HANDS-ON SESSION HERMES

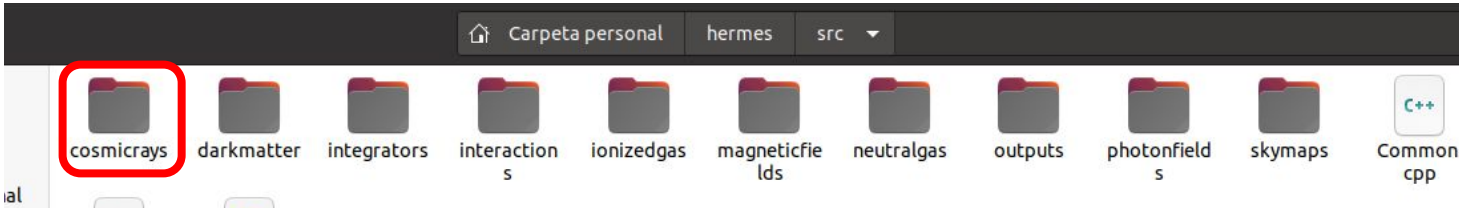
HERMES

Cosmic rays leave footprints when propagating

- Pion emission: gamma rays and neutrinos (among others)
- Inverse Compton Scattering
- Bremsstrahlung
- Synchrotron

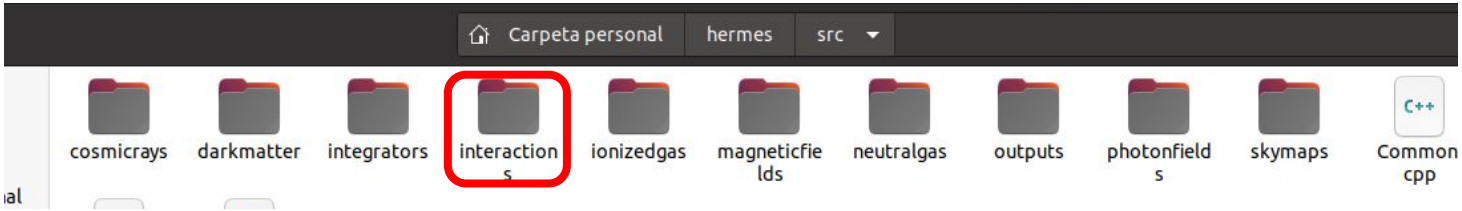
HERMES: modular code

- Cosmic-ray populations: results from DRAGON



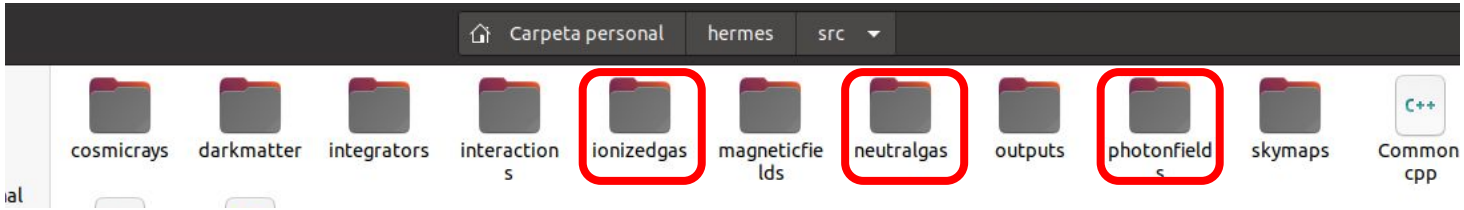
HERMES: modular code

- Cosmic-ray populations: results from DRAGON
- Interactions: those from the first slide



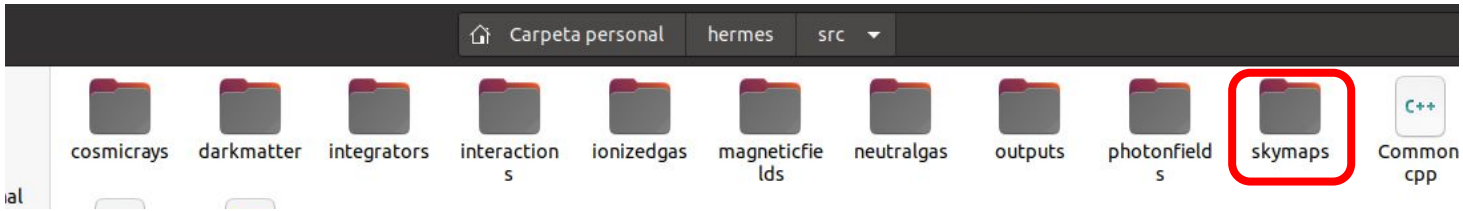
HERMES: modular code

- Cosmic-ray populations: results from DRAGON
- Interactions: those from the first slide
- Target field: gas or radiation fields



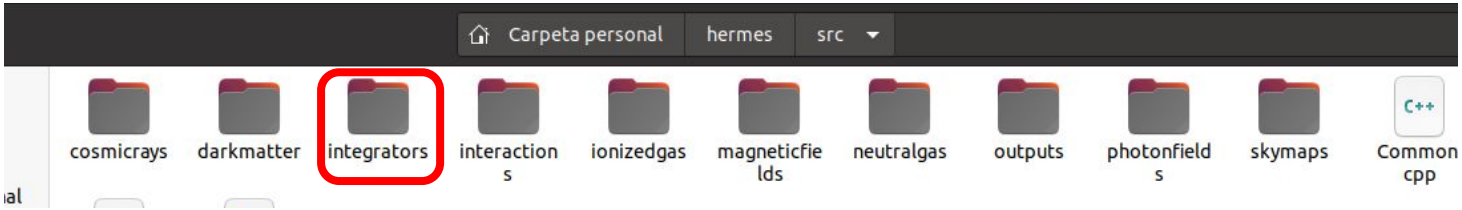
HERMES: modular code

- Cosmic-ray populations: results from DRAGON
- Interactions: those from the first slide
- Target field: gas or radiation fields
- Skymap: array where the emissions will be calculated



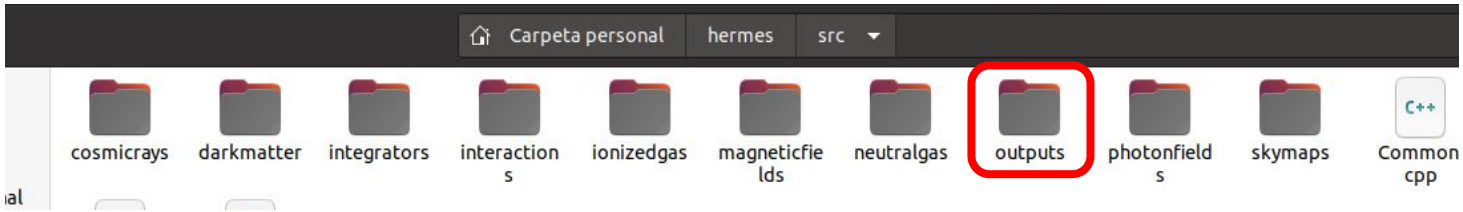
HERMES: modular code

- Cosmic-ray populations: results from DRAGON
- Interactions: those from the first slide
- Target field: gas or radiation fields
- Skymap: array where the emissions will be calculated
- Integrator: every interaction has its own integrator



HERMES: modular code

- Cosmic-ray populations: results from DRAGON
- Interactions: those from the first slide
- Target field: gas or radiation fields
- Skymap: array where the emissions will be calculated
- Integrator: every interaction has its own integrator
- Output: store the results in a fits file



Method: python

1. Load pyhermes module
2. Define the particle population, the interaction and the target fields

```
1 from pyhermes import *
2 from pyhermes.units import TeV, GeV, kpc, pc, deg, erg, sun_mass, m_proton,cm3
3
4 import numpy as np
5 import healpy
6 import matplotlib.pyplot as plt
7 import astropy.units as u
8
9
10 filename = '../../DRAGON2-Beta_version/output/BaseModel_DRAGONxsec.fits.gz'#DRAGON output file
11 dragon2D_leptons = cosmicrays.Dragon2D(filename,[Electron, Positron])#load particle population
12
13 xsec = interactions.BremsstrahlungTsai74()
14 HI = neutralgas.RingModel(neutralgas.GasType.HI)
15
```

Method: python

1. Load pyhermes module
2. Define the particle population, the interaction and the target fields
3. Define the integrator with the objects above as arguments

```
18
19 integrator = BremsstrahlungIntegrator(dragon2D_leptons,HI,xsec)#define the integrator
20
21
```

Method: python

1. Load pyhermes module
2. Define the Particle population, Interaction and Target field
3. Define the Integrator using the objects above as arguments
4. Set Observer position and introduce it in the integrator

```
20
21 sun_pos = Vector3QLength(8.3*kpc,0*kpc,0*kpc)#Earth's position in the Galaxy
22 integrator.setupCacheTable(60, 60, 20)  # To be adjusted based on performance needs
23 integrator.setObsPosition(sun_pos)#introduce Earth's position in the integrator
24
```

Method: python

1. Load pyhermes module
2. Define the Particle population, Interaction and Target field
3. Define the Integrator using the objects above as arguments
4. Set Observer position and introduce it in the integrator
5. Create the Skymap and introduce the integrator

```
26 nside = 128#map resolution
27 skymap_range = GammaSkymapRange(nside, 0.1*GeV,100*GeV,13)#define the map that we want to compute
28 skymap.setIntegrator(integrator)#introduce the integrator into the map
29
```

Method: python

1. Load pyhermes module
2. Define the Particle population, Interaction and Target field
3. Define the Integrator using the objects above as arguments
4. Set Observer position and introduce it in the integrator
5. Create the Skymap and introduce the integrator
6. Calculate the emissions

```
29
30 skymap.compute()
31
```

Method: python

1. Load pyhermes module
2. Define the Particle population, Interaction and Target field
3. Define the Integrator using the objects above as arguments
4. Set Observer position and introduce it in the integrator
5. Create the Skymap and introduce the integrator
6. Calculate the emissions
7. Optional (but recommendable): save the results

```
32 output_file = f'!HandsOn/2D_run_brems.fits'
33 output = outputs.HEALPixFormat(output_file)#define the file to save the results
34 skymap.save(output)#save results
```