

Open-Science Integration of a Combined Analysis of KM3NeT and CTA into the EOSC Infrastructure

On behalf of the KM3NeT collaboration



Mikhail
Smirnov



ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 824064.



Motivation



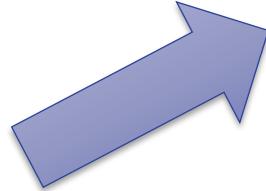
- ❖ Limitation of a single-experiment analysis in astrophysics
- ❖ Follow FAIR principles:
 - findable
 - accessible
 - interoperable
 - reusable
- ❖ Creation of the first open-science enabled repository with scientific analysis from KM3NeT
- ❖ Provide integration with European Open Science Cloud (EOSC) ESCAPE cell services

Raising a scientific problem



Multi-messenger particles (ν and γ) carry information about the origin of distant sources

Emission scenario



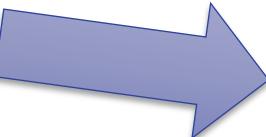
Leptonic

Inverse Compton (IC)

$$e^\pm + \gamma \rightarrow e^\pm + \gamma^*$$

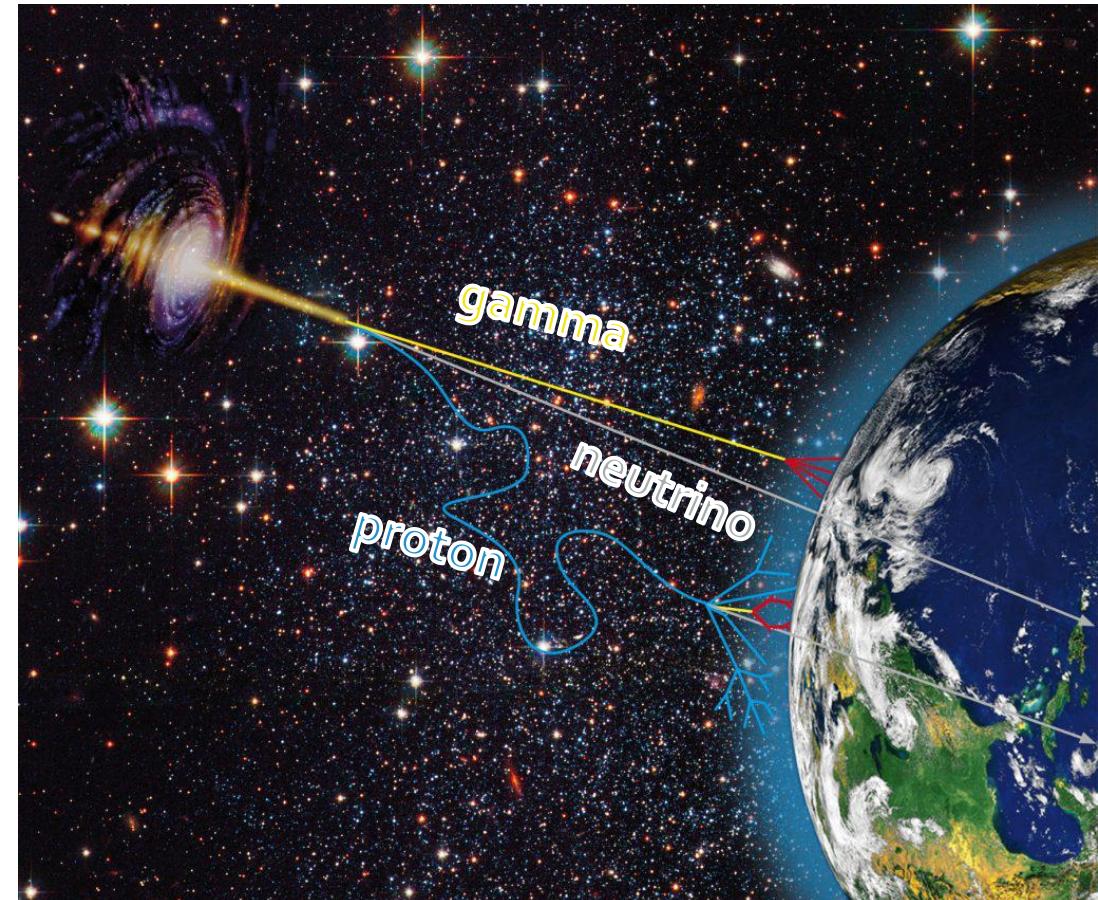
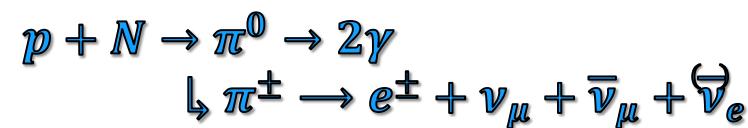
No neutrinos!

Neutrinos play a key role to disentangle two emission scenarios



Hadronic

Pions Decay (PD)



Astrophysical telescopes KM3NeT (neutrino)



KM3NeT (cubic kilometer neutrino telescope) [J.Phys. G43 \(2016\) 084001](#)

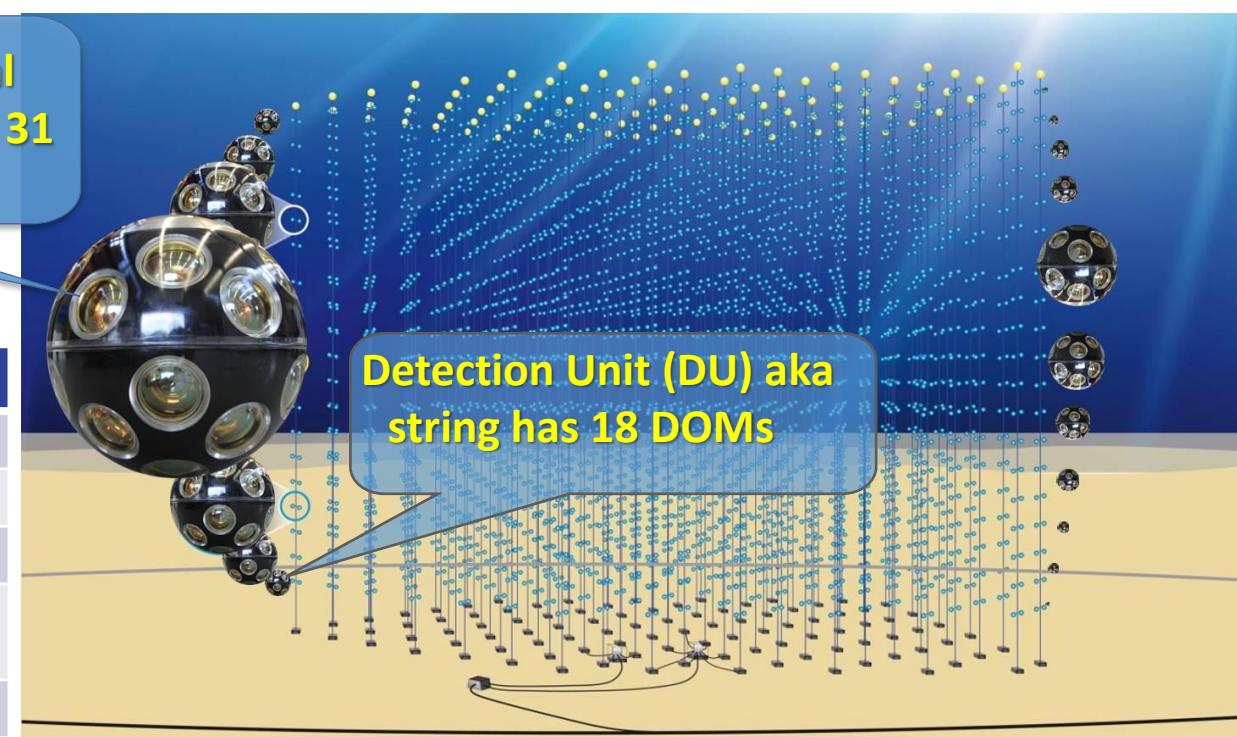
KM3NeT/ARCA (Astroparticle Research with Cosmics in the Abyss)
discovery and observation of HE cosmic neutrino sources
($E_\nu \sim \text{GeV-PeV}$) high energy neutrinos
Depth - 3500 m - offshore Sicily (Italy)

KM3NeT/ORCA (Oscillation Research with Cosmics in the Abyss)
determination of the neutrino mass hierarchy
($E_\nu \sim \text{MeV - GeV}$) low energy neutrinos
Depth - 2500 m - offshore Toulon (France)



Parameter	ARCA	ORCA
DU distance	90 m	20 m
DOM spacing	36 m	9 m
DU height	~ 800 m	~ 200 m
Instrumented mass	2*500 Mton	7 Mton
Amount of DUs	115*2	115
Current status	21	19

Digital Optical
Module (DOM) 31
of 3" PMTs



Astrophysical telescopes CTA (gamma)



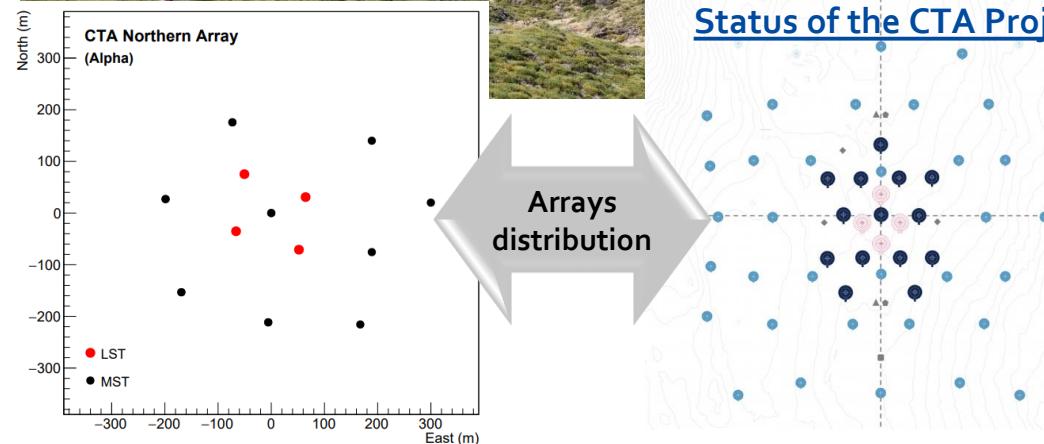
CTAO (Cherenkov Telescope Array Observatory) [ICRC21](#)

CTA Northern site La Palma in the Canary Islands, Spain.

Altitude 2180 m. Low and mid E_γ (20 GeV – 5 TeV)

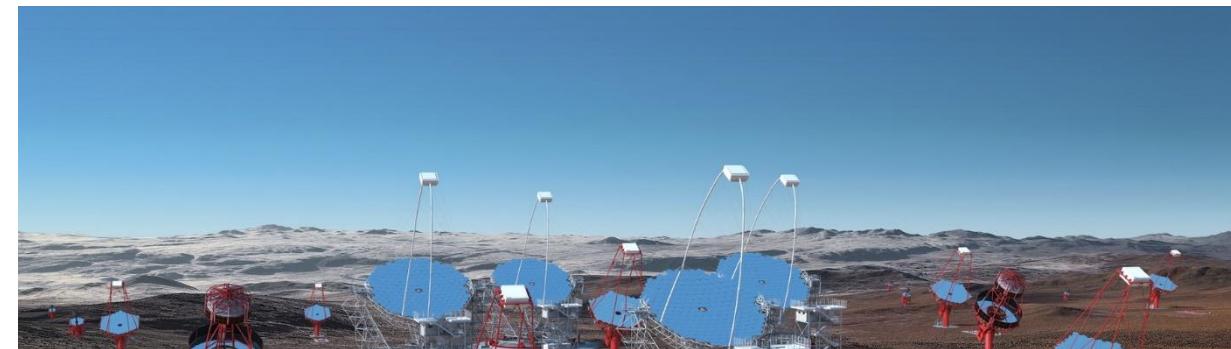
Extragalactic physics

- 4 large-sized telescopes (low gammas)
- 9 medium-sized telescopes (mid gammas)



CTA Southern site Paranal Observatory in the Atacama Desert, Chile. Altitude 2150 m, several km². Mid and high E_γ (150 GeV – 300 TeV). Galactic sources

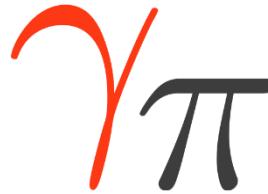
- 37 small-sized telescopes (high gammas)
- 14 medium-sized telescopes (mid gammas)
- 4 large-sized ???



Alpha Config	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	
LST North	Comissioning and Operation of LST1									Operation as 4 LST Array		
	CDR		Deployment of LST2-4									
MST North	Design and Finance		INFRA	Construction of 9MSTs								
	Array config, Finance and CDR									Construction and Deployment of 14 MSTs		
CTA South										Construction and Deployment of 37 SSTs		
Extension												
LST South	2020			2021			2022	2023	2024	2025	2026	
	Finance / CDR			Construction of 4 LSTs ???			2027					
	Operation ???											

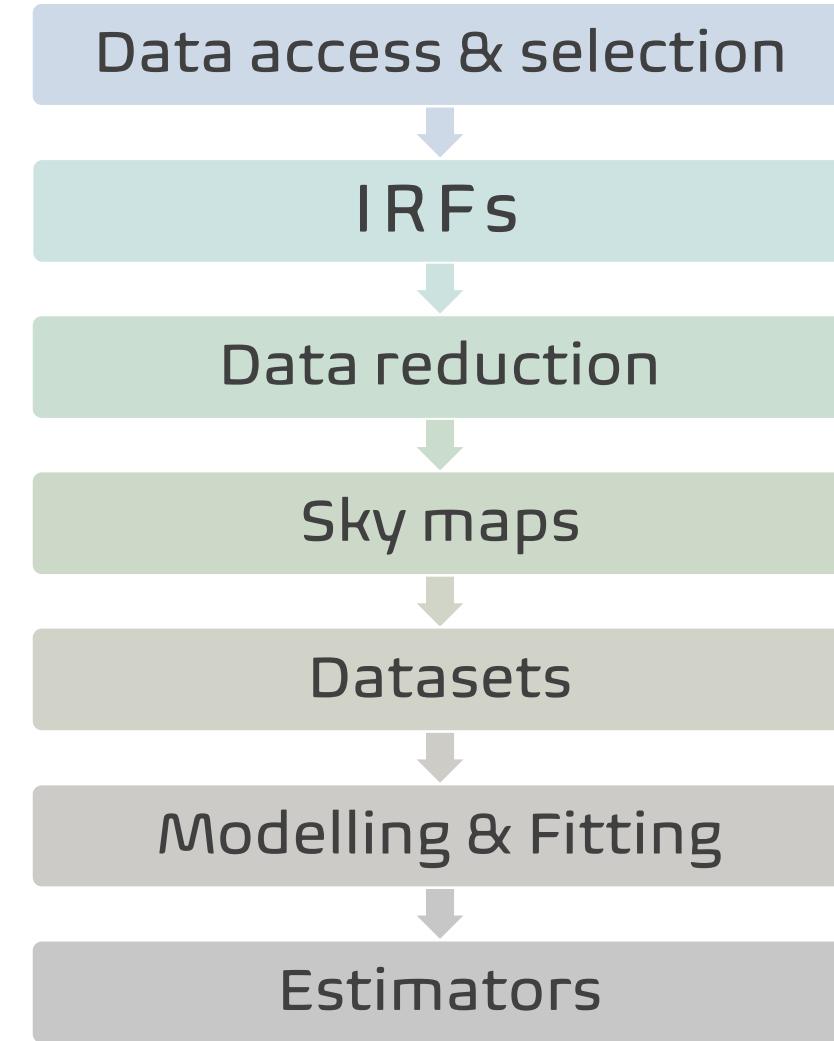
Gammapy python package

Open-source software for CTA



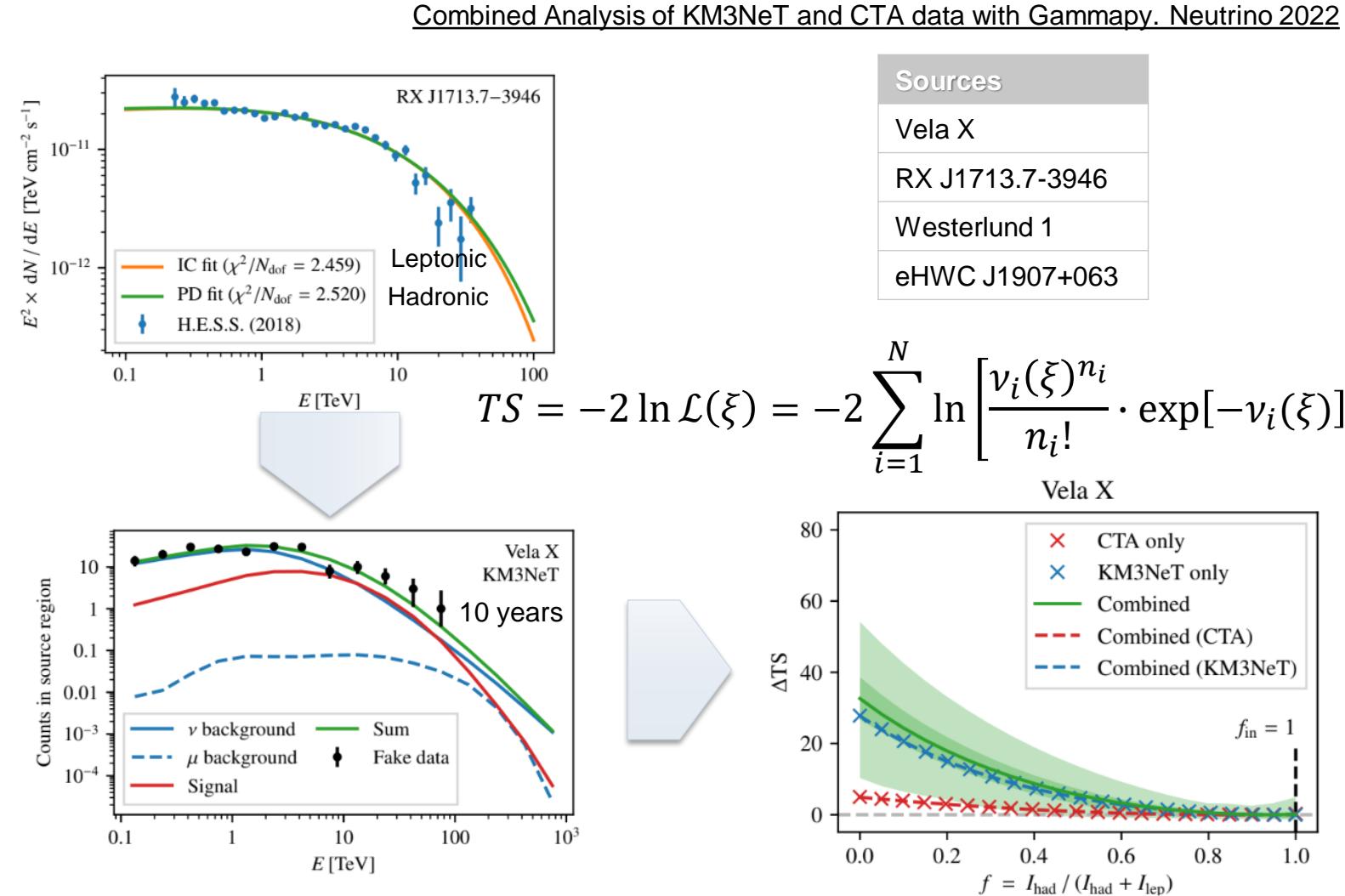
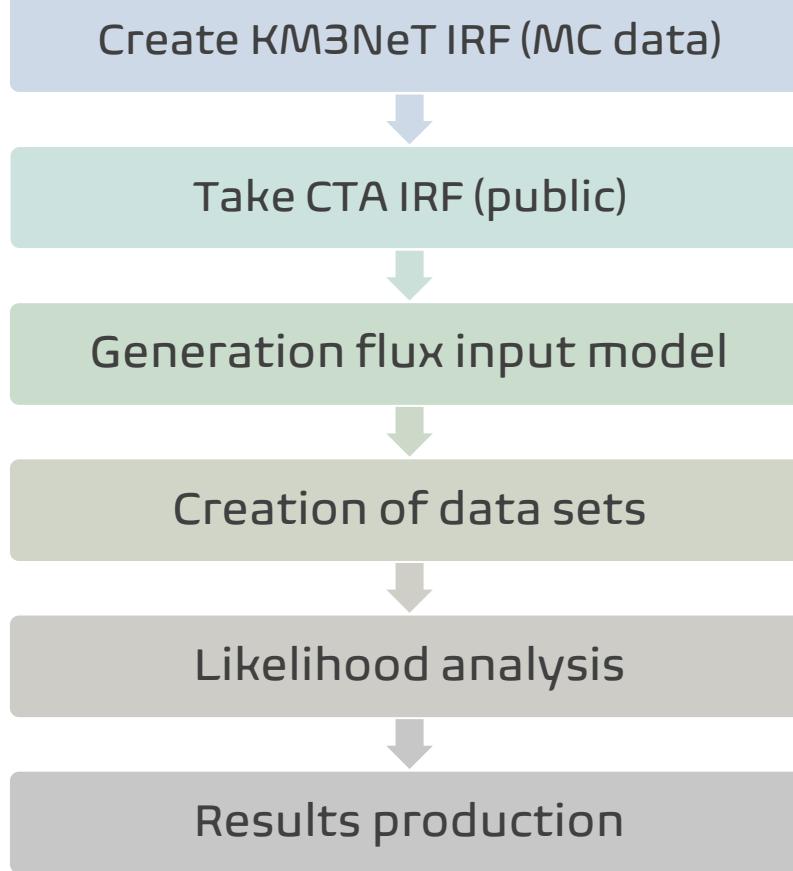
<https://gammapy.org/>

A **Python** package for
gamma-ray astronomy



Structure of the analysis

Performed by Tim Unbehaun et. al



EOSC/ESCAPE introduction



Science Clusters in eosc

Social sciences

Environmental sciences

Life sciences

Materials Health Energy

Astronomy Particle physics HPC

Astrophysics

Particle &
nuclear physics



<https://projectescape.eu/>

EOSC/ESCAPE test science projects (TSPs)



Data Lake:



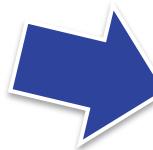
VRE:



Software Repo:



- TSPs
 - Dark Matter
 - Extreme Universe and Gravitational Waves



ESCAPE AAI
AAI Federated with EOSC

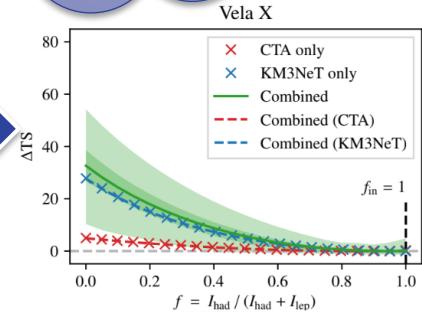
ESCAPE Data Lake
(Federated storage)
Includes EOSC Exchange provisioned storage

Virtual Research Environment
(old VRE)

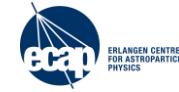
VRE@CERN
<https://vre-hub.github.io/>

Software Repository
Archive of reusable pipelines:
Onboarded to EOSC marketplace

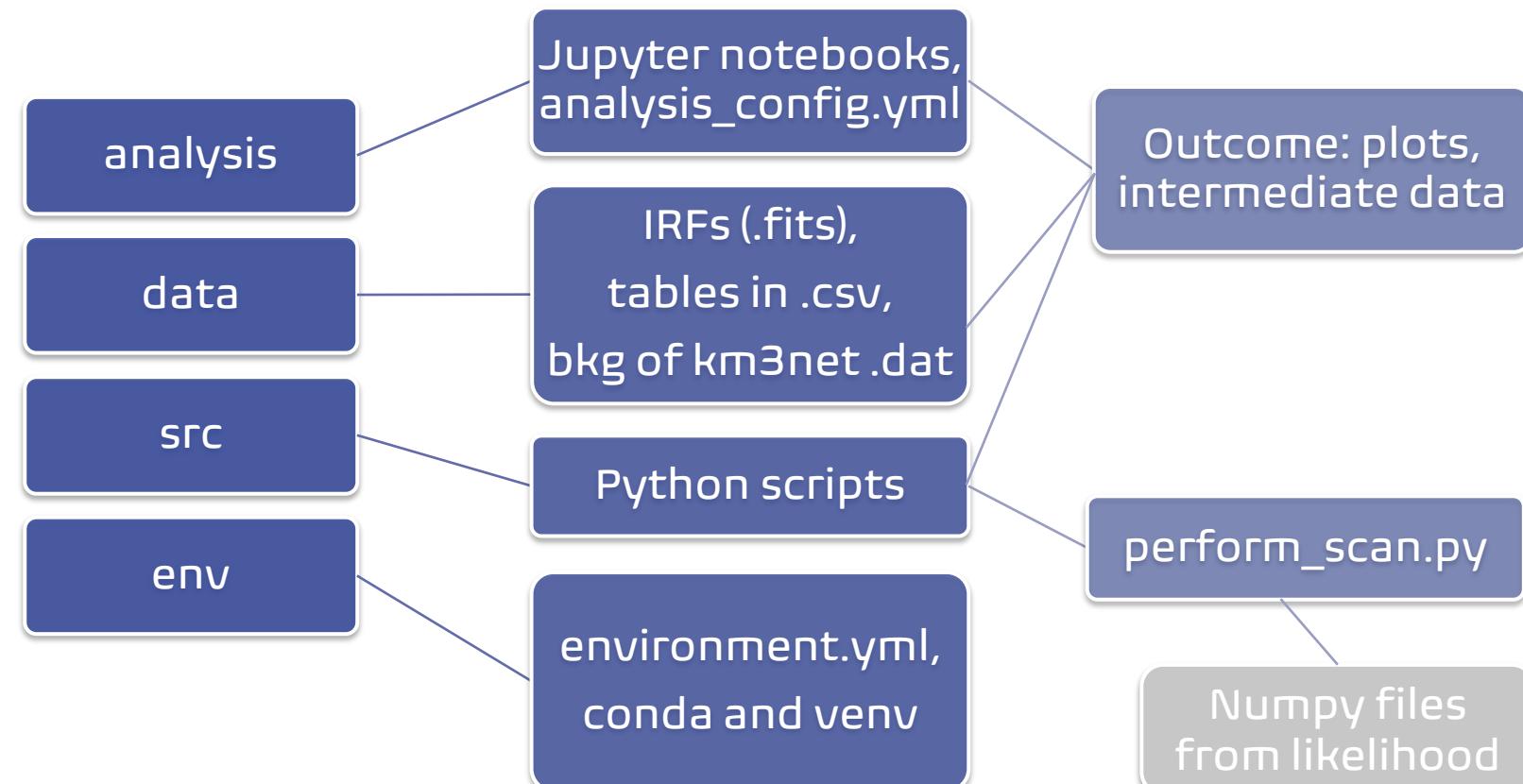
Cloud computing
using EOSC Exchange provisioned compute, & HPC



Repository with combined analysis CTA plus KM3NeT

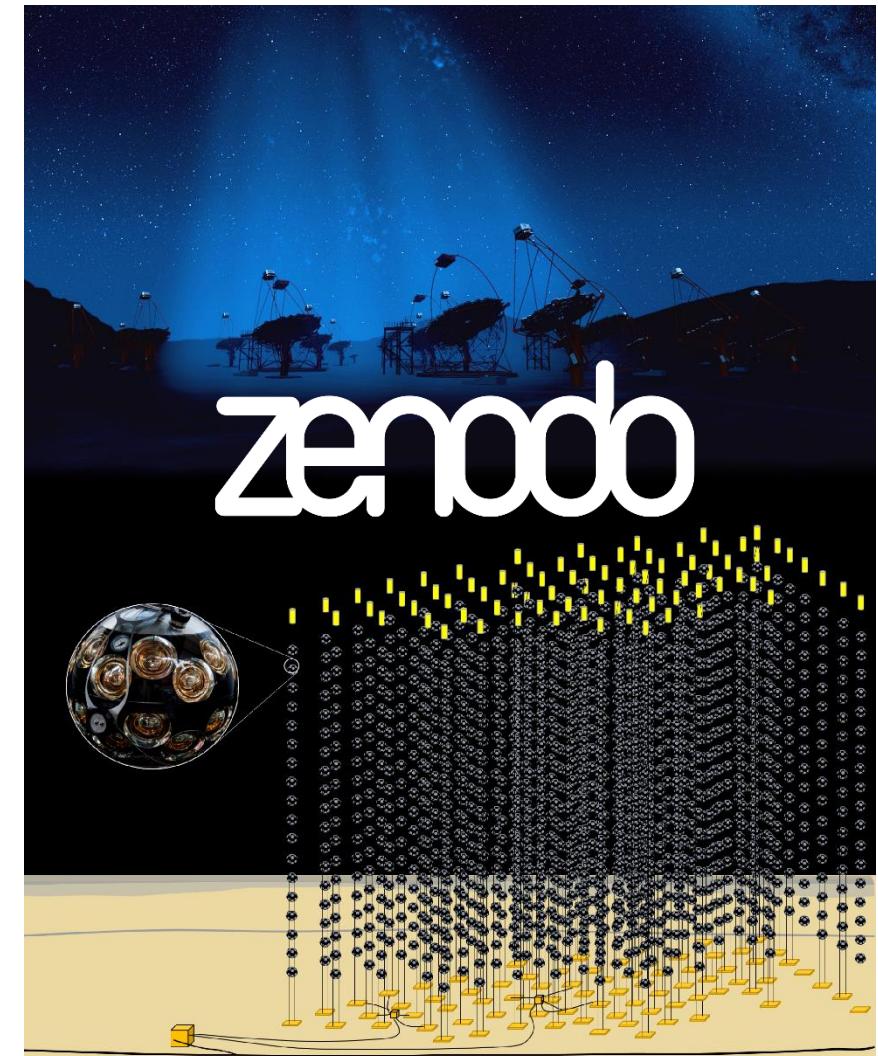


Analysis is based on `gammapy v0.17` and `python 3.8`

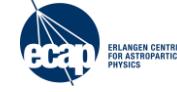


[DOI 10.5281/zenodo.8298096](https://doi.org/10.5281/zenodo.8298096)

<https://github.com/KM3NeT/Analysis-galactic-sources-CTA-KM3NeT>



Integration with EOSC/ESCAPE



REANA

DataLake
Rucio

Deployment
VRE github



```
reana_cta_dataset.yml 864 bytes
1 ---
2 version: 0.9.0
3 inputs:
4   directories:
5     - data/models/
6     - data/cta/
7     - src/
8
9 files:
10   - Analysis/reana/flux_models.py
11   - Analysis/reana/create_cta_dataset.py
12
13 parameters:
14   flux: flux_models.py
15   main: create_cta_dataset.py
16   path: Analysis/reana/
17
18 workflow:
19   type: serial
20   specification:
21     steps:
22       - name: analysis
23         environment: "gitlab-registry.in2p3.fr/escape2020/virtual-envi"
24         kubernetes_memory_limit: "4Gi"
25         commands:
26           # - mkdir "${plot}"
27           - cd "${path}" && python "${flux}"
28           - cd "${path}" && python "${main}"
29           #       - cd reana_analysis && python plot_from_fits.py
30         # --data "${data}"
31         # --plot "${plot}"
32     outputs:
33       directories:
34         - results/plots
35     files:
36       - results/CTA_VelaX_200h_p4.fits.gz
37
```

<https://reanahub.io/>

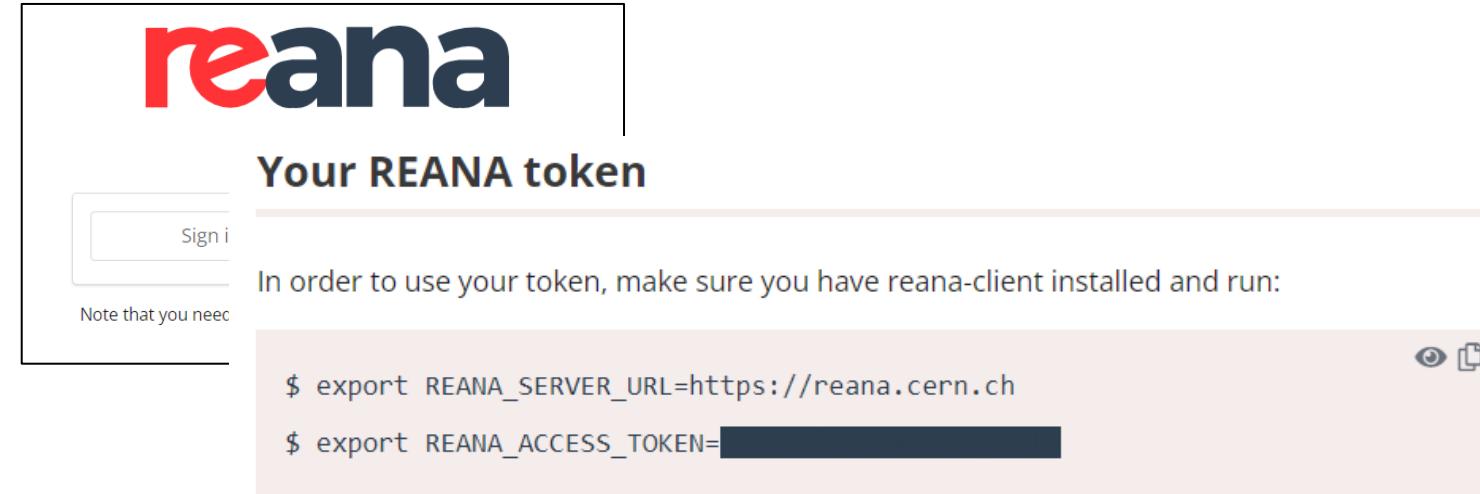
Integration with EOSC/ESCAPE



REANA

DataLake
Rucio

Deployment
VRE github



The image shows a screenshot of the REANA token generation interface. At the top, it says "Your REANA token". Below that, there is a note: "In order to use your token, make sure you have reana-client installed and run:". Underneath the note, there are two command-line export statements:

```
$ export REANA_SERVER_URL=https://reana.cern.ch  
$ export REANA_ACCESS_TOKEN=[REDACTED]
```

<https://reanahub.io/>

Integration with EOSC/ESCAPE



REANA

DataLake
Rucio

Deployment
VRE github

The screenshot shows the REANA hub interface. On the left, there is a sidebar with a "Sign in" button and a note: "Note that you need to have a REANA account to use this feature". Below this are two command-line snippets: "\$ export" and "\$ export". The main area is titled "Your workflows" and shows two completed entries:

- km3net_dataset #27**: finished in 6h 32m 42s, step 1/1. Last run: 2 months ago.
- km3net_dataset #26**: finished in 3h 22m 48s, step 1/1. Last run: 2 months ago.

At the top right, it says "Refreshed at 12:23:37 UTC".

<https://reanahub.io/>

Integration with EOSC/ESCAPE



REANA

DataLake
Rucio

Deployment
VRE github

The screenshot shows the REANA hub interface. At the top, there are two tabs: "Your RE" (selected) and "Your workflows". Below the tabs, there is a search bar and a dropdown menu for "Status". A large button labeled "\$ export" is prominently displayed. On the left, there are three sections: "jupyter" (with "Files", "Running", and "Clusters" tabs), "jupyter" (with "Files", "Running", and "Clusters" tabs), and a file browser showing a directory structure with files related to KM3NeT_VelaX_10y.

REANA

Your RE

In order to ...

\$ export

jupyter

jupyter

File browser:

- 0 / results / KM3NeT_VelaX_10y
- ..
- KM3NeT_VelaX_10y_data_nu1.fits
- KM3NeT_VelaX_10y_data_nu2.fits
- KM3NeT_VelaX_10y_data_nu3.fits
- KM3NeT_VelaX_10y_data_nu4.fits
- KM3NeT_VelaX_10y_data_nu5.fits
- KM3NeT_VelaX_10y_data_nu6.fits
- KM3NeT_VelaX_10y_data_nu7.fits
- KM3NeT_VelaX_10y_datasets.yaml
- KM3NeT_VelaX_10y_models.yaml
- KM3NeT_VelaX_10y_models_covariance.dat

File browser:

- 0 /
- Analysis
- data
- results
- src

<https://reanahub.io/>

Integration with EOSC/ESCAPE



REANA

DataLake
Rucio

Deployment
VRE github

Server Options

- Minimal environment**
Based on jupyter/scipy-notebook (active reana-client)
- ROOT environment**
ROOT v6.26.10, a C++ kernel is implemented too - DASK testing
- Minimal environment - python 3.9.13**
Contains a REANA client
- Virtual Observatory environment**
Contains Jupyter Notebooks examples with the basic usage of the IVOA tools
- Indirect Dark Matter Detection Environment**
Contains a GCC compiler and the MLFermiLATDwarfs and fermitools libraries - not fermipy (bugged)
- Common gamma analysis tools**
Contains a GCC compiler and astropy, sherpa, agnpy, gammipy libraries
- Wavelet Detection Filter (WDF) project environment**
Contains the full WDF env
- Compact stars Science Project environment**
Contains the matchmaker library
- KM3NeT Science Project environment**
Contains the common gamma analysis tools and the km3io, km3pipe and km3irf libraries
- KM3NeT & CTA combined analyses**
Compatible environment with gammipy and the km3io, km3pipe and km3irf libraries (env testing)

Integration with EOSC/ESCAPE



Server Options

- Minimal environment
Based on jupyter/scipy-notebook (active reana-client)
- ROOT environment
ROOT v6.26.10, a C++ kernel is implemented too - DASK testing
- Minimal environment - python 3.9.13
Contains a REANA client
- Virtual Observatory environment
Contains Jupyter Notebooks examples with IVOA compliant data
- Indirect Dark Matter Detection Environment
Contains a GCC compiler and the MLFermilac environment (bugged)
- Common gamma analysis tools
Contains a GCC compiler and astropy, sherpa, gammapy
- Wavelet Detection Filter (WDF) project environment
Contains the full WDF env
- Compact stars Science Project environment
Contains the matchmaker library
- KM3NeT Science Project environment
Contains the common gamma analysis tools
- KM3NeT & CTA combined analyses
Compatible environment with gammapy and cta-sim

RUCIO

File Edit View Run Kernel Tabs Settings Help

REAPER NOTEBOOK

Enter a Data Identifier (DID)

Search Everything ▾

SEARCH RESULTS

- KM3NET_ECAP_SP:Fits.Files.KM3NET
- KM3NET_ECAP_SP:aeff.fits

Available scopes

- IVOA_ASTRON_TEAM
- KM3NET_ECAP_MS
- KM3NET_ECAP_SP
- KM3NET_ECAP_TGAL
- KM3NET_FAU_JSCHNABEL
- KM3NET_FAU_SP
- KM3NET_NIKHEF_MBOUWHUIS

Launcher

ct-a-and-km3net

Notebook

Python 3 (ipykernel)

comb_cta

Console

Python 3 (ipykernel)

comb_cta

A screenshot of the REANA web interface. On the left, there's a sidebar with "Server Options" containing ten environment choices. The "Minimal environment" option is selected. To the right, there's a main area with the "RUCIO" interface showing search results for "KM3NET_ECAP_SP:Fits.Files.KM3NET" and "KM3NET_ECAP_SP:aeff.fits". Below the RUCIO interface is a "Launcher" section with icons for "ct-a-and-km3net", "Notebook", "Python 3 (ipykernel)", "comb_cta", "Console", "Python 3 (ipykernel)", and "comb_cta".

Integration with EOSC/ESCAPE



REANA

DataLake
Rucio

Deployment
VRE github



Virtual Research Environment

Analysis platform developed at CERN within the EOSC Future project to promote open science and collaboration between astroparticle physics communities.

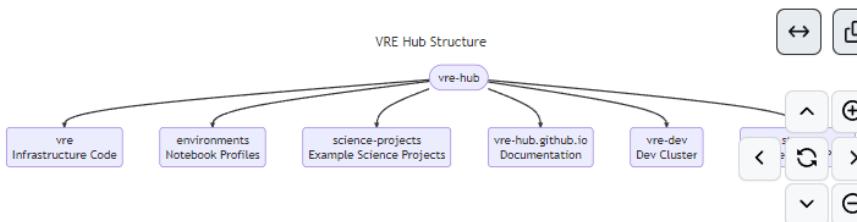
6 followers Switzerland <https://eoscfuture.eu> <https://vrehub.github.io/> escape-cern-ops@cern.ch

Follow

README.md

Virtual Research Environment

The Virtual Research Environment is an analysis platform developed at CERN serving the needs of scientific communities involved in European Projects. Its scope is to facilitate the development of end-to-end physics workflows, providing researchers with access to an infrastructure and to the digital content necessary to produce and preserve a scientific result in compliance with FAIR principles. The platform's development is aimed at demonstrating how sciences spanning from to could benefit from the usage of common technologies, initially born to satisfy CERN's exabyte-scale data management needs.



Find the VRE documentation and overview [here](#).

Any contribution is welcome! Feel free to create issues and start discussions! Find our wiki [here](#).

The platform development effort is part of the EOSC Future Project, an EU-funded H2020 project implementing the European Open Science Cloud (EOSC).

View as: Public ▾

You are viewing the README and pinned repositories as a public user.

Top discussions this past month

Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.

[Start a new discussion](#)

People

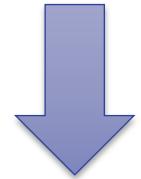


<https://github.com/vrehub>

Top languages

Shell C++ HTML HCL
Jupyter Notebook

Old VRE



Available in PYPI pip install km3irf

km3irf 0.3.1.dev1+g09c35be API Reference Examples Changelog Code Coverage 



Welcome to km3irf's documentation!

Note

This project is under active development.

Contents

[API Reference](#)

[km3irf](#)

[Examples](#)

[Collection of Examples](#)

[Changelog](#)

[Unreleased changes](#)

[Version 0](#)

[Code Coverage](#)

[pipeline](#)  [coverage](#) 85.76% [docs](#) latest

KM3NeT instrument response function

This project provides a versatile tool that can be used to quickly analyze the sensitivity of the **KM3NeT** detector for various source models. Currently it considers only point-like sources. The main feature of the tool is deep targeting to [gammapy](#) software. For further analysis in [gammapy](#), [km3irf](#) provides next modules:

- Event list
- Instrument response function (IRF)
- Data set

Installation

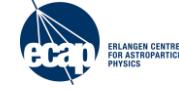
It is recommended to first create an isolated virtualenvironment to not interfere with other Python projects:

```
git clone https://git.km3net.de/km3py/km3irf
cd km3irf
python3 -m venv venv
. venv/bin/activate
```

- IRF generation for point-like sources
- Compatible with [gammapy](#)
- Output
 - DL3 format with [EventList](#) and [IRFs](#)
 - [DataSet](#)

km3irf: example №1

Production of combined IRF .fits file



**Neutrino
and Cosmic
Rays
generators**

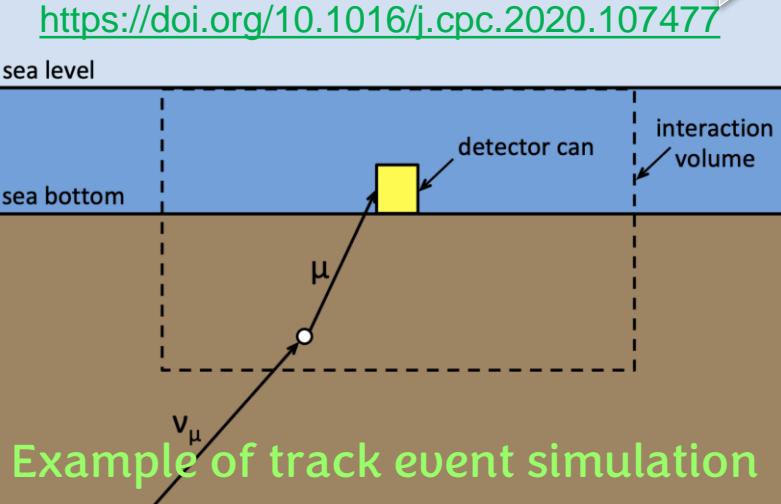
Example № 1

This example produces *Effective Area* (Aeff), *Point Spread Function* (PSF), *Energy Dispersion* (Edisp) files in .fits format from original KM3NeT simulation dst.root file. And finally merge them into one common .fits file.

**Light
generators**

```
data import data_path
rt build_irf
import fits
s import merge_fits
warnings.simplefilter("ignore")
```

**Light
propagation
and
collection**



Describing workflow

Input dst.root files

Create DataContainer object

Processing Aeff .fits

Processing PSF .fits

Processing Edisp .fits

Merge in combined .fits

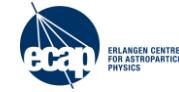
Check content with gammapy

Reconstruction

**Data
production**

km3irf: example №1

Production of combined IRF .fits file

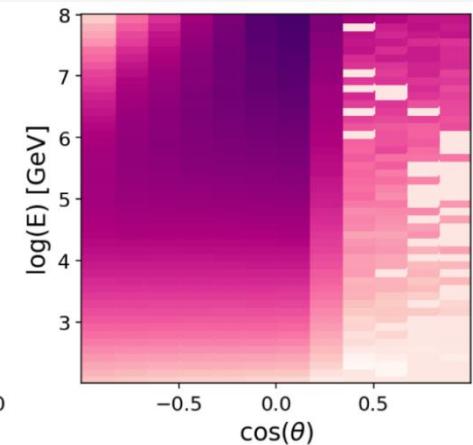
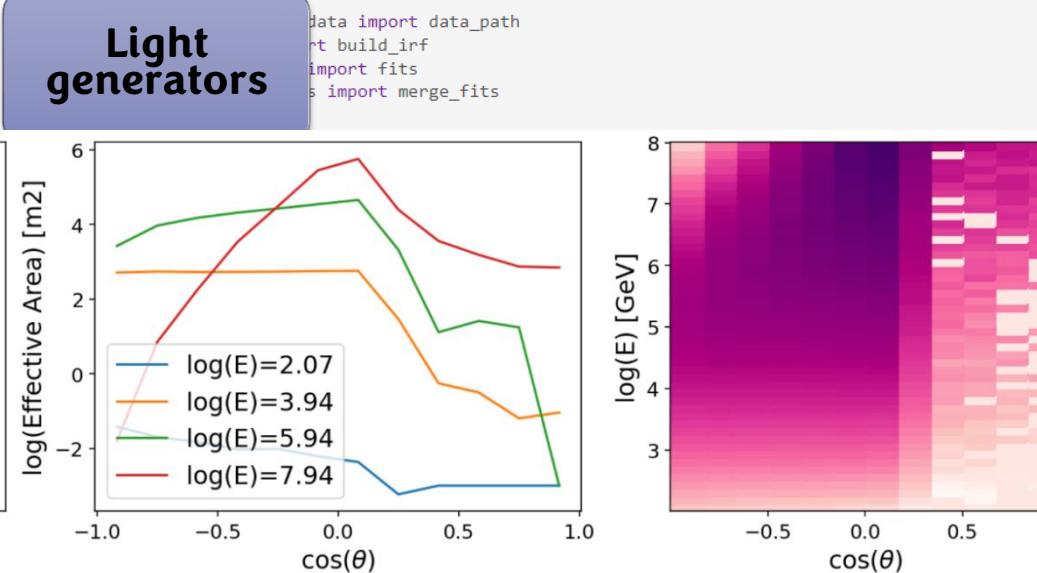
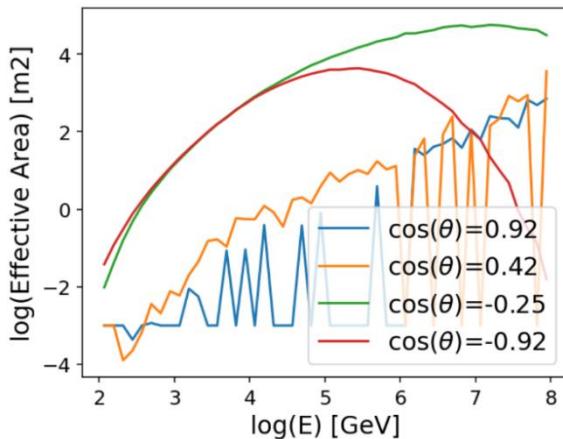


**Neutrino
and Cosmic
Rays
generators**

Example № 1

This example produces *Effective Area (Aeff)*, *Point Spread Function (PSF)*, *Energy Dispersion (Edisp)* files in .fits format from original KM3NeT simulation dst.root file. And finally merge them into one common .fits file.

**Light
generators**



Example of track event simulation

**Data
production**

Describing workflow

Input dst.root files

Create DataContainer object

Processing Aeff .fits

Processing PSF .fits

Processing Edisp .fits

Merge in combined .fits

Check content with gammapy

**Thank you
for your attention!**



More: [Open Science in KM3NeT](#)
Jutta Schnabel, 31.08, Outreach
and Education 4



Backup / analysis scheme in gammapy

