# TopoAna: A generic tool for the event type analysis of inclusive Monte-Carlo samples in high energy physics experiments

**Abstract:** Inclusive Monte-Carlo samples are indispensable for signal selection and background suppression in many high energy physics experiments. A clear knowledge of the physics processes involved in the samples, including the types of processes and the number of processes in each type, is a great help to investigating signals and backgrounds. To help analysts obtain the physics process information from the truth information of the samples, we develop a physics process analysis program, TopoAna, with C++, ROOT, and LaTeX. The program implements the functionalities of component analysis and signal identification with many kinds of fine, customizable classification and matching algorithms. It tags physics processes in individual events accurately in the output root files, and exports the physics process information at the sample level clearly to the output plain text, tex source, and pdf files. Independent of specific software frameworks, the program is applicable to many experiments. At present, it has come into use in three e+e- colliding experiments: the BESIII, Belle, and Belle II experiments. The use of the program in other similar experiments is also prospective. (**Please contact me (zhouxy@buaa.edu.cn) if you want to employ the program for other experiments. Now, you can try the program with examples under the directory** `examples` )

A detailed description of the program is in the document: `share/user_guide_v*.pdf` (**Please note that all the examples in the user guide can be found in the sub-directory** `examples/in_the_user_guide` )

An essential description of the program is in the document: `share/paper_draft_v*.pdf`
(**Please note that all the examples in the paper draft can be found in the sub-directory** `examples/in_the_paper` )

**Note:** The paper on the tool has been published by `Computer Physics Communications` . You can find this paper and the preprint corresponding to it in the links Comput. Phys. Commun. 258 (2021) 107540 and arXiv:2001.04016, respectively. If the tool really helps your researches, we would appreciate it very much if you could cite the paper in your publications.

**It is a good practice to learn how to use the program via the examples with the user guide and paper draft as references.**

Here, we just briefly introduce how to install and use the program.

1. How to install the program

   1.1. Set up the package path with the following command: `./Configure`

   - **Notably, you are recommended to manually set up the environment variable `PATH` according to the guidelines printed out by the command.**
   - Executing the command and setting the `PATH` once is sufficient, unless the package is moved.

   1.2. Compile the program with the following command: `make`

   - Executing the command once is sufficient, unless the C++ header, source, or script files are updated or revised, or the package is moved.

   1.3. Set up the experiment name with the following command: `./Setup ExpNm`

   - Here, the formal argument `ExpNm` is short for experiment name, and the actual values supported at present are `BESIII`, `Belle`, `Belle_II`, and `Example`.
     - If you want to employ the program for the BESIII experiment, please execute: `./Setup BESIII`.
     - If you want to employ the program for the Belle experiment, please execute: `./Setup Belle`.
     - If you want to employ the program for the Belle II experiment, please execute: `./Setup Belle_II`.
     - If you want to try the program with examples under the directory `examples`, please execute: `./Setup Example`.
   - Executing the command once is sufficient, unless you want to alter the targeted experiment.

2. How to use the program

   1.1. Prepare the input data

   - The input data of the program, namely the raw topology truth information of the inclusive MC samples, can be obtained with the interface to the program in the software system of your experiment. For the BESIII, Belle, and Belle II experiments, such interfaces have already been developed.

   1.2. Fill in the card file

   - You can refer to the template topoana card file: `share/template_topoana.card`.
   - **For the concision of your own card file, it is recommended just to 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 user guide and paper draft.**

- **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.**

1.3. Execute the command line: `topoana.exe cardFileName`

- The argument cardFileName is optional, and its default value is `topoana.card`.