Gammapy + VERITAS Perspectives and Challenges

Matthew Lundy, Ken Ragan, Sam Wong, Ste O'Brien, on behalf of the VERITAS Collaboration

Introduction

Validation

Outstanding Issues/Problems

Future Potential

Introduction to VERITAS Software: Our Pipeline and *Gammapy*

General Background

- VERITAS has two separate analysis chains both of which we use to produce DL3 files. These are EventDisplay (ED) and VEGAS.
- Currently we have ~30 gammapy users.
    - ~10 PhD Students and Postdocs working actively on validation
- Differences are normally expected between ED and VEGAS due to the implementation of different reconstruction technique.
- We are currently working toward producing DL3 files with ED/VEGAS and performing high level analyses in gammapy (significances, sky maps, spectra, light curves)
- Our converter from VERITAS .root to DL3 .fits is public : https://github.com/VERITAS-Observatory/V2DL3
- EventDisplay is also public, but does not contain IRFs: https://github.com/VERITAS-Observatory/EventDisplay_v4

Structure

Trace Level Data
VERITAS Bank Format
(.cvbf)

VEGAS

- Full event reconstruction
- Image Template Method
- Standard gamma-hadron separation cuts
- More *On* events, and higher backgrounds
- More sensitive to bright sources and better gamma-ray PSF
- Outputs .root event lists

V2DL3 Conversion to VEGAS DL3 Files

Currently only Point Source

EventDisplay

- Full event reconstruction
- Boosted decision tree determined gamma-hadron separation
- More gamma-ray pure signal but lower counts
- More sensitive to dimmer sources.
- Outputs .root event lists

V2DL3 Conversion to ED DL3 Files

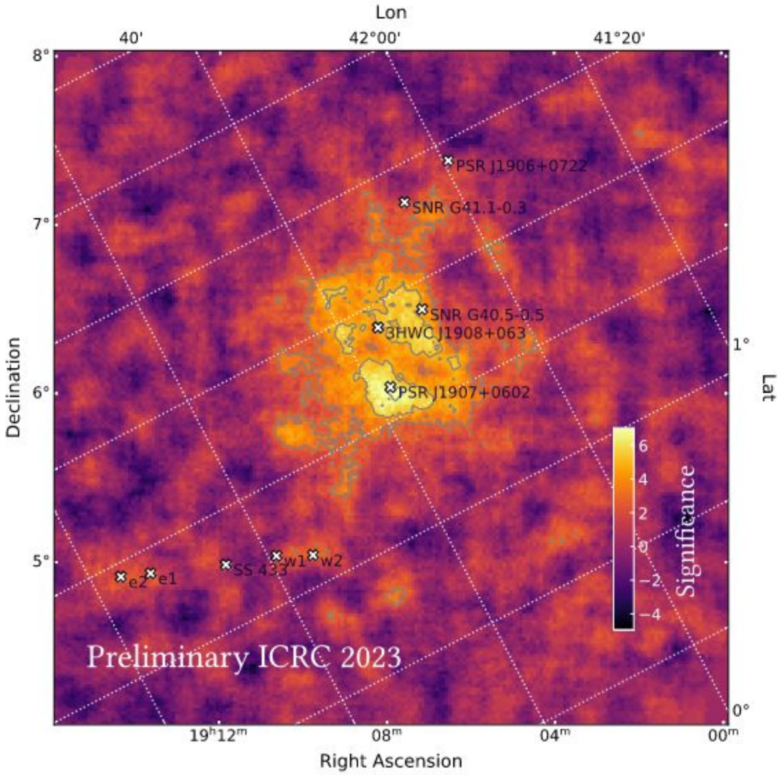
Point Like + Full Enclosure, No BKGs

Custom Scripts for BKG generation

$\gamma\pi$ A **Python** package for **gamma-ray** astronomy

Use of *Gammapy* - Adoption + Validation

- VERITAS plans to adopt *gammapy* **as a standard method** but there have been challenges due to validation. We have the joint problem of ensuring that there are no small errors in our files and then additionally ensuring that there are no errors in the gammapy pipeline that we are working on.
- The degeneracy between these two has caused difficulty in validation as we often spend a lot of time trying to work out if the error is **instrumental**, **expected** from the difference between methods, or **a bug** in some section of code (i.e. issue #5245).
- Gammapy is currently being used in a narrowly validated range of use cases (>800 GeV and point source spectra), and only as a cross-check of traditional analysis methods. We hope to be able to fully approve and adopt gammapy **by the end of the summer/early fall for 3D analysis** and plan to use it in a number of interesting papers that are being worked on (i.e. LHAASO source analysis)
- Gammapy education has been internal and we normally run ~2-3 tutorials per year. The main challenge in the past that we encountered with education mostly came down to preparation of DL3 files, but now we have streamlined most of this process.



Validation with VERITAS Data: Methodology, Reasoning, and Results

Validation

Validation means several things in the context of using VERITAS data in *gammapy*.

Are our DL3 files (full enclosure and point-like) generated correctly or are there **additional errors** introduced in the conversion?

How should we be **generating our BKG headers** for 2D and 3D analyses?

What are the **optimal cuts for the inputs into *gammapy*** that should be treated as defaults for the analysis?

Can *gammapy* reproduce the functionality of the legacy packages or **are there areas where the legacy products do something better**?

What are the **joint-systematics of VERITAS+gammapy** that need to be factored into results that incorporated into results produced with *gammapy*?

Validation

Validation sources include Crab, PG 1553+113, PKS 1424+240, Mrk 421, Mrk 501, and Segue 1. We are currently cross checking the following:

- ❑ *On* and *Off* counts, alpha, and significance at the source location match those in the legacy package
- ❑ No artifacts in the significance and excess maps
- ❑ Significance distribution normal, and matching the width of the legacy package
- ❑ Fitted spectrum parameters match those in the legacy spectrum
- ❑ Flux points match those in the legacy packages
- ❑ No outliers in the cumulative significance plot

Our validation sources include data from every era of VERITAS (V4, V5, V6) and include a mixture of camera offsets, zeniths, azimuths, and noise levels/conditions. In order to cross-check the *Off* counts we have to use a method that is also in the legacy data products (i.e. ring background or reflected regions)

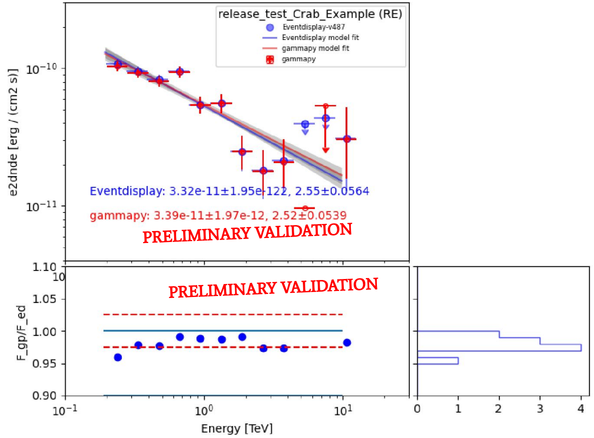
Validation- 1D ED Point Source

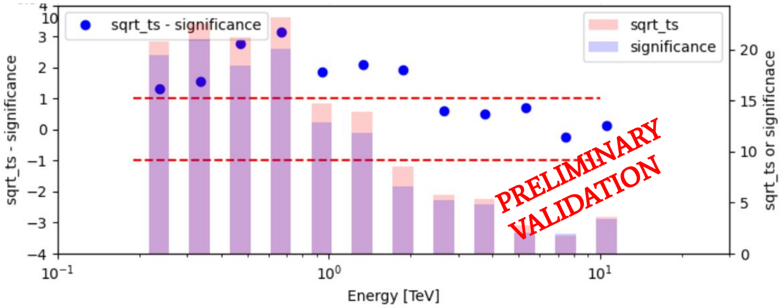
The only fully validated analysis with VERITAS and Gammapy is 1D spectral analysis with point source IRFs and ED.

Here we see agreement on the level of 5-10% in flux and <4 sigma in significance.

We attribute some of the spectral differences to:

- Different energy folding / unfolding methods applied
- Instrument response functions are interpolated per run for the γπ analysis, while it is done per event in the Eventdisplay analysis
- Differences possibly due to the accuracy of coordinate transformation

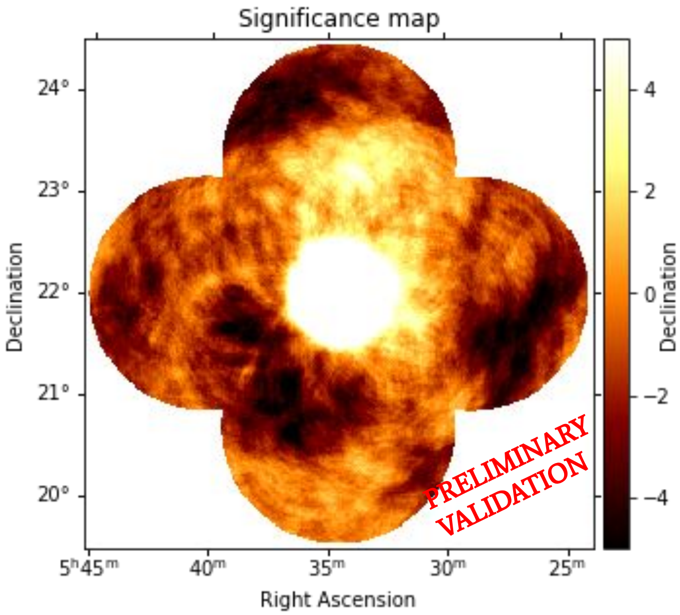


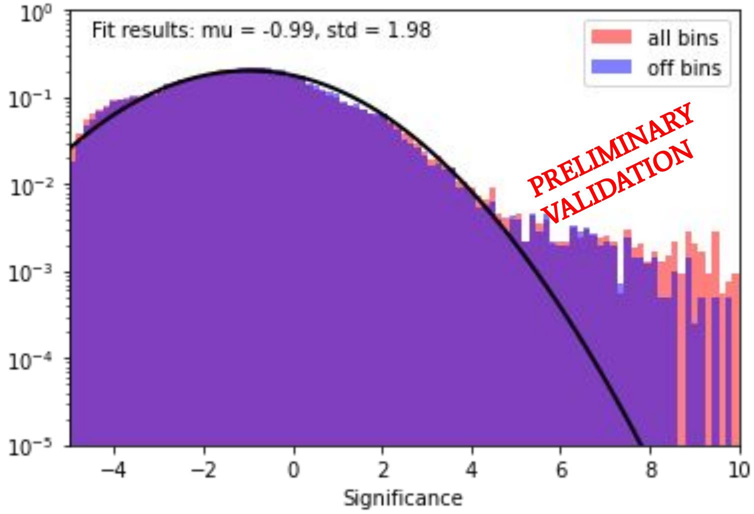


Validation – Ring background

- Analysis methods differ (e.g., only reflected region background calculations are available in our legacy packages), so we are **directly comparing ON counts** for identical integration regions and expect **significances to be in agreement**
- Resolved differences between the packages:
  - Sky map energy range must be set iteratively to match legacy package counts
  - Safe mask maker's offset-max must be set to ~1.5 for point sources to avoid ring artifacts in sky maps
  - correlate_off must be set to false to avoid double counting off counts and artificially reducing alpha
- Unresolved differences between the packages:
  - ON counts are still ~+/- 1 count/run (30 min) different than legacy packages (same events)
  - We see 5-10% differences in significance between packages
  - Integration radii of > 0.1 deg (for extended source analysis) create artifacts of the camera edges in sky maps

Validation – Ring background





Large positive and negative fluctuations & wide, shifted significance distributions for extended integration regions (0.35 deg)

Validation – Ring background

- The difference in *On* counts has a small amount to do with coordinate transformations but on top of that there are additional difference that are dependent on the energy axis in unintuitive ways (expanding the region decreases the counts).
- If we just take the *On* counts and sum them from the DL3 file manually we see agreement with the legacy packages, so at some point (with binning or a transformation) we are losing or gaining events.

Source	*ON* Counts *gammapy*	*ON* Counts ED	#counts difference/run
Crab v4	9029	9131	0.92
Crab v5	10150	10260	1.28
Crab v6	24670	24900	1.68
PG 1553	1494	1592	3.38
Mrk 501 v4	582	601	0.68
VER J0521+211	601	640	3.25

Validation – Ring background

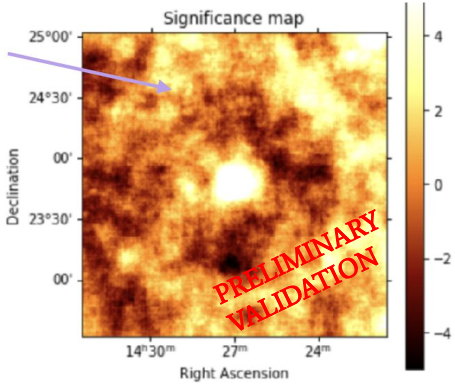
- The significances also differ quite dramatically and it's unclear if these are random errors or if there are regions where *gammapy* is performing worse/better.
- In this particular case, the WStat being used by *gammapy* is the same as the Li & Ma significance of the legacy package so this is not due to differences in the statistic being used.

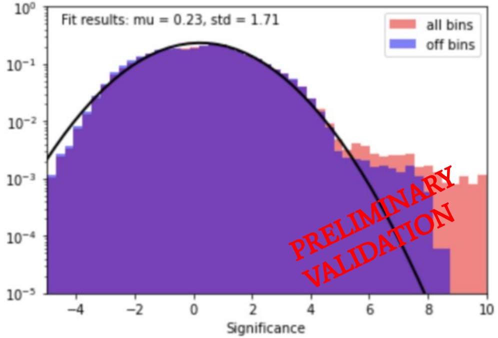
Source	Significance *gammapy*	Significance EventDisplay	% difference
Crab v4	174.74	161	7.86
Crab v5	161	172.7	7.27
Crab v6	302.59	269.3	11.00
PG 1553	34.24	32.71	4.47
Mrk 501 v4	36.07	34.09	5.49
VER J0521+211	21.129	20.077	4.98

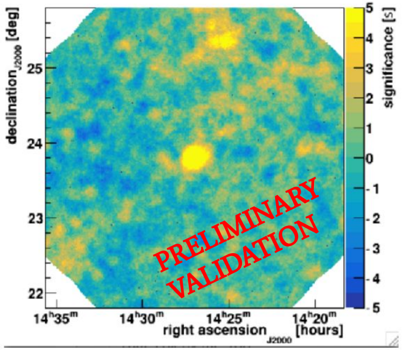
Validation – Ring background

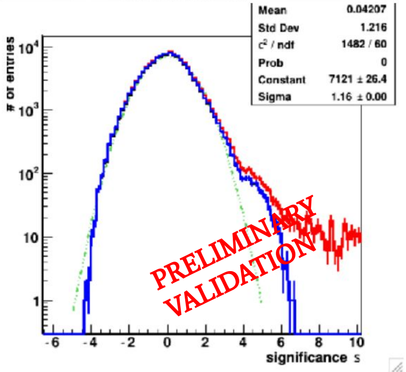
- We also find that *gammapy* is very sensitive to small changes in some parameters. A common feature that appears is a ring like structure and a very broad significance distribution. This occurs when we don't properly cut on the offset or have minor tweaks to the construction of BKGs. We find that this systematics is very common when analyses aren't done perfectly.

Ring structure!









Outstanding Problems and Quality of Life Improvements:

Discussion of User Experiences within Gammapy with VERITAS Data

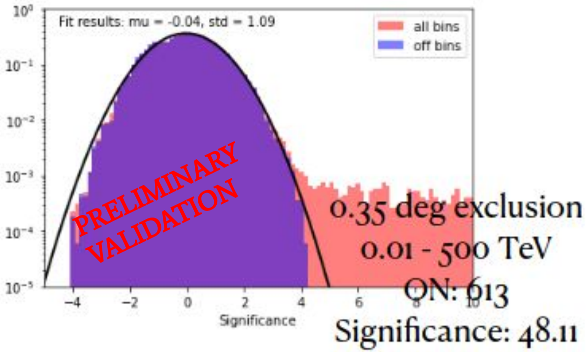
Issues - Internal Questions of Consistency

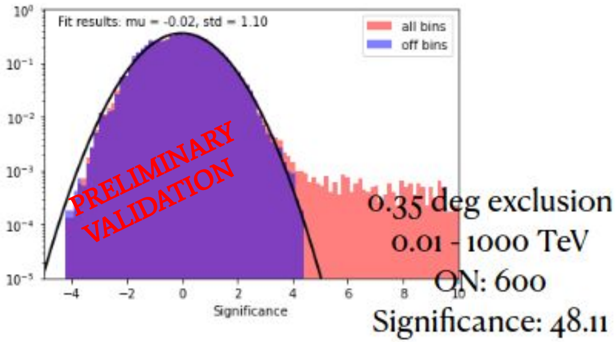
Since one of our validation checks is the number of *On* counts, we have found that *On* counts will often vary with parameters that should not have any effect on the number of *On* counts.

The two main ones that we've been exploring are due to the true energy axis and due to the containment correction.

To our understanding the containment correction should only affect exposure and should not have any direct impact on the number of counts yet it does, we are currently investigating where this comes from.

Other questions about fluxes reported from spectral and LC output are also being investigated for internal consistency.





Statistics - Improvements

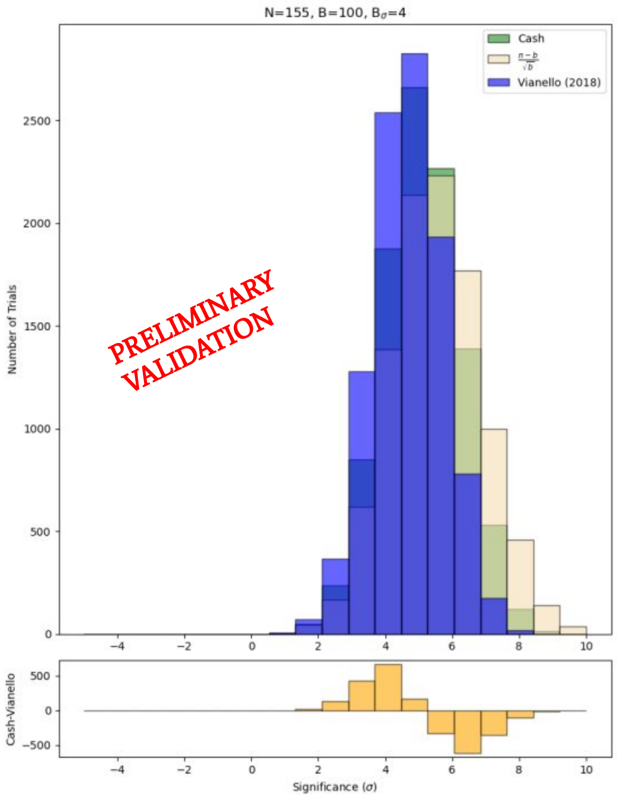
Due to the limited statistics available in gammapy, with some of these ported over from the X-ray or GeV domain, we see a bias in results for background dominated regimes like those present in most IACT datasets.

This gives the Test Statistics (TS) in gammapy an artificial boost when practically all that's happening is a lack of accounting of the conditional probability of your backgrounds.

By switching from Cash (1979) statistics to Vianello (2018) statistics that account for a gaussian error on the background estimation for FoVBackground we find that for many discovery sources a shift in significance on the order of ~0.5 sigma which accounts for some of the sensitivity "improvement" that certain methods seem to show over legacy packages.

We are also working on incorporating trials factors and other look-elsewhere effect corrections into the *gammapy* reporting of significance.

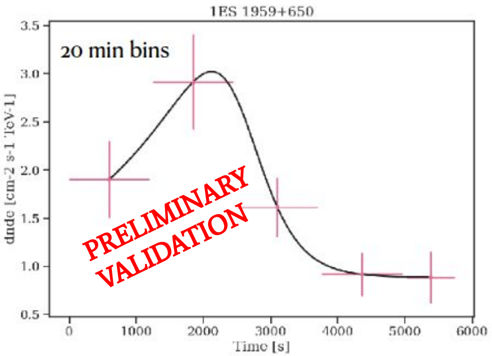
Internal memo on the safe use of certain statistics in *gammapy* is in progress and can be circulated after completion.

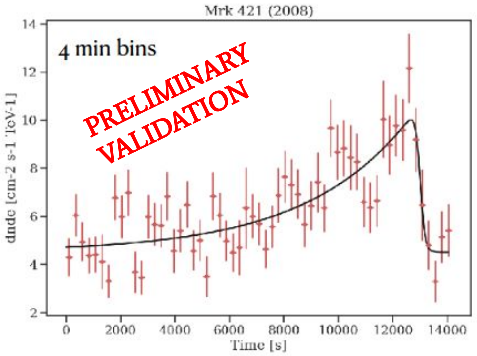


LightCurve - Convenience Issues

Multiple users have encountered some issues with the construction of custom LC templates and have also run into issues of importing LC objects from other telescopes with overlapping time bins (bin edges match). It's been difficult to create models that combine multiple temporal models and then additionally hard to fit piecewise broken models. Often users have resorted to having to use one of the default models even for simple flares.

Having more documentation about the format of LCs and a more flexible format might help with importing legacy data into gammapy objects rather than right now what's more convenient which is to export the LCs once they have been produced and do much of your work outside of gammapy.





gammapy flux points fit with minuit

Quality of life improvements

Right now, VERITAS users have been writing scripts and sharing them among ourselves to do some of the following but we would like to help incorporate some of these changes that are welcomed into the main bulk of the code rather than spending more time building this into a wrapper.

- Return alpha map from ExcessMapEstimator (github issue #5247)
  - Alpha maps are great diagnostics for methods or fields where you have a lot of complications. Having a these as a part of the FluxMaps object would be very useful.
- Easier to return analysis summaries.
  - High level results can be extracted from some of these methods but it's often confusing which are the true analysis results. Similarly to when you print(MapDatasetOnOff) it returns a list of basic information, excess map estimator it should be very simple to extract summary source information from a FluxMap. to_spectrum_dataset returns the proper information but the values mismatch those calculated by ExcessMapEstimator with the same integration region.
- Ring Background Spectrum
  - Knowing clearer which methods can return spectra and which cannot would be useful to know. There's a lot of flexibility in gammapy so understanding the regions where things have been tested and where they have not is very helpful.
- Fitting of extended objects
  - Extended objects currently have to be fit with an iterative approach and very tight constraints on parameters or else your fit will often try to model the entire FoV or escape to a null region on the edge. Even fitting the Crab with very flexible Gaussians will often cause errors if limits are not set for the fit parameters. I believe some of this has to do with NaN handling at the edges and the default stepping hyperparameters that *gammapy* uses.

Future Work with VERITAS/Gammapy:

Potential Projects and Discussion of the Development of a Reciprocal Relationship

Future Proposed Work

Dedicated VERITAS help:

- VERITAS members are willing to help with some of these changes but there is also substantial work that continues with the legacy packages that takes up most of our software development power.

Questions about future VERITAS related work:

- How to proceed with changes necessary for some specialized analyses?
  - Should we contribute recipes? Some analyses are important to gamma-ray analysis but don't work well within a map based framework (i.e. PBHs, and clustering analyses, pulsars).
- Notebook for working with VERITAS data on the tutorials page?
- Tutorials for VERITAS users from gammapy developers (the challenge here is the difference in levels of VERITAS users is very broad)?

Proposals:

- Implementation of errors on events in the construction of map datasets instead of PSF implementation. This would have major effects on *gammapy* and would require rewrites of sections where PSF is utilized. The benefit is that this incorporates information about poorly reconstructed events more accurately than using an averaged quantity like PSF.
- Implementation of more statistical methods like Vianello (2018) and look-elsewhere effect corrections.
- Using Gammapy dark matter templates for upper limits, currently requires rewriting of some of the templates to make them more general and then writing scripts for upper limits on parameters by modifying code written for parameter contours.

Open Questions

- How many features in *gammapy* should be **instrument specific** and how can these parameters be solidified for users in the future who may be working with data from VERITAS and CTA (or other current generation instrument)?
- VERITAS is currently discussing our **legacy data products** and *gammapy* users will clearly be some of the targeted users. Do developers of the software have any thoughts or opinions about our data release?
- Some validation results are of general interest to the community and so is there a forum to share certain work to decrease the overall burden?
- What parts of gammapy are widely used and “safer” and which parts are newer or more untested (like RBM vs. FoV Background).

VERITAS Analysis Contacts

These are up to dates contacts for reaching out to VERITAS members with future questions, and requests:

VERITAS Spokesperson

– Amy Furniss, UCSC, amy.furniss@gmail.com

Analysis Working Group Co-Chairs

– Brian Humensky, Goddard, thomas.b.humensky@nasa.gov

– Lucy Fortson, Minnesota, lffortson@gmail.com

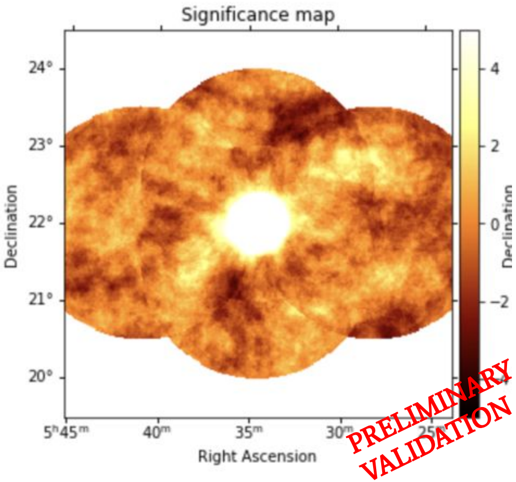
Package Leads (VEGAS/ED)

– Jodi Christiansen, Cal Poly, jlchrist@calpoly.edu

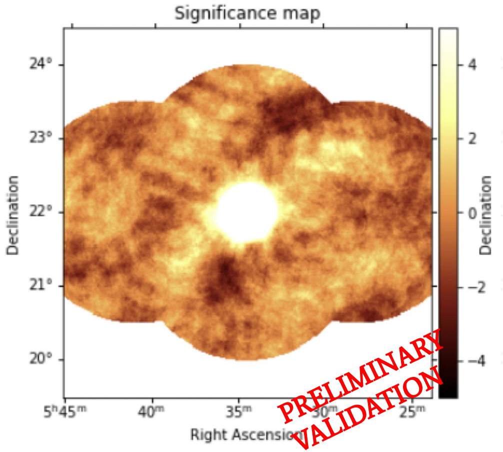
– Gernot Maier, DESY, gernot.maier@desy.de

Discussion & Questions?

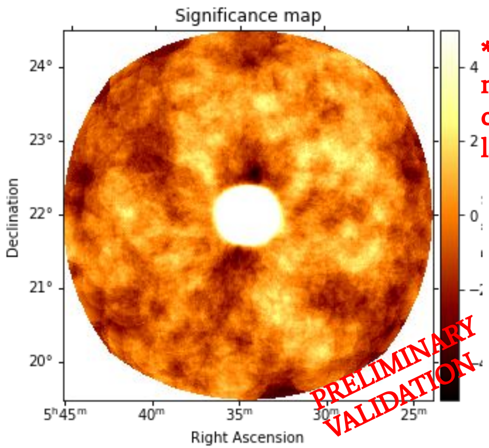
Validation – Ring background (stacking)



 # We need to extract the ring for each observation separately, hence, no stacking at this stage
config.datasets.stack = **False**



 # We need to extract the ring for each observation separately, hence, no stacking at this stage
config.datasets.stack = **True**



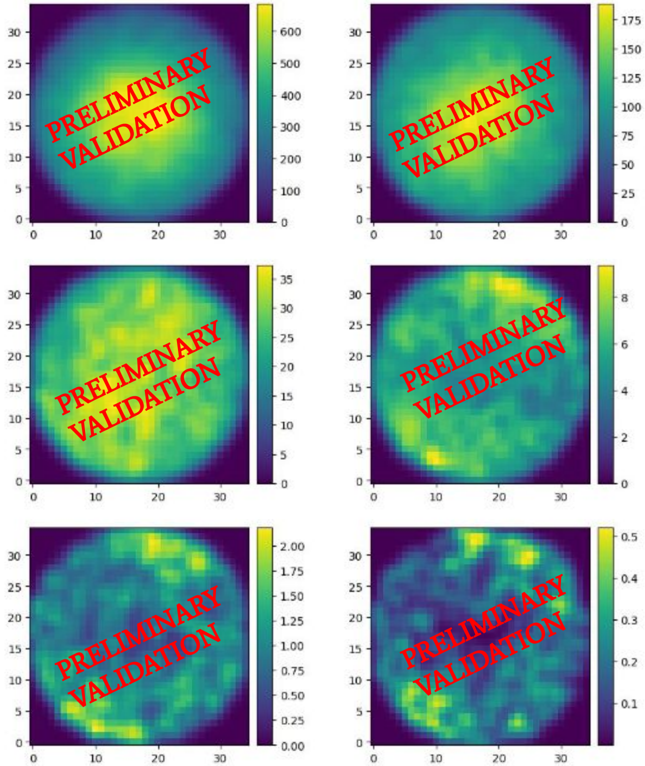
Apply offset-max in SafeMaskMaker after stacking

* different runs b
meant to show th
cutout shape and
lack of artifacts

Camera edge/offset-max artifacts show up if offset mask is applied before stacking.
This is either resolved by stacking the dataset before analysis or applying the SafeMaskMaker after stacking, but this results in a cropped sky map

Validation – Construction of Backgrounds

- Gammapy has no in built ability to construct BKGs and although a recipe exists there's not a good understanding of the different methods of constructing these BKGs from data and how sensitive results are to the number of backgrounds or the difference between an observation and the background used.
- Currently we have three methods for the generation of BKGs:
    - Use our archival data and create a list of backgrounds with weak or no gamma-ray sources and then apply a mask and stack these runs to the create 2D/3D backgrounds on a run-by-run basis based on the divergence of differences.
    - Generate a BKG look-up after binning our existing runs into basic observation parameters (i.e zenith, azimuth and noise) and then select the BKGs based on where the run falls on this axis.
    - Generate the BKG from the single run after source masking.
- We also would like to compare the difference between simulation created BKGs and data driven BKGs to estimate the systematics of using but these studies are planned for after we have fully validated the data driven backgrounds.



Validation – FoV background

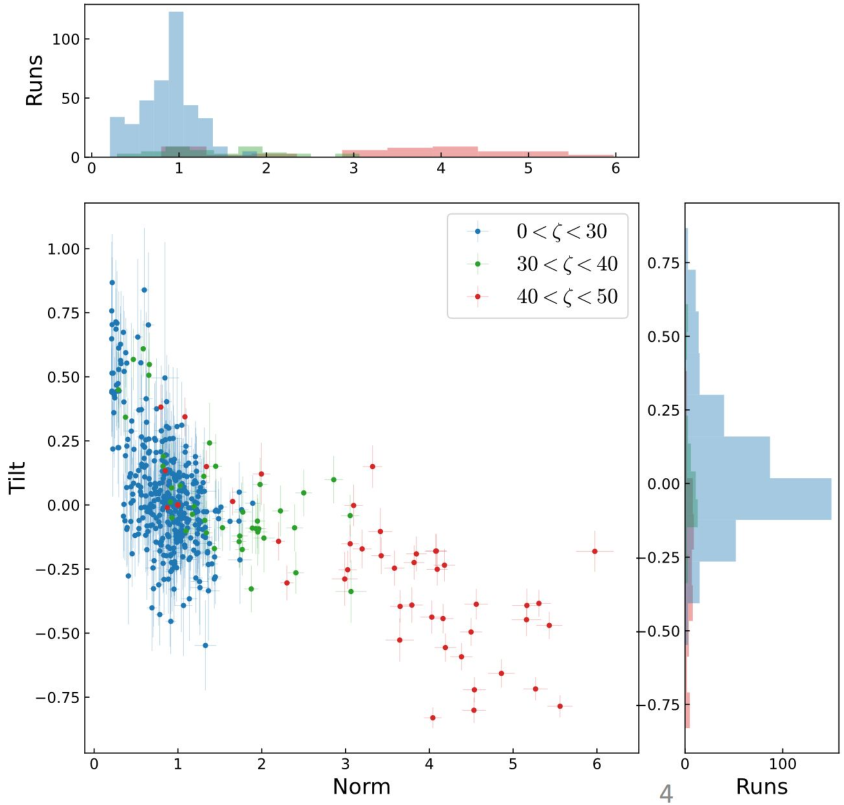
(Kleiner,2023)

FoV Background relies heavily on the validity of backgrounds.

Construction of data driven backgrounds in regions of low statistics often requires large scaling that introduces biases in spectra. We see that for runs >30 degrees zenith most of these issues are minor but become very significant for low zenith angle observations.

The plot to the left was generate using V6 Crab data all taken with good weather, across a range of zenith angles and shows the required normalization and tilt when fitting the data driven BKG using FoV Background.

Part of this is due to the low statistics of the data available to make backgrounds in regions where data we don't observe often but may additionally be caused by more complicated issues like those of the significance of atmospheric conditions becoming more important at low zenith angles.



Overview

- Introduction to VERITAS
    - Software overview with gammapy context
    - Explanation of the user base
    - Explanation of the current expected use cases
- Validation
    - What has been our process of validation and what have we been studying
- Problems
- Future Potential Uses, and QOL improvements
- VERITAS/Gammapy future discussion