# Complementarity of different Dark Matter searches

Pat Scott

Imperial College London

Slides at: tinyurl.com/patscott

# Outline

1. Background: combining dark matter searches
2. Indirect detection with neutrinos
3. Global fits and GAMBIT

# Outline

1. Background: combining dark matter searches
2. Indirect detection with neutrinos
3. Global fits and GAMBIT

**Question**

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

# Combining searches I

**Question**

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

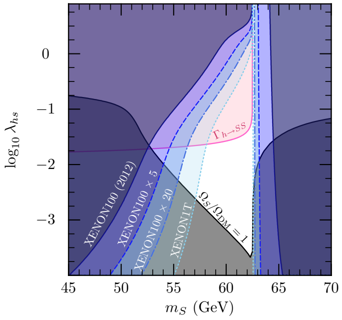
**Answer**

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

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

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



# Combining searches II

That's all well and good if there are only 2 parameters and few searches. . .

**Question**

What if there are many different constraints?

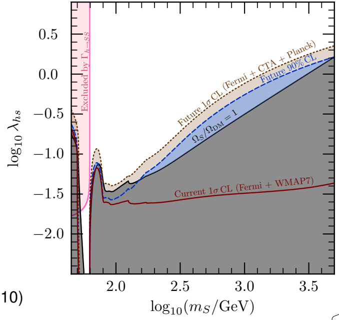
That's all well and good if there are only 2 parameters and few searches. . .

## Question

What if there are many different constraints?

## Answer

Combine constraints in a statistically valid way
$\longrightarrow$ composite likelihood




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

# Combining searches III

That's all well and good if there are only 2 parameters and few searches. . .

**Question**

What if there are many parameters?

# Combining searches III

That's all well and good if there are only 2 parameters and few searches…

**Question**

What if there are many parameters?

**Answer**

Need to

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

$\longrightarrow$ global fits

# Outline

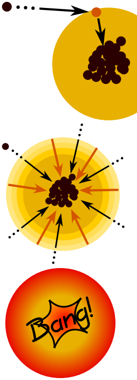
1. Background: combining dark matter searches
2. Indirect detection with neutrinos
3. Global fits and GAMBIT

- Indirect limits always presented in terms of hard process final states
- Actual experiments do not measure those final states – they detect one type of SM particle produced later: $\gamma$s, $\nu$s, etc
- Limits as presented cannot be combined and applied to models with mixed final states (= all non-toy models)
- Proper treatment of indirect detection for BSM searches requires full phenomenological recast abilities
$\rightarrow$ full experimental *and* theoretical treatment at the same time

# Neutrino telescope likelihoods: `nulike`

Unbinned $\nu$ telescope likelihood $\implies$ full event-level angular and energy info

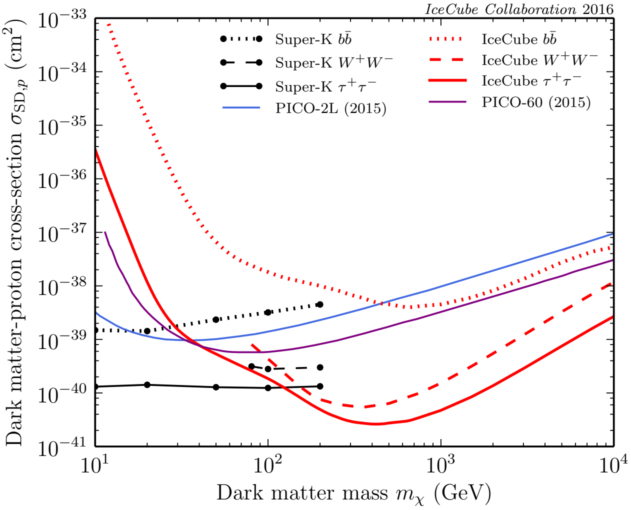
$$\mathcal{L}_{\mathrm{unbin}} \equiv \mathcal{L}_{\mathrm{num}}(n_{\mathrm{tot}}|\theta_{\mathrm{tot}}) \prod_{i=1}^{n_{\mathrm{tot}}} (f_{\mathrm{S}}\mathcal{L}_{\mathrm{S},i} + f_{\mathrm{BG}}\mathcal{L}_{\mathrm{BG},i})$$




Strategy: precompute partial likelihoods for each event, then reweight with the $\nu$ spectrum at Earth for each model

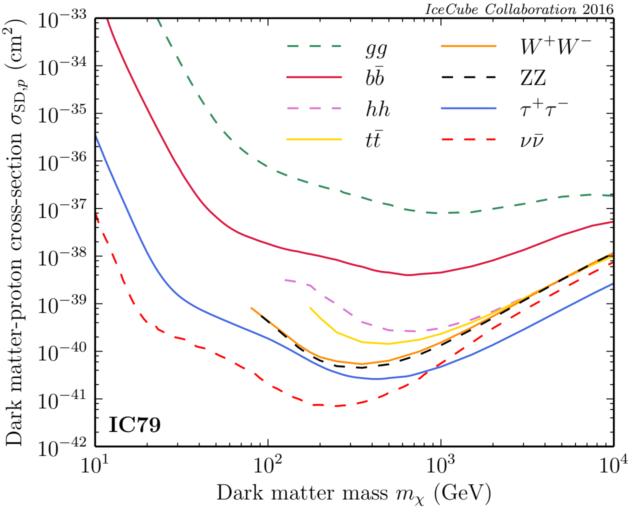
- precompute step uses `nusigma` with CTEQ6-DIS PDFs to get charged current $\nu - n$ and $\nu - p$ cross-sections as function of $x$ and $y$
- like step input: neutrino spectrum at Earth (from DarkSUSY or whatever else you want to use)
- like step output: num predicted events, likelihood
- → fully model-independent = future-proof for global fits

# Neutrino telescope likelihoods: `nulike`



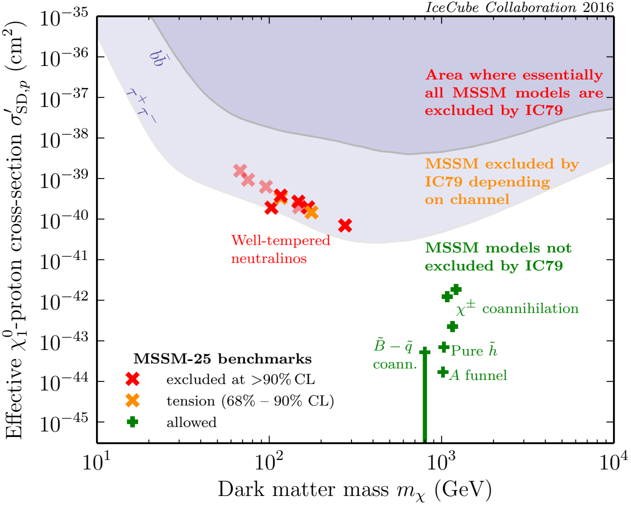

IceCube Collab. (contacts: PS + M. Danninger) arXiv:1601.00653, *JCAP* in press
`nulike`: model-independent unbinned limit calculator for generic BSM models
Publicly available at nulike.hepforge.org

# Neutrino telescope likelihoods: nulike




IceCube Collab. (contacts: PS + M. Danninger) arXiv:1601.00653, *JCAP* in press
nulike: model-independent unbinned limit calculator for generic BSM models
Publicly available at nulike.hepforge.org

# Neutrino telescope likelihoods: nulike




IceCube Collab. (contacts: PS + M. Danninger) arXiv:1601.00653, *JCAP* in press

nulike: model-independent unbinned limit calculator for generic BSM models

Publicly available at nulike.hepforge.org

# Outline

1. Background: combining dark matter searches
2. Indirect detection with neutrinos
3. Global fits and GAMBIT

# 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)

$\Longrightarrow$ *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)

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

How to quickly recast data, likelihood functions, scanning code 'housekeeping' and even theory predictions to new theories?
$\Longrightarrow$ a new, very general global fitting framework

# 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)

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

How to quickly recast data, likelihood functions, scanning code ‘housekeeping’ and even theory predictions to new theories?
$\Longrightarrow$ a new, very general global fitting framework

$\Longrightarrow$ GAMBIT

# GAMBIT: a *second-generation* global fit code

GAMBIT: The Global And Modular BSM Inference Tool

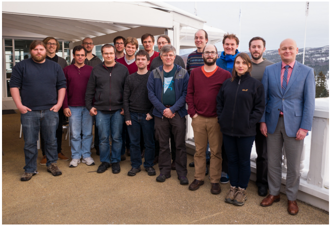
Overriding principles of GAMBIT: flexibility and modularity

- 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

# The GAMBIT Collaboration

30 Members, 17 institutions, 10 countries,
11 Experiments, 4 major theory codes

| | |
|---|---|
| **ATLAS** | A. Buckley, P. Jackson, C. Rogan, M. White, |
| **LHCb** | M. Chrząszcz, N. Serra |
| **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 |
| **AMS-02** | A. Putze |
| **CDMS, DM-ICE** | L. Hsu |
| **XENON/DARWIN** | J. Conrad |
| **Theory** | P. Athron, C. Balázs, T. Bringmann, J. Cornell, J. Edsjö, B. Farmer, A. Krislock, A. Kvellestad, M. Pato, F. Mahmoudi, A. Raklev, P. Scott, C. Weniger, M. White |



+recently joined: T. Gonzales, F. Kahlhoefer, J. McKay, R. Ruiz, R. Trotta
-recently retired: L. Dal, A. Saavedra, C. Savage

## Physics modules

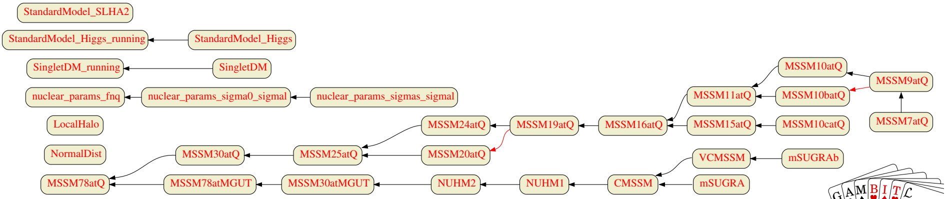
- DarkBit – dark matter observables (relic density, direct + indirect detection)
- ColliderBit – collider observables inc. Higgs + SUSY searches from ATLAS, CMS + LEP
- FlavBit – flavour physics inc. $g-2$, $b \to s\gamma$, $B$ decays (new channels, angular obs., theory uncerts, LHCb likelihoods)
- SpecBit – 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 ($W$ mass, $\Delta\rho$ etc)

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

+ScannerBit: manages stats, sampling and optimisation

# 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)

# Backends: mix and match

pat@xpspedition: ~/gambit 163x45

All relative paths are given with reference to /home/pat/gambit.

| BACKENDS | VERSION | PATH TO LIB | STATUS | #FUNC | #TYPES | #CTORS |
|---|---|---|---|---|---|---|
| DDCalc0 | 0.0 | Backends/installed/DDCalc/0.0/libDDCalc0.so | OK | 62 | 0 | 0 |
| DarkSUSY | 5.1.1 | Backends/installed/DarkSUSY/5.1.1/lib/libdarksusy.so | OK | 68 | 0 | 0 |
| FastSim | 1.0 | Backends/installed/fastsim/1.0/libfastsim.so | absent/broken | 1 | 0 | 0 |
| FeynHiggs | 2.11 | Backends/installed/FeynHiggs/2.11.2/lib/libFH.so | OK | 14 | 0 | 0 |
| HiggsBounds | 4.2.1 | Backends/installed/HiggsBounds/4.2.1/lib/libhiggsbounds.so | OK | 10 | 0 | 0 |
| HiggsSignals | 1.4 | Backends/installed/HiggsSignals/1.4.0/lib/libhiggssignals.so | OK | 11 | 0 | 0 |
| LibFarrayTest | 1.0 | Backends/examples/libFarrayTest.so | OK | 9 | 0 | 0 |
| LibFirst | 1.0 | Backends/examples/libfirst.so | OK | 8 | 0 | 0 |
| | 1.1 | Backends/examples/libfirst.so | OK | 15 | 0 | 0 |
| LibFortran | 1.0 | Backends/examples/libfortran.so | OK | 6 | 0 | 0 |
| MicrOmegas | 3.5.5 | Backends/installed/micromegas/3.5.5/MSSM/MSSM/libmicromegas.so | OK | 15 | 0 | 0 |
| MicrOmegasSingletDM | 3.5.5 | Backends/installed/micromegas/3.5.5/SingletDM/SingletDM/libmicromegas.so | OK | 13 | 0 | 0 |
| Pythia | 8.186 | Backends/installed/Pythia/8.186/lib/libpythia8.so | absent/broken | 0 | 27 | 105 |
| | 8.209 | Backends/installed/Pythia/8.209/lib/libpythia8.so | OK | 0 | 28 | 107 |
| SUSYPOPE | 0.2 | no path in config/backend_locations.yaml | absent/broken | 3 | 0 | 0 |
| SUSY_HIT | 1.5 | Backends/installed/SUSY-HIT/1.5/libsusyhit.so | OK | 55 | 0 | 0 |
| SuperIso | 3.4 | Backends/installed/SuperIso/3.4/libsuperiso.so | OK | 32 | 0 | 0 |
| gamLike | 1.0.0 | Backends/installed/gamLike/1.0.0/lib/gamLike.so | OK | 3 | 0 | 0 |
| nulike | 1.0.0 | Backends/installed/nulike/1.0.0/lib/libnulike.so | OK | 4 | 0 | 0 |

Gambit diagnostic backend line 1 (press h for help or q to quit)

# Backends: mix and match

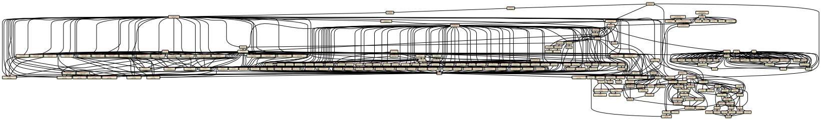
pat@xpspedition: ~/gambit 163x45

All relative paths are given with reference to /home/pat/gambit.

| BACKENDS | VERSION | PATH TO LIB | STATUS | #FUNC | #TYPES | #CTORS |
|---|---|---|---|---|---|---|
| DDCalc0 | 0.0 | Backends/installed/DDCalc/0.0/libDDCalc0.so | OK | 62 | 0 | 0 |
| DarkSUSY | 5.1.1 | Backends/installed/DarkSUSY/5.1.1/lib/libdarksusy.so | OK | 68 | 0 | 0 |
| FastSim | 1.0 | Backends/installed/fastsim/1.0/libfastsim.so | absent/broken | 1 | 0 | 0 |
| FeynHiggs | 2.11 | Backends/installed/FeynHiggs/2.11.2/lib/libFH.so | OK | 14 | 0 | 0 |
| HiggsBounds | 4.2.1 | Backends/installed/HiggsBounds/4.2.1/lib/libhiggsbounds.so | OK | 10 | 0 | 0 |
| HiggsSignals | 1.4 | Backends/installed/HiggsSignals/1.4.0/lib/libhiggssignals.so | OK | 11 | 0 | 0 |
| LibFarrayTest | 1.0 | Backends/examples/libFarrayTest.so | OK | 9 | 0 | 0 |
| LibFirst | 1.0 | Backends/examples/libfirst.so | OK | 8 | 0 | 0 |
| | 1.1 | Backends/examples/libfirst.so | OK | 15 | 0 | 0 |
| LibFortran | 1.0 | Backends/examples/libfortran.so | OK | 6 | 0 | 0 |
| MicrOmegas | 3.5.5 | Backends/installed/micromegas/3.5.5/MSSM/MSSM/libmicromegas.so | OK | 15 | 0 | 0 |
| MicrOmegasSingletDM | 3.5.5 | Backends/installed/micromegas/3.5.5/SingletDM/SingletDM/libmicromegas.so | OK | 13 | 0 | 0 |
| Pythia | 8.186 | Backends/installed/Pythia/8.186/lib/libpythia8.so | absent/broken | 0 | 27 | 105 |
| | 8.209 | Backends/installed/Pythia/8.209/lib/libpythia8.so | OK | 0 | 28 | 107 |
| SUSYPOPE | 0.2 | no path in config/backend_locations.yaml | absent/broken | 3 | 0 | 0 |
| SUSY_HIT | 1.5 | Backends/installed/SUSY-HIT/1.5/libsusyhit.so | OK | 55 | 0 | 0 |
| SuperIso | 3.4 | Backends/installed/SuperIso/3.4/libsuperiso.so | OK | 32 | 0 | 0 |
| gamLike | 1.0.0 | Backends/installed/gamLike/1.0.0/lib/gamLike.so | OK | 3 | 0 | 0 |
| nulike | 1.0.0 | Backends/installed/nulike/1.0.0/lib/libnulike.so | OK | 4 | 0 | 0 |

Gambit diagnostic backend line 1 (press h for help or q to quit)

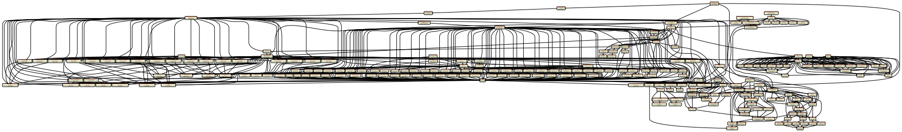
# Dependency Resolution



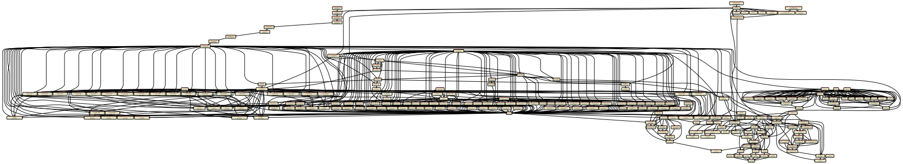
- Module functions and backend functions get arranged into a **dependency tree**
- Starting with requested observables and likelihoods, GAMBIT fills each dependency and backend requirement
- Obeys **rules** at each step: allowed models, allowed backends, constraints from input file, etc
- → tree constitutes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

# Dependency Resolution

CMSSM:

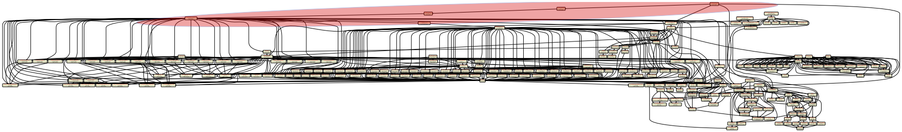


MSSM7:

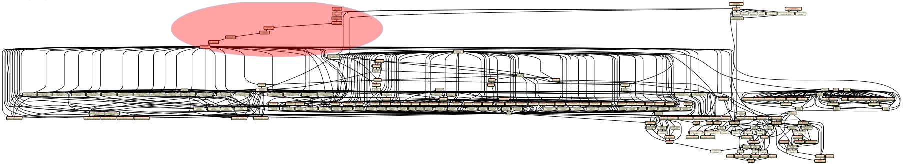


# Dependency Resolution

CMSSM:



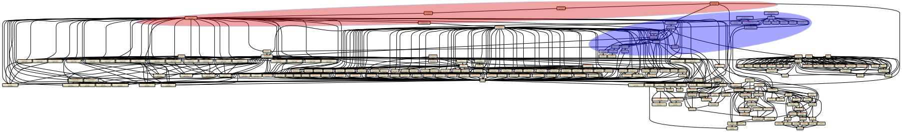
MSSM7:



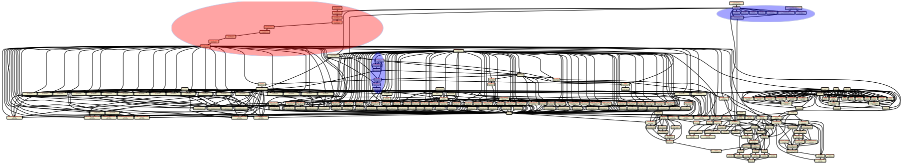
Red: Model parameter translations

# Dependency Resolution

CMSSM:



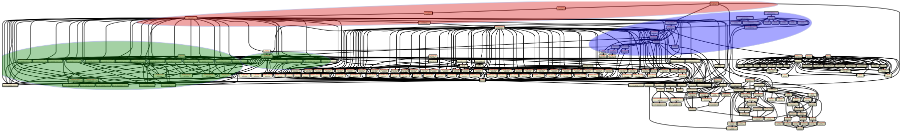
MSSM7:



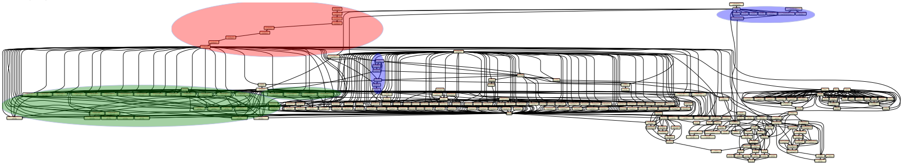
Red: Model parameter translations
Blue: Precision calculations

# Dependency Resolution

CMSSM:



MSSM7:



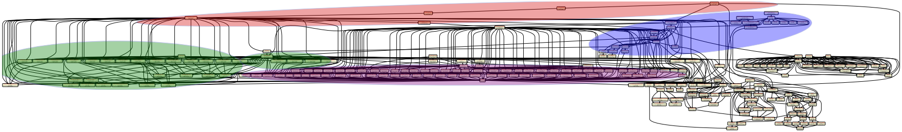
Red: Model parameter translations

Blue: Precision calculations

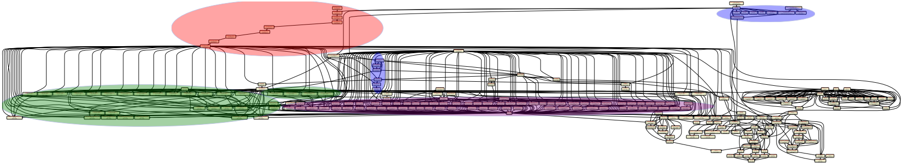
Green: LEP rates+likelihoods

# Dependency Resolution

CMSSM:



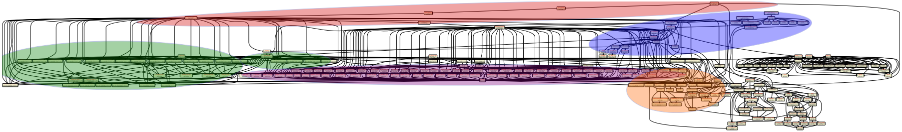
MSSM7:



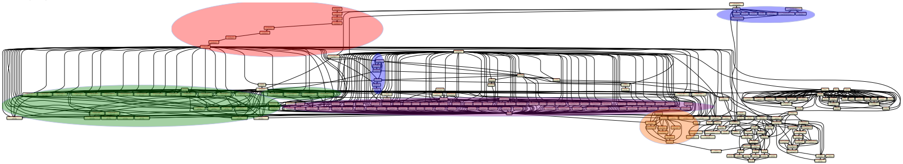
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays

# Dependency Resolution

CMSSM:



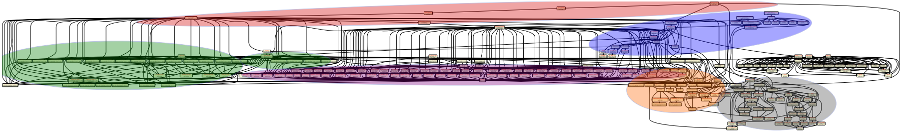
MSSM7:



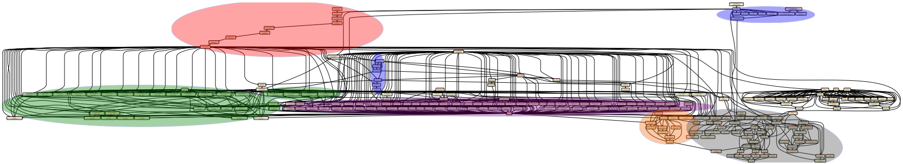
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods

# Dependency Resolution

CMSSM:



MSSM7:



Red: Model parameter translations

Blue: Precision calculations

Green: LEP rates+likelihoods

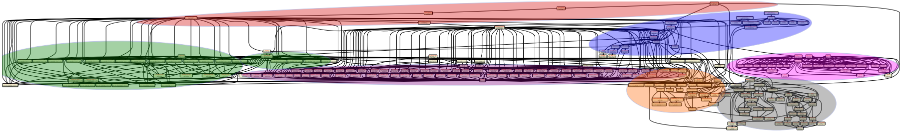
Purple: Decays

Orange: LHC observables and likelihoods

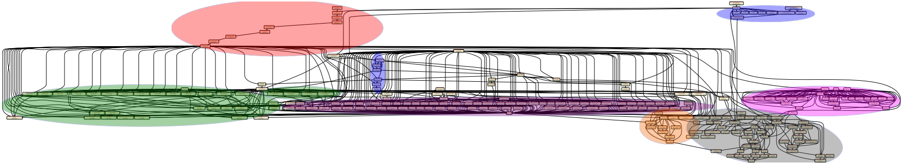
Grey: DM direct, indirect and relic density

# Dependency Resolution

CMSSM:



MSSM7:



Red: Model parameter translations

Blue: Precision calculations

Green: LEP rates+likelihoods

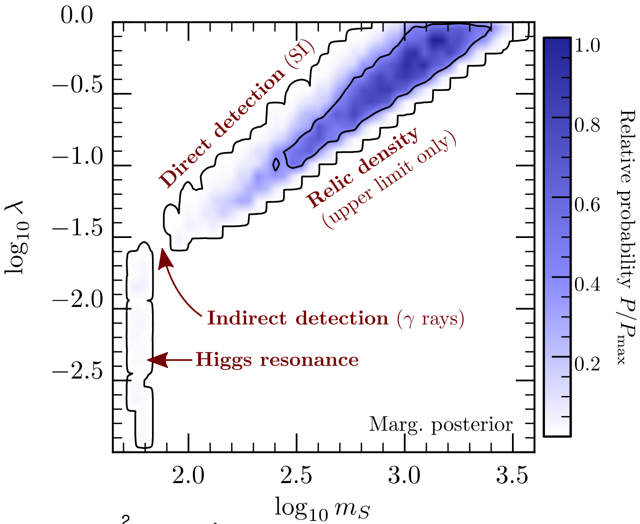
Purple: Decays

Orange: LHC observables and likelihoods

Grey: DM direct, indirect and relic density

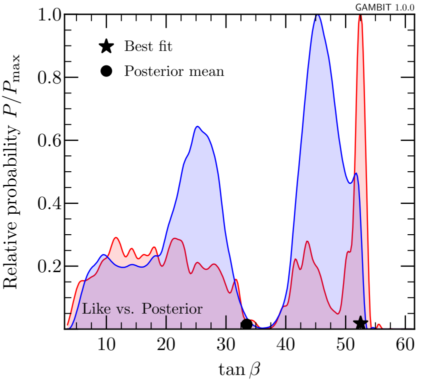
Pink: Flavour physics

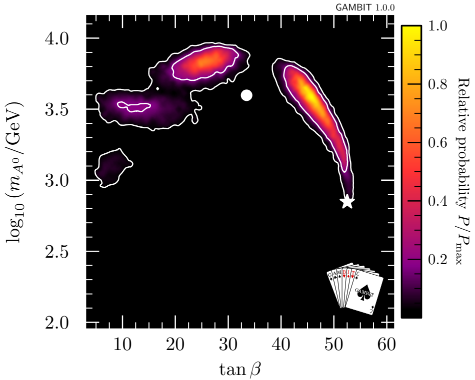
# Preliminary results: scalar singlet model




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

# Preliminary results: SUSY (CMSSM – *example only*)




**11 parameters**:

**4** $\times$ CMSSM + **6** $\times$ SM nuisances + **1** $\times$ astro nuis.($\rho_{\chi,\mathrm{local}}$)

# Outlook

- Neutrino telescope searches for DM give the strongest spin-dependent limits
  - highly complementary to other direct, indirect and collider searches
  - New IceCube analysis and nulike code allow full limit recast for any annihilating DM model
- GAMBIT is almost here:
  - → Global fits to many models for the first time
  - → Better global fits to familiar ones
  - → Highly modular, usable and extendible public code
  - → Faster, more complete and more consistent theory explorations + experimental analysis prototyping
  - → Initial series of 9 papers due out this summer
- After that:

  more models, more observables, more data!

  →→→ LHC Run 2!

# Backup Slides

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

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

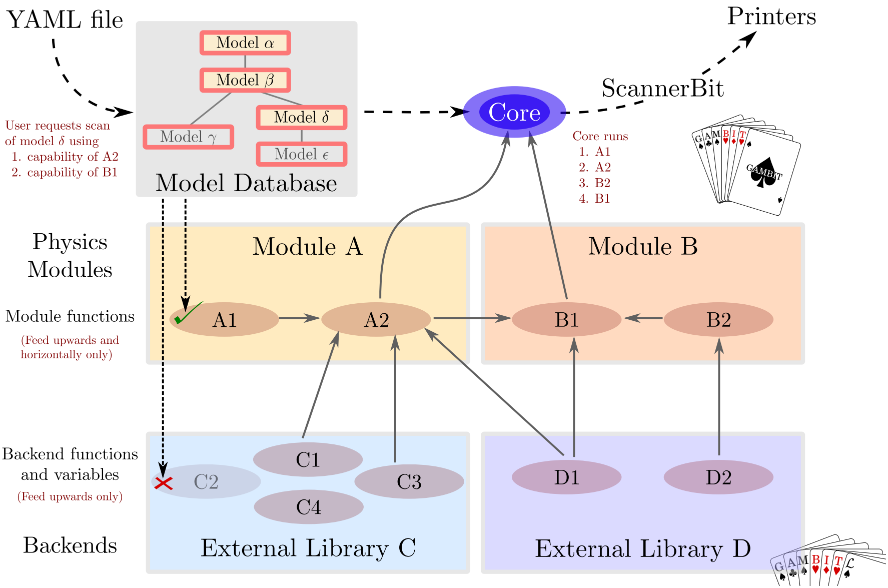
Goals:

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

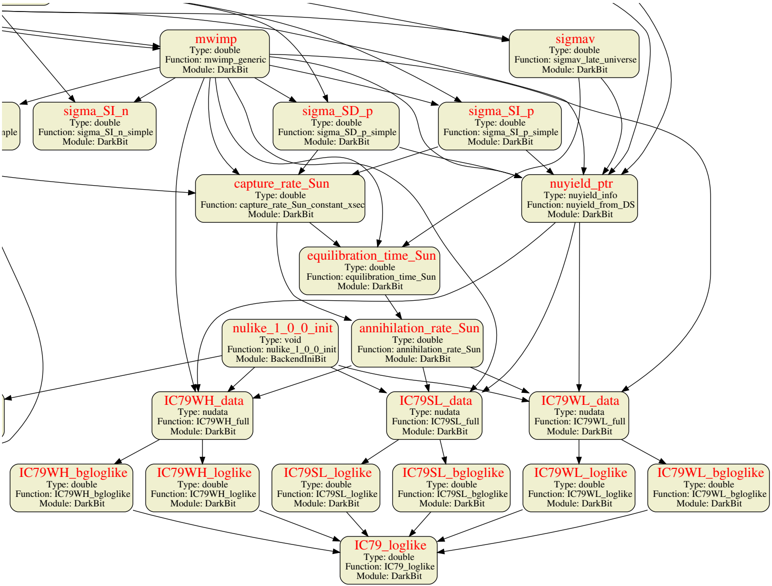
Why simple IN/OUT analyses are not enough. . .

- 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 $\Longrightarrow$ no statistical meaning.
- $\rightarrow$ statements about a theory's general ability to do one thing or another, based on such scans, are statistically invalid

# Functional overview of GAMBIT



# Dependency Resolution: a closer look



# GAMBIT: 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[1]
- **flexible output streams** (ASCII, databases, HDF5, ...)
- more more more...

[1] if a backend won't compile/crashes/steals your bank details, blame the authors (not us...except where we **are** the authors...)

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

- no simplified models, just faster direct simulation

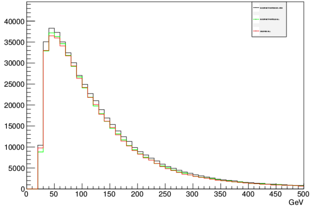
LHC likelihoods:

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

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on 4-vector smearing $\rightarrow$ matches DELPHES results very closely (but much faster)

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 – A few details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on 4-vector smearing $\rightarrow$ matches DELPHES results very closely (but much faster)
- Cross-sections: LO + LL from MC generator by default (fast NLO in works for SUSY)

# ColliderBit – A few details

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on 4-vector smearing $\rightarrow$ matches DELPHES results very closely (but much faster)
- Cross-sections: LO + LL from MC generator by default (fast NLO in works for SUSY)
- Analysis framework: custom event-level, independent of experiment or simulation

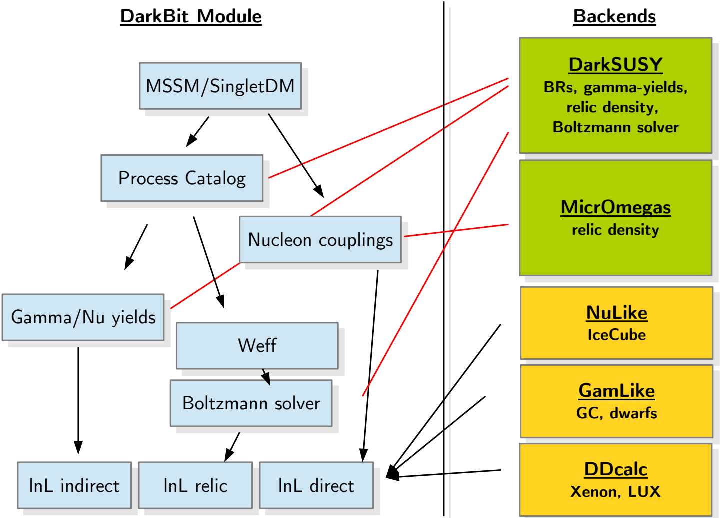
LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on 4-vector smearing $\longrightarrow$ matches DELPHES results very closely (but much faster)
- Cross-sections: LO + LL from MC generator by default (fast NLO in works for SUSY)
- Analysis framework: custom event-level, independent of experiment or simulation
- Likelihood: inline systematic error marginalisation (via `nulike`)

LHC likelihoods:

- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on 4-vector smearing $\rightarrow$ matches DELPHES results very closely (but much faster)
- Cross-sections: LO + LL from MC generator by default (fast NLO in works for SUSY)
- Analysis framework: custom event-level, independent of experiment or simulation
- Likelihood: inline systematic error marginalisation (via `nulike`)
- Initially shipping with:
  - ATLAS SUSY searches ($0\ell$, $0/1/2\ell$ $\tilde{t}$, $b$ jets + MET, $2/3\ell$ EW)
  - CMS multi-$\ell$ SUSY
  - CMS DM ($t$ pair + MET, mono-$b$, monojet)

# DarkBit – an incomplete overview



C++ library with simple interface to most relevant likelihood functions from Fermi LAT and IACTs

Particle physics input:

$$\frac{1}{m_\chi^2}\frac{d\sigma v}{dE}(v, E)$$

Output: lnL

Uncertainties in the DM distribution (or astrophysical foregrounds) are internally marginalized over.

Correct treatment of energy dispersion and spectral singularities (lines, virtual internal Bremsstrahlung, boxes).