Prototype Open Event Reconstruction Pipeline for the Cherenkov Telescope Array

Nöthe, M.; Kosack, K.; Nickel, L.; Peresano, M.; CTA Consortium

DOI
10.22323/1.395.0744

Publication date
2022

Document Version
Final published version

Published in
Proceedings of Science

License
CC BY-NC-ND

Citation for published version (APA):
Prototype Open Event Reconstruction Pipeline for the Cherenkov Telescope Array

M. Nöthe\textsuperscript{a,}*\textsuperscript{a}, K. Kosack\textsuperscript{b}, L. Nickel\textsuperscript{a} and M. Peresano\textsuperscript{b} for the CTA Consortium

\textsuperscript{a}TU Dortmund University, Otto-Hahn-Str. 4a, Dortmund, Germany
\textsuperscript{b}AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, F-91191 Gif-sur-Yvette, France
E-mail: maximilian.noethe@tu-dortmund.de

The Cherenkov Telescope Array (CTA) is the next-generation gamma-ray observatory currently under construction. It will improve over the current generation of imaging atmospheric Cherenkov telescopes (IACTs) by a factor of five to ten in sensitivity and it will be able to observe the whole sky from a combination of two sites: a northern site in La Palma, Spain, and a southern one in Paranal, Chile. CTA will also be the first open gamma-ray observatory. Accordingly, the data analysis pipeline is developed as open-source software. The event reconstruction pipeline accepts raw data of the telescopes and processes it to produce suitable input for the higher-level science tools. Its primary tasks include reconstructing the physical properties of each recorded shower and providing the corresponding instrument response functions.

ctapipe is a framework providing algorithms and tools to facilitate raw data calibration, image extraction, image parameterization and event reconstruction. Its main focus is currently the analysis of simulated data but it has also been successfully applied for the analysis of data obtained with the first CTA prototype telescopes, such as the Large-Sized Telescope 1 (LST-1).

pyirf is a library to calculate IACT instrument response functions, needed to obtain physics results like spectra and light curves, from the reconstructed event lists.

Building on these two, protopipe is a prototype for the event reconstruction pipeline for CTA. Recent developments in these software packages will be presented.
1. Introduction

The Cherenkov Telescope Array (CTA)\footnote{www.cta-observatory.org} will be the next generation very-high-energy gamma-ray observatory, sensitive to energies between $\sim 20$ GeV and $300$ TeV. It will be composed of over fifty imaging atmospheric Cherenkov telescopes (IACTs) built at two sites to achieve full sky coverage: one on the Canary Island of La Palma, Spain and the other near Paranal, Chile.

CTA will detect gamma rays by measuring the Cherenkov light emitted by extensive air showers, however these are also induced by charged cosmic rays, which form a large background to gamma-ray observations. The data analysis pipeline of CTA starts with the pre-calibrated raw data from the telescopes in the form of time series data for each pixel and for each telescope that registered a signal from the current shower. The pipeline proceeds to reconstruct the physical properties of the primary particle for each recorded shower, this includes the gamma ray’s energy and arrival direction. To remove most of the cosmic-ray induced air showers, also a particle type classification is required.

In the classical analysis approach, the raw data is reduced and aggregated into higher levels of abstraction before finally employing a set of machine learning and geometrical algorithms to reconstruct the physical properties of the primary particle. This is usually performed as a four step procedure, which is currently implemented in ctapipe: image extraction, image cleaning, image parametrization and finally the reconstruction of primary particle properties (see Figure 1). These steps will be detailed in section 2. After reconstruction of the shower events, one last step – based on the pyirf library described in section 3 – selects the best ones on the basis of the specific science case at hand and allows to produce the instrument response functions (IRFs). The pipeline prototype called protopipe and described in section 4 performs the analysis steps from raw simulated data to IRF production based on both libraries.
2. ctapipe

calipe is a python package providing library functions and command-line tools to perform the tasks listed in the previous section. It is developed as open-source software and the project, started in 2015, is hosted on Github. Since then, 26 versions have been released but it is still under heavy development (the latest release at the time of writing is 0.11.0 [7]). In total, 44 contributors have made this project possible. Releases are published to PyPI and conda packages are provided using conda-forge\(^2\). ctapipe builds upon the scientific python stack with the main dependencies being astropy [16] for astronomical computations and unit support, numpy [3] and scipy [17] for numerical algorithms and statistics and pytables\(^3\) for IO using HDF5\(^4\). The jit-compiler numba [8] is used to optimize performance-critical parts of the code base.

2.1 Image Extraction

The first step in the ctapipe analysis is to reduce the time-series information, i.e. the digitized signals of the Cherenkov photosensors, to the number of photons and their mean arrival time in each pixel. ctapipe supports different algorithms for extracting these quantities from single-pixels waveforms, from simple peak finding algorithms to more complex ones which combine the waveforms of multiple pixels or that fit the expected time evolution of the shower and use that to define the integration window for each pixel.

2.2 Image Cleaning

This operation is aimed at identifying pixels which are likely to host real Cherenkov signal. This is usually done by applying a pixel-wise selection via cleaning thresholds based on the photo-electron and peak time values output by the image extraction step. Again, ctapipe supports multiple algorithms to solve this task.

2.3 Image Parametrization

After removal of noise pixels, the cleaned image goes through a parametrization in order to make it exploitable by subsequent algorithms, in particular shower geometry reconstructors and/or machine-learning models that assist with with the event property reconstruction. Among the most important parameters are the classical Hillas parameters [5], which describe the orientation and extension of the shower image in the camera, which is needed for the following reconstruction steps. Additionally, ctapipe implements general descriptive statistics of the images, morphological features like the number of isolated pixel groups and parameters describing the containment of the shower’s image in each camera.

2.4 Reconstruction of Event Properties

While the first three steps can be performed individually for each telescope in the array (monoscopic), this step needs to combine the information from all telescopes to give one common estimate for a recorded shower (stereoscopic).

---

\(^2\) conda-forge.org  \(^3\) www.pytables.org  \(^4\) www.hdfgroup.org/solutions/hdf5/
The stereoscopic reconstruction of physical shower parameters can be performed in \texttt{ctapipe} by either of two currently supported approaches: moments-based and template-based.

The moments-based method makes use of a reconstructor which takes as input the parametrized moments of each image (in the default approach the Hillas parameters) from a candidate shower. This input is then combined with a pair-wise geometric reconstruction where each pair of images gets a weight based on the brightness of the images. In case of single, monoscopic telescopes, machine learning can also be used for the reconstruction of the origin, as the geometrical approaches require multiple telescopes.

\texttt{ctapipe} also supports the \texttt{ImPACT} \cite{14} algorithm, an advanced template-based likelihood optimization to reconstruct the event properties, where the expected image for a given set of event properties is calculated from simulations, stored in a database of template images which is then used to perform a likelihood fit to the observed image.

2.5 Input / Output, visualization and configuration

IACT events are read from input files using the \texttt{EventSource} interface, which can be implemented for custom file formats using the \texttt{ctapipe} plugin system\footnote{E.g.: \url{github.com/cta-observatory/ctapipe_io_lst}}. There are built-in event source implementations for the simulation file format and \texttt{ctapipe}'s own output data format. \texttt{ctapipe}'s data model uses its own data structure, called \texttt{Container}, which can be written to and loaded from HDF5 files, supporting transformations and metadata including units.

The \texttt{ctapipe.visualization} module provides classes to display both camera images and telescope array configurations. Two implementations currently exist, one using \texttt{matplotlib} \cite{6} and one using the \texttt{bokeh}\footnote{\url{bokeh.org/}} library.

The \texttt{ctapipe} configuration system is build using \texttt{traitlets}\footnote{\url{traitlets.readthedocs.io/}}, the configuration system developed for IPython. A full configuration tree is built by configurable classes called \texttt{Components} that can include configurable member attributes. Command-line tools use the same configuration mechanism and allow passing configuration for all configurable objects either on the command line or via a configuration file. Many options can be set per telescope type or even per telescope.

3. Calculating Instrument Response Functions (IRFs) using \texttt{pyirf}

To be able to estimate physical properties of gamma-ray sources from lists of reconstructed events, the instrumental response to the initial gamma-ray signal must be known. This will depend on the instrument, the specific analysis, environmental conditions and more. In general, the instrumental response of a gamma-ray telescope can be described by the following integral equation, transforming true properties of the gamma rays into the observable quantities:

\[
e(\hat{\alpha}, \hat{\delta}, \hat{E}, t) = \int R(\hat{\alpha}, \hat{\delta}, \hat{E}|\alpha, \delta, E, t) \cdot I(\alpha, \delta, E, t) \, d\Omega \, dE + b(\hat{\alpha}, \hat{\delta}, \hat{E})
\]

Where $\alpha$, $\delta$ and $E$ are the right ascension, declination of the gamma ray origin and its total energy, while $\hat{\alpha}$, $\hat{\delta}$, $\hat{E}$ are the corresponding reconstructed quantities obtained from the analysis pipeline. $I$ is the source term, the true gamma-ray signal arriving at earth at the given position, energy and time $t$. $R$ is the IRF, the convolution kernel translating true quantities to the observed ones, $b$ is
the irreducible background and $e$ is the expected event distribution as measured by the experiment. The solid angle integration over $\alpha, \delta$ is denoted using $d\Omega$.

The IRF can only be estimated from labeled data, where the true and reconstructed quantities are both known. In the case of CTA these labeled datasets are created via Monte Carlo simulations using CORSIKA \cite{4} to simulate the extensive air showers, followed by the detector simulation performed by sim_telarray \cite{1}.

In classical IACT analysis, the IRF is factorized into three independent components, making the strong assumption that the migrations between the different observables are statistically independent. This factorization yields:

$$ R(\hat{\alpha}, \hat{\delta}, \hat{E}|\alpha, \delta, E, t) = A_{\text{eff}}(\alpha, \delta, E, t) \cdot \text{PSF}(\hat{\alpha}, \hat{\delta}|\alpha, \delta, E, t) \cdot D(\hat{E}|\alpha, \delta, E, t) $$  \hspace{1cm} (2)

Where $A_{\text{eff}}$ is the effective area, the detection probability times the observed area for a gamma ray with given true properties, PSF is the point spread function, i.e. the convolution kernel for the reconstructed gamma-ray origin and $D$ is the energy dispersion, the migration between true energy $E$ and reconstructed energy $\hat{E}$. Instead of continuous functions, these IRFs are calculated and stored as binned quantities filled from simulated events.

**pyirf** is a python library for calculating these IRFs from labeled, reconstructed event lists as created by the event reconstruction pipeline. The latest version of **pyirf** at the time of writing is v0.5.0 \cite{12}, which supports calculating most IRFs formats defined in Gamma-Astro-Data-Formats (GADF, \cite{2}) and can export these into the FITS-based data format defined therein. Additionally, **pyirf** contains functionality to calculate flux sensitivity of gamma-ray instruments according to the requirements laid out for CTA and the optimization of event selection criteria to obtain the best flux sensitivity.

### 4. protopipe

**protopipe** is a pipeline prototype for CTA based on the **ctapipe** and **pyirf** libraries. It is distributed as a python package on the PyPI platform; the latest release at the time of writing is 0.4.0.post1 \cite{15}. Started as an independent project for image cleaning studies by the CTA Consortium group at CEA-Saclay/IRFU, it has been developed as an open-source package for the whole consortium since September 2019. Since then, its development has been steered by the will to substitute the historical pipelines currently in use for the production of the official IRFs for CTA. Such pipelines have been inherited from the VERITAS (EventDisplay \cite{10}) and MAGIC (MARS \cite{19}) experiments and adapted to the CTA scenario by their maintainers. Even if they provide satisfactory results, they are not in line with the software requirements of CTA and not easily exploitable by the whole consortium. The development of **protopipe** is strongly influenced by a step-by-step comparison with such pipelines, which translates in a continuous code migration into the **ctapipe** and **pyirf** libraries (algorithms and support of additional analysis operations).

**protopipe** has been built around the two libraries described in this work by constantly trying to support their latest stable releases. It also provides a module for multivariate analysis using supervised machine-learning techniques (**protopipe.mva**), used to reconstruct energy and particle type of the events.
The pipeline provides four tools based on ctapipe, protopipe.mva, ctapipe and pyirf respectively. Each tool is a python executable configurable via YAML-based configuration files. protopipe also provides a way to launch the analysis on computing grids featuring the DIRAC interware. The tools can be launched on the grid thanks to an interface code developed separately from the main package and based on CTADIRAC, a version of the DIRAC middleware customized for CTA.

4.1 Description of the pipeline workflow

A full dataset composed of simulated events from primary gammas, protons and electrons is split at the beginning of the analysis in sub-datasets. Depending on the workflow of choice, a step of the pipeline will correspond to a tool being applied to one or more sub-datasets. The currently tested workflow is the following:

- part of the gamma rays are used to train an energy reconstruction model,
- part of the gamma rays and part of the protons are used to train a particle classification model (making use also of the reconstructed energy),
- the remaining gamma rays and protons together with the full electron dataset are fully analyzed.

In a real scenario of a gamma-ray analysis, the entire third sub-dataset and the proton sub-dataset used to train the particle classifier would correspond to data observed by the telescope array. An overview of the workflow is shown in Figure 2 and the tools are defined in the following sections.

![Figure 2: Current pipeline workflow tested on full-scale analyses on the GRID. The actions performed by the tools 1, 2, 3 and 4 are highlighted by green, orange, red and pink arrows respectively. Reconstructed energy is used as a model feature when training particle classification (black dashed arrow)](image)

4.1.1 Tool 1: preparation of training data

This tool is based on ctapipe and it produces data in a format suitable for model training. This format is a combination of data levels as defined by the data models in ctapipe: DL1b (image parameters) and part of DL2 (reconstructed shower geometry). The transformation of raw data into DL1b data makes use of the library capabilities described in paragraphs 2.1 to 2.3. Since the pipeline workflow currently tested (see Fig.2) comprises the use of two models (one for energy reconstruction and the other for particle classification) this tool is used in two separate steps of that analysis.

---

*dirac.readthedocs.io/en/latest/*  
*github.com/cta-observatory/CTADIRAC*
4.1.2 Tool 2: production of models

The production of machine-learning models is performed by the tool based on the `protopipe.mva` module. The dependencies are few: numpy and pandas to deal internally with tables of data, joblib for I/O support and scikit-learn to create the models and fit the test data. The models currently tested are part of the `sklearn.ensemble` module: AdaBoostRegressor or RandomForestRegressor for energy reconstruction and RandomForestClassifier for particle classification. It is possible to perform tuning of the hyper-parameters via an exhaustive search over lists of parameter values specified by the user (`sklearn.model_selection.GridSearchCV`). The tool outputs both model and tables of the events selected for training and testing as gzip-compressed pickled objects.

4.1.3 Tool 3: production of fully-analyzed events

This tool performs the full reconstruction pipeline and is applied to events which have to be independent from those used by the previous tools. The operations performed are those described by paragraphs 2.1 to 2.4. In particular the tool requires as an input the models produced by Tool 2 in order to reconstruct both energy and particle type. The models’ input file format currently supported is the one output by the Tool 2.

4.1.4 Tool 4: production of IRFs and optimized cuts

This tool is based on the functions provided by the `pyirf` library and performs the following sequence of operations:

1. find the best cutoff in gammaness score, which is the result of the particle type classification, to best discriminate between signal and background, as well as the angular cut to obtain the best sensitivity for a given amount of observation time and a given template for the source of interest,
2. estimate the sensitivity from the optimized cuts,
3. compute the IRFs from the same selected events.

The current output format is the one supported by `pyirf`: it builds on the data format specification given by the GADF integrated by input coming from CTA optimizations.

5. Conclusions and Outlook

`ctapipe` and `pyirf` offer open-source tools to solve a critical part of the analysis of IACT data. Using the IO plugin system, `ctapipe` can be used to process data by all experiments, see for example [13] for a combined analysis of LST-1 and MAGIC observations. While the current version of `ctapipe` performs event property estimation using geometrical or template based algorithms, the use of modern machine learning techniques is also investigated (see for example [18] and [11]). Performance of an analysis using `ctapipe` and `pyirf` on simulated data and first results on data from observations performed by the LST-1 are reported in [9]. `protopipe` is being currently developed with the goal of superseding the reference analyses for the planned arrays, currently performed by EventDisplay and MARS. It takes into account all supported cameras, optics and array configurations for CTA, enabling a high degree of flexibility to accommodate diverse instrument
configurations. It will be used to produce sets of IRFs sufficiently large to describe and investigate the performance of CTA under any required observing condition and science case and to analyze data from the whole set of telescopes.

Acknowledgements: We gratefully acknowledge financial support from the agencies and organizations listed here: www.cta-observatory.org/consortium_acknowledgments

References


Prototype Open Event Reconstruction Pipeline for CTA

M. Nöthe et al. for the CTA Consortium


Prototype Open Event Reconstruction Pipeline for CTA

M. Nöthe et al. for the CTA Consortium

Prototype Open Event Reconstruction Pipeline for CTA

M. Nöthe et al. for the CTA Consortium

68. Center for Astrophysics and Cosmology, University of Nova Gorica, Vipavska 11c, 5270 Ajdovščina, Slovenia
69. Institut für Astronomie und Astrophysik, Universität Tübingen, Sand 1, 72076 Tübingen, Germany
70. Research School of Astronomy and Astrophysics, Australian National University, Canberra ACT 0200, Australia
71. Department of Physics and Astronomy, University of California, Los Angeles, CA 90095, USA
72. INFN Sezione di Bari and Politecnico di Bari, via Ora branca 4, 70124 Bari, Italy
73. Laboratoire de Physique des 2 infinis, Irene Joliot-Curie,IN2P3/CNRS, Université Paris-Saclay, Université de Paris, 15 rue Georges Clemenceau, 91406 Orsay, Cedex, France
74. INFN Sezione di Pisa, Largo Pontecorvo 3, 56217 Pisa, Italy
75. IRFU/DEIDIP, CEA, Université Paris-Saclay, Bat 141, 91191 Gif-sur-Yvette, France
76. INAF - Osservatorio Astronomico di Padova, Vicolo dell’Osservatorio 5, 35122 Padova, Italy
77. INAF - Osservatorio Astronomico di Palermo “G.S. Vaiana”, Piazza del Parlamento 1, 90134 Palermo, Italy
78. School of Physics, University of Sydney, Sydney NSW 2006, Australia
79. Sorbonne Université, Université Paris Diderot, Sorbonne Paris Cité, CNRS/IN2P3, Laboratoire de Physique Nucléaire et de Hautes Énergies, LPNHE, 4 Place Jussieu, F-75005 Paris, France
80. Instituto de Fisica de São Carlos, Universidade de São Paulo, Av. Trabalhador Sáo-carlense, 400 - CEP 13566-590, São Carlos, SP, Brazil
81. Departement de Physique Quantique et Astrofisica, Institut de Ciencias del Cosmos, Universitat de Barcelona, IEEC-UB, Martí i Franquès, 1, 08028, Barcelona, Spain
82. Department of Physics, Washington University, St. Louis, MO 63130, USA
83. Saha Institute of Nuclear Physics, Bidhannagar, Kolkata-700 064, India
84. INAF - Osservatorio Astronomico di Capodimonte, Via Salita Moiariello 16, 80131 Napoli, Italy
85. Università di Parma, CNRS, Artroparticule et Cosmologie, 10, rue Alice Domon et Léonie Duquet, 75013 Paris Cedex 13, France
86. Astronomy Department of Faculty of Physics, Sofia University, 5 James Bourchier Str., 1164 Sofia, Bulgaria
87. Institut de Recherche en Astrophysique et Planétologie, CNRS-INSU, Université Paul Sabatier, 9 avenue Colonel Roche, BP 44346, 31028 Toulouse Cedex 4, France
88. School of Physics and Astronomy, University of Minnesota, 116 Church Street S.E. Minneapolis, Minnesota 55455-0112, USA
89. IRFU, CEA, Université Paris-Saclay, Bât 141, 91191 Gif-sur-Yvette, France
90. INAF - Istituto di Radioastronomia, Via Gobetti 101, 40129 Bologna, Italy
91. INAF - Istituto di Astrofisica Spaziale e Fisica Cosmica di Palermo, Via U. La Malfa 153, 90146 Palermo, Italy
92. Astronomical Observatory, Department of Physics, University of Warsaw, Aleje Ujazdowskie 4, 00478 Warsaw, Poland
93. Armagh Observatory and Planetarium, College Hill, Armagh BT61 9DG, United Kingdom
94. INFN Sezione di Catania, Via S. Sofia 64, 95123 Catania, Italy
95. INAF - Osservatorio Astronomico di Brera, Via Brera 28, 20121 Milano, Italy
96. Kavli Institute for Particle Astrophysics and Cosmology, Department of Physics and SLAC National Accelerator Laboratory, Stanford University, 2575 Sand Hill Road, Menlo Park, CA 94025, USA
97. Universidade Cruzeiro do Sul, Núcleo de Astrofísica Teórica (NAT/UCS), Rua Galvão Bueno 8687, Bloco B, sala 16, Liberdade 01506-000 - São Paulo, Brazil
98. Universidad de Valparaíso, Blanco 951, Valparaíso, Chile
99. INAF - Istituto di Astrofisica e Planetologia Spaziali (IAPS), Via del Fosso del Cavaliere 100, 00133 Roma, Italy
100. Lund Observatory, Lund University, Box 43, SE-22100 Lund, Sweden
101. The Henryk Niewodniczański Institute of Nuclear Physics, Polish Academy of Sciences, ul. Radzikowskiego 152, 31-342 Cracow, Poland
102. Escola de Engenharia de Lorena, Universidade de São Paulo, Área I - Estrada Municipal do Campinho, s/n, CEP 12602-810, Pte. Nova, Lorena, Brazil
103. INFN Sezione di Trieste and Università degli Studi di Trieste, Via della Scienza 208, 34014 Trieste, Italy
104. Palacky University Olomouc, Faculty of Science, RCPTM, 17. listopadu 1192/12, 771 46 Olomouc, Czech Republic
105. IRFU/DEIDIP, CEA, Université Paris-Saclay, Bat 141, 91191 Gif-sur-Yvette, France
106. Monterrey Institute of Technology and Education, Faculty of Engineering and Sciences, Campus Tecnológico de Monterrey, Campus Monterrey, Monterrey, N.L., 64849, Mexico
107. Dublin City University, Glasnevin, Dublin 9, Ireland
108. Dipartimento di Fisica - Università di Milano, Via Celoria 16, 20133 Milano, Italy
109. Tata Institute of Fundamental Research, Homi Bhabha Road, Colaba, Mumbai 400005, India
110. Lund Observatory, Lund University, Box 43, SE-22100 Lund, Sweden
111. The Henryk Niewodniczański Institute of Nuclear Physics, Polish Academy of Sciences, ul. Radzikowskiego 152, 31-342 Cracow, Poland
112. University of Oxford, Department of Physics, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, United Kingdom
113. School of Physics & Astronomy, University of Southampton, University Road, Southampton SO17 1BJ, United Kingdom
114. Department of Physics and Technology, University of Bergen, Museplass 1, 5007 Bergen, Norway
115. Western Sydney University, Locked Bag 1797, Penrith, NSW 2751, Australia