TopoAna: A Generic Tool for the Event Type Analysis of Inclusive MC Samples

Xingyu Zhou[1] (Beihang Univ.)

Shuxian Du (Zhengzhou Univ.)

Gang Li (IHEP, CAS)

Chengping Shen (Fudan Univ.)

January 24, 2021

[1] zhouxy@buaa.edu.cn

Important information

- Latest version of the package at KEKCC:
  /home/belle2/zhouxy/workarea/tools/topoana/topoana-02-08-06[2]
- Documents in the share directory of each version of the package:
  README.pdf, quick-start_tutorial_v*.pdf, and user_guide_v*.pdf
- Repository at Stash:
  https://stash.desy.de/users/zhouxy/repos/topoana/browse
- Repository at GitHub:
  https://github.com/buaazhouxingyu/topoana
- Paper on ScienceDirect:
  Comput. Phys. Commun. 258 (2021) 107540
- Preprint on arXiv:
  arXiv:2001.04016

[2] The detailed version evolves over time.

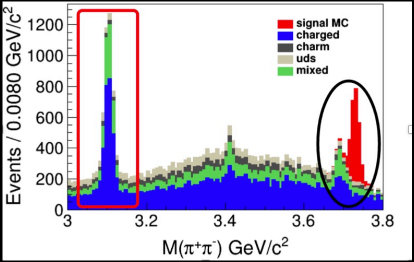
Outline

Introduction (I)

- One of the most important tasks in the data analysis of high energy physics experiments is to select signals, or in other words, to suppress backgrounds.
- As for the task, inclusive/generic MC samples are extremely useful, in that they provide basic, though not perfect, descriptions of the signals and/or backgrounds involved.
- To select signals with a higher efficiency and meanwhile suppress backgrounds to a lower level, a comprehensive understanding of the samples is required.
- In particular, a clear knowledge of the physics processes, or event types, involved in the samples is quite helpful.
- To be specific, the physics process information includes the types of processes and the number of processes in each type, involved both in the entire samples and in the individual events.

Introduction (II)

- With the physics process information, we can figure out the main backgrounds (especially the peaking ones).
- Then, we can optimize the selection criteria further by analyzing the differences between the main backgrounds and the signals.
- Even if it is difficult to further suppress these backgrounds, the knowledge of their types is beneficial to estimate the systematic uncertainties associated with them.



(Thank Junhao Yin for the plot!)

- Detailed analysis shows that the left peak mainly comes from $J/\psi \to e^+e^-/\mu^+\mu^-$, instead of $J/\psi \to \pi^+\pi^-$.

Introduction (III)

- Sometimes, we need to search for certain processes of interests.
- Mostly, signal and background events coexist in inclusive MC samples. It is useful to differentiate them in such cases.
- The identified signal events can be used to make up a signal sample in the absence of specialized signal samples, or they can be removed to avoid repetition in the presence of specialized signal samples.
- Occasionally, we have to pick out some decay branches in order to re-weight them according to new theoretical predictions or updated experimental measurements.

physics processs $\Longleftrightarrow$ event type $\Longleftrightarrow$ topology

- Since the raw topology truth information of inclusive MC samples is counter-intuitive, diverse, and overwhelming, it is difficult for analysts to check the topology information of the samples directly.
- To help them do the checks quickly and easily, a topology analysis program called TopoAna is developed with C++, ROOT, and LaTeX.

Basics of the program — Functionalities (I)

The program resolves counter-intuitive, diverse, and overwhelming input data

Number of MC generated particles	:	25
PDG codes of MC generated particles	:	433, −321, 223, 211, −413, 431, 111, 211, −211, 111, −411, 111, 321, 113, 22, 22, 22, 22, 321, −211, −211, 22, 22, 211, −211
Mother indices of MC generated particles	:	−1, −1, −1, −1, −1, 0, 0, 2, 2, 2, 4, 4, 5, 5, 6, 6, 9, 9, 10, 10, 10, 11, 11, 13, 13

into highly readable symbolic expressions of physics processes

0	$e^+e^- \to \pi^+\omega K^- D^{*-} D_s^{*+}$	-1	4	$D^- \to \pi^-\pi^- K^+$	2
1	$\omega \to \pi^0\pi^+\pi^-$	0	5	$D_s^+ \to \rho^0 K^+$	3
2	$D^{*-} \to \pi^0 D^-$	0	6	$\rho^0 \to \pi^+\pi^-$	5
3	$D_s^{*+} \to \pi^0 D_s^+$	0			

Basics of the program — Functionalities (II)

1. The program recognizes, categorizes, and counts physics processes in each event of the samples.
2. It tags the physics processes in the corresponding entry of the output root files.
   - Except for the tags, the input TTree object in the output root files is entirely the same as that in the input root files.
3. After processing the events, the program exports the obtained topology information at the sample level (topology maps) to the output plain text, tex source, and pdf files.
   - Although the files are in different formats, they have the same information.
   - The pdf file is the easiest to read. It is converted from the tex source file by the "pdflatex" command.
   - The plain text file is convenient to be checked with text processing commands as well as text editors.

Basics of the program — Package

```
[zhouxy@ccw02 topoana]$ ls
Configure*   Makefile    Setup*  examples/  share/  utilities/
LICENSE      README.md   bin/    include/   src/
```

- Files contained in the package:
  - README.md — readme file in markdown format
  - LICENSE — MIT license under which the program is released
  - Configure, Makefile, and Setup — scripts to install the software

- Directories contained in the package:
  - include — directory of the header file (topoana.h)
  - src — directory of the source files (topoana.C and *.cpp)
  - bin — directory of the binary files (topoana.exe and *.o)
  - share — directory of common data, style, card files and related documents (README.pdf, quick-start_tutorial_v*.pdf, and user_guide_v*.pdf)
  - utilities — directory of useful bash scripts
  - examples — directory of examples (in_the_quick-start_tutorial and in_the_user_guide)

Basics of the program — Installation

1. Configure the package path: ./Configure
   - Notably, you are recommended to manually set up the environment variable PATH according to the guidelines printed out by the command.
2. Compile and link the program: make
   - You succeed if you see the following line: "topoana.exe" installed successfully!
3. Setup the experiment name: ./Setup Belle_II
   - If you want to try the program with examples under the directory examples, please execute: ./Setup Example.

Basics of the program — Usage

1. Prepare the input data with the interfaces in basf2

2. Fill in the input card file
   - A template card file (template_topoana.card) can be found in the share directory.
   - For the concision of your own card file, it is recommended to just copy the setting items you need from the template card file and paste them to your own card file, just as we did for the examples in the quick-start tutorial and the user guide.
   - Since there are plenty of setting items in the template card file, it is NOT recommended to create your own card file simply by copying and revising the whole template card file.

3. Execute the program
   - Cmmand line: topoana.exe cardFileName
   - The default card file name is topoana.card
   - Execute topoana.exe −−help to see other optional arguments supported in the command line

Basics of the program — Interfaces in basf2

TopoAna is a tool independent of basf2, and it needs an interface in basf2 to prepare the input data for it.

- Before release-04, the official NtupleTool MCGenKinematics can be used.
- After release-04, the official variable list MCGenTopo developed by me can be used.
  1. Append the statement at the beginning part of python steering files

     from variables.MCGenTopo import mc_gen_topo
  2. Use the function mc_gen_topo(n) as a list of variables in the steering function variablesToNtuple as follow:

     variablesToNtuple(particleList, yourOwnVariableList + mc_gen_topo(n), treeName, fileName, path)

     Here, n is the number of MCGenPDG_i/MCGenMothIndex_i variables (see next slide), and its default value is 200.

Basics of the program
A typical entry containing the input information

```
nMCGen            = 94
MCGenPDG_0        = 300553
MCGenMothIndex_0 = -999
MCGenPDG_1        = -511
MCGenMothIndex_1 = 0
MCGenPDG_2        = 511
MCGenMothIndex_2 = 0
MCGenPDG_3        = -213
MCGenMothIndex_3 = 1
MCGenPDG_4        = 411
MCGenMothIndex_4 = 1
MCGenPDG_5        = -311
MCGenMothIndex_5 = 1
MCGenPDG_6        = 311
MCGenMothIndex_6 = 2
```

Only part of the varaibles are shown here!

The number of MCGenPDG_i/MCGenMothIndex_i is n set in mc_gen_topo(n).

- nMCGen — number of MC generated particles
- MCGenPDG_i — PDG code of the i$^{\mathrm{th}}$ MC generated particle
- MCGenMothInex_i — mother index of the i$^{\mathrm{th}}$ MC generated particle

Basics of the program — Essential functionality

Items set in the input card file (1)

```
# The following five items set the input of the program.

% Names of input root files
{
  ../input/jpsi1.root
  ../input/jpsi2.root
}

% TTree name
{
  evt
}

% TBranch name of the number of particles (Default: nMCGen)
{
  Nmcps
}

% TBranch name of the PDG codes of particles (Default: MCGenPDG)
{
  Pid
}
```

Basics of the program — Essential functionality
Items set in the input card file (2)

```
% TBranch name of the mother indices of particles (Default: MCGenMothIndex)
{
    Midx
}

# The following item sets the basic functionality of the program.

% Component analysis — decay trees
{
    Y
}

# The following item sets the output of the program.

% Common name of output files (Default: Name of the card file)
{
    jpsi_ta
}
```

Basics of the program — Essential functionality

Topology Map listed in the output pdf file

Table: Top six decay trees and their respective final states.

rowNo	decay tree	decay final state	iDcyTr	nEtr	nCEtr
1	$J/\psi \to \mu^+\mu^-$	$\mu^+\mu^-$	6	5269	5269
2	$J/\psi \to e^+e^-$	e^+e^-	4	4513	9782
3	$J/\psi \to \pi^0\pi^+\pi^+\pi^-\pi^-$	$\pi^0\pi^+\pi^+\pi^-\pi^-$	0	2850	12632
4	$J/\psi \to \pi^0\pi^+\pi^+\pi^+\pi^-\pi^-\pi^-$	$\pi^0\pi^+\pi^+\pi^+\pi^-\pi^-\pi^-$	2	1895	14527
5	$J/\psi \to \pi^0\pi^+\pi^-K^+K^-$	$\pi^0\pi^+\pi^-K^+K^-$	20	1698	16225
6	$J/\psi \to \rho^+\rho^-\omega, \rho^+ \to \pi^0\pi^+,$ $\rho^- \to \pi^0\pi^-, \omega \to \pi^0\pi^+\pi^-$	$\pi^0\pi^0\pi^0\pi^+\pi^+\pi^-\pi^-$	19	1453	17678

Only The top six decay trees and their respective final states are shown here!

- rowNo — row number
- iDcyTr — index of decay tree
- nEtr — number of entries
- nCEtr — number of cumulative entries

Basics of the program — Essential functionality
Topology Tag inserted in the output root file

- Here, iDcyTr is the topology tag for decay trees.
- Thus, it is also saved in the TTree object of the output root file, together with other quantities for physics analysis.
- Therefore, it can be used to pick out the entries of specific decay trees in order to examine the distributions of the other quantities over the decay trees.

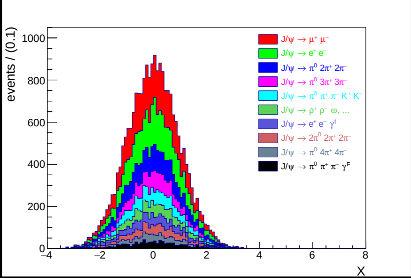


Figure: Distribution of X accumulated over the top ten decay trees.

Component analysis

Component analysis is the primary functionality of the program.

1. Over decay trees
2. Over decay initial-final states
3. With specified particles, to check their
   - decay branches
   - cascade decay branches
   - decay final states
   - production branches
   - mothers
4. With specified inclusive decay branches, to examine their exclusive components
5. With specified intermediate-resonance-allowed (IRA) decay branches, to investigate their inner structures

In this tutorial, we only take the component analysis over the decay branches of specified particles as an example.

Component analysis — decay branches of particles
Items set in the input card file

```
% Names of input root files
{
    ../input/mixed1.root
    ../input/mixed2.root
}

% TTree name
{
    evt
}

% Component analysis — decay branches of particles
{
    D*+     Dsp     5
    J/psi     Jpsi     5
}
```

Parameters in the functionality item

One particle information in each row.

- D*+ and J/psi — Particles to be investigated.
- Dsp and Jpsi — Aliases for D*+ and J/psi, respectively. They will be used in the TBranch names of topology tags.
- 5 and 5 — Maximum numbers of decay branches to be printed.

Component analysis — decay branches of particles
Topology maps listed in the output pdf file (1)

Table: Decay branches of D^{*+}.

rowNo	decay branch of D^{*+}	iDcyBrP	nCase	nCCase
1	$D^{*+} \to \pi^+ D^0$	0	31180	31180
2	$D^{*+} \to \pi^0 D^+$	1	13978	45158
3	$D^{*+} \to D^+ \gamma$	2	700	45858
4	$D^{*+} \to \pi^+ D^0 \gamma^F$	3	28	45886

- rowNo — row number
- iDcyBrP — index of decay branch of the specified particle
- nCase — number of cases
- nCCase — number of cumulative cases

Component analysis — decay branches of particles
Topology maps listed in the output pdf file (2)

Table: Decay branches of J/ψ.

rowNo	decay branch of J/ψ	iDcyBrP	nCase	nCCase
1	$J/\psi \rightarrow \mu^{+}\mu^{-}$	2	128	128
2	$J/\psi \rightarrow \pi^{0}\pi^{+}\pi^{+}\pi^{-}\pi^{-}$	9	101	229
3	$J/\psi \rightarrow e^{+}e^{-}\gamma^{F}$	1	65	294
4	$J/\psi \rightarrow e^{+}e^{-}$	20	56	350
5	$J/\psi \rightarrow \pi^{0}\pi^{+}\pi^{+}\pi^{+}\pi^{-}\pi^{-}\pi^{-}$	51	51	401
rest	$J/\psi \rightarrow \mathrm{others\ (639\ in\ total)}$	—	2253	2654

- rowNo — row number
- iDcyBrP — index of decay branch of the specified particle
- nCase — number of cases
- nCCase — number of cumulative cases

Component analysis — decay branches of particles
Topology tags inserted in the output root file

Table: Topology tags inserted in the output ROOT file.

Particle	Topology tag	Interpretation
D^{*+}	nPDcyBr_Dsp	number of D^{*+} (or its decay branches)
	iDcyBrP_Dsp_j	index of decay branch of the j^{th} D^{*+}
J/ψ	nPDcyBr_Jpsi	number of J/ψ (or its decay branches)
	iDcyBrP_Jpsi_j	index of decay branch of the j^{th} J/ψ

Signal identification

Signal identification is the other functionality of the program. Though it is a relatively simple functionality, it can help us identify the physics processes of interests we desire directly, quickly, and easily.

- decay trees
- decay initial-final states
- particles
- (regular) decay branches
- cascade decay branches
- inclusive decay branches
- inclusive cascade decay branches
- IRA decay branches

In this tutorial, we only take the identification of the (regular) decay branches as an example.

Signal identification — decay branches
Items set in the input card file

```
% Names of input root files
{
    ../input/mixed1.root
    ../input/mixed2.root
}

% TTree name
{
    evt
}

% Signal identification — decay branches
{
    anti-B0 --> mu- anti-nu_mu D*+   &   B2munuDsp
    B0 --> K_S0 J/psi   &   B2KsJpsi
}
```

Parameters in the functionality item

One decay branch information in each row.

- anti-B0 −−> mu− anti-nu_mu D*+ and B0 −−> K_S0 J/psi — Decay branches to be identified.
- B2munuDsp and B2KsJpsi — Aliases for the two decay branches, respectively. They will be used in the TBranch names of topology tags.
- & — Separator between the decay branches and their aliases.

Signal identification — decay branches
Topology map listed in the output pdf file

Table: Signal decay branches.

rowNo	signal decay branch	iSigDcyBr	nCase	nCCase
1	$\bar{B}^0 \to \mu^- \bar{\nu}_\mu D^{*+}$	0	4154	4154
2	$B^0 \to K_S^0 J/\psi$	1	45	4199

- rowNo — row number
- iSigDcyBr — index of signal decay branch
- nCase — number of cases
- nCCase — number of cumulative cases

Signal identification — decay branches
Topology tags inserted in the output root file

Table: Topology tags inserted in the output ROOT file.

Decay branch	Topology tag	Interpretation
$\bar{B}^0 \to \mu^- \bar{\nu}_\mu D^{*+}$	nSigDcyBr_B2munuDsp	number of $\bar{B}^0 \to \mu^- \bar{\nu}_\mu D^{*+}$
$B^0 \to K_S^0 J/\psi$	nSigDcyBr_B2KsJpsi	number of $B^0 \to K_S^0 J/\psi$

Common settings

About two dozen optional items are designed and implemented to control the execution of the program in order to meet practical needs.

1. Settings on the input of the program
   - input entries (3 items)
   - input branches (5 items)
   - ISR and FSR photons (4 items and 2 sorts of parameters)
2. Settings on the functionalities of the program
   - candidate based analysis (2 items)
   - charge conjugation (1 item)
   - Reconstruction restrictions on truth particles (1 sort of parameters)
   - settings only on signal identification (2 items)
3. Settings on the output of the program
   - output pdf files (4 items)
   - output root files (7 items)

In this tutorial, we only take the setting on charge conjugation as an example.

Common settings — charge conjugation
Items set in the input card file

```
% Names of input root files
{
   ../input/mixed1.root
   ../input/mixed2.root
}

% TTree name
{
   evt
}

% Component analysis — decay branches of particles
{
   D*+     Dsp     5
   J/psi     Jpsi     5
}

% Process charge conjugate objects together (Two options: Y and N. Default: N)
{
   Y
}
```

Parameters in the functionality item

One particle information in each row.

- D*+ and J/psi — Particles to be investigated.
- Dsp and Jpsi — Aliases for D*+ and J/psi, respectively. They will be used in the TBranch names of topology tags.
- 5 and 5 — Maximum numbers of decay branches to be printed.

Common settings — charge conjugation
Topology maps listed in the output pdf file (1)

Table: Decay branches of D^{*+} (with the charge conjugation setting).

rowNo	decay branch of D^{*+}	iDcyBrP	nCase	nCcCase	nAllCase	nCCase
1	$D^{*+} \rightarrow \pi^+ D^0$	0	31180	31291	62471	62471
2	$D^{*+} \rightarrow \pi^0 D^+$	1	13978	14166	28144	90615
3	$D^{*+} \rightarrow D^+ \gamma$	2	700	721	1421	92036
4	$D^{*+} \rightarrow \pi^+ D^0 \gamma^F$	3	28	36	64	92100
5	$D^{*+} \rightarrow \pi^0 D^+ \gamma$	4	0	1	1	92101

- rowNo — row number
- iDcyBrP — index of decay branch of the specified particle
- nCase — number of cases of the specified particle
- nCcCase — number of cases of the charge conjugate particle
- nAllCase — sum of nCases and nCcCases
- nCCase — number of cumulative cases

Common settings — charge conjugation
Topology maps listed in the output pdf file (2)

Table: Decay branches of J/ψ (with the charge conjugation setting).

rowNo	decay branch of J/ψ	iDcyBrP	nCase	nCcCase	nAllCase	nCCase
1	$J/\psi \rightarrow \mu^{+}\mu^{-}$	2	128	–	128	128
2	$J/\psi \rightarrow \pi^{0}\pi^{+}\pi^{+}\pi^{-}\pi^{-}$	9	101	–	101	229
3	$J/\psi \rightarrow e^{+}e^{-}\gamma^{F}$	1	65	–	65	294
4	$J/\psi \rightarrow \pi^{0}\pi^{-}\rho^{+}$	11	28	34	62	356
5	$J/\psi \rightarrow e^{+}e^{-}$	20	56	–	56	412
rest	$J/\psi \rightarrow \mathrm{others}\ (550\ \mathrm{in\ total})$	–	–	–	2242	2654

- rowNo — row number
- iDcyBrP — index of decay branch of the specified particle
- nCase — number of cases of the specified particle
- nCcCase — number of cases of the charge conjugate particle
- nAllCase — sum of nCases and nCcCases
- nCCase — number of cumulative cases

Common settings — charge conjugation
Topology tags inserted in the output root file

Table: Topology tags inserted in the output ROOT file.

Particle	Topology tag	Interpretation
D^{*+}	nPDcyBr_Dsp	number of D^{*+} (or its decay branches)
	iDcyBrP_Dsp_j	index of decay branch of the j^{th} D^{*+}
	iCcPDcyBr_Dsp	charge conjugate index of D^{*+}
	nCcPDcyBr_Dsp	number of D^{*-} (or its decay branches)
	iDcyBrCcP_Dsp_j	index of decay branch of the j^{th} D^{*-}
	nAllPDcyBr_Dsp	number of D^{*+} and D^{*-} (or their decay branches)
J/ψ	nPDcyBr_Jpsi	number of J/ψ (or its decay branches)
	iDcyBrP_Jpsi_j	index of decay branch of the j^{th} J/ψ
	iCcPDcyBr_Jpsi	charge conjugate index of J/ψ
	iCcDcyBrP_Jpsi_j	charge conjugate index of decay branch of the j^{th} J/ψ

Summary

- A generic tool for the event type analysis of inclusive MC samples, TopoAna, has been developed.
- It has rich functionalities and aims to solve all kinds of event type analysis tasks. Meanwhile, it is efficient and easy to use.
- These features make it powerful for analysts to investigate the signals and backgrounds involved in their works.
- The interface of basf2 to it, MCGenTopo, has been developed.
- A brief quick-start tutorial and a detailed user guide have been written up and are put under the share directory of the package.
- The paper on it has been published:

Comput. Phys. Commun. 258 (2021) 107540[3]

[3] If the tool really helps your researches, we would appreciate it very much if you could cite the paper in your publications.

Welcome to TopoAna

Best way to learn how to use TopoAna:

1. try the examples under examples/in_the_quick-start_tutorial, with the quick-start tutorial under share as a reference.
2. try the examples under examples/in_the_user_guide, with the user guide under share as a reference.

Please let us know:

- if you encounter any problems with it,
- if you have any questions about it,
- if you find any bugs in it,
- if you have any suggestions on improving it,
- if you want to extend its functionalities.

We hope it can be helpful to your studies.

Acknowledgements

- This work was supported by the National Natural Science Foundation of China [grant numbers 11575017, 11661141008, 11761141009, 11875262, 11975076] and the CAS Center for Excellence in Particle Physics (CCEPP).
- Here, we would like to thank all of the people who have helped us in the development of the program.
  - We first thank Prof. Changzheng Yuan, Bo Xin, and Haixuan Chen for their help at the early stage of developing the program.
  - We are particularly grateful to Prof. Xingtao Huang for his comments on the principles and styles of the program, to Remco de Boer for his suggestions on the tex output and the use of GitHub, and to Xi Chen for his discussions on the core algorithms.
  - We are especially indebted to Prof. Xiqing Hao, Longke Li, Xiaoping Qin, Ilya Komarov, Yubo Li, Guanda Gong, Suxian Li, Junhao Yin, Prof. Xiaolong Wang, Yeqi Chen, Hannah Wakeling, Hongrong Qi, Hui Li, Ning Cao, Sanjeeda Bharati Das, Kazuki Kojima, Tingting Han, Fang Yan, Lin Wang, and Meiru An for their advice in extending and perfecting the program.
  - Also, we thank Xi'an Xiong, Runqiu Ma, Wencheng Yan, Sen Jia, Lu Cao, Dong Liu, Hongpeng Wang, Jiawei Zhang, Jiajun Liu, Maoqiang Jing, Yi Zhang, Wei Shan, and Yadi Wang for their efforts in helping us test the program.

Thank you all for your help!