



Leibniz-Institut für  
Astrophysik Potsdam

# colab : Unterstützung für datenzentrierte Prozesse

Harry Enke & EScience @ AIP


# Softwareunterstützung für datenzentrierte Prozesse

Neben Werkzeugen zum Datenmanagement (RDMO) und Datenpublikation (Daiquiri)





- betreibt das AIP eine Reihe von Collaborative Research Environments (CRE)
  - zur kollaborativen Arbeit an grossen Datensammlungen (Kosmologische Simulationen)
    - [www.clues-project.org](http://www.clues-project.org)
  - zur kollaborativen Arbeit an Beobachtungsdaten (MUSE, GREGOR)
    - Daten-Center-Ring mit gemeinsamen Storage, Execution-Engine und Metadatenbank
    - Storage und Analysis Facility für GREGOR
- Die CRE haben jeweils sehr spezifische Workflows und Software-Tools, und lang andauernde Mitgliedschaft
- Damit nicht abgedeckt sind dabei kleine Arbeitsgruppen oder Ad Hoc Zusammenschlüsse
  - z.B. um gemeinsam Machine-Learning Anwendungen auf Daten zu entwickeln
  - temporäre Tasks wie die Vorbereitung von Datenreleases zu organisieren

# Softwareunterstützung für datenzentrierte Prozesse: colab.aip.de









 In case of any questions or problems, *do not hesitate* to create a [support ticket](#).  
We want to know if anything is broken!

## Help & Support

-  [How to teach a course with AIP](#)
-  [AIP documentation](#)
-  [Docker image for offline usage](#)
-  [Courses using AIP](#)

## Connect


-  [Mailing list](#)
-  [News and updates on our blog](#)
-  [Follow @co\\_calc on twitter](#)
-  [Like our facebook page](#)
-  [+1 our Google+ page](#)
-  [GitHub](#) — source code, bugs and issues


## Softwareunterstützung für datenzentrierte Prozesse: colab.aip.de






- Colab ist die AIP-Installation der OpenSource Software CoCalc (mit einigen Anpassungen)
- Colab bietet
  - eine Webapplikation mit Nutzermanagement als Interface zu einem Server (pool), derzeit Virtuelle Maschine in OVirt
  - die Webapplikation interagiert mit dem Server
    - stellt (virtualisierte) Umgebungen bereit
      - Kernel (wahlweise) ,es gibt auch weitere:
        - Python 2.7/ 3.5 / 3.6
        - R
        - Sagemath
        - Latex
      - Kernel können erweitert werden nach Bedarf durch Installation von Modulen und Paketen in der individuellen Umgebung
    - über die virtualisierten Umgebungen können Ressourcen des Servers für Computing, Grafik und wissenschaftliche Textbearbeitung kollaborativ genutzt werden

# Softwareunterstützung für datenzentrierte Prozesse: colab.aip.de

**Projects**  Deleted  Hidden

Search for projects... 

 Create new project...

<b>RAVE DR6</b>	3 minutes ago	No Description	 Running
<b>Gaia@AIP</b>	4 minutes ago	This is the shared workspace for everyone working on the Gaia data at AIP.	 Running
<b>APPLAUSE_DR3</b>	11 minutes ago	No Description	 Running
<b>DR6proto</b>	2 days ago	No Description	 Running
<b>Python club</b>	3 days ago	Python club meets monthly at AIP and is a discussion forum for everything concerning python programming language.	 Running

Liste der Mitglieder des jeweiligen Projekts

# Softwareunterstützung für datenzentrierte Prozesse: colab.aip.de

Beispiel:

- Vorbereitung eines Datenreleases
  - RAVE {RAdial Velocity Experiment}
  - Ca. 500000 Sternspektren
- Aufgaben:
  - Verschiedene, aufeinander aufbauende Pipelines zur Berechnung von
    - Radialgeschwindigkeiten,
    - Sterntemperaturen, Oberflächenschwerkraft, Metallizität
  - Verifikation der Ergebnisse der Pipelines
    - Statistische Analyse
    - Vergleich mit Sternpopulationen (HRD)
    - Vergleich mit anderen Katalogen
  - Publikation der Datenreduktionen/Analysen
    - Tabellen (Datenbank)
    - Spektren (Dateien im FITS format)

RAVE Survey (<https://www.rave-survey.org/project/gallery/movies/#RAVE-flightmovies>)



# Softwareunterstützung für datenzentrierte Prozesse: colab.aip.de

(Live Colab)

1. File handling (Import / Export / Shell)
2. Jupiter Notebook mit Python 3.6
  - Zusatzmodule installierbar
    - hier astropy, pandas, numpy,
3. Vergleich RAVE/Gaia Daten zeigt eine Gruppe von Sternen, die keine physikalische Deutung haben (live)
4. Mollweidediagramm von Photoplatten
  - (Daten sind aus [www.plate-archive.de](http://www.plate-archive.de) abrufbar, SQL-Statement und Python Code auf Anfrage )
  - Die folgenden 3 Folien wurden nachträglich eingefügt



# colab.aip.de: . Mollweidediagramm von Photoplatten (I)

Files + New Log Find Settings ExposuresPlates-gals\_edit.ipynb 2018-06-12-233342.ipynb

File Edit View Insert Cell Kernel Help

Trusted | Anaconda(Py3.6)

+ ⌘ ↻ ↺ ⬆ ⬇ ↻ ↺ A A ▶ ■ Markdown Save TimeTravel Classical Notebook...

## Import required Python modules

```
In [3]: import matplotlib.pyplot as plt
from matplotlib.patches import Circle, Wedge, Polygon, PathPatch
from matplotlib.path import Path
from matplotlib.collections import PatchCollection, LineCollection
import pandas as pd
import numpy as np

from astropy.wcs import WCS
from astropy.utils.data import get_pkg_data_filename
```

9 minutes ago  
2.245 seconds

## define some functions

```
In [4]: def map_coord(ar):
ard = ar
ra = np.radians(ar.ra)
ra[ra > np.pi] = ra - 2 * np.pi
ard.ra=ra
ard.de = np.radians(ar.de)
ard.fov1= np.radians(ar.fov1)
ard.fov2= np.radians(ar.fov2)
return ard
```

9 minutes ago  
0.002 seconds

```
In [5]: def plot_mwrfov(dr, loc, pth, nx, colx, xtlab, s):
fig = plt.figure(figsize=(16, 8))

if(len(colx) < 1):
ax = fig.add_subplot(111, projection="mollweide")
title=('APLAUSE %s\n%s Archive %s' % (dr, loc, nx))
imgname=('img/%s_archive%s_fov.png' % (dr, nx))
archnam=('s/ar%s_fov.csv' % (pth, nx))
ax.grid()
ax.set_xticklabels(xtlab)
ax.set_title(title);
ar = pd.read_csv(archnam)
ax.set_linewidth=0.2
ard = map_coord(ar)

lines = []
## generate lines_
for i in range(len(ard)):

ax.scatter(ard.ra, ard.de, s=3, c='r', alpha=0.6)
ax.add_collection(LineCollection(lines, linewidths=0.2))

if s == 1:
else:
plt.show()
```

# colab.aip.de: . Mollweidediagramm von Photoplatten (II)

```
In [8]: # Extraktion aus der Datenbank
archnam="csv/ar001_fov.csv"
ar = pd.read_csv(archnam)
# Info über den geladenen Table
ar.info()
ar[:3]
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 8 columns):
exp      150 non-null int64
pid      150 non-null int64
ra       150 non-null float64
de       150 non-null float64
fov1     150 non-null float64
fov2     150 non-null float64
jdmid    150 non-null object
expn     150 non-null int64
dtypes: float64(4), int64(3), object(1)
memory usage: 9.5+ KB
```

	exp	pid	ra	de	fov1	fov2	jdmid	expn
0	13438	19930	337.4605	78.8172	7.57484	7.61019	2418615.39455	1
1	13439	19928	337.4605	78.8172	7.55174	7.60120	2418615.25029	1
2	13440	19929	337.4605	78.8172	7.53457	7.56695	2418615.35775	1

Out[8]:

# colab.aip.de: . Mollweidediagramm von Photoplatten (III)

```
In [6]: dbg=1
xtlab=['14h', '16h', '18h', '20h', '22h', '0h', '2h', '4h', '6h', '8h', '10h']
ham_arch = ['101', '102', '104']
bam_arch = ['201', '202']
pot_arch = ['001']
DR='DR2'
COLR = '#33F4F0'
PTH = 'csv'
```

11 minutes ago  
0.004 seconds

call plot function for Potsdam Archive (150 observed fields from Zeiss Triplett)

```
In [7]: # Potsdam DR2 plots
loc='Potsdam'
for arnu in pot_arch:
    plot_mwrfov(DR, loc, PTH, arnu, COLR, xtlab, 1)
```

/projects/anaconda3/lib/python3.6/site-packages/matplotlib/projections/geo.py:485: RuntimeWarning: invalid value encountered in arcsin  
theta = np.arcsin(y / np.sqrt(2))

11 minutes ago  
3.616 seconds

