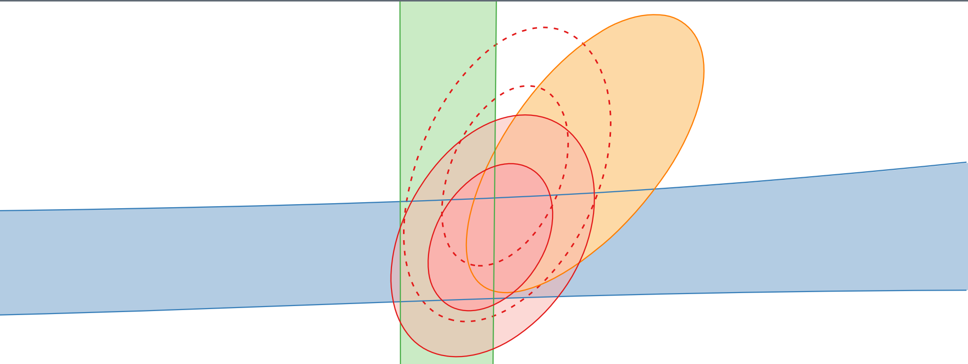
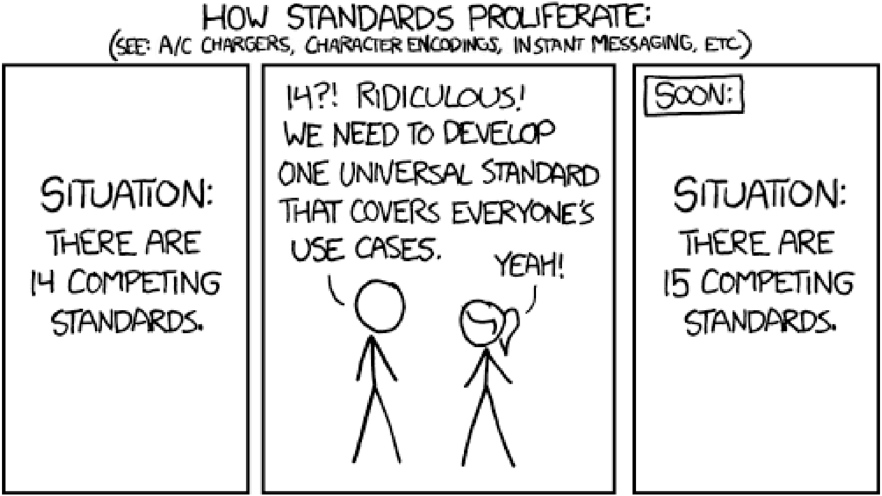
2019-06-13, SMEFT Tools 2019, IPPP Durham

flavio & smelli

David M. Straub Universe Cluster/TUM, Munich



Comment on WCxf



DsixTools 2.0 vs. WCxf

arXiv:1712.05298:

In principle, WCxf is not necessarily a *file* format but can also be used to directly exchange data structures between programs, bypassing the file system (e.g., in the case of Python this could be a wcxf.WC instance discussed in section 4.2 or simply a dictionary). The rest of this document will use the YAML format in all examples.

```
NewInput["SMEFT", {CLQ1[1,1,1,2] -> 1, CH -> -0.5}]
```

Why not:

```
WCxfInput["SMEFT", "Warsaw up", {"lq1_1112" -> 1, "phi" -> -0.5}]
```

Outline: smelli & flavio

1. What is it?
2. Observables: status
3. Comments on statistics
4. Tutorials

flavio: what is it?

flavio is a Python (3.5+) package with the following main features:

- General **observable calculator** (with uncertainties) in terms of WET or SMEFT WCs
- Database of **experimental meausurements**
- Construction of **likelihoods**

In addition, it contains

- convenient plotting routines
- interfaces to fitters (MCMC)
- a frequentist likelihood profiler

About

- Docs: `https://flav-io.github.io/`
- Repo: `https://github.com/flav-io/flavio`
- Paper: arXiv:1810.08132 (not a manual)
- Main developer: myself, contributions from the community

smelli: what is it?

The **SME**FT **L**ike**li**hood package is built on top of flavio and provides a **global likelihood** in the space of **SMEFT WCs**. The main motivation for smelli was:

- Providing a consistent set of **observables** included in the likelihood
- Correct treatment of **SM parameters** in the presence of $D = 6$ effects
- Construction of a **nuisance-free** likelihood
- More informative **presentation** of results (table of observables with pulls etc.)

About

- Docs: `http://smelli.github.io/`
- Repo: `https://github.com/smelli/smelli`
- Paper: arXiv:1810.07698
- Developers: Peter Stangl & myself (authors: Jason Aebischer & Jacky Kumar)

Relation between flavio & smelli

- flavio aims to be easy to modify and flexible:
  - modify parameters & uncertainties
  - change form factor parametrizations etc.
  - change experimental measurements used
  - General treatment of theory uncertainties, Bayesian or frequentist
- smelli is meant to be less flexible but easier to use:
  - pre-selected set of observables and measurements
  - Default values of parameters & uncertainties
  - self-consistent choice of observables, measurements, and parameters
  - *nuisance-free* likelihood in WC space

flavio & smelli: master plan

- **flavio** to include every observable where NP can be expressed in terms of WET or SMEFT WCs and all their experimental measurements
- **smelli** to become truly global to constrain as many directions in SMEFT WC space as at all possible

Observables in flavio v1.5

See `https://flav-io.github.io/docs/observables.html`

- B physics: $B \to (V, P, X)\ell\ell$, $B \to \ell\ell$, $B \to (V, X)\gamma$, $\Lambda_b \to \Lambda\ell\ell$, $B \to (V, P, X)\ell\nu$, $B \to \ell\nu$, mixing
- K physics: $K \to \pi\nu\nu$, $K \to \ell\ell$, $K \to \ell\nu$, $K \to \pi\ell\nu$, ε_K, ε'/ε
- D physics: $D \to \ell\nu$, CPV in mixing
- μ physics: $\mu \to e\gamma$, $\mu \to 3e$, μ-e conversion, ν trident
- τ physics: $\tau \to 3\ell$, $\tau \to \ell\gamma$, $\tau \to (P, V)\ell$, $\tau \to V\nu$, $\tau \to \ell\nu\nu$
- EWPT: All LEP-1 Z and W pole observables
- Dipole moments: $(g-2)_{e,\mu,\tau}$, d_n

Observables in flavio v1.5

See `https://flav-io.github.io/docs/observables.html`

- B physics: $B \to (V,P,X)\ell\ell$, $B \to \ell\ell$, $B \to (V,X)\gamma$, $\Lambda_b \to \Lambda\ell\ell$, $B \to (V,P,X)\ell\nu$, $B \to \ell\nu$, mixing
- K physics: $K \to \pi\nu\nu$, $K \to \ell\ell$, $K \to \ell\nu$, $K \to \pi\ell\nu$, ε_K, ε'/ε
- D physics: $D \to \ell\nu$, CPV in mixing
- μ physics: $\mu \to e\gamma$, $\mu \to 3e$, μ-e conversion, ν trident
- τ physics: $\tau \to 3\ell$, $\tau \to \ell\gamma$, $\tau \to (P,V)\ell$, $\tau \to V\nu$, $\tau \to \ell\nu\nu$
- EWPT: All LEP-1 Z and W pole observables
- Dipole moments: $(g-2)_{e,\mu,\tau}$, d_n

Work in progress (don't tell anyone):

- Nuclear and neutron β decays A. Falkowski, M. Gonzalez-Alonso, M. Jung, DS
- Higgs production and decay A. Falkowski, M. Spannowsky, DS
- Paramagnetic EDMs
- even more secret stuff

Observables in smelli v1.3

- Every *measured* observable in flavio that is relevant for SMEFT
- **except** semi-leptonic charged-current decays since we hadn't yet accounted for NP affecting CKM extractions
- This problem has now been solved thanks to the approach suggested in
  Descotes-Genon et al. 1812.08163
- A similar approach has been implement in smelli and will be public soon

Future treatment of CKM elements in smelli

1. Select 4 input observables
2. Given $\vec{C}$, extract the 4 parameters of the *true* CKM matrix from these 4 observables, e.g. in the presence of $C_{\varphi ud}$:

$$\text{BR}(B \to \tau\nu) \propto \left|V_{ub} - \tfrac{v^2}{2} C_{\varphi ud}^{23}\right|^2$$

   - The *true* CKM matrix is defined by:

$$M_d^{\text{diag}} = \frac{v}{\sqrt{2}} \left(Y_d - \frac{v^2}{2} C_{d\varphi} \right)$$

$$M_u^{\text{diag}} = \frac{v}{\sqrt{2}} V \left(Y_u - \frac{v^2}{2} C_{u\varphi} \right)$$

   - V is unitary, but it is not the matrix proportional to the W vertex
3. Use “true” V in all calculations of SM contributions

$\Rightarrow$ no need to profile/marginalize over CKM parameters

How to add additional observables?

To add an observable, all we need is:

1. The prediction in terms of parameters and SMEFT WCs (any WCxf EFT/basis!):

```
def my_obs(par, wc):
  ...
  return 1 + wc['phiD'] + ...
```

2. The experimental likelihood/measurement as YAML file:

```
Measurement of my obs:
  values:
    my_obs: 3 ± 0.2
```

Flavio likelihoods

The `flavio.statistics.likelihood` module defines two types of likelihoods:

1. `Likelihood`

Provides the likelihood as a function of parameters $\vec{p}$ and WCs $\vec{C}$, optionally including theory constraints L_{th} on them (Bayesian: priors, frequentist: pseudo-measurements)

$$L(\vec{p},\vec{C}) = L_{\text{exp}}(\vec{O}(\vec{p},\vec{C})) \times L_{\text{th}}(\vec{p},\vec{C})$$

where $L_{\text{exp}}(\vec{O})$ is a *measurement* of observables $\vec{O}$

Flavio likelihoods

The `flavio.statistics.likelihood` module defines two types of likelihoods:

1. `Likelihood`

Provides the likelihood as a function of parameters $\vec{p}$ and WCs $\vec{C}$, optionally including theory constraints L_{th} on them (Bayesian: priors, frequentist: pseudo-measurements)

$$L(\vec{p},\vec{C}) = L_{\text{exp}}(\vec{O}(\vec{p},\vec{C})) \times L_{\text{th}}(\vec{p},\vec{C})$$

where $L_{\text{exp}}(\vec{O})$ is a *measurement* of observables $\vec{O}$

2. `FastLikelihood`

Here all measurements are approximated as Gaussian and combined with the correlated theory uncertainties into a pseudo-measurement:

$$\ln L_{\text{pseudo}}(\vec{C}) = -\frac{1}{2} x(\hat{\vec{p}},\vec{C})^T (C_{\text{exp}} + C_{\text{th}})^{-1} x(\hat{\vec{p}},\vec{C})$$

where x is the difference between theory predictions and measurements, C_i are the covariances, and $\hat{\vec{p}}$ the central values of the parameters

Nuisance-free likelihood in smelli

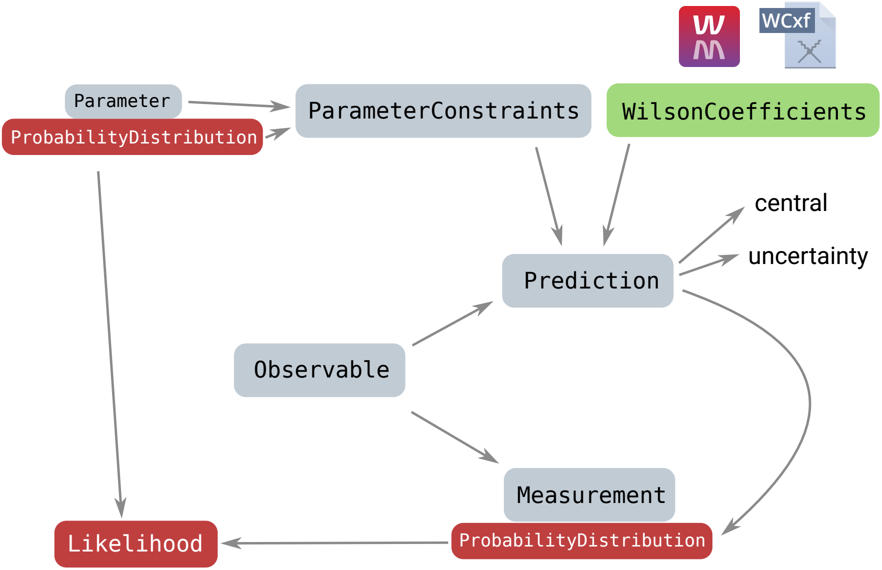
Problem

- We want a nuisance-free likelihood (avoid costly marginalization/profiling)
- For $b \to s\ell\ell$ fits, custom to use `FastLikelihood`: correlated uncertainties accounted for, NP-dependence of uncertainties smallish
- Sometimes, cannot justify Gaussian approximation. E.g. strongly asymmetric measurements or Poisson upper limits

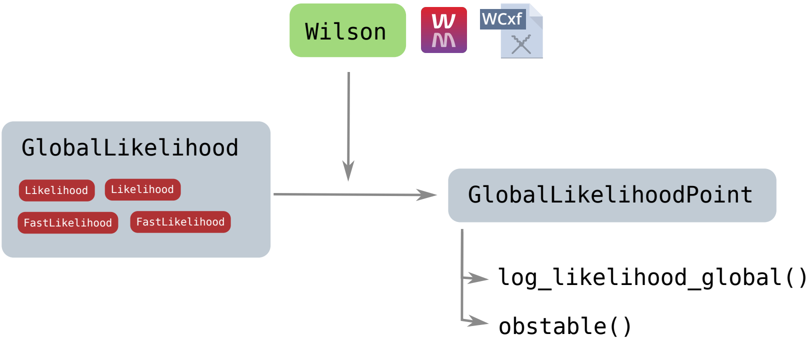
Solution

- Use `FastLikelihood` where NP-dependence of uncertainties $<$ exp uncertainties and Gaussian approximation justified
- Use full experimental likelihood *but* ignore theory uncertainties where justified
- *Do not include* observables where neither approach works (e.g.: neutron EDM)
  - thinking about ways to overcome this limitation, happy about suggestions

flavio: structure of the code



smelli: structure of the code



Tutorials

`https://github.com/DavidMStraub/flavio-smelli-mini-tutorials`