FAIR Data

Findable

F1. (Meta)data are assigned a globally unique and persistent identifier
F2. Data are described with rich metadata
F3. Metadata clearly and explicitly include the identifier of the data they describe
F4. (Meta)data are registered or indexed in a searchable resource available

Interoperable

I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (Meta)data use vocabularies that follow FAIR principles
I3. (Meta)data include qualified references to other (meta)data

Accessible

A1. (Meta)data are retrievable by their identifier using a standardised communications
A1.1 The protocol is open, free, and universally implementable
A1.2 The protocol allows for an authentication and authorization procedure, where necessary
A2. Metadata are accessible, even when the data are no longer available

Re-Usable

R1. (Meta)data are richly described with a pluralityof accurate and relevant attributes
R1.1. (Meta)data are released with a clear and accessible data usage license
R1.2. (Meta)data are associated with detailed provenance
R1.3. (Meta)data meet domain-relevant community standard

http://gavo1.aip.de:22228/e/how-fair-is-your-data

Use the QR-Code

- access the website
- select your favorite wavelength
- provide your estimate of the FAIRness of the data in the field

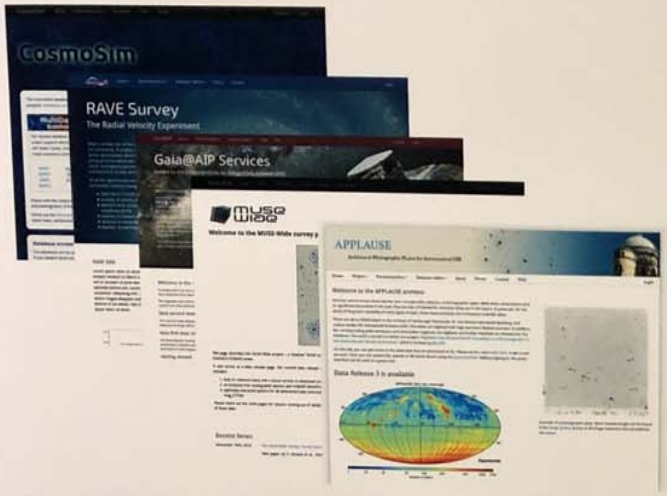
Example:

- data in the field of visible wavelength
- some pointers to arguments

Leibniz-Institut für
AIP Astrophysik Potsdam

Daiquiri - Python based framework for the publication of scientific databases

A. Galkin, J. Klar, K. Riebe, G. Matijevic, H. Enke



Data services @AIP, proudly powered by Daiquiri – an open source software for publishing scientific data, based on Python, developed at AIP

At Leibniz Institute for Astrophysics Potsdam (AIP) we host, curate and publish terabytes of astrophysical data using the Daiquiri framework. Dedicated web applications allow scientists from all around the world to run SQL queries via the web interface or scripted access and get their desired data in reasonable time. In the last two years, Daiquiri was completely re-written in Python and received major updates - upload of VOTables, VO TAP support and many more features. Daiquiri has been developed in close cooperation with scientists and having support for collaborations in mind. All components are Open Source software and available on GitHub.

The Daiquiri package enables collaborations and institutions to create customized websites. The new framework architecture splits the application into different layers, so user, job and queue management can be developed and maintained as separate packages. It is based on the Django framework in Python and utilised the Astropy Python package. This facilitates the integration of special modules for individual projects into the Daiquiri applications, such as the Cut out service in MuseWIDE. The Queryparser Python package translates the queries from ADQL to the backend SQL. The access permissions are checked depending on user accounts and groups.

Daiquiri has many features, just to name a few: an interactive query interface, asynchronous database queries, visualization tools, an IVOA compliant cone search API and a registry of registries implementation, metadata and user management, UCDs support, a cut-out API for data cubes, a contact form, DOI integration, an OAI PMI interface for harvesters, a meeting module and an integrated Wordpress for the documentation.

The role of the data curators

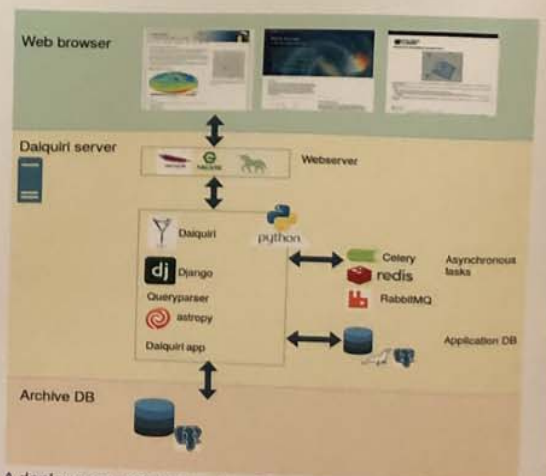
The tools we use and develop to publish the scientific datas are based on the long years of experience and knowledge of people. They curate and publish the diverse astronomical datasets, design and develop the software tools and the web applications in close collaboration with the scientists. The role of the data curators and research software engineers has become even more crucial with the implementing of the FAIR Principles for scientific data management and stewardship.

Links

You are welcome to try out Daiquiri as a provider:
Daiquiri on GitHub https://github.com/django-daiquiri/
Docker setup https://github.com/django-daiquiri/daiquiri-docker-compose

Data services @AIP

Scientific archives powered by Django Daiquiri
Gaia@AIP https://gaia.aip.de
APPLAUSE archive https://www.plate-archives.org
MuseWIDE https://www.musewide.aip.de



A deployment architecture for a Django Daiquiri installation.

A new multi-band optical image pipeline for the Magellan 6.5 m telescope.

Zohreh Ghaffari[1], Catalina Sobrino Figaredo[1], Martin Haas[1], Rolf Chini[1], Steve Willner[2]

1 *Astronomisches Institut Ruhr-Universität Bochum, Universitätsstraße 150, D-44801 Bochum, Germany*
2 *Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA.*

Abstract

We present a new image reduction pipeline for *PISCO*, the *Parallel Imager for Southern Cosmology Observations*, attached to the 6.5 m Magellan telescope at the *Las Campanas Observatory* (*LCO*), Chile. *PISCO* obtains simultaneous g, r, i, z band images with 0.2" pixel size and 5' x 8' field of view.

Our pipeline package performs all basic standard reduction steps on the raw images of each CCD and corrects for several instrumental effects. We apply astrometry, construct deep co-added images in each band and implement photometric calibration. We show the procedure of reducing *LCO* images from raw data to the final results and illustrate the quality of our data reduction by comparison with *Hubble Space Telescope* (*HST*) imaging.

Special emphasis is placed on a high-fidelity photometric calibration. This is indispensable for our research purpose to study the evolution of galaxy clusters around 3C radio sources in the early universe (z > 1). For a typical data set with 20 exposures of 120 s duration, the reduction allows for the reliable detection of faint sources down to r = 26 mag. For seven 3C fields, a cross-match with *HST* catalogues in a 2' x 2' field of view demonstrates the exceptional depth and significance of our source catalogue.

Motivation

While superb and cost-expensive telescopes and cameras have been built, an associated high fidelity image reduction pipeline for an efficient common usage is yet missing – often due to budget saving. Therefore, even within a part of a PhD thesis like here, it became necessary to develop such a pipeline. We invite *PISCO* observers to learn from our pipeline and to save multiple efforts.

Acknowledgements

We wish to thank Tony Stark, CfA, for help with the *PISCO* observations, and Marco Chiaberge, STScI, for providing us with the *HST* source catalog for comparison.

Reduction pipeline

We observed 13 high redshift 3C radio galaxies in June 2018 with *PISCO*.
One example, 3C255 at redshift z=1.355, is shown here.

Fig. 1: Multi-filter raw image mosaic for g, r, i, z. The arrangement contains 8 CCDs, two for each filter.

Fig. 2: Single r band image after bias, dark and flatfield correction. The background level shows a step between left and right CCD. It is further corrected by our pipeline.

Fig. 3: Final r band image with a 4' x 4' FoV. It is combined of 21 single exposures of 120 s. The seeing is ~0.8" measured by the mode of the FWHM of all isolated round sources.

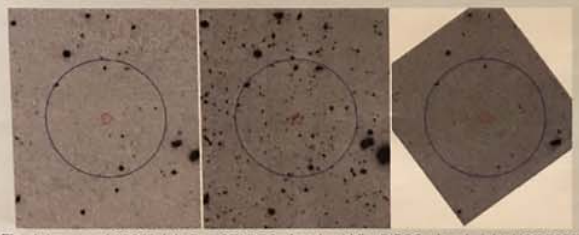


Fig. 4. Images of 3C255; left: Pan-STARSS r-band, middle: PISCO r-band, right: HST F606W. The red, green and blue circles mark a radius of 5", 30" and 55" around 3C255.

Science Achievement

Candidate galaxies at the redshift of the 3C source are selected by magnitude and color cuts:

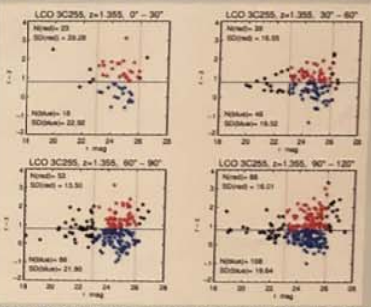


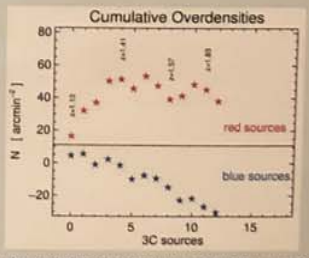
Fig. 7: Color Magnitude Diagrams (CMDs) of the 3C255 field for 4 bins on the distance from the 3C itself. Top left: < 30", top right: 30" – 60", bottom left: 60" – 90", bottom right: 90" – 120".
The candidates are selected by 23 < r < 26 (red dotted lines).

To distinguish between red and blue candidate galaxies, a color cut r – z = 0.8 is applied (blue solid line). Black symbols are excluded as fore- and background sources.

The number N of candidates and their surface density SD are given.

For the CMD selected galaxy candidates we calculate the galaxy overdensity (OD), i.e. central (< 30") surface density minus surrounding surface density in the annulus 90" - 120". To study the evolution of galaxy overdensities between redshift 1 and 2, we sort the 3Cs by redshift and calculate the cumulative overdensities (a: 1 < z < 2, 30" corresponds to ~250 kpc). This is done for blue and red candidates separately.

Fig. 8: Cumulative overdensities of CMD selected candidates. The x-axis is the index of the 3C fields sorted by redshift.



Red sources show OD between redshift 1 and 1.4: for the cosmological evolution this means that red ODs are not found in the earlier universe at z > 1.5 and start to show up at the epoch of z ~ 1.4.

Blue galaxies show on average a negative OD (i.e. underdensity) for all redshifts from z = 2 to z = 1.
This means that blue star forming galaxies are ubiquitously found in the outskirts.

The deep *LCO* data corroborate the results on the evolution of red galaxy overdensities around high redshift 3C sources found by *Spitzer* + *PanSTARRS* (Ghaffari et al. 2017). The *LCO* images are also wide enough to clearly reveal the ubiquitous central lack of blue galaxies.

PISCO Data quality

Fig. 5: log(N) - mag histogram.
The detection limit is taken as the magnitude at which 98% of the sources are brighter.

The detection limit of our *PISCO* data (r = 26.6 mag, red dotted line) is about 4-5 mag fainter than that of Pan-STARRS.

The blue dashed line shows a linear fit to the log(N) - mag histogram. The completeness limit (blue solid line) and the completeness fraction (> 90%) are estimated relative to the extrapolation of the linear fit.

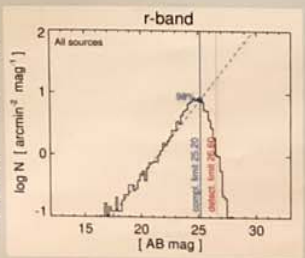
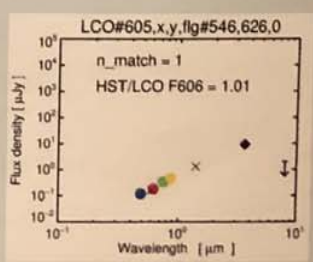


Fig. 6: Photometric comparison with *HST*.

Cross-matched Spectral Energy Distribution (SED) of a faint (r = 26 mag) and red galaxy in the field of 3C255. Colored circles denote *PISCO* griz bands, black crosses show *HST* F606W and F140W from Kotyla et al. (2016), and the black diamond / upper limit give *Spitzer/IRAC* photometry (Ghaffari et al. 2017).

The source match is unique within 0.5" and the flux ratio *HST* / *LCO*@F606W is close to 1, on average 1.003 +/- 0.06 for the entire sample, demonstrating the high quality of our *LCO* photometry.



References

[1] Brian Stalder et al., "PISCO: the Parallel Imager for Southern Cosmology Observations", Proc. SPIE 9147, Ground-based and Airborne Instrumentation for Astronomy V, 91473Y (2014)
[2] Z. Ghaffari et al., "Galaxy overdensities around 3C radio galaxies and quasars at 1<z<2.5 revealed by Spitzer 3.6/4.5µm and Pan-STARRS", Astronomische Nachrichten, 338, pp. 823-840 (2017)
[3] J.P. Kotyla et al., "The environment of z>1 3CR Radio Galaxies and QSOs: From Proto-Clusters to Clusters of Galaxies", ApJ, 826, pp. 46-57 (2016)

Outflow Detection in G327 : A 3D Approach

N. Kandpal[1], A. Sanchez Monge[1], Peter Schilke[1]
[1] First Institute Physics, Cologne, Germany

2 Dimensional Analysis

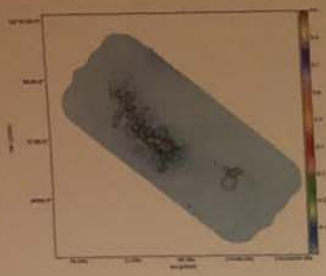


Figure 1: Red and blue shifted contour in SiO overlayed over ALMA+ATLASGAL continuum. Green lines show direction of the outflow and in yellow are the continuum sources using Sextractor.

- We used 2D contour analysis to detect the outflows in G327.
- In Fig1 we can see blue and red shifted outflow distribution.
- In all around 20 outflows were found which we can see in green in Fig 2 showing their direction.
- Method was not efficient specially in most crowded regions to assign and detect outflows.
- We need alternative methods to detect outflows.

3 Dimensional Analysis

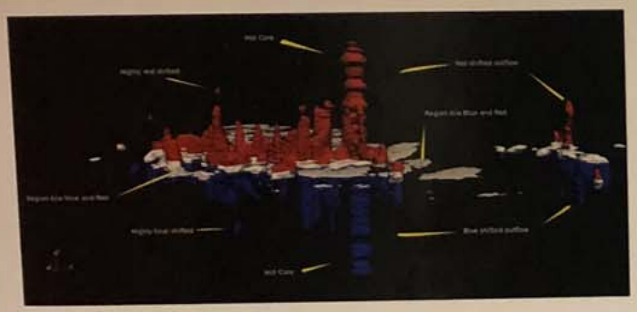


Figure 2: 3 Dimensional representation of SiO outflow. Red and blue colors represent red and blue shifted outflow. Pancake like structure in the middle is hot core and the region between red and blue shifted outflow is around systemic velocity.

- We needed to disentangle the outflows in order to get a complete picture of all the outflows.
- A 3D conversion of ALMA data was one of the solutions we found.
- We could detect around 43 outflows which is twice the number which we got using 2D analysis.
- We divided the G327 full region into subregions for better analysis.
- The outflow direction and relation can be concluded using skeleton structure of the subregions.

Figure 3: Subregion of G327 on the right showing skeletal structure of outflow.

Outflow Detection in 3D : Method

- We divided the G327 into subregions. We can see on such a subregion in Fig4.
- We found that individual outflows in 3D had complex geometry with branched outflows and multiple outflows.
- In order to calculate the outflow parameters we developed method using rotating ellipsoids.
- The ellipsoid was rotated using 2 angles in the direction of outflow and ellipsoid parameters were adjusted to fit the outflow lobe.
- The flux was calculated by using ellipsoid as mask which gave us outflow parameters.

Outflow Detection in 3D : Method

Figure 4: Section of G327 with ellipsoid in red being rotated to the position of the outflow in both the figures.

Results : Using 3D method

Figure 5: Kernel Density Estimation(KDE) for all the individual outflow lobes. Rugs in red are individual outflow lobes.

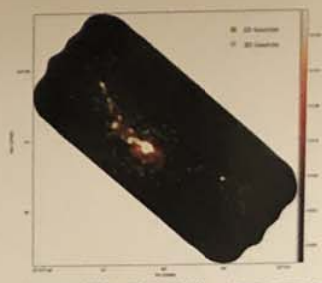


Figure 6: G327 ALMA+ATLASGAL continuum image with 2D sources(SExtractor) and possible 3D sources.

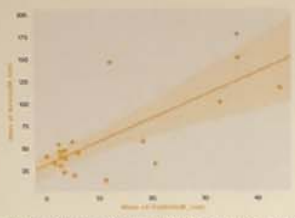


Figure 7: Plot of source mass from Sextractor vs Outflow mass form 3D method with linear fit.

- Kernel Density Estimation in Fig 5 shows masses of individual outflows in the range '0.4-17.7' solar mass.
- It was found in Fig6 that 2D Sextractor sources do not always overlap with possible 3D sources implies there could be more sources driving the outflow.
- A plot of source mass to the outflow mass(Fig 7) shows outflow mass not very high which could imply that our sources are in early stages of stellar evolution.

Position Angle : Monte Carlo Simulation

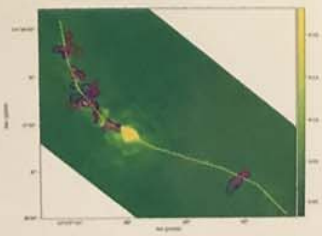


Figure 8: Outflow direction represented by pink line on top of blue and red shifted lobes close to filament in yellow.

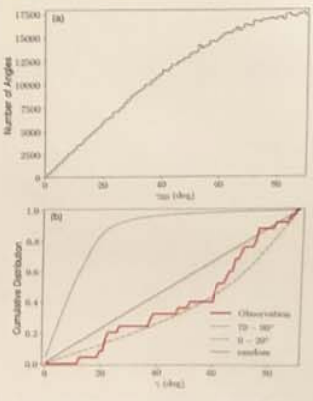


Figure 9: (a) CDF of the simulated gamma3D (b) CDF of gamma2D based on randomly generated gamma3D.

- The 3D lobes selected from rotated ellipsoid are projected back in 2D.
- Using Monte Carlo simulation form (Kong et. el. 2019) we found that outflows in G327 are preferentially oriented orthogonal to filament.
- This can suggest that angular momentum of protostellar accretion disk is correlated with host filament.

■ Link for 3D model and Virtual Reality : https://sketchfab.com/3d-models/2g327-sio-colored2-b510a2f11b1447fa8908afb01f79ef19

Searching Pulsars Using Neural Networks

Lars Künkel[1], Joris P. W. Verbiest[1,2] & Rajat M. Thomas[3]

[1]Universität Bielefeld, [2]MPIfR Bonn,
[3]Department of Psychiatry, Amsterdam UMC, University of Amsterdam

lars.kuenkel@uni-bielefeld.de

Supported by:

UNIVERSITÄT BIELEFELD
Faculty of Physics

Bundesministerium für Bildung und Forschung

Introduction

Pulsars are rotating neutron stars which can be observed by faint periodic pulsations. While pulsars can be used for the study of a multitude of astrophysical phenomena, finding new pulsars is not a trivial task due to their faintness, the modulation of the pulse period by binary companions and the vast parameter space of possible dispersion measures (DM) and pulse periods. The DM of a pulsar, which introduces a dispersive delay to the pulses, is not inherently known, which forces current approaches to blindly dedisperse the data at a range of different DM values. Analysing these DM trials results in a huge amount of pulsar candidates which are mostly the result of radio frequency interference (RFI) or other forms of noise. These candidates have to be classified subsequently. Quickly classifying pulsar observations becomes more and more important since new pulsar surveys will not be able to store all observations[1]. We propose a pipeline based on neural networks that is able to directly classify survey observations with great confidence and can be trained in an end-to-end manner.

Main Points

1. A convolutional neural net is able to directly detect pulsars in survey observations.
2. The neural network is sensitive to a range of different DM values (currently 80-700). The dedispersing part of network has two output channels which contain the low and the high DM pulsars respectively.
3. The neural network is trained on unaccelerated pulsars with periods between 20 ms and 650 ms.
4. The classification is based on result of the fast Fourier transform (FFT), short-time Fourier transform (STFT) and fast folding algorithm (FFA)[2].
5. The output of the dedispersing part of the network, the intermediate time series, can be investigated using conventional techniques.
6. The network is trained using a combination of real survey noise and simulated pulsars.
7. The simulated pulses are scaled down during training by a factor $1/N$ which decreases during training.
8. Current input size: $14 \times 400\,000$.

Architecture

The dedispersing part of the network is based on 1D convolutions. To reduce the data volume strided convolutions are used in the first layer of the nerual network. In subsequent layers dilated convolutions provide a sufficient receptive field.

The classifying part of the network combines the predictions of several individual classifier which are based on the FFT, STFT or the FFA. The gradients can propagate freely through the FFT and STFT which allows end-to-end training.

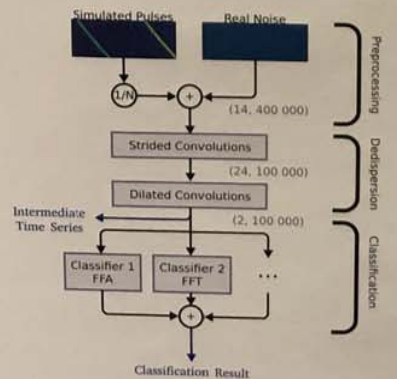


Figure 1: Sketch of the neural network based classification pipeline.

Results

The performance of our pipeline is tested on a set of 103 survey observations which include 23 observations containing known pulsars. We can test how strong the known pulsars are in the intermediate time series and how well the classification performs.

When comparing the strength of the pulsar in the intermediate time series with the strength of the pulsar in the time series that was dedispersed at the correct pulsar DM, wee see that our model provides comparable or better results even though the neural network does not know the pulsar DM.

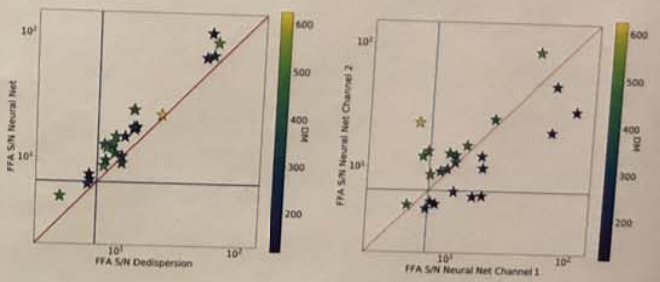


Figure 2: Left: Comparison of the S/N of real pulsar observations in the dedispersed time series and the output of the dedispersing part of the neural network. Right: Same Comparison between the two output channels of the dedispersing part of the neural network.

The neural network is able to find the majority of the real pulsars in the test set without false positives. The best classifier of the individual classifiers is the classifier based on the FFA. Only the very faint pulsars elude the network. Our classifiers only take into account the most significant signal in the FFA/FFT/STFT. In the case of faint pulsars signals resulting from noise may reach a similar strength as the pulsar signal. These pulsars could still be detected using traditional search techniques in the intermediate time series, since these techniques create multiple pulsar candidates per observation.

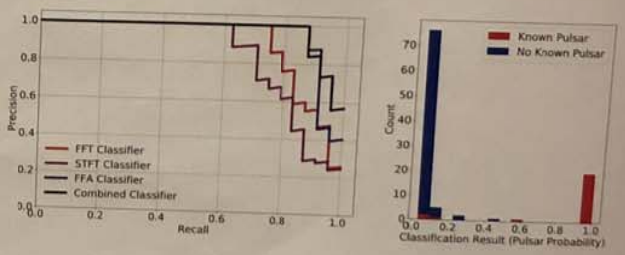


Figure 3: Left: Precision-recall curve of the neural net model on the test set containing real pulsars. Right: Histogram of the classification result for all observations in the test set.

Conclusions

- Our neural network model is able to dedisperse pulses with a wide range of possible DM values without knowing the DM.
- Using neural nets to dedisperse filterbank data allows us to reduce the number of time series which have to be classified subsequently.
- Very faint pulsars which elude our classifiers can still be detected in the intermediate time series with traditional search techniques.
- Our classifiers are able to detect the majority of our test pulsars without false positives.
- Training the classifiers provides useful gradients for the training of the dedispersing part of the network.

References

[1] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles. Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach. *MNRAS*, 459:1104, Jun 2016.

[2] A. D. Cameron, E. D. Barr, D. J. Champion, M. Kramer, and W. W. Zhu. An investigation of pulsar searching techniques with the fast folding algorithm. *MNRAS*, 468(2):1994–2010, Jun 2017.

Acknowledgements

This work was done as part of the D-MeerKAT consortium. The Quadro P6000 used to develop the network architecture was donated by the NVIDIA Corporation. The networks were trained using the GPU cluster of the Bielefeld Lattice QCD group.

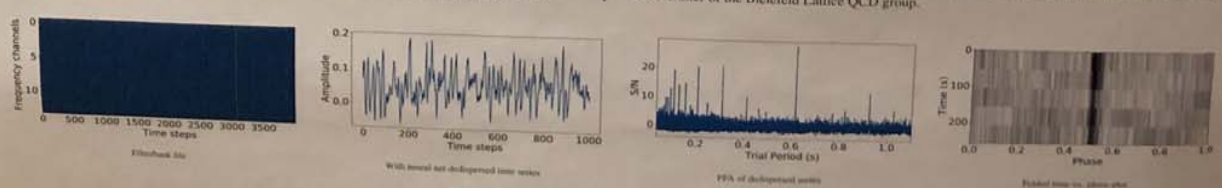


Figure 4: Different Representations of pulsar survey observations.

UNIVERSITÄT BIELEFELD

ASTRON
Netherlands Institute for Radio Astronomy

Metadata and User-Provided Data in the LOFAR Long Term Archive

Jörn Künsemöller[1], H.A. Holties[2] and G.A. Renting[2]

[1]Bielefeld University, Bielefeld, Germany; [2]ASTRON, Dwingeloo, Netherlands

Data challenge LOFAR

Across Europe, 52 stations of the International LOFAR Telescope (ILT) produce a continuous data stream of up to 200 Gbps in total. This data is transported via a fibre-optics network to central facilities in Groningen in the Netherlands. For some stations, this means that data travels more than 1000 kilometers. All incoming data is correlated in real time and the result (up to 14 GB/sec) is written to a large compute cluster where it can be further pre-processed.

The telescope consists of 38 Stations in the Netherlands, 6 in Germany, 3 in Poland, 1 in France, Ireland, Sweden, the UK, and Latvia. In Italy, a further station will be built.

Each LOFAR station has two antenna types for different frequency bands between 10 and 240 Mhz.

To archive and serve its large and growing dataset, the ILT operates the LOFAR LTA (long-term archive). In three physical locations (SURFsara in the Netherlands; FZ Jülich in Germany, and PSNC in Poland), the LTA currently stores about 49 Petabyte in 10 million dataproducts. The archive has an annual growth rate of roughly 7 PB.

Data growth in the LTA storage sites over time.

Metadata in the LOFAR LTA (Long Term Archive)

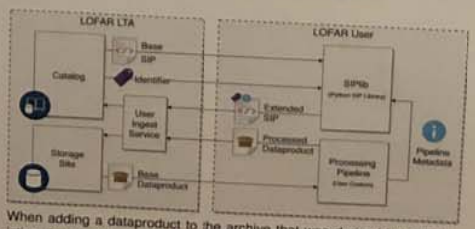
While the data itself is distributed on the three LTA sites (see left), the metadata for discovery and data access is kept in a separate catalog database and describes each dataproduct and its full provenance in detail. Since LOFAR is not static and new functionality is added over time, valid properties of metadata and the LTA datamodel are a moving target. The LTA follows principles of the Open Archival Information System (OAIS) model to deal with this. When adding data to the archive, the ILT control software provides a Submission Information Package (SIP) in XML format. It contains information about the original measurements and every processing step that has been applied to create each particular dataproduct. To ensure that metadata is correct and valid, each SIP gets validated against a rather strict and versioned schema before the data is stored in one of the storage sites and the metadata is added to the catalog database. System changes are reflected in both an updated version of the SIP schema as well as in the database itself. New data has to be ingested based on the most recent SIP version, and existing catalog entries are updated to contain all currently represented parameters.

Left: A small section of a SIP document in XML format. It contains detailed information about a dataproduct and its provenance. Right: An XSD schema defines valid elements and properties according to the underlying datamodel and SIPs must validate against this schema before data is accepted for the archive.

Ingesting User-derived Dataproducts

User-provided derived dataproducts pose a challenge concerning the consistency and cross-linking of dataproducts in the LTA. When a user processes data outside of ILT control and wants to add relevant dataproducts to the archive, the user has to provide a valid SIP for it. It not only has to completely describe the entire genesis of the new dataproduct, but has to correctly refer to any existing dataproducts in the LTA that it was derived from.

We provide services and Python modules (currently in a pilot user stage) to request information and LTA identifiers on the base data, which is already known to the LTA, in form of a latest-version SIP. The users can then extend that by programmatically adding processing steps they applied to create the derived dataproduct. Unique LTA identifiers can be linked to custom user-specific labels for reference by the providing user. We further provide tools to validate and visualize the outcome on the use-end to mitigate bouncing ingest requests due to incorrectly specified SIPs or nonsensical information due to human error. Each catalog entry is further associated with an identifier for the providing user to allow filtering in data discovery and for potential rollbacks.



When adding a dataproduct to the archive that was derived from an existing dataproduct, a SIP document for the original dataproduct can be requested from the archive. It can then be used as provenance information to the SIP describing the new dataproduct, so the user only has to provide additional information about the applied new processing steps.

A small code example to demonstrate the process of adding base data information to a new SIP for the derived dataproduct.

ASTRON - The Netherlands Institute for Radio Astronomy

ASTRON

The Netherlands Institute for Radio Astronomy (ASTRON) designed LOFAR, is the major partner of the ILT, and owns the 38 Dutch stations. ASTRON and operates LOFAR for interferometric observations. → www.astron.nl

GLOW - The German Long-Wavelength Consortium

The German Long Wavelength Consortium

The German Long Wavelength Consortium (GLOW) was formed in 2006 by German universities and research institutes. The six German ILT stations are owned and operated by GLOW partners. → www.glowconsortium.de

Acknowledgements

We acknowledge contributions and support by various members of the Radio Observatory at ASTRON, particularly from Marcel Iacobelli, Jorrit Schaap, and Raimund Boekhorst. We further acknowledge support and operation of computing and storage facilities by the FZ-Jülich and Bielefeld University and financial support from BMBF D-LOFAR III (grant 05A14PBA) and D-LOFAR IV (grant 05A17PC1).

References

[1] G. A. Renting, H. A. Holties: LOFAR Long Term Archive, Proc. ADASS XX, ASP Conf. Ser. 442, 49 (2011)

[2] H. Holties, A. Renting, Y. Grange: The LOFAR long-term archive: a wholesome set petabyte scale, Proc. SPIE 8451, 451-458 (2012)

PyParadise:
A simultaneous pipeline of stellar and gas kinematics

Man I Lam[1], Bernd Husemann[2], Omar S Choudhury[1], Anika Beer[1] and C. Jakob Walcher[1]

mlam@aip.de

[1]Leibniz-Institut für Astrophysik Potsdam (AIP), An der Sternwarte 16, 14482 Potsdam, Germany
[2] European Southern Observatory, Karl-Schwarzschild-Str. 2, 85748 Garching b. München, Germany

Summary

PyParadise is a state-of-the-art stellar population synthesis code, which is based on the MCMC. It derives stellar population, stellar kinematics, and gas kinematics at the same time. In this poster, we present the fitting results based on mock and observed data. We also compare our result to some existed methods, and conclude that our method is stable in velocity measurement up to z~1.

Modules

Non-linear (MCMC) fit of kinematics
v, sigma, (h3, h4)

Linear (NNLS) inversion for stellar populations
<age>, <Fe/H>, etc

None-linear (MCMC) fit of emission lines
F, v, sigma

Procedures

Original spectra
↓
Kinematics fit (Library 1)
↓
Stellar populations fit (Library 1)
↓
...
↓
Kinematics fit (Library N)
↓
Stellar populations fit (Library N)
↓
Emission lines fit

Error Measurements (Bootstrap)

N times → Random spectrum

Kinematics | Stellar populations | Emissions

Means, Errors

Spectral Fitting Example

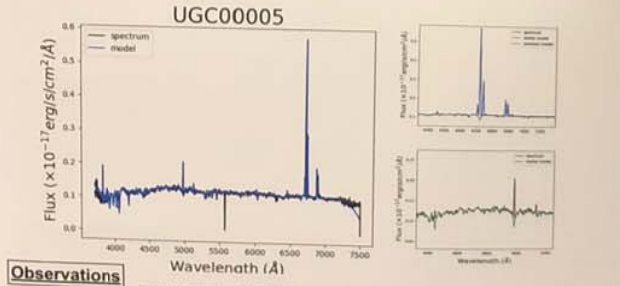


Figure 1: Examples of UGC00005 spectra from CALIFA survey.
Left: an example of whole spectrum fitting. *Right:* stellar population model and emission lines model.

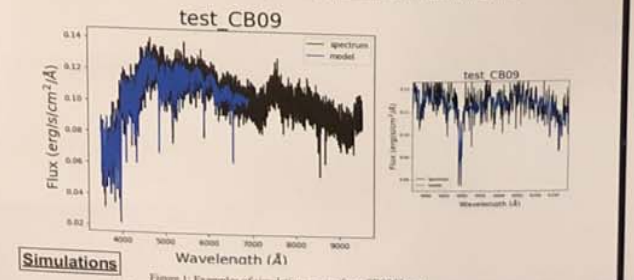


Figure 1: Examples of simulation spectra from CB09 library.
Left: an example of whole spectrum fitting. *Right:* stellar population model.

UnitTest Results

Table 1: Configuration file of CB09 spectra

Nr	Velocity Guess	Template	Sample	Burns	Wavelength
1	(-20, 20)	CB09	500	150	3500 - 6900
2	(-200, 200)	CB09	300	100	3500 - 6900
3	(-200, 200)	MIUSCAT	300	100	3500 - 6900
4	(-200, 200)	RG04	300	100	3500 - 6900
5	(-200, 200)	MIUSCAT	500	150	3500 - 6900
6	(-200, 200)	RG04	500	150	3500 - 6900
7	(-200, 200)	CB09	500	150	3500 - 6900

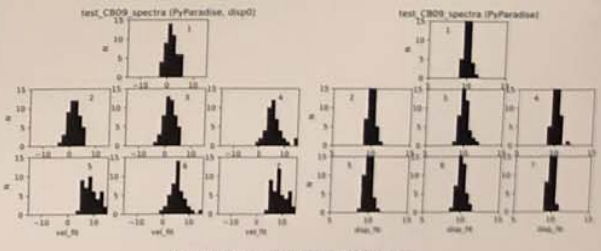


Figure 3: Retrieved test results from CB09 mock data.
Initial setup file is shown in Table 1. The kinematics recovery of PyParadise is accurate but a systematics offset is found in different libraries.

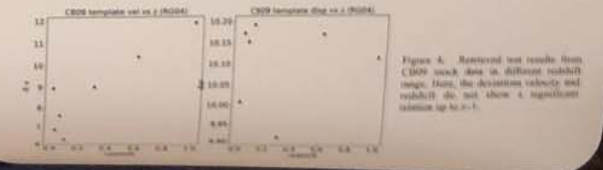


Figure 4: Retrieved test results from CB09 mock data in different redshift range. Here, the deconvolve velocity and redshift do not show a significant variation up to z~1.

References

Walcher, C. J., Coelho, P. R. T., Gallazzi, A., et al. 2011, A&A, 582, A46
Husemann, B., Bennert, V. N., Scharwächter, J., Woo, J. H., & Choudhury, O. S. 2018, MNRAS, 475, 1065

Exoplanet detection using Machine Learning

A. Malik[1]

[1]Universitäts-Sternwarte, Ludwig-Maximilians-Universität, München, Germany

INTRODUCTION

We present a machine learning based technique to detect exoplanets using the transit method. In this approach, time series features are extracted from light curves and a tree-based classifier using a popular machine learning tool 'XGBoost'. This model was able to correctly predict whether a transit is present in a light curve with an accuracy of around 85%.

Machine learning and deep learning techniques have proven to be very useful in various areas of scientific research. We would like to exploit some of these methods to improve the conventional algorithm based approach used in astrophysics today to detect exoplanets.

METHOD

For this analysis, we used 6000 light curves from the K2 mission, out of which 3000 light curves contained a randomly injected transit. From these 6000 samples, we used 4500 samples to train our model and remaining 1500 samples to evaluate its performance. We made sure ratio of light curves with transit vs without them remains 1:1 in both test and training set. The data was prepared in the following steps:

- We removed all known sources from each light curve.
- Removed 3σ outliers (discrepant points like cosmic ray hits) and flattened the light curves
- After that, we randomly injected transits in half the cases.

We then performed a few additional steps to prepare the light curves to be used as inputs to our model:

- We used popular time-series analysis library 'TSFresh' to extract features.
- For each light curve, we extracted 789 features. Mathematical formulation and other details of these features can be found at: https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html
- Later features with constant values were dropped and undefined numbers (NaNs) were interpolated. Finally, we were left with 706 features.

These features capture information about the characteristics of a light curve and used as input to our model.

RESULTS

We processed our validation set of 1500 samples the same way as described in previous section before using them to make inference from our trained model. Our validation set consisted of 752 cases with an injected random transit and 748 cases without it. The results on our validation set are:

- The model had a prediction accuracy of around 85 % i.e. it was able to classify planet vs non-planet signal correctly in 85% of the cases.
- It was able to identify a transit light curves with a precision of 0.78.
- The model predicted only 72 false positives.

The model also predicted 162 false negatives, where it missed the transit signal but it is important to note that transits were injected randomly which resulted in many non-detectable or special cases:

- Cases where injected transit signal was weaker than noise -> low S/N ratio
- Cases with very low inclination angle for the given star -> Hard to detect.

These cases will be rectified in the next version of training data. Above results are summarised in Figure 1.

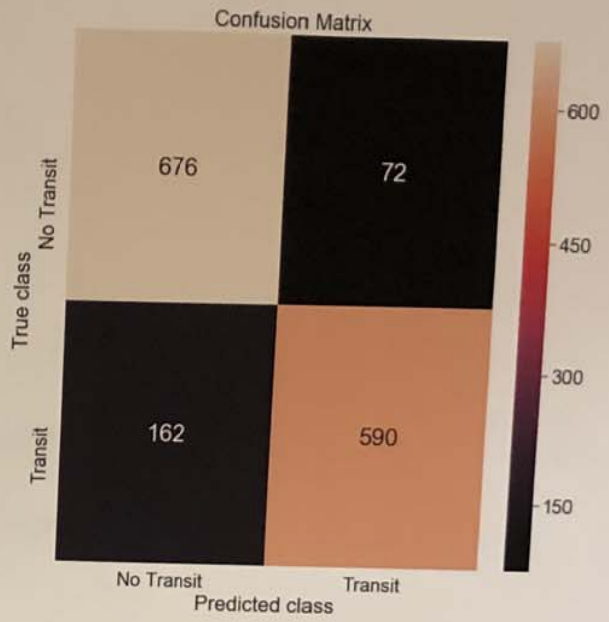


Figure 1: Confusion matrix, describing performance of the machine learning model

CONCLUSIONS

Machine Learning methods have proven to be very useful various areas of science and technology where we need to deal with large datasets. In some cases, it even surpasses human performance in certain tasks such as classifying various images into different classes.

With the new data coming in with a rate higher than ever before, we need systems that can extract results while making judicial use of computing power. Conventional methods like BLS(box least squares) are not efficient enough to deal with it.

The suggested method takes less than a second to classify 1500 samples. These methods are not only more efficient but more robust as well, this approach can be easily extended to classify single vs multi-planet transit signals without removing any previously detected transit signals. Hopefully, this analysis encourage the reader to explore the capacity of machine learning and deep learning methods to support the ongoing scientific research in their respective areas.

REFERENCES

- Shallue, C. J., Vanderburg, A. 2017, ArXiv e-prints, arXiv:1712.05044v1
- Ansdell, M., Ioannou, Y. et al. 2018, The Astrophysical Journal Letters, 869, 1
- Wang, X., Smith-Miles, K., & Hyndman, R.J. 2005, *Data Mining and Knowledge Discovery, 13*, 335-364.
- Maximilian Christ M., Kempa-Liehr A. W., & Feindt M. 2017, ArXiv e-prints, arXiv:1610.07717v3

CONTACT

Abhishek Malik, MSc. Astrophysics (Final Semester), USM, LMU München
Email: a.malik@campus.lmu.de, a.malik@usm.lmu.de
Phone: +49 15739761956
GitHub: https://github.com/amalik2205

Modern software can be adapted to **create**, **preserve**, and **share** the full **workflow** from data to publication!

Entering NeuLAND: Analysis workflow preservation for a *fair* FAIR

CHALLENGES

- Data quantity growth
- Analysis complexity growth
- Team size growth
- Implement *fair* data principles (findable, accessible, interoperable, reusable)

Can you simply share your analysis scripts and expect them to work, be comprehensible, and useful?
(Probably not)

EXAMPLE

- NeuLAND (New Large Area Neutron Detector) for R³B (Reactions with Relativistic Radioactive Beams)
  - 2.5 m x 2.5 m x 3.0 m active plastic scintillator
  - 6000 channels
- R3BRoot framework steered with ROOT macros
- Machine Learning (ML) approaches to reconstruct events (interaction points)

How to ...
... integrate ML solutions?
... preserve the workflow?
... convey trust in results?

IDEAS

- Code development
  - Version control (git)
  - Forking & Sharing
  - Continuous Integration (CI)
    → **Continuous Analysis**
- Cloud infrastructure
  - AWS, Azure, GCP exist
    → Pilfer ideas & training
  - Google Colab:
    → **JupyterLab** frontend + Cloud computing backend
- Data Science
  - Many solutions exists
  - Python *everywhere*
  - ROOT modules work in Python automatically
    → **Steer** compiled software

Foundations available, adopt existing software:
Analysis-as-a-Service

Image: GSI Helmholtzzentrum für Schwerionenforschung GmbH, D. Fehrenz

University of Cologne

Jan Mayer, Andreas Zilges
University of Cologne, Institute for Nuclear Physics
jan.mayer@ikp.uni-koeln.de

Data Infrastructure at the University of Cologne's Institute for Nuclear Physics

M. Müller, J. Mayer and A. Zilges

E-Mail: mmueller@ikp.uni-koeln.de

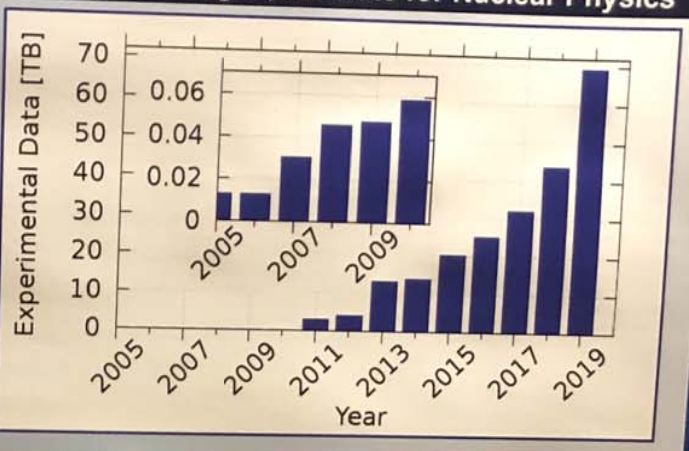
University of Cologne, Institute for Nuclear Physics

Introduction

The universitiy of Cologne's Institute for Nuclear Physics oversees its own computing and data storage resources, operated and maintained by master's and Ph.D. students.

This system has been established over the past 40 years with the current storage system having been installed 15 years ago. During this time continuous development and investments have led up to the current sophisticated infrastructure.

However, developments in data acquisition and detector physics as well as an increasing amount of collaborations have led to ever larger amounts of data to be transferred, stored, and processed.



Infrastructure

Computing

- **Kronos and Thanathos:**
  - 56 core user servers for resource intensive calculations
- **Helios**:
  - 16 core user server mostly used as access point
- **Poseidon and Gaia:**
  - 32 core user servers for data analysis
- **Hermes 1 and 2:**
  - Hosts for virtual machines providing services like:
    - DHCP, DNS, DB and identity management
    - Webpage and e-mail
    - Experimental logbooks, shift planning, etc.
- **Athene 1 and 2:**
  - Storage managers (28 cores)
  - Distribute data to the other servers via infiniband (48 Gbit/s)
- **Ares 1 and 2:**
  - Virtual machine hosts for machines used in data analysis
- **Hades**:
  - Backup for essential services

Storage

- 2 fibre channel switches (16 Gbits/s)
- 5 storage controllers
- 4 HPE MSA 2040 (1 not operational yet)
  - Support up to 7 additional disk enclosures each
    - Each supports 12 3.5" or 24 2.5" SAS disks
    - Disks arranged in RAID5 (+1 hot-spare) volumes
  - Enclosures communicate via 4 fibre channel ports
  - Total storage capacity of about 560 TB (4th MSA not included)
  - Controllers can be expanded by 17 additional disk enclosures
  - 4th storage controller will be usable soon and will provide even quicker access through solid-state-disks
- 1 DELL PowerVault 3600 for archiving
- Storage managed by two servers running IBM's **G**eneral **P**arallel **F**ile **S**ystem (**GPFS**)
- Uninterrupted power supply capable of powering the system for about half an hour

→ High parallel performance

Assessment of the current state

Pros:

- Hardware allows for easy expansion during the next few years
- Members of the institute have unlimited access to all resources
- Very fast parallel storage access and large computing power
- Independence
- Cheapest short term solution since the hardware is already there

Cons:

- Maintenance requires a lot of time
- Supporting infrastructure, like the server room, not designed as such
- Potential for expansion limited
- Constant investments in new hardware required
- Adherence to fair-principles difficult

Conclusion

As long as the available hardware can be easily expanded, there is no need to switch to a Cloud solution and comparatively slow internet connections would probably lead to a decrease in overall performance. However, once the current hardware's potential for expansion is exhausted, large investments will be required and the effort for maintaining the additional hardware will increase drastically. Therefore, Cloud solutions will at some point in time become more viable then the localized approach.

Ghost from the past
Planetary engulfment as a possible explanation for observed high stellar rotation in metal poor main sequence stars

A.Oetjens and M. Bergemann and L. Carone
Max-Planck-Institute for Astronomy, Heidelberg
Ruprecht-Karls Universität, Heidelberg

Abstract

The analysis of stellar rotation, is a standard astronomical method to determine the ages of stars. Since the fundamental Skumanich (1972) relationship that predicts that main sequence stars spin down as $t^{-1/2}$, this canonical method has been widely used to provide a diagnostic of stellar ages. However, recent advances in age tagging of stars by asteroseismology have revealed severe discrepancies with ages derived by gyrochronology (e.g. Nielsen et al. (2015)). This work aims to provide an alternative scenario: The engulfment of a massive planetary companion during the main sequence. The model relies on numerical models for angular momentum evolution (Bouvier et al. (1997)), tidal friction (Privitera et al. (2016)), and stellar spin-up (Carone (2012)). The model is applied to an ensemble of synthetic star-planet configurations. It is found that the dynamical evolution of the star-planet system leads to a gradual spin-up of the main sequence star but the time it takes for a planet to be engulfed by the star critically depends on the initial orbit, mass and metallicity of the system. In stark contrast with gyrochronology models, about 10 % of metal-poor old main sequence stars observed by the Kepler space mission exhibit very high rotation rates (Huber et al. (2014), McQuillan et al. (2014)). This contradiction is explained naturally by the model of planetary engulfment presented in this work.

Introduction

The aim of this work is to analyze, whether tidal interaction can lead to a stellar spin-up during the main sequence.
A Kepler sample of fast rotating stars is shown below. About 10 % of the metal-poor stars are high rotators which would infer young ages when using gyrochronology. However, they are extreme metal-poor, and we know from the age-metallicity relationship that they are older than 8 Gyr (Bensby (2014)).

Fig 1.: Stars from the Kepler mission

Methods

Stellar-spin:

The evolution of the angular velocity for a single, low-mass star is computed with (Bouvier et al. (1997)).
Shown are different initial rotation periods 4, 8 and 16 days . During the PMS the star spins up due to radius contraction and a decreased moment of inertia. After the contraction stops, the magnetized wind dominates and the star spins down.

Conservation of angular momentum:

In the case of close-in extrasolar planets, the central star rotates typically with a period of 10–30 days, while the planet revolves at a period of 2–8 days for orbital radii less than 0.1 AU. The tidal torque acting on the star spins up the star which consequently leads to a reduction of the planetary orbit due to the conservation of angular momentum. (Carone et al. (2007))

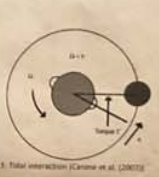
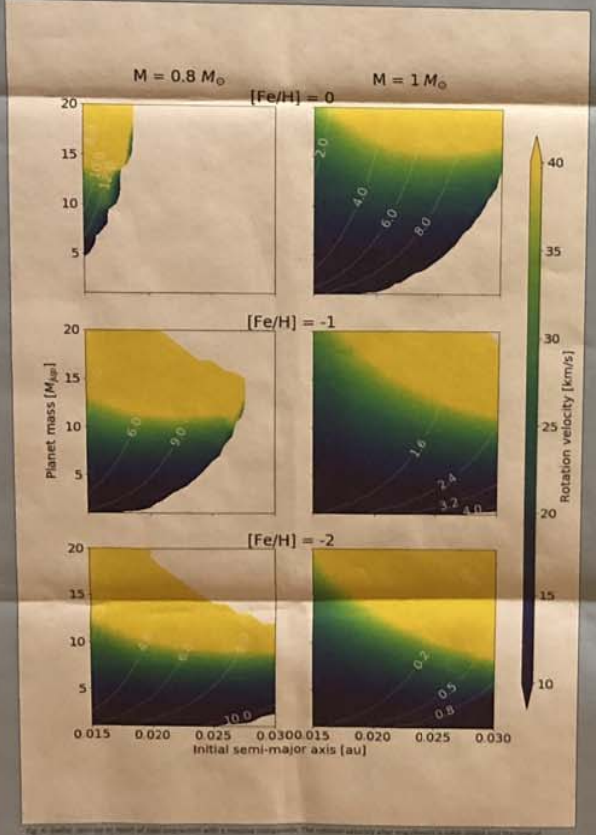


Fig 3.: Tidal interaction (Carone et al. (2007))

The impact of the decreasing semi-major axis,also effects the rotation velocity of the star.
We present a model that combines the change of rotation velocity caused by magnetized winds with the effect of the tidal interaction.

$$\frac{d\Omega}{dt} = \frac{d\Omega}{dt}_{wind} + \frac{d\Omega}{dt}_{tidal}$$



Results

Fig. 4 shows the mass of the planetary companion as a function of the initial semi-major axis . We compare a 0.8 and 1 solar mass star at metallicities from $-2 < [Fe/H] < 0$.
The simulation start at the beginning of the main sequence and stop when the planet gets engulfed. When no engulfment occurs the computation ends at the main sequence turnoff.
The metallicity of the star plays a key role in whether a planet gets engulfed during the main sequence or not. For a given mass, stars with lower metallicities are larger, have smaller convective envelopes, and they also evolve faster on the main sequence. The computations show that this leads to a quicker engulfment and a higher stellar spin-up.

Discussion

Here, the Kepler target KIC 9024795 serves as example: From the age-metallicity relationship we know that the star, with metallicity of [Fe/H] = -1.5, should be older than 8 Gyr. It has a rotation velocity of 26 km/s. With the gyrochronology approach the star is, even in the full range of uncertainties, younger than 6.5 Gyr (Angus et al. (2019)). With the tidal-interaction model, a companion with 7-10 jupyter masses on an initial distance of 0.015 - 0.026 au, could have caused this spin-up, as shown in Fig.5

Fig 5.: The parameter space for a high rotating, metal poor star.

The search for exoplanets delivered a lot of material during the past few years. The percentile of objects orbiting metal-poor stars on tight orbits is rather small though. The reason for that is not certain, as this might be an observational bias. Metal poor stars are usually older and fainter. That not only makes them harder to observe in the first place, but it is also more difficult to detect orbiting objects. The model presented in this work serves as a first approach to describe stellar spin-up for metal-poor main sequence stars.

To conclude, tidal interaction and the engulfment of a massive companion within the presented parameter space, lead to stellar spin-up during the main sequence. This not only solves the paradox of age deviations from different methods, but also reveals that gyrochronology ages can be misleading.

References:
Angus et al. (2019) "Toward Precise Stellar Ages: Combining Isochrone Fitting with Empirical Gyrochronology." 158, no. 5, 173 (November): 173. doi:10.3847/1538-3881/ab3c53. arXiv: 1908.07528 [astro-ph.SR].
Bensby et al. (2014) "Abundances of stars in different Galactic subsystems." 85 (January): 214. arXiv: 1312.4592 [astro-ph.GA].
Bouvier et al. (1997) "The angular momentum evolution of low-mass stars." 326 (October): 1023–1043.
Carone et al. (2007) "Constraints on the tidal dissipation factor of a main sequence star: The case of OGLE-TR-56b." 55 (April): 643–650. doi:10.1016/j.pss.2006.05.044.
Privitera et al. (2016) "Star-planet interactions. I. Stellar rotation and planetary orbits." 591, A45 (June): A45. doi:10.1051/0004-6361/201528044. arXiv: 1604.06005 [astro-ph.EP].

Improving reliability of photometric redshifts using machine learning methods

O. Razim[1] and G. Longo[1]

[1]Department of Physics, Strada Vicinale Cupa Cintia, 21, 80126, University Federico II, Napoli, Italy

What is photo-z

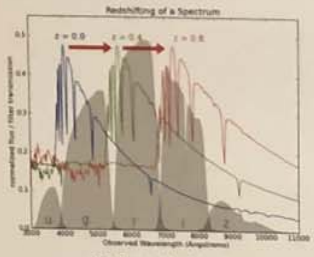


Fig 1. The principle of photo-z

When spectroscopic features, like Lyman and Balmer breaks, fall into a certain photometric band, they increase or decrease the corresponding magnitude (Fig. 1). From this change we calculate so called photometric redshift (photo-z) [1].

There are two main methods of obtaining photo-z: Spectral Energy Distribution (SED) fitting, and Machine Learning methods (ML) [2]. SED fitting requires preexisting spectral templates, obtained from theoretical models. ML instead requires representative spectr-z sample for the investigated photometric catalog.

Both methods have their limitations and can outperform each other in different cases.

COSMOS2015 + MLPQNA = bad results

COSMOS2015 [3] is a photometric catalog for approx. 0.5 million galaxies with $0 < \text{spectr-z} < 8$.

MLPQNA [4] is a ML algorithm for photo-z calculation, which showed good results on SDSS, KiDS, etc. [5,6].

Photo-z for COSMOS2015 produced with MLPQNA are much worse than in previous works (Tab. 1). The probable reason is that spectr-z catalog for COSMOS2015 was compiled from multiple catalogs from different instruments, and therefore is non-homogeneous. To fix this, we filter galaxies with non-typical spectr-z using SOM.

Catalog	Mean	NMAD	Std	% outl
SDSS DR9	10^{-5}	0.02	0.02	0.1
KiDS DR2	10^{-4}	0.02	0.03	0.3
COSMOS2015	10^{-3}	0.02	0.05	2

Tab 1. Some of the statistical estimators for MLPQNA photo-z for several catalogs

Self-organizing maps (SOM)

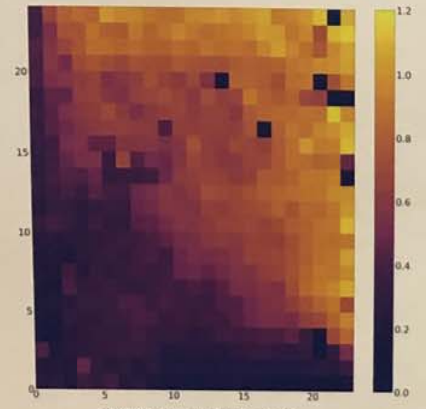


Fig 2. SOM with post-labelled mean spectr-z

SOM algorithm projects a dataset from a multi-dimensional parameter space to the 2D map. SOM does it in such a way that data points neighboring in the original space remain neighbors on the map [7].

Photometrically similar galaxies end up being in the same or in the neighboring cells. They also should have similar redshifts (Fig. 2). So we check spectr-z distribution within each cell and drop out outliers. In best cases it lessens the percentage of outliers from 2 down to 0.25 and standard deviation from 0.05 to 0.023 (Fig. 3). The effect on NMAD is negligible.

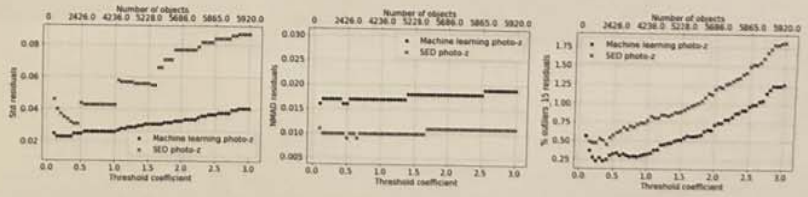


Fig 3. The change of standard deviation, NMAD and % of outliers of residuals with spectr-z filtering threshold

SOM for photometry calibration

In order to apply our trained ML model to the whole COSMOS2015 dataset, we must ensure that it is photometrically similar of the spectr-z catalog that we used for training.

Usually it is done by cutting the tail of the distribution, but this is not very accurate.

Instead we place the whole catalog on the trained SOM and discard galaxies that differ significantly from typical magnitude vectors of all cells. Magnitude distribution of the whole catalog becomes more like magnitude distribution of train spectr-z catalog (Fig. 4).

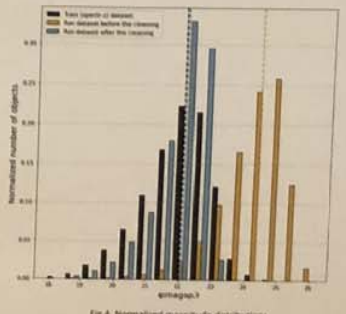


Fig 4. Normalized magnitude distributions

Conclusions

SOM filtering of galaxies, atypical in terms of spectr-z and photometry, allows to significantly lessen number of outliers and ensure that both train and run datasets lay within the same area of the parameter space. In future, we plan to investigate the nature of the filtered outliers and to try outlier detection methods for the same purposes.

[1] Baum W., 1962, Proceedings from IAU Symposium no. 15, p. 300
[2] Salvato M., Ilbert O., Hoyle B., 2019, Nature Astronomy, 3, 212
[3] Laigle C., et al., 2016, ApJS, 224, 24
[4] Cavuoti S., Brescia M., Longo G., Mercurio A., 2012, A&A, 546, A13
[5] Brescia M., Cavuoti S., Longo G., De Stefano V., 2014, A&A, 568, A126
[6] Cavuoti S., et al., 2017b, MNRAS, 466, 2039
[7] Kohonen T., 1982, Biological Cybernetics, 43, 59

Alex Razim: shr.razim@gmail.com
January 2019

This project has received financial support from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 721463 to the SUNDIAL ITN network.

Multi-Cloud GPU Burst for Multi-Messenger Astrophysics

Benedikt Riedel[1] David Schultz[1] Igor Sfiligoi[2] Frank Wuerthwein[2]
[1] University of Wisconsin - Madison, Madison WI 53715 USA
[2] University of California San Diego, La Jolla CA 92093, USA

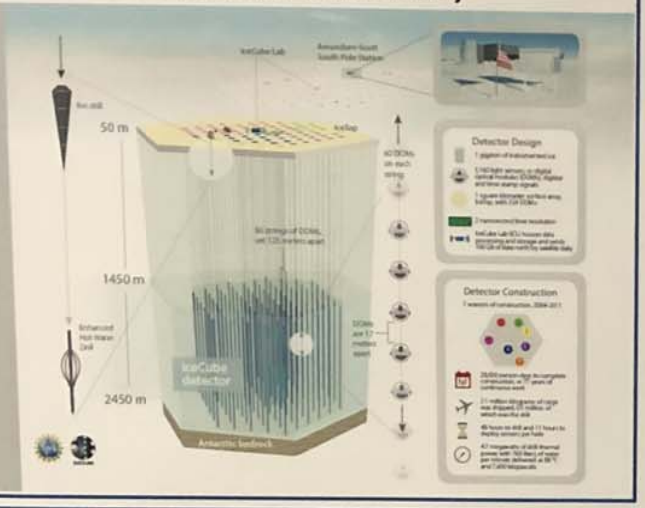
As we approach the Exascale era, it is important to verify that the existing frameworks and tools will still work at that scale. Moreover, public Cloud computing has been emerging as a viable solution for both prototyping and urgent computing. Using the elasticity of the Cloud, we have put in place a pre-exascale HTCondor setup for running scientific simulation in the Cloud, with the chosen application being IceCube's photon propagation simulation. This was not a purely demonstration run, but it was used to produce valuable and much needed scientific results for the IceCube collaboration. In order to reach the desired scale, we aggregated GPU resources across 8 GPU models from many geographic regions across Amazon Web Services, Microsoft Azure, and the Google Cloud Platform. Using this setup we reached a peak of over 51k GPUs corresponding to almost 380 PFLOP32s, for a total integrated compute of about 100k GPU hours. In this paper we provide the description of the setup, the problems that were discovered and overcome, as well as a short description of the actual science output of the exercise.

Introduction

Multi-messenger astrophysics has developed into a full-fledged observational over the last 30 years. Current multi-messenger experiments, such as the IceCube Neutrino Observatory, require a large amount of compute. With the exascale systems coming online in the coming years, "bursting" (quickly filling in gaps) will become especially important for experiments to utilize spare capacity on these systems. Cloud providers can provide us the scale to test whether experiments can utilize these resources effectively.

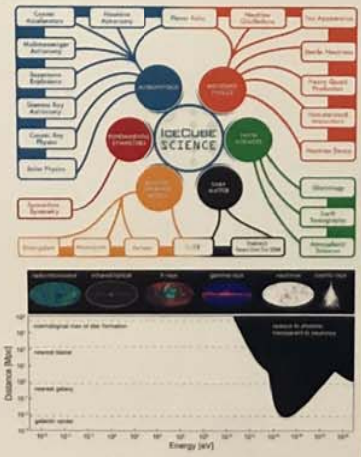
IceCube Neutrino Observatory

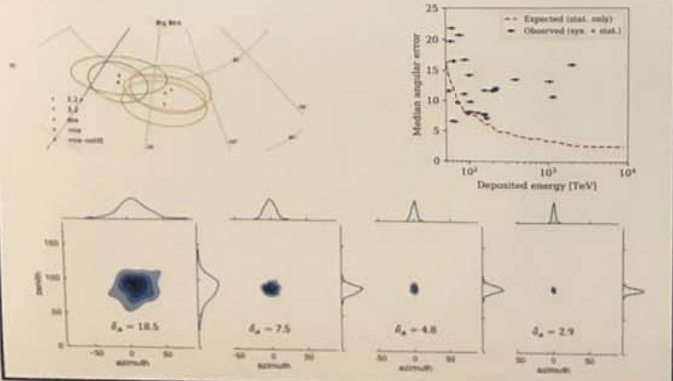
The IceCube Neutrino Observatory
Design and construction



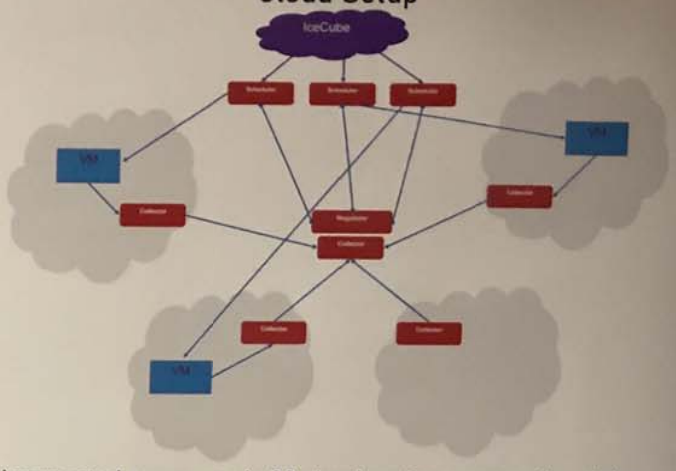
Science Case

- IceCube has broad science case beyond neutrino astrophysics
- Universe opaque to light at highest energies and large distances
- Only gravitational waves and neutrinos pinpoint to most violent events in universe
- Ice produces large systematic effects at highest energy, esp. for ν_e and ν_τ
- Improve event selection and pointing resolution needs





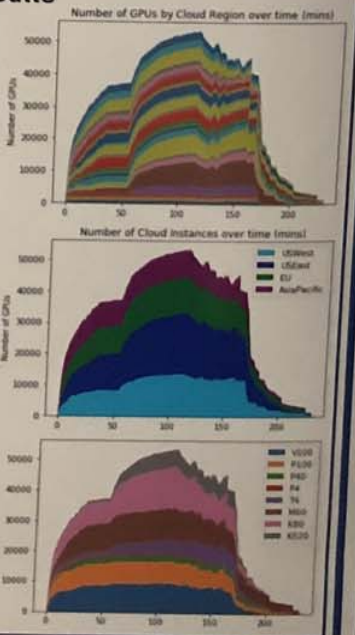
Cloud Setup

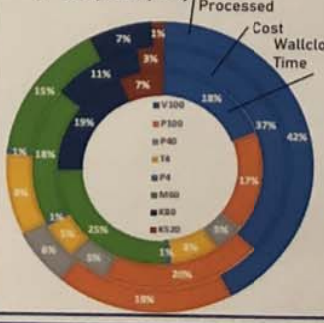


- Aggregated resources in 28 cloud regions across the globe into single HTCondor pool
- Tiered pool - Each region's resources collected into a regional pool and then joins global pool
- Existing technology and expertise from Open Science Grid
- On the same scale as compute pools being run for IceCube, CMS, LIGO, etc.
- Input and output data was staged to cloud storage in each region – Reduce a source of complexity
- Simple wrapper script to paper over differences in storage APIs, location, etc.

Results

- Initial setup tested using CPUs – Cheap
- Network testing showed over 1 Tbit/s networking inside a region
- No single region or GPU type contributed more than 11% in compute effort
- 8 generations of nvidia GPUs
- Geographical even split
- Best cost/science on newest GPUs (V100, T4)





Conclusions

We have shown that IceCube can utilize exascale class resources for their simulation production effectively in a "burst" mode. This yielded the largest GPU resource pool every created in the cloud.

SDSC San Diego Supercomputer Center · PRP Pacific Research Platform · Internet2 · HT Center for High Throughput Computing · HTCondor · IceCube · Open Science Grid

5

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Machine Learning and Big Data in Cherenkov Astronomy

B.Schleicher and D. Dorner

Abstract: Being a data-intensive and analysis-intensive field, Cherenkov astronomy has several use cases for machine learning methods. For example, in the background suppression, it can be used to differentiate between gamma rays and cosmic rays as primary particle of the shower that produced the observed Cherenkov light. Also for reconstructing the origin or the energy of the events, machine learning can be applied. The challenge for all these use cases is that for training the methods, simulated data are needed. If the simulated events do not describe the real data correctly, the machine learning methods do not provide equally good results on the real data. On the other hand, generative adversarial networks might help to reduce the mismatch between real and simulated data. Furthermore, machine learning methods are interesting for the high-level analysis, e.g. for studying light curves and predicting the behavior of a source. Therefore systematic studies of variability and periodicity can profit from machine learning approaches. The long-term goal is to predict the flux for variable sources and coordinate multi-wavelength observations and studies based on this.

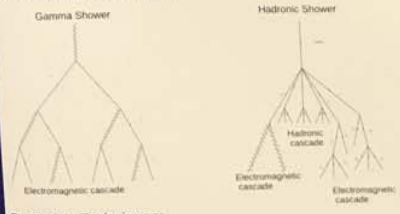
Overview of Imaging Air Cherenkov and Water Cherenkov Telescopes

→ Big Data Challenge

Example FACT:
- Total amount of physics data: >14720 hours → 2.4 Mio Events
- Physics data per year: ~ 2500 hours → 0.4 Mio Events
- Data Rate: 5-10 TB/month → ~80 TB/year

Air Showers

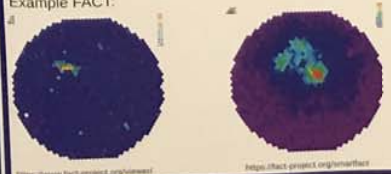
Simulated particle shower:



Detection Techniques:
- Imaging Air Cherenkov Technique
- Water Cherenkov Technique

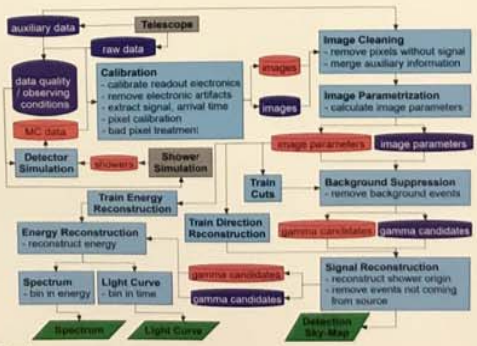
Indirect measurement:
- Atmosphere part of the detector
→ Air showers need to be included in simulation

Detector Response
Example FACT:



Data Analysis

Data Flow



Challenge:
Atmosphere is part of the detector→ Changes need to be included in simulation

Observational conditions:
- Zenith distance
- Ambient light (e.g. moon)
- Star field

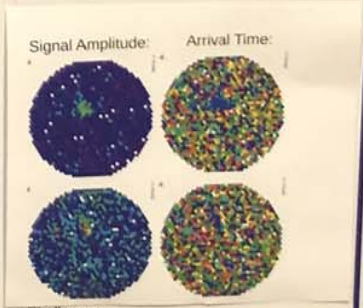
Atmospheric conditions:
- Weather (e.g. clouds)
- Dust in the atmosphere (e.g. calima)
- Seasonal changes (e.g. water vapor concentration)

Possible Solution:
Tailored Simulations → Large-scale computing needed

Machine Learning Use Cases

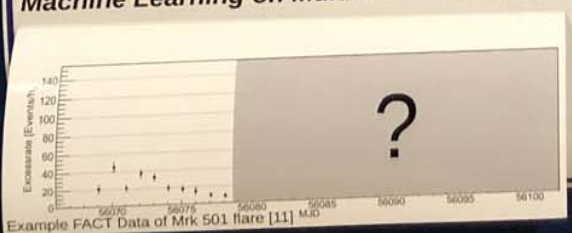
Classification and Regression problems:
- Particle classification
- Reconstruction of shower origin
- Energy reconstruction

Can you guess the kind of the particle?



Challenge:
Mismatch between real and simulated events
Possible solution:
Generative adversary networks [10]

Machine Learning on Multi-Dimensional Time Series



Example FACT Data of Mrk 501 flare [11]

- Predict flares based on long-term light curves
- Include multi-wavelength information to improve predictions
- Long-term monitoring and/or all-sky coverage needed to provide training sample
- Simulated light curves might help, but they should describe real light curves well
- Computing intensive

References
[1] https://veritas.sao.arizona.edu/
[2] The Cherenkov Telescope Array Consortium, arXiv:1709.07997
[3] https://magic.mpp.mpg.de/
[4] H.Anderhub et al. (FACT Collab), JINST 8 (2013) P06008, arXiv:1304.1710
[5] D. Dorner et al., arXiv:1610.06625v1
[6] https://www.hawc-observatory.org/
[7] H. Schoorlemmer on behalf of the SWGO collaboration, PoS(ICRC2019)785
[8] W. Hofmann, for the H.E.S.S. Collaboration, Proceedings of ICRC 2001: 2785
[9] X. Bai et al., arXiv:1905.02773v1
[10] M. Erdmann et al., arXiv:1802.03325
[11] Ahnen et al. (MAGIC,FACT,VERITAS,others), A&A 620 (2018) A181

Affiliation:
Universität Würzburg, Germany - Institute for Theoretical Physics and Astrophysics
Emil-Fischer-Str. 31, 97074 Würzburg
E-mail: bernd-schleicher@stud-mail.uni-wuerzburg.de

PAHN-PaN: Particle, Astroparticle, Hadron & Nuclear- Physics accelerate the NFDI

Andreas Haungs (KIT), Gregor Kasieczka (Hamburg), Arnulf Quadt (Göttingen),
Thomas Schörner (DESY), Kilian Schwarz (GSI)

Bonn-Königswinter, December 2019

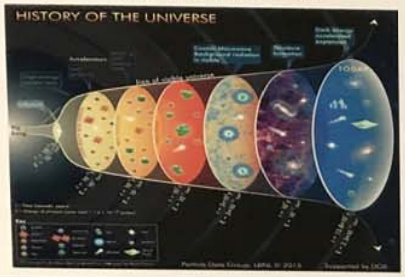
Research

Fundamental particles, forces and symmetries

Some of the key questions in PAHN

- Dynamics of Quark Gluon Plasma
- Formation of structure from strong interactions
- Mass and properties of the Neutrinos
- Asymmetry of matter and antimatter
- "Known" baryonic matter is only 5% of the universe
- Nature of dark matter and dark energy

Close relations to astronomy and cosmology



Experimental and Computing Infrastructures

- German astro-particle-, particle- and hadron physics community strong player in the large global collaborations
- Also involved in a number of international experiments with 100+ members: Belle2, IceCube, Pierre Auger, KM3NeT, KATRIN
- R&D projects for future detectors and more
- Several "smaller" complementary research infrastructures, e.g. MAMI, MESA, S-DALINAC
- Re-use of successful building blocks from WLCG, middleware and scientific software packages within PAHN-PaN and by partner consortia in NFDI

Data Volumes

PAHN has significant experience in operating federated compute and data infrastructures for research data. An example is the WLCG, consisting of more than 170 computing centres in 42 countries.

ALICE@LHC Run3

PAHN-PaN@NFDI:

an exemplary view

- novel discoveries by sharing data
- interconnect global data stores
- sustainable data accessibility
- data and meta data management using FAIR principles

The PAHN-PaN Consortium

Our mission

Particle, astroparticle, and hadron&nuclear physics

- Decade-long experience in operating a self-developed global big data management infrastructure (WLCG, the world's largest grid).

PAHN-PaN goals

- Innovative, industry-standard solutions for *FAIR* Exabyte data managementand scientific services
- Foster data management at small sites: accelerators like MAMI, S-DALINAC, on-site experiments like KATRIN, theory.

Synergies, solutions, services

- Using NFDI synergies for common developments.
- Knowlegde and technology transfer to entire
- Accessible to PAHN-PaN and the entire NFDI.

Necessary for analysis by O(10000 users):
- 1 PB/a non-zero-suppressed raw input data / experiment
- > 500 000 grid jobs running at any time
- Example ATLAS storage: 223+315 PB

Unprecented scale & complexity of data!

But: facing massive challenges: volumes, rates – order of magnitude in next decade: HL-LHC upgrade, CTA, FAIR facility etc.!

Solutions for
- the entire data lifecycle (including data publication and archiving) and
- long-term re-usability of data

FAIR Data Lifecycle Concepts and Open Data

Overview

- **Each arrow requires *FAIR* data management**
- **Each step needs appropriate metadata**
- **The cycle includes data, metadata and workflows**

- Where possible, establish common standards to foster interoperability (FAIR)
- Importance of "data stewards" as data lifecycle managers and metadata curators
- The lifecycle has to provide a FAIR data
  (i) availability (ii) method development
  (iii) analysis (iv) education
  (v) open access (vi) archival

Professional Training, Education, Outreach

WP1: Training
- guidance for career planning
- science management skills
- foreign talents/gender aspects

WP2: Education
- curricula
- collect/develop material
- virtual NFDI-wide seminars

WP2: Outreach
- Masterclass
- science faires
- open access publications

Consortium Partners

Research in (astro-) particle physics and hadron physics is spread all over Germany
- Research centres
- Universities

National research coordinated via committees
- Komitee für Elementarteilchenphysik (KET)
- Komitee für Hadronen und Kerne (KHuK)
- Komitee für Astroteilchenphysik (KAT)

Three committees pursue a joined NFDI approach
~3,300 scientists with PhD degree represented by PAHN-PaN NFDI consortium

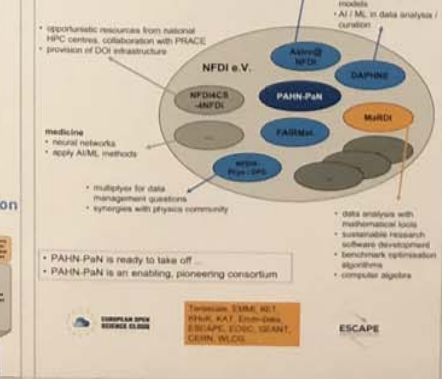
Governance and Organisation

A community used to collaboration

- Viable operating model for providers + users
- Clear interfaces to the NFDI + other consortia
- Dynamical reaction to new requirements

Synergies

And relations to other NFDI consortia



- mastering data irreversibility challenge (PAHN-PaN → Astro)
- open data and meta data definitions (Astro → PAHN-PaN)
- big data management models
- AI / ML in data analysis / curation
- opportunistic resources from national HPC centres, collaboration with PRACE
- provision of DOI infrastructure

medicine
- neural networks
- apply AI/ML methods

- multiplyer for data management questions
- synergies with physics community
- data analysis with mathematical tools
- sustainable research software development
- benchmark optimisation algorithms
- computer algebra

- PAHN-PaN is ready to take off ...
- PAHN-PaN is an enabling, pioneering consortium

Tentacles: EMMI, KIT, RWuK, KAT, Exzellenz, ESVAPE, EOSC, GIANT, CERN, WLCG

ESCAPE

Summary

PAHN-PaN: established international communities with
- decade-long experience in large-scale data management,
- existing infrastructures and
- successful handling of thousands of users.

A pioneering consortium – exciting challenges ahead in terms of data volumes and rates. Solutions
- new technologies and concepts
- NFDI as incubator for future ideas.

Already now, PAHN-PaN can
- assist the NFDI to tackle medium- and long-term challenges in data management,
- help to keep the German science system competitive and
- strengthen the industry location via technology development and training.

PAHN-PaN needs from NFDI: improved software, FAIR compliant data management, sustainability
PAHN-PaN contributes to NFDI: well tested software, Big Data operational experience, know-how transfer via schools and outreach.

PAHN-PaN: Particle, Astroparticle, Hadron & Nuclear Physics accelerates the NFDI

Astronomy meets big data:
Improving the Milky Way model with the billion-star surveyor *Gaia*

Kseniia Sysoliatina and Andreas Just

Astronomisches Rechen-Institut, Mönchhofstr. 12-14, 69120 Heidelberg, Germany
Contact: Sysoliatina@uni-heidelberg.de

1. Gaia mission and Galaxy modelling

The ESA's mission Gaia has mapped about 1.6 billion of stars in the Milky Way (DR2, [1]) that corresponds to about 200 Tb of raw data during its five-year nominal mission. For most of these stars five astrometric parameters (positions, proper motions, and parallaxes) are known, and a 7.2-million subset of bright stars additionally contains radial velocities providing us with the full 6D dynamic information for the extended solar neighbourhood. The high quality and abundance of these data strongly stimulate the development of existing Milky Way models, thus improving our undertanding of the Galaxy structure and evolution.
Semi-analytic, physically and observationally motivated models of the Galaxy usually assume stellar density distrbutions and kinematic properties of the thin and thick disks, halo, and bulge (TRILEGAL code, [2]), or derive the density profiles in a self-consistency with the gravitational potential (BGM, [3,4]; JJ model, see below). When combined with the data from large astrometric and spectroscopic surveys, these models predictions shed light on the past of the Galaxy by providing constraints on its star formation, dynamical heating, and chemical enrichment history.

2. The Galactic disk model

Just-Jahreiß (JJ) model [5] is a chemo-dynamic model that concentrates on the detailed vertical structure of the thin disk. It views the present-day thin disk properties as a function of:

- a declining star formation (SF) rate (**SFR**) with a recent SF outburst possible
- a monotonously increasing power-law age-velocity dispersion relation (**AVR**)
- a four-slope broken power law initial mass function (**IMF**) [6]
- a simple enrichment law in the form of age-metallicity relation (**AMR**).

Calibrated locally with the *Hipparcos* data [5] and SDSS star counts [7], and tested recently with the RAVE DR5 and Gaia DR1 data in the solar cylinder [8], this model can be further improved with Gaia DR2 (this work).

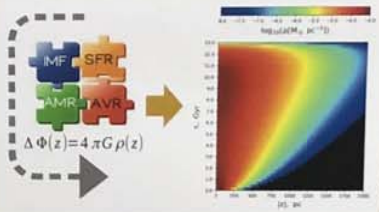


Figure 1.
Left: The JJ model structure. Poisson eq. is iteratively solved, given SFR and AVR, to get a self-consistent pair of the vertical gravitational potential and stellar density law.
Right: Vertical density profiles of the different thin disk mono-age populations. Complemented with IMF, AMR, and a stellar library, they give stellar number density and can be represented in a form of such observational quantites as Hess diagrams and stellar velocity or metallicity distribution functions.

3. Using Gaia DR2 data effectively

As many of the model parameters can be naturally correlated, the most robust way to find their best values is to investigate full parameter space. An effective way to do this is to perform a parallelized MCMC sampling of posterior distribution; the latter characterises model-to-data goodness of fit and takes into account our preliminary knowledge on parameter values (Bayes' theorem).
Such a fitting approach imposes a strong constraint on the maximum model-to-data comparison time for a single combination of fitting parameters, which leads to special constraints on the data selection. We define magnitude-complete data samples de-reddened with the local extinction map [9] (about 10^6 stars in total):

- Sampels of A-, F-, RC/RGB-stars, G- and K-dwarfs (Fig.2). For these stars we calculate:
  - Vertical number density profiles (Fig.3, a)
  - W-velocity distributions (Fig. 3, b; for subsets with known radial velocities)
- All stars in a cone perpendicular to the Galatic plane with 10° opening angle. A significant fraction of this sample are the thick disk stars, so it is used as an additional constraint on the thick disk parameters in the form of apparent Hess diagram (Fig.3, c).

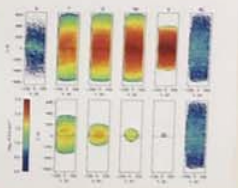


Figure 2. Spatial distribution in XZ coordinates of the six Gaia samples (top) and their subsamples with radial velocities available (bottom).

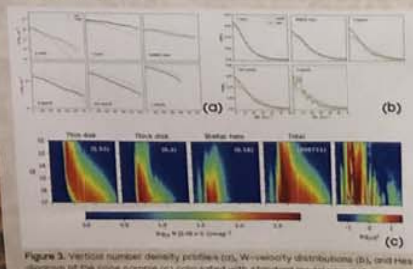


Figure 3. Vertical number density profiles (a), W-velocity distributions (b), and Hess diagram of the cone sample (c) calculated with standard model parameters.

For all three types of quantities (vertical profiles, velocity distributions, and Hess diagrams) we perform data binning, and thus reduce our data sample size from 10^8 (stars) to about 10^4 (bins).
Ten parameters we choose to adapt are:

$$\theta = \{\Sigma_{d1}, \Sigma_{d2}, \Sigma_{dh}, \sigma_1, \alpha_{\rm AVR}, \sigma_1, \alpha_{\rm AVR}, \beta_{\rm SFR}, \tau_p, \Sigma_p\}$$

Local surface density normalisations (thin disk, thick disk, dark matter halo) — AVR and thick disk vertical kinematics — Thin disk SFR shape, with a secondary SF peak 2–3 Gyr ago allowed

Preliminary tests indicate that it might be difficult to reproduce all observed data features within the assumed framework, such that an additional variation of IMF parameters might be required; alternatively, this one-zone Galactic model reaches the limit of its performance and further improvements can be achieved by switching on additional physical processes, such as stellar migration (Sysoliatina and Just, in prep.).
To sum up, the efficient exploration of the parameter space even of a relatively simple Galactic model requires reduction of the large initial catalogue to much smaller set of its statistical properties.

References & Acknowledgements

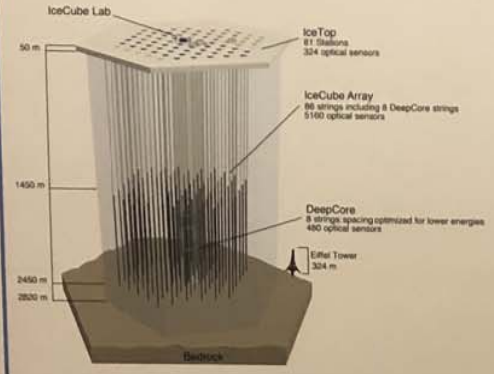
[1] Gaia Collaboration et al. 2018, A&A, 16, A1
[2] Girardi, L. et al., 2005, A&A, 436, 895
[3] Czekaj M. A., et al. 2014, A&A, 564, A102
[4] Sharma S., et al. 2011, Astrophysics Source Code Library
[5] Just A., Jahreiß H., 2010, MNRAS, 402, 461
[6] Rybizki J., Just A., 2015, MNRAS, 447, 3880
[7] Just A., Gao S., Vallrih S., 2011, MNRAS, 411, 2586
[8] Sysoliatina et al. 2018, A&A, 620, A71
[9] Lallement, R. et al. 2018, A&A, 616, A132

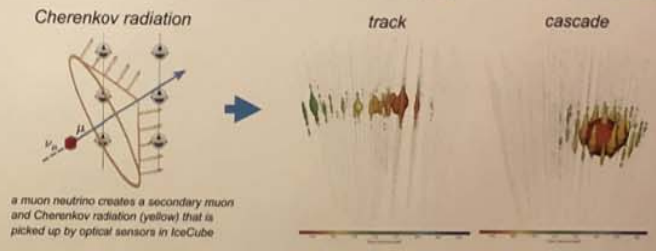
This work is supported by Sonderforschungsbereich SFB881 «The Milky Way System» (subproject A6) of the German Research Foundation (DFG).

From 2D to 3D to Graphs: Representing Detector Geometries in Neural Networks.

A. Trettin for the IceCube Collaboration

The IceCube array: a detector for neutrinos from the sky



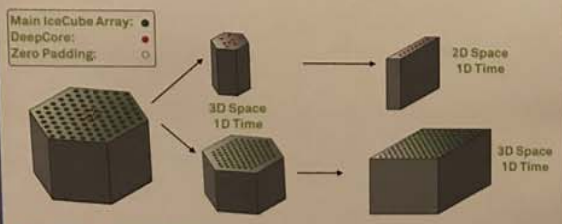


a muon neutrino creates a secondary muon and Cherenkov radiation (yellow) that is picked up by optical sensors in IceCube

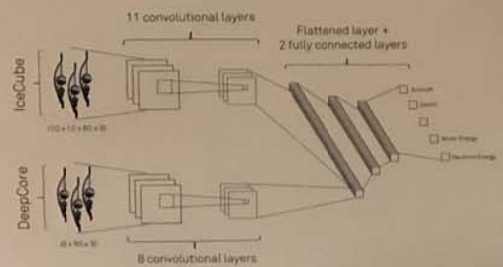
- sensors are activated in different patterns depending on the particle's properties
- aim: determine **type** of particle (classification) as well as its **energy and direction** (regression) from observed patterns
- approximate translation invariance of patterns suggest use of convolutional neural networks to approach classification and regression tasks

3D Convolution Kernels

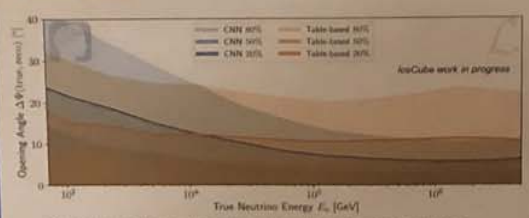
Re-arranging a hexagonal into a regular grid



Network Architecture



Performance

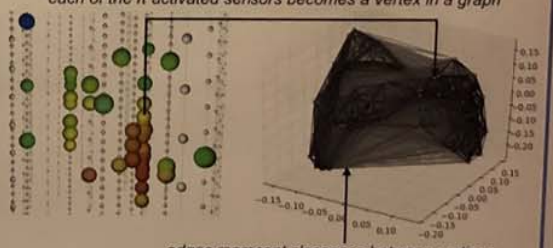


Comparison of the angular errors in cascade direction reconstructions between the CNN and a conventional likelihood-based reconstruction method. Thick lines represent the median error. Lower is better.

Graph Neural Networks

Theory

each of the n activated sensors becomes a vertex in a graph



edges represent closeness between vertices

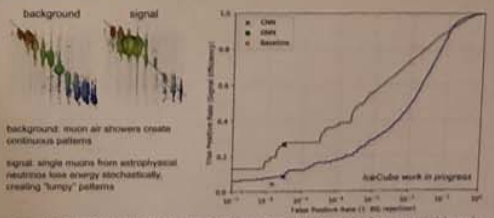
- closeness between sensors i and j described by trainable adjacency matrix A_{ij}
- can apply "graph convolution" to $(n \times d)$ input matrix X:

Graph Convolution

$$\mathrm{Spread}(X) = AX||IX$$
$$\mathrm{GConv}(X) = \mathrm{Spread}(X)\theta_w + \theta_b$$

|| = concatenation
I = identity matrix
θ_w = trainable weights
θ_b = trainable biases

Performance



Receiver operating curve for the task of identifying astrophysical muon neutrinos in IceCube. Baseline is a conventional likelihood-based method.

References

- N. Choma et al., "Graph Neural Networks for IceCube Signal Classification", *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, 2018, pp. 386-391. doi: 10.1109/ICMLA.2018.00064
- M. Hünnefeld, *Master Thesis*, Technische Universität Dortmund, 2017

HELMHOLTZ

DESY.

IceCube

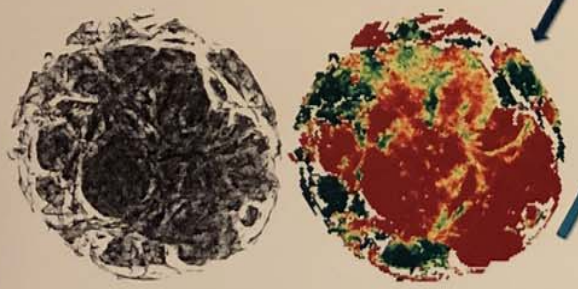
Leibniz-Institut für
Astrophysik Potsdam

Classification of high-resolution solar Hα spectra using t-SNE

Meetu Verma[1], Gal Matijevič[1], Carsten Denker[1], Andrea Diercke[1,2], Christoph Kuckein[1], Horst Balthasar[1], Ekaterina Dineva[1,2], Ioannis Kontogiannis[1], and Partha S. Pal[1,3]

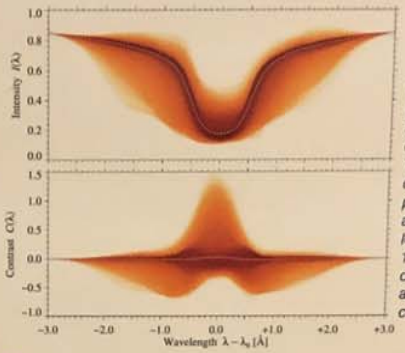
1 Leibniz-Institut für Astrophysik Potsdam (AIP), Germany 2 Universität Potsdam, Institut für Physik and Astronomie, Germany
3 University of Delhi, Bhaskaracharya College of Applied Sciences, Delhi, India

Abstract. Starting mid-2020, the Daniel K. Inouye Solar Telescope (DKIST), 4-meter solar telescope will become operational. With five post-focus instruments equipped with large format detectors, the expected annual data rate is around 3 PB. Data include not only images, but imaging spectropolarimetric as well as high-precision full-Stokes spectropolarimetric spectra. With this amount of data, it will be impossible for astronomers to inspect each and every spectrum individually. We propose a framework to classify solar spectra using t-distributed Stochastic Neighbor Embedding (t-SNE) to speed up the basic spectral inversion. This study is based on high-spectral resolution Hα spectra obtained with the Echelle spectrograph of the Vacuum Tower Telescope (VTT) located at Observatorio del Teide, Tenerife, Spain. The Hα spectral line is a well-studied absorption line, revealing properties of the highly structured and dynamic solar chromosphere. Typical features with distinct spectral signatures in Hα include filaments/prominences, bright active region plages, superpenumbrae around sunspots, surges, flares, Ellerman bombs, filigree, and mottles/rosettes, among others.

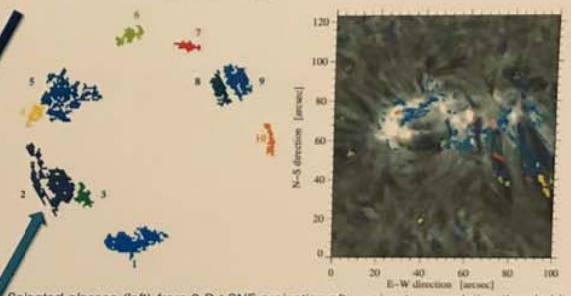


2-D t-SNE projection (left) of 630 x 660 contrast profiles showing various clusters. 2-D t-SNE projection (right), colored using the linear and rank-order correlations when comparing observed profiles with cloud model (CM) inversion, depicts two classes. One with contrast profiles (green) suitable for CM inversions and another with quiet-Sun and emission profiles (red), which cannot be inverted using CM inversion.

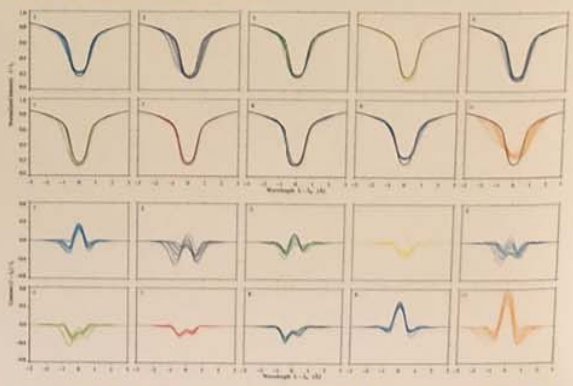
t-SNE is a machine learning algorithm, which is used for nonlinear dimensionality reduction. In this application, it projects the Hα spectra onto a 2-D map, where it is easy to classify them according to results of Cloud Model (CM) inversions, i.e., optical depth, Doppler width, line-of-sight velocity, and source function of the cloud material. Initial results of t-SNE indicate its strong discriminatory power to separate quiet-Sun and plage profiles from those that are suitable for CM inversion. In addition, the identified classes are linked to chromospheric features, the impact of seeing conditions on the classification is assessed, the projection of new Hα spectral data (different observing time and target) onto the 2-D tSNE maps is inspected to optimize CM inversions, and representative Hα spectra are determined as input for deep neural networks speeding up the CM inversion.



Two-dimensional histograms of observed, noise-stripped Hα intensity (top) and contrast (bottom) profiles. The distributions were divided by the number of profiles (about 8.7 million) and are displayed on a logarithmic scale between 10^{-6} and 10^{0}. The red-white dashed curves refer to the average Hα intensity and contrast profiles.



Selected classes (left) from 2-D t-SNE projection after using t a correlation threshold of 0.9 and better. The color code is based on the number of profiles in the classes (blue to red), where as numbers are marked by counting the classes counter-clockwise starting from the bottom. The location of extracted classes are marked on the slit-reconstructed Hα line-core intensity map. These are the regions, where the Hα profiles can surely be inverted using cloud model.



Intensity (top) and contrast (bottom) profiles belonging to ten selected classes. The shown 20 profiles in each panel are randomly chosen. The quiet-Sun intensity profile (black) is plotted for reference. The variation across all ten classes is evident in both intensity and contrast profiles.

NuRadioReco and NuRadioMC.

A Software Framework for the Radio Detector Community

Christoph Welling for the RNO-G Collaboration

Web-based Event Viewer
> Use latest web technologies
> Deployable locally or online

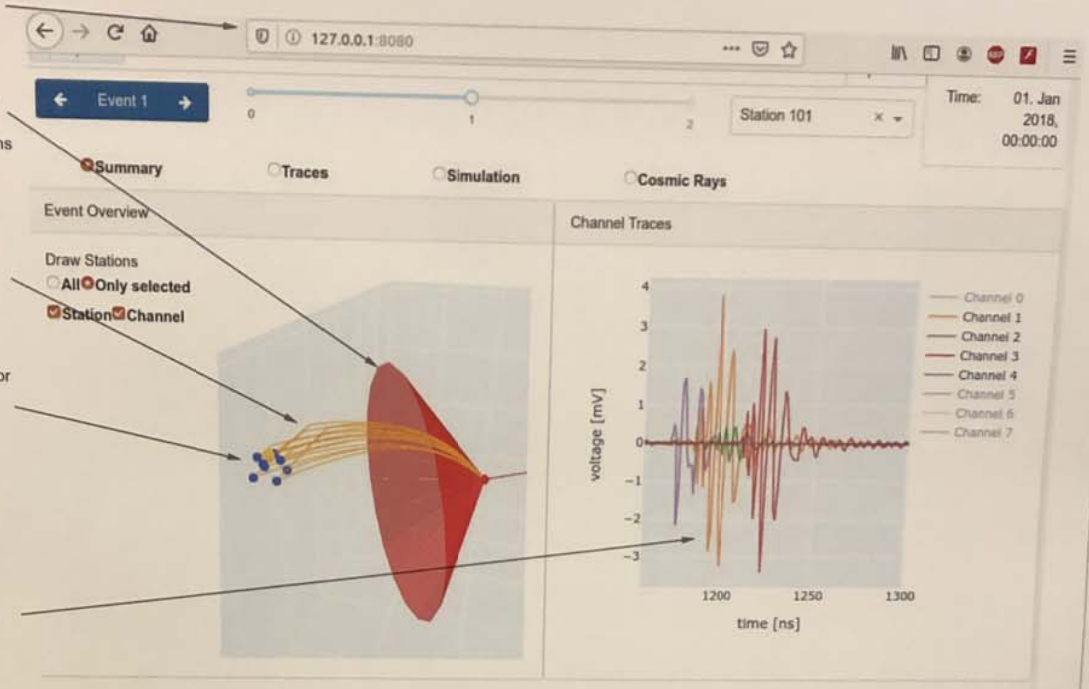
Signal Generation
> Signal generated using state-of-the-art semi-analytic model
> Support for older parametrizations for cross-checks
> Treatment of LPM elongation

Signal Propagation
> Analytic ray-tracing through medium with refractive index $n(z) = n_{ice} - \Delta_n e^{\frac{z}{z_0}}$
> Fast ray-tracing via C++ raytracing module

Detector Description
> Configuration & state of detector
> Time-dependent
> Detector database access
> Custom detectors in JSON format

Detector Simulation
> Detector effects on signal
> Generic or measured noise
> Trigger simulation

Event Reconstruction
> Split into individual modules
> Write data to disk at any time



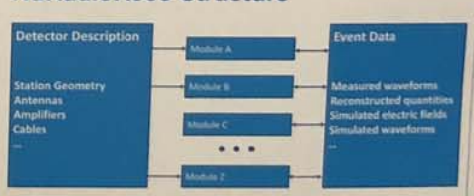
Experimental Background

> Targeting neutrinos at energies > 10 PeV
> First discovery-scale radio detector for neutrinos to be built in 2020
> 35 in-ice stations in Greenland
> Radio to be part of IceCube Gen2
> Simulation & design studies ongoing
> Radio detection of cosmic rays is well established
> Cosmic ray signals valuable calibration source

Design Goals & Philosophy

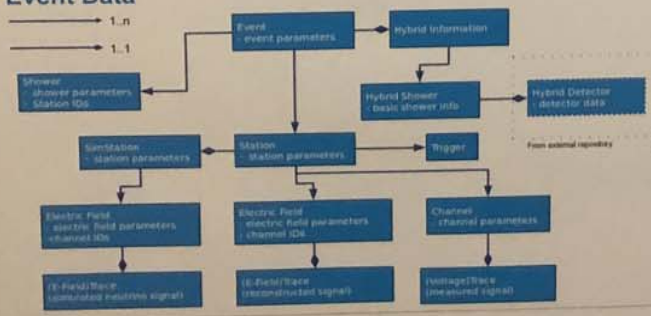
> Complete simulation and reconstruction package
> Python-based
> Open Source / GNU
> Community-driven
> Modular
> Flexible detector layouts
> Support most radio detector experiments
> For in-ice neutrino and cosmic ray detectors
> Accessible
> Utilize GitHub workflow: Pull requests, issues, unit tests...
> Build on experience from previous experiments

NuRadioReco Structure



> Detector Description:
Provides information about detector
> Event Data:
Stores measured raw data
Stores simulated and reconstructed quantities
> Modules:
Perform detector simulation and event reconstruction
Can manipulate event data
Can only read detector description

Event Data



> Hierarchical structure of data objects
> Channel voltage and electric field traces stored in dedicated classes
> Can serialize itself into pickle-like *.nur* format
> Can be written to disk at any time
> Can hold additional data from other detectors

References

> C. Glaser et al. "NuRadioReco: a Reconstruction Framework for Radio Neutrino Detectors." EPJ-C 79.6 (2019)
> C. Glaser et al. „NuRadioMC: Simulating the radio emission of neutrinos from interaction to detector" arxiv:1906.01670, submitted to EPJ-C
> NuRadioReco on GitHub: github.com/nu-radio/NuRadioReco
> NuRadioMC on GitHub: github.com/nu-radio/NuRadioMC

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS
DFG Deutsche Forschungsgemeinschaft

Prototypes for the Next Generation of Computing Backends in Radio-Astronomy

T. Winchen, A. Bansod, E. Barr, M. Heininger, S. P. Sathyanarayanan, G. Wieching, J. Wu
Max Planck Institute for Radio Astronomy, Auf dem Hügel 69, 53121 Bonn, Germany

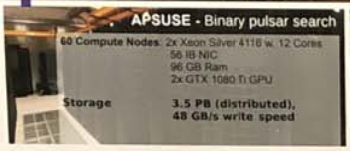
Max-Planck-Institut für Radioastronomie

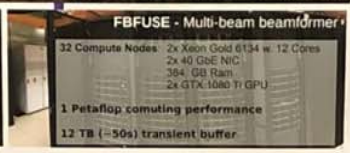
Different Telescopes and Computing Clusters


Effelsberg, Germany


MeerKAT, South Africa


TNRT, Thailand





One Modular Backend-Software Design

Service: Backend comprised of multiple services deployed in cluster with high-speed network connected to telescope data stream

High-speed network

Control network

Component Control

Data Processing

Docker Registry

Redis

Data Transport

- High-speed data network with RDMA (56Gb IB, 100 GbE, ...)
- Telescope / data products stream to individual multicast groups
- Spead2 protocol (but also VDIF)
- Capture / send from ring buffer (dada) via specialized programs:
  `mkrecv, mksend, vdif_send`

Telescope Connectors

- Interface to telescope
- Translation of telescope commands to internal katcp protocol, e.g. from SCPI for Effelsberg
- Connection to telescope data stream:
- Only customized component

Component Control

- Control applications based on `mpikat`, client/server framework based on `katcp`:
  - Manage buffers
  - Start/stop sub-processes for send, receive, process programs
  - Load FPGA firmware
  - Provide sensors for telemetry data
- Access to global data storage realized as Redis key/value store

Data Processing

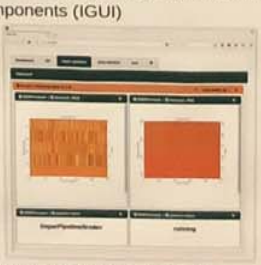
- Multiple services for different data products, e.g. spectrometer, pulsar, vlbi ...
- Data processing by different technologies CPU, GPU, FPGA
- Dada ring buffer via `psrdada_cpp`
- CUDA for GPU computing

Provisioning

- All services containerized using docker
- Containers stored in local docker registry
- Services launched on demand for different observations
- Container build + launched via Ansible
- Ansible executed by `master-controller` controlled via katcp
- Future Kubernetes instead of Ansible?
  → NUMA management problematic

Monitoring

- Web based monitoring of all backend components (IGUI)



- Future: Connection to control ?

Design Summary

Adaptability:
Rapid adaption to individual observations and new/exotic science cases, e.g. real time transients with machine learning, detection of extreme energy particles, ...

\+ **Commensality:**
Simultaneous processing for multiple disciplines and also offline data processing

\+ **Simplification:**
Minimize in-house solutions, maximize reuse of components and prefer COTS computing hardware.

\+ **Standardization:**
Use established open source software and industry operation standards

= **Maximize science / €**
Reduce development time (=money)

Resources

https://github.com/TobiasWinchen/mpikat
https://github.com/TobiasWinchen/psrdada_cpp
https://gitlab.mpifr-bonn.mpg.de/mhein/mkrecv
https://gitlab.mpifr-bonn.mpg.de/mhein/mksend

Contact: twinchen@mpifr-bonn.mpg.de

LHELM UND ELSE RAEUS-STIFTUNG

711. WE-Heraeus-Seminar

ne Science Cloud – Towards a Research Data Ecosystem for the ext Generation of Data-intensive Experiments and Observatories

SFB 1258 Neutrinos Dark Matter Messengers

BoostNumpy

BoostNumpy: Big Data Processing in C++ with Python convenience

Martin Wolf martin.wolf@tum.de
Experimental Physics with Cosmic Particles (ECP),
Technical University Munich, Germany

Efficient processing of big data can only be achieved with algorithms implemented in a compiled language like C++. However, for convenient steering of these compiled algorithms an interpreted scripting language like Python is desired. This contribution presents the meta-programming library "BoostNumpy" that serves as interface between C++ and Python by utilizing the Boost.Python library [1] and the numpy software package [2] for high-performance big data storage management and processing.

Overview & Example

BoostNumpy is an extension of boost::python to handle numpy arrays in C++ code. It introduces the `boost::numpy::ndarray` class derived from `boost::python::object` to manage `PyArrayType` objects, i.e. numpy arrays.

This project is based on an implementation by Jim Bosch et. al. [3]. The major development of BoostNumpy is the `dstream`, a.k.a. data stream, sub-library for the vectorization of (scalar) C++ functions. It implements the Generalized Universal Functions approach described by the numpy community. BoostNumpy uses meta-programming (MPL) to achieve the vectorization of a C++ function with only one line of code.

Example:

```
#include <boost/python.hpp>
#include <boost/numpy/dstream.hpp>

namespace bp = boost::python;
namespace bn = boost::numpy;

double square(double v) { return v*v; }

BOOST_PYTHON_MODULE(my_py_module)
{
    bn::initialize();

    bn::dstream::def(
        "square", &square, bp::arg("v"),
        "Calculates the square of v.");
}
```

The square function in Python will accept a numpy array as input and will return a numpy array as output:

```
import numpy as np
import my_py_module

in = np.array([1, 2, 3])
out = my_py_module.square(in)
```

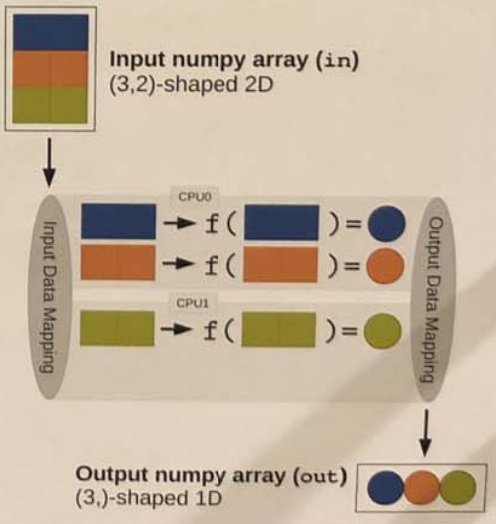
```
print(out)
[1 4 9]
```

The C++ function will be called for every entry in the given input numpy array. This also works for C++ class member functions:

```
bp::class_<...>(...).def(bn::dstream::method(...));
```

Schematic Data Flow

In this schematic the function `f` operates on the second axis and iterates over the first axis of the input array. For the iteration numpy broadcasting rules apply.



Multi-Threading

BoostNumpy supports multi-threading via the C++ pthread library. By exposing a C++ function to Python using the `bn::dstream::allow_threads()` option, the Python function gets the additional optional keyword argument `nthreads=1`. Hence, calculations can be distributed over several CPU cores.

Conclusion

BoostNumpy allows to operate on numpy arrays within C++ in an efficient, transparent, and easy way. Hence, copying of data between Python and C++ becomes unnecessary and calculations can be performed on the data directly.

References

[1] D. Abrahams and R. W. Grosse-Kunstleve, *Building Hybrid Systems with Boost.Python*, C/C++ Users Journal, (July 2003).
[2] S. van der Walt, S. C. Colbert and G. Varoquaux, *The NumPy Array: A Structure for Efficient Numerical Computation*, Computing in Science & Engineering, **13**, 22-30 (2011).
[3] https://github.com/ndarray/Boost.NumPy

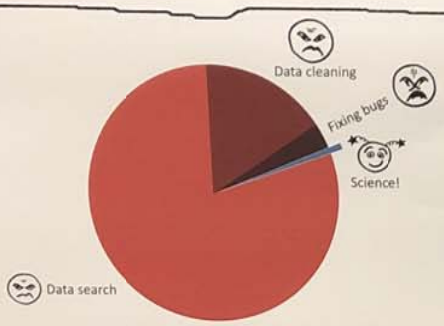
https://github.com/martwo/BoostNumpy/

Small Problems with Big Data in Astronomy

Oleksandra Razim[1], Kseniia Sysoliatina[2]

[1]Department of Physics, Strada Vicinale Cupa Cintia, 21, 80126, University Federico II, Napoli, Italy,

[2]Astronomisches Rechen-Institut, Mönchhofstr. 12-14, 69120 Heidelberg, Germany

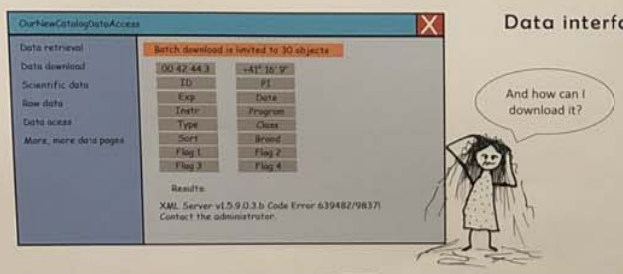


Pareto principle states that 20% of work takes 80% of time.
Data scientists say that data preparation takes 80% of their time.
For astroinformatics it is common that data retrieval takes 80% of our time. It can require weeks and months to find the right catalog, then to select only the data you need, then to find out what are the meanings, corrections and errors of every column. Often it is almost impossible without the help of someone who already worked with these data.
Then you have data preparation,
and then you have actual work.

Catalog search

Problem	Solution
Multitude of non-unified and disconnected data search engines	Virtual Observatory
No "catalog of catalogs"	
Uninformative catalog names	Standartization
No tradition on where to give catalog link in the papers	
No guidelines on where and how to upload your own catalogs	Guidelines and templates similar of those for publications

Data interface



Problem	Solution
Complicated web-sites without site maps	User scenarios during development phase
Search interfaces with multitude of fields, but not the one you need	ADQL
	Asynchronous download
404 Errors	Automated feedback forms

Formatting

Problem	Solution
FormatsZOO: CSV, FITS, TXT, HTML, DAT, CAT...	FITS
ColNamsZOO (Fig. 1)	Naming guidelines
Different NaN values, not specified in the header	"Read tests" in most popular instruments, e.g. TopCat, Python, IDL
Numerical columns read as strings	
Unreadable symbols in column names	
Meaningless column names, e.g. Col1, Col2, Col3	
Sudden renaming of columns in a new data releases	

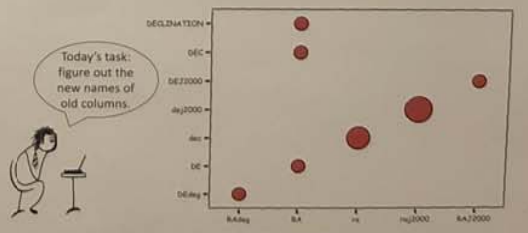
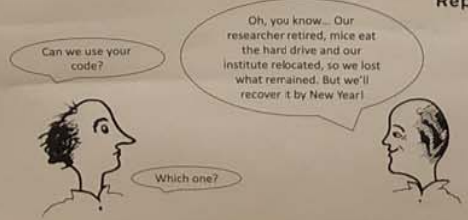


Fig. 1. ColNamesZOO of the of the 12 most popular astrometric, spectroscopic and photometric stellar catalogs

Reproducibility



Problem	Solution
No code and data published	"Better bad code than no code" policy
Plots/diagrams without source tables	Git
	Examples of typical user scripts

Contacts

Alex Razim: shr.razim@gmail.com

Kseniia Sysoliatina: sysoliatina@uni-heidelberg.de

This work was supported by Sonderforschungsbereich SFB 881'The Milky Way System' of the German Research Foundation (DFG).

This project has received financial support from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 721463 to the SUNDIAL ITN network.