

# 2D Alphabet Software Documentation

Lucas Corcodilos

March 8, 2018

## Introduction to the method

## Prerequisites

## Installation

## JSON Configuration

While I've tried to keep the format of the JSON configuration file obvious, it's not fair to assume that everyone else will feel the same way. I've included "HELP" keys in dictionaries where I felt I could easily explain the formatting. This section is a necessary supplement to those. Please note that I may use phrases like, "The key 'HELP' is defined inside 'PROCESS'." I'm aware this is not correct since "PROCESS" is a key for a dictionary which has a key "HELP" and so "HELP" can't be "inside" "PROCESS" but it simplifies the explanation so I'm going to roll with it.

The goal of the JSON is to have an easily configurable input that allows the 2D Alphabet software to read and organize analysis files and histograms to its liking while also giving the user the ability to easily configure values like bin sizes and ranges. This means that while the user is encouraged to add what they need, there are some keys and values that must stay the same. **These static strings are always in capital letters to make them easy to distinguish.** I'll now describe what each of the five static sections is designed for.

## PROCESS

In this section, the user can define as many processes as they need to account for. This includes data and MC. This section is NOT used to define the background to be estimated via the transfer function. That background is naturally defined by the difference between data and the other backgrounds defined here.

Each key in PROCESS is the name of each process of interest. Please name

your data as “data\_obs.” This is a Combine convention that I’d like to maintain. Each process name is a key for a sub-dictionary that specifies

- **FILE:** the path to the file containing the nominal pass and fail histograms for the process (string);
- **HISTPASS:** the name of the histogram in the above file with the passing distribution (string);
- **HISTFAIL:** the name of the histogram in the above file with the failing distribution (string);
- **:** a list of strings that correspond to the names of systematics in **SYSTEMATIC** (note **SYSTEMATIC** not **SYSTEMATICS**) that are applicable to this process (list of strings);
- **CODE:** a way to classify the treatment of the process: 0 (signal), 1 (data), 2 (unchanged MC), 3 (MC to be renormalized) (int).

## SYSTEMATIC

Because it bears repeating, please note the difference between this section, **SYSTEMATIC**, and the list of **SYSTEMATICS** defined inside the **PROCESS** dictionary. The **SYSTEMATIC** dictionary is a place to define as many systematics as a user may need. Similar to the processes, each key in **SYSTEMATIC** is the name of the systematic in the analysis and each is classified by a code that determines how the systematic will be treated. However, the dictionary the user defines for a given systematic is different depending on what type it is. The self-explanatory types are:

- Symmetric, log-normal
  - **CODE:** 0 (int)
  - **VAL:** uncertainty (float)
- Asymmetric, log-normal
  - **CODE:** 1 (int)
  - **VALUP:**  $+1\sigma$  uncertainty (float)
  - **VALDOWN:**  $-1\sigma$  uncertainty (float)

Less obvious are codes 2 and 3 which are for shape based uncertainties (and thus have corresponding histograms) and are either in the same file as the process’s nominal histogram (code 2) or in a separate file (code 3).

- Shape based uncertainty, in same file as nominal histogram

- **CODE:** 2 (int)
- **HISTUP:** the name of the histogram (in the same file as the nominal histogram) for  $+1\sigma$  uncertainty (string)
- **HSITDOWN:** the name of the histogram (in the same file as the nominal histogram) for  $-1\sigma$  uncertainty (string)
- Shape based uncertainty, in different file as nominal histogram
  - **CODE:** 3 (int)
  - **FILEUP:** /path/to/fileup.root which contains the  $+1\sigma$  uncertainty histogram (string)
  - **FILEDOWN:** /path/to/filedown.root which contains the  $-1\sigma$  uncertainty histogram (string)
  - **HIST:** the name of the histogram in each of these files (string)

This scheme obviously relies on the user’s ability to be somewhat organized. If for example, you have your  $+1\sigma$  shape based uncertainty in file\_up.root with histogram name “SystUp” and  $-1\sigma$  shape based uncertainty in file\_down.root with histogram name “SystDown” then there’s no applicable code for you to use.

I designed the scheme this way because 2 and 3 are the only applicable classifications to my analysis approach and so I ask the user to be at least as organized as I am so that this scheme works for them.

## BINNING

One set of important user defined values is the 2D binning of the space being analyzed. This dictionary is the opportunity to define the axes values. The values are split into x and y axis definitions where the x-axis describes the variable whose signal region is blinded.

- X
  - **NAME:** name of your variable on the x-axis (string)
  - **LOW:** lower bound of x-axis (int)
  - **HIGH:** upper bound of x-axis (int)
  - **NBINS:** number of x bins from LOW to HIGH (int)
  - **SIGSTART:** lower bound of signal region of x-axis (int)
  - **SIGEND:** upper bound of signal region of x-axis (int)
- Y
  - **NAME:** name of your variable on the y-axis (string)

- **LOW**: lower bound of y-axis (int)
- **HIGH**: upper bound of y-axis (int)
- **NBINS**: number of x bins from **LOW** to **HIGH** (int)

## **FIT**

The other set of important user defined values is the fit parameters for the transfer function from the fail sideband to the passing (or pass-fail ratio). The 2D fit of the transfer function assumes a polynomial in both directions. The order of the polynomials are up to the user (technically they are capped at order 10) as long as each parameter is defined. For example, if the highest order parameter defined is 2 in x and 1 in y, the user needs to define six parameters here.

The naming for the parameters must follow the form **X#Y&** where **#** and **&** are the polynomial orders

- **X0Y0**
  - **NOMINAL**: nominal value (float)
  - **LOW**: lower bound (float)
  - **HIGH**: upper bound (float)
- **X1Y0**
  - **NOMINAL**: nominal value (float)
  - **LOW**: lower bound (float)
  - **HIGH**: upper bound (float)
- **X0Y1**
  - **NOMINAL**: nominal value (float)
  - **LOW**: lower bound (float)
  - **HIGH**: upper bound (float)
- **X1Y1**
  - **NOMINAL**: nominal value (float)
  - **LOW**: lower bound (float)
  - **HIGH**: upper bound (float)
- ...

## How To Run

Command line options

Plotting