

RDMと連携する データ解析サービス

藤原 一毅

オープンサイエンス基盤研究センター

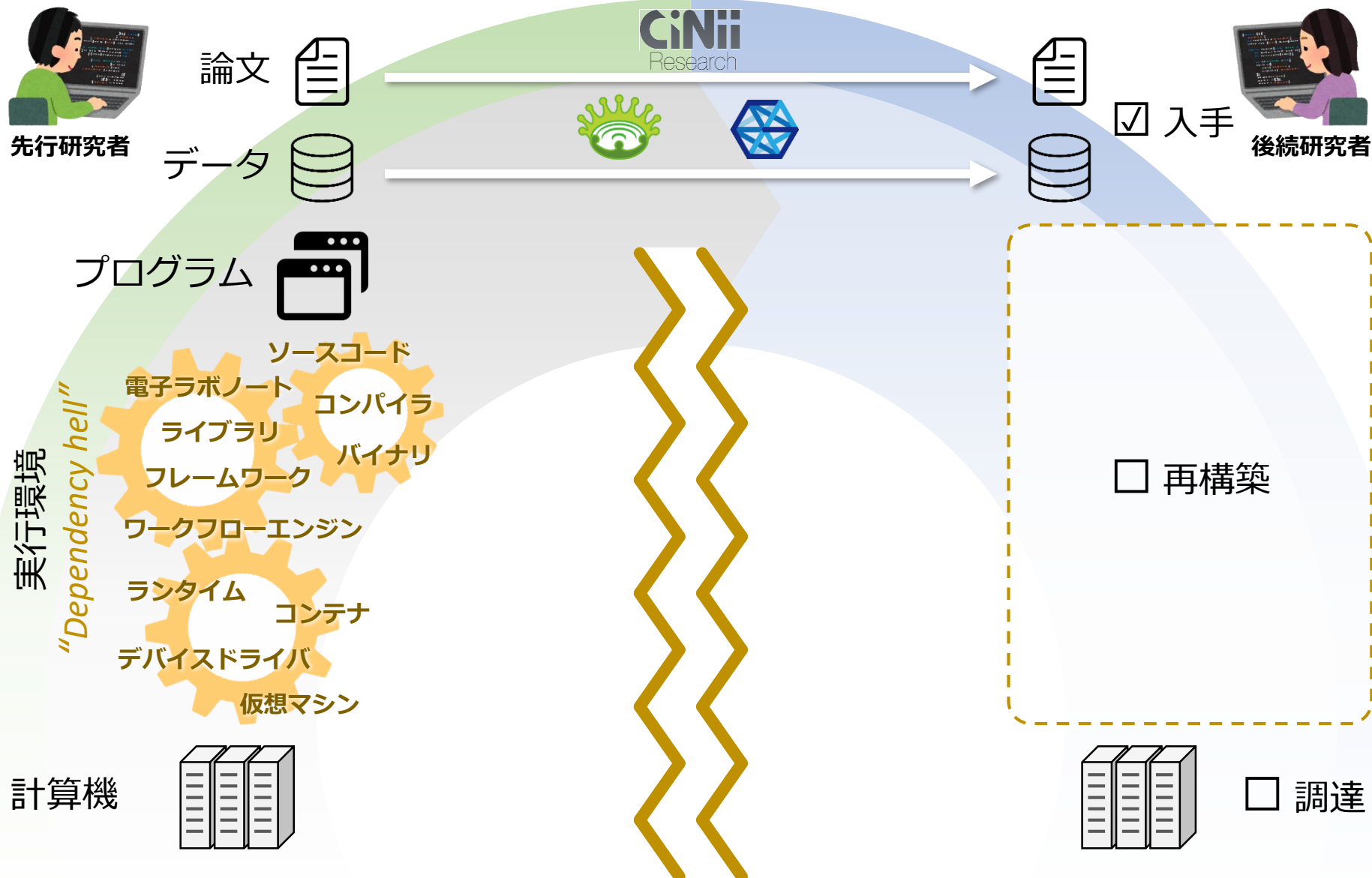
国立情報学研究所

データ + プログラム → 研究成果

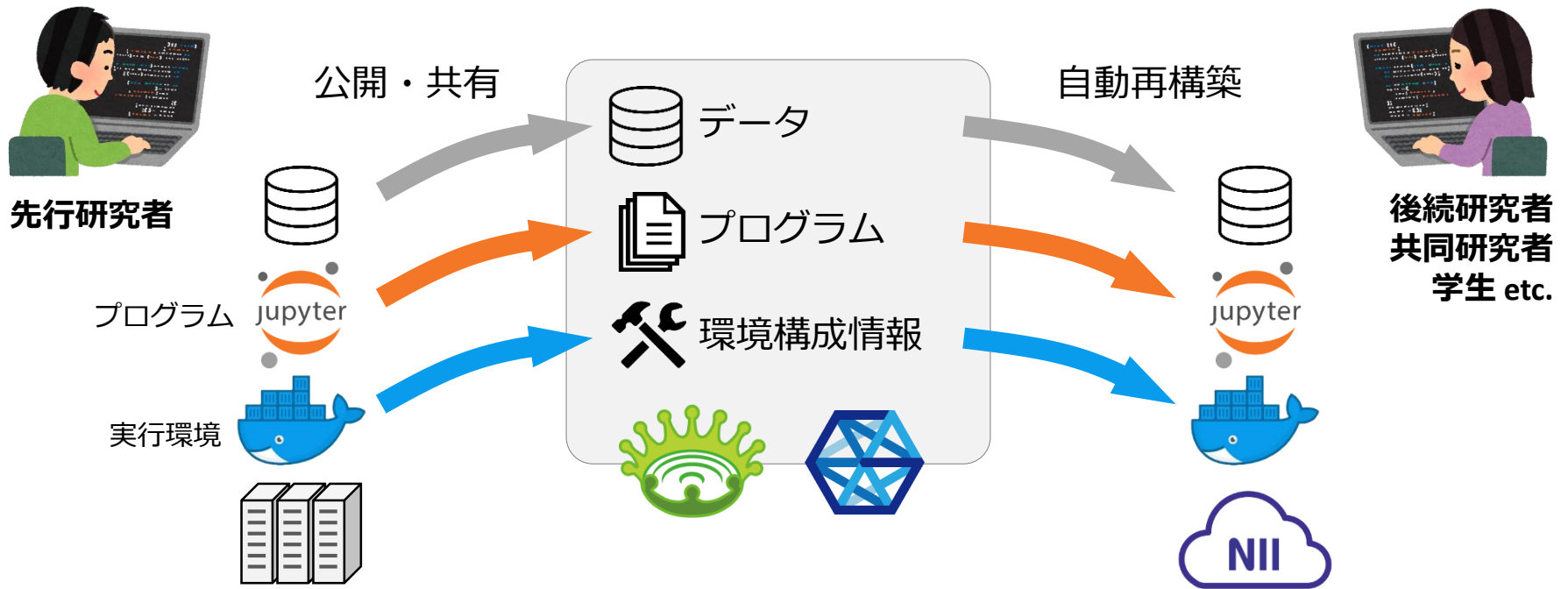


- データから知見を得るには、**プログラム**とその**実行環境**が不可欠
 - いまどき研究者がデータを直接眺めるわけではない
- 研究の再現性を確保するために、**ソフトウェア**をデータや文献と同様、研究活動の一級市民として扱う必要がある

オープンサイエンスの環をつなげたい



研究のボタンタッチを簡単に



- 先行研究者は、データ・プログラム・環境構成情報をまとめてアップロードする
 - 一般公開なら WEKO / JAIRO Cloud へ、内部共有なら GakuNin RDM へ
- 後続研究者は、個人用のコンテナ環境上に実行環境を再構築し、プログラムを実行する
- 本サービスは、このコンテナ再構築作業を自動化する

🔧 環境構成情報

● 例: environment.yml

```
name: ipyvolume
channels:
  - conda-forge
dependencies:
  - nodejs
  - pip:
    - traitletypes
    - traitlets
    - ipywidgets>=7.4
    - Pillow
    - scipy
    - numpy
    - scikit-image>=0.13
    - requests
    - ipywebbrtc>=0.4
    - pythreejs>=1.0
    - matplotlib
    - jupyterlab>=0.34
    - notebook>=5.3
```

<https://github.com/maartenbreddels/ipyvolume/blob/master/binder/environment.yml>

● 例: Dockerfile

```
FROM jupyter/scipy-notebook:latest

# Install TF libraries
RUN wget
  https://storage.googleapis.com/tensorflow/libtensorflow/libtensorflow-
  cpu-linux-x86_64-1.14.0.tar.gz
RUN tar -C /usr/local -xzf libtensorflow-cpu-linux-x86_64-1.14.0.tar.gz
RUN rm libtensorflow-cpu-linux-x86_64-1.14.0.tar.gz
RUN ldconfig
# Install .NET Core SDK
ENV DOTNET_SDK_VERSION 3.1.101
RUN curl -SL --output dotnet.tar.gz
  https://dotnetcli.blob.core.windows.net/dotnet/Sdk/$DOTNET_SDK_VERSION/
  dotnet-sdk-$DOTNET_SDK_VERSION-linux-x64.tar.gz && mkdir -p
  /usr/share/dotnet && tar -xzf dotnet.tar.gz -C /usr/share/dotnet && rm
  dotnet.tar.gz && ln -s /usr/share/dotnet/dotnet /usr/bin/dotnet
# Enable detection of running in a container
ENV DOTNET_RUNNING_IN_CONTAINER=true
ENV DOTNET_USE_POLLING_FILE_WATCHER=true
ENV NUGET_XMLDOC_MODE=skip
ENV DOTNET_TRY_CLI_TELEMETRY_OPTOUT=true

# Install Microsoft.DotNet.Interactive
RUN dotnet tool install -g dotnet-interactive --add-source
  "https://dotnet.myget.org/F/dotnet-try/api/v3/index.json"
ENV PATH="${PATH}:${HOME}/.dotnet/tools"
RUN echo "$PATH"
# Install kernel specs
RUN dotnet interactive jupyter install
# Enable telemetry once we install jupyter for the image
ENV DOTNET_TRY_CLI_TELEMETRY_OPTOUT=false
# Set root to notebooks
WORKDIR ${HOME}/notebooks/
```

<https://github.com/javiercp/BinderTF.NET/blob/master/Dockerfile>

- コンテナに pip や Conda で任意のパッケージをインストールする
- コンテナ構築時に任意のコマンドを実行することもできる

📄 プログラム

● 例: Jupyter Notebook

```
In [ ]: import ipyvolumes as ipv
import numpy as np
from matplotlib.pyplot import cm

In [ ]: # download the dataset
!wget -q https://www.dropbox.com/s/eqht79b7j4jqit2/petct.npz?dl=1 -O petct.npz

We show a CT scan and overlay the PET scan

In [ ]: full_scan = {k: v.swapaxes(0, 1)[::-1] for k,v in np.load('petct.npz').items()}
print(list(full_scan.keys()))

In [ ]: table_ct = cm.gray_r(np.linspace(0, 1, 255))
table_ct[:,50, 3] = 0 # make the lower values transparent
table_ct[50:, 3] = np.linspace(0, 0.05, table_ct[50:].shape[0])
tf_ct = ipv.TransferFunction(rgba=table_ct)

In [ ]: ct_vol = ipv.quickvolshow(full_scan['ct_data'],
                                tf=tf_ct, lighting=False,
                                data_min=-1000, data_max=1000)
ct_vol
```

Zoom

Zoom in by clicking the magnifying icon, or keep the alt/option key pressed. After zooming in, the higher resolution version displayed.

Multivolume rendering

Since version 0.5, ipyvolumes supports multivolume rendering, so we can render two volumetric datasets at the same time

```
In [ ]: table_pet = cm.hot(np.linspace(0, 1, 255))
table_pet[:,50, 3] = 0 # make the lower values transparent
table_pet[50:, 3] = np.linspace(0, 1, table_pet[50:].shape[0])
tf_pet = ipv.TransferFunction(rgba=table_pet)
```

<https://github.com/maartenbreddels/ipyvolumes/blob/master/notebooks/multi-volume.ipynb>

● 例: Python

```
10 import warnings
11 import numpy as np
12 from numpy import cos, sin, pi
13
14 try:
15     import scipy.ndimage
16     import scipy.special
17 except:
18     pass # it's ok, it's not crucial
19 # __all__ = ["example_ylm"]
20
21
22 def xyz(shape=128, limits=[-3, 3], spherical=False, sparse=True, centers=False):
23     dim = 3
24     try:
25         shape[0]
26     except:
27         shape = [shape] * dim
28     try:
29         limits[0][0] # pylint: disable=unsubscriptable-object
30     except:
31         limits = [limits] * dim
32     if centers:
33         v = [
34             slice(vmin + (vmax - vmin) / float(N) / 2, vmax - (vmax - vmin) / float(N) / 4, (vmax - vmin) / float(N))
35             for (vmin, vmax), N in zip(limits, shape)
36         ]
37     else:
38         v = [
39             slice(vmin, vmax + (vmax - vmin) / float(N) / 2, (vmax - vmin) / float(N - 1))
40             for (vmin, vmax), N in zip(limits, shape)
41         ]
42     if sparse:
43         x, y, z = np.ogrid.__getitem__(v)
44     else:
45         x, y, z = np.mgrid.__getitem__(v)
```

<https://github.com/maartenbreddels/ipyvolumes/blob/master/ipyvolumes/examples.py>

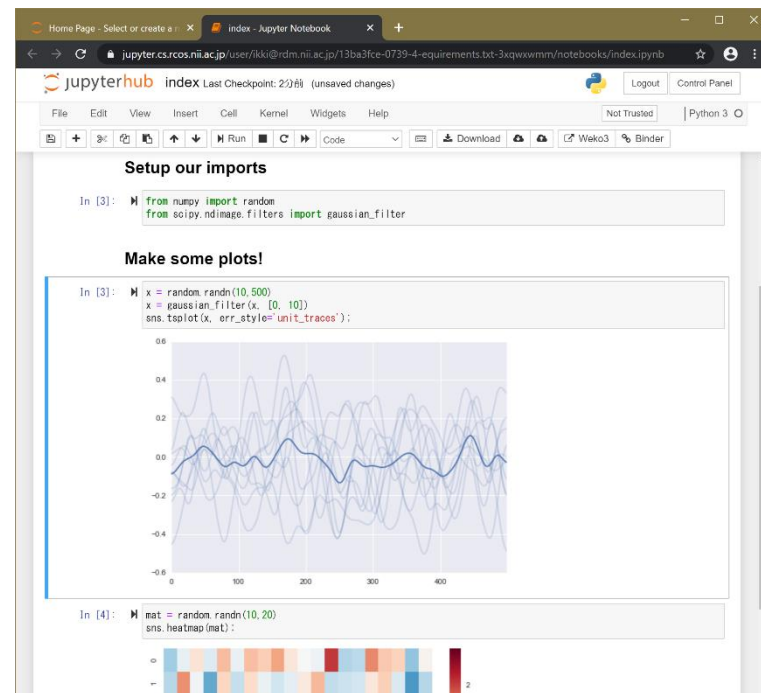
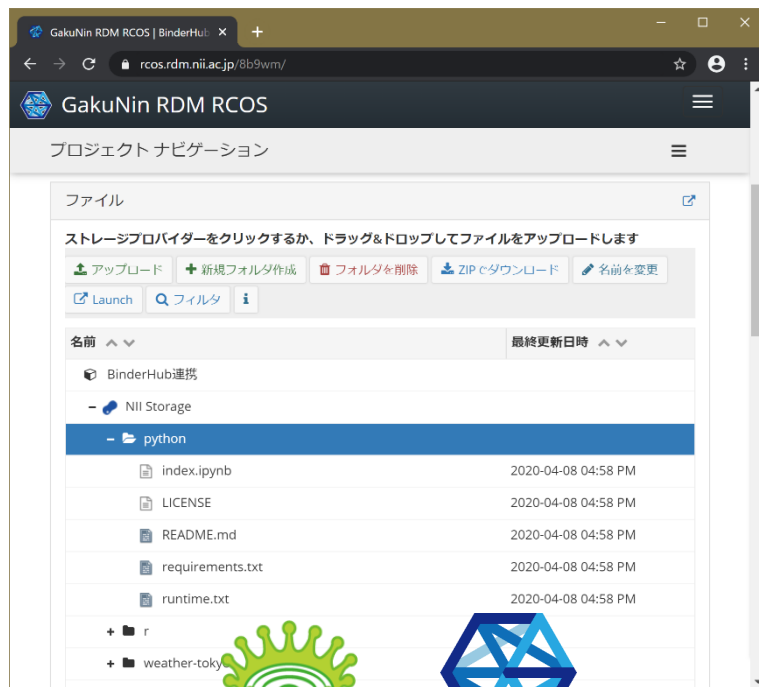
- 基本は Jupyter Notebook
- 構築されたコンテナ上で実行可能な任意の言語を使える

再構築手順

1 フォルダを選んで **Launch** を押す

2 3分待つ ※

3 起動したJupyter環境でプログラムを実行する



※待ち時間は再構築内容により前後します。初回はOAuth認証が必要です

ユースケース

研究公正

- 研究環境を別の研究者が再現し、成果を検証する



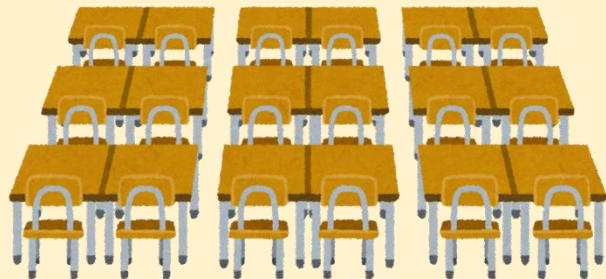
二次研究

- 研究環境を別の研究者が再現し、新たな研究を開始する



教育・学習

- 授業や研修の教材を受講者が各々再現し、学習する



引き継ぎ

- 先輩の研究環境を後輩が再現し、研究を継続する



類似サービスとの比較

	本サービス (構想中)	Google Colab	mybinder.org	GESIS Notebooks	Pangeo's BinderHub	Codalab Worksheets
ターゲット	汎用	主に深層学習	汎用	社会科学	地球科学	深層学習
提供元	NII (日・学術機関)	Google (米・民間企業)	Project Jupyter (任意団体)	GESIS (独・学術機関)	Pangeo (米・学術団体)	Microsoft (米・民間企業)
アカウント	学認	Google	不要	GESIS	GitHub	Codalab
対応言語	Python, R, Julia (需要により追加)	Python, R, Julia, Swift	Python, R, Julia	Python, R, Julia	Python	Python
対応リポジトリ	WEKO3, GakuNin RDM, GitHub, Gist, GitLab, Zenodo, Figshare, Hydroshare, Dataverse	Google Drive, GitHub	GitHub, Gist, GitLab, Zenodo, Figshare, Hydroshare, Dataverse	GitHub, Gist, GitLab, Zenodo, Figshare, Hydroshare, Dataverse	GitHub, Gist, GitLab, Zenodo	?
メモリ CPU ストレージ	(これから決める)	13GB 2コア 40GB	2GB 1コア ?	32GB 2コア 10GB	8GB 4コア ?	?
タイムアウト	永続	90分 / 12時間	10分	40分	11分 / 3時間	?
インフラ	オンプレ	Google	Google, OVH, Turing Institute	オンプレ	Google	Microsoft

オープンサイエンスの環をつなげよう

- NII では、研究再現性をサポートするため、RDMと連携するデータ解析サービスを構想しています
- 先行研究者が開発したプログラムとその実行環境を後続研究者が容易に再構築できるようになります
- 今後の開発について、将来のユーザーとなる皆さまのアイデアをお聞かせください

