

データガバナンス機能

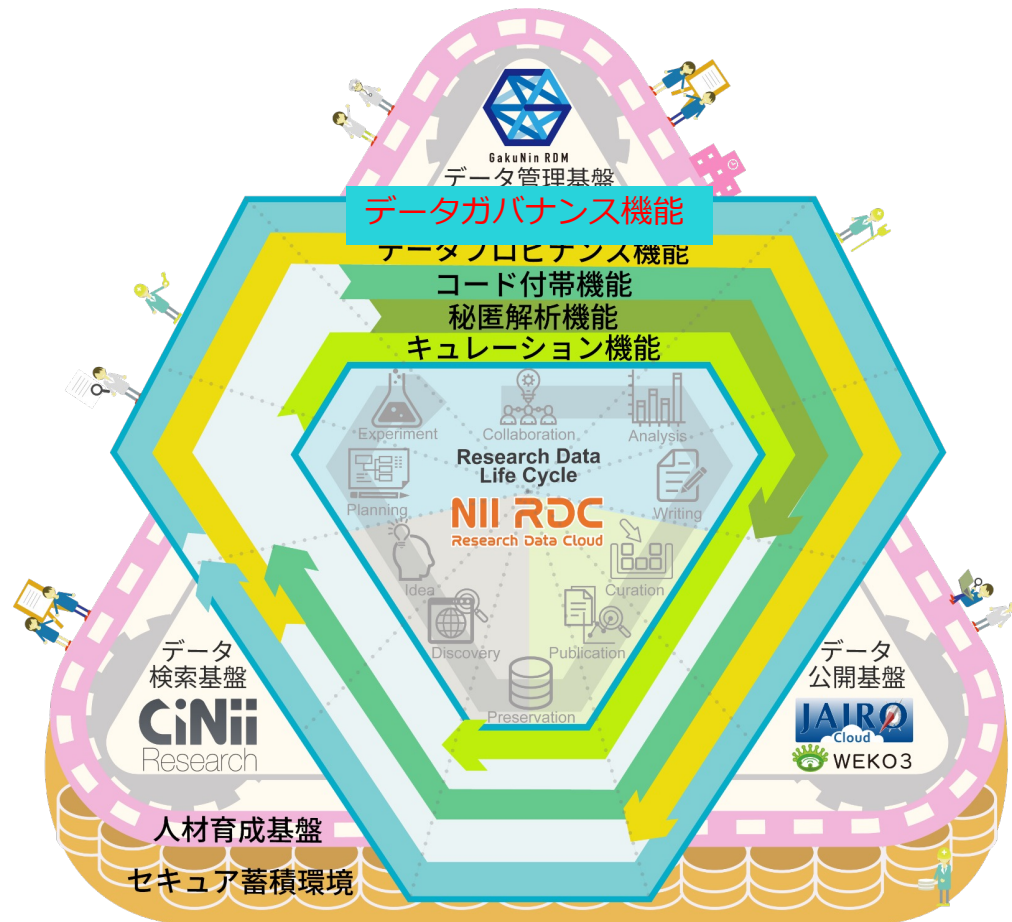
国立情報学研究所
オープンサイエンス基盤研究センター
データガバナンス機能担当
横山重俊

2022年6月1日

1. データガバナンス機能とは

データガバナンス機能の位置付け

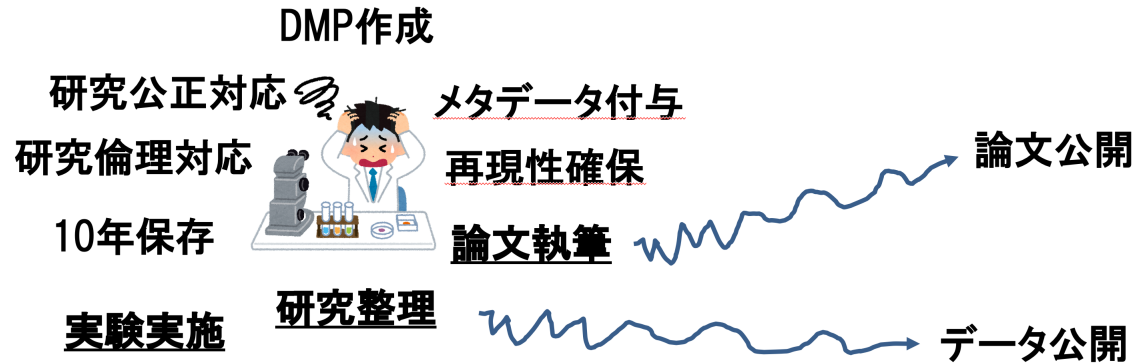
研究データ管理品質を向上させるために
ワークフロー、モニタリング、研究記録機能を提供



RDC (Research Data Cloud) の必要性

AS IS

研究者の
努力に依存

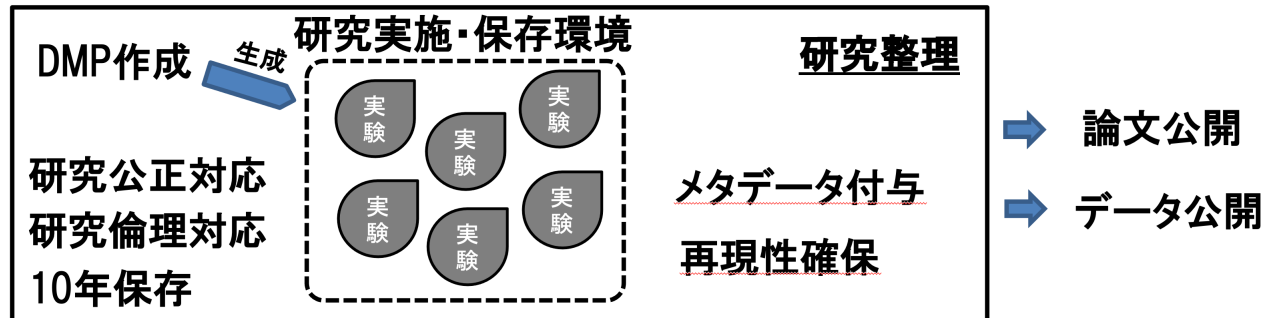


TO BE

研究管理
基盤による
支援

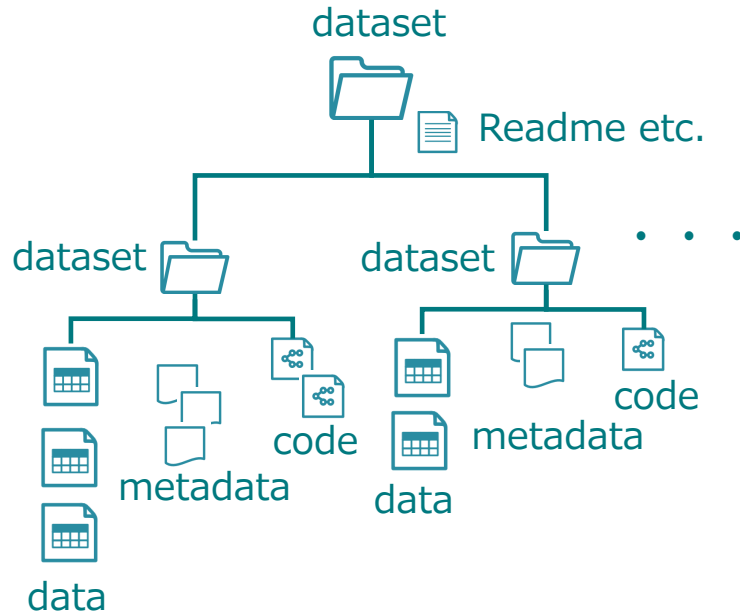


研究管理基盤



研究データ

研究データの構成



研究データの例

[goltschlab / Schreyer_and_Goltsch_2021_salamander_retinal_bipolar_cell_recordings](#)

Data sets accompanying the paper by Schreyer and Goltsch, 2021.

Voltage traces of salamander retinal bipolar cells under visual stimulation.

ファイル	コミット	変更	リリース
Spot	a95e43e3ac	Spot data and code	3ヶ月前
naturalMovies	1f8251376a	Natural movies data and code	3ヶ月前
spatiotemporalWhiteNoise	7f854642cc	Spatiotemporal white noise data and code	3ヶ月前
Documentation_Data_BCR...	518aa7462b	Documentation bipolar cell data	3ヶ月前
LICENSE	c95614b585	Initial commit	4ヶ月前
README.md	8521764916	Updated README	1ヶ月前
datacite.yml	cc8382f88b	Added information for publishing with DataCite	1ヶ月前

README.md

Schreyer_and_Goltsch_2021_salamander_retinal_bipolar_cell_recordings

Membrane potential recordings of salamander retinal bipolar cells under visual stimulation.

Accompanying the paper:

Schreyer HM, Goltsch T (2021) Nonlinear spatial integration in retinal bipolar cells shapes the encoding of natural stimuli. *Neuron* 109:1692-1706.

Pubmed link: <https://pubmed.ncbi.nlm.nih.gov/33798407/>

The repository contains voltage traces of salamander retinal bipolar cells under stimulation with various light patterns on questions of spatial signal integration and encoding of natural stimuli. Furthermore, software code (in Matlab) the stimulus-response relation is supplied.

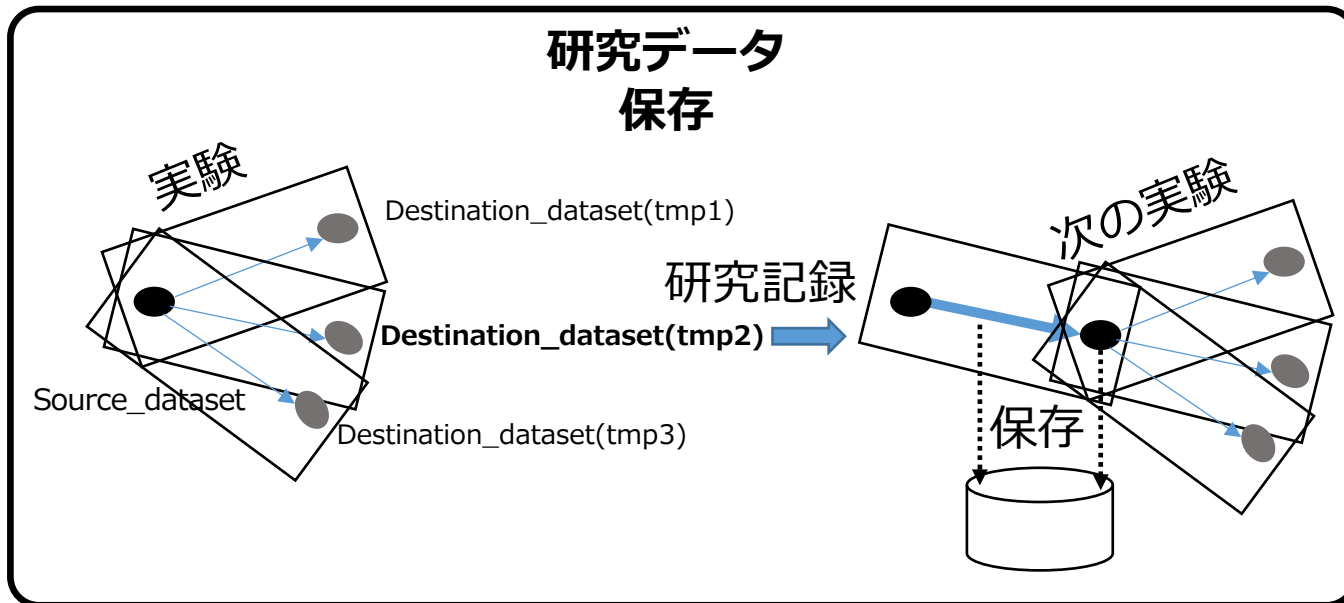
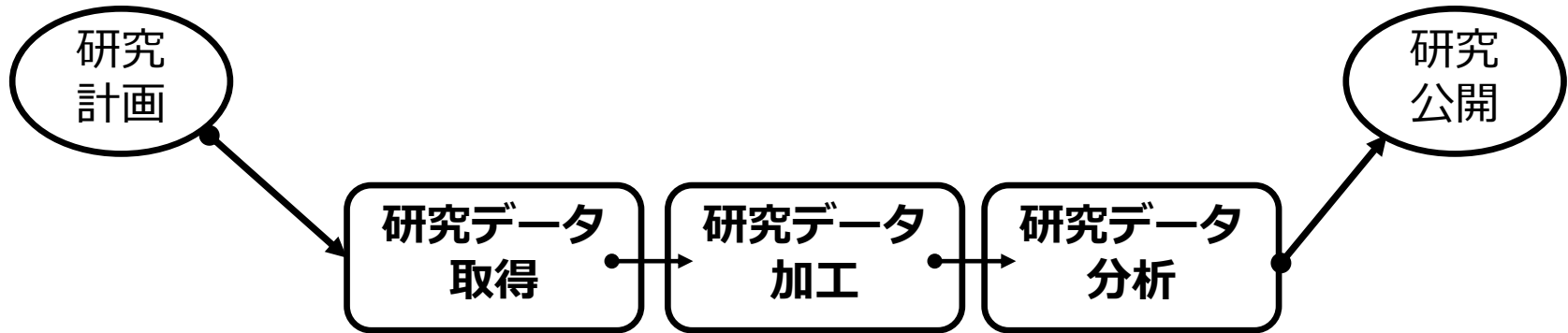
The structure of the repository is explained in "Documentation_Data_BCRRecordings.pdf". The file contains info on the format of the data files, the applied visual stimuli, and the supplied code for analysis.

If you plan to use this data for a publication, please inform us about it and don't forget to cite the original paper (see above) as well as the source of the data.

datacite.yml

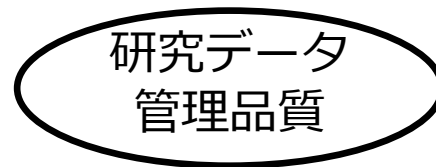
Title	Dataset - Salamander retinal bipolar cell membrane potential measured under visual stimulation
Authors	Schreyer,Helene;Marianne,University Medical Center Gottingen Goltsch,Tim,University Medical Center Gottingen;ORCID:0000-0003-3988-533X
Description	Data sets accompanying Schreyer and Goltsch, Neuron 2021: Nonlinear spatial integration in retinal bipolar cells shapes the encoding of natural stimuli.
License	Creative Commons Attribution-ShareAlike 4.0 International (https://creativecommons.org/licenses/by-sa/4.0/)
References	Schreyer HM, Goltsch T (2021), Nonlinear spatial integration in retinal bipolar cells shapes the encoding of natural stimuli. <i>Neuron</i> 109:1692-1706.

研究データ管理



研究データ管理 と データガバナンスの関係

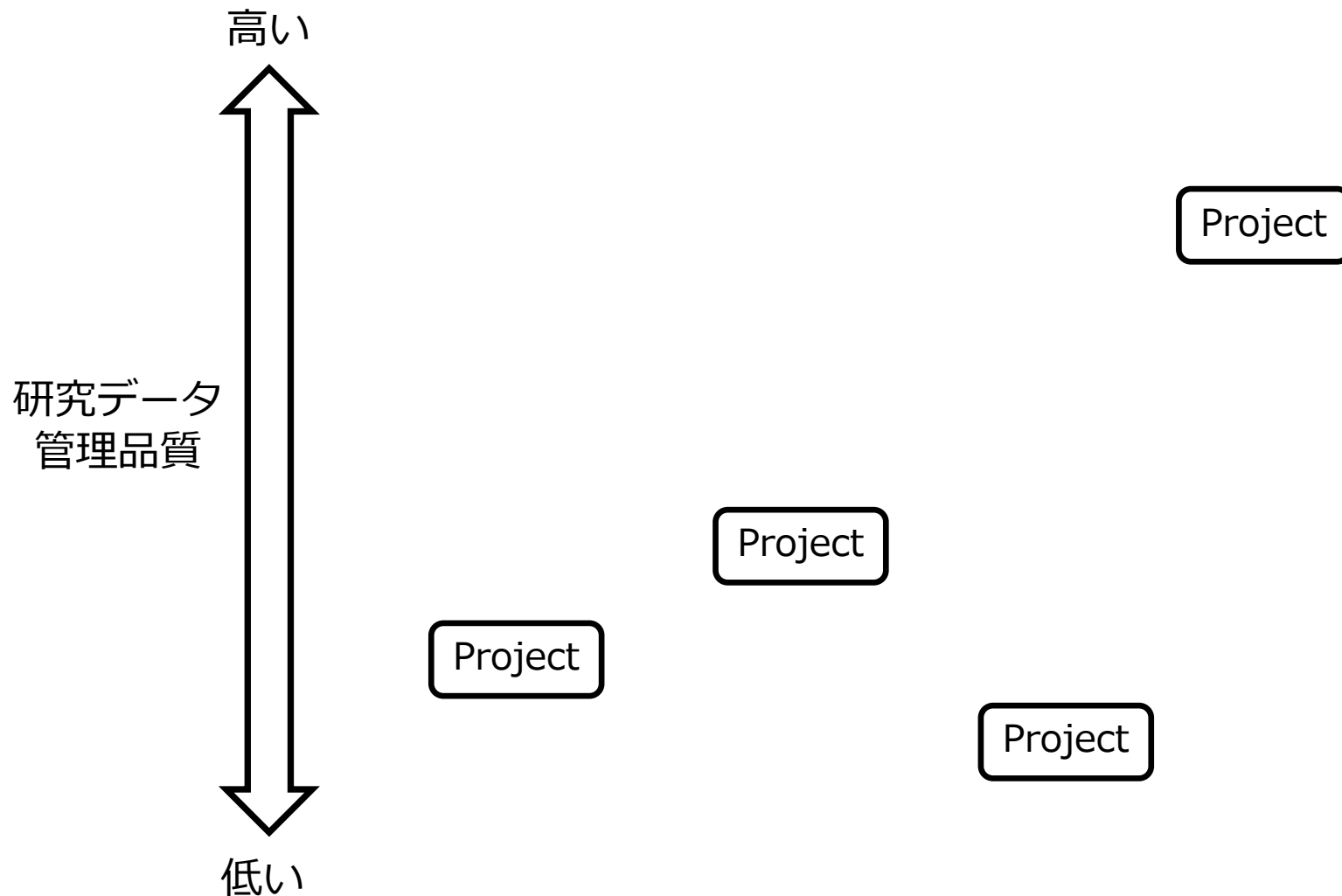
データガバナンス : 研究データ管理品質を守る規律に従う行為



研究データ管理 : 研究データの取得・加工・分析・保存・公開などの行為

仮説 :
研究者や研究グループ自身が、データガバナンスを、
日頃から行うことが、研究データ管理品質の向上に大きく寄与する。
(ガバナンスは監視ではない)

研究データ管理品質のばらつき

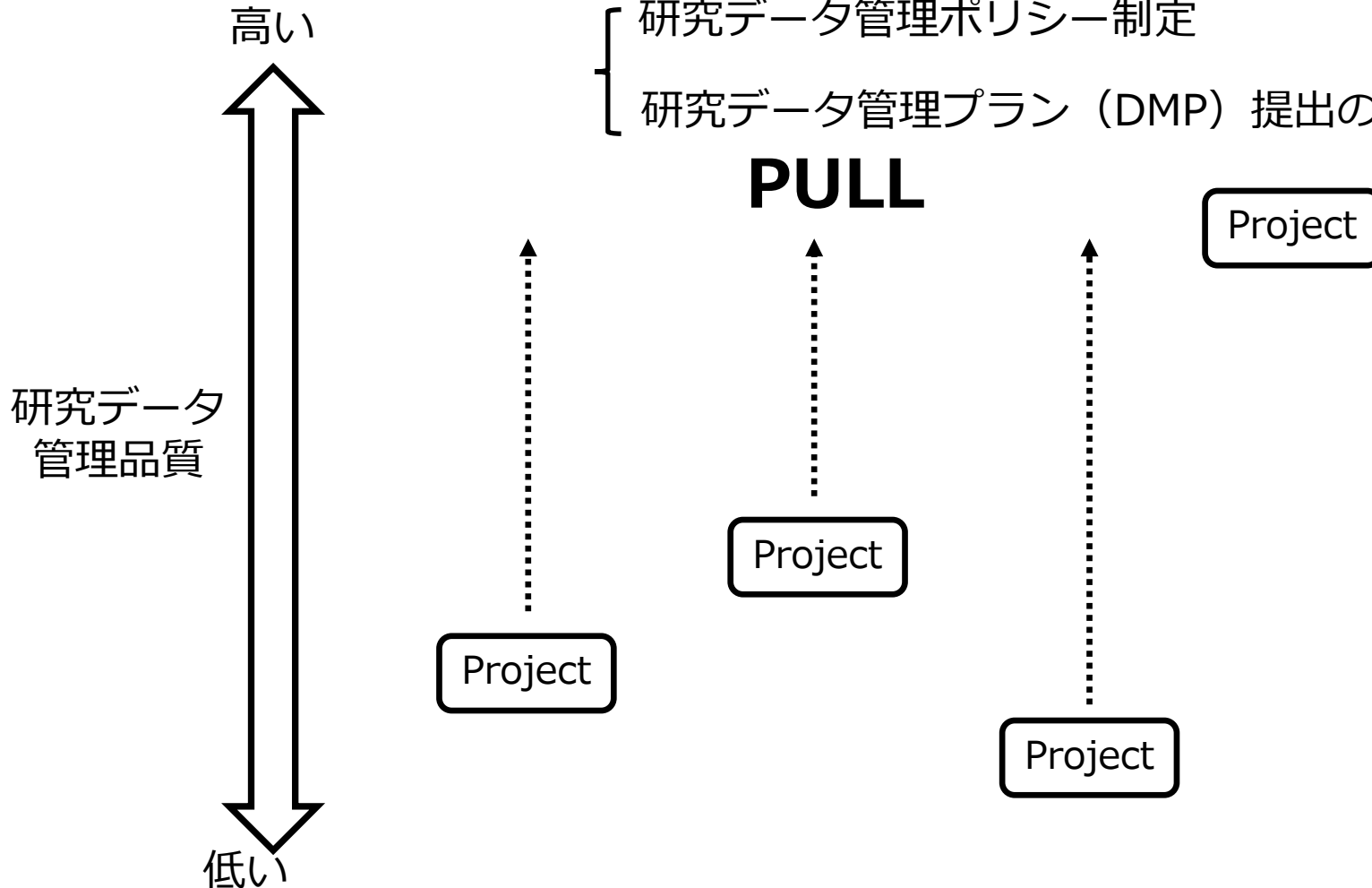


施策の展開による研究データ管理品質向上

データガバナンス施策

- 研究データ管理ポリシー制定
- 研究データ管理プラン（DMP）提出の義務化 など

PULL

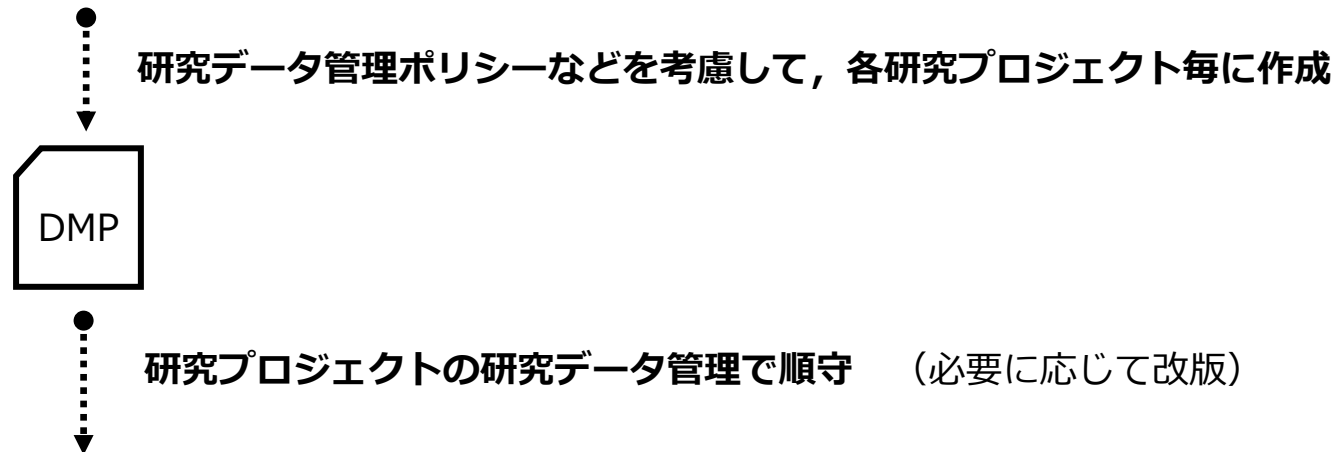


データマネジメントプラン (DMP)

データマネジメントプラン(DMP)は、研究のために収集・作成する研究データの取扱いや整備・保存・公開についての計画を定めた**文書**である。

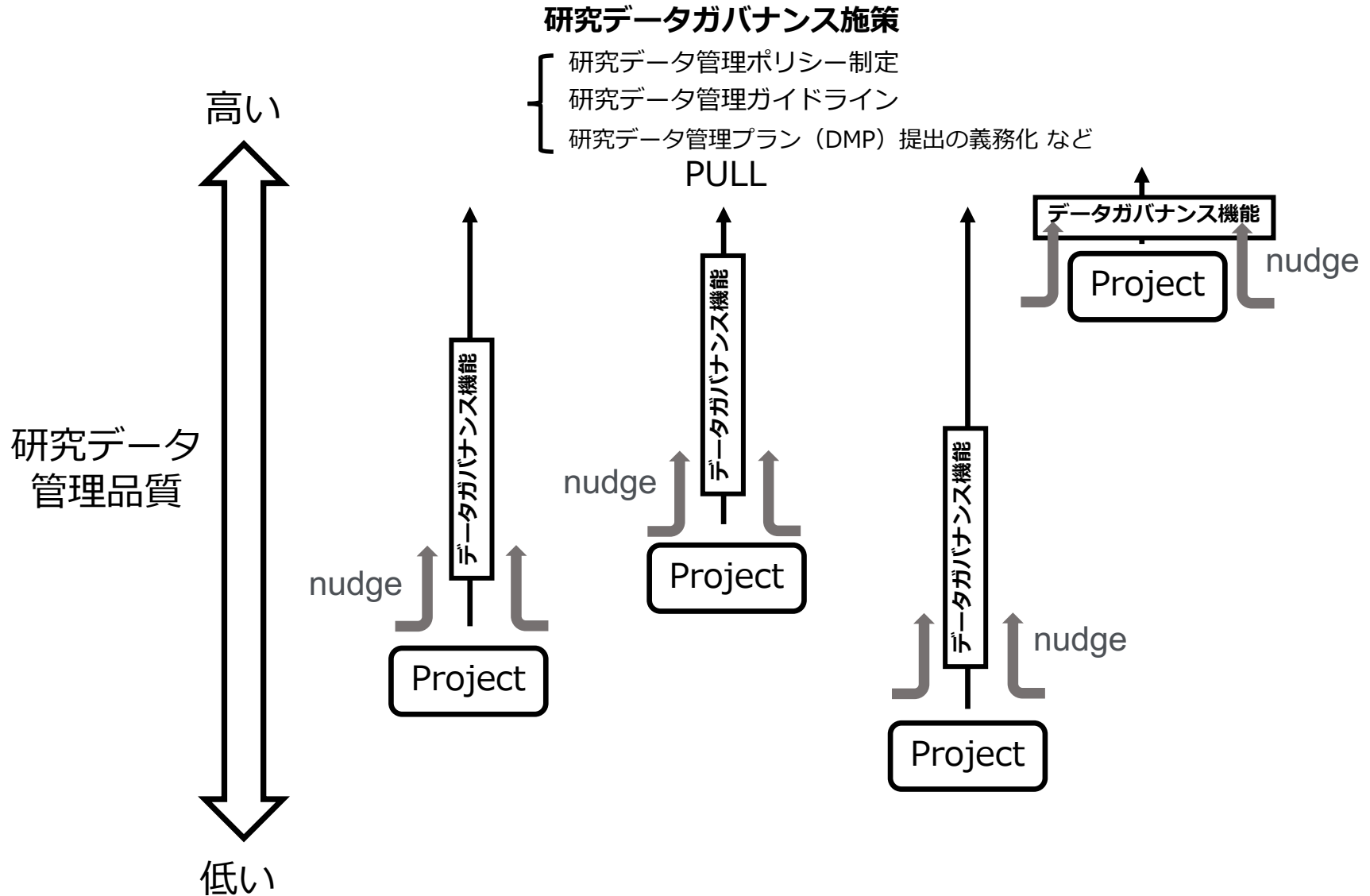
DMPの記述内容は提出先の助成機関によって異なるけれど、データ管理者、データの説明、データの種類、データ量、公開レベルなどが記述される。

研究データガバナンス : 研究データ管理品質を守る規律に従う行為



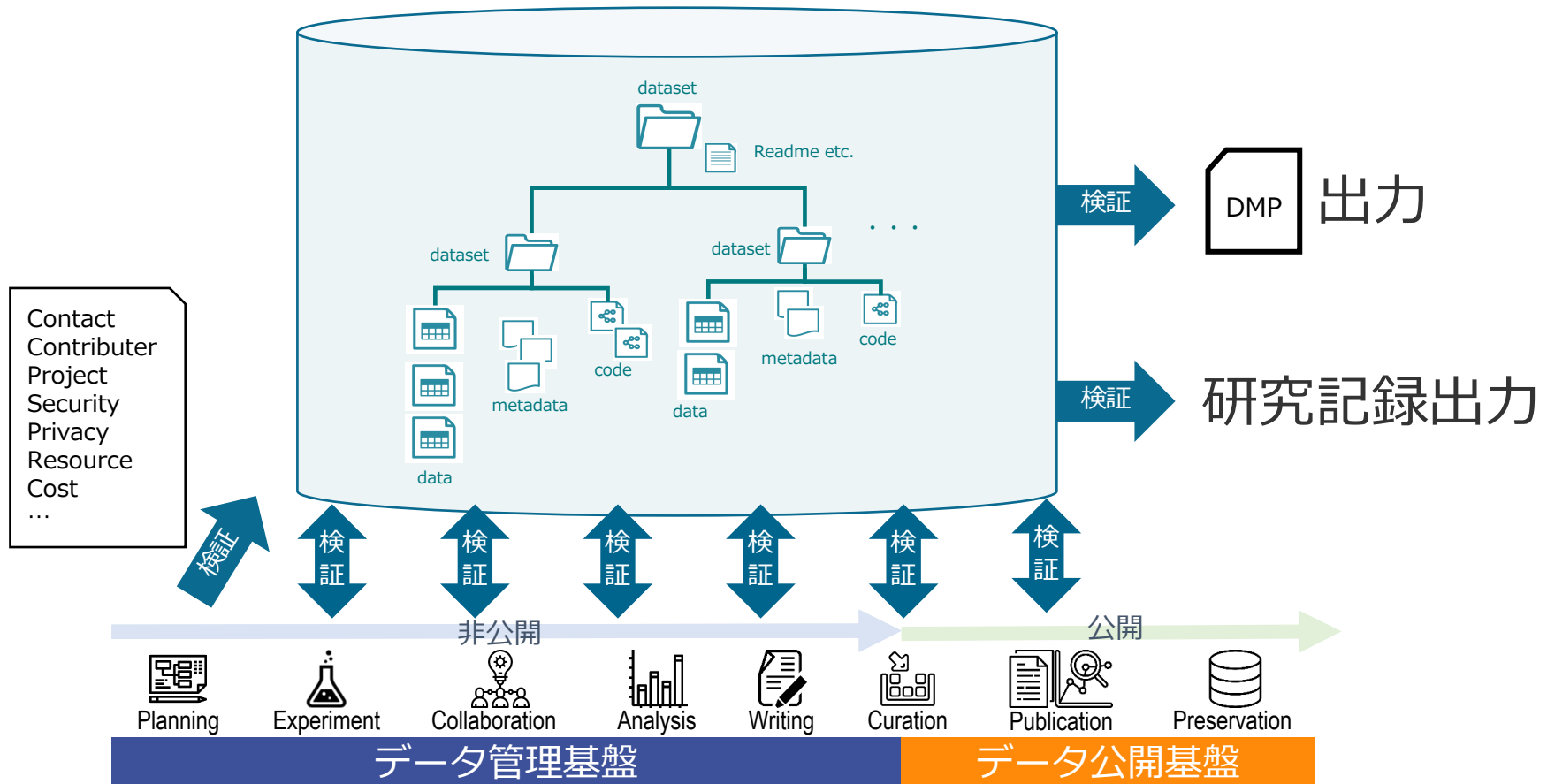
研究データ管理 : 研究データの取得・加工・分析・保存・公開などの行為

データガバナンス機能の役割



「ガバナンスする。」とは

ステート（＝メタデータ・データセット）の変化を検証し、検証プロセスを保存し、提供する、こと



2. プロトタイプシステム

プロトタイプシステムの機能と特徴

機能

研究データ管理計画（DMP）に基づき研究に適した環境を準備し、データ管理品質の維持を支援

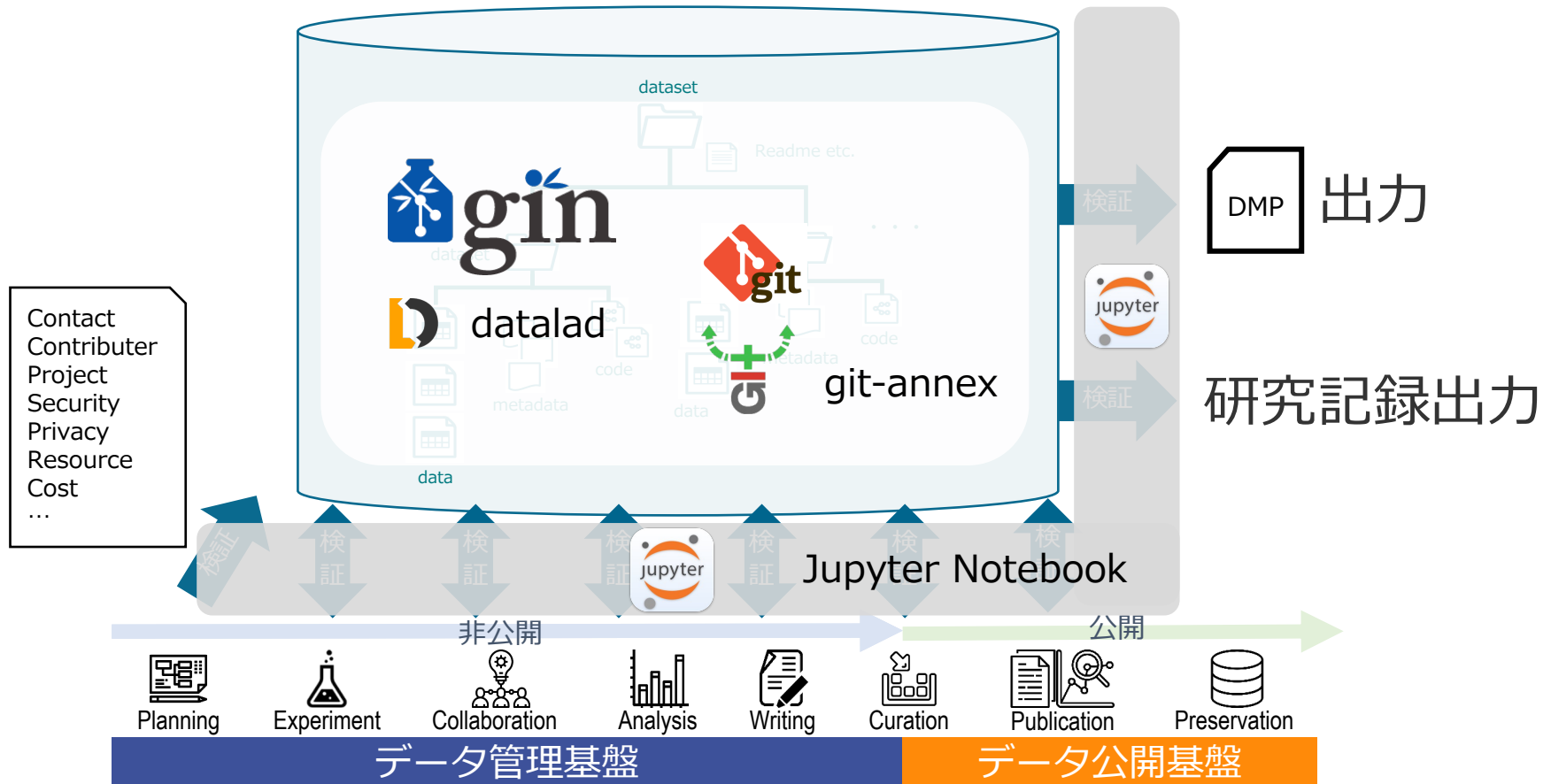
- ① 入力されたDMPから最適なワークフローを生成
- ② ワークフローを実行することで、実験の準備、実験の実施、結果の保存、およびその一連の手順を記録
- ③ 実験で作成されたデータがDMPと整合するかを確認

特徴

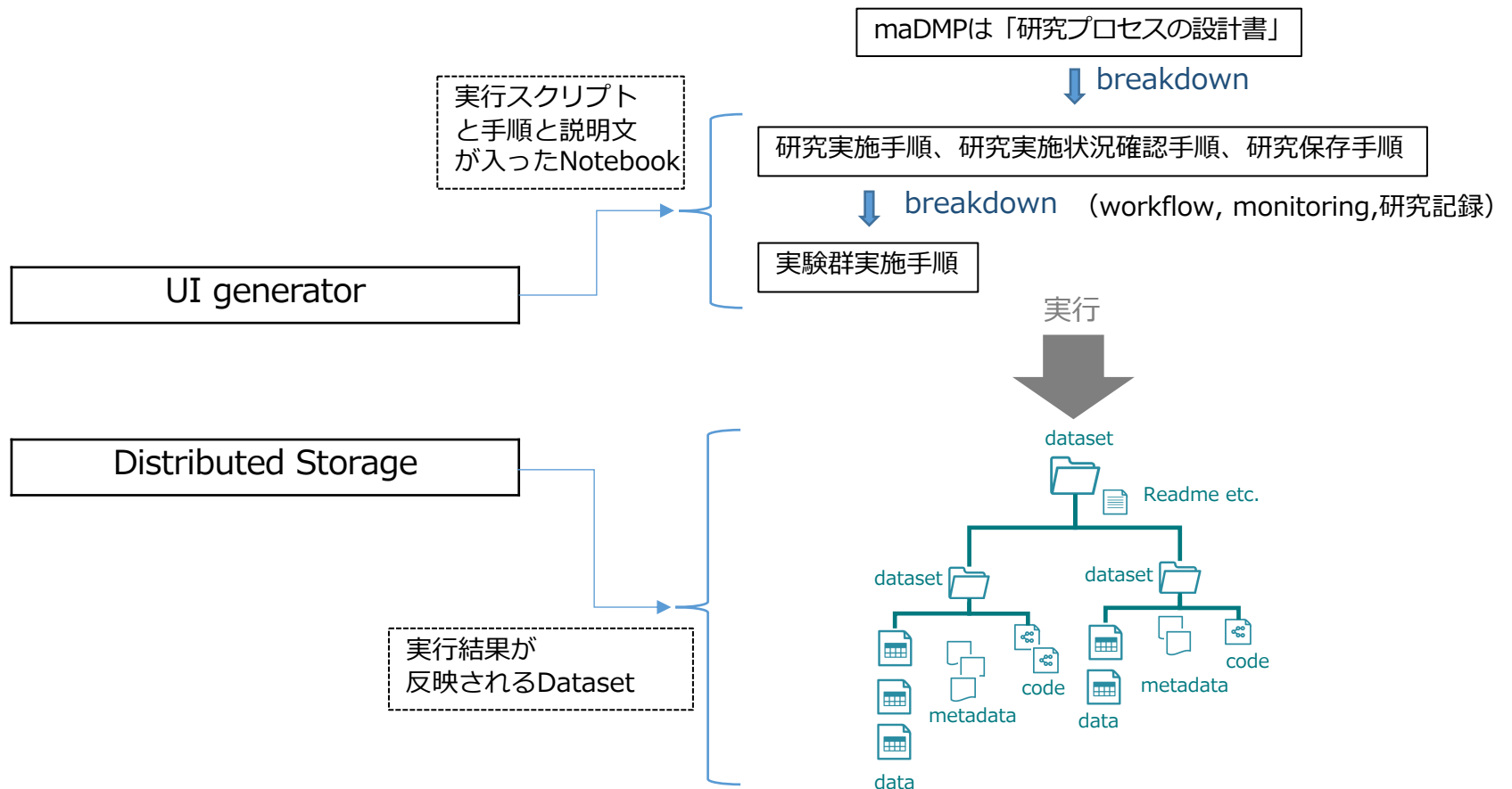
- ① 作成されたDMPからmaDMPを生成し、研究や実験手順を最適な形で展開するワークフローを作成
- ② ワークフローは、データ分析の分野で広く利用されているJupyter Notebook
- ③ ワークフローにしたがって半自動的に実験結果を記録、その記録はGitによるバージョン管理を行うことが可能
- ④ 実験に関するデータや手順、実験結果と記録文書が系統的にパッケージ化

プロトタイプシステムのアーキテクチャ (1/2)

ステート (=メタデータ・データセット) の保存をgit, git-annexを使ったgin/dataladというオープンソースプロダクトで実装、検証プロセスをJupyter Notebookで実現

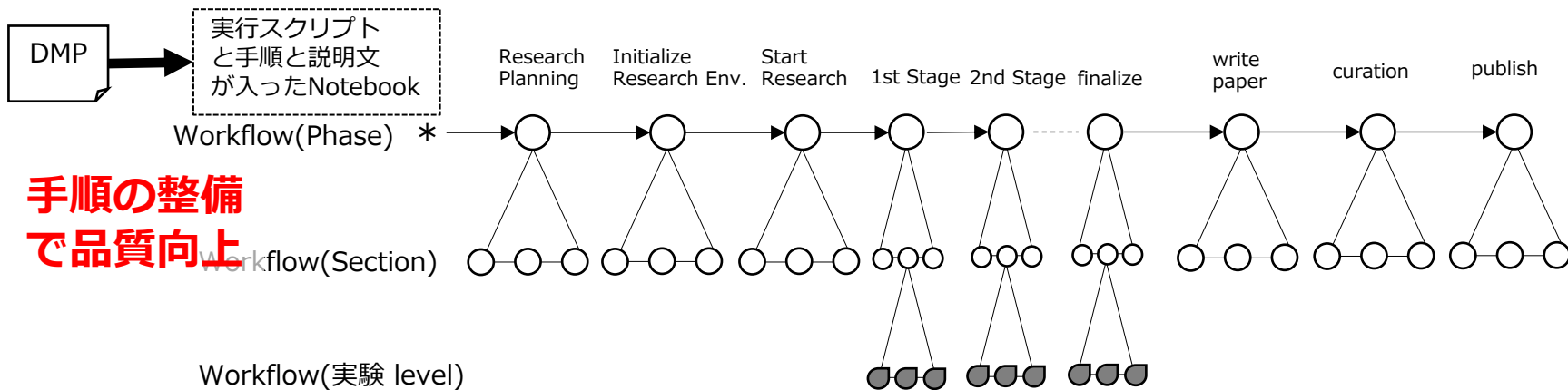


プロトタイプシステムのアーキテクチャ (2/2)

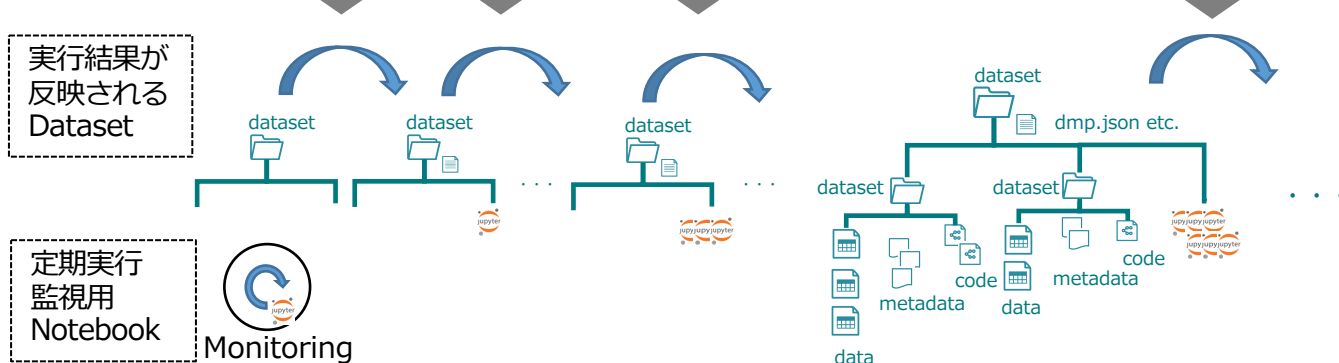


プロトタイプシステムの機能

データガバナンスを実施する実行可能な手順書(Notebook)をDMPから生成



手順の整備
で品質向上



継続監視
で品質向上



研究活動支援範囲

ワークフロー機能の実行準備

研究準備フェーズ

実験フェーズ

実験終了フェーズ

研究終了後

初期セットアップ

- データガバナンスログイン
- 研究リポジリー作成
- DMP作成
 - 日付/タイトル/研究情報
 - データサイズ/構造
 - スキーマ等
- ワークフロー生成 (maDMP生成)

ワークフロー生成時に最適化に必要な情報が埋め込まれています
順番に実行することで、システムがDMP情報を活用した最適化処理を行います

最適化情報
テンプレート

データガバナンス機能

実験環境準備

- ワークフローのセルを順に実行
- 実験環境の情報入力
 - 実験環境接続
 - 実験環境の生成

実験実施

- ワークフローのセルを順に実行
- 実験リポジリーを作成
 - 実験記録管理の準備

実験記録の保存

- ワークフローのセルを順に実行
- 実験記録の保存

実験のパッケージング

- ワークフローのセルを順に実行
- 論文等研究文書を保存

実験環境

- ワークフローのセルを順に実行
- 実験実施

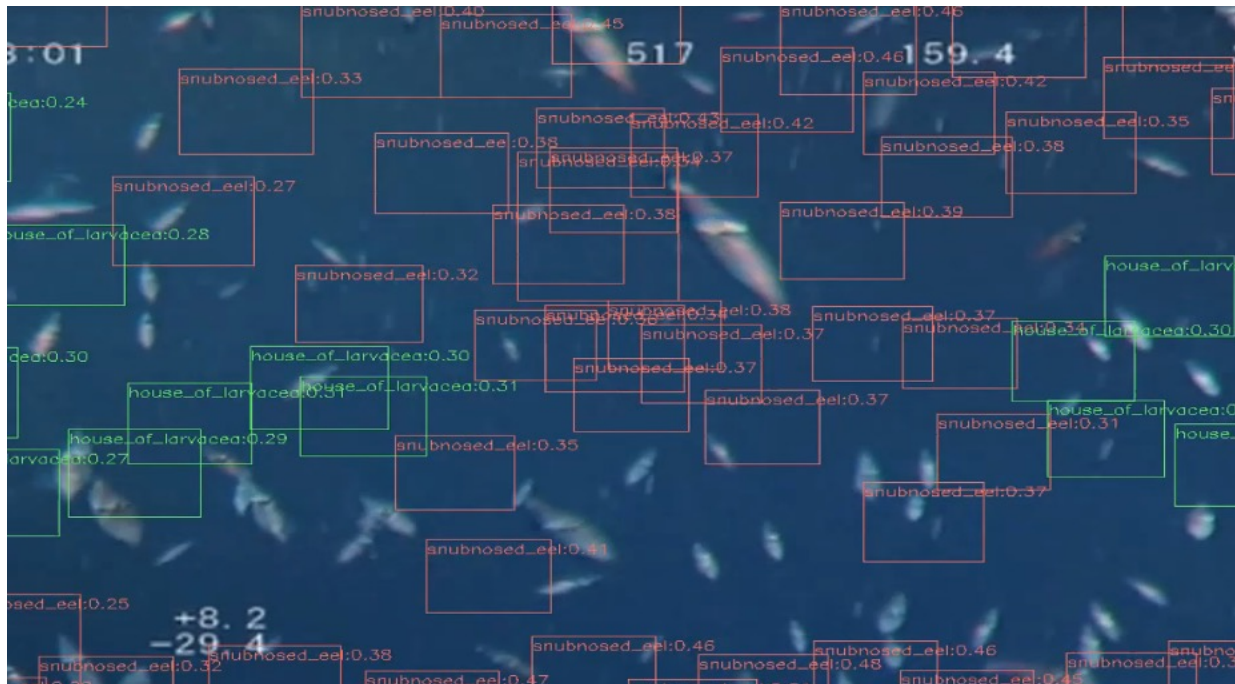
- ワークフローのセルを順に実行
- 研究データ容量モニタリング

- ワークフローのセルを順に実行
- データセット構成モニタリング

- ワークフローのセルを順に実行
- 再現性モニタリング

実験適用例

底生生物の検知／追跡

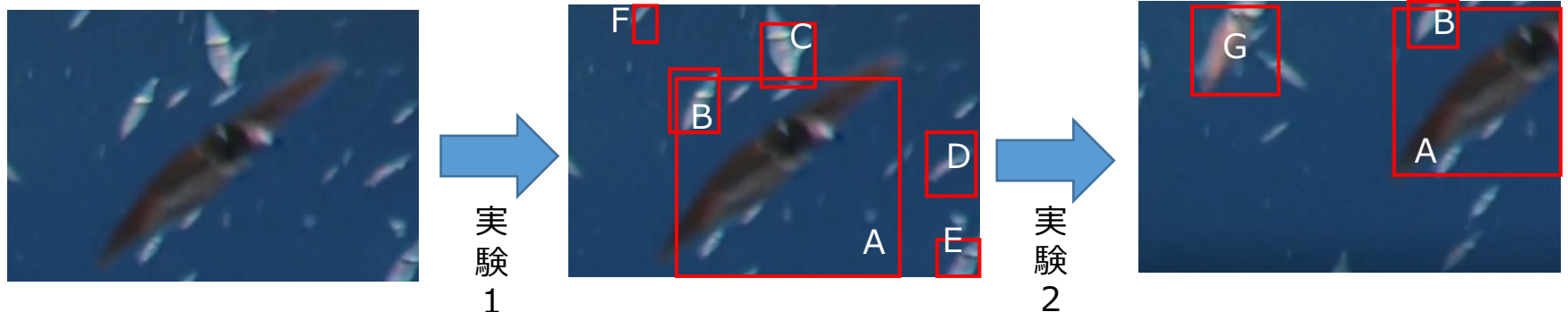


データ提供：海洋研究開発機構(JAMSTEC)

実験適用例の概要

海中を映した動画中に含まれる、底生生物を検知／追跡する。

- 実験1 物体検出
動画の各フレームで生物を検知する検知器を作成・評価する。
- 実験2 物体検出+追跡
動画のあるフレームで検出した生物が、
次フレームでどこに移動したかを追跡するAIを作成し、
追跡性能を評価する。



データ提供：海洋研究開発機構(JAMSTEC)

実験 1 「detr(物体検出の学習と評価)」の説明

実験概要、目的

概要：機械学習による物体検出の学習とその評価の出力。

目的：良い物体検出モデルを作成する。

実験構成

以下を構築

input_data

configs 設定ファイル（実験のパラメータ）
data 機械学習の教師データと評価用データ（実験の入力データ）
weight 機械学習の初期重み（実験の入力データ）

source

train.py 学習用プログラム
eval.py 評価用プログラム

実験の実行

main.ipynbを実行（環境構築、学習用プログラム実行）



約10分後、以下のフォルダに結果が格納される。

output_data

hogehoge_20220309183628 モデルと学習曲線

評価値を取得して、以下を確認する



- ・モデルと学習曲線（モデルの性能が良くなってるか（数値））



モデルと学習曲線のフォルダパスをconfigに適用して、eval.pyを実行し、推論結果の評価値と推論動画を得る。そして、モデルの性能が良くなっているかを目視確認する。

実験リポジトリ

```

input_data
├── configs
│   └── hogehoge.yaml
├── data
│   ├── 11175505
│   │   ├── annotations
│   │   │   ├── instances_train.json
│   │   │   └── instances_val.json
│   │   └── images
│   │       ├── frame_000000.png
│   │       ├── frame_000010.png
│   │       ├── ...
│   │       └── frame_001440.png
│   └── video_and_score_text
│       ├── 11195506.mov
│       ├── 11195506
│       │   ├── frame_000000.txt
│       │   ├── frame_000001.txt
│       │   ├── ...
│       │   └── frame_001754.txt
└── weight
    └── detr-r50-e632da11.pth

output_data
├── hogehoge_20220309183628
│   ├── hogehoge_valloss_min.pth
│   └── 2022_0309_1836
│       └── [train.pyによる評価データ（学習曲線）]
└── inference
    └── hogehoge_20220309183628
        └── [eval.pyによる評価データ（推論の評価値と推論動画）]

source
├── eval.py
├── train.py
├── main.ipynb
├── detr
└── Object-Detection-Metrics
    
```

実験 2 「DeepSORT」 + 「ByteTrack」 の説明

実験概要、目的

概要：物体の検出 + 追跡と、その評価の出力。
 実験 1 の出力（検出量の重み）を使って行う。
 目的：良い追跡パラメータを見つけ、追跡する。

実験構成

以下を構築

input_data

config 設定ファイル（実験のパラメータ）
 video 処理対象の動画と評価データ（実験の入力データ）
 weight 検出用の重み（実験の入力データ）

source

inference_detr+tracker.py 追跡用プログラム
 main.ipynb 実行用プログラム

実験の実行

main.ipynbを実行（環境構築、検出 + 追跡用プログラム実行）

↓
 約 1 時間後、以下のフォルダに結果が格納される。

output_data

result.txt 計算ログ
 result_metric.txt 評価値
 video.mp4 検出枠 + 追跡枠を動画に書き込んだもの

↓
 評価値を取得して、以下を確認する

・推論結果の評価値と推論動画（追跡の性能がよくなったか）

実験リポジトリ

input_data

config

colors.txt
 hogegege.yaml
 labels.txt

video

11195506.mov
 11195506
 frame_000000.txt
 ...

weight

ckpt.t7
 hogegege_valloss_min.pth（実験1の出力）

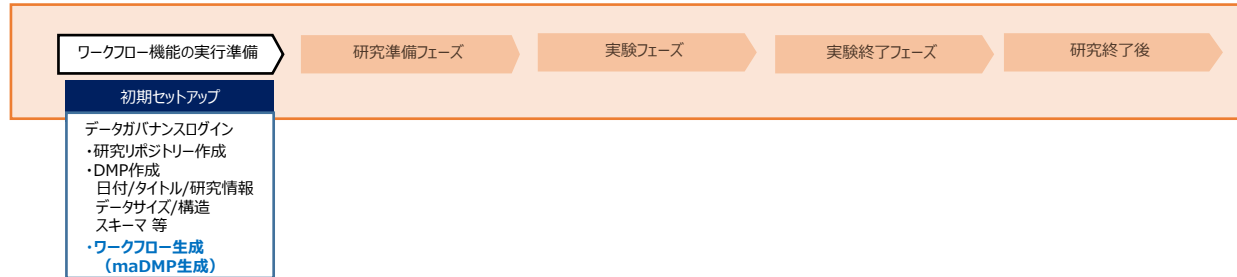
output_data

result.txt
 result_metric.txt
 video.mp4

source

inference_detr+tracker.py
 main.ipynb
 detection
 tracking

① DMPの作成および、maDMPの生成

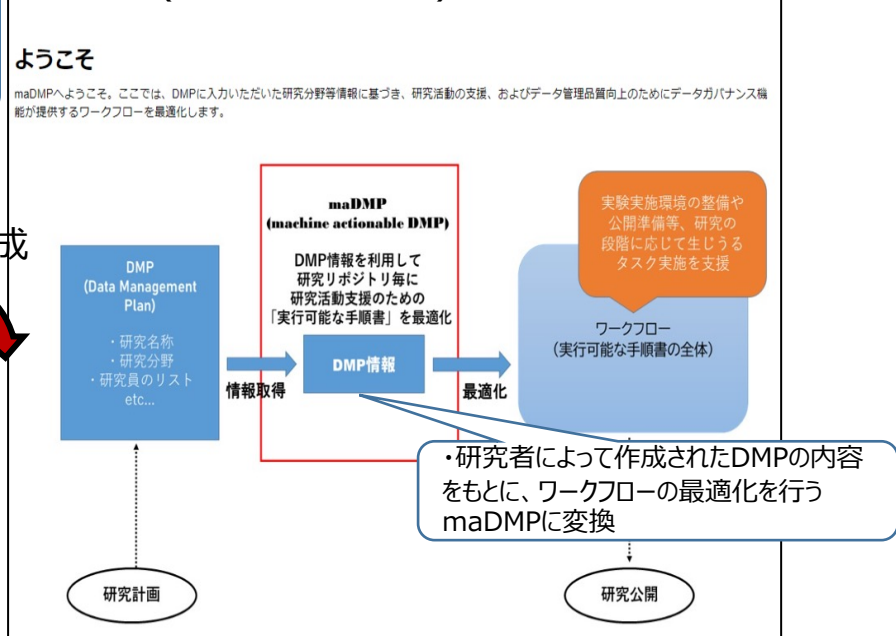


DMP入力画面

・各FA(AMED,JST,METI...)に対応したDMPの作成を、Jsonエディタにて作成可能

・各項目ごとに、Jsonエディタから入力する

maDMP(Notebook 形式)



②高性能実験環境準備用のワークフロー



高性能実験環境を準備する

高性能実験環境として、`mdx`環境を利用するための設定をします。
以下のセルを上から順番に実行してください。
2回目以降の実行の場合、このセルが選択された状態で画面上部に表示される以下のボタンをクリックしてから実行して下さい。



◆◆◆開発メモ◆◆◆
将来的には、[学認クラウドオンデマンド構築サービス](#)による動的な環境構築を実現する。

1. 高性能実験環境のアカウント情報の入力

以下のセルを実行し、表示されるフォームに高性能実験環境におけるアカウント情報を入力してください。

```
[1]: from IPython.display import clear_output
import getpass
name_mdx = input("高性能実験環境におけるSSHユーザ名:")
clear_output()
```

2. アカウント認証のための設定

[こちら](#)を押下し、ファイル一覧画面に遷移してください。
遷移後、`id_rsa`ファイルをドラッグアンドドロップによりアップロードしてください。
アップロード後、以下のセルを実行してください。

```
[3]: !mkdir -p /home/jovyan/.ssh/
!mv ~/id_rsa* ~/.ssh/id_rsa
!chmod 600 ~/.ssh/id_rsa
```

・ `maDMP`に記載された内容をもとに、高性能実験環境を準備する用のワークフロー（手順書）を提供。

・ 研究者は、`NoteBook`のセルを実行するのみで、高性能実験環境を準備できる。

③実験の実施 研究サンプル(底生生物の検知/追跡)

ワークフロー機能の実行準備

研究準備フェーズ

実験フェーズ

実験終了フェーズ

研究終了後

実験実施

ワークフローのセルを
順に実行
・実験実施

```
[1]: !pip install opencv-python cython-bbox motmetrics
!pip install torch==1.11.0+cu113 torchvision==0.12.0+cu113 -f https://download.pytorch.org/whl/cu113/torch_stable.html
!conda install -c conda-forge lap -y
!python3 inference_detr+tracker.py --config ../input_data/config/hogehoge.yaml
```

・実験の開始方法について、自由な実行をサポート
(再現性担保のため、一部制限あり)

```
Collecting opencv-python
  Downloading opencv_python-4.5.5.64-cp36-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (60.5 MB)
  _____ 60.5/60.5 MB 46.4 MB/s eta 0:00:00:01:00:01

Collecting cython-bbox
  Downloading cython_bbox-0.1.3.tar.gz (41 kB)
  _____ 41.3/41.3 KB 12.7 MB/s eta 0:00:00

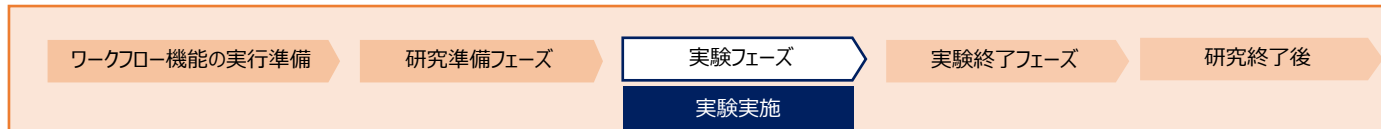
  Preparing metadata (setup.py) ... done
Collecting motmetrics
  Downloading motmetrics-1.2.0-py3-none-any.whl (151 kB)
  _____ 151.6/151.6 KB 41.9 MB/s eta 0:00:00

Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.9/site-packages (from opencv-python) (1.19.5)
Collecting pytest
  Downloading pytest-7.1.1-py3-none-any.whl (297 kB)
  _____ 297.0/297.0 KB 61.7 MB/s eta 0:00:00

Collecting pytest-benchmark
  Downloading pytest_benchmark-3.4.1-py2.py3-none-any.whl (50 kB)
  _____ 50.1/50.1 KB 16.4 MB/s eta 0:00:00

Collecting flake8-import-order
  Downloading flake8_import_order-0.18.1-py2.py3-none-any.whl (15 kB)
```

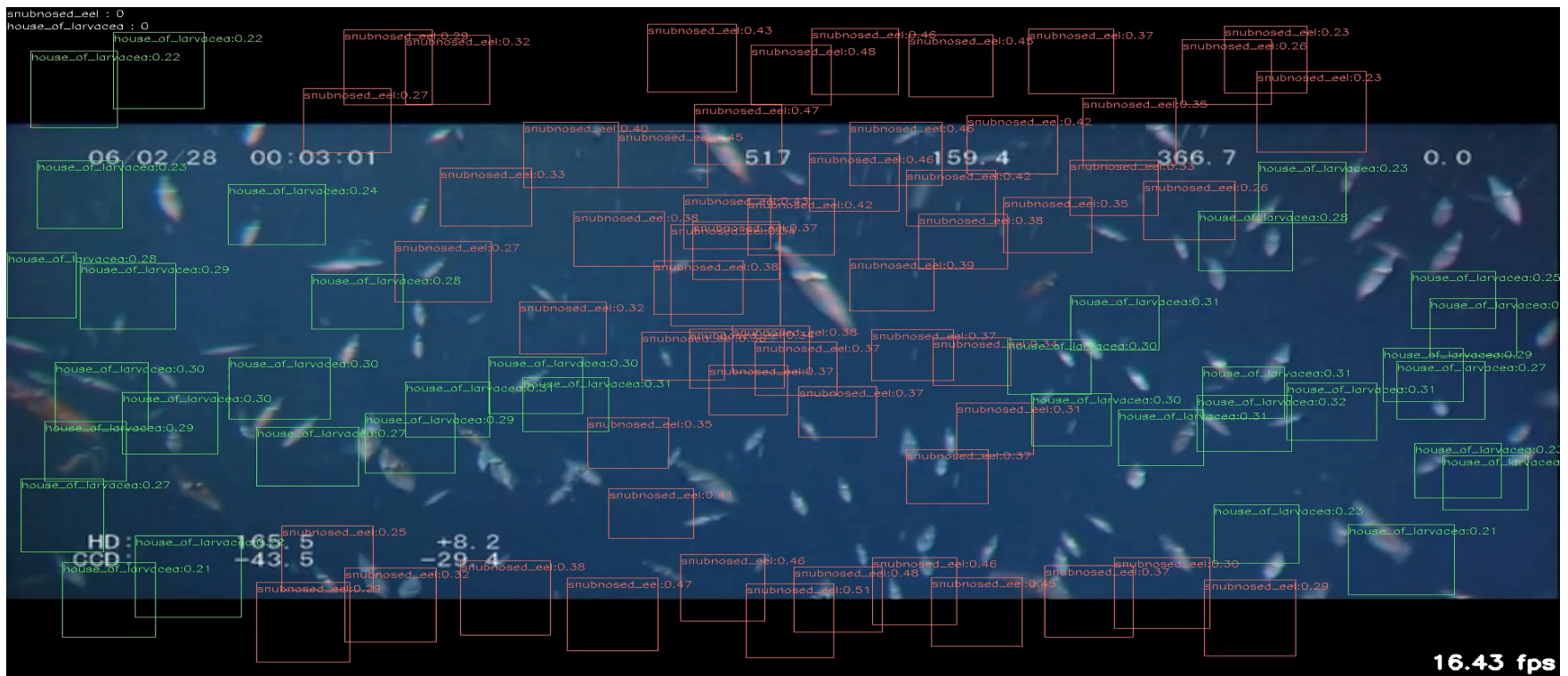
③実験の実施 研究サンプル(底生生物の検知/追跡)



ワークフローのセルを順
に実行
・実験実施

イカ アナゴ

※動画は開発中のものです



データ提供：海洋研究開発機構(JAMSTEC)

④ 実験の記録

ワークフロー機能の実行準備

研究準備フェーズ

実験フェーズ

実験終了フェーズ

研究終了後

実験記録の保存

ワークフローのセルを
順に実行
・実験記録の保存

2. 実験をデータガバナンス機能に途中保存する

※データの保存先としてAWS S3準拠のオブジェクトストレージを利用する場合は、
[こちら](#)を実行してください。

```
from IPython.display import display, Javascript
display(Javascript('IPython.notebook.save_checkpoint();'))
```

```
import os
import glob

# Git管理のパスのリストを作成する
%cd ~/
files = os.listdir()
# ディレクトリー一覧からGit-annex管理するディレクトリ(input_dataとoutput_data)を排除する
dirs = [f for f in files if os.path.isdir(f)]
dirs.remove('input_data')
dirs.remove('output_data')
# HOME直下のファイルを取得
files = [f for f in files if os.path.isfile(f)]
# Git管理するパスの配列を作成する
files.extend(dirs)
save_path = files

# Git-annex管理するパスの配列を作成する
annexed_save_path = ['input_data', 'output_data']
```

```
import papermill as pm

%cd ~/
# Git-annex管理ファイルを保存
pm.execute_notebook(
    'EX-WORKFLOW/util/base_data_lad_save_push.ipynb',
    '/home/jovyan/.local/push_log.ipynb',
    parameters = dict(SAVE_MESSAGE = message + ' (1/2)', PATH = annexed_save_path, IS_RECURSIVE = False)
)
```

・実験の記録について、NoteBookのセルによる手順書を提供
セルの実行のみで、実験の記録をGitによるバージョン管理で
行うことができるようにする

3. 今後の進め方

取り組み方針

1. 実研究プロジェクトからのフィードバックへ対応
 - プロトタイプシステムの継続的更新
 - 実研究プロジェクト候補（機械学習、ゲノム分析、数学など）
2. システム連携
 - RDC内の基盤・機能との連携
 - 他機関基盤との連携
3. 先読み
 - 分散ストレージ対応
4. 国内外へのプロモーション



Thank you

問合せ先：yoko@nii.ac.jp