Dark matter theory
Implications and future prospects for *Fermi*

Pat Scott

*Imperial College*

*London*
Even the simplest particle theories have *many* different consequences...
Even the simplest particle theories have *many* different consequences... theoretical
Even the simplest particle theories have *many* different consequences... theoretical, experimental
Even the simplest particle theories have *many* different consequences... theoretical, experimental, computational...
Even the simplest particle theories have *many* different consequences... theoretical, experimental, computational...

Gamma rays and *Fermi* are an important part of this, but need to be considered in the context of *real models* and constraints coming from *complementary probes*:

- realistic annihilation branching fractions
- direct detection
- collider searches
- relic density considerations
- related particles $\rightarrow$ precision/flavour physics
An illustrative example: lines from singlet dark matter

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

Claim: gamma-ray lines are an important constraint on scalar singlet dark matter

An illustrative example: lines from singlet dark matter

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

Claim: gamma-ray lines are an important constraint on scalar singlet dark matter

Problem: only true when

- singlet-Higgs coupling \( \lambda_{hs} \) is large
- Singlet mass \( m_S \sim m_{h^0}/2 \), on the resonance where \( S \) annihilates via SM Higgs \( h^0 \)
- but the relic density is very low when either of these is the case \( \rightarrow \) strongly suppresses the signal
An illustrative example: lines from singlet dark matter

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

Claim: gamma-ray lines are an important constraint on scalar singlet dark matter

Problem: only true when

- singlet-Higgs coupling \( \lambda_{hs} \) is large
- Singlet mass \( m_S \sim m_{h^0}/2 \), on the resonance where \( S \) annihilates via SM Higgs \( h^0 \)
- but the relic density is very low when either of these is the case \( \rightarrow \) strongly suppresses the signal

\[ \Rightarrow \] channel is never observable + therefore basically irrelevant for this model
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: $\gamma$, $\nu$, 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 → full experimental and theoretical treatment at the same time

Actually not so dissimilar to the LHC in this respect...

extra complications with neutrinos from capture-annihilation balance...
OK OK, enough pontificating and complaining already...
OK OK, enough pontificating and complaining already...  

So how to achieve all of this consistently? And how to be able to redo it all easily for many different dark matter models?
OK OK, enough pontificating and complaining already...

So how to achieve all of this consistently? And how to be able to redo it all easily for many different dark matter models?

⇒ GAMBIT
GAMBIT: The Global And Modular BSM Inference Tool

gambit.hepforge.org

- 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
F. Bernlochner, A. Buckley, P. Jackson, M. White

LHCb
M. Chrząszcz, N. Serra

Belle-II
F. Bernlochner, P. Jackson

Fermi-LAT
J. Edsjö, G. Martinez, P. Scott

CTA
C. Balázs, T. Bringmann, M. White

CMS
C. Rogan

IceCube
J. Edsjö, P. Scott

XENON/DARWIN
B. Farmer, R. Trotta

Theory

29 Members in 9 Experiments, 12 major theory codes, 11 countries

arXiv:1705.07908 (EPJC 2017)
DarkBit Module

MSSM/SingletDM

Process Catalog

Nucleon couplings

Gamma/Nu yields

Weff

Boltzmann solver

lnL indirect

lnL relic

Boltzmann solver

lnL direct

Backends

DarkSUSY
BRs, gamma-yields, relic density, Boltzmann solver

MicrOmegas
relic density

NuLike
IceCube

GamLike
GC, dwarfs

DDcalc
Xenon, LUX

Pat Scott – Oct 19 2017 – 7th Fermi Symposium

Dark matter theory: implications and future prospects for Fermi
C++ library with simple interface to most relevant likelihood functions from Fermi LAT and IACTs

Particle physics input:

\[
\frac{1}{m^2} \frac{d\sigma v}{dE}(v, E)
\]

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).
Scalar singlet DM \((m_S, \lambda h_S + 13 \text{ nuisances})\) (arXiv:1705.07931)

Simplest BSM example: \(\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda h_S}{2} S^2 H^\dagger H + \ldots\)

All dark matter signals consistently scaled for predicted abundance
Scalar singlet DM \((m_S, \lambda_{hS} + 13\) nuisances\) (arXiv:1705.07931)

Simplest BSM example: \(\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H + \ldots\)

All dark matter signals consistently scaled for predicted abundance
Scalar singlet DM \( (m_S, \lambda_{hS} + 13 \text{ nuisances}) \) (arXiv:1705.07931)

Simplest BSM example: \( \mathcal{L}_S = -\frac{\mu^2_S}{2} S^2 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H + \ldots \)

All dark matter signals consistently scaled for predicted abundance
Scalar singlet DM \((m_S, \lambda_{hS} + 13 \text{ nuisances})\) (arXiv:1705.07931)

Simplest BSM example: \(\mathcal{L}_S = -\frac{\mu^2_S}{2} S^2 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H + \ldots\)

All dark matter signals consistently scaled for predicted abundance
Scalar singlet DM ($m_S$, $\lambda_{hS} + 13$ nuisances)  (arXiv:1705.07931)

Simplest BSM example: $\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hS}}{2} S^2 H^\dagger H + \ldots$

All dark matter signals consistently scaled for predicted abundance
An advance health warning about CTA projections

CTA sensitivity will be dominantly determined by:

- statistical variation of diffuse emission
- systematics (differential error over FoV on effective area, event reconstruction, atmospheric conditions)

Official CTA projections assume 0% systematic errors & 500 hr exp. ...with 0.3% sys err, CTA is already systematic-limited @200 hr...
GUT-scale MSSM (CMSSM) (arXiv:1705.07935)

- $m_0$, $m_{1/2}$, $A_0$, tan $\beta$ + 5 nuisances
- $H/A^0$ funnel, $\chi^{\pm}$ co-annihilation, $\tilde{t}$ co-annihilation
- $\tilde{\tau}$ co-annihilation now ruled out
- Includes LUX 2016, Panda-X + direct simulation of LHC Run 1 & Run 2 limits.
GUT-scale MSSM (CMSSM)

- $m_0$, $m_{1/2}$, $A_0$, tan $\beta$ + 5 nuisances
- $H/A^0$ funnel, $\chi^\pm$ co-annihilation, $\tilde{t}$ co-annihilation
- $\tilde{\tau}$ co-annihilation now ruled out
- $\tilde{t}$ co-annihilation tough to get at with DD, ID or LHC – but some hope with LHC for seeing light stops at the ‘tip’
m_{0}, m_{H}, m_{\frac{1}{2}}, A_0, \tan \beta + 5 nuisances

H/A^0 funnel, \chi^\pm co-annihilation, \tilde{t} co-annihilation + \tilde{\tau} co-annihilation due to additional freedom to fit m_h

Includes LUX 2016, Panda-X + direct simulation of LHC Run 1 & Run 2 limits.
• $m_0$, $m_{H_u}$, $m_{H_d}$, $m_{\tilde{t}^\pm}$, $A_0$, $\tan \beta + 5$ nuisances

• $H/A^0$ funnel, $\chi^\pm$ co-annihilation, $\tilde{t}$ co-annihilation, $\tilde{\tau}$ co-annihilation

• Includes LUX 2016, Panda-X + direct simulation of LHC Run 1 & Run 2 limits.
Weak-scale MSSM (MSSM-7)

$$m_{\tilde{f}}, M_2, A_u, A_d, m_{H_u}, m_{H_d}, \tan \beta + 5 \text{ nuisances}$$

- $H/A^0$ funnel, $h/Z$ funnel, $\chi^\pm$ co-annihilation, $\tilde{t}/\tilde{b}$ co-annihilation
- Includes LUX 2016, Panda-X + direct simulation of LHC Run 1 & Run 2 limits.
Summary

- Scalar singlet restricted to ‘resonant triangle’ and high-coupling/high-mass solutions
- Fermi dwarfs place limits in the region of the Higgs resonance
- Dwarfs are providing significant constraints on some versions of SUSY + will do more
- Heavy Higgs funnel models are especially promising targets for Fermi going forwards
- CTA will also probe many of the same models
- $\gamma$-rays are still ~complementary with XENONnT/LZ/Panda-X, but analyses will need to be fast!

- GAMBIT results, samples, run files, best fits, benchmarks, etc are all available to download from Zenodo:
  www.zenodo.org/communities/gambit-official/
- GAMBIT code is public: gambit.hepforge.org
Backup slides
Some more scalar singlet DM \((m_S, \lambda h_S + 13)\) (arXiv:1705.07931)

\[\Omega h^2 = 0.119\]

\[\lambda h_S > 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\]

Profile likelihood ratio \(\Lambda = \frac{L}{L_{\text{max}}}\)

\[\log_{10}(\sigma_{SI}/\text{cm}^2)\]

\[\log_{10}(f \cdot \sigma_{SI}/\text{cm}^2)\]

\[\log_{10}(f^2 \cdot \langle \sigma v \rangle_0 / \text{cm}^3 \text{s}^{-1})\]

Fermi-LAT \((\tau^+\tau^-)\), Fermi-LAT \((b\bar{b})\)

PandaX 2016, LUX 2016
Some more CMSSM

Profile likelihood ratio $\Lambda = L/L_{\text{max}}$

$\tilde{t}_1$ co-annihilation

$A/H$ funnel

$\tilde{\chi}_1^\pm$ co-annihilation
Some more CMSSM

Profile likelihood ratio \( \Lambda = \frac{L}{L_{\text{max}}} \)

-\( t_1 \) co-annihilation
-\( A/H \) funnel
-\( \tilde{\chi}_1^\pm \) co-annihilation

Pat Scott – Oct 19 2017 – 7th Fermi Symposium

Dark matter theory: implications and future prospects for Fermi
Some more CMSSM

Pat Scott – Oct 19 2017 – 7th Fermi Symposium

Dark matter theory: implications and future prospects for Fermi
Some more CMSSM

![Plot of log10 of f_\chi_0^0 / (σSI p / cm^2) vs. m_\chi_0^0 (GeV)]

- Red: \tilde{t}_1 co-annihilation
- Orange: A/H funnel
- Yellow: \tilde{\chi}_1^\pm co-annihilation

- Profile likelihood ratio Λ = L/L_{max}

- Neutrino floor

- Best fit

LUX 2016
XENON1T (2ty)
XENONnT/LZ (20ty)
DARWIN (200ty)
Some more MSSM-7

$\Lambda = \frac{L}{L_{\text{max}}}$

$M_1$ (GeV)

$\mu$ (GeV)

- $\tilde{t}_1$ co-annihilation
- $A/H$ funnel
- $\tilde{\chi}_1^\pm$ co-annihilation
- $\tilde{b}_1$ co-annihilation
- $h/Z$ funnel

Pat Scott – Oct 19 2017 – 7th Fermi Symposium

Dark matter theory: implications and future prospects for Fermi
Some more MSSM-7

Profile likelihood ratio $\Lambda = \frac{L}{L_{\text{max}}}$

$\Omega h^2 = 0.119$

$\log_{10}(\Omega \chi h^2)$ vs $m_{\tilde{\chi}_0}$ (GeV)

- $\tilde{t}_1$ co-annihilation
- $A/H$ funnel
- $\tilde{\chi}_1^\pm$ co-annihilation
- $\tilde{b}_1$ co-annihilation
- $h/Z$ funnel
Some more MSSM-7

![Plot of MSSM7](image)

- $t_1$ co-annihilation
- $A/H$ funnel
- $\tilde{\chi}_1^\pm$ co-annihilation
- $\tilde{b}_1$ co-annihilation
- $h/Z$ funnel

Profile likelihood ratio $\Lambda = \frac{L}{L_{\text{max}}}$

- LUX 2016
- XENON1T (2ty)
- XENONnT/LZ (20ty)
- DARWIN (200ty)

Neutrino floor

$\log_{10}(f \cdot \sigma_{SI}/\text{cm}^2)$
Some more MSSM-7

(Fermi $\tau^+\tau^-$, Fermi $b\bar{b}$, Fermi $b\bar{b}$ 15y, 60 dwarfs, CTA Galactic Halo $b\bar{b}$ 500h)

$\log_{10}(f^2 \cdot \langle \sigma v \rangle_{\text{cm}^3 \text{s}^{-1}})$

Profile likelihood ratio $\Lambda = L/L_{\text{max}}$

$0 \quad 1000 \quad 2000 \quad 3000$

$m_{\tilde{\chi}_0^1} (\text{GeV})$

Fermi $\tau^+\tau^-$

Fermi $b\bar{b}$

Fermi $b\bar{b}$ 15y, 60 dwarfs

CTA Galactic Halo $b\bar{b}$ 500h

$\tilde{t}_1$ co-annihilation

$A/H$ funnel

$\tilde{\chi}_1^\pm$ co-annihilation

$\tilde{b}_1$ co-annihilation

$h/Z$ funnel

Pat Scott – Oct 19 2017 – 7th Fermi Symposium

Dark matter theory: implications and future prospects for Fermi
Functional overview of GAMBIT

User requests scan of model δ using:
1. capability of A2
2. capability of B1

Core runs:
1. A1
2. A2
3. B2
4. B1

Model Database

YAML file

Printers

Physics Modules

Module functions
(Feed upwards and horizontally only)

Backend functions and variables
(Feed upwards only)

Backends

External Library C

External Library D

Module A

Module B

Pat Scott – Oct 19 2017 – 7th Fermi Symposium

Dark matter theory: implications and future prospects for Fermi
Models are defined by their parameters and relations to each other.

Models can inherit from (be subspaces of) **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.

⇒ maximum safety, maximum reuse.
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, so backend functions are tagged according to what they calculate.

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 Scott – Oct 19 2017 – 7th Fermi Symposium**

Dark matter theory: implications and future prospects for Fermi

---

<table>
<thead>
<tr>
<th>BACKENDS</th>
<th>VERSION</th>
<th>PATH TO LIB</th>
<th>STATUS</th>
<th>#FUNC</th>
<th>#TYPES</th>
<th>#CTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDCalc0</td>
<td>0.9</td>
<td>Backends/installed/DDCalc/0.8/libDDCalc0.so</td>
<td>OK</td>
<td>62</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DarkSUSY</td>
<td>5.1.1</td>
<td>Backends/installed/DarkSUSY/3.1.1/lib/libdarksusy.so</td>
<td>OK</td>
<td>68</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FastSim</td>
<td>1.0</td>
<td>Backends/installed/fastsim/1.0/libfastsim.so</td>
<td>absent/broken</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FeynHiggs</td>
<td>2.11</td>
<td>Backends/installed/FeynHiggs/2.11.2/lib/libFeynHiggs.so</td>
<td>OK</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HiggsBounds</td>
<td>4.2.1</td>
<td>Backends/installed/HiggsBounds/4.2.1/lib/libhiggsbounds.so</td>
<td>OK</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HiggsSignals</td>
<td>1.4</td>
<td>Backends/installed/HiggsSignals/1.4.0/lib/libhiggsSignals.so</td>
<td>OK</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LibFarrayTest</td>
<td>1.0</td>
<td>Backends/examples/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/examples/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/examples/libfirst.so</td>
<td>OK</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LibFortran</td>
<td>1.0</td>
<td>Backends/examples/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>Backends/installed/micromega/3.5.5/MSSM/MSSM/libmicromega.so</td>
<td>OK</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MicrOmegasSingletDM</td>
<td>3.5.5</td>
<td>Backends/installed/micromega/3.5.5/SingletDM/SingletDM/libmicromega.so</td>
<td>OK</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pythia</td>
<td>8.186</td>
<td>Backends/installed/Pythia/8.186/lib/libpythia8.so</td>
<td>absent/broken</td>
<td>0</td>
<td>27</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>8.209</td>
<td>Backends/installed/Pythia/8.209/lib/libpythia8.so</td>
<td>OK</td>
<td>0</td>
<td>28</td>
<td>107</td>
</tr>
<tr>
<td>SUSYPOPE</td>
<td>0.2</td>
<td>no path in config/backend_locations.yaml</td>
<td>absent/broken</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SUSY_HIT</td>
<td>1.5</td>
<td>Backends/installed/SUSY-HIT/1.5/lib/susyhit.so</td>
<td>OK</td>
<td>55</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SuperIso</td>
<td>3.4</td>
<td>Backends/installed/SuperIso/3.4/lib/superiso.so</td>
<td>OK</td>
<td>32</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>gamLike</td>
<td>1.0.0</td>
<td>Backends/installed/gamlike/1.0.0/lib/gamlike.so</td>
<td>OK</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>nulike</td>
<td>1.0.0</td>
<td>Backends/installed/nulike/1.0.0/lib/nulike.so</td>
<td>OK</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
## Backends: mix and match

Pat Scott – Oct 19 2017 – 7th Fermi Symposium

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

<table>
<thead>
<tr>
<th>BACKENDS</th>
<th>VERSION</th>
<th>PATH TO LIB</th>
<th>STATUS</th>
<th>#FUNC</th>
<th>#TYPES</th>
<th>#CTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDCalc0</td>
<td>0.9</td>
<td>Backends/installed/DDCalc/0.8/libDDCalc0.so</td>
<td>OK</td>
<td>62</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DarkSUSY</td>
<td>5.1.1</td>
<td>Backends/installed/DarkSUSY/3.1.1/lib/libdarksusy.so</td>
<td>OK</td>
<td>68</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FastSim</td>
<td>1.0</td>
<td>Backends/installed/fastsim/1.0/libfastsim.so</td>
<td>absent/broken</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FeynHiggs</td>
<td>2.11</td>
<td>Backends/installed/FeynHiggs/2.11.2/lib/libFH.so</td>
<td>OK</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HiggsBounds</td>
<td>4.2.1</td>
<td>Backends/installed/HiggsBounds/4.2.1/lib/libhiggsbounds.so</td>
<td>OK</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HiggsSignals</td>
<td>1.4</td>
<td>Backends/installed/HiggsSignals/1.4.0/lib/libhiggsSignals.so</td>
<td>OK</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LibFarrayTest</td>
<td>1.0</td>
<td>Backends/examples/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/examples/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/examples/libfirst.so</td>
<td>OK</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LibFortran</td>
<td>1.0</td>
<td>Backends/examples/libfortran.so</td>
<td>OK</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MicrOmegaes</td>
<td>3.5.5</td>
<td>Backends/installed/micrOmegaes/3.5.5/MSSM/MSSM/libmicrOmegaes.so</td>
<td>OK</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MicrOmegaesSingletDM</td>
<td>3.5.5</td>
<td>Backends/installed/micrOmegaes/3.5.5/SingletDM/SingletDM/libmicrOmegaes.so</td>
<td>OK</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pythia</td>
<td>8.186</td>
<td>Backends/installed/Pythia/8.186/lib/libpythia8.so</td>
<td>absent/broken</td>
<td>0</td>
<td>27</td>
<td>185</td>
</tr>
<tr>
<td></td>
<td>8.209</td>
<td>Backends/installed/Pythia/8.209/lib/libpythia8.so</td>
<td>OK</td>
<td>0</td>
<td>28</td>
<td>107</td>
</tr>
<tr>
<td>SUSYPOPE</td>
<td>0.2</td>
<td>no path in config/backend_locations.yaml</td>
<td>absent/broken</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SUSY_HIT</td>
<td>1.5</td>
<td>Backends/installed/SUSY-HIT/1.5/lib/susyhit.so</td>
<td>OK</td>
<td>55</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SuperIso</td>
<td>3.4</td>
<td>Backends/installed/SuperIso/3.4/lib/superiso.so</td>
<td>OK</td>
<td>32</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>gamLike</td>
<td>1.0.0</td>
<td>Backends/installed/gamLike/1.0.0/lib/gamLike.so</td>
<td>OK</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>nulike</td>
<td>1.0.0</td>
<td>Backends/installed/nulike/1.0.0/lib/libnulike.so</td>
<td>OK</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

MSSM7:

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

MSSM7:

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

MSSM7:

Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods
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
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
Extensive scanner tests on scalar singlet model with different numbers of nuisance parameters

Diver scales far better with dimensionality than MultiNest or other scanners
Adding a new module function is easy:

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

```
#define MODULE FlavBit // A tasty GAMBIT module.
START_MODULE

#define CAPABILITY Rmu // Observable: BR(K->mu nu)/BR(pi->mu nu)
START_CAPABILITY

#define FUNCTION SI_Rmu // Name of a function that can compute Rmu
START_FUNCTION(double) // Function computes a double precision result
BACKEND_REQ(Kmunu_pimunu, (my_tag), double, (const parameters*)) // Needs function from a backend
BACKEND_OPTION( (SuperIso, 3.6), (my_tag) ) // Backend must be SuperIso 3.6
DEPENDENCY(SuperIso_modelinfo, parameters) // Needs another function to calculate SuperIso info
ALLOW_MODELS(MSSM63atQ, MSSM63atMGUT) // Works with weak/GUT-scale MSSM and descendents

#undef FUNCTION
#undef CAPABILITY
```

2. Write the function as a standard C++ function
   (one argument: the result)
Expansion: adding new models

1. Add the model to the **model hierarchy**:
   - Choose a model name, and declare any **parent model**
   - Declare the model’s parameters
   - Declare any **translation function** to the parent model

```c
#define MODEL NUHM1
#define PARENT NUHM2
START_MODEL
  DEFINEPARS(M0,M12,mH,A0,TanBeta,SignMu)
  INTERPRET_AS_PARENT_FUNCTION(NUHM1_to_NUHM2)
#undef PARENT
#undef MODEL
```

2. Write the translation function as a standard C++ function:

```c
void MODEL_NAMESPACE::NUHM1_to_NUHM2 (const ModelParameters &myP, ModelParameters &targetP) {
  // Set M0, M12, A0, TanBeta and SignMu in the NUHM2 to the same values as in the NUHM1
  targetP.setValues(myP,false);
  // Set the values of mHu and mHd in the NUHM2 to the value of mH in the NUHM1
  targetP.setValue("mHu", myP["mH"]);
  targetP.setValue("mHd", myP["mH"]);
}
```

3. If needed, declare that existing module functions work with the new model, or add new functions that do.
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)
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)
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)
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)
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)
Other nice technical features

- **Scanners**: Nested sampling, differential evolution, MCMC, t-walk...
- Mixed-mode **MPI + openMP** parallelisation, mostly automated → 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 backends¹
- **Flexible output streams** (ASCII, databases, HDF5, ...)
- Available as docker plugin or vagrant virtual machine
- More more more more...

¹If a backend won’t compile/crashes/kills your cat, blame the authors (not us...except where we are the authors...)
ColliderBit 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, CP info, 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
ColliderBit details

LHC likelihoods:

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

Leading electron $p_T$ distribution (CMSSM example):
- red: detector-level simulation with DELPHES
- green: 4-vector smearing with GAMBIT
- blue: truth-level distribution
ColliderBit details

LHC likelihoods:

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

LHC likelihoods:

- **MC generation**: Pythia8 parallelised with OpenMP + other speed tweaks
- **Detector simulation**: fast simulation based on 4-vector smearing → 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 → 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)
ColliderBit details

LHC likelihoods:

- **MC generation**: Pythia8 parallelised with OpenMP + other speed tweaks
- **Detector simulation**: fast simulation based on 4-vector smearing → 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 (0ℓ, 0/1/2ℓ ℓ̃, b jets + MET, 2/3ℓ EW)  
  - CMS multi-ℓ SUSY  
  - CMS DM (t pair + MET, mono-ℓ, monojet)  
  - ATLAS + CMS Run II 0ℓ  
  - refresh coming with more Run II analyses shortly