Global BSM fitting with GAMBIT

AND OTHER MACHINERY FOR LHC RECASTING

Andy Buckley
University of Glasgow

BSM FITS FOR LHC RUN 2,
HIGGS CENTRE FOR THEORETICAL PHYSICS,
UNIVERSITY OF EDINBURGH, 11 OCT 2016

Global fits and GAMBIT

Global BSM fits – combining searches

Question

How do we know which models are in and which are out?

Global BSM fits – combining searches

Question

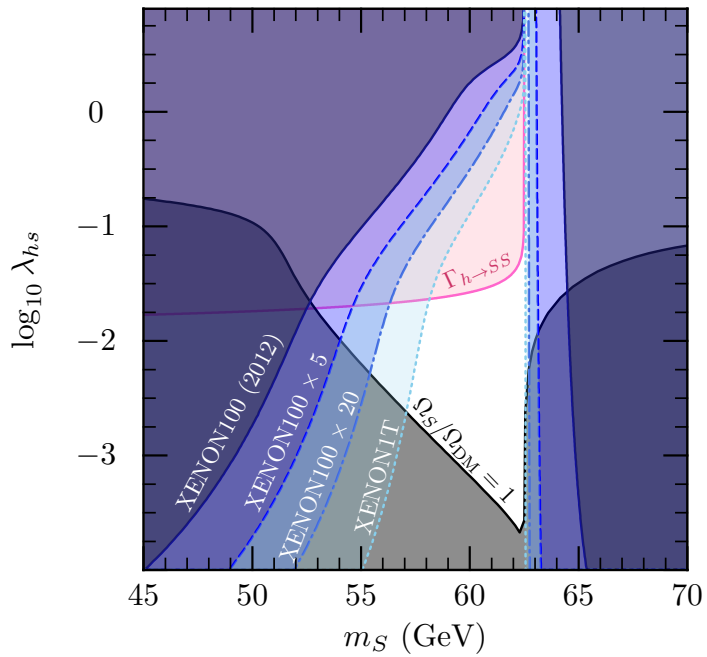
How do we know which models are in and which are out?

Answer

Combine the results from different searches

- Simplest method: take different exclusions, overplot them, conclude things are "allowed" or "excluded"
- Simplest BSM example: the scalar singlet model

$$\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hs}}{2} S^2 H^\dagger H + \dots$$



(Cline, Kainulainen, Scott & Weniger, *PRD*, 1306.4710)

Global BSM fits – combining searches (2)

That's fine if there are only 2 parameters and a few searches...

Question

What if there are many different constraints?

Global BSM fits – combining searches (2)

That's fine if there are only 2 parameters and a few searches...

Question

What if there are many different constraints?

Answer

Statistically valid constraint combination $\Rightarrow$ composite likelihood

Global BSM fits – combining searches (2)

That's fine if there are only 2 parameters and a few searches...

Question

What if there are many different constraints?

Answer

Statistically valid constraint combination $\Rightarrow$ composite likelihood

Question

What if there are many parameters?

Global BSM fits – combining searches (2)

That's fine if there are only 2 parameters and a few searches...

Question

What if there are many different constraints?

Answer

Statistically valid constraint combination $\Rightarrow$ composite likelihood

Question

What if there are many parameters?

Answer

- scan the parameter space (smart numerics)
- interpret the combined results (Bayesian / frequentist)
- project on to parameter planes of interest (marginalise / profile)

$\Rightarrow$ global fits

Scanning BSM model space

Goals:

1. Given multiple theories, determine which fit the data better, and quantify how much better
2. Given a particular theory, determine which parameter combinations fit all experiments, and how well

Scanning BSM model space

Goals:

1. Given multiple theories, determine which fit the data better, and quantify how much better $\Longrightarrow$ model comparison
2. Given a particular theory, determine which parameter combinations fit all experiments, and how well $\Longrightarrow$ parameter estimation

Scanning BSM model space

Goals:

1. Given multiple theories, determine which fit the data better, and quantify how much better $\Longrightarrow$ model comparison
2. Given a particular theory, determine which parameter combinations fit all experiments, and how well $\Longrightarrow$ parameter estimation

Why simple "in/out" analyses are not enough:

- Only partial goodness of fit, no measure of convergence, no idea how to generalise to regions or whole space;
- Frequency/density of models in in/out scans is not proportional to probability $\Rightarrow$ no statistical meaning;
- $\Rightarrow$ statements about a theory's general ability to do one thing or another, based on such scans, are statistically invalid.

A statistically robust, convergent scan needs *feedback* in the parameter space exploration $\Rightarrow$ adaptive sampling

Global fits for dark matter and new physics

Current global fit codes are hardcoded to deal with only a few

- theories (MSSM and/or mSUGRA+friends)
- theory calculators (often interfaced in a very ad hoc way)
- datasets and observables (often missing detailed likelihoods)
- scanning algorithms and statistical methods (generally just one)

$\Rightarrow$ *hitting the wall on theories, data & computational methods*

Global fits for dark matter and new physics

Current global fit codes are hardcoded to deal with only a few

- theories (MSSM and/or mSUGRA+friends)
- theory calculators (often interfaced in a very ad hoc way)
- datasets and observables (often missing detailed likelihoods)
- scanning algorithms and statistical methods (generally just one)

$\Rightarrow$ *hitting the wall on theories, data & computational methods*

GAMBIT: a *second-generation* global fit code

GAMBIT: the Global And Modular BSM Inference Tool

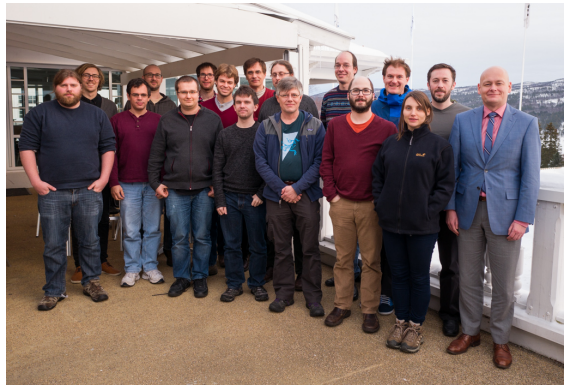
Key principles/aims: flexibility, modularity, generality

- General enough to allow fast definition of new datasets and theoretical models
- Plug and play scanning, physics and likelihood packages
- Extensive model database – not just small modifications to constrained MSSM (NUHM, etc), and not just SUSY!
- Extensive observable/data libraries (likelihood modules)
- Many statistical options – Bayesian/frequentist, likelihood definitions, scanning algorithms
- A smart and *fast* LHC likelihood calculator
- Massively parallel
- Full open-source code release (along with first papers)

The GAMBIT collaboration

30 members, 10 countries, 9 experiments,
4 major theory codes

ATLAS	A. Buckley, P. Jackson, C. Rogan, M. White,
LHCb	M. Chrząszcz, N. Serra
Belle-II	F. Bernlochner, P. Jackson
Fermi-LAT	J. Conrad, J. Edsjö, G. Martinez P. Scott
CTA	C. Balázs, T. Bringmann, J. Conrad, M. White
HESS	J. Conrad
IceCube	J. Edsjö, P. Scott
XENON/DARWIN	J. Conrad. R. Trotta
Theory	P. Athron, C. Balázs, T. Bringmann, J. Cornell, J. Edsjö, B. Farmer, T. Gonzalo, A. Fowlie, S. Hoof, J. Harz F. Karlhoefer, A. Krislock, A. Kvellestad, M. Pato, F. Mahmoudi, J. McKay, A. Raklev, R. Ruiz, P. Scott, R. Trotta, C. Weniger, M. White, S. Wild



And several years of work so far. Yes, a bit behind schedule!

Results and code this year – currently scanning & bug-finding/fixing

Modules

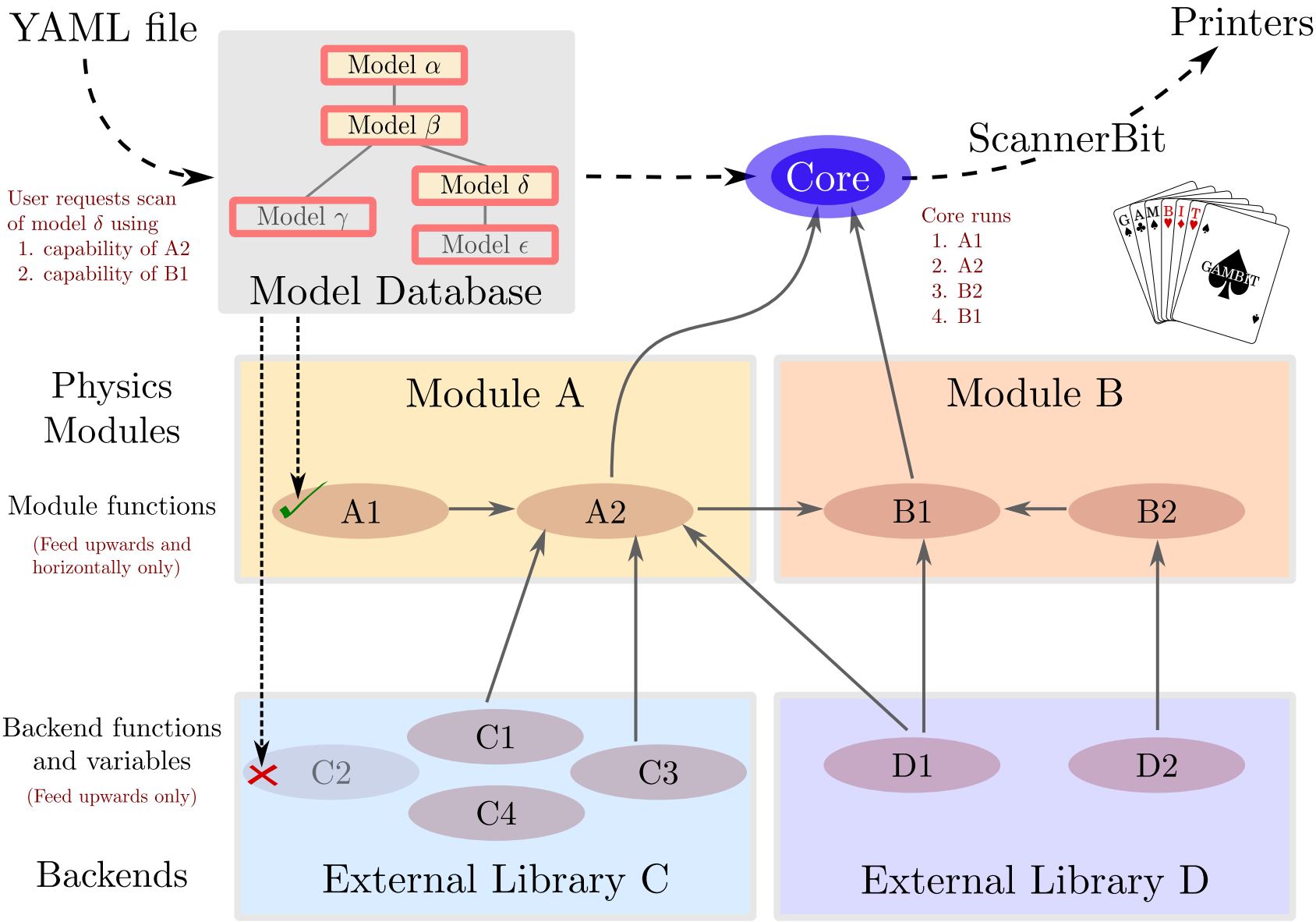
Physics modules

- DarkBit – dark matter observables (relic density, direct + indirect detection)
- ColliderBit – collider observables inc. Higgs + SUSY searches from ATLAS, CMS + LEP
- FlavBit – flavour physics inc. $g-2$, $b \to s\gamma$, B decays (new channels, angular obs., 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
- PrecisionBit – SM likelihoods, precision BSM tests e.g. W mass

Each consists of a number of **module functions** that can have **dependencies** on each other

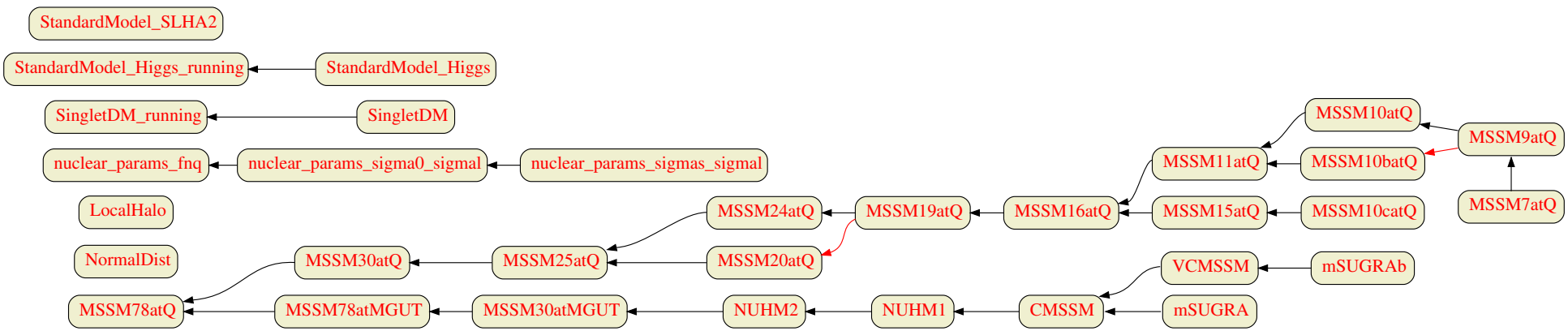
+ ScannerBit – manages stats, sampling and optimisation: DIVER, TWALK, MULTINEST...

Functional overview of GAMBIT



Hierarchical model database

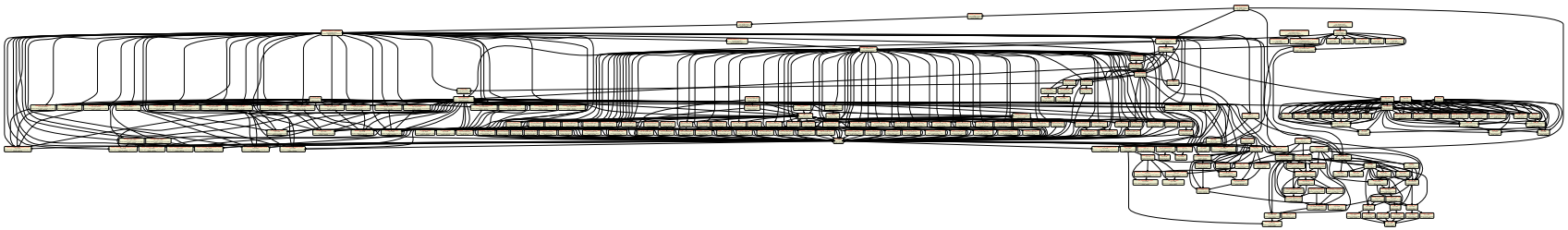
- Models are defined by their parameters and relations to each other
- Models can inherit from **parent models**
- Points in child models can be **automatically translated** to ancestor models
- **Friend models** also allowed (cross-family translation)
- Model dependence of every function/observable is tracked
  $\Longrightarrow$ maximum safety, maximum reuse



Backends: mix and match

- Module functions can require specific functions from **backends**
- Backends are external code libraries (DarkSUSY, FeynHiggs, etc) that include different functions
- GAMBIT automates and abstracts the interfaces to backends $\rightarrow$ backend functions are tagged according to what they calculate
- $\rightarrow$ with appropriate module design, different backends and their functions can be used interchangeably
- GAMBIT dynamically adapts to use whichever backends are actually present on a user's system (+ provides details of what it decided to do of course)

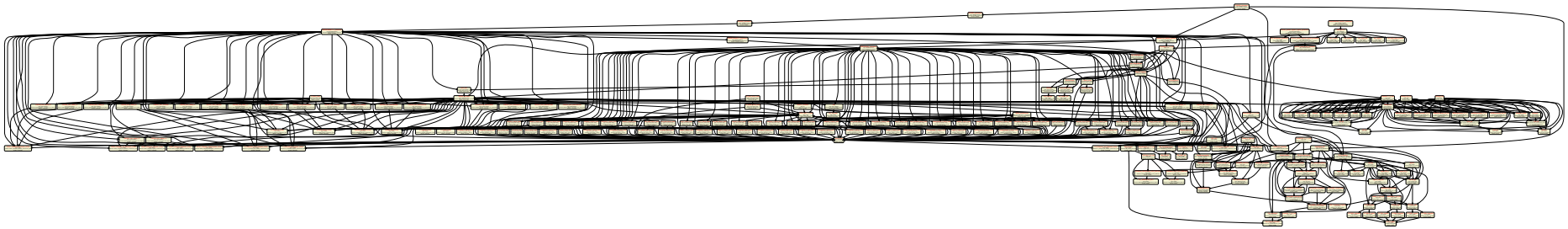
Dependency resolution



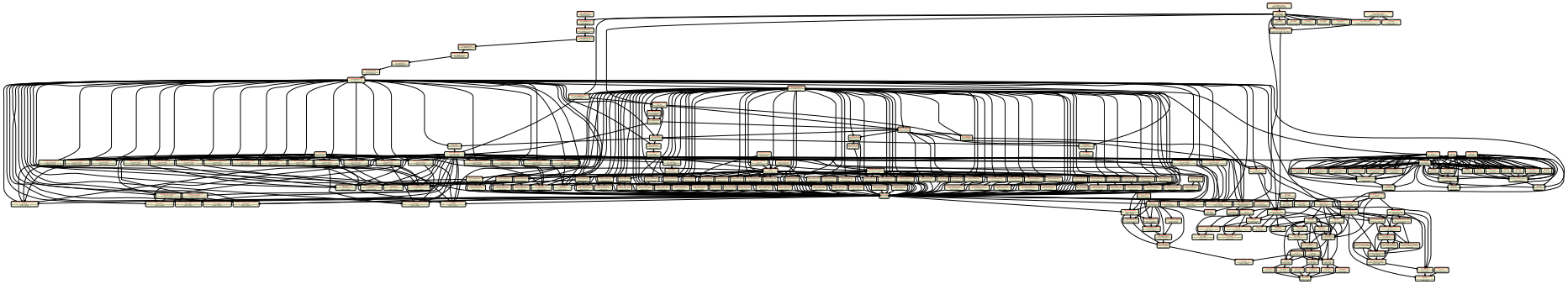
- Module functions and backend functions get arranged into a **dependency tree**
- Starting with requested observables and likelihoods, GAMBIT fills each dependency and backend requirement
- Obeys **rules** at each step: allowed models, allowed backends, constraints from input file, etc
- $\Rightarrow$ tree constitutes a directed acyclic graph
- $\Rightarrow$ GAMBIT uses graph-theoretic methods to 'solve' the graph to determine function evaluation order

Dependency resolution

CMSSM:

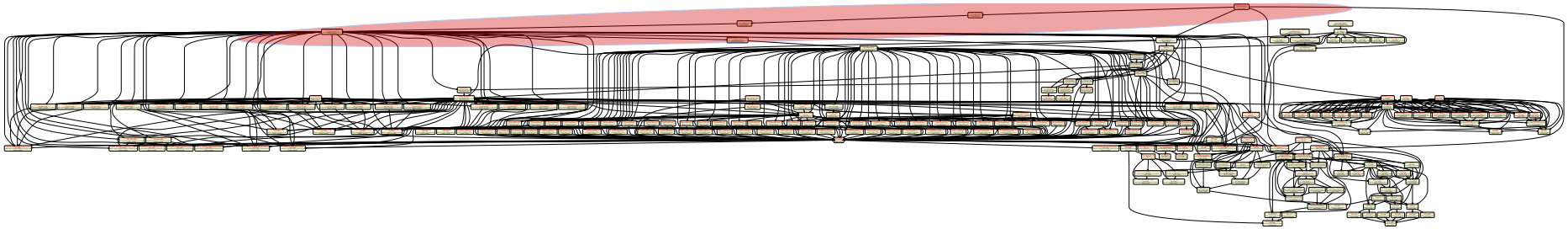


MSSM7:

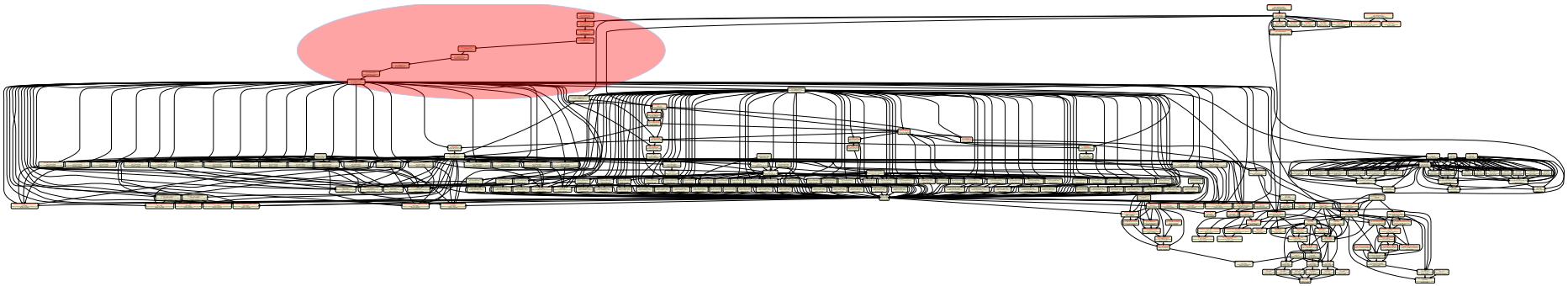


Dependency resolution

CMSSM:



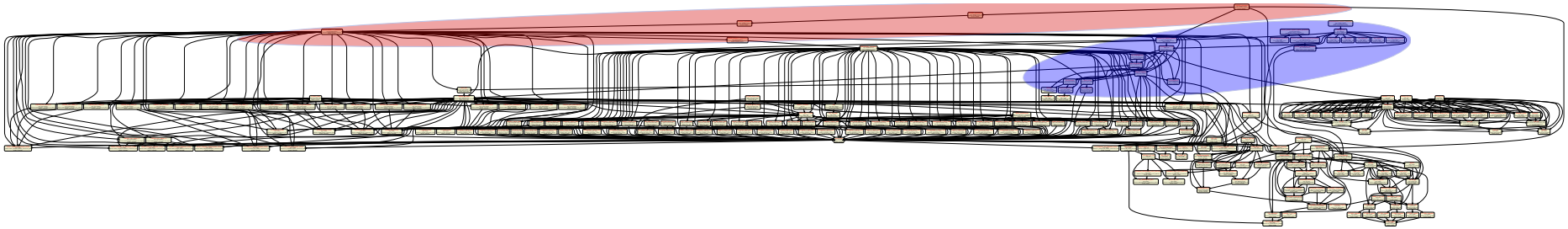
MSSM7:



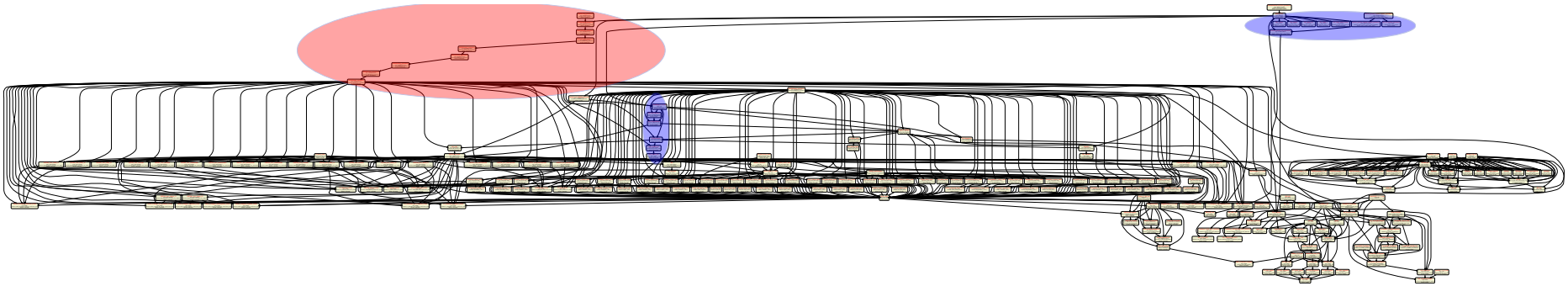
Red: Model parameter translations

Dependency resolution

CMSSM:



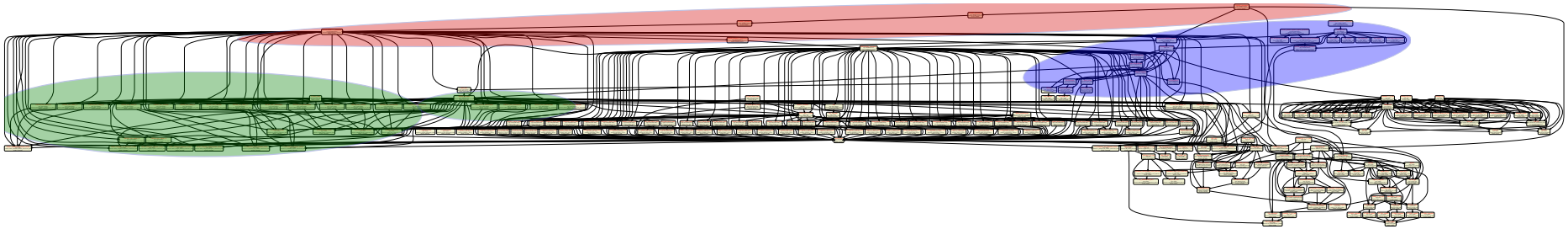
MSSM7:



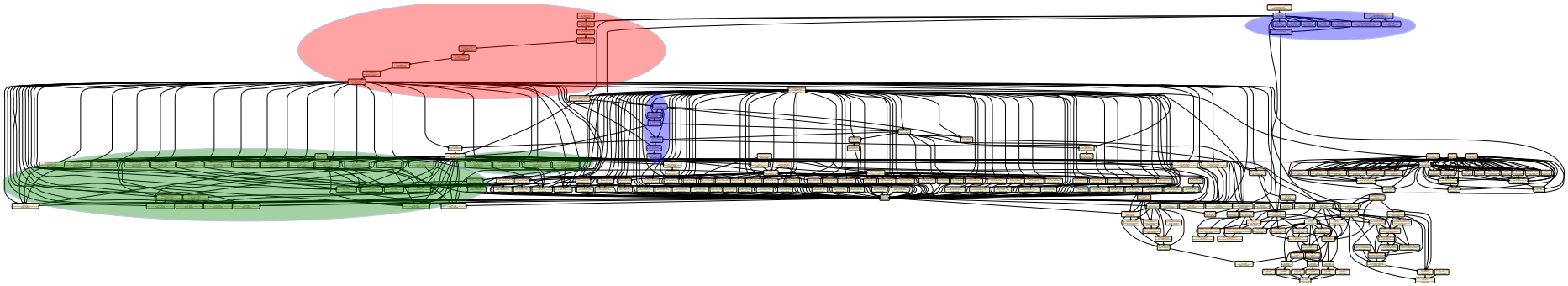
Red: Model parameter translations
Blue: Precision calculations

Dependency resolution

CMSSM:



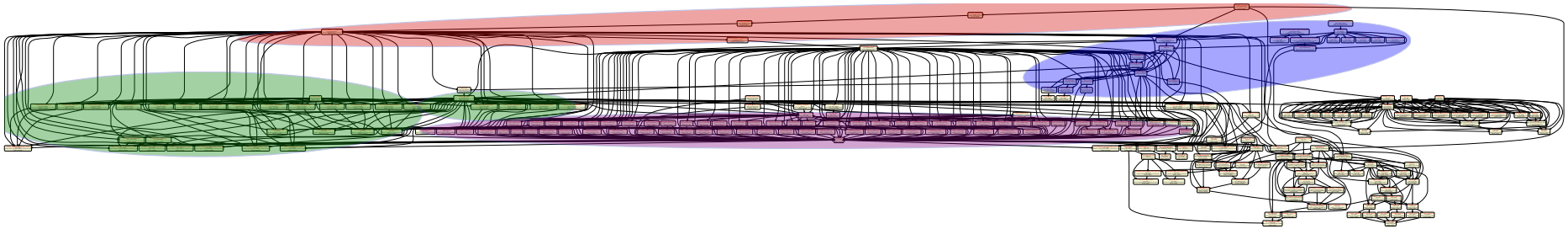
MSSM7:



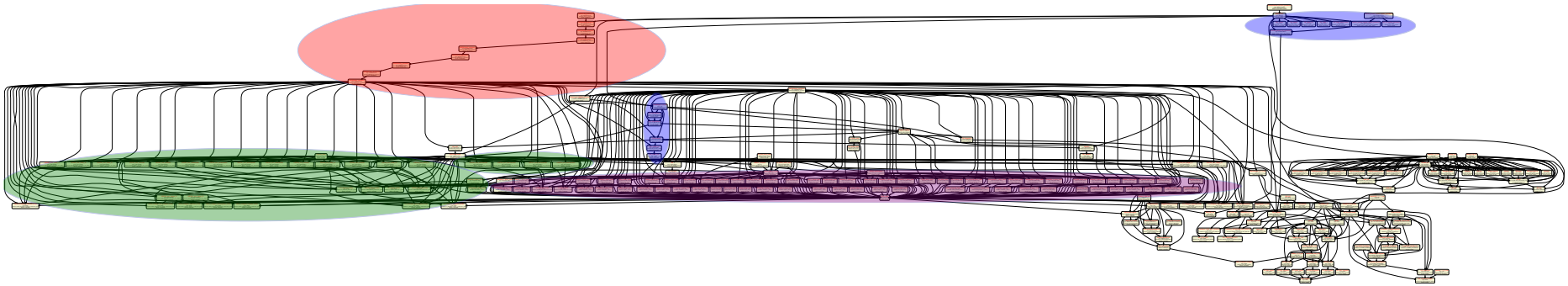
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods

Dependency resolution

CMSSM:



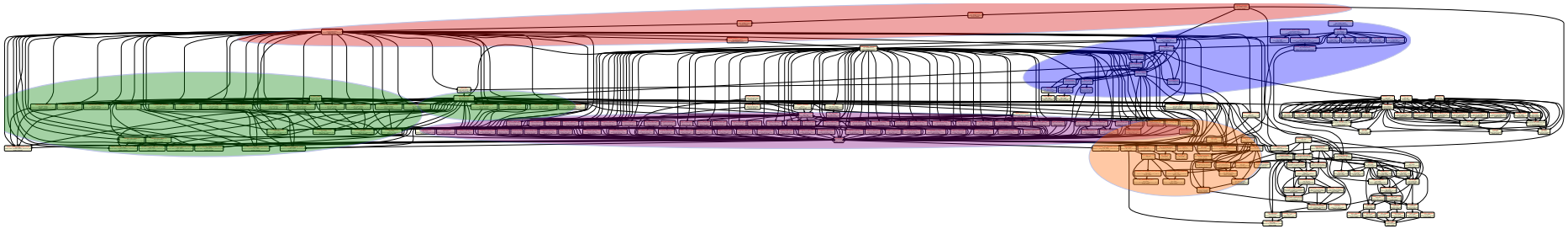
MSSM7:



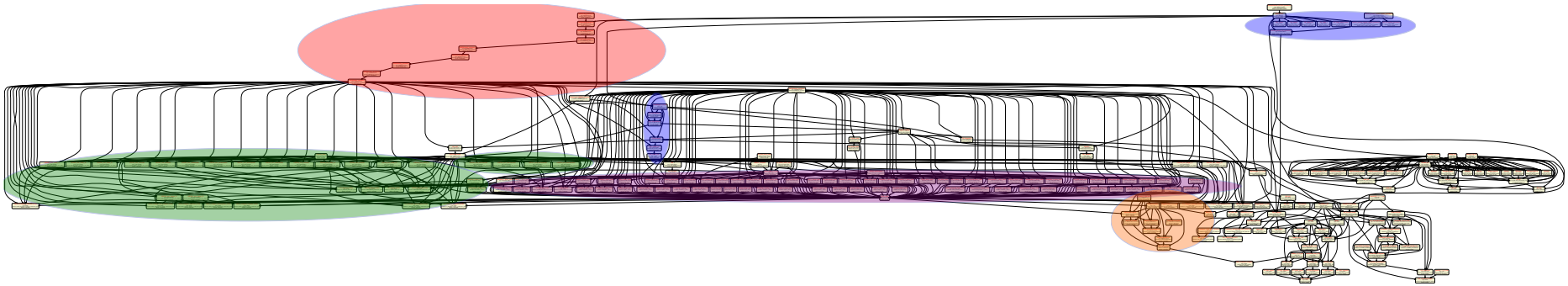
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays

Dependency resolution

CMSSM:



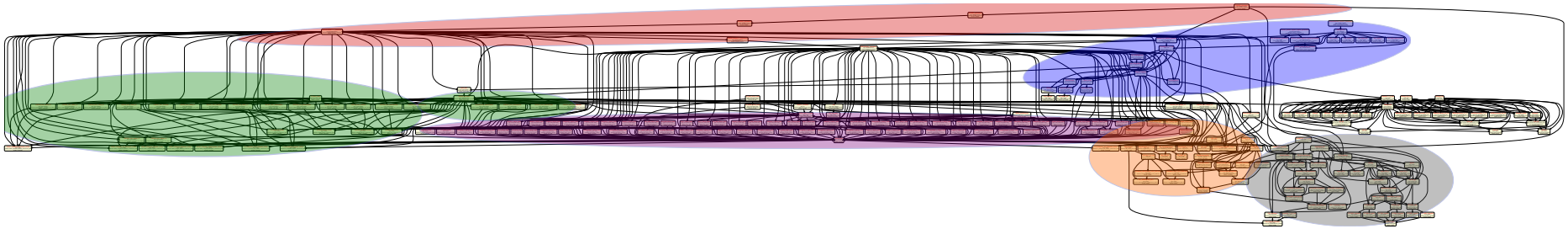
MSSM7:



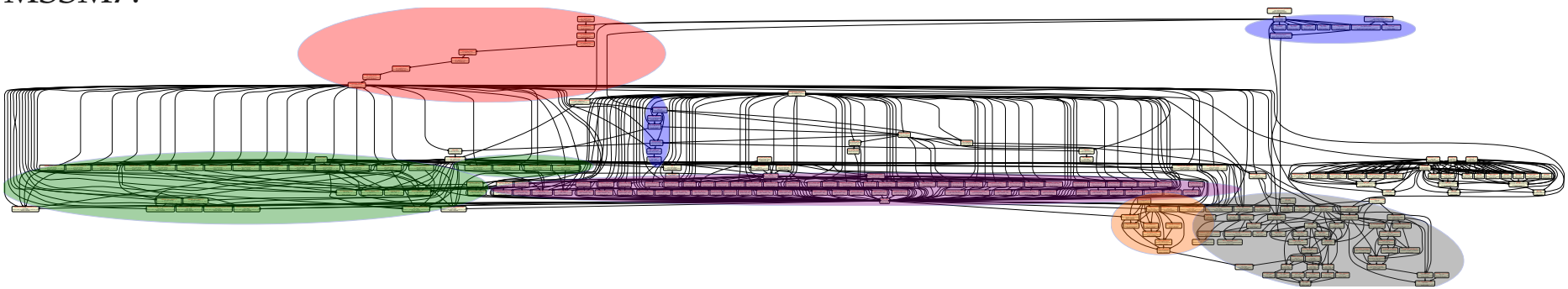
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods

Dependency resolution

CMSSM:



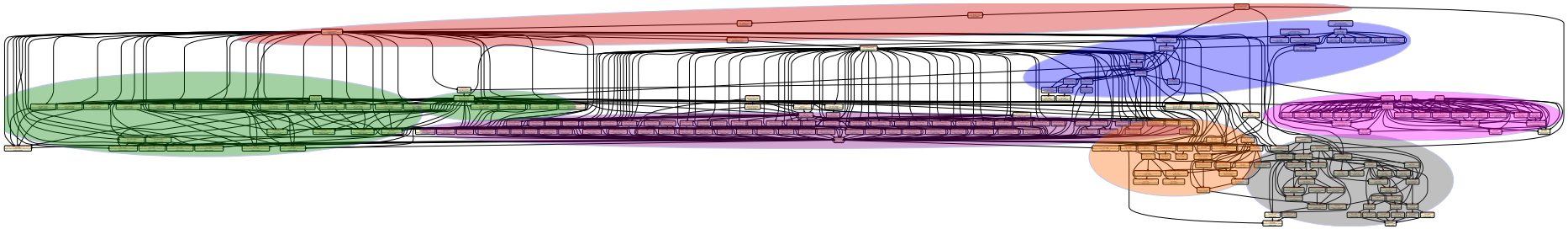
MSSM7:



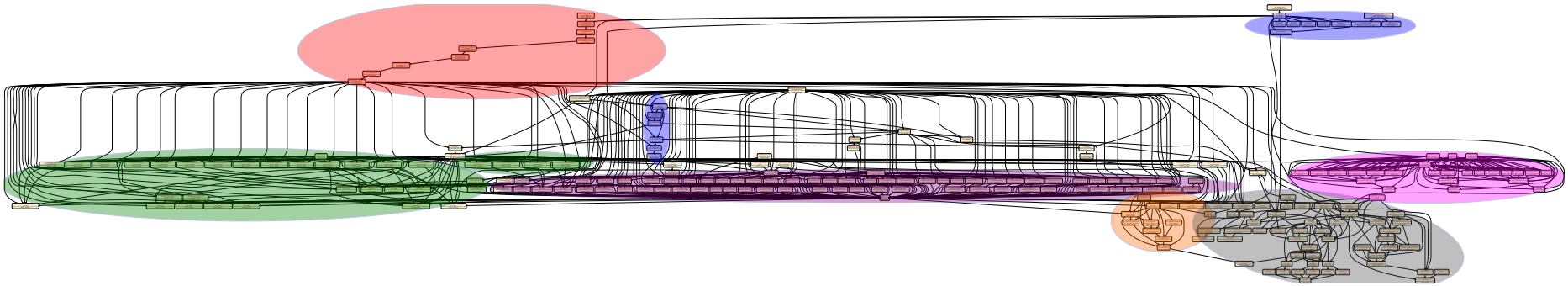
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods
Grey: DM direct, indirect and relic density

Dependency resolution

CMSSM:



MSSM7:



Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods
Grey: DM direct, indirect and relic density
Pink: Flavour physics

Expansion: adding new functions

Adding a new module function is ∼easy:

1. Declare the function to GAMBIT in a module's **rollcall header**
   - Choose a capability
   - Declare any **dependencies**
   - Declare any **backend requirements**
   - Declare any specific **allowed models**
   - Other more advanced declarations also available

```
#define MODULE FlavBit
START_MODULE

  #define CAPABILITY Kmunu_pimunu               // Observable: BR(K->mu nu)/BR(pi->mu nu)
  START_CAPABILITY
    #define FUNCTION SI_Kmunu_pimunu            // Name of specific function providing the observable
    START_FUNCTION(double)                      // Function calculates a double precision variable
      DEPENDENCY(FlavBit_fill, parameters)      // Needs some other function to caluclate FlavBit_fill data
      BACKEND_REQ(Kmunu_pimunu, (libsuperiso), double, (struct parameters*)) // Needs a function from a backend
      BACKEND_OPTION( (SuperIso, 3.4), (libsuperiso) )                       // Backend must be SuperIso v3.4
      ALLOW_MODELS(MSSM78atQ, MSSM78atMGUT)     // Can be used with GUT-scale or other-scale MSSM-78, and all their children
    #undef FUNCTION
  #undef CAPABILITY
```

2. Write the function as a simple C++ function
   (one argument: the result)

ColliderBit – a few details

LEP likelihoods:

- complete model-independent recast of direct sparticle searches

Higgs likelihoods:

- for now: HiggSignals + HiggsBounds + constraints from invisible fits (Bernon, Dumont, Kraml et al)
- future: full simulation and ATLAS+CMS combination, more correlations, no SM-like coupling assumptions

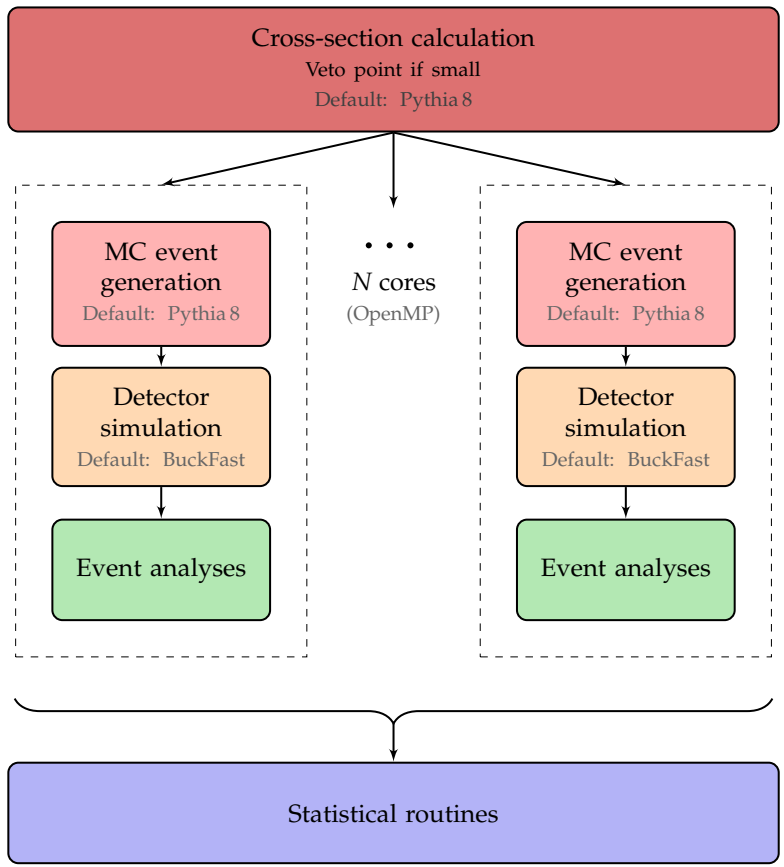
Fast LHC likelihoods:

- event generation is expensive: only simulate if much cheaper exclusion checks have passed
- no simplified models, just faster direct simulation

ColliderBit – more details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks



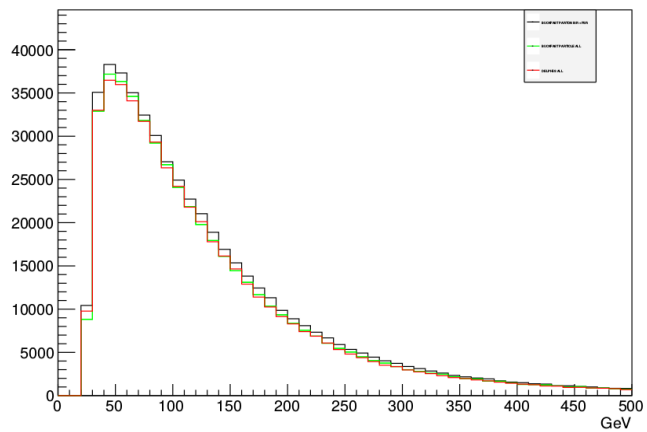
ColliderBit – more details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on physics-object effs + "smearing" ⇒ more later; matches DELPHES results very closely (but faster, both fundamentally & threadable)

Jet energy distribution (a CMSSM example):

black: parton-level analysis with smearing
green: particle-level analysis with smearing
red: detector-level simulation with DELPHES



ColliderBit – more details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on physics-object effs + "smearing"
  $\Rightarrow$ more later; matches DELPHES results very closely (but faster, both fundamentally & threadable)
- Cross-sections: LO + LL from MC generator by default
  (fast NLO in works for SUSY. . . or piggy-back on BSM-AI etc.?)

ColliderBit – more details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on physics-object effs + "smearing"
  $\Rightarrow$ more later; matches DELPHES results very closely (but faster, both fundamentally & threadable)
- Cross-sections: LO + LL from MC generator by default
  (fast NLO in works for SUSY. . . or piggy-back on BSM-AI etc.?)
- Analysis framework: custom event-level, independent of experiment or simulation; tagging, isolation, etc. at analysis level

ColliderBit – more details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on physics-object effs + "smearing"
  $\Rightarrow$ more later; matches DELPHES results very closely (but faster, both fundamentally & threadable)
- Cross-sections: LO + LL from MC generator by default
  (fast NLO in works for SUSY. . . or piggy-back on BSM-AI etc.?)
- Analysis framework: custom event-level, independent of experiment or simulation; tagging, isolation, etc. at analysis level
- Likelihood: inline systematic error marginalisation (via `nulike`)

ColliderBit – more details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on physics-object effs + "smearing"
  $\Rightarrow$ more later; matches DELPHES results very closely (but faster, both fundamentally & threadable)
- Cross-sections: LO + LL from MC generator by default
  (fast NLO in works for SUSY…or piggy-back on BSM-AI etc.?)
- Analysis framework: custom event-level, independent of experiment or simulation; tagging, isolation, etc. at analysis level
- Likelihood: inline systematic error marginalisation (via `nulike`)
- Initially shipping with:
  - ATLAS R1 SUSY searches (0ℓ, $0/1/2\ell$ $\tilde{t}$, b jets + MET, $2/3\ell$ EW)
  - CMS multi-ℓ SUSY
  - CMS DM (t pair + MET, mono-b, monojet)
  - (ATLAS R2 0ℓ, monojet)

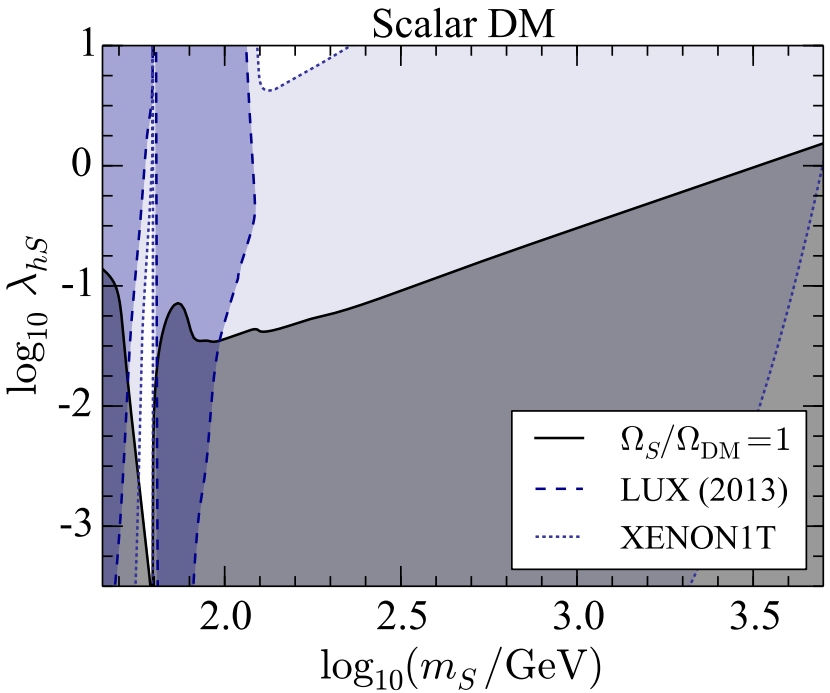
Other nice technical features

- **Scanners**: Nested sampling, differential evolution, MCMC, genetic algorithm, t-walk...
- Mixed-mode **MPI + openMP** parallelisation, mostly automated $\rightarrow$ scales to 10k+ cores
- 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, HDF5, ...)

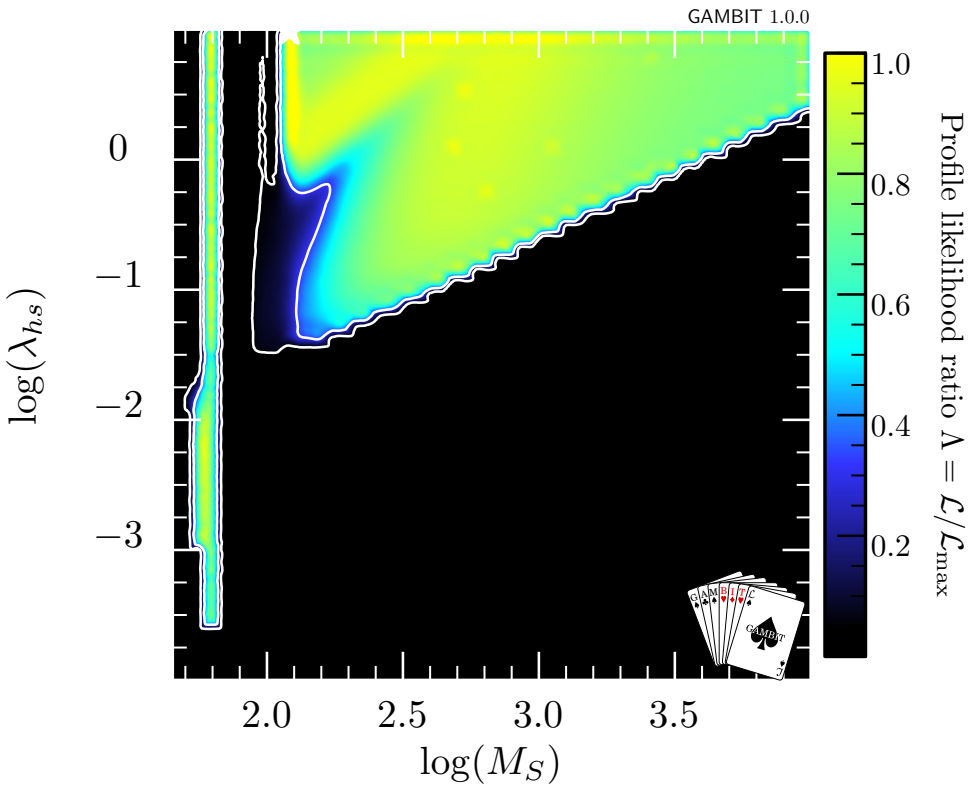
Preliminary results: scalar singlet model

$$\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hs}}{2} S^2 H^\dagger H + \ldots$$

Simple scan



Global fit

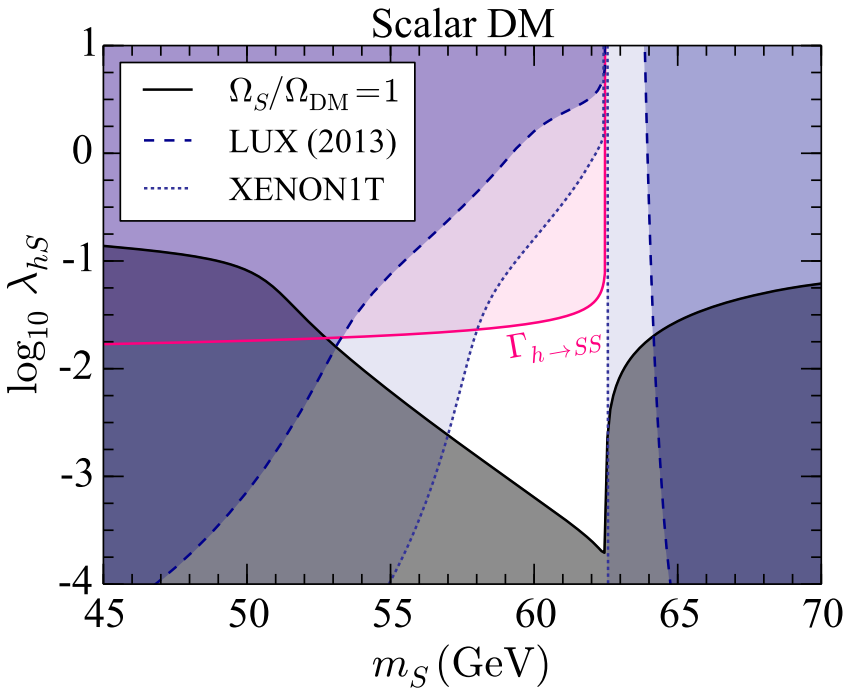


High-mass region

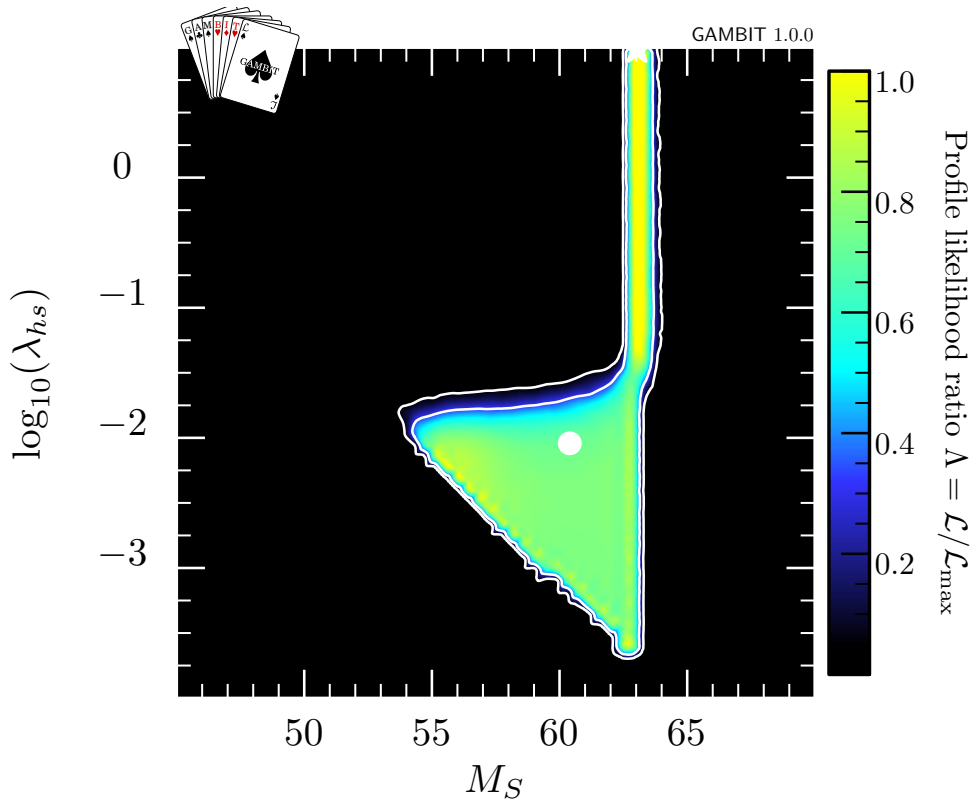
Preliminary results: scalar singlet model

$$\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hs}}{2} S^2 H^\dagger H + \ldots$$

Simple scan



Global fit

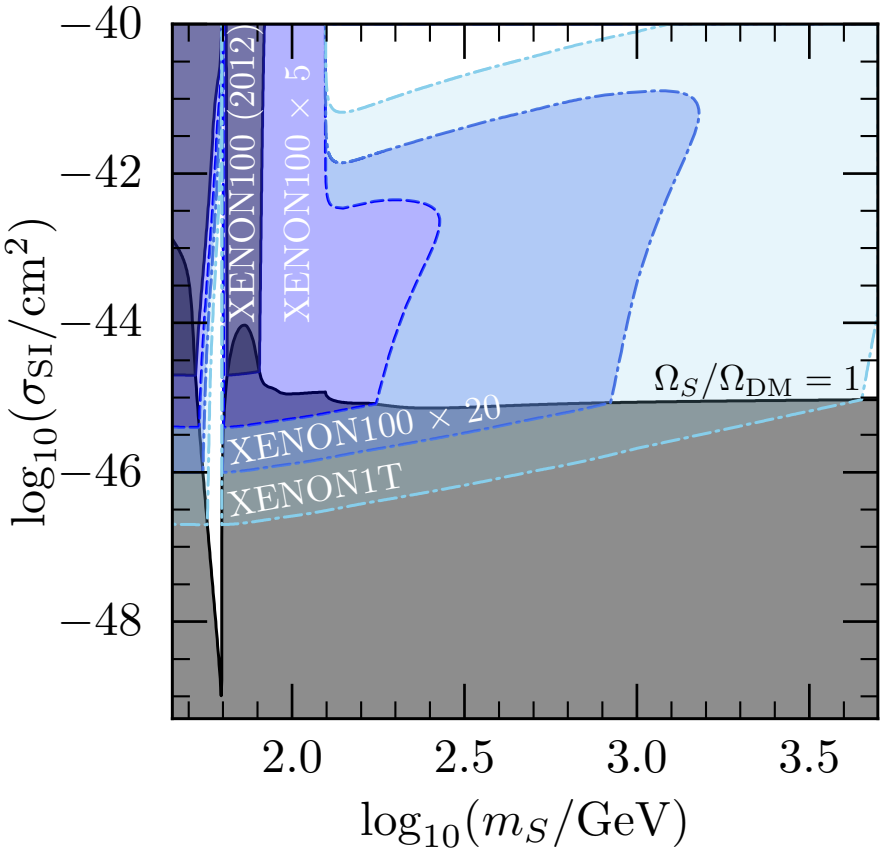


Low-mass region

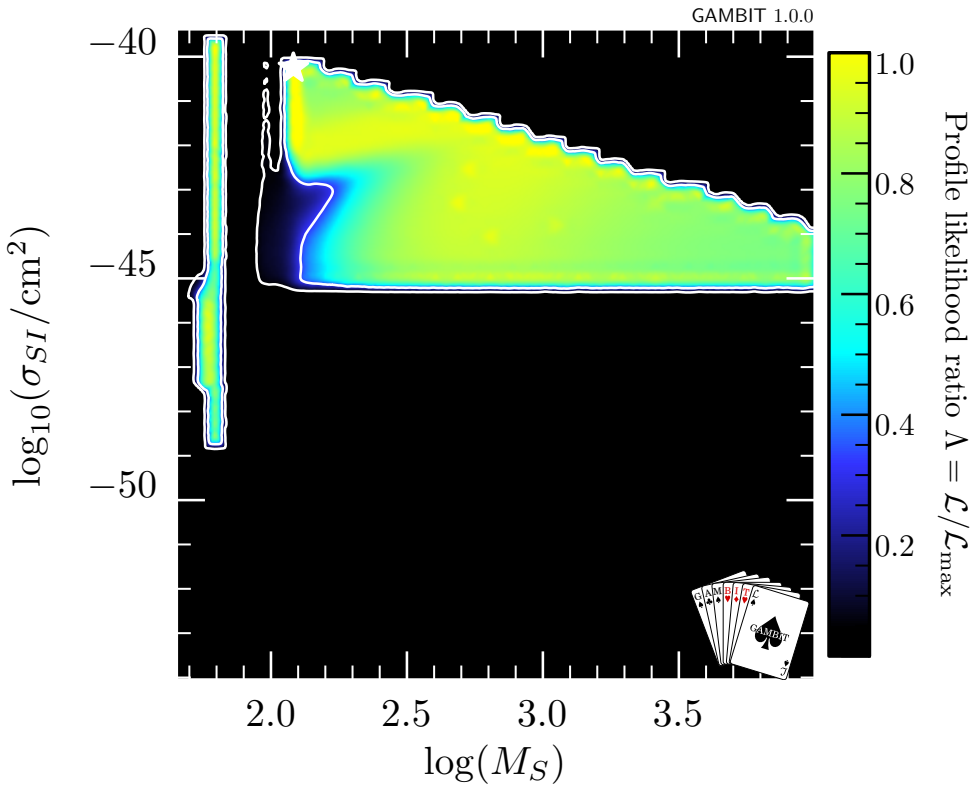
Preliminary results: scalar singlet model

$$\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hs}}{2} S^2 H^\dagger H + \dots$$

Simple scan



Global fit



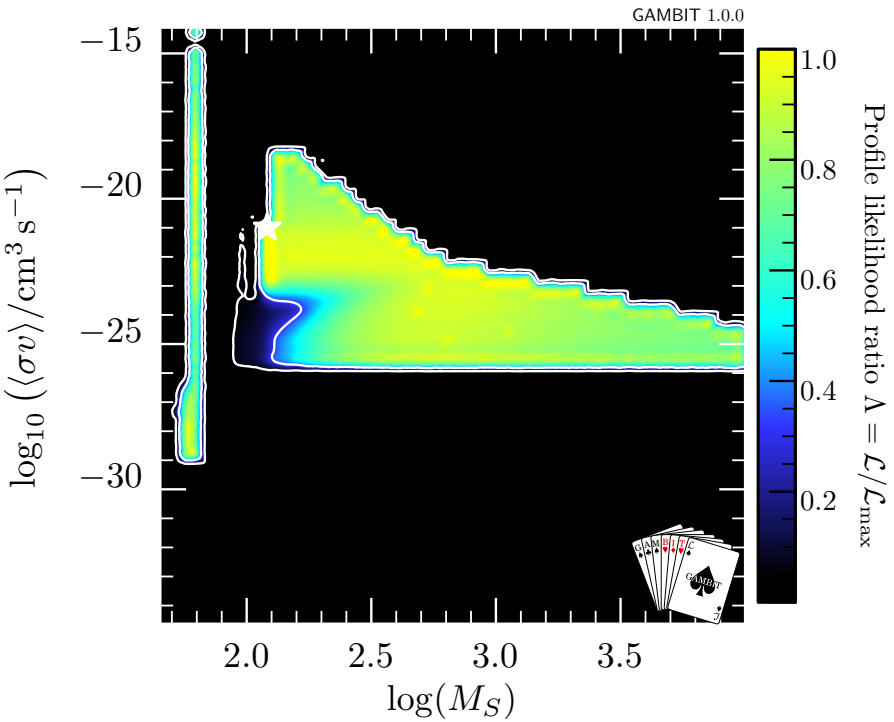
Cline, Kainulainen, PS, Weniger, *PRD* 2013, arXiv:1306.4710

Direct detection cross-section

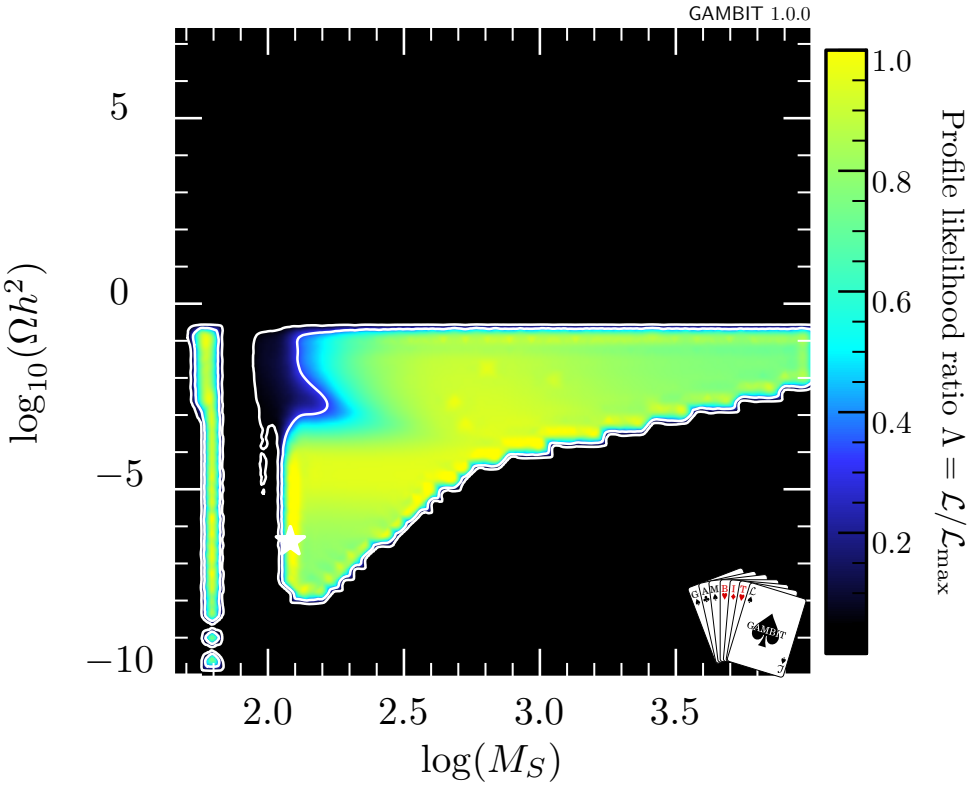
Preliminary results: scalar singlet model

$$\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hs}}{2} S^2 H^\dagger H + \dots$$

Global fit

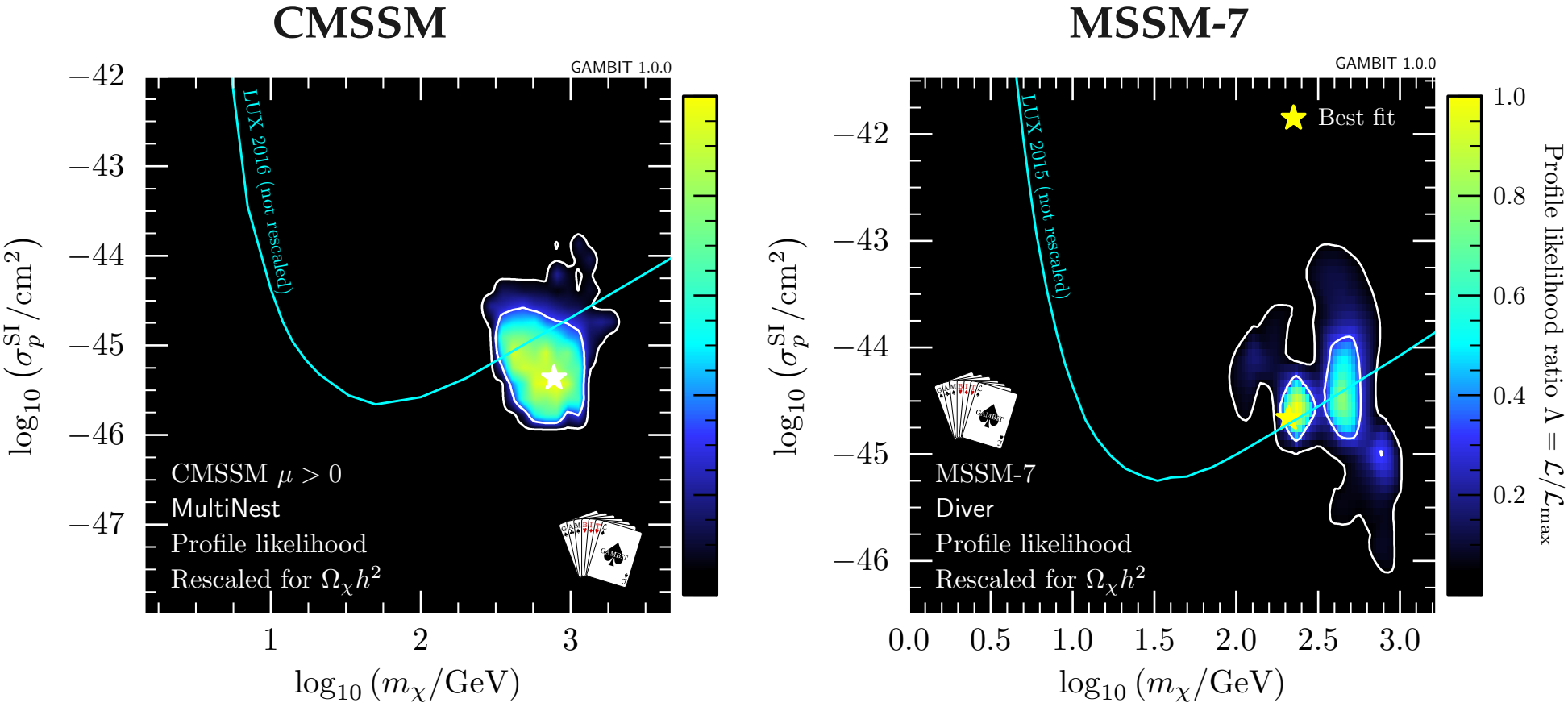


Global fit



Annihilation + relic density

Preliminary results: a SUSY teaser

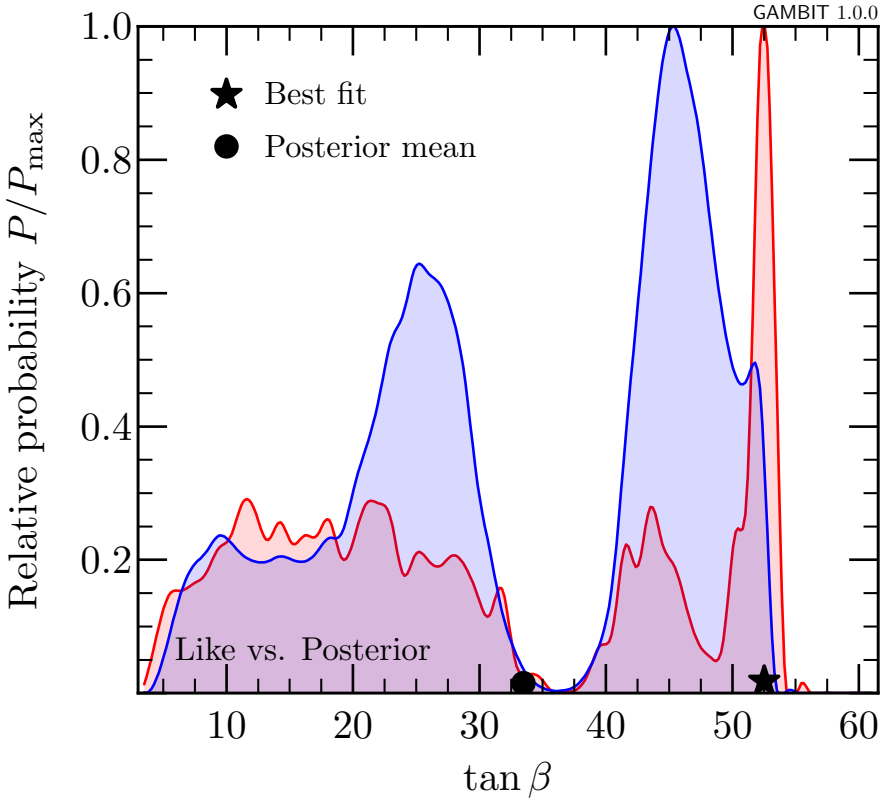


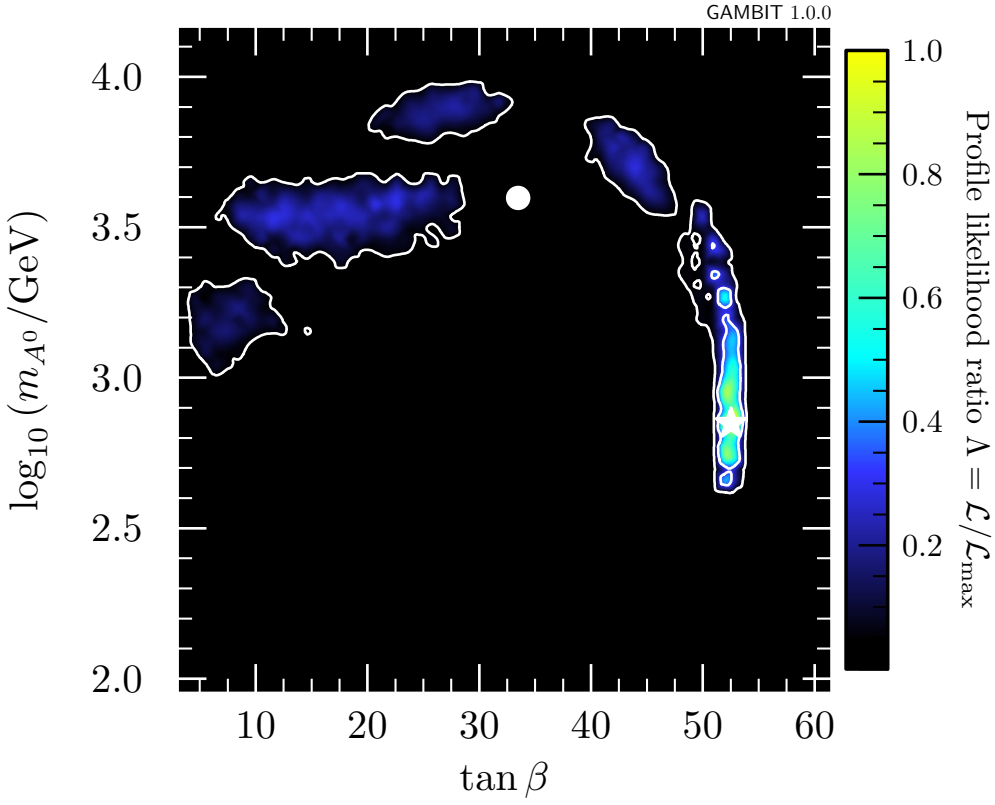
Includes direct simulation of all relevant LHC Run 1 limits.

9(12) parameters:

$$\mathbf{4(7)} \times \text{CMSSM(MSSM7)} + \mathbf{2} \times \text{SM nuisances}(\alpha_S, m_t)$$
$$+\mathbf{2} \times \text{nuclear}(\sigma_s, \sigma_l) + \mathbf{1} \times \text{astro nuis.}(\rho_{\chi,\text{local}})$$

Preliminary results: more CMSSM – *still finalising*





+ NUHM1 & 2 scans just finished – interesting results there, too.
First GAMBIT physics papers approaching…

GAMBIT outlook

- **GAMBIT is almost here:**
  - → Global fits to many models for the first time
  - → Better global fits to familiar ones
  - → Highly modular, usable and extendable public code
  - → Faster, more complete and more consistent theory explorations + experimental analysis prototyping

- **Series of 9 papers in preparation / largely written:**
  - EW-scale MSSMs, CMSSM$\pm\epsilon$ (NUHM, etc), Scalar Singlet
  - DarkBit, ColliderBit, FlavBit, Spec+Decay+PrecisionBits
  - GAMBIT framework, ScannerBit

- **After that: more models, more observables, more data!**
  - Unbinned/shape collider likelihoods, more flexibility, NLO xsecs, RIVET?

RIVET, fast-sim, PROFESSOR, etc. etc.

What is RIVET?

Since RIVET isn't currently a big deal in BSM-land...

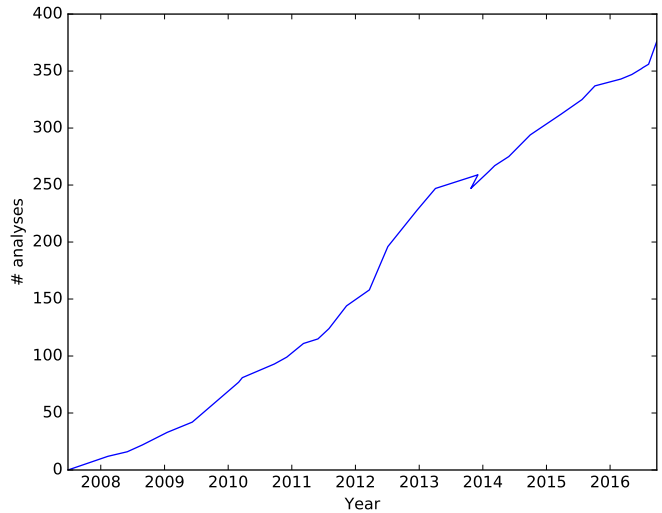
- **RIVET is an MC analysis toolkit, originally developed for MC validation vs. SM data**
  - Designed for MC generator independence and close reproduction of expt methods
  - And low entry barriers – uncluttered "physicist-friendly" interface
- **Has become the LHC standard for archiving measurements in SM, top & Higgs**
  - Important: in ATLAS & CMS it's *standard for the analysis team to write a* RIVET *routine*
  - Along with paper preparation and HepData submissions – and RIVET is integrated with HD
  - After submission we quality-control, maintain, update, regression check, ...
  - Major "feedback" roles in MC generator development, tuning, comparison, ...

⇒ "central" source of LHC analyses for BSM recasting?

Analysis coverage

Currently $>$ 350 analyses total & $>$ 200 LHC alone

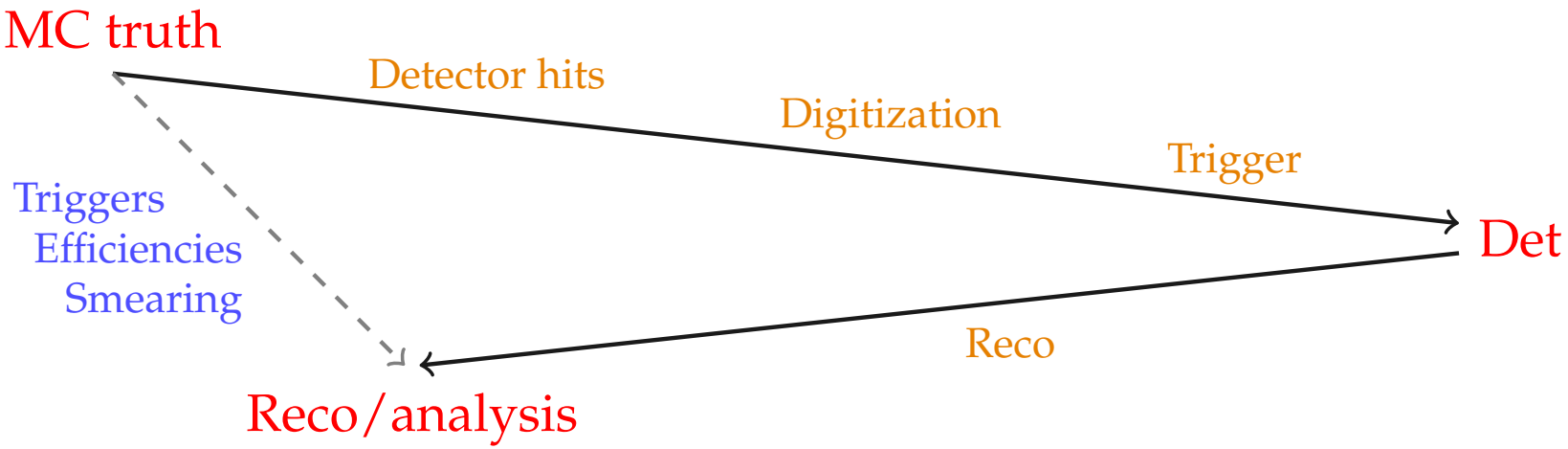
- But only 27 dedicated BSM searches, and not recent
- More BSM-sensitive measurements cf. e.g. Jon Butterworth/CONTUR talk
  $\Rightarrow$ but BSM is currently niche
- Because of focus so far on *unfolded* observables? That's more emphasis than fundamental
- **RIVET 2.5 introduces detector smearing machinery!** BSM only!
- **Want to change this perception and extend the LHC RIVET programme for SM measurements $\rightarrow$ BSM searches**



NB. glitch is RIVET 1.x $\rightarrow$ 2.x migration!

BSM & detector effects

A problem with explicit fast detector simulation:



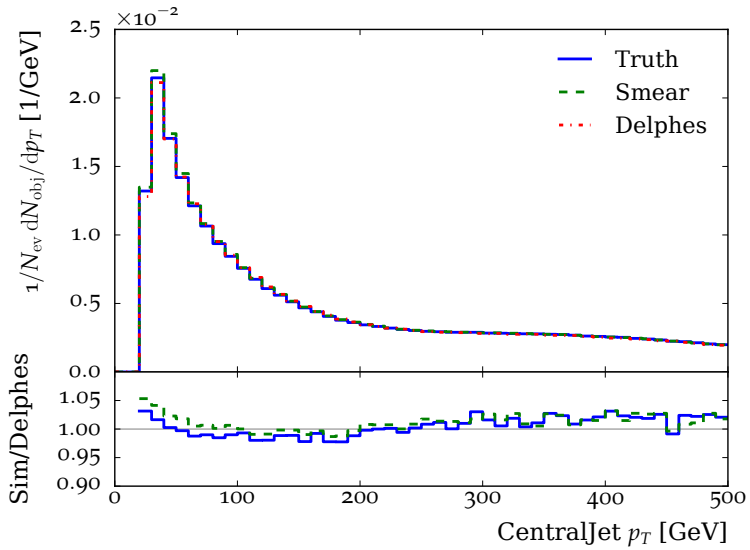
- **Reco already reverses most detector effects!**
- **Explicit fast-sim takes the "long way round".**
- Calibration means that smearing is a few-percent effect. (Lepton) efficiency & mis-ID functions dominate – and are modelled the same as in explicit fast-sim.
- Much harder to fast-sim reco algorithms than detector geometry. . . and they change more frequently.

Smearing vs. fast sim vs. MC truth

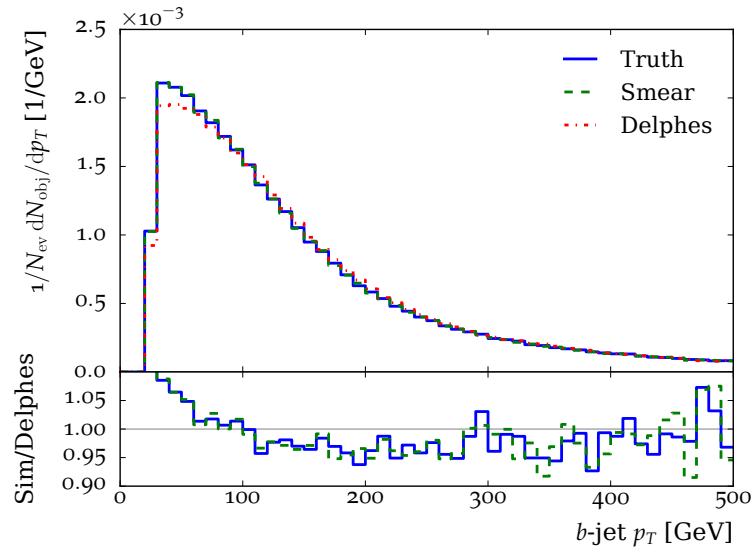
CMSSM eff/smearing effects from RIVET == GAMBIT smearing, in turn using some DELPHES and paper/note calibration functions:

From GAMBIT internal "BuckFast" smearer, with SUSY events:

Central jet p_T



b-jet p_T



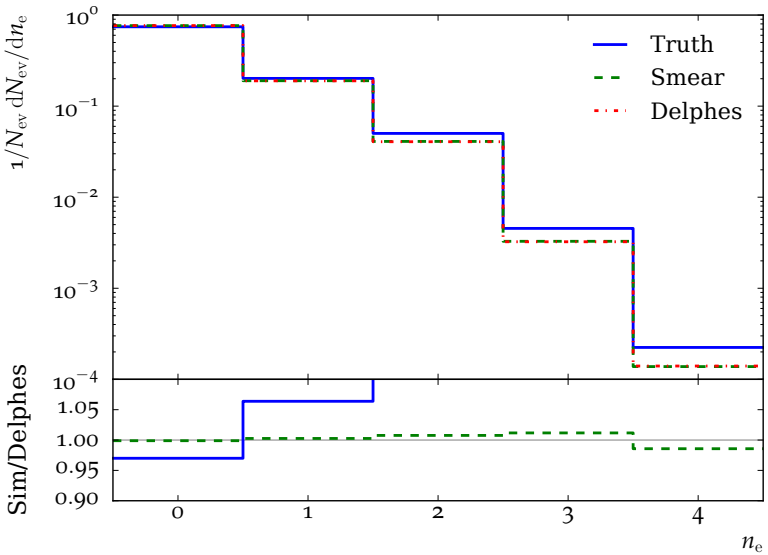
Note major lepton shifts from blue truth to green smeared: difference w.r.t red DELPHES very small

Smearing vs. fast sim vs. MC truth

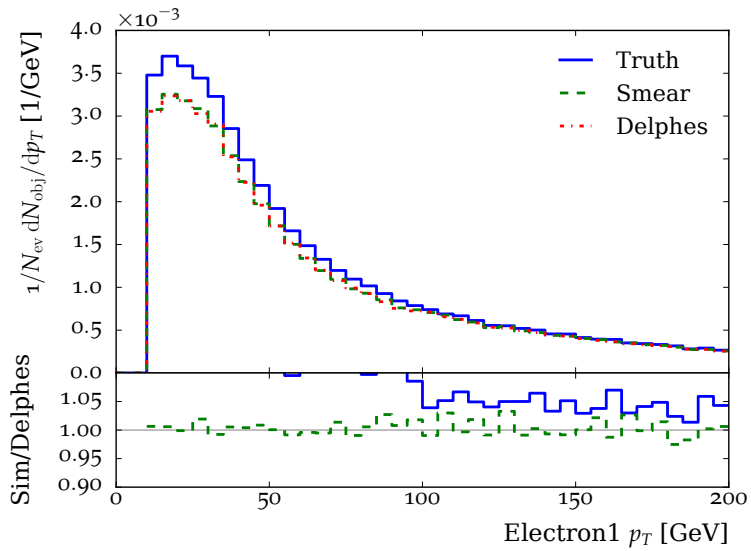
CMSSM eff/smearing effects from RIVET == GAMBIT smearing, in turn using some DELPHES and paper/note calibration functions:

From GAMBIT internal "BuckFast" smearer, with SUSY events:

Electron multiplicity



Leading electron p_T



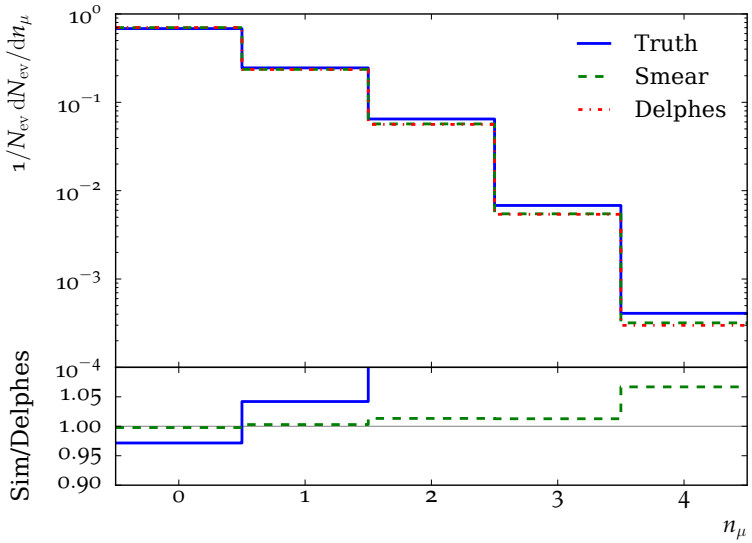
Note major lepton shifts from blue truth to green smeared: difference w.r.t red DELPHES very small

Smearing vs. fast sim vs. MC truth

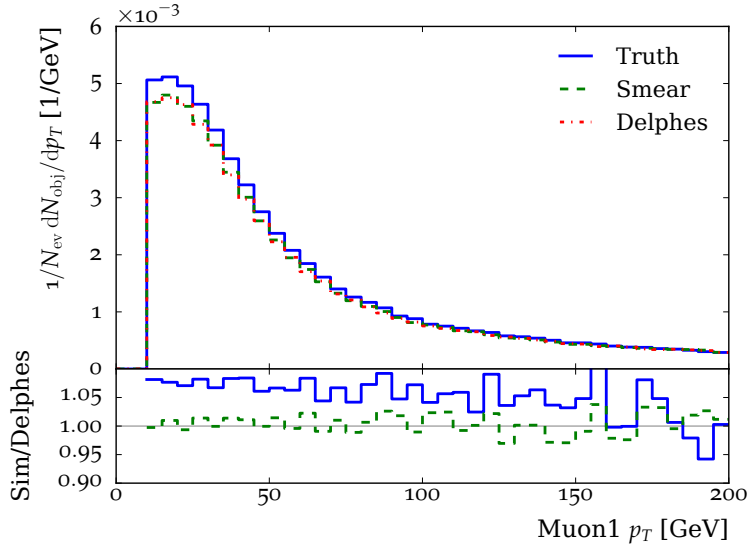
CMSSM eff/smearing effects from RIVET == GAMBIT smearing, in turn using some DELPHES and paper/note calibration functions:

From GAMBIT internal "BuckFast" smearer, with SUSY events:

Muon multiplicity



Leading muon p_T



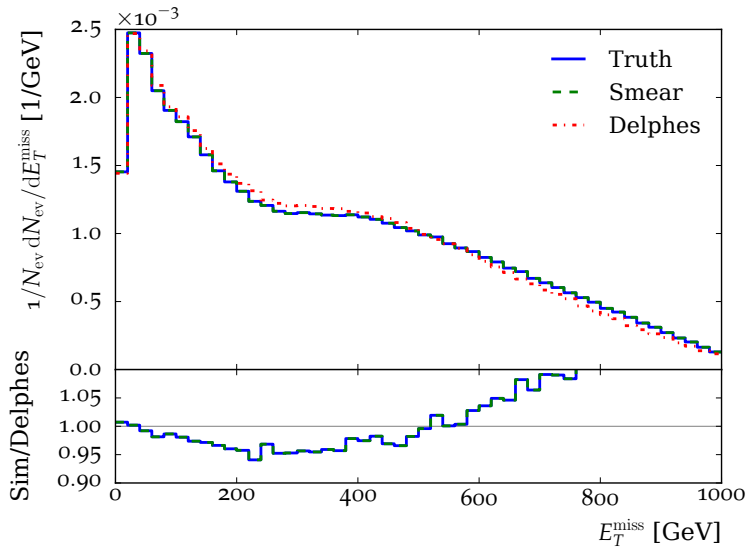
Note major lepton shifts from blue truth to green smeared: difference w.r.t red DELPHES very small

Smearing vs. fast sim vs. MC truth

CMSSM eff/smearing effects from RIVET == GAMBIT smearing, in turn using some DELPHES and paper/note calibration functions:

From GAMBIT internal "BuckFast" smearer, with SUSY events:

Missing E_T



Note major lepton shifts from blue truth to green smeared: difference w.r.t red DELPHES very small and not always rational...

High MET, τ eff, b-tag extrapolation...

BSM & detector effects (II) ⇒ RIVET 2.5

In addition to last slides, *flexibility* of det-sim is important:

- "Global" fast-sims hence difficult for coverage of multiple experiments, multiple runs, multiple reco calibrations, etc.
- Analysis-specific efficiencies and smearings are both more precise and more politically acceptable than "official fast-sims"

⇒ RIVET det-sim as effs+smearing, localised per-analysis

RIVET internally caches results, so global effect sim still efficient

- Functions for generic ATLAS & CMS performance in Runs 1 & 2
- Inline or analysis-specific functions easy to write & *chain*
- Eff/smearing functions can be used directly, e.g. with new functions for candidate object filtering
- Working on embeddability for multithreaded GAMBIT CB.

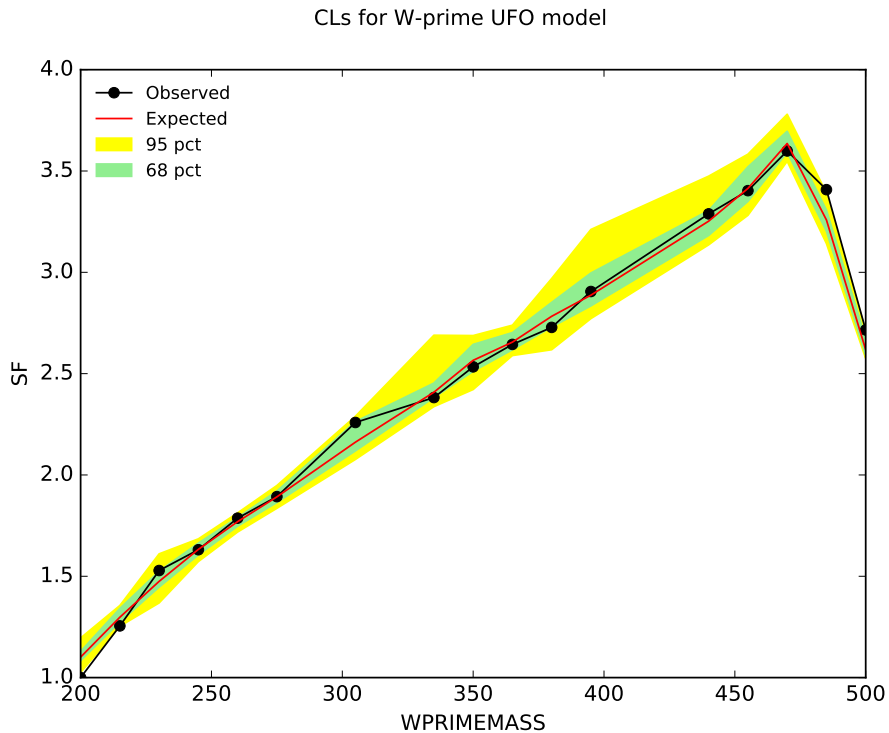
Existing RIVET experience, integration & acceptance in the experiments is worth building on.

Happy to support all BSM users from experiments to recasting tools!

ATOM, LHADA…?

Limit setting with RIVET & PROFESSOR

- Smearing functions allow fair comparison with reco-level data
  - e.g. missing E_T distribution from monojet search (arXiv:1502.01518)
- Data and (total) background readily available on HepData
- Run MC generator with BSM model (UFO), analyze with RIVET
- $\Rightarrow$ hypothesis testing: H_0 background only, $H_1(\vec{p})$ $S + B$
- Can use PROFESSOR for parametrisation and trivial parallelisation in P params, $\vec{p}$
- $\Rightarrow$ Implementation of CL_s purely based on RIVET, in $\sim$ 20 LoC:

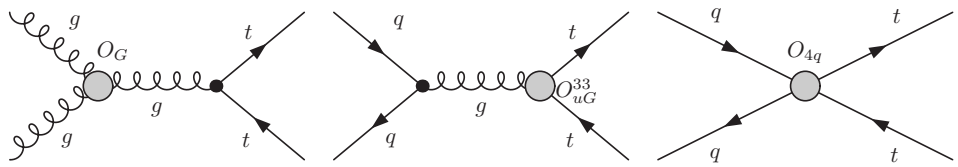


Bellm & Schulz
@ RIVET BSM workshop
Sept 2016

TopFitter– from MC tuning to BSM limit setting

Global effective field theory fits, i.e. BSM modes integrated into *generic* extra operators in $\mathcal{L}$. Leading terms are mass-dimension 6:

$$\mathcal{L}_{\text{eff}} = \mathcal{L}_{\text{SM}} + \frac{1}{\Lambda^2}\sum_i C_i O_i(\ldots) + \mathcal{O}(\Lambda^{-4})\,.$$



Full set of dim 6 EFT operators → FeynRules; feed into MadGraph → analyse events. Rescale to NLO SM + theory uncertainties.

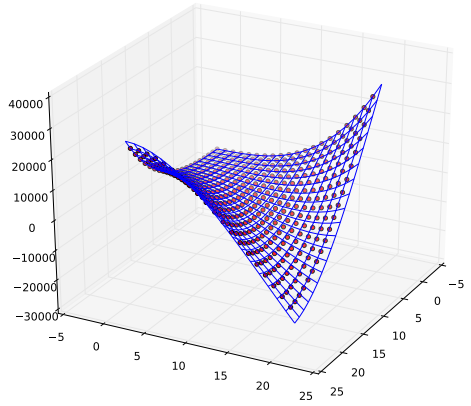
Use Professor MC tuning tool to parametrise LHC & Tevatron observables as fns of EFT coefficients

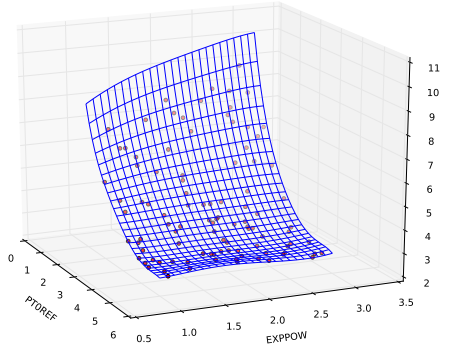
⇒ Currently polynomials, can also use MVAs

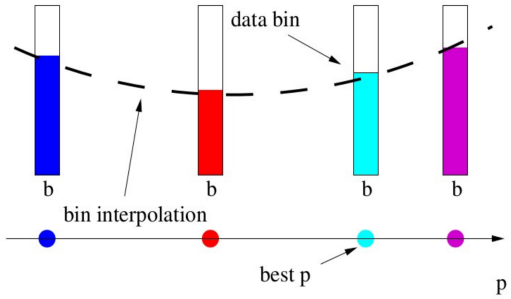
⇒ Currently partonic: Rivet for pseudo-tops, etc.

⇒ χ^2 or $\Delta\chi^2 \to \mathcal{L} \to p$? (Correlations/systs. . .)

cf. Christoph's talk yesterday

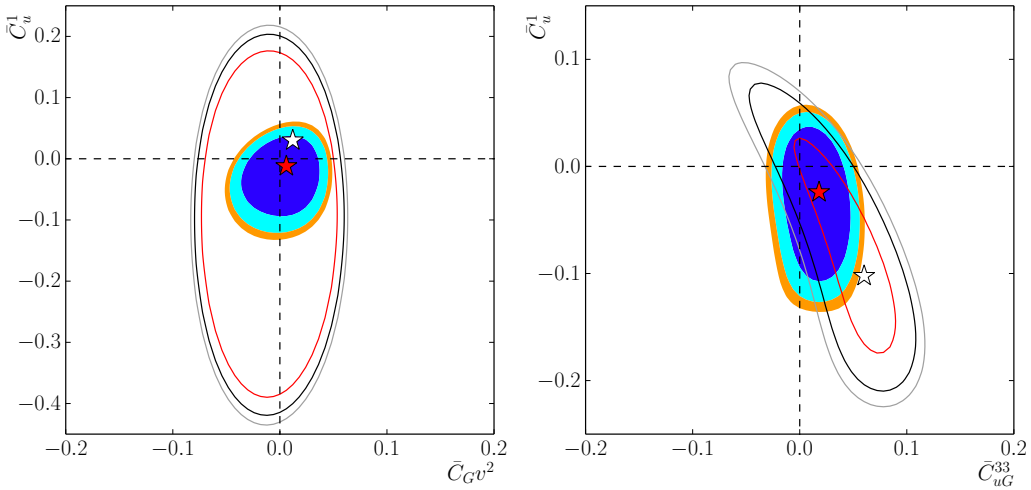


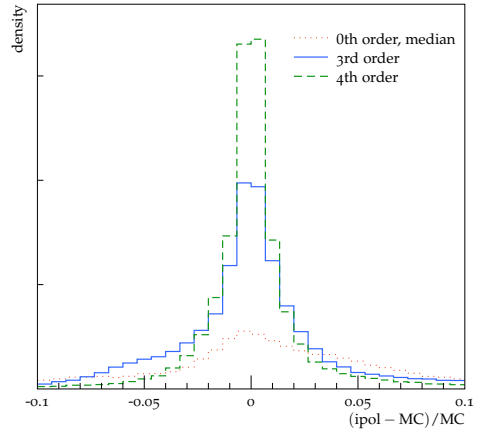


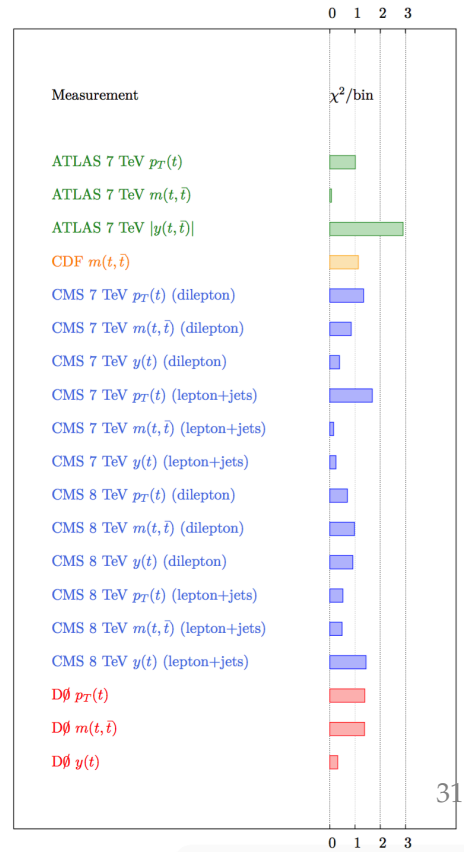


TopFitter results

Split for now into coeff subspaces – simul-fit next







Care with polynomials, speed, stats ⇒ Professor 2 complete rewrite in C++.

***Marginalise* over 200 observables and 5 dimensions: fast parametrisation essential. Scan now possible on GPUs, plus more optimisations in mind.**

Detailed studies possible: tensions between datasets, correlations between operators, optimising observables…

Summary

GAMBIT

- GAMBIT is machinery & expertise for unprecedented global BSM fits
- State-of-the-art components from cosmo/astro to direct DM, to flavour physics, to LHC
- Collider physics is super-expensive in adaptive sampling:
  - Only explicitly generate events if faster constraints don't exclude
  - Lots of work to optimise PYTHIA 8 init, multithreading strategy, etc.
  - Using LO cross-sections for now; NLO pMSSM24 xsec ANN parametrisation attempt failed, but try again / BSM-AI?
- Currently scanning (on Prometheus 41k-core cluster) $\Rightarrow$ finding real-world logistics problems! Papers & code public *soon*

RIVET and other machinery

- RIVET has been a huge success for LHC SM preservation; rolling out now for BSM, with new smearing machinery
- Push to get new population of BSM search analyses, and adding search routine tools & ecosystem
- $\Rightarrow$ *sustainable* supply of *official* analyses for recasting tools, e.g. GAMBIT, TOPFITTER, ...