GammaCombo

Moritz Karbach
moritz.karbach@cern.ch

CERN

25.02.2015

Outline

1. Introduction
2. Two less academic examples
   - $B \to D^* \tau \nu$
   - V_{ub}
3. PLUGIN method
4. Interfacing custom code
5. Control plots
6. A possible scenario
7. A real-life example: LHCb γ
8. Summary

Outline

1. Introduction
2. Two less academic examples
   - $B \to D^* \tau \nu$
   - V_{ub}
3. Plugin method
4. Interfacing custom code
5. Control plots
6. A possible scenario
7. A real-life example: LHCb γ
8. Summary

Combining measurements

- This talk is about software, not statistics.

Combining measurements

- This talk is about software, not statistics.
- Basic approach: work with a number of experimental observables, assume they are Gaussian.
- There is often no other choice that works in practice (e.g. people won't interface their fitters to original data).

Combining measurements

- This talk is about software, not statistics.
- Basic approach: work with a number of experimental observables, assume they are Gaussian.
- There is often no other choice that works in practice (e.g. people won't interface their fitters to original data).
- Form likelihood

$$\mathcal{L}(\vec{\alpha}) = \exp\left(-\frac{1}{2}\left(\vec{A}_{\mathrm{th}}(\vec{\alpha}) - \vec{A}_{\mathsf{obs}}\right)^T V^{-1}\left(\vec{A}_{\mathrm{th}}(\vec{\alpha}) - \vec{A}_{\mathsf{obs}}\right)\right)\,, \quad (1)$$

- Express observables as functions of theory parameters: $\vec{A}_{\mathrm{th}}(\vec{\alpha})$

Combining measurements

- This talk is about software, not statistics.
- Basic approach: work with a number of experimental observables, assume they are Gaussian.
- There is often no other choice that works in practice (e.g. people won't interface their fitters to original data).
- Form likelihood

$$\mathcal{L}(\vec{\alpha}) = \exp\left(-\frac{1}{2}\left(\vec{A}_{\rm th}(\vec{\alpha}) - \vec{A}_{\sf obs}\right)^T V^{-1}\left(\vec{A}_{\rm th}(\vec{\alpha}) - \vec{A}_{\sf obs}\right)\right)\,, \quad (1)$$

- Express observables as functions of theory parameters: $\vec{A}_{\rm th}(\vec{\alpha})$
- Then, e.g., profile the likelihood, determine CL intervals through Wilk's theorem ("go up 0.5").
- Other ways to construct intervals exist, each has pros and cons.

GammaCombo: What does it do for you?

You have several observables, and know how they are related to theory parameters. You have 30 minutes to spare (or a paper to write!).

GammaCombo: What does it do for you?

You have several observables, and know how they are related to theory parameters. You have 30 minutes to spare (or a paper to write!).

- GammaCombo computes their best fit values, along with **confidence intervals**

GammaCombo: What does it do for you?

You have several observables, and know how they are related to theory parameters. You have 30 minutes to spare (or a paper to write!).

- GammaCombo computes their best fit values, along with **confidence intervals**
- it is a **combination framework:**
  - bookkeeping of measurements
  - plug-and-play on the command line: What happens if I add this new result? What if that parameter is fixed?
  - visualize anatomy of combinations

GammaCombo: What does it do for you?

You have several observables, and know how they are related to theory parameters. You have 30 minutes to spare (or a paper to write!).

- GammaCombo computes their best fit values, along with **confidence intervals**
- it is a **combination framework:**
  - bookkeeping of measurements
  - plug-and-play on the command line: What happens if I add this new result? What if that parameter is fixed?
  - visualize anatomy of combinations
- it makes **plots**

GammaCombo: What does it do for you?

You have several observables, and know how they are related to theory parameters. You have 30 minutes to spare (or a paper to write!).

- GammaCombo computes their best fit values, along with **confidence intervals**
- it is a **combination framework:**
  - bookkeeping of measurements
  - plug-and-play on the command line: What happens if I add this new result? What if that parameter is fixed?
  - visualize anatomy of combinations
- it makes **plots**
- **tools to debug your likelihood** (multiple maxima, pull plots, . . .)

GammaCombo: What does it do for you?

You have several observables, and know how they are related to theory parameters. You have 30 minutes to spare (or a paper to write!).

- GammaCombo computes their best fit values, along with **confidence intervals**
- it is a **combination framework:**
  - bookkeeping of measurements
  - plug-and-play on the command line: What happens if I add this new result? What if that parameter is fixed?
  - visualize anatomy of combinations
- it makes **plots**
- **tools to debug your likelihood** (multiple maxima, pull plots, . . .)
- it doesn't make it *too* easy—statistics is hardly ever black boxable (please *do* look at your generated toys . . .)

Implemented in GammaCombo

- profile likelihood with Wilk's theorem (the "`Prob`" method)
- frequentist method based on toys with nuisances at best-fit values (the "plug-in" method)
- frequentist Berger-Boos method (hardly usable in practice)

Wishlist

- Cousins-Highland (Bayesian-frequentist hybrid)
- ...

GammaCombo

Was originally developed to combine LHCb measurements of the CKM angle γ.

- LHCb γ combination: Phys. Lett. B726 (2013) 151, arXiv:1305.2050.
- Now used in the context of $B_s^0 \to \mu\mu$ (Max Schlupp). Here the fits to original data are being re-run!
- Used for LHCb-internal charm combinations.
- Used for LHCb-internal combinations of B_s^0-system observables.
- RooFit based, c++
- Under active development (CERN fellows, Ph.D. student).

GammaCombo installation

The code lives on GitHub:

- fork me on `https://github.com/mkarbach/gammacombo`

Manual:

- `https://github.com/mkarbach/gammacombo/blob/master/manual/GammaComboManual.pdf`

Dependencies:

- `git`, `cmake`, `C++11`, boost, `ROOT`, `RooFit`

```
git clone https://github.com/mkarbach/gammacombo.git
cd gammacombo
mkdir build; cd build
cmake ..
make install -j4
cd ../tutorial
```

try it now!

Simple example

- The code contains c++ classes that represent "measurements".
- They capsule the numerical inputs: central values, uncertainties, correlations
- Let's assume a simple single Gaussian measurement:

$$\mathrm{PDF} = G(a_{\mathrm{obs}} | \mu = a_{\mathrm{th}}, \sigma = \sigma_{a_{\mathrm{obs}}}) \tag{2}$$

- $a_{\mathrm{obs}} \pm \sigma_{a_{\mathrm{obs}}}$ — observable and its error
- a_{th} — theory relation (in this case identity)

Simple example

- Define parameters:

```
void PDF_Gaus::initParameters(){
  ParametersGammaCombo p;
  parameters = new RooArgList("parameters");
  parameters->add(*(p.get("a_gaus")));
}
```

- Define observables:

```
void PDF_Gaus::initObservables(){
  observables = new RooArgList("observables");
  observables->add(*(new RooRealVar("a_gaus_obs",
    "a_gaus_obs", 0, -1e4, 1e4)));
}
```

Simple example

- Define theory relations:

```
void PDF_Gaus::initRelations()
{
  theory = new RooArgList("theory");
  theory->add(*(new RooFormulaVar("a_gaus_th",
    "a_gaus_th", "a_gaus", *(RooArgSet*)parameters)));
}
```

Simple example

- Set observed central values:

```
void PDF_Gaus::setObservables(TString c){
  if ( c.EqualTo("year2013") ){
    obsValSource = c;     // <-- to help your bookkeeping
    setObservable("a_gaus_obs",-0.5);
}}
```

- Set errors:

```
void PDF_Gaus::setUncertainties(TString c){
  if ( c.EqualTo("year2013") ){
    obsErrSource = c;
    StatErr[0] = 1; // a_gaus_obs
    SystErr[0] = 0;
}}
```

Simple example

- Set correlations:

```
void PDF_Gaus::setCorrelations(TString c)
{
  if ( c.EqualTo("year2013") ){
    corSource = c;
    //corStatMatrix[1][0] = 0.6; // no correlation for only
    //corSystMatrix[1][0] = 0.0; // one observable ...
}}
```

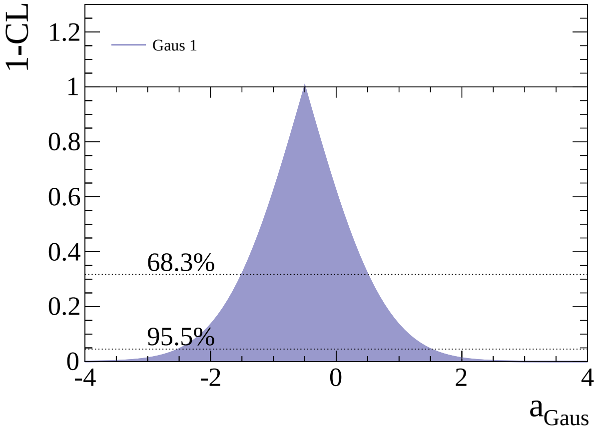
- Set PDF

```
void PDF_Gaus::buildPdf()
{
  pdf = new RooMultiVarGaussian("pdf_"+name, "pdf_"+name,
       *(RooArgSet*)observables, *(RooArgSet*)theory, covMatrix);
}
```

Simple example

Run scan 1: in directory `cd gammacombo/tutorial`

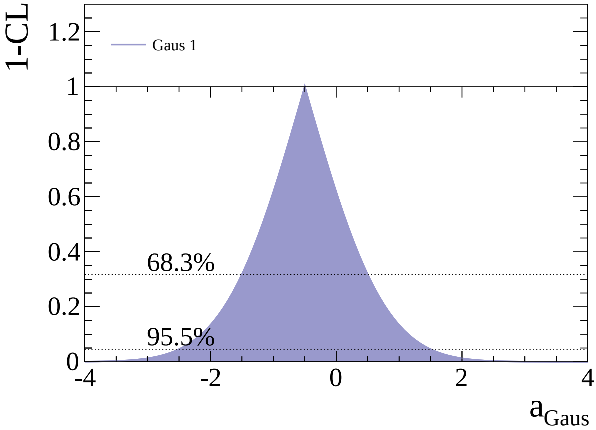
`bin/tutorial --var a_gaus -c 1`



Simple example

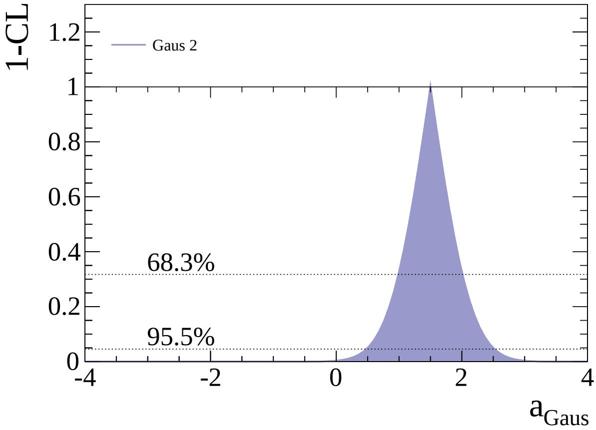
Run scan 1: in directory `cd gammacombo/tutorial`

`bin/tutorial --var a_gaus -c 1 -i`



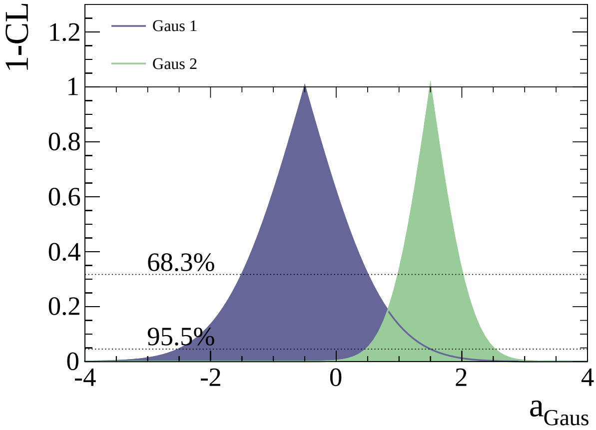
Simple example

Run scan 2, using a different Gaussian measurement:
`bin/tutorial --var a_gaus -c 2`



Simple example

Run scans 1 and 2 in same plot:
`bin/tutorial --var a_gaus -c 1 -c 2`



Simple example

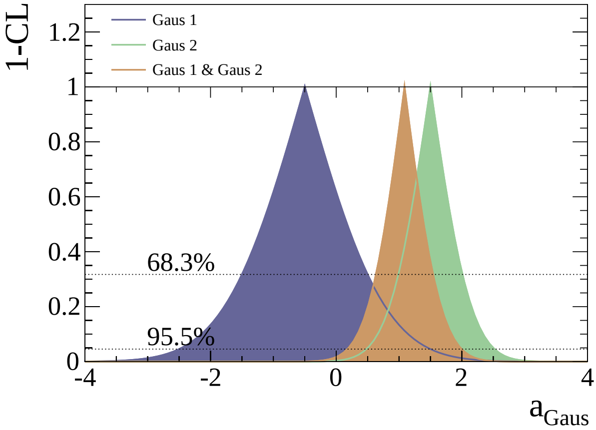
- Define what measurements get combined:

```
int main(int argc, char* argv[])
{
GammaComboEngine gc("tutorial", argc, argv);
// add measurements
gc.addPdf(1, new PDF_Gaus("year2013","year2013","year2013"), "1D
    Gaussian (a_{obs} = -0.5)");
gc.addPdf(2, new PDF_Gaus("year2014","year2014","year2014"), "1D
    Gaussian (a_{obs} = 1.5)");
// define combinations
gc.newCombiner(1, "tutorial1", "Gaus 1",      1);
gc.newCombiner(2, "tutorial2", "Gaus 2",      2);
gc.newCombiner(3, "tutorial3", "Gaus 1 & Gaus 2", 1,2);
// Run
gc.run();
}
```

Simple example

Run scans 1, 2 and combination 3 in same plot:

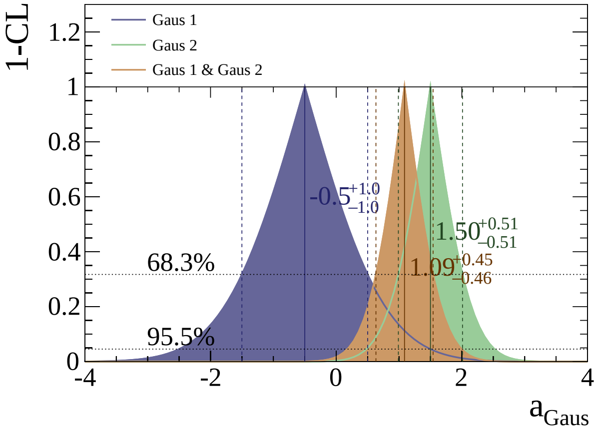
`bin/tutorial --var a_gaus -c 1 -c 2 -c 3`



Simple example

Add central values to plot:

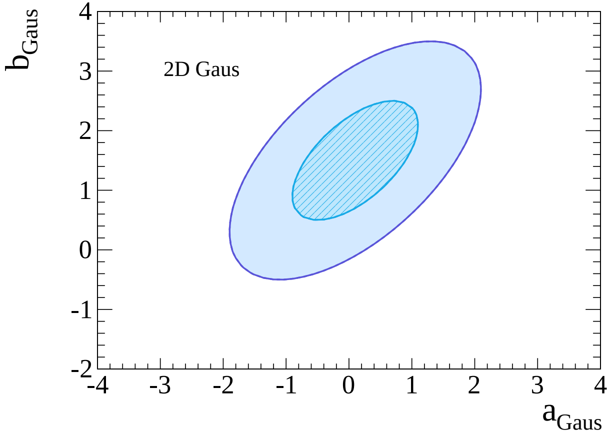
`bin/tutorial --var a_gaus -c 1 -c 2 -c 3 --ps 1`



2D example

Define a 2D Gaussian PDF with correlation:

```
bin/tutorial --var a_gaus --var b_gaus -c 4
```



"Circle" example

- Let's encode a more complicated relation between theory and observables:

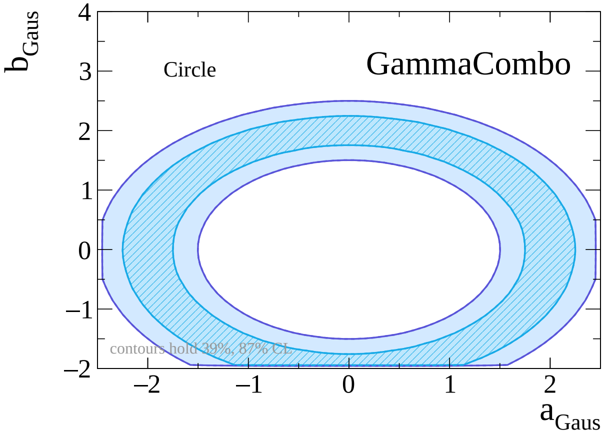
$$r_{\rm th} = \sqrt{a^2 + b^2}$$

```
void PDF_Circle::initRelations()
{
  theory = new RooArgList("theory");
  theory->add(*(new RooFormulaVar("radius_th", "radius_th",
    "sqrt(a_gaus^2+b_gaus^2)", *(RooArgSet*)parameters)));
}
```

"Circle" example

Scan circle:

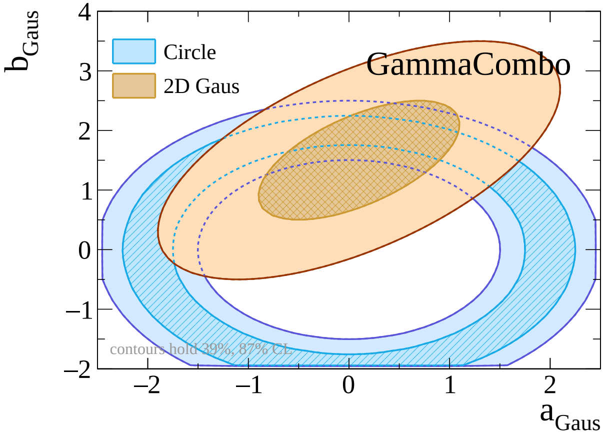
```
bin/tutorial --var a_gaus --var b_gaus -c 6
```



"Circle" example

Scan circle:

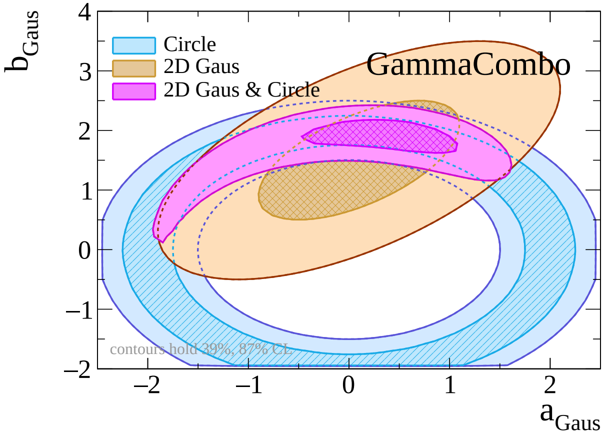
```
bin/tutorial --var a_gaus --var b_gaus -c 6 -c 4
```



"Circle" example

Scan circle:

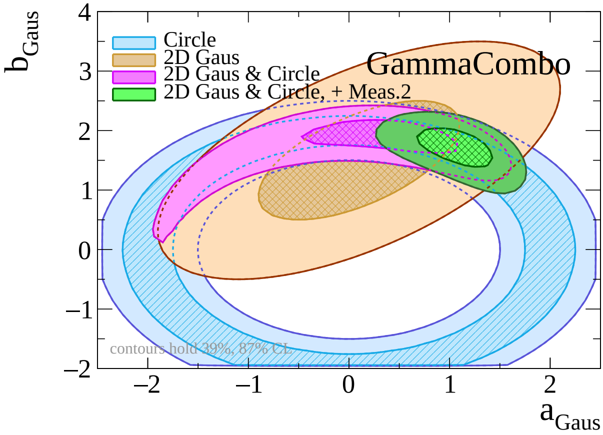
```
bin/tutorial --var a_gaus --var b_gaus -c 6 -c 4 -c 7
```



"Circle" example

Scan circle: make a new combination on the fly!

```
bin/tutorial --var a_gaus --var b_gaus -c 6 -c 4 -c 7 -c 7:+2
```



Outline

$B \to D^* \tau \nu$

Can the anomalous measurement of $B \to D^* \tau \nu$ be explained by new physics?

$$R_D = R_D^{\mathrm{SM}} \times (1 + a \cdot (S_R + S_L) + b \cdot (S_R + S_L)^2) \tag{3}$$
$$R_D^* = R_D^{*\mathrm{SM}} \times (1 + c \cdot (S_R - S_L) + d \cdot (S_R - S_L)^2) \tag{4}$$

where the S_R and S_L are the real values of the type III 2HDM parameters. BaBar observables and constants [1, 2]

$$R_D = 0.440 \pm 0.058 \pm 0.042, \tag{5}$$
$$R_D^* = 0.332 \pm 0.024 \pm 0.018, \tag{6}$$
$$R_D^{\mathrm{SM}} = 0.297, \quad R_D^{*,\mathrm{SM}} = 0.252, \tag{7}$$
$$a = 1.5, \quad b = 1.0, \quad c = 0.12, \quad d = 0.05 \tag{8}$$

$B \to D^*\tau\nu$

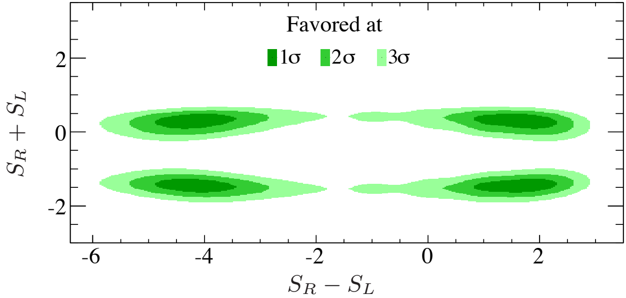


Figure: Figure taken from Babar [1]

$B \to D^*\tau\nu$

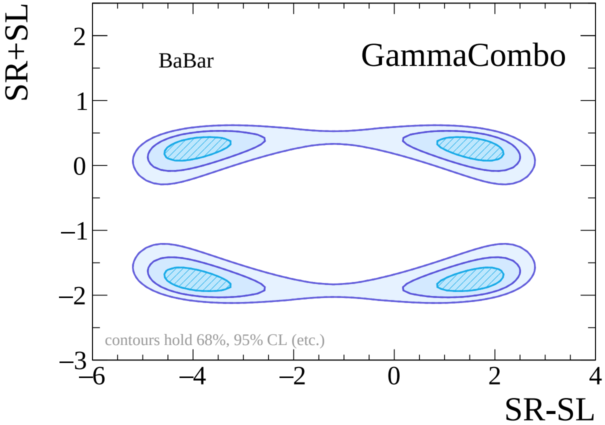


Figure: Reproduced in GammaCombo

$B \to D^*\tau\nu$

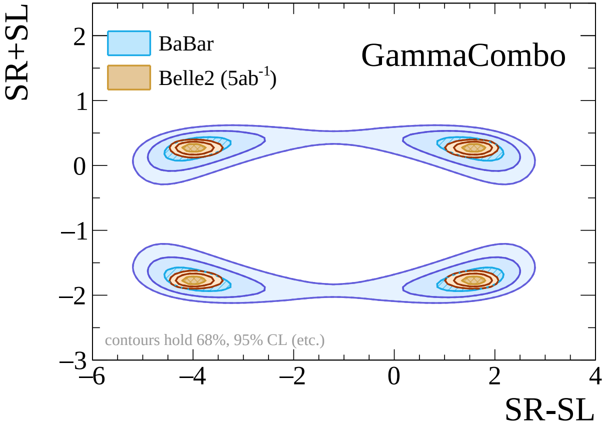


Figure: Reproduced in GammaCombo

$B \to D^* \tau\nu$

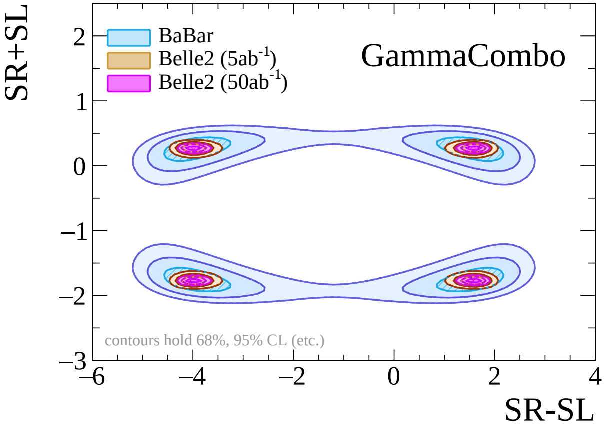


Figure: Reproduced in GammaCombo

Are there right handed currents in $V_{\rm ub}$?

$$V_{\rm ub}^{\pi} = V_{\rm ub}^{L} \cdot (1+\epsilon_R), \quad V_{\rm ub}^{X_u} = V_{\rm ub}^{L} \cdot \sqrt{1+\epsilon_R^2}, \quad V_{\rm ub}^{\tau} = V_{\rm ub}^{L} \cdot (1-\epsilon_R)$$

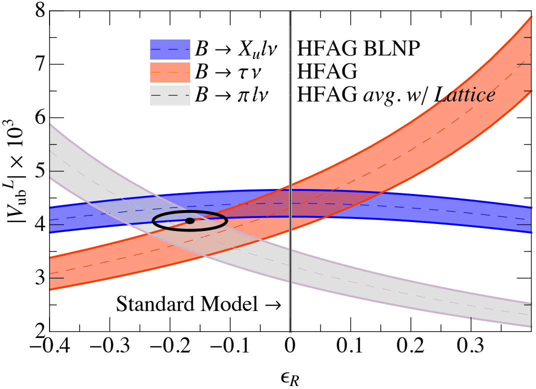


Figure: taken from Ref. [3]

Are there right handed currents in V_{ub}?

$$V^{\pi}_{\mathrm{ub}} = V^{L}_{\mathrm{ub}} \cdot (1+\epsilon_R), \quad V^{X_u}_{\mathrm{ub}} = V^{L}_{\mathrm{ub}} \cdot \sqrt{1+\epsilon_R^2}, \quad V^{\tau}_{\mathrm{ub}} = V^{L}_{\mathrm{ub}} \cdot (1-\epsilon_R)$$

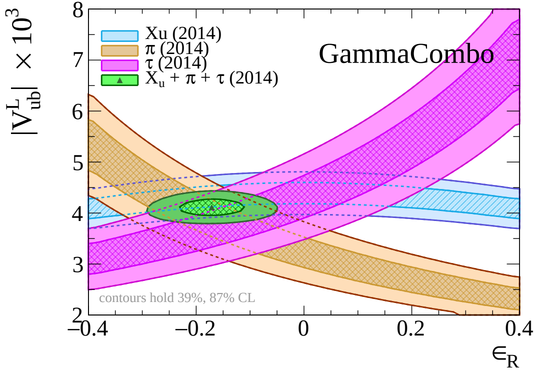


Figure: reproduced in GAMMACOMBO

Are there right handed currents in V_{ub}?

$$V_{\mathrm{ub}}^{\pi} = V_{\mathrm{ub}}^{L} \cdot (1+\epsilon_R), \quad V_{\mathrm{ub}}^{X_u} = V_{\mathrm{ub}}^{L} \cdot \sqrt{1+\epsilon_R^2}, \quad V_{\mathrm{ub}}^{\tau} = V_{\mathrm{ub}}^{L} \cdot (1-\epsilon_R)$$

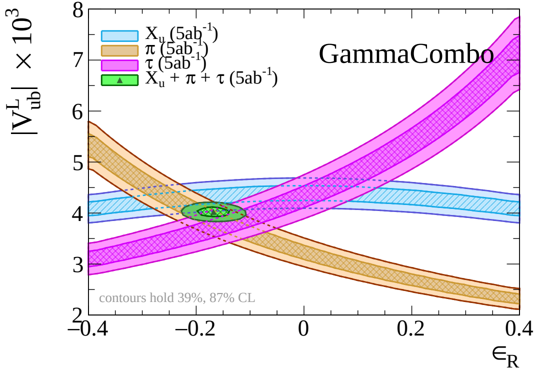


Figure: extrapolation to $5\,\mathrm{ab}^{-1}$ from Belle2

Outline

PLUGIN method

Running the pseudo-experiment based PLUGIN method consists of two steps:

1. generate and fit the pseudo experiments, possibly on a batch farm;
2. read in the fit results and compute the CL intervals.

```
# Run three jobs to generate and fit toy experiments, possibly on
# different cores or batch nodes:
bin/tutorial -c 7 --var a_gaus -a pluginbatch --ntoys 100 --nrun
    1
bin/tutorial -c 7 --var a_gaus -a pluginbatch --ntoys 200 --nrun
    2
# Compute the plugin intervals from all toys:
bin/tutorial -c 7 --var a_gaus -a plugin -j 1-3 -i
```

Plugin method

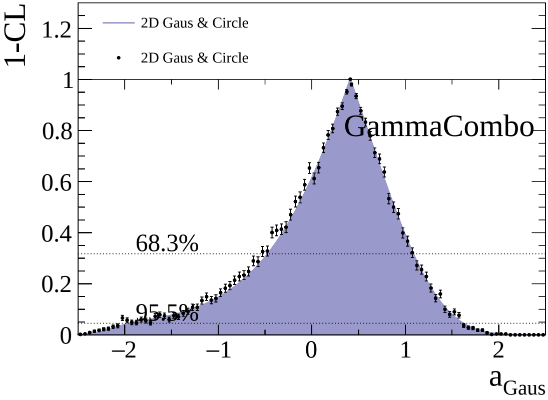


Figure: Result of running the Plugin method with combination number 7: a 2D Gaussian with a circular constraint. The points are the result from the toys, while the filled curve corresponds to the Prob method. In this case, there is good agreement between both.

PLUGIN method

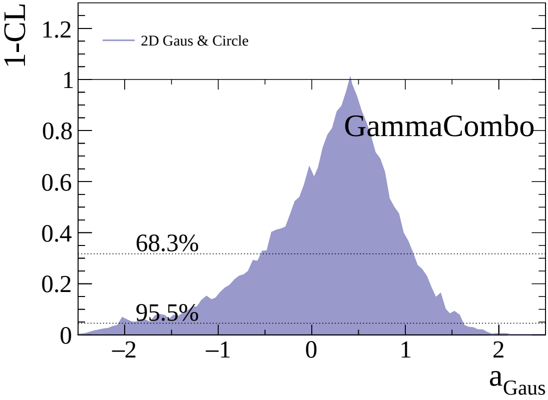


Figure: Result of running the PLUGIN method with combination number 7: a 2D Gaussian with a circular constraint. Plotting the result of the PLUGIN as a filled curve (–po), without the profile likelihood curve.

PLUGIN method: The standard Feldman-Cousins example

Lets reproduce the standard FC example: A Gaussian $G(\mu = -0.5, \sigma = 1)$, with a physical boundary at $\mu > -1.5$.

The option $-\mathrm{pr}$ will keep the parameter in its defined physical range.

```
# run plugin toys for combination 1 (simple Gaussian), with physical
    boundary
$ bin/tutorial -c 1 --var a_gaus -a pluginbatch --ntoys 400 --pr
    --scanrange -1.5:2.5 --nrun 1
# compute the plugin intervals from all toys:
$ bin/tutorial -c 1 --var a_gaus -a plugin -j 1 --pr --scanrange
    -1.5:2.5 -i

a_gaus = [ -1.24,   0.51] ( -0.50 -  0.74 +  1.01) @0.68CL, Plugin
a_gaus = [  -1.5,    1.4] (  -0.5 -   1.0 +   1.9) @0.95CL, Plugin
```

PLUGIN method: The standard Feldman-Cousins example

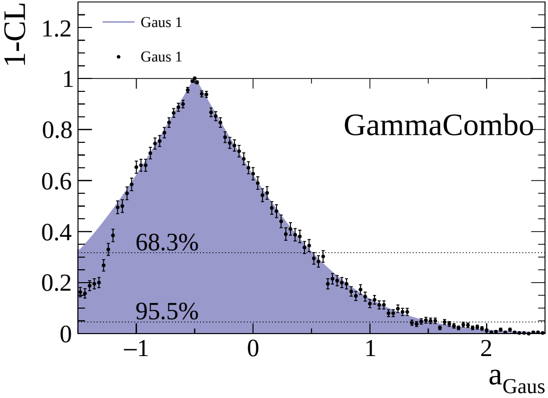
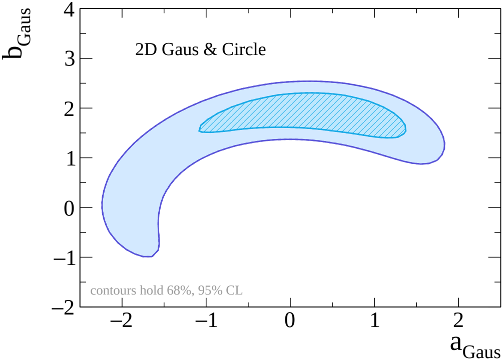


Figure: Applying a physical range to combination 1 and running the PLUGIN method. In this case the PLUGIN method is equivalent to the method by Feldman-Cousins. The difference to the profile likelihood method is most apparent close to the physical boundary at $a = -1.5$.

PLUGIN method: in 2D

The PLUGIN method can also used in two dimensions.



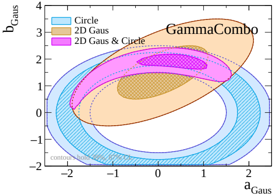
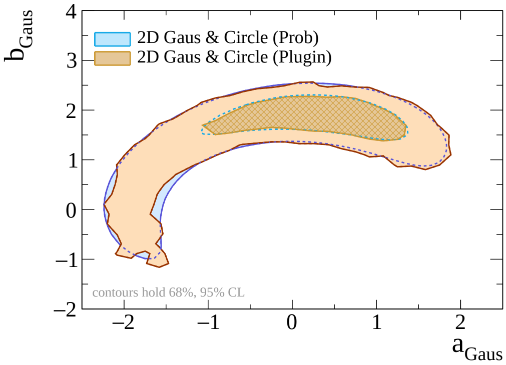


Figure: Profile likelihood contours for combination number 7: a 2D Gaussian with a circular constraint.

PLUGIN method: in 2D

The PLUGIN method can also used in two dimensions.



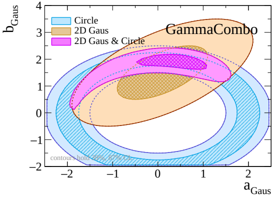


Figure: Profile likelihood contours for combination number 7: a 2D Gaussian with a circular constraint. Adding the PLUGIN contour.

Outline

Interfacing custom code

- can interface custom C++ theory relations (external computations, numerical integrations, etc.)

```
void PDF_Gaus::initRelations()
{
  // can use anything derived from RooAbsReal!
  theory = new RooArgList("theory");
  theory->add(*(new RooYourCustomClass("yourpar_th",
    "yourpar_th", yourTheoryPar1, yourTheoryPar2)));
}
```

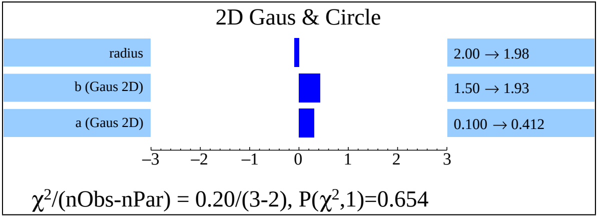
- can use different PDFs (e.g. Rfit, histograms, ...):

```
void PDF_Gaus::buildPdf()
{
  // can use anything derived from RooAbsPdfs!
  pdf = new RooYourCustomPdf("pdf_"+name, "pdf_"+name,
       *(RooArgSet*)observables, *(RooArgSet*)theory);
}
```

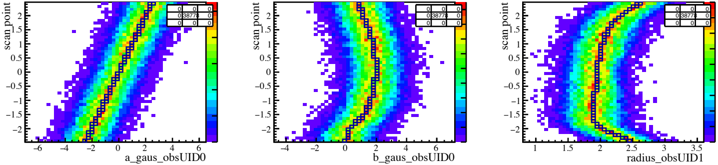
Outline

Control plots: two examples

- make a pull plot:
  `bin/tutorial -c 7 --var a_gaus --pulls`



- look at the generated PLUGIN toys



Control plots: combination anatomy

- GammaCombo writes out `.dot` files to be used with `graphviz`

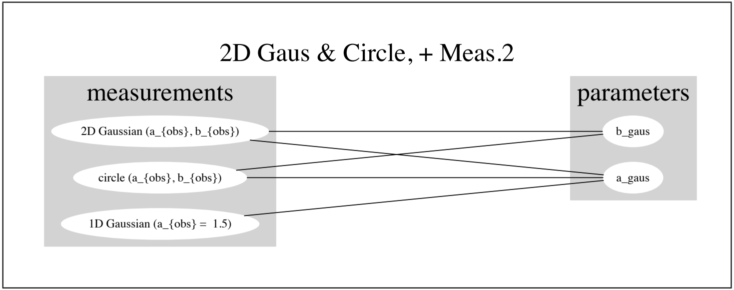


Figure: Anatomy of a combination: How are measurements linked to parameters?

Outline

A possible scenario

- We take a list of key observables and their estimated uncertainties for $5\,\mathrm{ab}^{-1}$ and $50\,\mathrm{ab}^{-1}$ of data, possibly also their correlations
- We provide ready-made measurement classes
- The user modifies the classes to add his/her theory relation
- GAMMACOMBO does the fits/plots/etc.

Semileptonic & Leptonic WG page	$V_{cb}[10^{-3}]$ inclusive	$B \to X_c \ell \nu$		$41.6(1 \pm 0.024_{fit})$		1.2%	
	$V_{cb}[10^{-3}]$ exclusive	$B \to D^* \ell \nu$		$37.5(1 \pm 0.030_{exp} \pm 0.027_{thy})$		1.8%	1.4%
	$V_{ub}[10^{-3}]$ inclusive	$B \to X_u \ell \nu$		$4.47(1 \pm 0.060_{exp} \pm 0.025_{thy})$		3.4%	3.0%
	$V_{ub}[10^{-3}]$ exclusive	$B \to \pi \ell \nu$ (Hadronic tag)		$3.52(1 \pm 0.95_{fit})$		4.4%	2.3%
	$\mathcal{B}[10^{-6}]$	$B \to \tau \nu$ (Hadronic tag)		$96(1 \pm 0.26)$		10%	5%
	$\mathcal{B}[10^{-6}]$	$B \to \mu \nu$				20%	7%
	$\mathcal{R}$	$B \to D \tau \nu$ (Hadronic tag)			$0.440(1 \pm 0.165)$	5.6%	3.4%
	$\mathcal{R}$	$B \to D^* \tau \nu$ (Hadronic tag)			$0.332(1 \pm 0.090)$	4.4%	2.3%

Figure:
`https://belle2.cc.kek.jp/~twiki/bin/view/Public/B2TIPGoldenModes`

Outline

A real-life example: LHCb γ

GammaCombo was developed for the combination of all LHCb measurements with tree-level sensitivity to γ/ϕ_3.

- 23 parameters for public results, internally tested up to 29
- used histogram PDFs
- tricky likelihood with several local minima
- ran coverage test

Tue following few slides I have shown at the CKM2014 conference in Vienna.

LHCb-CONF-2014-004

CKM2014: Result of the robust combination

Table: Credibility regions and most probable values for the hadronic parameters extracted from the robust combination. The second part of the table repeats the frequentist results for comparison.

Observable	Central value	Intervals	
Frequentist		68%	95%
$\gamma[^\circ]$	72.9	$[63.0, 82.1]$	$[52.0, 90.5]$
$\delta_B^{DK}[^\circ]$	126.8	$[115.3, 136.7]$	$[101.6, 145.2]$
r_B^{DK}	0.0914	$[0.0826, 0.0997]$	$[0.0728, 0.1078]$
Bayesian		68%	95%
$\gamma[^\circ]$	71.9	$[61.9, 81.8]$	$[50.9, 91]$
$\delta_B^{DK}[^\circ]$	127.4	$[115.6, 137.3]$	$[100.3, 147.1]$
r_B^{DK}	0.091	$[0.0826, 0.0984]$	$[0.0740, 0.106]$

The agreement of frequentist and Bayesian results is quite good.

LHCb-CONF-2014-004

CKM2014: Result of the robust combination

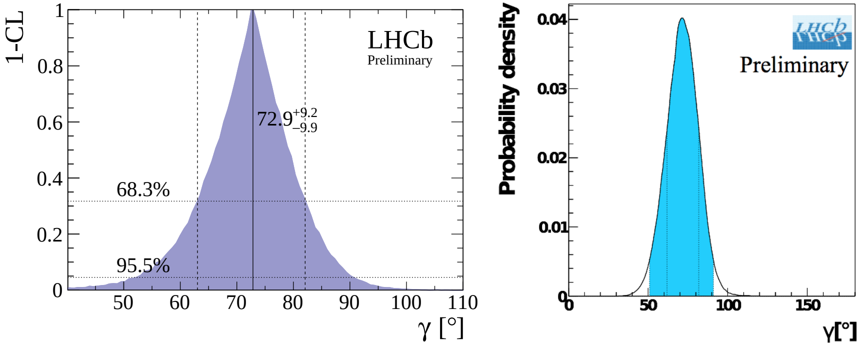


Figure: Graphs for the robust combination (left: frequentist $1 - \mathrm{CL}$ curves, right: Bayesian posterior).

CKM2014: Result of the full combination

LHCb-CONF-2014-004

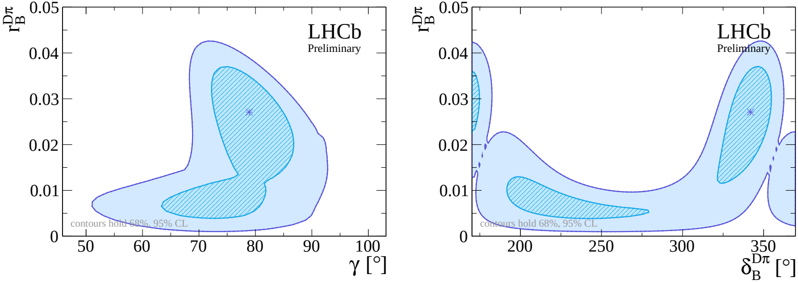


Figure: Profile likelihood contours, full combination. The contours are the two-dimensional 1σ and 2σ contours.

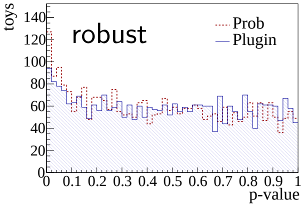
The high observed value of $r_B^{D\pi}$ also pulls up γ.

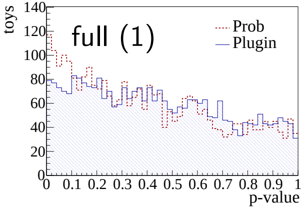
CKM2014: Coverage test

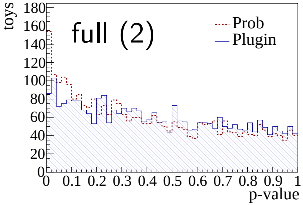
- We test the frequentist coverage at the minima of the combinations.
- We find that the profile likelihood construction undercovers quite a bit.
- The robust plugin method has good coverage.
- The coverage of the full combination is worse than of the robust. Expected due to the low value of $r_B^{D\pi}$.

$\eta = 0.683$	α (prof. LH.)	α (plugin)
robust	0.6158	0.6494
full (1), $r_B^{D\pi} = 0.027$	0.5593	0.6154
full (2), $r_B^{D\pi} = 0.006$	0.5454	0.6120

LHCb-CONF-2014-004







Outline

Summary

GammaCombo is still young, but already very useful for many areas—so far only within LHCb.

Try it out!

```
git clone https://github.com/mkarbach/gammacombo.git
cd gammacombo
mkdir build; cd build
cmake ..
make install -j4
cd ../tutorial
```

Thanks to: Max Schlupp, Matt Kenzie, Florian Bernlochner

BaBar Collaboration, J. Lees *et al.*, *Measurement of an Excess of $\bar{B} \to D^{(*)}\tau^-\bar{\nu}_\tau$ Decays and Implications for Charged Higgs Bosons*, Phys. Rev. **D88** (2013), no. 7 072012, `arXiv:1303.0571`.

BaBar Collaboration, J. Lees *et al.*, *Evidence for an excess of $\bar{B} \to D^{(*)}\tau^-\bar{\nu}_\tau$ decays*, Phys. Rev. Lett. **109** (2012) 101802, `arXiv:1205.5442`.

F. U. Bernlochner, Z. Ligeti, and S. Turczyk, *New ways to search for right-handed current in $B \to \rho l \bar{\nu}$ decay*, Phys. Rev. **D90** (2014), no. 9 094003, `arXiv:1408.2516`.