

Table of Contents

: Analytical morphing class in.....	1
Introduction.....	1
Example.....	1
Supported processes.....	1
Format of the input file.....	1
Number of input files.....	2
Macro for building the param_card.....	2

: Analytical morphing class in

- Introduction

-

Introduction

RooEFTMorphFunc is a standalone RooFit class implementing the analytical morphing method. Analytical morphing is a method to construct a continuous signal model in the coupling parameter space. Basic assumption is that shape and cross section of a physical distribution is proportional to its squared matrix element. The signal model is constructed by a weighted sum over N input distributions of an observable O.

todo weighted sum

Calculation of the weights is based on Matrix Elements.

The number of input files depends on the number of couplings in production and decay vertices, and also whether the decay and production vertices describe the same process or not.

number of input samples

The implementation of RooEFTMomentMorph is based on the Higgs Characterisation Model (todo link) implemented in MadGraph5_MC@NLO. It is possible to add another theoretical framework, if such a use case arises, please contact the developers.

Example

Supported processes

1. Production: ggF and VBF
2. Decays: $H \rightarrow ZZ$ and $H \rightarrow WW \rightarrow l\nu l\nu$
3. Theoretical framework: EFT implemented in Higgs Characterisation Model

Supported processes can be easily expanded to other decays, theoretical framework. If you have a use-case please contact the developers <hello here we are>.

Format of the input file

The format of the input file is at the moment restricted to the following layout:

The param_card histogram

List of folders

The rootfile should contain a list of folders with arbitrary names. Also the number of folders is not restricted. The user can specify in the constructor which folders represent the morphing input files <see section XY>.

Each container has to contain a list of TH1, from which exactly one is called 'param_card'. This histogram contains the parameter configuration for each input file and has also a specified format <see section XY>. The other histograms are the distribution of the observables. The user can add several observables. In the morphing class constructor it has to be specified which observable is used in the morphing. The layout of the subfolders has to be the same

Number of input files

The number of input files needed for successful morphing depends on

- 1) the process
- 2) the couplings included in the process

For ggf

For VBF

For example

In some cases, the morphing class is returning warnings when the input is not reasonable. But at the moment it does not catch all possible problems and the user should make sure that her/his input is reasonable to ensure successful morphing.

Macro for building the param_card

This is an example, how the param_card histogram can be created by using the param card configuration of the respective MC production as input. *

This topic: Sandbox > KatharinaMariaEckerSandbox

Topic revision: r2 - 2015-06-29 - KatharinaMariaEcker



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback