nnpdf++ data layout

Maintainer: Nathan Hartland
nathan.hartland@physics.ox.ac.uk

October 19, 2015

Contents

1 Revision history

Date	Version	Major/Minor	Author	Comments
9/9/15	1.0.0	-	nh	First version.
16/9/15	1.1.0	Major	vb/sc/nh	Implementing many new theory parameters. Adding header to `CFACTOR` files.
19/10/15	1.1.1	Minor	nh	Updating paths after move to git repository.

2 Introduction

In the `nnpdf++` project, data files used by the code may be grouped into two categories, theory and experiment. Experimental data and the information pertaining to the treatment of systematic errors are held in `CommonData` and `SYSTYPE` files. `FK` tables, `COMPOUND` and `CFACTOR` files store the precomputed information for use when calculating theoretical predictions corresponding to information held in the equivalent `CommonData` file. In this document the file formats and naming conventions for these files will be detailed, along with the directory structure employed by the `nnpdf++` code.

For NNPDF3.1 and later fits, a considerably larger number of theory options will be explored than in previous determinations. In NNPDF3.0 the main theory variations used were perturbative order, value of the strong coupling and the number of active flavours in the VFNS. For NNPDF3.1 and later, variations in additional parameters must be accommodated, such as treatments of the heavy quark mass (pole vs $\overline{\text{MS}}$), scale variations, intrinsic charm, resummation effects etc. The book-keeping used to enable efficient variations of the theoretical treatment used in fits post-3.0 will therefore also be outlined here.

This document will begin by detailing the specifications for the file formats used by the code, first with the experimental data file formats and layouts in Section 3 and secondly with the file formats used for theoretical predictions in Section 4. Finally the organisation of these files within the `nnpdf++` structure will be described in Section 5.

2.1 Important definitions

In order to clarify the later description, here are a few important terminological points to note.

Dataset vs *Experiment*

When referring to a collection of data points two words are used in the `nnpdf++` code which have specific meanings. *Dataset* refers to the result of a specific measurement, typically associated with a single experimental paper and corresponds to the `DataSet` class in the `nnpdf++` code. *Experiment* refers to a collection of *Datasets* which are associated by experimental cross-correlations. For example, the ATLAS 2010 $R = 0.4$ inclusive jet measurement [1] and the ATLAS high-mass Drell-Yan measurement of [2] are both examples of *Datasets* as used in the NNPDF3.0 analysis. Both of these datasets are grouped into the ATLAS *Experiment* as they have systematic uncertainties that are cross-correlated with each other. In this document, when using these terms in this sense, they will be italicised for clarity.

***Dataset* and *Experiment* names**

When referred to, the *Dataset* and *Experiment* names refer to the short identifying string used in the code for each *Dataset* and *Experiment*. For example, the *Dataset* name for the aforementioned ATLAS 2010 inclusive jet measurement with $R = 0.4$ is ATLASR04JETS36PB.

3 Experimental data files

Data made available by experimental collaborations comes in a variety of formats. For use in a fitting code, this data must be converted into a common format that contains all the required information for use in PDF fitting. Existing formats commonly used by the community, such as in HepData, are generally unsuitable. Principally as they often do not fully describe the breakdown of systematic uncertainties. Therefore over several years an NNPDF standard data format has been iteratively developed, now denoted `CommonData`. In addition to the `CommonData` files themselves, in the `nnpdf++` project the user has the ability to vary the treatment of individual systematic errors by use of parameter files denoted `SYSTYPE` files. In this section we shall detail the specifications of these two files.

In principle, the file specification and classes described in this section are independent of the `nnpdf++` project and may be generated by whatever means the user sees fit. In practice, the `CommonData` and `SYSTYPE` files are generated by the `buildmaster` project of `nnpdf++` from the raw experimental data files.

3.1 Process types and kinematics

Before going into the file formats, we shall summarise the identifying features used for data in the `nnpdf++` code.

Each datapoint has an associated *process type* string. This can be specified by the user, but **must** begin with one of a selection of identifying base process types. Additionally for each datapoint three kinematic values are given, the *process type* being primarily to identify the nature of these values. Typically the first kinematic variable is the principal differential quantity used in the measurement. The second kinematic variable is typically the scale of the process. The third is generally the centre-of-mass energy of the process, or inelasticity in the case of DIS. The allowed basic process types, and their corresponding three kinematic variables are outlined below.

- **DIS** - Deep inelastic scattering measurements: (x, Q^2, y)
- **DYP** - Fixed-target Drell-Yan measurements: $(y, M^2, \sqrt{s})$
- **EWK** - Collider electroweak production: $(\eta, M^2, \sqrt{s})$
- **JET** - Jet production: $(\eta, p_T, \sqrt{s})$

- **HQP** - Heavy quark production: $(y, p_T, \sqrt{s})$
- **INC** - A total inclusive cross-section: $(0, 0, 0)$

As examples of *process type* strings, consider **EWK_WASYM** for a collider W boson asymmetry measurement, and **DIS_F2P** for the F_2^p structure function in DIS. The user is free to choose something identifying for the second segment of the process type, the important feature being the first three characters.

Notes for the future

In the future it would be nice to have a more flexible treatment of the kinematic variables, both in their number and labelling.

3.2 `CommonData` file format

Each experimental *Dataset* has its own `CommonData` file. `CommonData` files contain the bulk of the experimental information used in the `nnpdf++` project, with the only other experimental data files controlling the treatment and correlation of systematic errors. Each `CommonData` file is a plaintext file with the following layout.

The first line begins with the *Dataset* name, the number of systematic errors, and the number of datapoints in the set, whitespace separated. For example for the ATLAS 2010 jet measurement the first line of the file reads:

ATLASR04JETS36PB 91 90

Which demonstrates that the set *name* is 'ATLASR04JETS36PB', that there are 91 sources of systematic uncertainty, 90 datapoints, one associated `FK` table, and that the `FK` table corresponds to a proton initial state. As another example, consider the NMCPD *Dataset*:

NMCPD 5 211

Here there are 5 sources of systematic uncertainty and 211 datapoints.

Following this, each line specifies the details of a single datapoint. The first value being the datapoint index $1 < i_{\rm dat} \leq N_{\rm dat}$, followed by the *process type* string as outlined above, and the three kinematic variables in order. These are followed by the value of the experimental datapoint itself, and the value of the statistical uncertainty associated with it (absolute value). Finally the systematic uncertainties are specified. The layout per datapoint is therefore

$i_{\rm dat}$ *ProcessType* $\rm kin_1$ $\rm kin_2$ $\rm kin_3$ data_value stat_error [.. systematics ..]

For example, in the case of a DIS datapoint from the BCDMSD *Dataset*:

1 DIS_F2D 7.0e-02 8.75e+00 5.666e-01 3.6575e-01 6.43e-03 [.. systematics ..]

In these lines the systematic uncertainties are laid out as so. For each uncertainty, additive and multiplicative versions are given. The additive uncertainty is given by absolute value, and the multiplicative as a percentage of the data value. The systematics string is formed by the sequence of N_{sys} pairs of systematic uncertainties:

$$[.. \text{ systematics } ..] = \sigma_0^{\text{add}} \quad \sigma_0^{\text{mul}} \quad \sigma_1^{\text{add}} \quad \sigma_1^{\text{mul}} \quad .... \quad \sigma_n^{\text{add}} \quad \sigma_n^{\text{mul}}$$

where σ_i^{add} and σ_i^{mul} are the additive and multiplicative versions respectively of the systematic uncertainty arising from the ith source. While it may seem at first that the multiplicative error is spurious given the presence of the additive error and data central value, this may not be the case. For example in a closure test scenario, the data central values may have been replaced in the `CommonData` file by theoretical predictions. Therefore if you wish to use a covariance matrix generated with the original multiplicative uncertainties via the t_0 method, you must also store the original multiplicative (percentage) error. For flexibility and ease of I/O this is therefore done in the `CommonData` file itself

For a *Dataset* with N_{dat} datapoints and N_{sys} sources of systematic uncertainty, the total `CommonData` file should therefore be $N_{\text{dat}} + 1$ lines long. Its first line contains the set parameters, and every subsequent line should consist of the description of a single datapoint. Each datapoint line should therefore contain $7 + 2N_{\text{sys}}$ columns.

3.3 `SYSTYPE` file format

The explicit presentation of the systematic uncertainties in the `CommonData` file allows for a great deal of flexibility in the treatment of these errors. Specifically, whether they should be treated as additive or multiplicative uncertainties, and how they are correlated, both within the *Dataset* and within a larger *Experiment*. A specification for how the systematic uncertainties should be treated is provided by a `SYSTYPE` file. As there is not always an unambiguous method for the treatment of these uncertainties, these information is kept outside the (unambiguous) `CommonData` file. Several options for this treatment are often provided in the form of multiple `SYSTYPE` files which may be selected between in the fit.

Each `SYSTYPE` file begins with a line specifying the total number of systematics. Naturally this must match with the N_{sys} variable specified in the associated `CommonData` file. This is presented as a simple number. For example, in the case of the BCDMSD `SYSTYPE` files, the first line is

```
8
```

As there are $N_{\text{sys}} = 8$ sources of systematic uncertainty for this *Dataset*. Following this line are N_{sys} lines, describing each source of systematic uncertainty. For each source two parameters are provided, the *uncertainty treatment* and the *uncertainty description*. These are laid out for each systematic as:

$$i_{\text{sys}} \quad [\textit{uncertainty treatment}] \quad [\textit{uncertainty description}]$$

Where $1 < i_{\rm sys} \leq N_{\rm sys}$ enumerates each systematic. The *uncertainty treatment* determines whether the uncertainty should be treated as additive, multiplicative, or in cases where the choice is unclear, as randomised on a replica by replica basis. These choices are selected by using the strings **ADD**, **MULT**, or **RAND**. The *uncertainty description* specifies how the systematic is to be correlated with other datapoints. There are two special cases for the *uncertainty description*, specified by the strings **CORR** and **UNCORR**. These two specify whether the systematic is fully correlated **only** within the *Dataset* (**CORR**), or whether the systematic is totally uncorrelated (**UNCORR**). If the user wishes to correlate a specific uncertainty between multiple *Datasets* within an *Experiment*, then they should use a custom *uncertainty description*. When building a covariance matrix for an *Experiment*, the `nnpdf++` code checks for matches between the *uncertainty descriptions* of systematics of its constituent *Datasets*. If a match is found, the code will correlate those systematics over the relevant datasets.

As an example, let us consider an NNPDF2.3 standard `SYSTYPE` for the BCDMSD *Dataset*.

```
8
1 ADD BCDMSFB
2 ADD BCDMSFS
3 ADD BCDMSFR
4 MULT BCDMSNORM
5 MULT BCDMSRELNORMTARGET
6 MULT CORR
7 MULT CORR
8 MULT CORR
```

Here the first five systematics have custom *uncertainty descriptions*, thereby allowing them to be cross-correlated with other *Datasets* in a larger *Experiment*. Systematics six to eight are specified as being fully correlated, but only within the BCDMSD *Dataset*. Additionally note that the first three systematics are specified as additive, and the remainder are multiplicative. If we compare now to the equivalent `SYSTYPE` file for the BCDMSP *Dataset*:

```
11
1 ADD BCDMSFB
2 ADD BCDMSFS
3 ADD BCDMSFR
4 MULT BCDMSNORM
5 MULT BCDMSRELNORMTARGET
6 MULT CORR
7 MULT CORR
8 MULT CORR
9 MULT CORR
```

10 MULT CORR
11 MULT CORR

It is clear that the first five systematics are the same as in the BCDMSD *Dataset*, and therefore should the two sets be combined into a common *Experiment*, the code will cross-correlate them appropriately. The combination of `SYSTYPE` and `CommonData` is quite flexible. As stated previously, once generated from the original raw experimental data, the `CommonData` file is fixed and should not be altered apart from for the purpose of correcting errors. In practice the full details on the systematic correlation and their treatment is often not precisely specified. This system allows for the safe variation of these parameters for testing purposes.

4 Theory data files

In the `nnpdf++` project, `FK` tables (or grids) are used to provide the information required to compute pQCD cross sections in a compact fashion. With the `FK` method a typical hadronic observable datapoint $\mathcal{O}$, is computed as,

$$\mathcal{O}_d = \sum_{\alpha,\beta}^{N_x} \sum_{i,j}^{N_{\rm pdf}} \sigma^{(d)}_{\alpha\beta ij} N_i^0(x_\alpha) N_j^0(x_\beta). \tag{1}$$

Where $\sigma^{(d)}_{\alpha\beta ij}$, the `FK` table, is a five index object with two indices in flavour (i, j), two indices in x (α, β) and a datapoint index d. $N_i^0(x_\alpha)$ is the $i^{\rm th}$ initial scale PDF in the evolution basis at x-grid point $x = x_\alpha$. Each `FK` table has an internally specified x-grid upon which the PDFs are interpolated. The full 14-PDF evolution basis used in the `FK` tables is given by :

$$\{\gamma, \Sigma, g, V, V3, V8, V15, V24, V35, T3, T8, T15, T24, T35\}\,. \tag{2}$$

Additional information may be introduced via correction factors known internally as C-factors. These consist of datapoint by datapoint multiplicative corrections to the final result of the `FK` convolution $\mathcal{O}$. These are provided by `CFACTOR` files, typical applications being the application of NNLO and electroweak corrections. For processes which depend nonlinearly upon PDFs, such as cross-section ratios or asymmetries, multiple FK tables may be required for one observable. In this case information is provided in the form of a `COMPOUND` file which specifies how the results from several `FK` tables may be combined to produce the target observable. In this section we shall specify the layout of the `FK`, `COMPOUND` and `CFACTOR` files.

4.1 `FK` file format

`FK` preamble layout

The first entry in the file consists of a small description of the `FK` grid. This must be exactly three lines long, and at least contain the name of the *Dataset*. Other than this, the user is free to populate this at will, the description will typically be shown as a banner whenever the FK table is opened. For example the CDFR2KT grid description is:

FK_CDFR2KT.dat

This is followed by the number of data points in the grid and number of x-points in the interpolation grid. Like most parameters this is given as a single descriptive header line followed by the actual parameters. An example with $N_{\rm dat} = 76$, $N_{\rm x} = 50$ would be laid out as:

"ndata nx"
76 50

Next comes information on whether or not the grid represents a hadronic or DIS observable, and its perturbative order. In the case of hadronic data, the string '**HAD**' is used as the parameter, while for DIS the parameter is naturally '**DIS**'. As an example,

"hadronic ptord"
HAD N2LO

Where perturbative order is given as: 0-LO 1-NLO 2-NNLO. This example is therefore for NNLO hadronic data. The next block describes the flavour structure of the grid by means of a flavour map. This map details which flavour channels are active in the grid, using the basis specified in Eqn. 2. For DIS processes, an example section would be

"flavourmap"
0 1 1 0 0 0 0 0 0 0 1 0 0 0

Which specifies that only the Singlet, gluon and T_8 channels are populated in the grid. In the case of hadronic FK tables, the full 14×14 flavour combination matrix is specified in the same manner. Consider the flavourmap for the CDFR2KT *Dataset*:

"flavourmap"
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0

This flavourmap contains 9 nonzero entries, demonstrating the importance of only computing those flavour combinations that are relevant to the process. Additionally this map instructs the `nnpdf++` convolution code as to which elements of the FastKernel grid should be read, to minimise holding zero entries in memory.

Next we have a line describing the initial scale at which the evolution in the FK grid is performed from:

"q0"
1.0

In this case the initial scale PDFs $N^0(x)$ are taken at $Q^2 = 1$ GeV. The next few lines describe further details of the evolution kernel in the FK grid. Firstly, a string describing the method of PDF evolution.

"FNS Modev NF"
FONLL-C TRN 5

Here we are using the FONNL-C VFNS, with truncated evolution up to a maximum of $n_f = 5$ active flavours. For the FNS parameter, the currently permitted values are: **FFNS** for the fixed-flavour number scheme, **ZMVFN** for the zero-mass scheme, **FONLL-A/B/C** for variants of the FONLL general mass scheme. For the Modev parameter, **TRN** for truncated solution, **ITE** for iterated solution of the evolution equations.

This is followed by the value of α_S and the reference scale at which it was evaluated (typically M_Z):

"alphas qref"
0.118 91.2

The details of the heavy quark masses used in the evolution are now specified, for example

"mass mc, mb, mt"
POLE 1.275 4.18 173.07

Where the *mass* parameter can take the values **POLE** when using the heavy quark pole masses, and **MSBAR** when using $\overline{\text{MS}}$ masses. The *mc*, *mb*, *mt* parameters control the charm, bottom and top masses respectively.

The final theory parameters specified directly in the `FK` table preamble control whether QED corrections are present, the value of α_{QED}, and it's associated reference scale. Typically this is switched off as so:

"QED alphaqed qref"
OFF 0 0

When required, the *QED* parameter should be set to **ON** and the *alphaqed* and *qref* parameters filled.

The final section of the preamble before the grid itself defines the x grid upon which the `FK` grid is defined, given as an N_x long list of the x-grid points. This grid should be optimised to minimise `FK` grid zeros in x-space. Here is an example of an x-grid with $N_x = 5$ entries:

"NNPDFgrid"
0.10000000000000001
0.13750000000000001
0.17499999999999999
0.21250000000000002
1.00000000000000000

This grid ends the preamble. For examples of complete DIS and hadronic `FK` table headers, see Appendix B.

`FK` grid layout

To start the section of the file with the `FK` grid itself, we begin with a header line

"FKGrid"

The grid itself is now written out. For hadronic data, the format is line by line as follows:

$$d \ \alpha \ \beta \ \sigma^d_{\alpha\beta 11} \ \sigma^d_{\alpha\beta 12} \ \ldots. \ \sigma^d_{\alpha\beta nn} \tag{3}$$

Where d is the index of the data point for that line, α is the x-index of the first PDF, β is the x-index of the second PDF, the $\sigma^d_{\alpha\beta ij}$ are the values of the FastKernel grid for data point d as in Eqn 1, and $n = 14$ is the total number of parton flavours in the grid. Therefore the full 14×14 flavour space for one combination of the indices $\{d, \alpha, \beta\}$ is written out on each line.

These lines should be written out first in β, then α and finally d so that the FK grids are written in blocks of datapoints. All FK grid values should be written out in double precision. For DIS data the FK grids must be written out as

$$d \ \alpha \ \sigma^d_{\alpha 1} \ \sigma^d_{\alpha 2} \ .... \ \sigma^d_{\alpha n} \tag{4}$$

Therefore here all $n = 14$ values are written out for each combination of $\{d, \alpha\}$.

When writing out the grids, note that only x-grid points for which there are nonzero FK entries are written out. For example, there should be no lines such as:

d α β 0 0 0 ... 0

However, for those x-grid points which do have nonzero σ contributions, the full set of flavour contributions must be written out regardless of the number of zero entries. This choice was made in order that the nonzero flavour entries may be examined/optimised by hand after the FK table is generated.

The FK file should end on the last entry in the grid, and without empty lines at the end of file.

4.2 CFACTOR file format

Additional multiplicative factors to be applied to the output of the FK convolution may be introduced by the use of CFACTOR files. These files have a very simple format. The first lines give a description of the C-factor information stored in the file. This segment is initialised and terminated by a line beginning with a star (*) character and consists of six mandatory fields

- **SetName** - The *Dataset* name.
- **Author** - The author of the CFACTOR file.
- **Date** - The date of authorship.
- **CodesUsed** - The code or codes used in generating the $C-$factors.
- **TheoryInput** - Theory input parameters used in the $C-$factors (e.g α_S, scales).
- **PDFset** - The PDF set used in the $C-$factors.

These fields are formatted as

FieldName: FieldEntry

and may be accompanied by any additional information, within the star delineated header region. Consider the following as a complete example of the header,

```
****************************************
SetName: D0ZRAP
Author: John Doe john.doe@cern.ch
Date: 2014
CodesUsed: MCFM 15.01
TheoryInput: as 0.118, central scale 91.2 GeV
PDFset: NNPDF30_as_0118_nnlo
Warnings: None
Additional Information here
****************************************
```

The remainder of the file is simply given by the C-factors themselves, with the C-factor for each datapoint being given line by line. For example, for *Dataset* with five points, the data section of a `CFACTOR` file may be:

```
1.1
1.2
1.3
1.4
1.5
```

Where the i^{th} line corresponds to the C-factor to be applied to the `FK` prediction for the $(i-1)^{\text{th}}$ datapoint. For a complete example of a `CFACTOR` file, please see Appendix A.

4.3 `COMPOUND` file format

Some *Datasets* cover observables that depend nonlinearly upon the input PDFs. For example, the NMCPD *Dataset* is a measurement of the ratio of deuteron to proton structure functions. In the `nnpdf++` code such sets are denoted *Compound Datasets*. In these cases, a prescription for how the results from FK convolutions as in Eqn 1 should be combined must be given.

The `COMPOUND` files are a simple method of providing this information. For each *Compound Dataset* a `COMPOUND` file is provided that contains the information on how to build the observable from constituent `FK` tables. The following operations are currently implemented.

Here $N_{\texttt{FK}}$ refers to the number of tables required for each compound operation. $\mathcal{O}_d$ is final observable prediction for the d^{th} point in the *Dataset*. $\mathcal{O}_d^{(i)}$ refers to the observable prediction for the d^{th} point arising from the i^{th} `FK` table calculation. Note that here the ordering in i is important.

The `COMPOUND` file layout is as so. The first line is once again a general comment line and is not used by the code, and therefore has no particular requirements other than its presence. Following this line should come a list of the FK tables required for the

Operation ($N_{\texttt{FK}}$)	Code	Output Observable
Null Operation(1)	**NULL**	$\mathcal{O}_d = \mathcal{O}_d^{(1)}$
Sum (2)	**ADD**	$\mathcal{O}_d = \mathcal{O}_d^{(1)} + \mathcal{O}_d^{(2)}$
Normalised Sum (4)	**SMN**	$\mathcal{O}_d = (\mathcal{O}_d^{(1)} + \mathcal{O}_d^{(2)})/(\mathcal{O}_d^{(3)} + \mathcal{O}_d^{(4)})$
Asymmetry (2)	**ASY**	$\mathcal{O}_d = (\mathcal{O}_d^{(1)} - \mathcal{O}_d^{(2)})/(\mathcal{O}_d^{(1)} + \mathcal{O}_d^{(2)})$
Ratio (2)	**RATIO**	$\mathcal{O}_d = \mathcal{O}_d^{(1)}/\mathcal{O}_d^{(2)}$

calculation. This must be given as the table's filename *without* its path, preceded by the string '**FK:** '. For example,

FK: FK_SETNAME_1.dat
FK: FK_SETNAME_2.dat

The ordering of the list is once again important, and must match the above table. For example the observables $\mathcal{O}^{(i)}$ arise from the computation with the i^{th} element of this list. The final line specified the operation to be performed upon the list of tables, and must take the form

OP: [**CODE**]

where the [**CODE**] is given in the above table. Here is an example of a complete `COMPOUND` file

COMPOUND FK
FK: FK_NUMERATOR.dat
FK: FK_DENOMINATOR.dat
OP: RATIO

5 Organisation of data files

The `nnpdf++` code needs to be able to handle a great deal of different options with regard to the treatment of both experimental data and theoretical choices. In the code, every effort has been made to keep experimental and theoretical parameters strictly separate.

In this section we shall specify the layout of the various `nnpdf++` data directory. It is in this directory that all of the read-only data to be used in the fit are accessed. The data directory is located in the `nnpdfcpp` git repository, under the path `/nnpdfcpp/data/`. This path is typically supplied to the different projects by way of the compile time macro `DATA_PATH` which is by default set to `../nnpdfcpp/data` therefore assuming that the different `nnpdf++` project repositories have been checked out in the same directory. However the user may change this to specify a custom data directory location.

5.1 Experimental data storage

The central repository for `CommonData` in use by `nnpdf++` projects is located in the `nnpdfcpp` git repository at

```
/nnpdfcpp/data/commondata/
```

Where a separate `CommonData` file is stored for each *Dataset* with the filename format

```
DATA_<SETNAME>.dat
```

Information on the treatment of systematic uncertainties, provided in `SYSTYPE` files, are located in the subdirectory

```
/nnpdfcpp/data/commondata/systypes
```

Here several `SYSTYPE` files may be supplied for each *Dataset*. The various options are enumerated by an integer suffix to the filename. The filename format for `SYSTYPE` files is therefore

```
SYSTYPE_<SETNAME>_<SYSID>.dat
```

As an example, consider the first `SYSTYPE` file for the D0ZRAP *Dataset*:

```
SYSTYPE_D0ZRAP_0.dat
```

In addition to the `SYSTYPE` files, this directory also includes files containing a description of each treatment of systematic errors. They have the filename format

```
SYSTYPES_<SETNAME>.info
```

and have no particular layout, but must enumerate and give a brief explanation for all of the possible systematics treatments for the associated *Dataset*. For example, here is the info file for the D0ZRAP systypes:

```
# SYSTYPES.info : SysType options description.
# Please fill this in when you add a new SysType Option

1: NNPDF2.3 Standard (Default)
2: All systematic uncertainties assumed to be multiplicative
3: All systematic uncertainties assumed to be additive
4: Baseline Standard (Collider data: All systematics random)
```

5.2 Theory lookup table

In order to organise the various different theoretical treatments available, a lookup table is provided in `sqlite3` format. This lookup table can be found in the `nnpdfcpp` repository data directory at:

```
/nnpdfcpp/data/theory.db
```

This file should only be edited in order to add new theory options. It may be edited with any appropriate sqlite3-supported software. A script is provided to give a brief overview of the various theory options available. It can be found at

```
/nnpdfcpp/data/disp_theory.py
```

and should be run without any arguments.

Theory options are enumerated by an integer *TheoryID*. The parameters of each theory option are described in the lookup table under the appropriate ID. The current available parameters are summarised in Table 1.

5.3 Theory storage

Each theory configuration is stored as a GZIP compressed Tar archive with filename format

```
theory_<THEORYID>.tgz
```

and are stored at

```
pcteserver.mi.infn.it:/home/apfelcomb/WEB/commondatatheory/
```

For easy access, they can be downloaded through the `disp_theory.py` script mentioned above, or through the web interface at

```
http://pcteserver.mi.infn.it/~apfelcomb/fkviewer/
```

The uploading of a new theory archive can be performed through the `upload_theory.sh` script which generates and uploads the archive. Each archive contains the following directory structure

```
theory_X/
     -cfactor/
     -compound/
     -fastkernel/
```

Inside the directory `theory_X/cfactor/` are stored `CFACTOR` files with the filename format

```
CF_<TYP>_<SETNAME>.dat
```

where `<TYP>` is a three-letter designation for the source of the C-factor (i.e EWK or QCD) and `<SETNAME>` is the typical *Dataset* designation.

The directory `theory_X/compound/` contains the `COMPOUND` files described earlier, this time with the filename format

```
FK_<SETNAME>-COMPOUND.dat
```

Finally the `FK` tables themselves are stored in `theory_X/fastkernel/` with the filename format

```
FK_<SETNAME>.dat
```

Naturally, all of the FastKernel and C-factor files within the directory `theory_X/` have been determined with the theoretical parameters specified in the theory lookup table under ID `X`.

For use in the `nnpdf++` code, these theory archives should be uncompressed in the relevant data directory. The code then may switch between different theory settings by using the relevant files from the relevant theory directory.

Field	Type	Description	Comments
ID	Integer	*TheoryID*	Theory enumerating ID
PTO	Integer	pQCD order	(0/1/2 = LO/NLO/NNLO)
FNS	Text	Flavour Number Scheme	e.g FONLL-A/B/C or ZM-VFNS or FFNS
DAMP	Integer	FONLL damping factor switch	Boolean
IC	Integer	Intrinsic charm switch	Boolean
ModEv	Text	DGLAP solution mode	EXA/EXP/TRN
XIR	Real	ξ_R	μ_R/Q
XIF	Real	ξ_F	μ_F/Q
NfFF	Integer	Number of flavours in the FFNS	3/4/5/6
MaxNfAs	Integer	$n_{f,\max}^{(\alpha_S)}$	Max active flavours in α_s
MaxNfPdf	Integer	$n_{f,\max}^{(\mathrm{PDF})}$	Max active flavours in PDFs
Q0	Real	Q_0	FK Table initial scale
alphas	Real	Strong coupling	Format: $\alpha_S(Q_{\mathbf{ref}})$
Qref	Real	$Q_{\mathbf{ref}}$	Reference scale for α_S in GeV
QED	Integer	QED switch	Boolean
alphaqed	Real	QED coupling	Format: $\alpha_{\mathrm{QED}}(Q_{\mathbf{qedref}})$
Qedref	Real	$Q_{\mathbf{qedref}}$	QED reference scale (GeV)
SxRes	Integer	small-x resummation switch	Boolean
SxOrd	Text	small-x pt order	("LL", "NLL", "NNLL")
HQ	Text	HQ mass treatment	POLE/MSBAR
mc	Real	c quark mass $M_c/m_c(Q_{\mathrm{mc}})$	Units: GeV
Qmc	Real	Q_{mc}	c reference scale (GeV)
mb	Real	b quark mass $M_b/m_b(Q_{\mathrm{mb}})$	Units: GeV
Qmb	Real	Q_{mb}	b reference scale (GeV)
mt	Real	t quark mass $M_t/m_t(Q_{\mathrm{mt}})$	Units: GeV
Qmt	Real	Q_{mt}	t reference scale (GeV)
CKM	Text	CKM matrix elements	3X3 matrix
MZ	Real	M_Z	Z mass (GeV)
MW	Real	M_W	W mass (GeV)
GF	Real	G_F	Fermi coupling constant
SIN2TW	Real	$\sin^2\theta_W$	
TMC	Integer	Target mass corrections	Boolean
MP	Real	M_P Proton mass	Units: GeV
Comments	Text	General comments	

Table 1: Table describing the contents and required parameters of the `theory.db` lookup table. The Type column refers directly to the `sqlite3` data type of the field.

A CFACTOR file format example

```
****************************************
SetName: D0ZRAP
Author: John Doe john.doe@cern.ch
Date: 2014
CodesUsed: MCFM 15.01
TheoryInput: as 0.118, central scale 91.2 GeV
PDFset: NNPDF30_as_0118_nnlo
Warnings: None
7 datapoints in total. No smoothing applied
****************************************
1.1
1.2
1.3
1.4
1.5
1.6
1.7
```

B FK preamble examples

B.1 DIS preamble - BCDMSD

'_____'
' FK_BCDMSD.dat '
'_____'
'ndata nx'
254 50
"process ptord"
DIS N2LO
"flavourmap"
0 1 1 0 0 0 0 0 0 0 1 0 0 0
"q0"
1.0000000000000000
"FNS Modev nf"
FONLL-C TRN 5
"alphas qref"
0.118 91.200000000000003
"mass mc, mb, mt"
POLE 1.2749999999999999 4.1799999999999997 173.06999999999999
"QED alphaqed qref"
OFF 7.4962520000000001E-003 1.7769999999999999
"NNPDFgrid"
9.9999999999999995E-008
1.7378008287493754E-007
3.0199517204020160E-007
5.2480746024977254E-007
9.1201083935590974E-007
1.5848931924611137E-006
2.7542287033381659E-006
4.7863009232263851E-006
8.3176377110267111E-006
1.4454397707459272E-005
2.5118864315095808E-005
4.3651583224016600E-005
7.5857757502918358E-005
1.3182567385564074E-004
2.2908676527677748E-004
3.9810717055349714E-004
6.9183097091893666E-004

1.2022644346174139E-003
2.0892961308540386E-003
3.6307805477010144E-003
6.3095734448019372E-003
1.0964781961431845E-002
1.9054607179632473E-002
3.3113112148259127E-002
5.7543993733715673E-002
0.10000000000000001
0.13750000000000001
0.17499999999999999
0.21250000000000002
0.25000000000000000
0.28749999999999998
0.32500000000000001
0.36250000000000004
0.40000000000000002
0.43750000000000000
0.47499999999999998
0.51250000000000007
0.55000000000000004
0.58750000000000002
0.62500000000000000
0.66249999999999998
0.69999999999999996
0.73750000000000004
0.77499999999999991
0.81250000000000000
0.84999999999999998
0.88750000000000007
0.92500000000000004
0.96249999999999991
1.0000000000000000

B.2 Hadronic preamble - CDFR2KT

FK_CDFR2KT.dat

"ndata nx"
76 50
"hadronic ptord"
HAD N2LO
"flavourmap"
0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
"q0"
1
"FNS Modev NF"
FONLL-C TRN 5
"alphas qref"
0.118 91.2
"mass mc, mb, mt"
POLE 1.275 4.18 173.07
"QED alphaqed qref"
OFF 0 0
"NNPDFgrid"
1e-07
1.73780082874938e-07
3.01995172040202e-07
5.24807460249773e-07
9.1201083935591e-07
1.58489319246111e-06

2.75422870333817e-06
4.78630092322639e-06
8.31763771102671e-06
1.44543977074593e-05
2.51188643150958e-05
4.36515832240166e-05
7.58577575029184e-05
0.000131825673855641
0.000229086765276777
0.000398107170553497
0.000691830970918937
0.0012022644346174 1
0.0020892961308540 4
0.0036307805477010 1
0.0063095734448019 4
0.010964781961431 8
0.019054607179632 5
0.033113112148259 1
0.057543993733715 7
0.1
0.1375
0.175
0.2125
0.25
0.2875
0.325
0.3625
0.4
0.4375
0.475
0.5125
0.55
0.5875
0.625
0.6625
0.7
0.7375
0.775
0.8125
0.85
0.8875

0.925
0.9625
1

References

[1] G. Aad *et al.* [ATLAS Collaboration], Phys. Rev. D **86** (2012) 014022 [arXiv:1112.6297 [hep-ex]].

[2] G. Aad *et al.* [ATLAS Collaboration], Phys. Lett. B **725** (2013) 223 [arXiv:1305.4192 [hep-ex]].