An introduction to GAMBIT

or

the Future of Global Fits

to New Physics

Anders Kvellestad, University of Oslo
• Lots of models for TeV scale
New Physics
• Lots of models for TeV scale New Physics

• For each model, a parameter space of varying phenomenology
What New Physics scenario is preferred?
What New Physics scenario is preferred?

Compare to data!
What New Physics scenario is preferred?

Compare to data!

...But how?
Only a couple of parameters:

Compare preferred/excluded regions for different analyses
Only a couple of parameters:

Compare preferred/excluded regions for different analyses

Allanach, Kvellestad, Raklev: 1504.02752
If many different searches:

Combine searches in a total likelihood function
If many different searches:

Combine searches in a total likelihood function

Allanach, Kvellestad, Raklev: 1504.02752
Many parameters and many searches:

Perform a statistical fit to all available data - a global fit
Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

\[ \mathcal{L} = \mathcal{L}_{\text{Collider}} + \mathcal{L}_{\text{Higgs}} + \mathcal{L}_{\text{DM}} + \mathcal{L}_{\text{EWPO}} + \mathcal{L}_{\text{Flavor}} \]
Many parameters and many searches:

Perform a statistical fit to all available data - a global fit

• Explore likelihood across entire parameter space (smart sampling)
Many parameters and many searches:

Perform a statistical fit to all available data - a **global fit**

- Explore likelihood across entire parameter space (smart sampling)
- Interpretation: Bayesian/frequentist
Many parameters and many searches:

Perform a statistical fit to all available data - a **global fit**

- Explore likelihood across entire parameter space (smart sampling)

- Interpretation: Bayesian/frequentist

- Project down to 1 or 2 parameters (profile/marginalise)
Many parameters and many searches:

Perform a statistical fit to all available data - a **global fit**

- Explore likelihood across entire parameter space (smart sampling)
- Interpretation: Bayesian/ frequentist
- Project down to 1 or 2 parameters (profile/marginalise)
Global fit goals:
Global fit goals:

- Assume a theory is true: What parameter values are preferred?
Global fit goals:

- Assume a theory is true: What parameter values are preferred?
- Given multiple theories: Which is in best agreement with data?
Current challenges:
Current challenges:

- Many experimental searches (almost) neglected
Current challenges:

- Many experimental searches (almost) neglected
- Available tools limited to only a few theories
Current challenges:

- Many experimental searches (almost) neglected
- Available tools limited to only a few theories
- ...and a limited selection of theory tools
Current challenges:

- Many experimental searches (almost) neglected
- Available tools limited to only a few theories
- ...and a limited selection of theory tools
- ...and a limited set of sampling algorithms
Global And Modular BSM Inference Tool
GAMBIT

• A collaboration of ~30 theorists and experimentalists


• Members from several major particle and astroparticle experiments

  Fermi-LAT, ATLAS, CTA, HESS, LHCb, IceCube, AMS-02, CDMS, DM-ICE, XENON, DARWIN
GAMBIT

- Main design principles: modularity and flexibility
GAMBIT

• Main design principles: modularity and flexibility
• Allow user to define the model - not limited to SUSY & friends
GAMBIT

- Main design principles: modularity and flexibility
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
GAMBIT

- Main design principles: *modularity* and *flexibility*
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
GAMBIT

- Main design principles: modularity and flexibility
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
- External tools as (interchangeable) plug-ins
Each module presents the rest of GAMBIT with a set of requirements and a set of capabilities.
The core solves the dependency tree and decides the evaluation order.
GAMBIT

• Main design principles: **modularity** and **flexibility**
• Allow user to define the model - not limited to SUSY & friends
• Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
• Physics modules: Large set of observables/likelihoods (all replaceable)
• External tools as (interchangeable) plug-ins
Anders Kvellestad

GAMBIT

• Main design principles: modularity and flexibility
• Allow user to define the model - not limited to SUSY & friends
• Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
• Physics modules: Large set of observables/likelihoods (all replaceable)
• External tools as (interchangeable) plug-ins
• C++
GAMBIT

- Main design principles: modularity and flexibility
- Allow user to define the model - not limited to SUSY & friends
- Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
- Physics modules: Large set of observables/likelihoods (all replaceable)
- External tools as (interchangeable) plug-ins
- C++
- Both MPI and openMP parallelisation
GAMBIT

• Main design principles: modularity and flexibility
• Allow user to define the model - not limited to SUSY & friends
• Many statistical options: Bayesian/frequentist, scanning algorithms as plug-ins
• Physics modules: Large set of observables/likelihoods (all replaceable)
• External tools as (interchangeable) plug-ins

• C++
• Both MPI and openMP parallelisation
• Open-source release (in a few months)
<table>
<thead>
<tr>
<th>Aspect</th>
<th>GAMBIT</th>
<th>MasterCode</th>
<th>SuperBayes</th>
<th>Fittino</th>
<th>Rizzo et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Modular, Adaptive Frequency, Bayesian</td>
<td>Monolithic</td>
<td>Monolithic</td>
<td>(~)Monolithic</td>
<td>Monolithic</td>
</tr>
<tr>
<td>Statistics</td>
<td>Differential evolution, genetic algorithms, random forests, t-walk, t-nest, particle swarm, nested sampling, MCMC, gradient descent</td>
<td>Frequentist</td>
<td>Frequentist</td>
<td>Frequentist</td>
<td>None</td>
</tr>
<tr>
<td>Scanners</td>
<td>(p)MSSM-25, CMSSM±ε, GMSB, AMSB, gaugino mediation, E6MSSM, NMSSM, BMSSM, PQMSSM, effective operators, iDM, XDM, ADM, UED, Higgs portals/extended Higgs sectors</td>
<td>Nested sampling, MCMC, grad. descent</td>
<td>Nested sampling, MCMC</td>
<td>Nested sampling, MCMC</td>
<td>None</td>
</tr>
<tr>
<td>Theories</td>
<td>CMSSM±ε</td>
<td>(p)MSSM-15, CMSSM±ε, mUED</td>
<td>CMSSM±ε</td>
<td>(p)MSSM-19</td>
<td>None (random)</td>
</tr>
<tr>
<td>LHC</td>
<td>ATLAS+CMS multi-analysis with neural net and fast detector simulation. Higgs multi-channel with correlations and no SM assumptions. Full flavour inc. complete B → X_sτ and B → K^*τ angular set.</td>
<td>ATLAS resim, HiggsSignals, basic flavour.</td>
<td>ATLAS direct sim, Higgs mass only, basic flavour.</td>
<td>ATLAS resim, HiggsSignals, basic flavour.</td>
<td>ATLAS+CMS +Tevatron direct sim, basic flavour.</td>
</tr>
<tr>
<td>SM, theory and related uncerts.</td>
<td>m_t, m_b, α_s, α_{EM}, DM halo, hadronic matrix elements, detector responses, QCD+EW corrections (LHC+DM signal+BG), astro BGs, cosmic ray hadronisation, coalescence and p'gation.</td>
<td>m_t, m_b, α_s, α_{EM}, DM halo, hadronic matrix elements.</td>
<td>m_t, m_b, α_s, α_{EM}, DM halo, hadronic matrix elements.</td>
<td>m_t</td>
<td>None</td>
</tr>
</tbody>
</table>

Borrowed from P. Scott
Thank you for your attention!
Backup slides
<table>
<thead>
<tr>
<th>Backends</th>
<th>Version</th>
<th>Path to lib</th>
<th>Status</th>
<th>#funcs</th>
<th>#types</th>
<th>#ctors</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOSSMinimalExample</td>
<td>1.0</td>
<td>Backends/lib/libminimal_1_0.so</td>
<td>OK</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1.1</td>
<td>Backends/lib/libminimal_1_1.so</td>
<td>OK</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>Backends/lib/libminimal_1_2.so</td>
<td>OK</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DDCalc0</td>
<td>0.0</td>
<td>Backends/lib/libDDCalc0.so</td>
<td>OK</td>
<td>44</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DarkSUSY</td>
<td>5.1.1</td>
<td>../extras/DarkSUSY/lib/libdarksusy.so</td>
<td>OK</td>
<td>44</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FastSim</td>
<td>1.0</td>
<td>Backends/lib/libfastsim.so</td>
<td>absent/broken</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FeynHiggs</td>
<td>2.10</td>
<td>Backends/lib/libfeynhiggs.so</td>
<td>absent/broken</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HiggsBounds</td>
<td>4.1</td>
<td>Backends/lib/libhiggsbounds.so</td>
<td>absent/broken</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HiggsSignals</td>
<td>1.2</td>
<td>Backends/lib/libhiggssignals.so</td>
<td>absent/broken</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LibFarrayTest</td>
<td>1.0</td>
<td>Backends/lib/libFarrayTest.so</td>
<td>OK</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LibFirst</td>
<td>1.0</td>
<td>Backends/lib/libFirst.so</td>
<td>OK</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.1</td>
<td>Backends/lib/libFirst.so</td>
<td>OK</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LibFortran</td>
<td>1.0</td>
<td>Backends/lib/libFortran.so</td>
<td>OK</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MicrOMegas</td>
<td>3.5.5</td>
<td>/no/path/in/config/backend_locations/</td>
<td>absent/broken</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pythia</td>
<td>8.186</td>
<td>Backends/lib/libpythia8.so</td>
<td>OK</td>
<td>0</td>
<td>6</td>
<td>37</td>
</tr>
<tr>
<td>SUSY_HIT</td>
<td>1.4</td>
<td>../..../SUSY-HIT/susyhit.so</td>
<td>OK</td>
<td>51</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SuperIso</td>
<td>3.4</td>
<td>Backends/lib/libsuperiso.so</td>
<td>absent/broken</td>
<td>31</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>gamLike</td>
<td>1.0.0</td>
<td>Backends/lib/libgamlike.so</td>
<td>OK</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>nulike</td>
<td>1.0.0</td>
<td>../extras/nulike/lib/libnulike.so</td>
<td>OK</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
It is **not** meaningful to draw *general* conclusions directly from the number allowed/excluded points...
It is **not** meaningful to draw *general* conclusions directly from the number allowed/excluded points...
Only a couple of parameters:

Compare preferred/excluded regions for different analyses

- Simple to understand (at a qualitative level)
- Per-point interpretation is not straightforward
- Gets worse with increasing number of experimental analyses

Anders Kvellestad
If many different searches:

Combine searches in a total likelihood function

Clear per-point interpretation

...but what if there are many parameters?
Doing genuinely ‘model-independent’ DM pheno

All experimental limits in terms of simplified models: effective WIMP, one annihilation channel, etc

\[\Rightarrow\] need something to apply limits to arbitrary DD couplings and ID decay/annihilation branching fractions

\[\Rightarrow\] must include accurate treatment of experimental effects

Impacts of new unstable particles (e.g. extra Higgs) are hard

\[\Rightarrow\] need to simulate decays ‘on the fly’

Calculating relic densities for general models also challenging

\[\Rightarrow\] want to feed in partial annihilation rates, co-annihilations, resonances, etc (not only set up model in LanHEP)

→ nulike, gamlike, DDcalc, cascade sim → Christoph’s talk
The LHC likelihood monster

Time per point:
$O(\text{minute})$ in \textit{best} cases

Time per point for global fits to converge:
$O(\text{seconds})$ in \textit{worst} cases

Challenge:
About 2 orders of magnitude too slow to actually include LHC data in global fits properly

→ More in Martin’s presentation
Parameter space $\rightarrow$ Theory space

CMSSM, MSSM, Simplified Models $\neq$ BSM

Want to do model comparison to actually work out which theory is the best. . .

Challenge:
How do I easily adapt a global fit to different BSM theories?
Basic interface for a scan is a YAML initialisation file

- specify parameters, ranges, priors
- select likelihood components
- select other observables to calculate
- define generic rules for how to fill dependencies
- define generic rules for options to be passed to module functions
- set global options (scanner, errors/warnings, logging behaviour, etc)
Modules

Physics Modules

- **ColliderBit** (Martin’s talk)
- **DarkBit** (Christoph’s talk)
- **FlavBit** – flavour physics inc. $g - 2$, $b \rightarrow s\gamma$, $B$ decays (new channels, theory uncerts, LHCb likelihoods)
- **SpecBit** – generic BSM spectrum object, providing RGE running, masses, mixings, etc via interchangeable interfaces to different RGE codes
- **DecayBit** – decay widths for all relevant SM & BSM particles
- **EWPOBit** – precision tests (mostly by interface to FeynHiggs, alt. SUSY-POPE)

**ScannerBit**: manages statistics, parameter sampling and optimisation algorithms
Other nice technical features

- **Scanners:** MultiNest, Diver (diff. evolution), PIKAIA (genetic algorithms), GreAT (MCMC)
- **Statistics:** Bayesian, Profile Likelihood, later full Neyman
- Mixed-mode **MPI + openMP**, mostly automated
- diskless generalisation of various Les Houches Accords
- **BOSS:** dynamic loading of C++ classes from backends (!)
- **all-in or module standalone** modes – easily implemented from single cmake script
- **automatic getters** for obtaining, configuring + compiling backends

- **flexible output streams** (ASCII, databases, binary, ...)
- more more more...

---

1 if a backend breaks, won’t compile and/or kills your dog, blame the authors (not us... unless we are the authors... )