Martin White (on behalf of the GAMBIT collaboration)

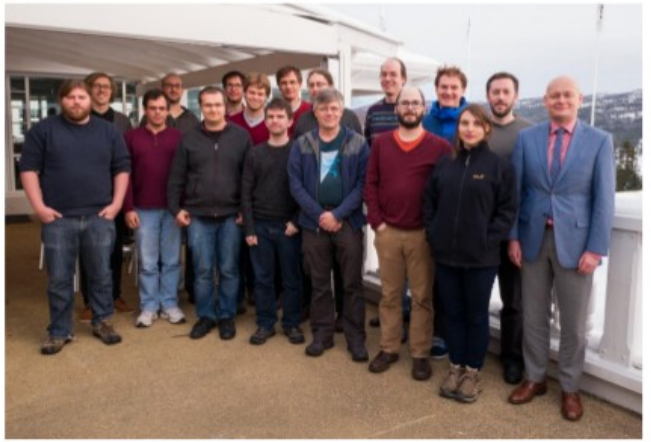
THE UNIVERSITY of ADELAIDE

CoEPP
ARC Centre of Excellence for Particle Physics at the Terascale

GAMBIT: The Global And Modular BSM Inference Tool

- Fast definition of new datasets and theoretical models
- Plug and play scanning, physics and likelihood packages
- Extensive model database – not just SUSY
- Extensive observable/data libraries
- Many statistical and scanning options (Bayesian & frequentist)
- *Fast* LHC likelihood calculator
- Massively parallel
- Fully open-source

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, S. Hoof, F. Kahlhoefer, A. Krislock, A. Kvellestad, M. Pato, F. Mahmoudi, J. McKay, A. Raklev, R. Ruiz, P. Scott, R. Trotta, C. Weniger, M. White



27 Members, 9 Experiments, 4 major theory codes, 10 countries

GAMBIT modules

- **ColliderBit:** collider observables including Higgs + SUSY Searches from ATLAS, CMS, LEP
- **DarkBit:** dark matter observables (relic density, direct & indirect detection, helioseismology, CMB)
- **FlavBit:** including $g-2$, $b \to s\gamma$, B decays (new channels), angular obs., theory unc., LHCb likelihoods
- **SpecBit:** generic BSM spectrum object,providing RGE running, masses, mixings via interchangable interfaces to different RGE codes
- **DecayBit:** decay widths for all relevant SM and BSM particles
- **PrecisionBit:** precision EW tests (mostly via interface to FeynHiggs or SUSY-POPE)
- **ScannerBit:** manages stats, sampling and optimisation

Module functions can depend on:

a) Other module functions (i.e. C++ code within GAMBIT)

b) Backend functions (from external codes)

We would like to backend your code

e

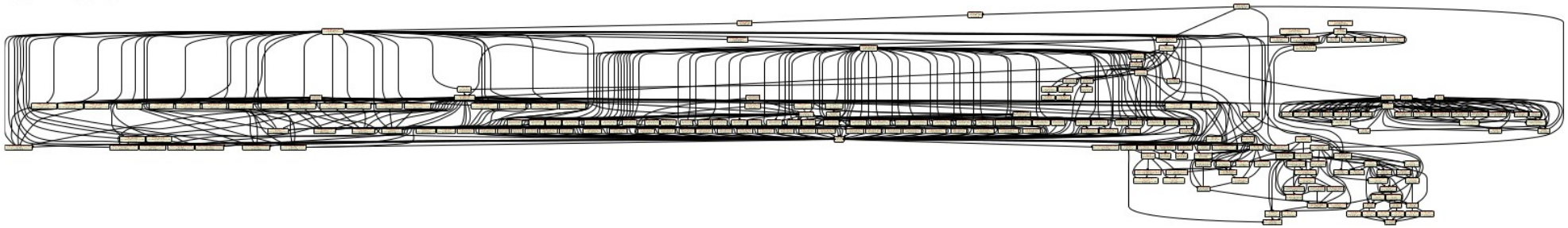
How GAMBIT operates (very roughly)

- The user states what they want to calculate

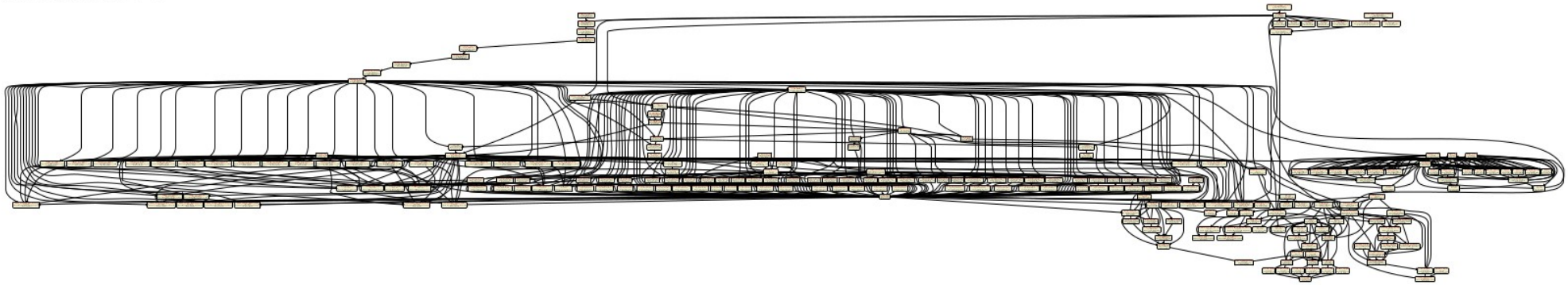
- GAMBIT *resolves dependencies*

  - Module functions and backend functions get arranged into a dependency tree
  - Starting with requested observables and likelihoods, fills each dependency and backend requirement
  - Obeys rules at each step: allowed models, allowed backends, constraints from input file, etc

CMSSM:



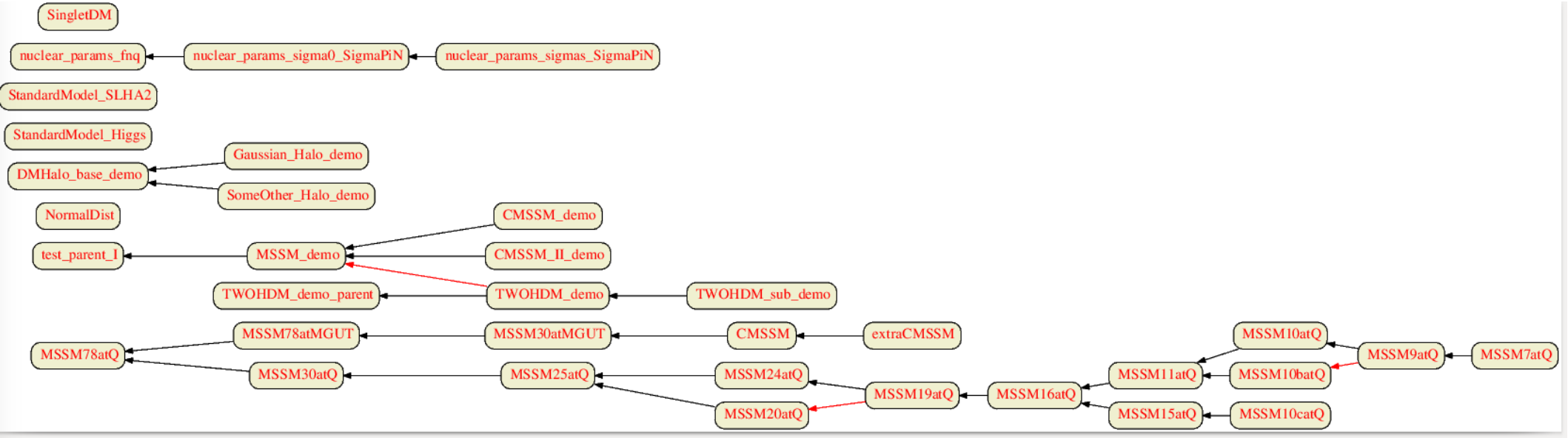
MSSM7:



- GAMBIT calculates observables plus likelihoods, outputs results in user-specified form

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 module/backend function is tracked (safety)



How do we recast LHC limits?

ColliderBit: a GAMBIT module for the calculation of high-energy collider observables and likelihoods

Csaba Balazs[1], Andy Buckley[2], Lars Dal[11],, Ben Farmer[3], Paul Jackson[4], Abram Krislock[5], Anders Kvellestad[6], Antje Putze[10], Are Raklev[5], Christopher Rogan[7], Aldo Saavedra[12], Pat Scott[8], Christoph Weniger[9], Martin White[4]

- **SM-like Higgs:** HiggsSignals and HiggsBounds via backend (more to come after first release)
- **New particle searches:** Calculate limits from first principles using MC generation + fast sim

Can run as a standalone code, or in a full GAMBIT run

Reminder: LHC cut and count analyses

- Given *s* expected signal events, *b* background events and *o* observed events:

$$L = e^{-(s+b)}(s+b)^o \div o!$$

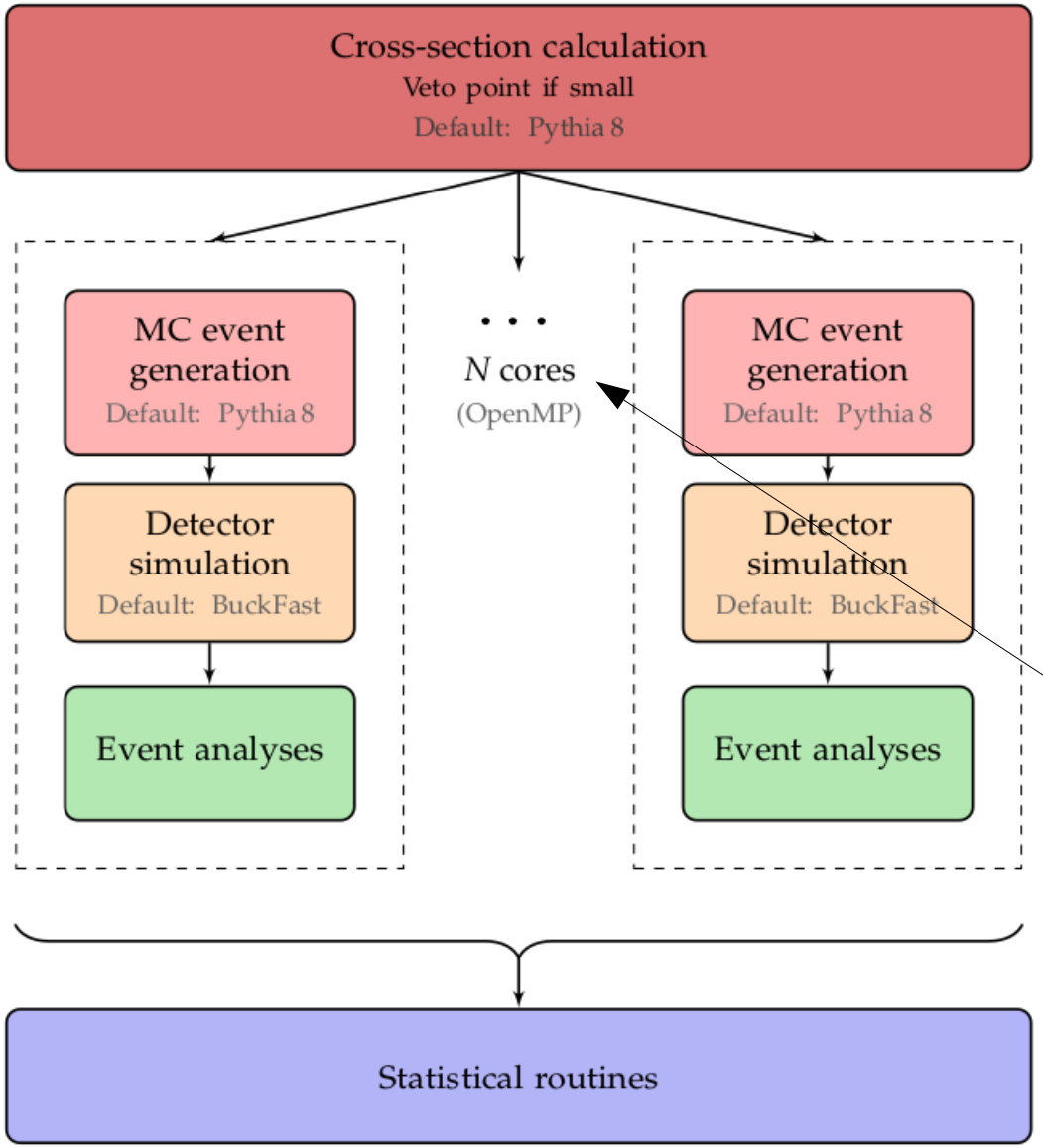
- Systematics on signal and background can be handled via:

$$\mathcal{L}(n|s,b) = \int_0^\infty \frac{[\xi(s+b)]^n e^{-\xi(b+s)}}{n!} P(\xi)\mathrm{d}\xi$$

- With, e.g. , $P(\xi|\sigma_\xi) = \frac{1}{\sqrt{2\pi}\sigma_\xi}\frac{1}{\xi}\exp\left[-\frac{1}{2}\left(\frac{\ln\xi}{\sigma_\xi}\right)^2\right]$

quadrature sum of signal and background systematics

Calculating *s*



$$s = \sigma \times \varepsilon \times L$$

- To be fully rigorous, need to run entire chain
- GAMBIT approach is to make each calculation run as fast as possible

Cross-section calculation(s)

- Use LO+LL cross-sections supplied by MC generator as a default

  - this is all that exists for most WIMP models
  - lags behind state of the art for others (SUSY), but will give conservative limits

- A look up table of NLO cross-sections has been in development for some time for SUSY

  - tricky to do generally in large dimensional spaces
  - the only successful approaches use significant approximations that reduce generality (e.g. NLLFAST)
  - not ready for first release

- After first release, will implement a generic user-supplied cross-section option

MC event generation

- Custom OpenMP parallelised version of the Pythia 8 generator

  - includes other hacks to speed up Pythia execution (e.g. remove redundant flag checks)
  - can generate tens of thousands of events on 8 CPUs in seconds

- First release will be capable of everything Pythia is out of the box plus:

  - can add matrix elements using Madgraph-Pythia interface (see later)

- A more general solution for future versions:

  - interface ColliderBit to general matrix element calculators (e.g. Madgraph)

- Speed can be increased without degrading performance though careful tuning of options:

Configuration	t (10^5 events)	Speed-up
All	1,841 sec	1
$\hookrightarrow -$MPI	671 sec	2.7
$\hookrightarrow -\tau$ correlations	531 sec	3.5
$\hookrightarrow -$FSR	256 sec	7.2
$\hookrightarrow -$hadrons	141 sec	13.0

(1 core)

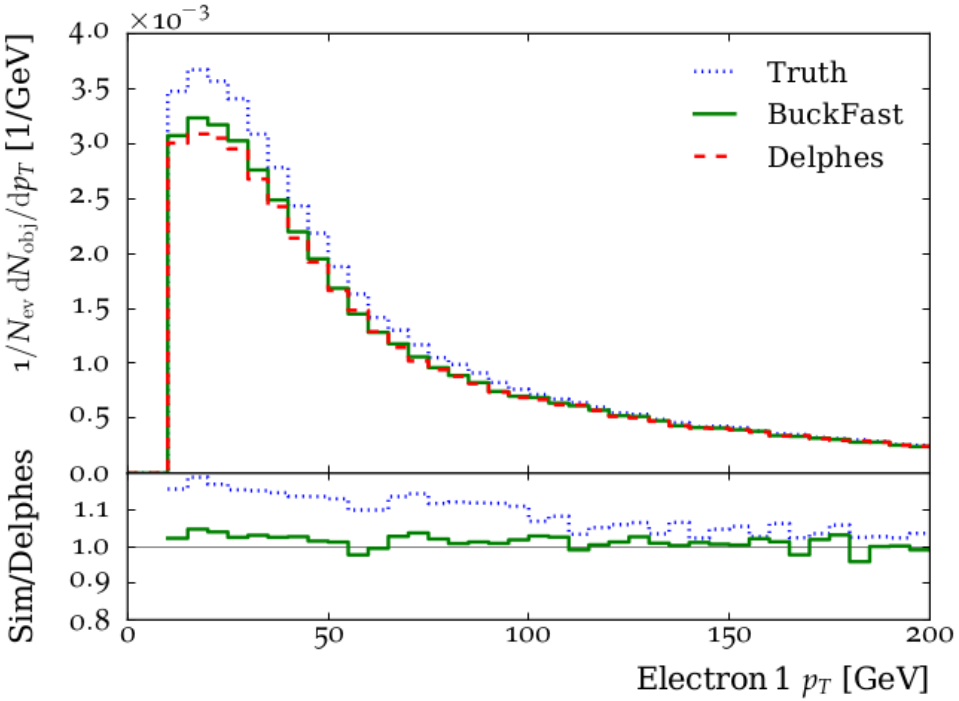
The ColliderBit approach: Detector simulation

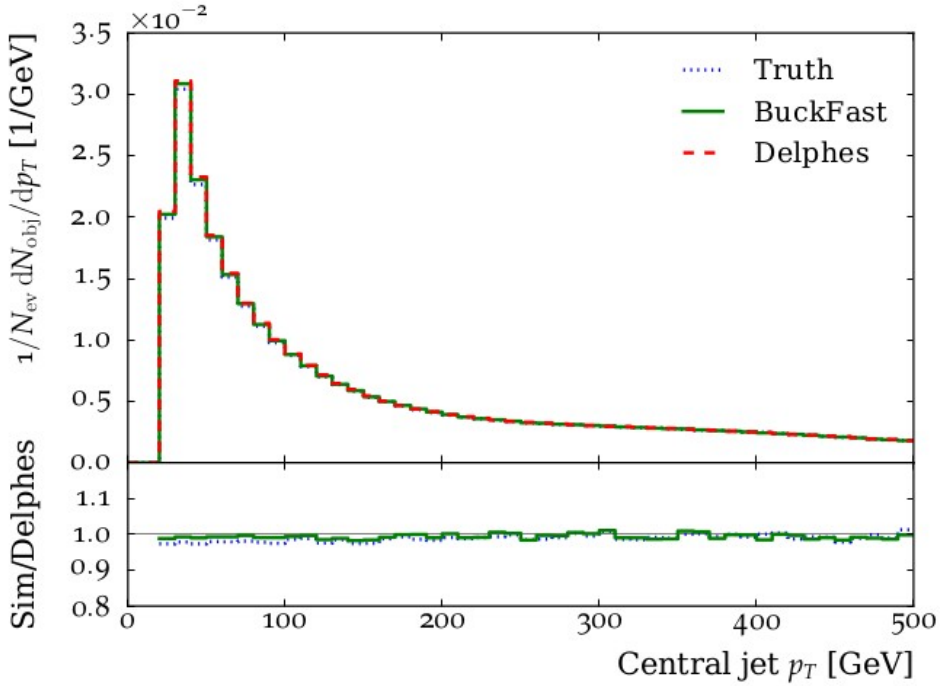
- Have an interface to DELPHES (the standard public LHC detector simulation)

  - unfortunately ROOT is not threadsafe!

- Have also written a threadsafe detector simulation based on four-vector smearing

  - results agree very closely with DELPHES





- The simulation is run in the main event loop, so that it is implicitly parallelised

Collider analysis cuts

- Custom event analysis framework

  - "simulation blind": can be used on event info from any source
  - uses public HepUtils event, jet and particle classes

- First release will contain implementations of (8 TeV):

  - ATLAS SUSY searches (0 lep, 0-1-2 lep stop, b jets plus MET, 2 lep EW, 3 lep EW)
  - CMS DM searches (top pair plus MET, mono-b, mono-jet)
  - CMS multilepton SUSY search
  - good coverage for SUSY and DM effective field theory

Cut	ATLAS	GAMBIT	Ratio
$E_{\rm T}^{\rm miss}$ + jet $p_{\rm T}$ cuts	89.6%	91.0%	1.02
$\Delta\phi_{\rm min} > 0.4$	81.0%	82.5%	1.02
$E_{\rm T}^{\rm miss}/\sqrt{H_{\rm T}} > 15\,{\rm GeV}^{-1/2}$	56.0%	56.8%	1.01
$m_{\rm eff}^{\rm incl} > 1600\,{\rm GeV}$	31.6%	33.4%	1.06

ATLAS 0 lepton cutflow performance, with **BuckFast**

- Likelihoods: use **nulike** implementation of marginalised Poisson likelihoods

DarkBit

- Most astrophysics limits are published using simplified models

  - e.g. effective WIMP, one annihilation channel,etc

- **DarkBit** can cope with:

  - arbitrary direct detection couplings (i.e. potential q and v dependence of cross-section)
  - arbitrary indirect detection decay/annihilation fractions

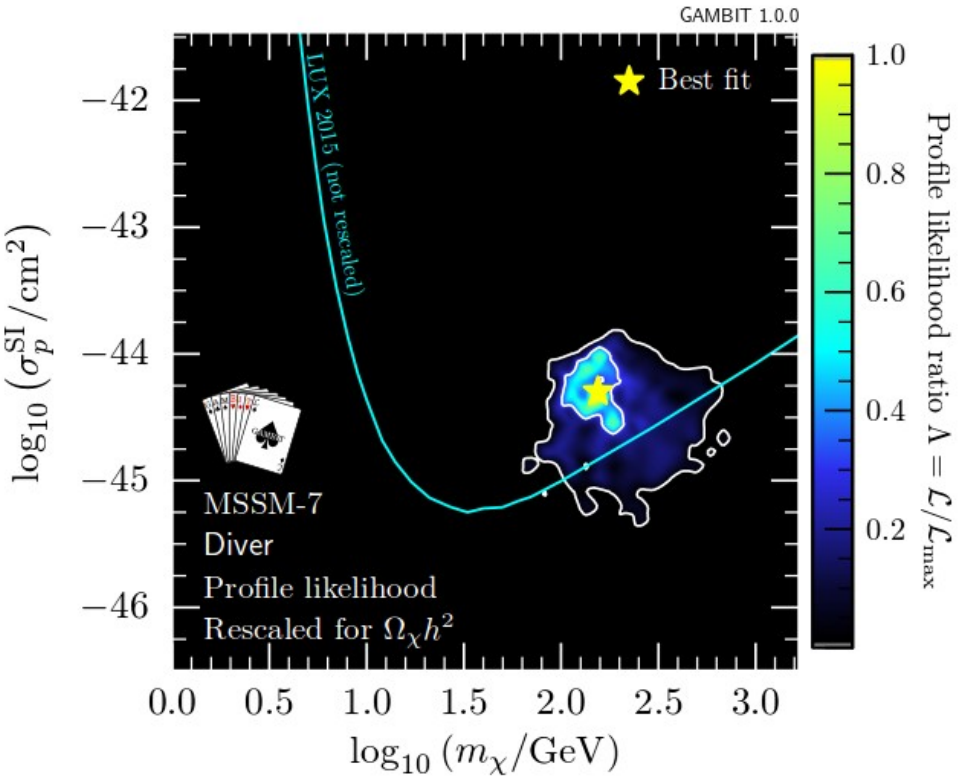
- Impacts of new unstable particles are hard

  - need to simulate decays “on the fly”

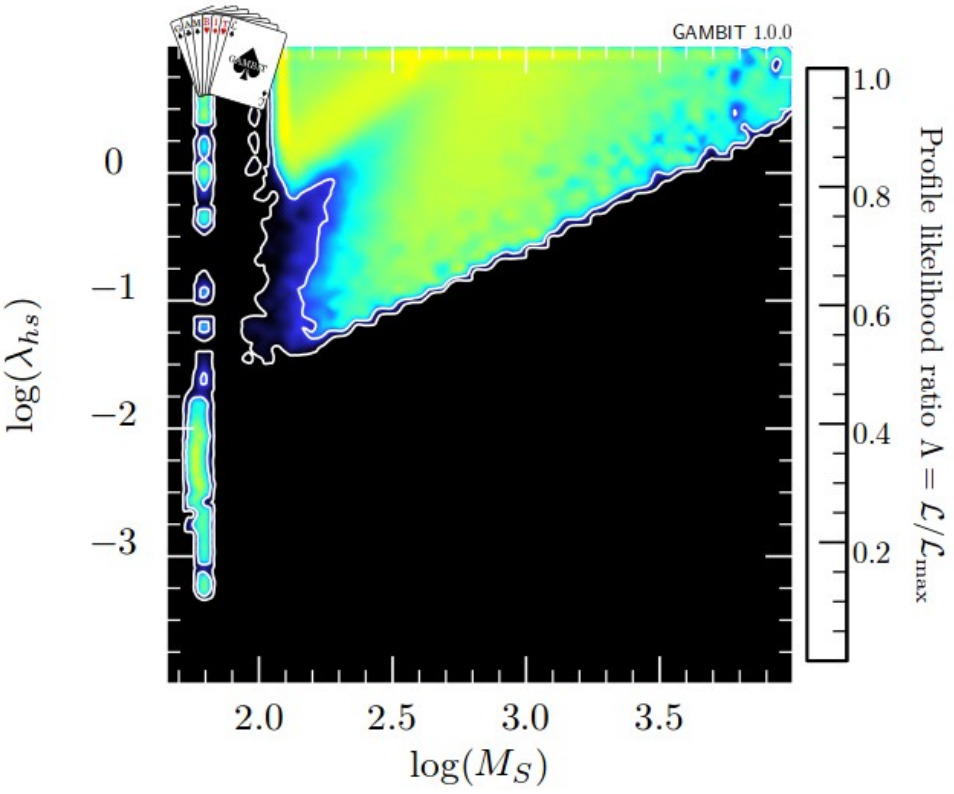
- Calculating relic density for general models is also challenging

  - we abstract the calculation rather than have a presumed model
  - i.e. feed in partial annihilation rates, co-annihilations, resonances, etc

7-parameter minimal supersymmetry



Scalar singlet dark matter



Other nice GAMBIT features

- **Scanners:** MultiNest, Diver (diff. evolution), t-walk (population MC), GreAT (MCMC)
- **Statistics:** Bayesian, Profile Likelihood, later full Neyman
- **Parallelisation:** Mixed-mode MPI + openMP, mostly automated
- **Data handling:** *diskless* generalisation of various Les Houches Accords
- **Backends:** dynamic loading of C++ classes from backends (BOSS)
- **Compilation:**

  - all-in or module standalone modes – easily implemented from single cmake script
  - automatic getters for obtaining, configuring + compiling backends

- **Output:** flexible output streams (ASCII, databases, HDF5, . . .)
- more more more. . .

Summary

- GAMBIT is to be released later this year

  - faster, more complete exploration of theoretical BSM options
  - easy way to generate benchmark points for experimental optimisations

We are very happy to speak to prospective users!