CTA DARK MATTER ANALYSIS WITH GAMMAPY

An extension of the Dark Matter module

Judit Pérez-Romero, judit.perez@uam.es, IFT UAM-CSIC
Jose Enrique Ruiz, jer@iaa.es, IAA-CSIC
Miguel Á. Sánchez-Conde, miguel.sanchezconde@uam.es, IFT UAM-CSIC
Gonzalo Rodríguez, grferna@roma2.infn.it, Sezione INFN di Tor Vergata

OUTLINE

- Context of the research project
- CTA Dark Matter community
- Brief introduction to Dark Matter science
- Proposal for Gammapy: Dark Matter analysis pipeline and dm-module extension
- An example notebook and workflow
- Summary

CONTEXT OF THE RESEARCH PROJECT

- Project started in July 2018:

Perform an state-of-the-art study of the capabilities CTA has for Dark Matter searches, mainly using Galaxy Clusters as targets

↓

Need of a software to perform the analysis: Gammapy

- Existence of a basic dark matter utilities notebook:

https://docs.gammapy.org/dev/notebooks/astro_dark_matter.html

- Embedded in the CTA Dark Matter SWG:

General need of a software to perform the standard Dark Matter analysis

CTA DARK MATTER COMMUNITY

- CTA Collaboration developed an ambitious Dark Matter Programme: CTAC17

  https://arxiv.org/abs/1709.07997

- Dark Matter SWG Coordinators:
  - Miguel Á. Sánchez-Conde (IFT UAM-CSIC, Spain)
  - Gabrijela Zaharijas (U. Nova Garica, Slovenia)

- General document for CTA Dark Matter searches:

  https://forge.in2p3.fr/projects/dmep-swg-standards-and-conventions-for-cta-dm-analyses/wiki

- More than 150 people working in different projects!

- Consortium papers within Dark Matter SWG:
  - Galactic Center (KSP) → Shared with Diffuse SWG
  - Dwarf Spheroidal Galaxies → Dark Matter Exclusively!
  - Galaxy Clusters (KSP) → Shared with Cosmic Rays SWG
  - Large Magellanic Cloud (KSP) → Shared with Diffuse SWG
  - Axion-Like Particles → Shared with Gamma-ray Propagation Task Force
- More non-consortium projects from CTA researchers
- CTA Dark Matter SWG webpage for more info:

https://portal.cta-observatory.org/WG/PHYS/SitePages/Dark Matter and Exotic Physics SWG.aspx

WIMP DARK MATTER SCIENCE

- Dark Matter (DM) flux:

$$\frac{d\Phi_{ann}}{dE}(E, l.o.s, \Delta\Omega, z) = \frac{d\phi}{dE}(E, z) \times J(l.o.s, \Delta\Omega, z)$$

Total DM flux coming from an astrophysical object

E = Energy
l.o.s = Line of sight
ΔΩ = Angular extension
z = redshift

Particle Physics Model

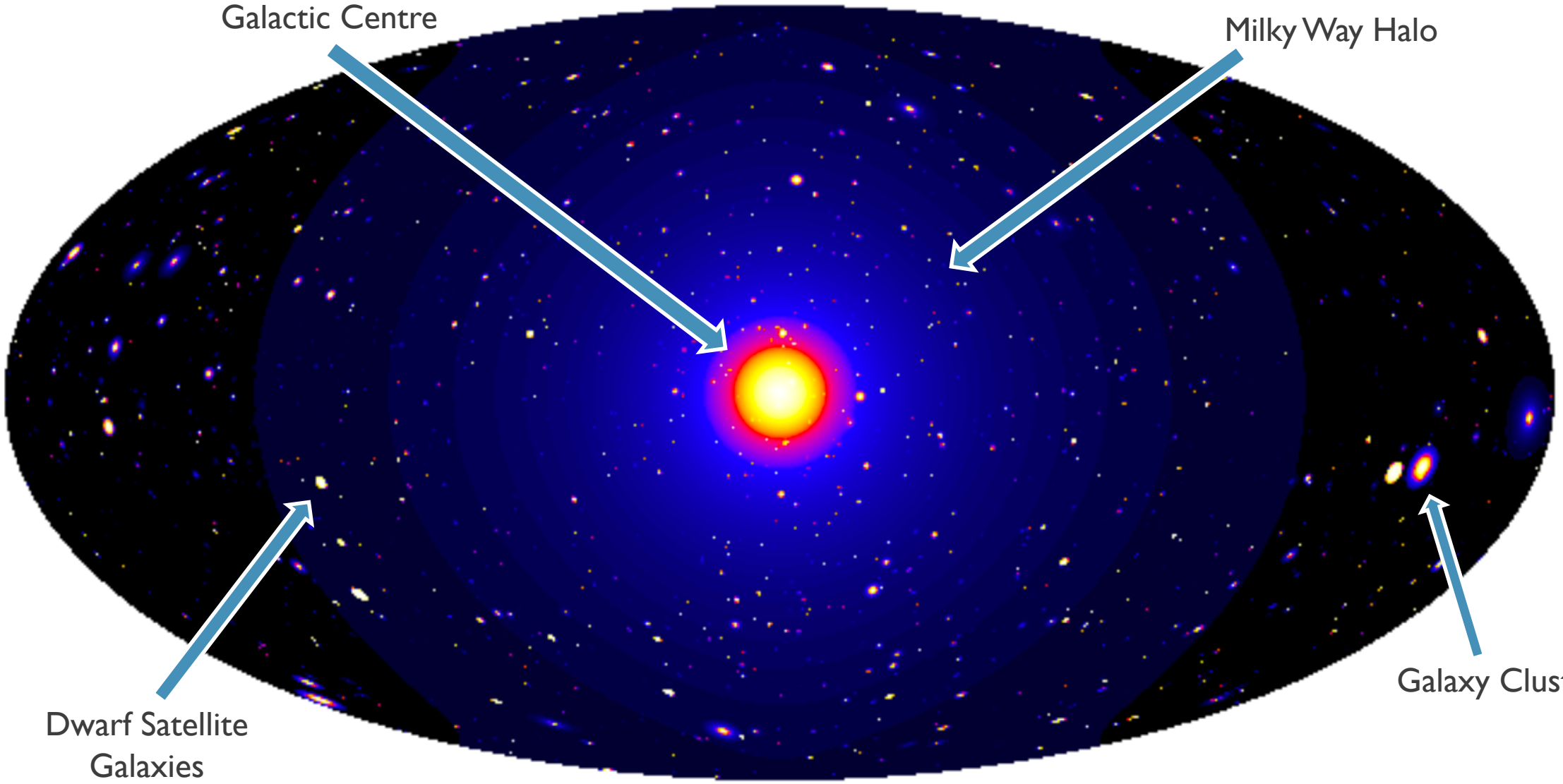
- Difficult to parametrize analytically
- All WIMP models are tabulated in Cirelli+12

Astrophysical Model

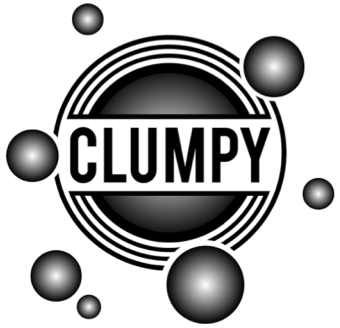
- Accounts for the DM distribution of the object
- State-of-the-art computations:

Strong simulation efforts are needed

Dark Matter induced gamma-ray sky simulation



Pieri+09, arXiv:0908.0195



https://clumpy.gitlab.io/CLUMPY/

- C/C++ code widely used by the DM community to compute J-factors for different structures

A lot of parameters accounting for the inner structure of the objects → State-of-the-art modelling of the astrophysical DM component

J-factor calculations in dm-module: Toy models

- Already suggested the possibility of cross checking functionality with Gammapy:

  https://docs.gammapy.org/dev/astro/darkmatter/index.html?highlight=CLUMPY

- Clumpy outputs:
  - Single value of the J-factor
  - 2D skymap of the J-factor

PROPOSAL FOR GAMMAPY

Create a standard analysis pipeline for DM searches

CTA Dark Matter SWG needs

- **Create tutorial notebooks** for this analysis pipeline

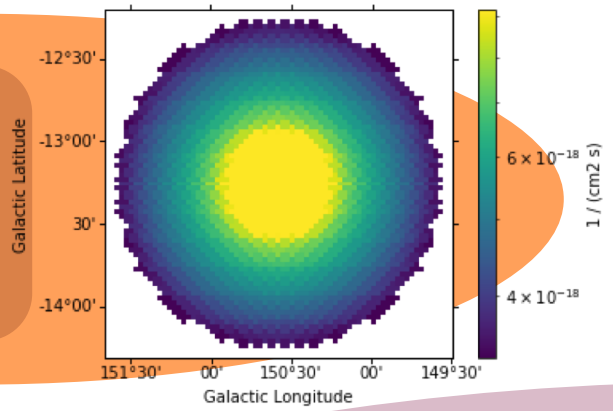
Incorporate functionalities ⟳ Identify notebook needs

- Make an **extension of the existing dm-module**
  - Add some new classes
  - Refactor code

Gammapy Implementations

Input J factor (Clumpy/own cooked file):

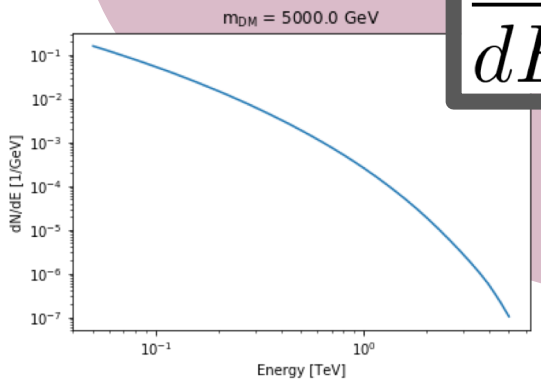
- One value
- 2D skymap
- 2D skymap + mask



J +

Particle Physics Model in Gammapy (dm-module)

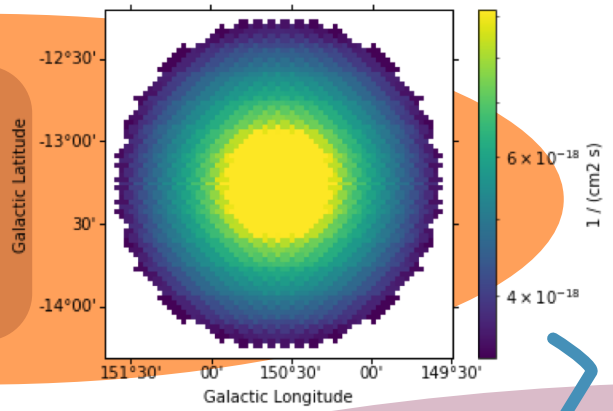
$\frac{d\phi}{dE}$



Colour legend:

- Already implemented in Gammapy (or minor improvements)
- Needs to be implemented in Gammapy (or major improvements)
- Needs to be read in Gammapy (created outside Gammapy)

Input J factor
(Clumpy/own cooked file):
- One value
- 2D skymap
- 2D skymap + mask

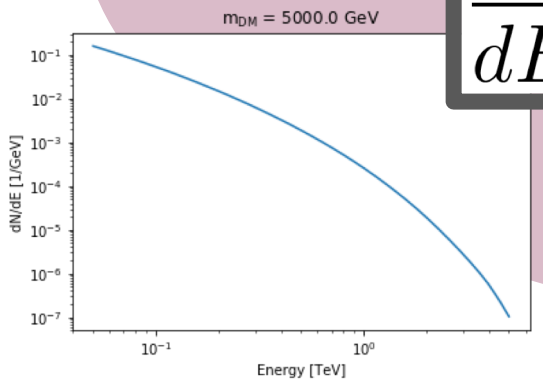


J

+

Particle Physics Model in Gammapy (dm-module)

$\frac{d\phi}{dE}$

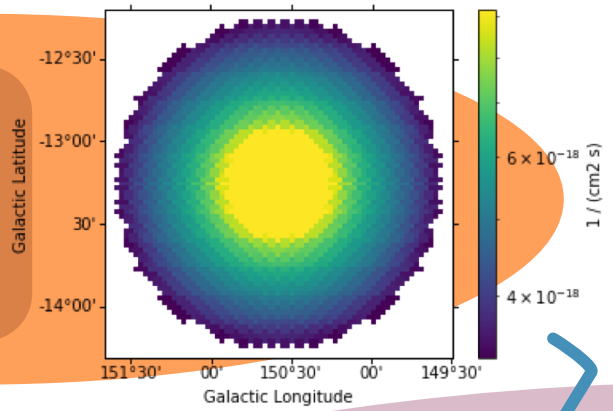


Total DM flux:
- Spatial information
- Spectral information

$\frac{d\Phi_{ann}}{dE}$

Colour legend:
- Already implemented in Gammapy (or minor improvements)
- Needs to be implemented in Gammapy (or major improvements)
- Needs to be read in Gammapy (created outside Gammapy)

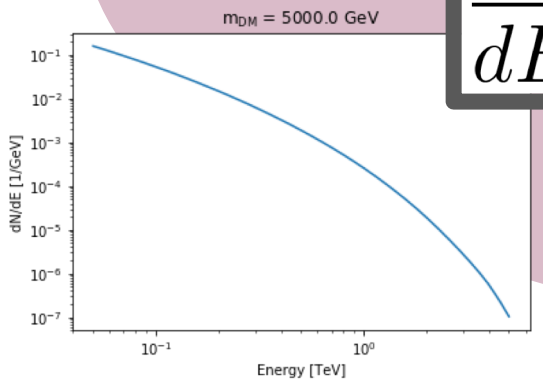
Input J factor (Clumpy/own cooked file):
- One value
- 2D skymap
- 2D skymap + mask



J +

Particle Physics Model in Gammapy (dm-module)

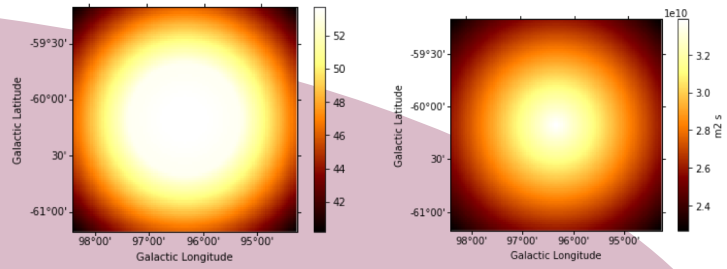
$\frac{d\phi}{dE}$



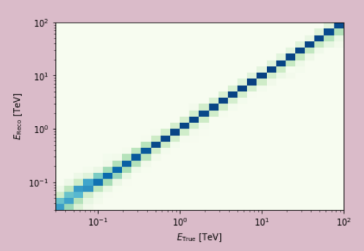
Total DM flux:
- Spatial information
- Spectral information

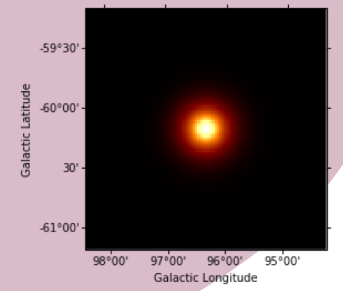
$\frac{d\Phi_{ann}}{dE}$

Colour legend:
- Already implemented in Gammapy (or minor improvements)
- Needs to be implemented in Gammapy (or major improvements)
- Needs to be read in Gammapy (created outside Gammapy)

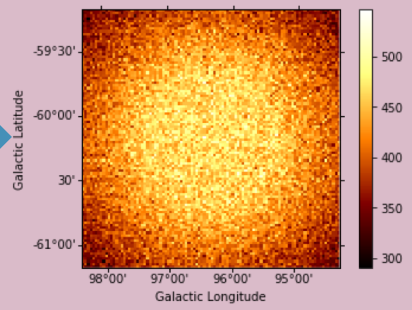


CTA IRFs





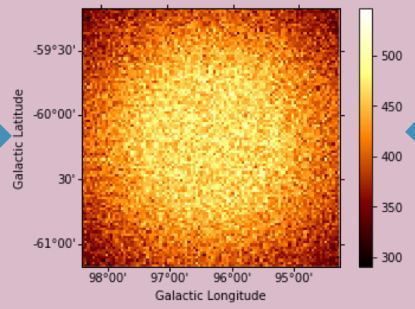
CTA IRFs



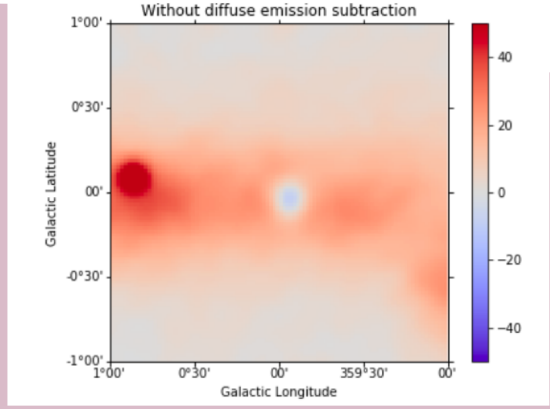
Colour legend:

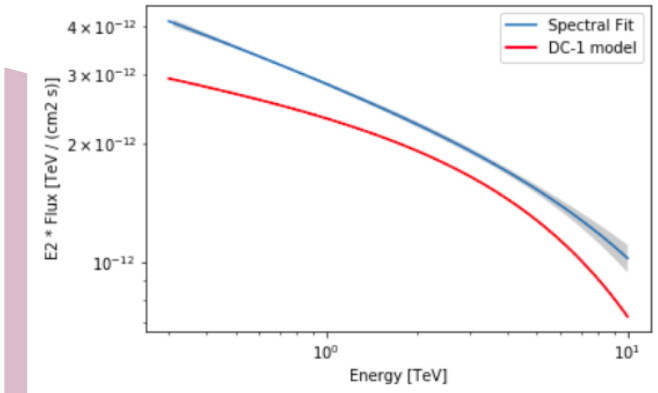
- Already implemented in Gammapy
- Needs to be implemented in Gammapy
- Needs to be read in Gammapy

CTA IRFs



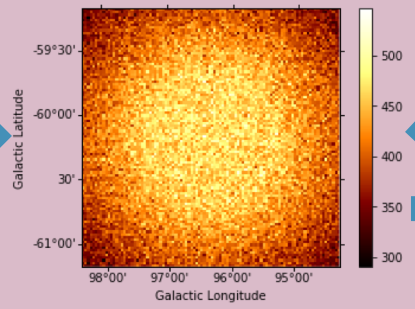
Signal!:
- Treat as an usual astrophysical source
- 3D analysis



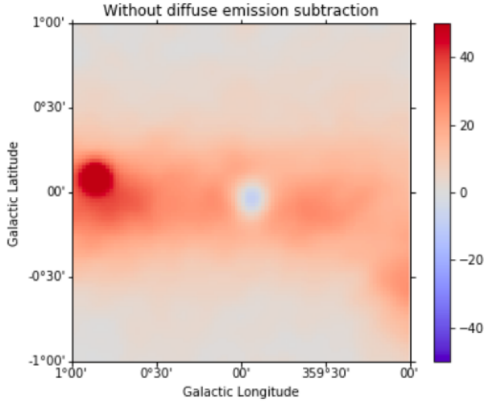


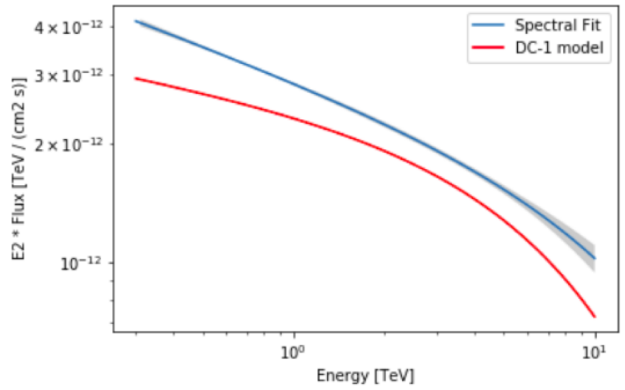
Colour legend:
- Already implemented in Gammapy
- Needs to be implemented in Gammapy
- Needs to be read in Gammapy

CTA IRFs



Signal!:
- Treat as an usual astrophysical source
- 3D analysis



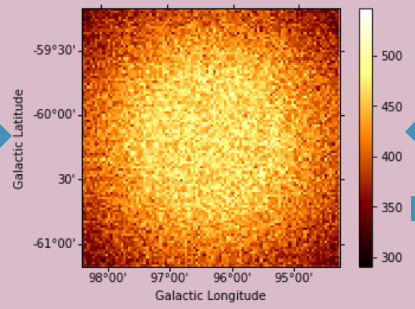


No Signal
(typical in DM searches)

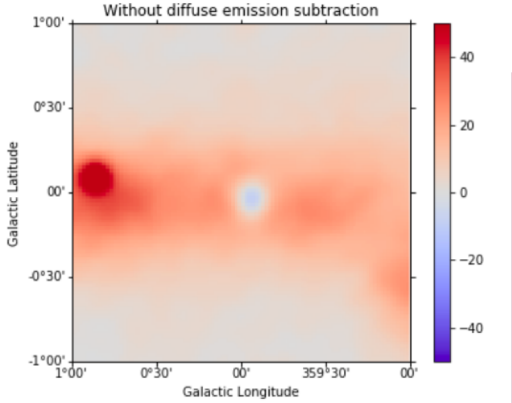
Flux upper limits obtained by fitting the simulation

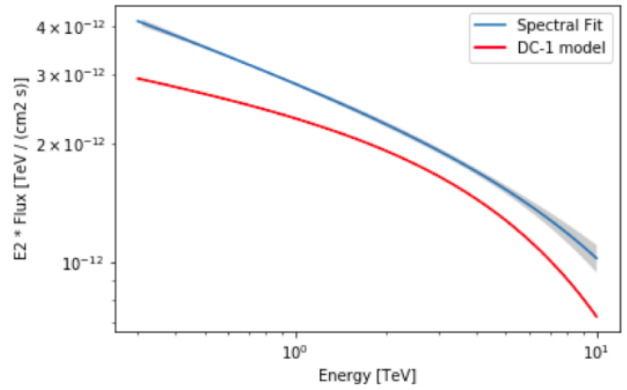
Colour legend:
- Already implemented in Gammapy
- Needs to be implemented in Gammapy
- Needs to be read in Gammapy

CTA IRFs



Signal!:
- Treat as an usual astrophysical source
- 3D analysis



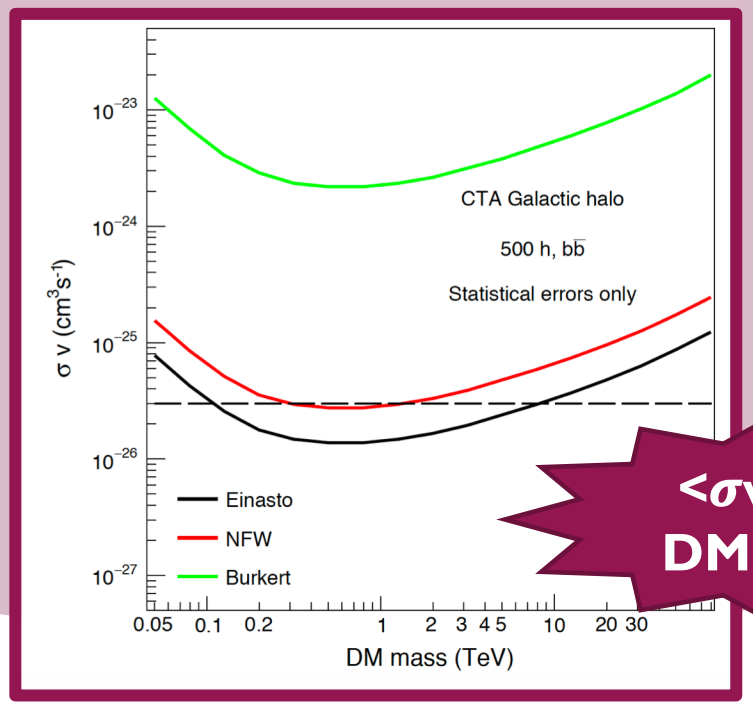


No Signal
(typical in DM searches)

Flux upper limits obtained by fitting the simulation

Constraints in the DM parameter space

CTAC17,
arXiv:1709.07997



<σv> vs. DM mass

Colour legend:
- Already implemented in Gammapy
- Needs to be implemented in Gammapy
- Needs to be read in Gammapy

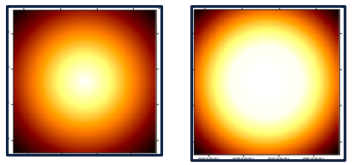
AN EXAMPLE NOTEBOOK

Load IRFs — i.e. South z=20° T=50h

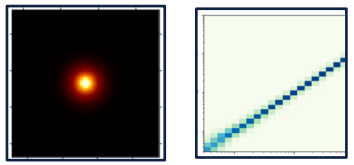
Derive ancillary IRF maps

- Define map geometry
  - source position
  - spatial range /bins
  - energy range /bins
- Observational values
  - pointing
  - lifetime
  - offset

exposure background



PSF Edisp



using ancillary IRF maps

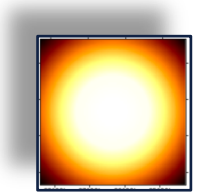
Build DM Emission Model

- Spatial
  - **SkyPointSource**
    - j-factor *.number*
  - **SkyDiffuseMap**
    - j-factor *.2D Map*
- Spectral

Parametrized User Model

- cross-section
  - *mass*
  - *channel*

Evaluate Fixed Reference 3D Model



- mass 5000 GeV
- annihilation channel b

https://github.com/peroju/dm-gammapy/blob/master/notebooks/DarkMatterUseCase.ipynb

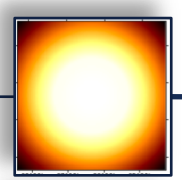
Fixed Reference Model

- mass 5000 GeV
- annihilation channel b

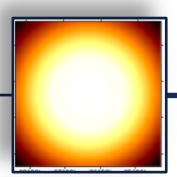
add random Poison noise

Build a set of models

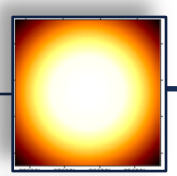
- mass range
- channel list



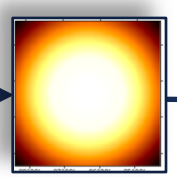
MapFit → *LRT* → *cross section value i*



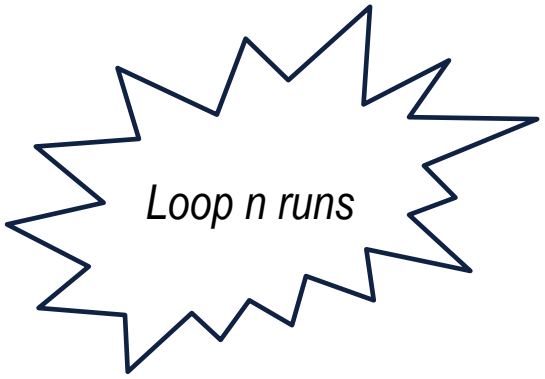
MapFit → *LRT* → *cross section value j*



MapFit → *LRT* → *cross section value k*



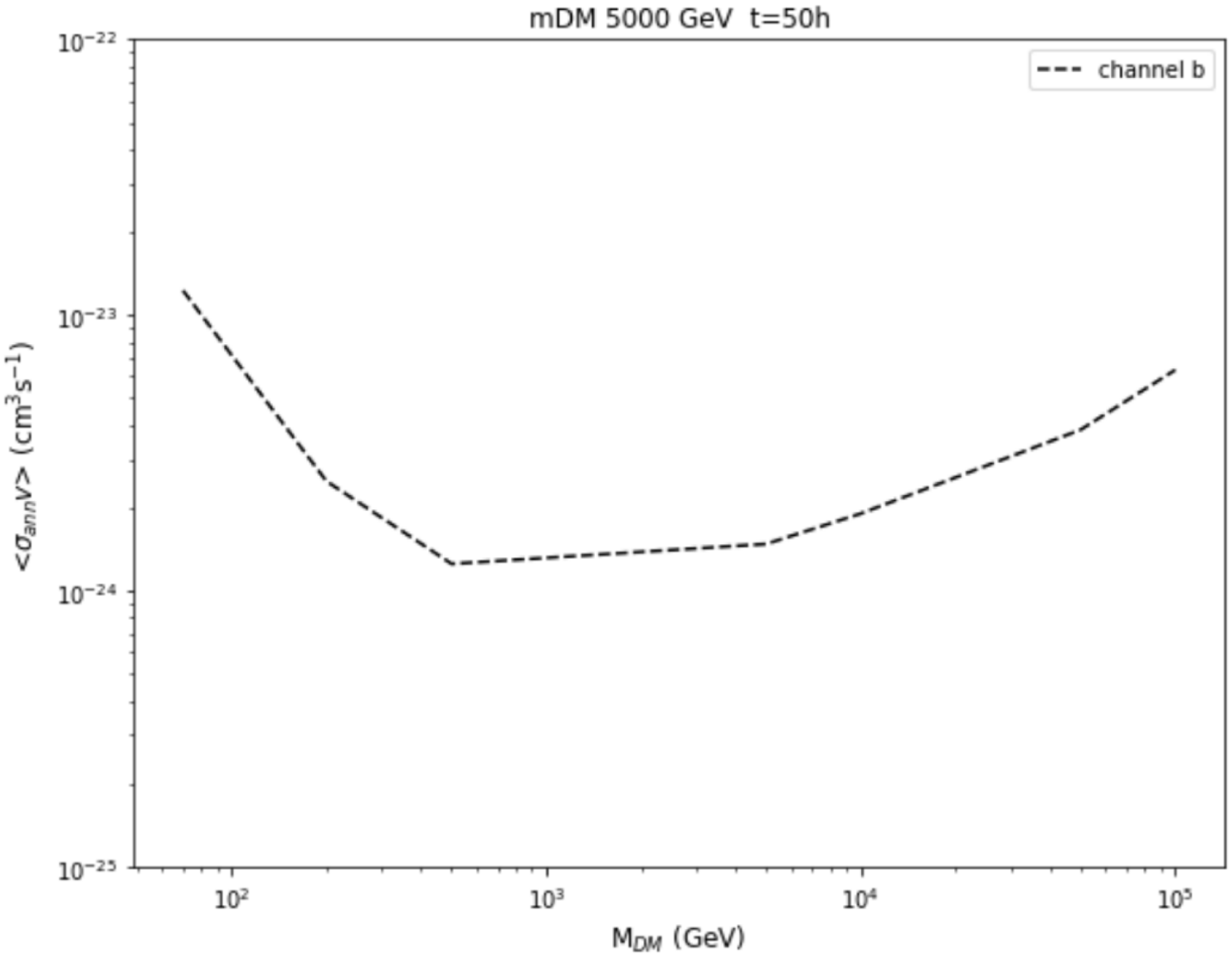
MapFit → *LRT* → *cross section value n*



take the mean values

derive cross section values on 95% CI of detection

9 runs of 7 Mapfit + LRT each ~ 4.5 min



for a reference model

- mass 5000 GeV
- annihilation channel b

SUMMARY

- Identified the need in the CTA DM community of the standard analysis software for DM searches
- Why Gammapy?
  - Open code
  - Easy to extend to the DM community needs
  - Computationally fast
- Reviewed the WIMP Dark Matter scenario, common to CTA DM community:
  - Need of tolerating J-factors computed outside Gammapy → Compatibility with Clumpy outputs
  - Gammapy already includes state-of-the-art models for the particle physics term
- Creation of standard DM analysis in the form of tutorial notebooks
  - Methodology: iterative progress between the notebook needs and the dm-module implementations
- Work during the Coding Sprint:
  - Need of general feedback: Are there already functions in Gammapy built or that can be adapted?
  - Specific need: Likelihoods in MapFit