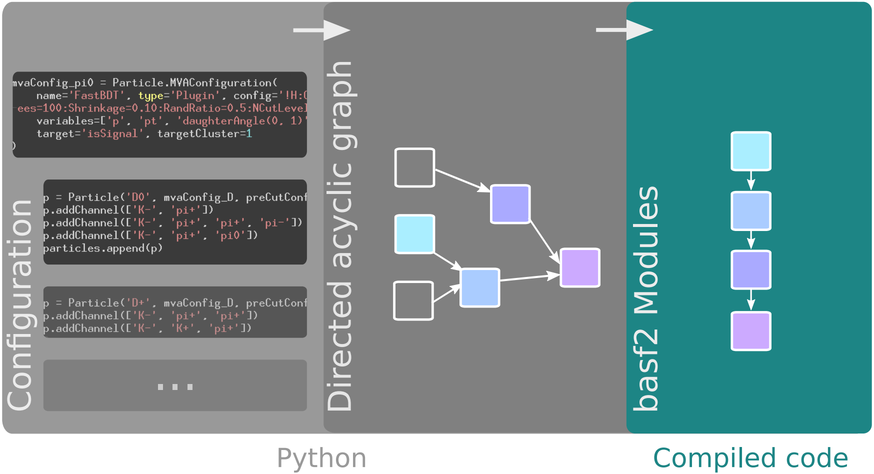
Main Goals

- Generic framework for meta-reconstruction
- User-friendly interface, simple addition of channels
- Automatisation
  - Dependency resolution
  - Determination of necessary cuts
  - Plots/statistics for evaluation
- Build on existing high-level reconstruction tools

Motivation

- Changing reconstruction from hadronic channels to include semileptonic channels, or to do the same on $\Upsilon(5S)$ requires only configuration changes.
- Signal selection changes tag-side priors, introduces bias that can by avoided by retraining (by analysis authors).

Architecture



Channel Configuration

Belle II analysis software framework (basf2) uses Python for steering files; this allows us to configure using a high-level abstraction:

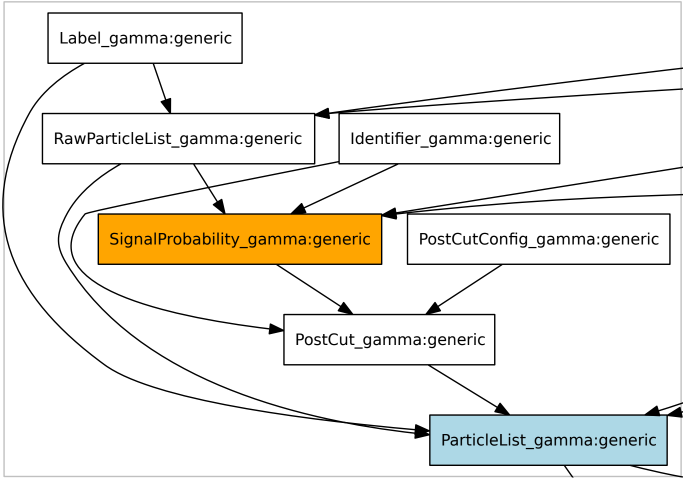
```
#...
p = Particle('B0', mvaConfig_B, preCutConfiguration_B, postCutConfig_B)
p.addChannel(['D0', 'pi0'])
p.addChannel(['D-', 'pi+'])
p.addChannel(['D-', 'pi0', 'pi+'])
p.addChannel(['D*-', 'pi+', 'pi+', 'pi-'])
p.addChannel(['D*-', 'pi+', 'pi+', 'pi-', 'pi0'])
particles.append(p)

FullEventInterpretation(modulePath, particles)
```

Configuration converted into tasks — e.g. for combining particles, vertexing, multivariate classification — with inputs and outputs that define their dependencies.

Dependency Resolution

Dependencies are represented by a directed acyclic graph, e.g.:



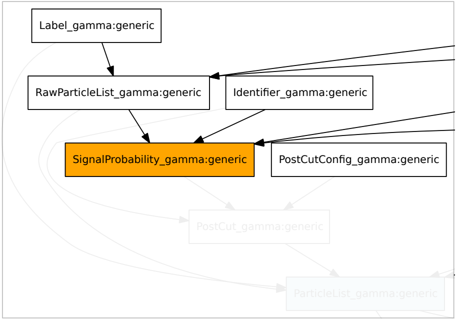
Dependency Resolution

Tasks may not provide output immediately, e.g. multivariate classification requires a training file.

Dependency Resolution

Tasks may not provide output immediately, e.g. multivariate classification requires a training file.

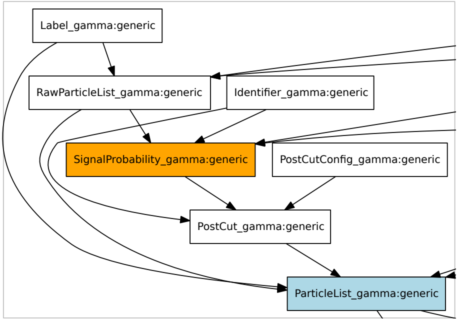
- When first run, SignalProbability will create a training file and provide nothing.



Dependency Resolution

Tasks may not provide output immediately, e.g. multivariate classification requires a training file.

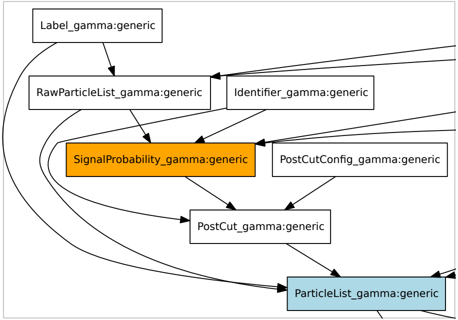
- When first run, SignalProbability will create a training file and provide nothing.
- If training file present, provide classification output.



Dependency Resolution

Tasks may not provide output immediately, e.g. multivariate classification requires a training file.

- When first run, SignalProbability will create a training file and provide nothing.
- If training file present, provide classification output.



Configuration changes are propagated by including hashes that depend on **all** previous input in file names, lists of particles, . . .

Executing basf2 Modules

Data-intensive tasks translate to high-level reconstruction tools (also used for manual analysis):

```
selectParticle('gamma', '', path=main)

reconstructDecay('pi0 -> gamma gamma', '0.11 < M < 0.15', path=main)
matchMCTruth('pi0', path=main)

#...
```

Actual reconstruction, selection, classification is handled by compiled modules (written in C++11).

Automatic Cut Determination

To reduce combinatorical background in the reconstruction, cuts need to be applied in between stages. When combining particles, we might cut on:

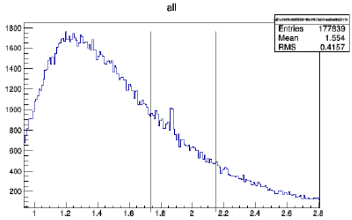
- Invariant mass of candidate
- Product of output probabilities of candidate's daughters
- $M(D^*) - M(D)$ for D^* mesons
- ...

Instead of deciding manually on cuts, they are determined automatically from signal/background ratio histograms according to user-provided efficiency and purity values.

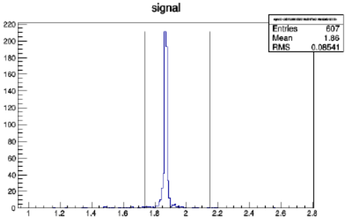
Automatic Cut Determination

1 Overview: D+ ==> K- pi+ pi+

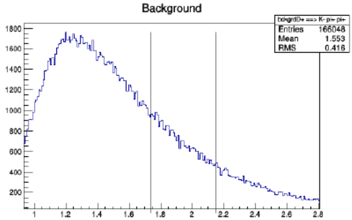
1.1 Pre-cut determination



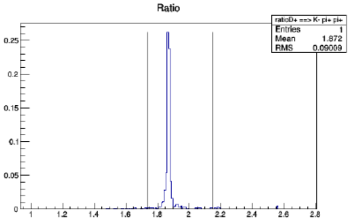
(a) All possible combinations



(b) Only signal combinations



(c) Only background combinations



(d) Ratio of signal and background histograms

TMVA Interface

Multivariate classifications are performed via modules that calculate a given set of variables on particles and pass them to TMVA:

```
selectParticle('e-', path=main)
#...
teacher = register_module('TMVATeacher')
teacher.param('methods', methods)
teacher.param('variables', variables)
teacher.param('target', 'isSignal')
teacher.param('listNames', 'e-')
main.add_module(teacher)
```

Allows to easily compare different methods, configurations, and reuse of TMVA's evaluation plots.

Reporting

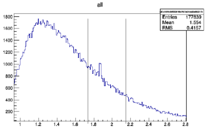
Even with the reuse of particles through the hierarchical approach, this quickly results in hundreds of modules.
Because of this, a reporting framework provides an overview of the performance of each particle and channel.

Reporting

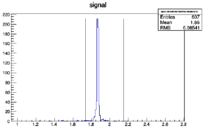
Even with the reuse of particles through the hierarchical approach, this quickly results in hundreds of modules.
Because of this, a reporting framework provides an overview of the performance of each particle and channel.

1 Overview: `D+ ==> K- pi+ pi+`

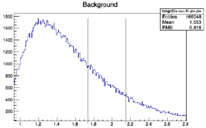
1.1 Pre-cut determination



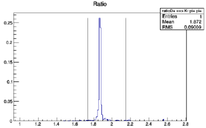
(a) All possible combinations



(b) Only signal combinations



(c) Only background combinations

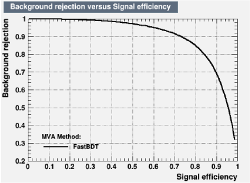


(d) Ratio of signal and background histograms

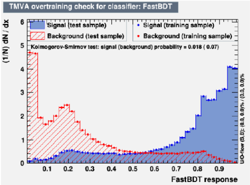
- Pre-cut plots

Reporting

Even with the reuse of particles through the hierarchical approach, this quickly results in hundreds of modules.
Because of this, a reporting framework provides an overview of the performance of each particle and channel.



(a) Purity over efficiency plot

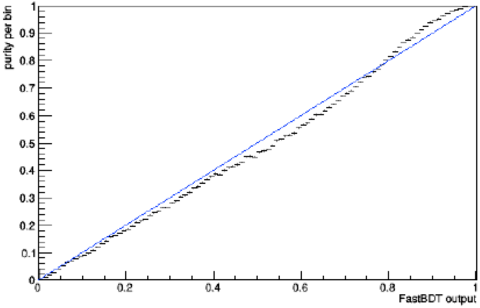


(b) Overtraining plot

- Pre-cut plots
- Plots to evaluate MVA classification, plus variables (ranked)

Reporting

Even with the reuse of particles through the hierarchical approach, this quickly results in hundreds of modules.
Because of this, a reporting framework provides an overview of the performance of each particle and channel.



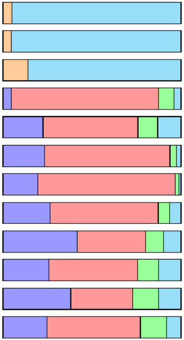
(c) Diag plot

- Pre-cut plots
- Plots to evaluate MVA classification, plus variables (ranked)

Reporting

Even with the reuse of particles through the hierarchical approach, this quickly results in hundreds of modules.
Because of this, a reporting framework provides an overview of the performance of each particle and channel.

Decay	CPU time	by module
pi+	2836.1	
K+	2574.7	
γ	812.0	
pi0 $\rightarrow$ γ γ	47013.9	
D0 $\rightarrow$ K- pi+	1348.8	
D0 $\rightarrow$ K- pi+ pi+ pi-	14229.7	
D0 $\rightarrow$ K- pi+ pi0	51061.6	
D+ $\rightarrow$ K- pi+ pi+	2749.7	
D+ $\rightarrow$ K- K+ pi+	997.7	
D*+ $\rightarrow$ D0 pi+	1040.0	
D*+ $\rightarrow$ D+ pi0	895.7	
D*0 $\rightarrow$ D0 pi0	7148.9	



- Pre-cut plots
- Plots to evaluate MVA classification, plus variables (ranked)
- CPU usage statistics

Reporting

Even with the reuse of particles through the hierarchical approach, this quickly results in hundreds of modules.
Because of this, a reporting framework provides an overview of the performance of each particle and channel.

Final state particle	Efficiency in %		Purity in %	
	detector	post-cut	detector	post-cut
pi+	81.4	73.4	69.237	73.260
K+	79.9	71.6	12.282	28.639
gamma	89.0	86.8	53.737	54.294

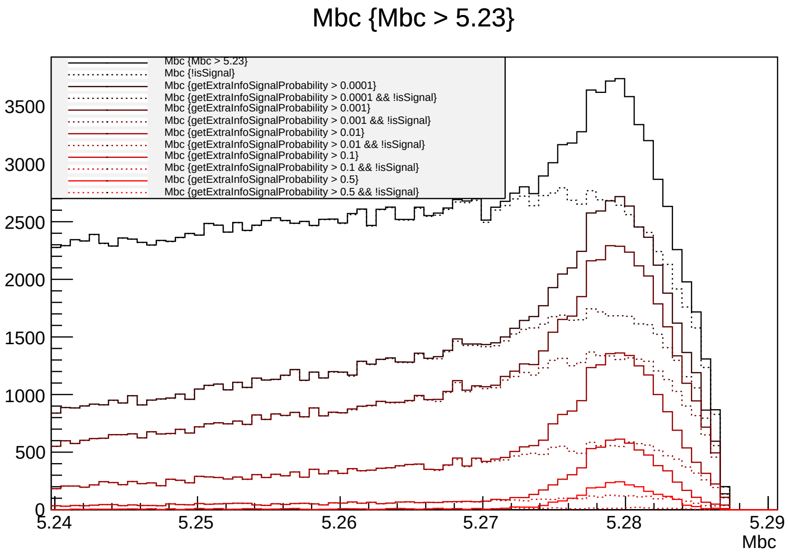
For each decay channel of each intermediate particle a multivariate analysis method was trained after performing a fast pre-cut on the candidates. Afterwards the signal probability calculated by the method was used to perform a post-cut on the intermediate particle candidates. This reduces combinatorics in the following stages of the full event interpretation.

Table 2: Per-particle efficiency and purity before and after the applied pre- and post-cut.

Particle	Efficiency in %			Purity in %		
	detector	pre-cut	post-cut	detector	pre-cut	post-cut
pi0	85.1	65.9	60.9	2.564	5.083	5.235
$D0$	10.6	8.1	6.5	0.014	0.465	0.669
$D+$	4.6	3.9	3.3	0.101	2.698	4.490
$D*+$	2.2	1.8	1.7	0.077	1.852	3.518
$D*0$	1.9	1.6	1.1	0.003	0.020	0.034
$B0$	0.1	0.0	0.0	0.000	0.008	0.008

- Pre-cut plots
- Plots to evaluate MVA classification, plus variables (ranked)
- CPU usage statistics
- Global efficiencies, purities

Plot of Beam-Constrained Mass for B^0_{tag}



Summary

We're building a meta-reconstruction framework that:

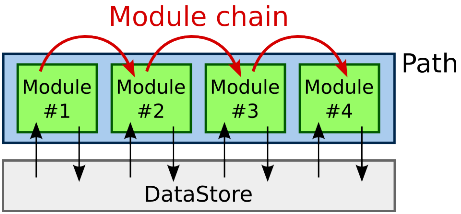
- reconstructs decays hierarchically, reducing the effort required to cover many exclusive channels,
- can easily be reconfigured to work on different resonances or include more channels,
- resolves dependencies between tasks and retrains only those affected by configuration changes,
- and automatically produces a set of plots and tables for evaluation.

In combination with a signal-side selection, this allows us to reconstruct $\Upsilon(4S) \rightarrow B\bar{B}$ and improve many analyses.

Backup

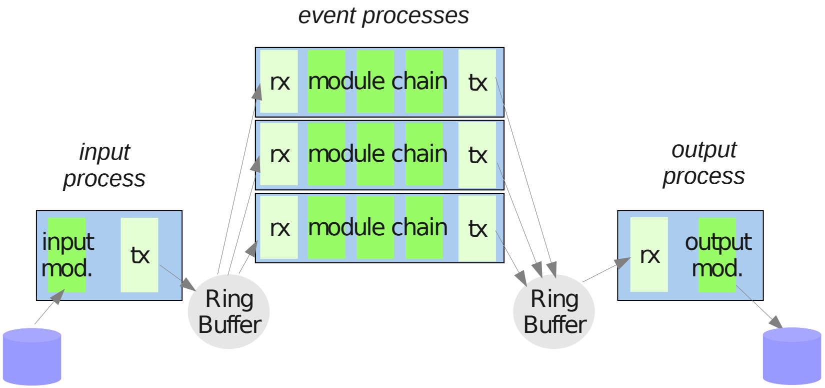
Overview of basf2

- Written in C++11
- Modular simulation / reconstruction / analysis toolchain
- Python steering files



Parallel Processing

- Event based (i.e. embarrassingly parallel)
- No threads (would require additional effort, libraries mostly not thread-safe)
- Processes are fork()ed, data exchanged through shared memory



Variables

Extendable list of variables shared by many analysis tools. Some examples:

- **M** invariant mass (determined from particle's 4-momentum vector)
- **dM** mass minus nominal mass
- **isSignal** 1.0 if Particle is correctly reconstructed, 0.0 otherwise
- **daughterAngle(i,j)** cosine of the angle between i-th and j-th daughters, in lab frame

Meta variables:

- **abs(variable)** Returns absolute value of the given variable.
- **daughter(n, variable)** Returns value of variable for the nth daughter.