Searches for dark matter and new physics with GAMBIT

Pat Scott

Imperial College London

on behalf of the GAMBIT Collaboration

Slides at: www.imperial.ac.uk/people/p.scott/research.html
GAMBIT: gambit.hepforge.org

Outline

1. Global fits
   - Why?
   - GAMBIT
2. Status updates for key theories
   - Higgs-portal dark matter
   - Supersymmetry
   - Axions and ALPs

Outline

1. Global fits
   - Why?
   - GAMBIT
2. Status updates for key theories
   - Higgs-portal dark matter
   - Supersymmetry
   - Axions and ALPs

Combining searches I

Question

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

Question

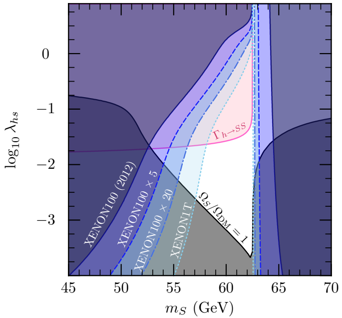
How do we know which models are in and which are out?

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

(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?

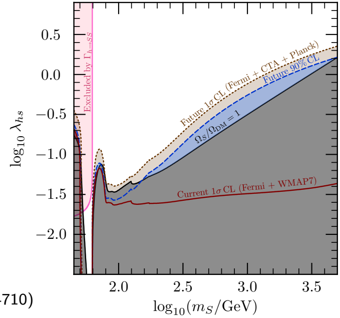
That's all well and good if there are only 2 parameters and few searches. . .

Question

What if there are many different constraints?

Answer

Combine constraints in a statistically valid way
→ 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)

→ global fits

Why don't we just pick models randomly?

Old-style scans: random, IN/OUT Not appropriate.

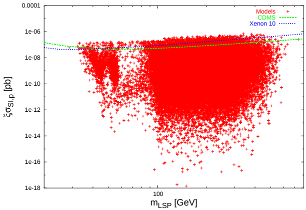
- no indication of goodness of fit $\implies$ no parameter estimation
- no concept of how indicative of the parameter space your points actually are, i.e. how much of the space you sampled, nor where 'most' of the 'good' theory space resides.
- $\implies$ impossible to generalise from points to regions or whole theory
- attempts to make probabilistic statements with such scans are statistically invalid

Why don't we just pick models randomly?

Old-style scans: random, IN/OUT Not appropriate.

- no indication of goodness of fit $\implies$ no parameter estimation
- no concept of how indicative of the parameter space your points actually are, i.e. how much of the space you sampled, nor where 'most' of the 'good' theory space resides.
- $\implies$ impossible to generalise from points to regions or whole theory
- attempts to make probabilistic statements with such scans are statistically invalid

Example:



"Values are possible"

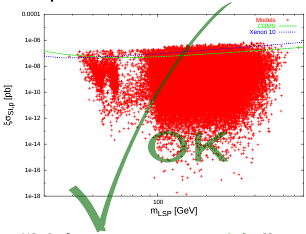
Berger, Gainer, Hewett & Rizzo, *JHEP* 2009

Why don't we just pick models randomly?

Old-style scans: random, IN/OUT Not appropriate.

- no indication of goodness of fit $\Longrightarrow$ no parameter estimation
- no concept of how indicative of the parameter space your points actually are, i.e. how much of the space you sampled, nor where 'most' of the 'good' theory space resides.
- $\Longrightarrow$ impossible to generalise from points to regions or whole theory
- attempts to make probabilistic statements with such scans are statistically invalid

Example:



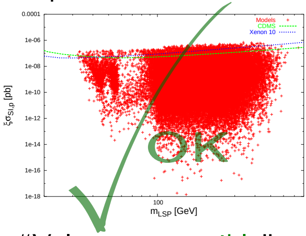
"Values are possible"

Berger, Gainer, Hewett & Rizzo, *JHEP* 2009

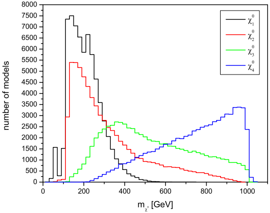
Old-style scans: random, IN/OUT Not appropriate.

- no indication of goodness of fit $\implies$ no parameter estimation
- no concept of how indicative of the parameter space your points actually are, i.e. how much of the space you sampled, nor where 'most' of the 'good' theory space resides.
- $\implies$ impossible to generalise from points to regions or whole theory
- attempts to make probabilistic statements with such scans are statistically invalid

Example:



"Values are possible"



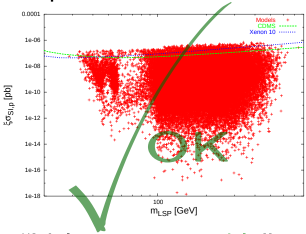
"Values are probable"

Berger, Gainer, Hewett & Rizzo, *JHEP* 2009

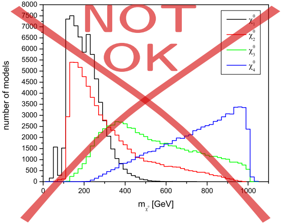
Old-style scans: random, IN/OUT Not appropriate.

- no indication of goodness of fit $\Longrightarrow$ no parameter estimation
- no concept of how indicative of the parameter space your points actually are, i.e. how much of the space you sampled, nor where 'most' of the good' theory space resides.
- $\Longrightarrow$ impossible to generalise from points to regions or whole theory
- attempts to make probabilistic statements with such scans are statistically invalid

Example:



"Values are possible"



"Values are probable"

Berger, Gainer, Hewett & Rizzo, *JHEP* 2009

Q. OK, so what about combining *all* the constraints consistently then?

Q. OK, so what about combining *all* the constraints consistently then?

Q. And varying SM parameters within their allowed ranges?

Q. OK, so what about combining *all* the constraints consistently then?

Q. And varying SM parameters within their allowed ranges?

Q. And the nuclear uncertainties?

Q. OK, so what about combining *all* the constraints consistently then?

Q. And varying SM parameters within their allowed ranges?

Q. And the nuclear uncertainties? And astrophysical models?

Q. OK, so what about combining *all* the constraints consistently then?

Q. And varying SM parameters within their allowed ranges?

Q. And the nuclear uncertainties? And astrophysical models?

Q. And dealing properly with the statistics of big parameter spaces?

Q. OK, so what about combining *all* the constraints consistently then?

Q. And varying SM parameters within their allowed ranges?

Q. And the nuclear uncertainties? And astrophysical models?

Q. And dealing properly with the statistics of big parameter spaces?

Q. And doing that for lots of different theories?

Q. OK, so what about combining *all* the constraints consistently then?

A. GAMBIT

Q. And varying SM parameters within their allowed ranges?

A. GAMBIT

Q. And the nuclear uncertainties? And astrophysical models?

A. GAMBIT

Q. And dealing properly with the statistics of big parameter spaces?

A. GAMBIT

Q. And doing that for lots of different theories?

A. GAMBIT

Outline

1. Global fits
   - Why?
   - GAMBIT
2. Status updates for key theories
   - Higgs-portal dark matter
   - Supersymmetry
   - Axions and ALPs

GAMBIT: The Global And Modular BSM Inference Tool

gambit.hepforge.org   EPJC **77** (2017) 784   arXiv:1705.07908

- 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
- Fast definition of new datasets and theories
- Plug and play scanning, physics and likelihood packages

Members of: ATLAS, Belle-II, CMS, CTA, *Fermi*-LAT, DARWIN, IceCube, LHCb, SHiP, XENON

Authors of: DarkSUSY, DDCalc, Diver, FlexibleSUSY, gamlike, GM2Calc, IsaJet, nulike, PolyChord, Rivet, SOFTSUSY, SuperIso, SUSY-AI, WIMPSim

Collaborators:
Peter Athron, Csaba Balázs, Ankit Beniwal, Florian Bernlochner, Sanjay Bloor, Torsten Bringmann, Andy Buckley, Eliel Camargo-Molina, Marcin Chrząszc, Jan Conrad, Jonathan Cornell, Matthias Danninger, Tom Edwards, Joakim Edsjö, Ben Farmer, Andrew Fowlie, Tomás Gonzalo, Will Handley, Sebastian Hoof, Selim Hotinli, Felix Kahlhoefer, Suraj Krishnamurthy, Anders Kvellestad, Julia Harz, Paul Jackson, Tong Li, Greg Martinez, Nazilla Mahmoudi, James McKay, Are Raklev, Janina Renk, Chris Rogan, Roberto Ruiz de Austri, Patrick Stoecker, Roberto Trotta, Pat Scott, Nicola Serra, Daniel Steiner, Puwen Sun, Aaron Vincent, Christoph Weniger, Sebastian Wild, Martin White, Yang Zhang

40+ participants in 10 Experiments & 14 major theory codes

Modules

Physics modules

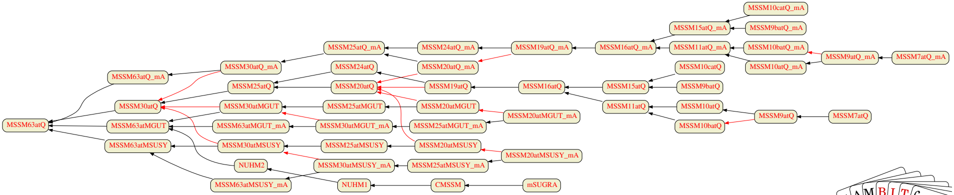
- DarkBit – dark matter observables (relic density, direct + indirect detection) (EPJC, arXiv:1705.07920)
- ColliderBit – collider observables inc. Higgs + SUSY searches from ATLAS, CMS + LEP (EPJC, arXiv:1705.07919)
- FlavBit – flavour physics inc. $g-2$, $b \to s\gamma$, B decays (new channels, angular obs., theory uncerts, LHCb likelihoods) (EPJC, arXiv:1705.07933)
- SpecBit – generic BSM spectrum object, providing RGE running, masses, mixings, etc via interchangeable interfaces to different RGE codes (EPJC, arXiv:1705.07936)
- DecayBit – decay widths for all relevant SM & BSM particles (EPJC, arXiv:1705.07936)
- PrecisionBit – SM likelihoods, precision BSM tests (W mass, $\Delta\rho$ etc) (EPJC, arXiv:1705.07936)

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

Hierarchical Model Database (EPJC, arXiv:1705.07908)

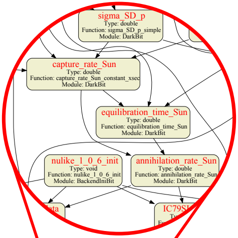
- Models are defined by their parameters and relations to each other
- Models can inherit from (be subspaces of) **parent models**
- Points in child models can be **automatically translated** to ancestor models
- **Friend models** also allowed (cross-family translation)
- Model dependence of every function/observable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

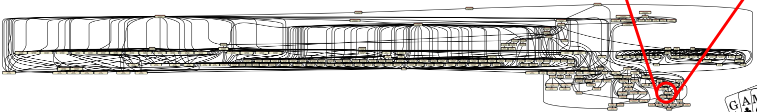
$\implies$ maximum safety, maximum reuse



Anatomy of a GAMBIT run (EPJC, arXiv:1705.07908)

- User chooses a model to scan, which observables to include, and the scanning method
- GAMBIT constructs a **dependency tree**
  1. Identifies which functions and inputs are needed to compute the requested observables
  2. Obeys **rules** at each step: allowed models, allowed backends, constraints from input file, etc
     $\rightarrow$ tree constitutes a directed acyclic graph
  3. Uses graph-theoretic methods to 'solve' the graph to determine function evaluation order
- GAMBIT scans the parameter space by calling the necessary module and backend functions in the optimal order, for each parameter point



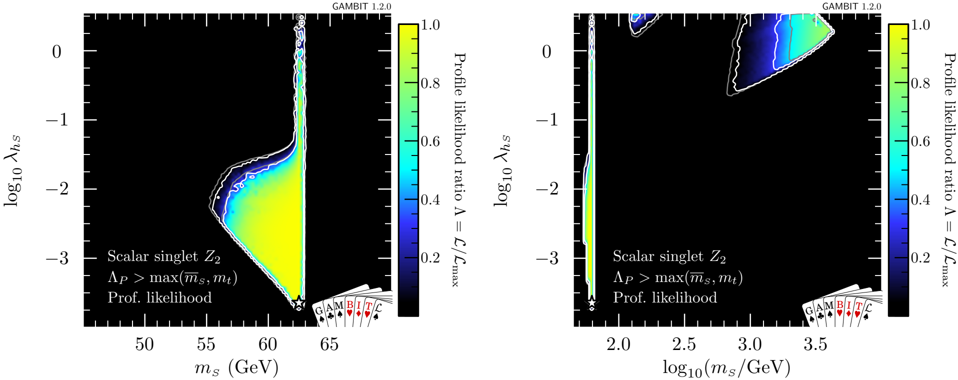


Outline

1. Global fits
   - Why?
   - GAMBIT
2. Status updates for key theories
   - Higgs-portal dark matter
   - Supersymmetry
   - Axions and ALPs

$\mathbb{Z}_2$-symmetric scalar singlet DM

(EPJC, arXiv:1806.11281)



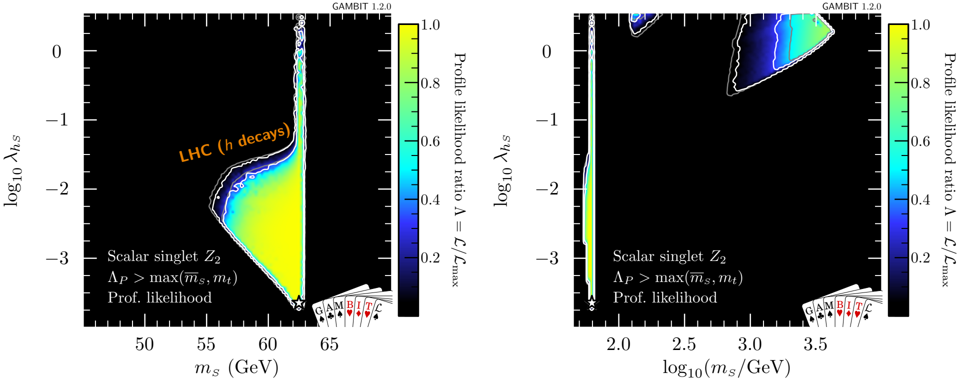
White = 1σ, 2σ with XENON1T 2018

Grey = 1σ, 2σ with XENON1T 2017

- 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
- Scan includes 3 singlet parameters $(m_S, \lambda_{hS}, \lambda_S)$ + 7 nuisances

(Milky Way halo model, nuclear matrix elements, Standard Model params.)

$\mathbb{Z}_2$-symmetric scalar singlet DM (EPJC, arXiv:1806.11281)

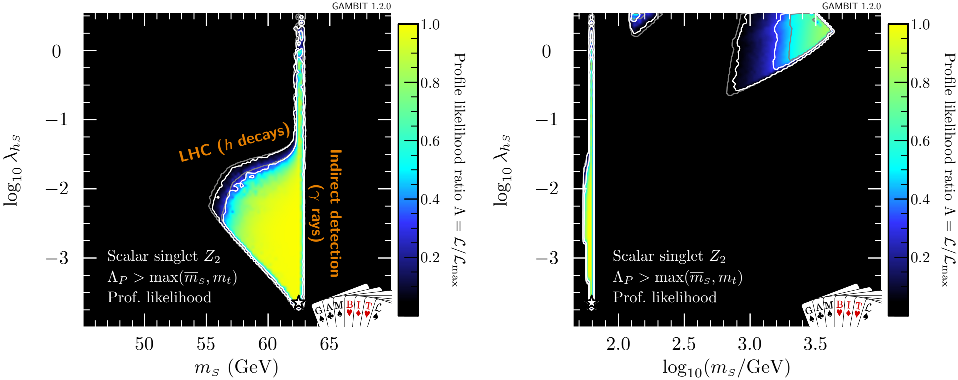


White = 1σ, 2σ with XENON1T 2018

Grey = 1σ, 2σ with XENON1T 2017

- 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
- Scan includes 3 singlet parameters $(m_S, \lambda_{hS}, \lambda_S)$ + 7 nuisances

(Milky Way halo model, nuclear matrix elements, Standard Model params.)



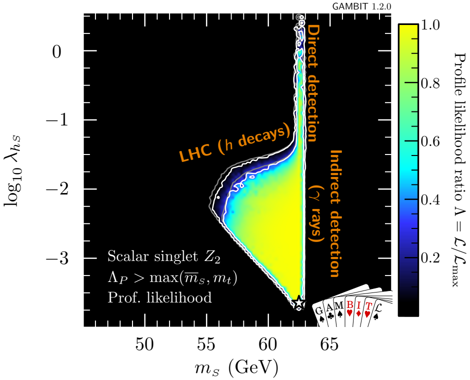
White = 1σ, 2σ with XENON1T 2018
Grey = 1σ, 2σ with XENON1T 2017

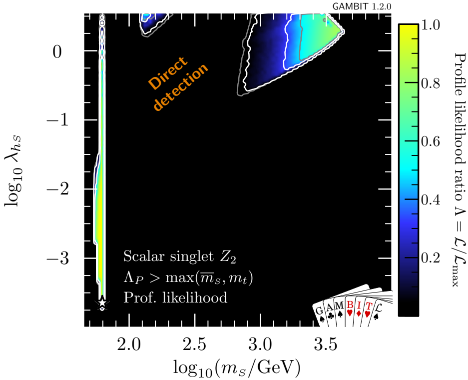
- Simplest BSM example: $\mathcal{L}_S = -\frac{\mu_S^2}{2}S^2 - \frac{\lambda_{hs}}{2}S^2H^\dagger H + \ldots$
- All dark matter signals consistently scaled for predicted abundance
- Scan includes 3 singlet parameters $(m_S, \lambda_{hS}, \lambda_S)$ + 7 nuisances

(Milky Way halo model, nuclear matrix elements, Standard Model params.)

$\mathbb{Z}_2$-symmetric scalar singlet DM

(EPJC, arXiv:1806.11281)



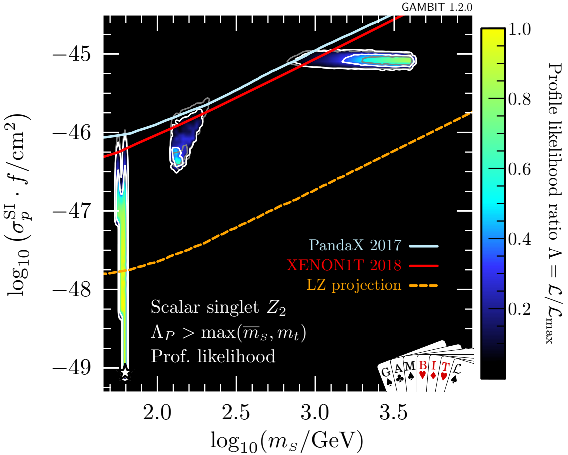


White = 1σ, 2σ with XENON1T 2018

Grey = 1σ, 2σ with XENON1T 2017

- Simplest BSM example: $\mathcal{L}_S = -\frac{\mu_S^2}{2} S^2 - \frac{\lambda_{hs}}{2} S^2 H^\dagger H + \dots$
- All dark matter signals consistently scaled for predicted abundance
- Scan includes 3 singlet parameters $(m_S, \lambda_{hS}, \lambda_S)$ + 7 nuisances

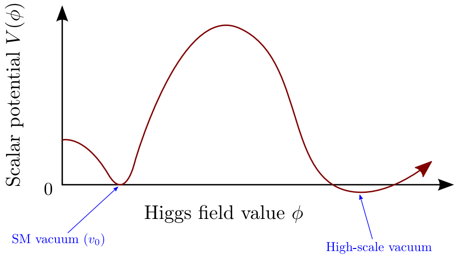
(Milky Way halo model, nuclear matrix elements, Standard Model params.)



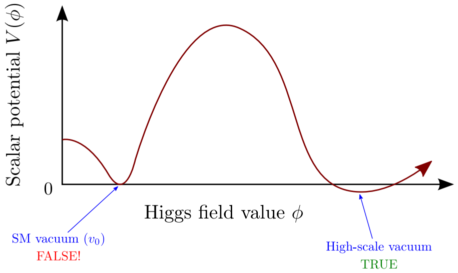
White = 1σ, 2σ with XENON1T 2018

Grey = 1σ, 2σ with XENON1T 2017

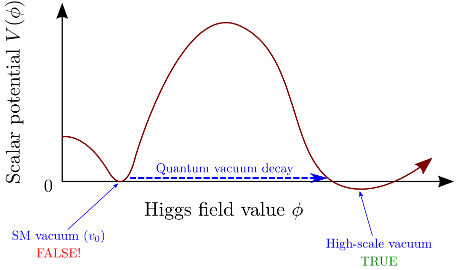
Aside: Vacuum Stability



- The electroweak vacuum of the Standard Model is not stable

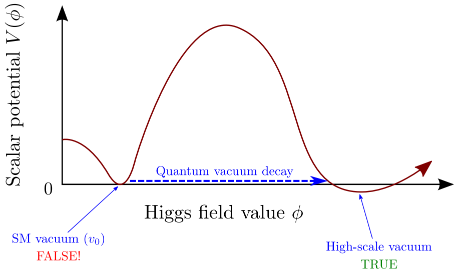


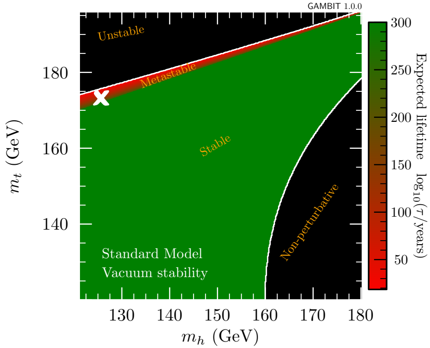
- The electroweak vacuum of the Standard Model is not stable



- The electroweak vacuum of the Standard Model is not stable

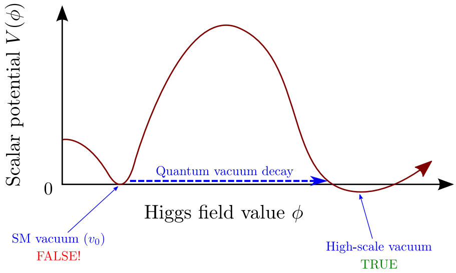
Aside: Vacuum Stability

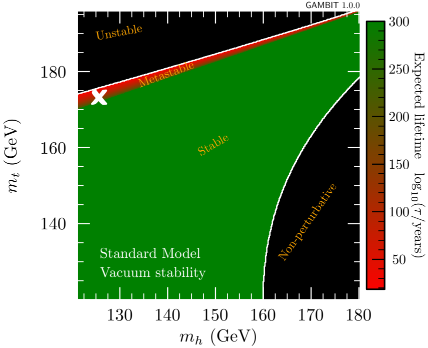




- The electroweak vacuum of the Standard Model is not stable
- Lifetime for decay to the global minimum is $\gg$ age of the Universe
  $\implies$ metastable

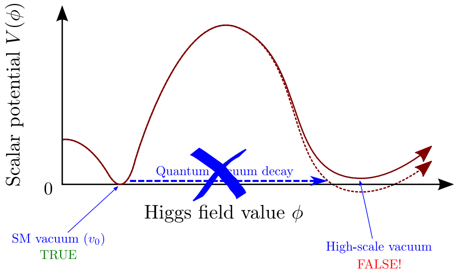
Aside: Vacuum Stability

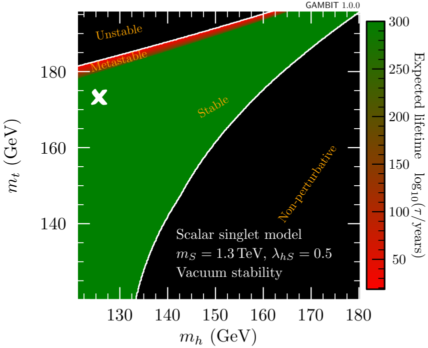




- Can be spoilt by Planck-scale effects
- Unclear how inflation would have put us in a metastable state
  $\rightarrow$ metastability makes Standard Model seem rather fine-tuned

Aside: Vacuum Stability



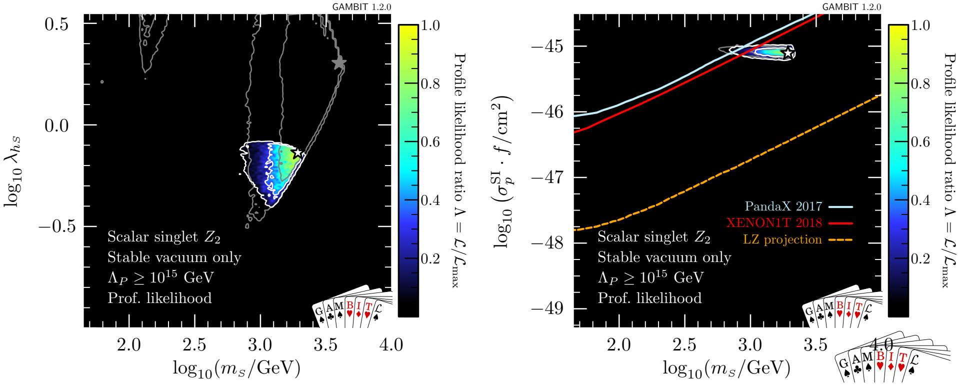


- Exact depth of minimum is very sensitive to running of couplings due to renormalisation
  $\rightarrow$ new particles can make our vacuum absolutely stable
  & remove the fine-tuning issue

Can the $\mathbb{Z}_2$ scalar singlet provide vacuum stability?

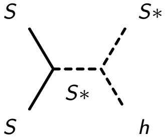
$\mathbb{Z}_2$ scalar singlet can stabilise the electroweak vacuum

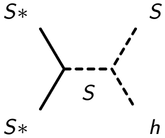
- Preferred mass of $\sim$2 TeV, $\sigma_{\rm SI} \sim 10^{-45}$ cm^2 to do so
- explains all of DM
- matches slight preference for signal in XENON1T data
- good fit to all observables ($p \sim 0.5$) $\Longrightarrow$ interesting. . . (?)

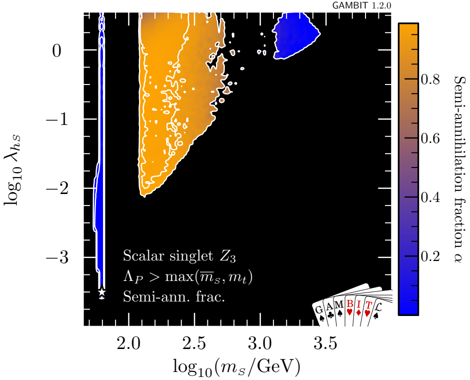


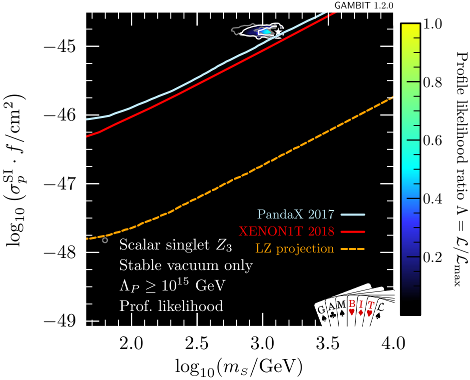
But why only $\mathbb{Z}_2$? $\mathbb{Z}_3$ works too, right? (EPJC, arXiv:1806.11281)

- All we were trying to achieve with the $\mathbb{Z}_2$ symmetry was to prevent $S \to SM + SM$
- Can be achieved with any other discrete symmetry, e.g. $\mathbb{Z}_3$
- $\mathcal{L}_S = -\mu_s S^\dagger S - \lambda_{hs} S^\dagger S H^\dagger H - \frac{\mu_3}{2}(S^3 + S^{\dagger 3}) + \ldots$
- Singlet (S) and anti-singlet (S^*) dark matter
- Semi-annihilation: $SS \to S^* h$, $S^* S^* \to Sh$









Grey = without demanding perturbativity

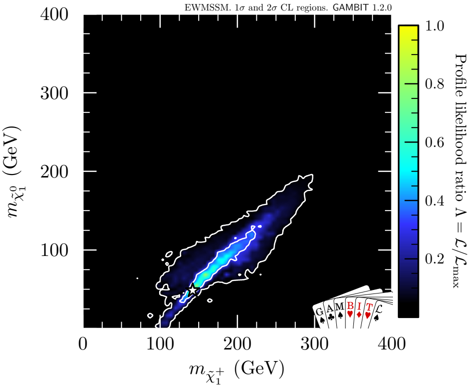
Excluded at >99% CL ($p < 0.01$) as all of dark matter
Excluded at >98% CL ($p < 0.02$) as *any* of dark matter

Also full global fits of **vector, Dirac & Majorana fermion Higgs portal** models

See backup slides or arXiv:1808.10465 (EPJC)

Outline

1. Global fits
   - Why?
   - GAMBIT
2. Status updates for key theories
   - Higgs-portal dark matter
   - Supersymmetry
   - Axions and ALPs

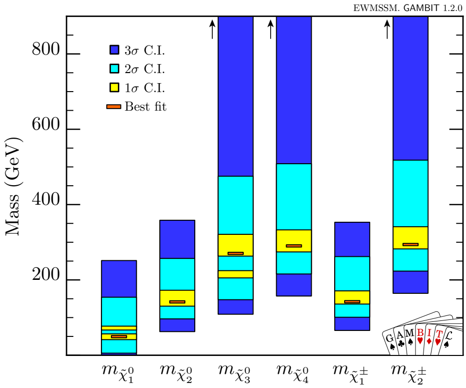


- Low-mass neutralinos & charginos
- everything else decoupled
- $M_1, M_2, \mu, \tan\beta$ free
- m_h fixed to 125.09 GeV

→ 3.5σ (local) combined signal significance

Electroweak analyses included in likelihood:

- ATLAS multi-lepton: $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$, $\tilde{\chi}_2^\pm\tilde{\chi}_1^\pm$, $\tilde{l}\tilde{l}$; final states with 2–3 leptons + 0–5 jets
- ATLAS 2/3-lepton recursive jigsaw searches for $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$
- ATLAS 4-lepton SUSY search
- ATLAS 4-b Higgsino search
- CMS 1lep(H)bb: single-lepton final states including $H \to bb$
- CMS 2SFOSlep-soft: $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$, virtual W^* and $Z^* \to ll$; final state with two same-flav. opp. sign leptons
- CMS 2SFOSlep: as above but with hard leptons (W, Z not virtual)
- CMS multi-lepton: similar to ATLAS, but exclusively $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$ production
- Assorted LEP likelihoods & h/Z invisible widths



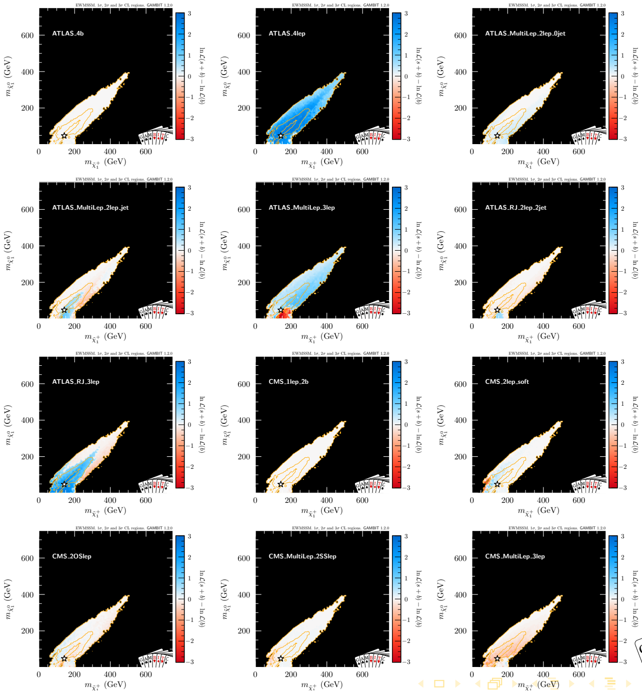
- Low-mass neutralinos & charginos
- everything else decoupled
- $M_1, M_2, \mu, \tan\beta$ free
- m_h fixed to 125.09 GeV

→ 3.5σ (local) combined signal significance

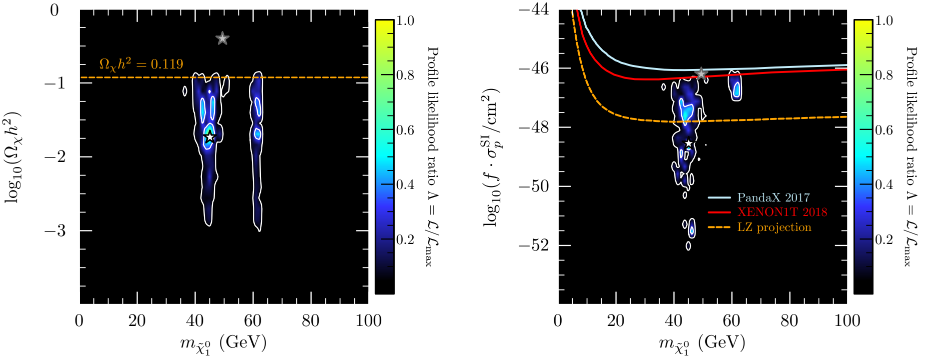
Electroweak analyses included in likelihood:

- ATLAS multi-lepton: $\tilde\chi_2^0\tilde\chi_1^\pm$, $\tilde\chi_2^\pm\tilde\chi_1^\pm$, $\tilde{l}l$; final states with 2–3 leptons + 0–5 jets
- ATLAS 2/3-lepton recursive jigsaw searches for $\tilde\chi_2^0\tilde\chi_1^\pm$
- ATLAS 4-lepton SUSY search
- ATLAS 4-b Higgsino search
- CMS 1lep(H)bb: single-lepton final states including $H \to bb$
- CMS 2SFOSlep-soft: $\tilde\chi_2^0\tilde\chi_1^\pm$, virtual W^* and $Z^* \to ll$; final state with two same-flav. opp. sign leptons
- CMS 2SFOSlep: as above but with hard leptons (W, Z not virtual)
- CMS multi-lepton: similar to ATLAS, but exclusively $\tilde\chi_2^0\tilde\chi_1^\pm$ production
- Assorted LEP likelihoods & h/Z invisible widths

Likelihood contributions of individual analyses



Just taking the points within our 3σ regions from the LHC fit:



Z and h funnel mechanisms can give sensible relic densities
$\rightarrow$ models consistent with LHC excesses can also naturally explain dark matter

Also global fits of full **GUT-scale and weak-scale MSSM models**

See:

- backup slides
- arXiv:1705.07917 (EPJC)
- arXiv:1705.07935 (EPJC)

Outline

1. Global fits
   - Why?
   - GAMBIT
2. Status updates for key theories
   - Higgs-portal dark matter
   - Supersymmetry
   - Axions and ALPs

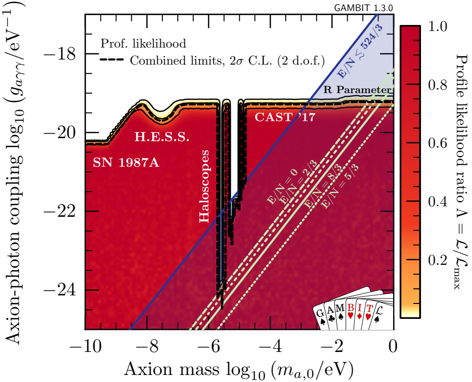
Axion-like Particles (ALPs) (arXiv:1810.07192)

Parameters:

- couplings $g_{a\gamma\gamma} + g_{aee}$
- decay constant f_a
- initial misalignment angle θ_i
- zero-temperature mass $m_{a,0}$
- 2× mass-related nuisance parameters

Likelihoods:

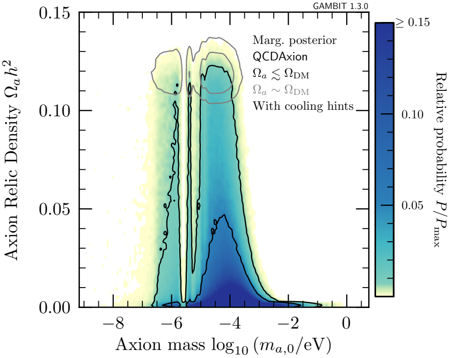
- Light shining through wall: ALPS
- Helioscopes: CAST(2007), CAST(2017)
- Haloscopes: RBF, UF, ADMX(1998-2009), ADMX(2018)
- DM relic density: *Planck*
- Astrophysics: HESS, SN1987a, HB/RGB stars (R parameter)

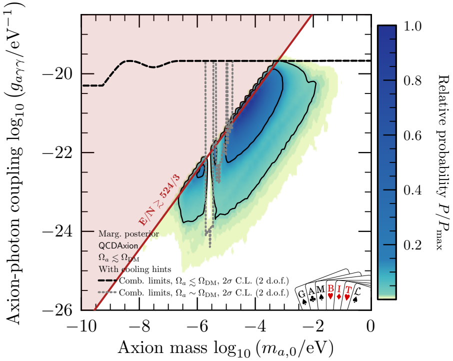


QCD axions

Bayesian analysis gives preferred axion mass range and couplings:

- small $m_a \Rightarrow$ fine-tuning in θ_i to avoid DM overproduction
- large $m_a \Rightarrow$ fine-tuning in E/N (i.e. $g_{a\gamma\gamma}$) to avoid experiments





(assuming log priors on f_a, C_{aee}; flat priors prefer lower masses)

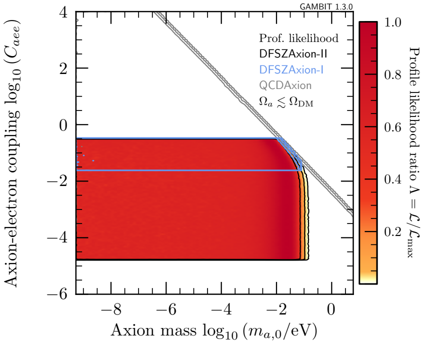
Summary

- Higgs portal models are getting pretty well constrained nowadays
- Vacuum stability + XENON1T restrict scalar variant to a small region at high mass (consistent with tiny excess)
- Axions and ALPs are getting a lot more interesting, but...
- 3.5σ hint of light SUSY in LHC electroweak searches?

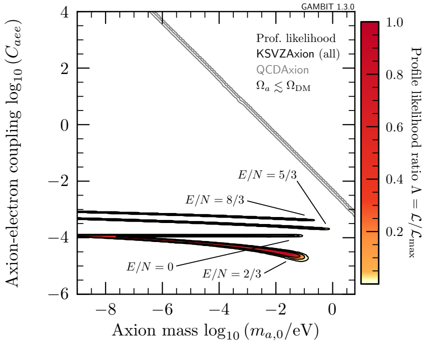
- Global analyses complete for many models
- GAMBIT results, samples, run files, best fits, benchmarks, etc are *all* available to download from Zenodo: www.zenodo.org/communities/gambit-official/
- GAMBIT code is public: gambit.hepforge.org

Backup slides

White dwarfs are cooling faster than expected
$\rightarrow$ due to emission of axions?



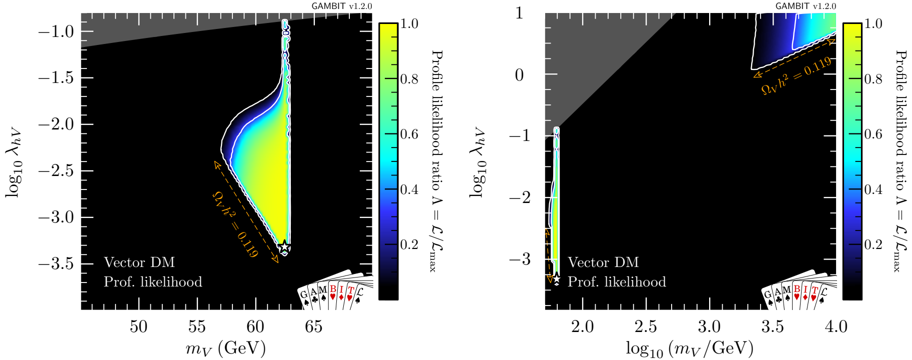
$E/N = 2/3, 8/3$



$E/N = 8/3, 0, 5/3, 2/3$

KSVZ, DFSZ: specific QCD axion models, with different anomaly ratios E/N

ALPs, QCD axion & DFSZ all give good fits; KSVZ excluded at >99% CL

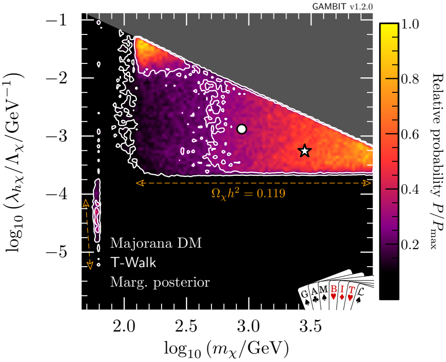


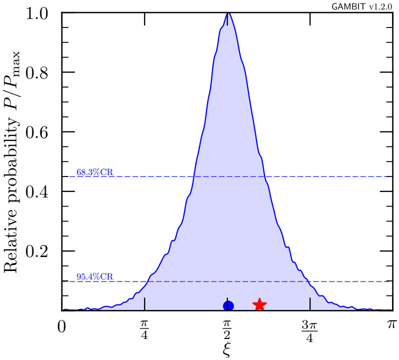
Unitarity bound $\lambda_{hs} \leq \left(\frac{2m_V}{v_0}\right)^2$ cuts out intermediate masses and 'upper neck' region

Only resonance and high-mass solutions remain

Fermionic Higgs-Portal DM

(EPJC, arXiv:1808.10465)





Model has mixed CP-even and CP-odd portal coupling

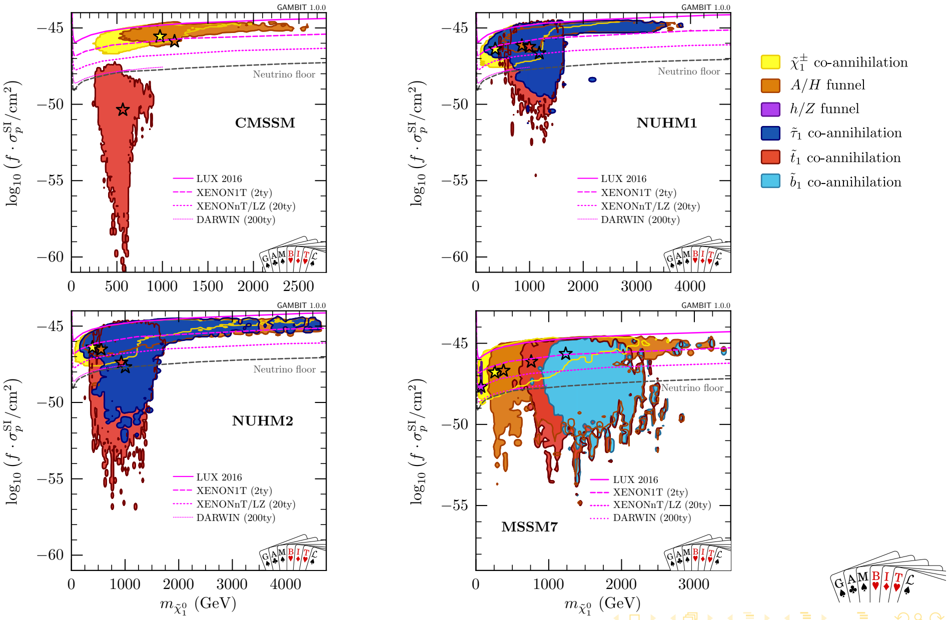
$$\mathcal{L}_\chi = \lambda_{h\chi}/\Lambda_\chi(\cos\xi\,\overline{\chi}\chi + \sin\xi\,\overline{\chi}i\gamma_5\chi)H^\dagger H + \ldots$$

→Momentum-independent (from CP-even) and q^2-dependent (from CP-odd) nuclear-scattering cross-sections

→Evading direct detection requires reduced CP-even coupling ($\xi \to \frac{\pi}{2}$)

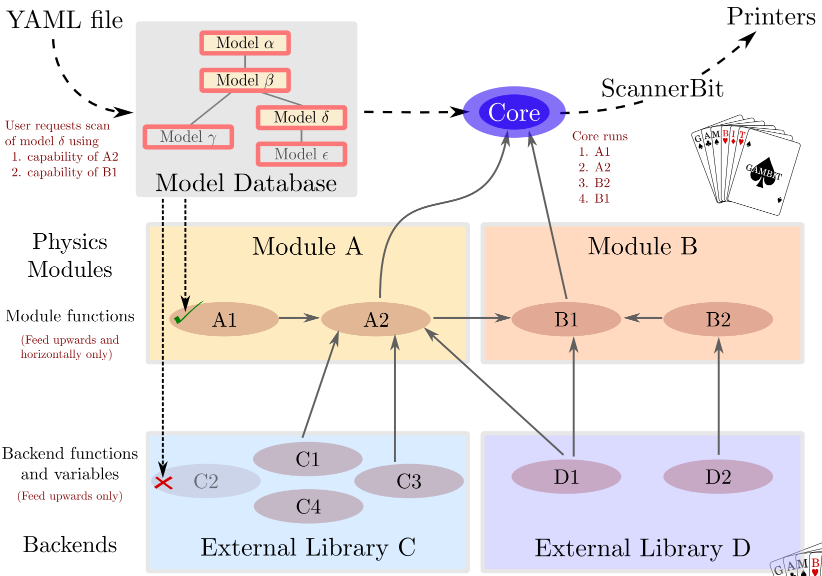
→Bayesian model selection favours CP-violating version roughly 100:1

MSSM: 4/5/6/7 par. + 5 nuisances (EPJC, arXiv:1705.07917/35)



Functional overview of GAMBIT

(EPJC, arXiv:1705.07908)



- 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 (EPJC, arXiv:1705.07908)

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 (EPJC, arXiv:1705.07908)

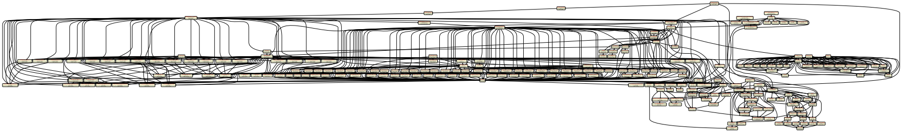
pat@xpspedition: ~/gambit 163x45

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

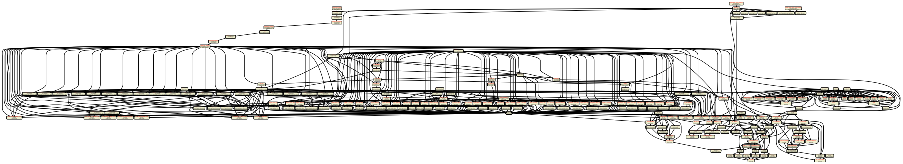
BACKENDS	VERSION	PATH TO LIB	STATUS	#FUNC	#TYPES	#CTORS
DDCalc0	0.0	Backends/installed/DDCalc/0.0/libDDCalc0.so	OK	62	0	0
DarkSUSY	5.1.1	Backends/installed/DarkSUSY/5.1.1/lib/libdarksusy.so	OK	68	0	0
FastSim	1.0	Backends/installed/fastsim/1.0/libfastsim.so	absent/broken	1	0	0
FeynHiggs	2.11	Backends/installed/FeynHiggs/2.11.2/lib/libFH.so	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)

CMSSM:

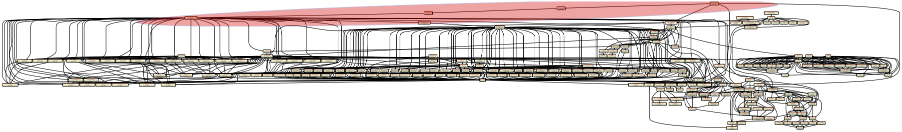


MSSM7:

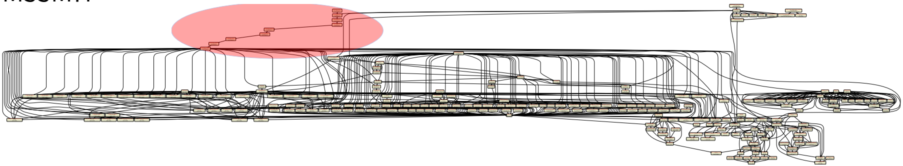


Dependency Resolution

CMSSM:



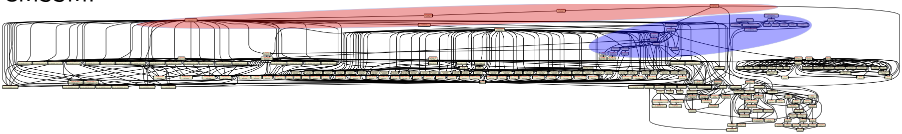
MSSM7:



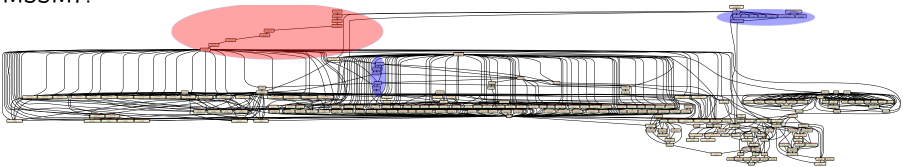
Red: Model parameter translations

Dependency Resolution

CMSSM:

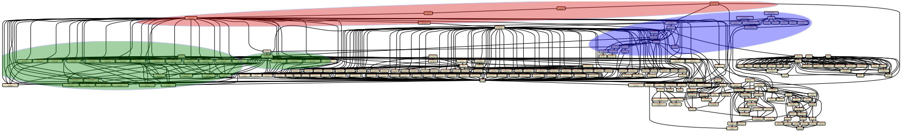


MSSM7:

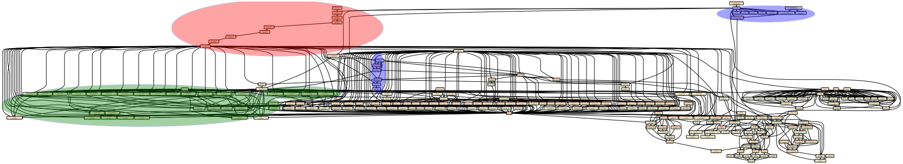


Red: Model parameter translations
Blue: Precision calculations

CMSSM:



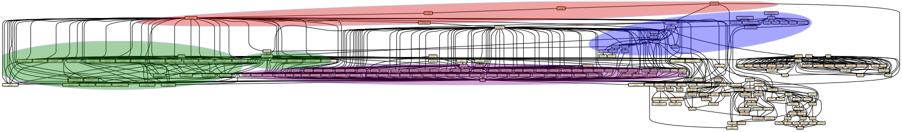
MSSM7:



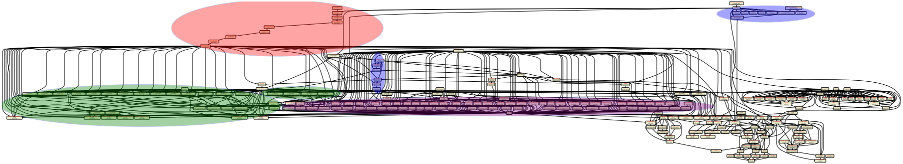
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods

Dependency Resolution

CMSSM:



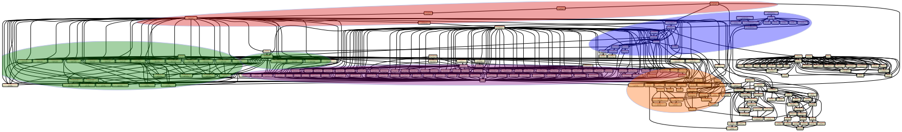
MSSM7:



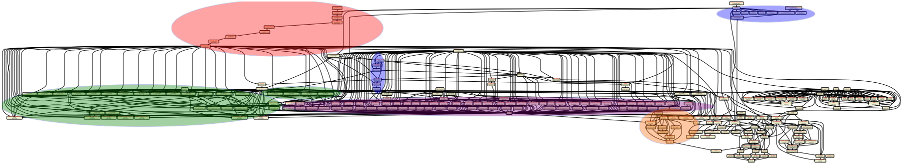
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays

Dependency Resolution

CMSSM:



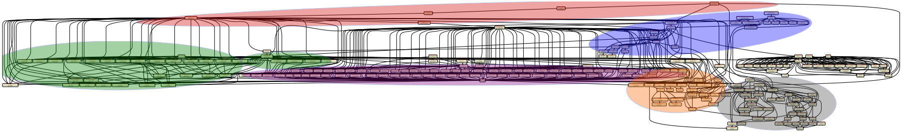
MSSM7:



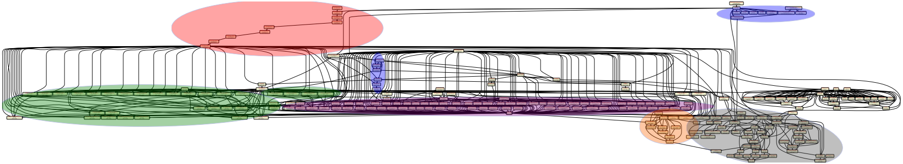
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods

Dependency Resolution

CMSSM:



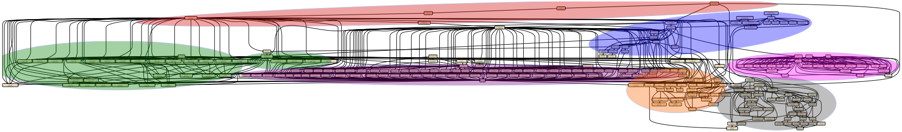
MSSM7:



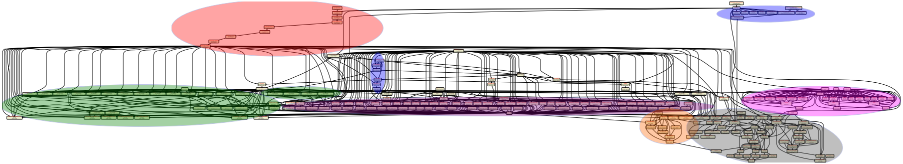
Red: Model parameter translations
Blue: Precision calculations
Green: LEP rates+likelihoods
Purple: Decays
Orange: LHC observables and likelihoods
Grey: DM direct, indirect and relic density

Dependency Resolution

CMSSM:

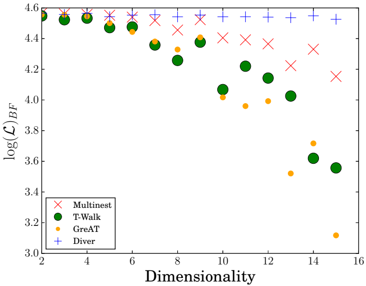


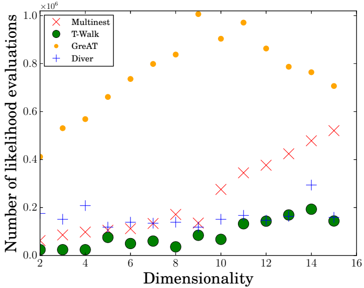
MSSM7:



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

Extensive scanner tests on scalar singlet model with different numbers of nuisance paramters





Diver scales far better with dimensionality than MultiNest or other scanners

Expansion: adding new observables and likelihoods

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

```
#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:

```
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.

Interface: yaml file

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)

```yaml
Parameters:
  StandardModel_SLHA2: !import StandardModel_SLHA2_default
  MSSM25atQ: !import LesHouches.in.MSSM_1.yaml

Priors:
    # none: all parameters fixed in this example.

Scanner:

  use_scanner: toy_mcmc

  scanners:
    toy_mcmc:
      plugin: toy_mcmc
      point_number: 2000
      output_file:  output
      like:  Likelihood

ObsLikes:

  # Test DecayBit
  - purpose:       Test
    capability:    decay_rates
    type:          DecayTable

  # 79-string IceCube likelihood
  - capability: IceCube_likelihood
    purpose: Likelihood
    function: IC79_loglike

Rules:

  - capability: MSSM_spectrum
    function: get_MSSMatQ_spectrum
    options:
      invalid_point_fatal: true
```

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)

```yaml
Parameters:
  StandardModel_SLHA2: !import StandardModel_SLHA2_default
  MSSM25atQ: !import LesHouches.in.MSSM_1.yaml

Priors:
    # none: all parameters fixed in this example.

Scanner:

  use_scanner: toy_mcmc

  scanners:
    toy_mcmc:
      plugin: toy_mcmc
      point_number: 2000
      output_file:  output
      like:  Likelihood

ObsLikes:

  # Test DecayBit
  - purpose:      Test
    capability:   decay_rates
    type:         DecayTable

  # 79-string IceCube likelihood
  - capability: IceCube_likelihood
    purpose: Likelihood
    function: IC79_loglike

Rules:

  - capability: MSSM_spectrum
    function: get_MSSMatQ_spectrum
    options:
      invalid_point_fatal: true
```

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)

```yaml
Parameters:
  StandardModel_SLHA2: !import StandardModel_SLHA2_default
  MSSM25atQ: !import LesHouches.in.MSSM_1.yaml

Priors:
    # none: all parameters fixed in this example.

Scanner:

  use_scanner: toy_mcmc

  scanners:
    toy_mcmc:
      plugin: toy_mcmc
      point_number: 2000
      output_file:  output
      like:  Likelihood

ObsLikes:

  # Test DecayBit
  - purpose:      Test
    capability:   decay_rates
    type:         DecayTable

  # 79-string IceCube likelihood
  - capability: IceCube_likelihood
    purpose: Likelihood
    function: IC79_loglike

Rules:

  - capability: MSSM_spectrum
    function: get_MSSMatQ_spectrum
    options:
      invalid_point_fatal: true
```

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)

```yaml
Parameters:
  StandardModel_SLHA2: !import StandardModel_SLHA2_default
  MSSM25atQ: !import LesHouches.in.MSSM_1.yaml

Priors:
    # none: all parameters fixed in this example.

Scanner:

  use_scanner: toy_mcmc

  scanners:
    toy_mcmc:
      plugin: toy_mcmc
      point_number: 2000
      output_file:  output
      like:  Likelihood

ObsLikes:

  # Test DecayBit
  - purpose:      Test
    capability:   decay_rates
    type:         DecayTable

  # 79-string IceCube likelihood
  - capability: IceCube_likelihood
    purpose: Likelihood
    function: IC79_loglike

Rules:

  - capability: MSSM_spectrum
    function: get_MSSMatQ_spectrum
    options:
      invalid_point_fatal: true
```

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)

```yaml
Parameters:
  StandardModel_SLHA2: !import StandardModel_SLHA2_default
  MSSM25atQ: !import LesHouches.in.MSSM_1.yaml

Priors:
    # none: all parameters fixed in this example.

Scanner:

  use_scanner: toy_mcmc

  scanners:
    toy_mcmc:
      plugin: toy_mcmc
      point_number: 2000
      output_file:  output
      like:  Likelihood

ObsLikes:

  # Test DecayBit
  - purpose:      Test
    capability:   decay_rates
    type:         DecayTable

  # 79-string IceCube likelihood
  - capability: IceCube_likelihood
    purpose: Likelihood
    function: IC79_loglike

Rules:

  - capability: MSSM_spectrum
    function: get_MSSMatQ_spectrum
    options:
      invalid_point_fatal: true
```

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 backends[1]
- **flexible output streams** (ASCII, databases, HDF5, . . .)
- available as docker plugin or vagrant virtual machine
- more more more. . .

[1] if 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

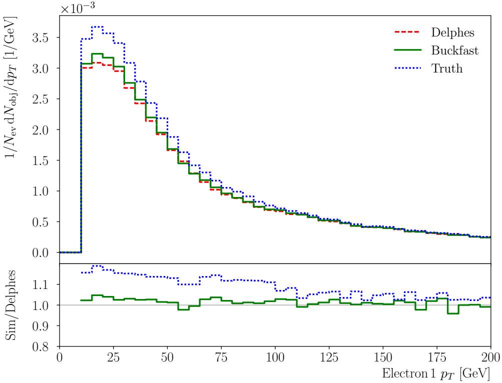
- no simplified models, just faster direct simulation

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:

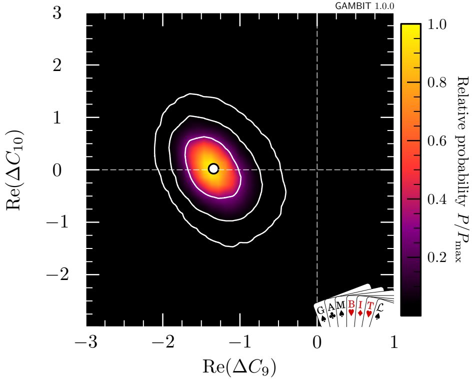
- MC generation: Pythia8 parallelised with OpenMP + other speed tweaks
- Detector simulation: fast simulation based on 4-vector smearing
  $\rightarrow$ matches DELPHES results very closely (but much faster!)
- Cross-sections: LO + LL from MC generator by default (fast NLO in works for SUSY)
- Analysis framework: custom event-level, independent of experiment or simulation
- Likelihood: inline systematic error marginalisation (via `nulike`)
- v1.0 shipped with:
  - ATLAS 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 + CMS Run II 0ℓ
- v1.2 now released with a bucketload of additional Run II analyses; 80 fb^{-1} analyses coming soon too

Flavour physics global fits

LHCb sees possible hints of lepton flavour non-universality in neutral currents
$\rightarrow$ GAMBIT flavour EFT global fit (Wilson coefficients as model parameters)

Flavour likelihoods in GAMBIT:

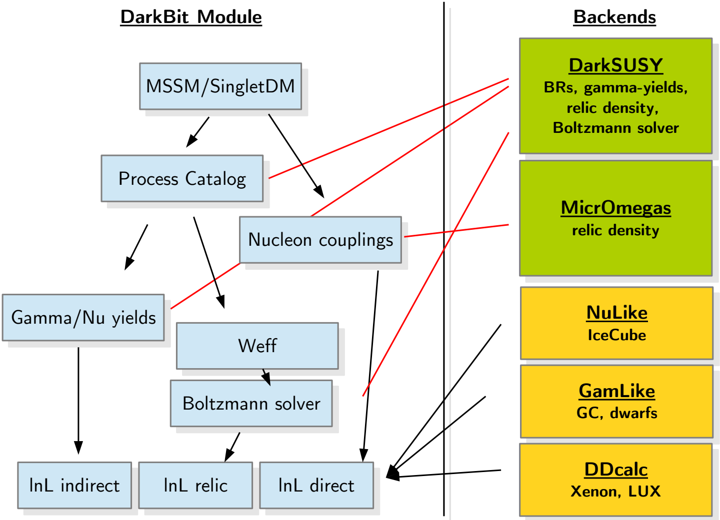
- $(g-2)_\mu$
- $B \rightarrow X_s\gamma$
- $B \rightarrow \mu\mu$
- $B_s \rightarrow \mu\mu$
- $B \rightarrow K^*\mu\mu$ + angular observables
- $B \rightarrow \tau\nu$
- $B \rightarrow D\mu\nu$
- $B \rightarrow D\tau\nu$
- $B \rightarrow D^*\mu\nu$
- $B \rightarrow D^*\tau\nu$
- $D \rightarrow \mu\nu$
- $D_s \rightarrow \mu\nu$
- $D_s \rightarrow \tau\nu$
- $\frac{\mathcal{B}(K \rightarrow \mu\nu)}{\mathcal{B}(\pi \rightarrow \mu\nu)}$



(EPJC, arXiv:1705.07933)

Fit to $\mathcal{O}_7$ (photons), $\mathcal{O}_9$ (leptons, vector), $\mathcal{O}_{10}$ (leptons, axial-vector)

DarkBit 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).