



JAEA-Testing 2023-002

J-PARC 23-01

DOI:10.11484/jaea-testing-2023-002

JAEA-Testing

MLFデータ処理環境"空蟬"インストールマニュアル

Installation Manuals for "Utsusemi"

稲村 泰弘

Yasuhiro INAMURA

原子力科学研究部門

J-PARCセンター

物質・生命科学ディビジョン

Materials and Life Science Division

J-PARC Center

Sector of Nuclear Science Research

December 2023

Japan Atomic Energy Agency

日本原子力研究開発機構

本レポートは国立研究開発法人日本原子力研究開発機構が不定期に発行する成果報告書です。
本レポートはクリエイティブ・コモンズ 表示 4.0 国際 ライセンスの下に提供されています。
本レポートの成果（データを含む）に著作権が発生しない場合でも、同ライセンスと同様の
条件で利用してください。（<https://creativecommons.org/licenses/by/4.0/deed.ja>）
なお、本レポートの全文は日本原子力研究開発機構ウェブサイト（<https://www.jaea.go.jp>）
より発信されています。本レポートに関しては下記までお問合せください。

国立研究開発法人日本原子力研究開発機構 JAEA イノベーションハブ 研究成果利活用課
〒319-1112 茨城県那珂郡東海村大字村松 4 番地 49
E-mail: ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency.

This work is licensed under a Creative Commons Attribution 4.0 International License
(<https://creativecommons.org/licenses/by/4.0/deed.en>).

Even if the results of this report (including data) are not copyrighted, they must be used under
the same terms and conditions as CC-BY.

For inquiries regarding this report, please contact Institutional Repository and Utilization Section,
JAEA Innovation Hub, Japan Atomic Energy Agency.

4-49 Muramatsu, Tokai-mura, Naka-gun, Ibaraki-ken 319-1112, Japan

E-mail: ird-support@jaea.go.jp

MLF データ処理環境"空蟬"インストールマニュアル

日本原子力研究開発機構 原子力科学研究部門 J-PARC センター
物質・生命科学ディビジョン

稲村 泰弘

(2023 年 8 月 29 日受理)

空蟬とは、J-PARC の物質・生命科学実験施設 (MLF) において実施される中性子散乱実験にて得られたデータ処理に使用されるソフトウェア群の一つである。MLF の多くの装置 (ビームライン) では、測定されたデータから条件に応じて抽出し、強度の規格化、装置機器依存の強度補正、単位変換などを行い、物理量へと変換し可視化する必要がある。空蟬はこの一連の処理を効率的に行うために開発されたソフトウェア群である。空蟬は主にデータの抽出機能 (ヒストグラム化) や各種補正機能、それらを効率よく実行するためのグラフィカルインターフェースを持つソフトウェア、さらには処理されたデータを可視化するプロッタアプリケーションなどから構成されている。既に MLF の多数の装置において解析作業の要として動作している。

本書は、MLF の中性子散乱装置を運用する装置担当者やユーザーが空蟬を利用したデータ処理を自分自身の計算機環境にて実施する際に、その環境構築やインストール作業を容易に実施できるよう、オペレーションシステムごとに具体的な手順を記述したマニュアルである。なお、Windows や macOS といった一般向けオペレーションシステムであれば、通常のアプリケーションを扱う程度の知識があれば本書を利用してインストールが可能である。

Installation Manuals for “Utsusemi”

Yasuhiro INAMURA

Materials and Life Science Division, J-PARC Center, Sector of Nuclear Science Research
Japan Atomic Energy Agency
Tokai-mura, Naka-gun, Ibaraki-ken

(Received August 29, 2023)

'*Utsusemi*' is a suite of software used to process data obtained from measurements of neutron scattering experiments at the Materials and Life Science Experimental Facility (MLF), J-PARC. To directly obtain the physical quantities which scientists want to get from the data produced by instruments at MLF, many processes are required, such as creating histogram format data, easily visualizing and converting units and correcting intensity adapting the instrument conditions. '*Utsusemi*' software consists of many software components, many functions for data processing, graphical interface software for executing Utsusemi functions, data visualization applications, and so on. '*Utsusemi*' has already played an important role in data processing and has been widely employed in MLF beamlines. This document describes how to install the '*Utsusemi*' software on each operating system to be of help of instrument staff and users who want to process data by themselves. Installation of '*Utsusemi*' on Windows and macOS requires only general knowledge of working with PC applications according to this document.

Keywords : Neutron Scattering, Software, Installation, Manual, Data Treatment, Data Visualization, Utsusemi, MLF, J-PARC

目 次

| | |
|---|----|
| 1. はじめに..... | 1 |
| 1.1. 本書について..... | 1 |
| 1.2. インストールの概要..... | 1 |
| 1.3. 空蟬の概要..... | 1 |
| 2. Ubuntu 20.04/22.04 へのインストール (debian package 版) | 5 |
| 2.1. 概要..... | 5 |
| 2.2. 動作環境..... | 5 |
| 2.3. 手順の概要..... | 6 |
| 2.4. インストール..... | 6 |
| 2.5. 初期設定..... | 9 |
| 2.6. 空蟬の実行..... | 14 |
| 2.7. アンインストール..... | 15 |
| 2.8. 問題が発生した場合..... | 15 |
| 3. Windows 用バイナリ版インストール | 17 |
| 3.1. 概要..... | 17 |
| 3.2. 動作環境..... | 17 |
| 3.3. 手順の概要..... | 17 |
| 3.4. インストール..... | 17 |
| 3.5. 初期設定..... | 22 |
| 3.6. 空蟬の実行..... | 24 |
| 3.7. アンインストール..... | 26 |
| 3.8. 問題が発生した場合..... | 26 |
| 3.9. Appendix..... | 27 |
| 4. macOS (Big Sur および Monterey) 用バイナリ版インストール | 29 |
| 4.1. 概要..... | 29 |
| 4.2. 動作環境..... | 29 |
| 4.3. 手順の概要..... | 29 |
| 4.4. インストール..... | 29 |
| 4.5. 初期設定..... | 32 |
| 4.6. 空蟬の実行..... | 36 |
| 4.7. アンインストール..... | 38 |
| 4.8. 問題が発生した場合..... | 38 |
| 5. macOS (Catalina、Mojave および High Sierra) 用バイナリ版インストール | 40 |
| 5.1. 概要..... | 40 |
| 5.2. 動作環境..... | 40 |
| 5.3. 手順の概要..... | 40 |
| 5.4. インストール..... | 40 |
| 5.5. 初期設定..... | 44 |
| 5.6. 空蟬の実行..... | 47 |
| 5.7. アンインストール..... | 49 |
| 5.8. 問題が発生した場合..... | 49 |
| 6. Ubuntu 18.04 LTS および 20.04 LTS への汎用インストール..... | 51 |
| 6.1. 概要..... | 51 |
| 6.2. 動作環境..... | 51 |
| 6.3. 手順の概要..... | 51 |
| 6.4. インストール..... | 52 |
| 6.5. 初期設定..... | 58 |

| | | |
|------|---|----|
| 6.6. | 空蟬の実行 | 59 |
| 6.7. | アンインストール | 59 |
| 6.8. | 問題が発生した場合 | 59 |
| 7. | CentOS 7 へのインストール | 60 |
| 7.1. | 概要 | 60 |
| 7.2. | 動作環境 | 60 |
| 7.3. | 手順の概要 | 62 |
| 7.4. | インストール | 63 |
| 7.5. | 環境設定 | 67 |
| 7.6. | 空蟬の実行 | 69 |
| 7.7. | アンインストール | 69 |
| 7.8. | 問題が発生した場合 | 69 |
| 8. | AlmaLinux 8 および RockyLinux 8 へのインストール | 70 |
| 8.1. | 概要 | 70 |
| 8.2. | 動作環境 | 70 |
| 8.3. | 手順の概要 | 73 |
| 8.4. | インストール | 73 |
| 8.5. | 環境設定 | 77 |
| 8.6. | 空蟬の実行 | 79 |
| 8.7. | アンインストール | 79 |
| 8.8. | 問題が発生した場合 | 79 |
| | 参考文献 | 80 |

Contents

| | |
|---|----|
| 1. Introduction..... | 1 |
| 1.1. About this document | 1 |
| 1.2. Utsusemi installations | 1 |
| 1.3. Utsusemi information..... | 1 |
| 2. Installation on Ubuntu 20.04/22.04 using debian package | 5 |
| 2.1. Outline..... | 5 |
| 2.2. Required environment..... | 5 |
| 2.3. Procedure | 6 |
| 2.4. Installation..... | 6 |
| 2.5. Settings..... | 9 |
| 2.6. Executing Utsusemi | 14 |
| 2.7. Uninstall..... | 15 |
| 2.8. Bugs reporting..... | 15 |
| 3. Binary Installation on Windows..... | 17 |
| 3.1. Outline..... | 17 |
| 3.2. Required environment..... | 17 |
| 3.3. Procedure | 17 |
| 3.4. Installation..... | 17 |
| 3.5. Settings..... | 22 |
| 3.6. Executing Utsusemi | 24 |
| 3.7. Uninstall..... | 26 |
| 3.8. Bugs reporting..... | 26 |
| 3.9. Appendix..... | 27 |
| 4. Binary Installation on macOS (Big Sur, Monterey)..... | 29 |
| 4.1. Outline..... | 29 |
| 4.2. Required environment..... | 29 |
| 4.3. Procedure | 29 |
| 4.4. Installation..... | 29 |
| 4.5. Settings..... | 32 |
| 4.6. Executing Utsusemi | 36 |
| 4.7. Uninstall..... | 38 |
| 4.8. Bugs reporting..... | 38 |
| 5. Binary Installation on macOS (Catalina, Mojave and High Sierra)..... | 40 |
| 5.1. Outline..... | 40 |
| 5.2. Required environment..... | 40 |
| 5.3. Procedure | 40 |
| 5.4. Installation..... | 40 |
| 5.5. Settings..... | 44 |
| 5.6. Executing Utsusemi | 47 |
| 5.7. Uninstall..... | 49 |
| 5.8. Bugs reporting..... | 49 |
| 6. Common Installation on Ubuntu 18.04 LTS, 20.04 LTS..... | 51 |
| 6.1. Outline..... | 51 |
| 6.2. Required environment..... | 51 |
| 6.3. Procedure | 51 |
| 6.4. Installation..... | 52 |
| 6.5. Settings..... | 58 |
| 6.6. Executing Utsusemi | 59 |
| 6.7. Uninstall..... | 59 |
| 6.8. Bug reporting | 59 |
| 7. Installation on CentOS 7 | 60 |
| 7.1. Outline..... | 60 |
| 7.2. Required environment..... | 60 |

| | | |
|------|---|----|
| 7.3. | Procedure | 62 |
| 7.4. | Installation..... | 63 |
| 7.5. | Settings..... | 67 |
| 7.6. | Executing Utsusemi | 69 |
| 7.7. | Uninstall..... | 69 |
| 7.8. | Bugs reporting..... | 69 |
| 8. | Installation on AlmaLinux 8 and RockyLinux 8..... | 70 |
| 8.1. | Outline..... | 70 |
| 8.2. | Required environment..... | 70 |
| 8.3. | Procedure | 73 |
| 8.4. | Installation..... | 73 |
| 8.5. | Settings..... | 77 |
| 8.6. | Executing Utsusemi | 79 |
| 8.7. | Uninstall..... | 79 |
| 8.8. | Bugs reporting..... | 79 |
| | References..... | 80 |

図表リスト

| | | |
|----------|---|----|
| Figure 1 | Structure of Utsusemi in a broad sense | 2 |
| Table 1 | Available operation systems for Utsusemi | 2 |
| Table 2 | Available DAQ electronics modules on Utsusemi | 3 |
| Table 3 | Beam lines which imported Utsusemi | 4 |
| Table 4 | Environment for Development | 4 |
| Table 5 | Available beam lines for Utsusemi debian packages | 5 |
| Table 6 | Available Ubuntu Linux version | 5 |
| Table 7 | List of available debian packages | 7 |
| Table 8 | Required packages for each beamline | 7 |
| Table 9 | Example names of actual debian packages | 7 |
| Table 10 | Default path of MLF software installation | 10 |
| Table 11 | Default path of data folder | 10 |
| Table 12 | Utsusemi applications on Ubuntu Linux | 11 |
| Table 13 | Environment variables for Utsusemi | 12 |
| Table 14 | Path to the environment setting file for each BL | 12 |
| Table 15 | Environment variables for Utsusemi | 12 |
| Table 16 | Key codes to set environment | 14 |
| Table 17 | Package name required for Manyo library installation | 16 |
| Table 18 | Package name required for Utsusemi installation | 16 |
| Table 19 | Available beam lines for Utsusemi on Windows | 17 |
| Table 20 | Available Windows version | 17 |
| Table 21 | Account name examples | 17 |
| Table 22 | Type of installation | 18 |
| Table 23 | list of the packages required by a minimum BinaryInstaller for win | 21 |
| Table 24 | List of Utsusemi Applications installed | 22 |
| Table 25 | Path to folders required by Utsusemi | 22 |
| Table 26 | Example of path to folders required by Utsusemi | 22 |
| Table 27 | Environment variables for Utsusemi on Windows | 23 |
| Table 28 | Examples of beam line RAW data folder | 24 |
| Table 29 | About Release number | 25 |
| Table 30 | Available beam lines for Utsusemi on Windows | 29 |
| Table 31 | Confirmed macOS versions | 29 |
| Table 32 | List of Utsusemi applications | 32 |
| Table 33 | Path to folders required by Utsusemi | 33 |
| Table 34 | Example of path to folders required by Utsusemi | 33 |
| Table 35 | Environment Variables for Utsusemi on macOS after bigsur | 34 |
| Table 36 | Examples of beam line RAW data folder | 35 |
| Table 37 | About release number | 38 |
| Table 38 | Available beam lines for Utsusemi on Windows | 40 |
| Table 39 | Installed Utsusemi applications | 44 |
| Table 40 | Path to folders required by Utsusemi | 44 |
| Table 41 | Example of path to folders required by Utsusemi | 44 |
| Table 42 | Environment Variables for Utsusemi on macOS before 10.15 (macOS Catalina) | 46 |
| Table 43 | Examples of beam line RAW data folder | 47 |
| Table 44 | About Release number | 48 |
| Table 45 | Available Beam lines for script installation for Ubuntu | 51 |
| Table 46 | Available Ubuntu Linux version for general installation script | 51 |

| | | |
|----------|--|----|
| Table 47 | Menu number and meanings on Install.py | 54 |
| Table 48 | Default Installation path..... | 56 |
| Table 49 | Default Installation path for private area..... | 57 |
| Table 50 | Keywords for Utsusemi environment setting | 58 |
| Table 51 | Environment variables for Utsusemi | 58 |
| Table 52 | External CentOS7 repositories required for Utsusemi installation..... | 60 |
| Table 53 | External Shared Library required for Utsusemi environment (CentOS7)..... | 61 |
| Table 54 | Command List of do_make_Manyo.py and do_makeUtsusemi.py (CentOS7) | 64 |
| Table 55 | Installation paths of python codes specialized to beam lines (CentOS7)..... | 65 |
| Table 56 | External Shared Libraries for BL02 environment (CentOS7)..... | 66 |
| Table 57 | Environment variables for Utsusemi (CentOS7)..... | 68 |
| Table 58 | External AlmaLinux8 and RockyLinux8 repositories required before installation..... | 70 |
| Table 59 | External Shared Library required for Utsusemi environment (RH8 series) | 71 |
| Table 60 | Command List of do_make_Manyo.py and do_makeUtsusemi.py(RH8 series) | 74 |
| Table 61 | Installation paths of python codes specialized to beam lines (RH8 series)..... | 75 |
| Table 62 | External shared libraries for BL02 environment (RH8 series)..... | 76 |
| Table 63 | Environment variables for Utsusemi (RH8 series)..... | 78 |

1. はじめに

1.1. 本書について

本書は J-PARC、物質・生命科学実験施設 (MLF) の中性子散乱実験装置で使用される汎用データ処理環境である空蟬¹⁾を使用するために必要なインストールに関するマニュアルである。主に本書では空蟬を PC にインストールし、データ処理を行う環境を構築する方法を記述している。空蟬自体の基本的な使用方法については、今後発行予定の空蟬ユーザーズマニュアルを参照していただきたい。

また本書の内容は執筆時 (2023 年 10 月現在) の情報であり、対応する OS のアップデートなどで常に更新される。最新の情報は空蟬ポータルサイト²⁾より入手可能である。

1.2. インストールの概要

空蟬のインストール方法は、大きく分けて 2 種類用意されている。一つはバイナリインストーラであり、もう一つはソースコードからのコンパイルである。

本書執筆時では、下記のようにインストール方法が確立されている。

- バイナリインストーラによる方法
 - Ubuntu Linux 用バイナリパッケージ (Ubuntu 20.04 と 22.04 のみ)
 - Windows 用バイナリインストーラ
 - macOS 用バイナリインストーラ
- ソースコードからのコンパイルによる方法
 - Ubuntu Linux 用 (18.04 20.04)
 - RedHat 互換 Linux 用 (CentOS 7.x, AlmaLinux 8.x, RockyLinux 8.x)

次章以降で、上記のそれぞれのインストール方法について章に分けて記述する。

1.3. 空蟬の概要

空蟬は、J-PARC、MLF で使用されている DAQ ミドルウェア³⁾の MLF コンポーネントが出力するイベントデータのデータリダクション、可視化や補正を行うためのコード群である。MLF で開発が進められているデータ処理基盤である万葉ライブラリ⁴⁾を利用し、柔軟なヒストグラム化、特に試料環境や装置状態を加味したデータ分別機能を使用した高度な解析を目指す。また測定中のデータを安全に解析できる Live Data Reduction の MLF での実用化も目指している。

現在既に多くの装置に導入が進み、MLF における解析ソフトウェアのフレームワーク的な役目を担う。

以下に本書で取り扱う空蟬についてその概要を述べる。

1.3.1. 何ができるか

MLF における多くの装置では、核破砕型中性子源から発生した中性子 (入射中性子) を測定試料に照射し、試料内で原子分子と相互作用して出てきた中性子 (散乱中性子) を検出器にて検知・収集する。いわゆる中性子散乱測定では、散乱された中性子の方向やエネルギーを捉える必要があるが、散乱の方向は検出器の位置情報を用いて、エネルギーは飛行時間法 (Time-Of-Flight 法: 発生中性子のエネルギーを距離と時間で分別する手法) を用いて知ることができる。この散乱中性子の検出された時間と位置情報はイベント記録方式で保存 (イベントデータ) されるが、このデータを柔軟に活用し試料内部の情報 (原子分子の構造や振動状態など) を得るためのデータ処理を行うことが、空蟬の主要な目的である。同様にいくつかのデータ補正機能や可視化機能が搭載されている。

1.3.2. 空蟬の構成

空蟬は万葉ライブラリと呼ばれる MLF 標準として使用されるデータ処理基盤ソフトウェアをベースとして構築されている。万葉ライブラリでは基本的なコードは C++ で書かれてい

るが、そのインターフェースとして Python を利用しているため、ユーザーは Python コードから万葉ライブラリの関数を自由に呼び出せる。同様に空蟬も多数の独自関数を含み、それらの大半は Python から使用される。ただし C++ のコア関数は非常にプリミティブなものが多いため、実際のユーザーはさらに Python のコードで使いやすい形にされた関数（ファサード関数）を利用することになる。このファサード関数を組み合わせてシーケンスを作成し簡単に実行するためのソフトウェアも用意されている。

また、データを可視化するために必要なソフトウェアも用意されている。かつては空蟬に含まれていた汎用可視化ソフトウェアの一部は、夕顔という名前で分離させて別モジュールとして扱うようになっている。そのため多くの空蟬のソフトウェアは夕顔が必須である。

したがって空蟬と呼ばれるものの広義な意味での構成は Figure 1 のように、万葉ライブラリ、夕顔、そして空蟬の 3 種類のコードやモジュール群が組み合わさったものとも言える。

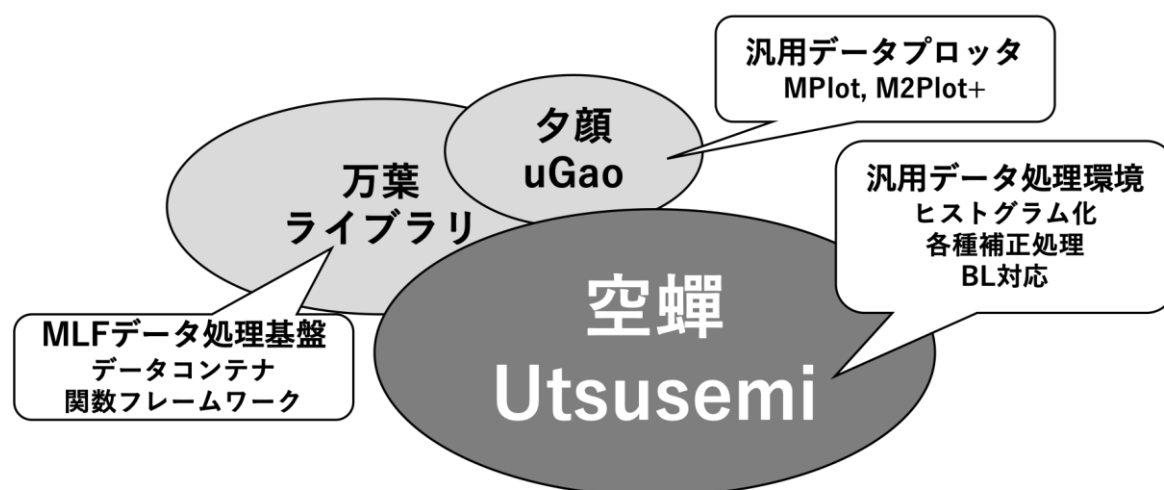


Figure 1 Structure of Utsusemi in a broad sense

よって空蟬にて使用できる機能(関数)やソフトウェアは以下のような構成となっている。

- データ処理機能
 - 基本 Python 関数群（空蟬環境の Python スクリプトで利用できるコマンドなど）
 - ファサード関数群（基本関数を使いやすくまとめたコマンド群）
 - コマンド実行ソフトウェア（ファサード関数を組み合わせてスクリプト作成・実行するソフトウェア）
- 可視化ソフトウェア
 - 各種汎用可視化ソフトウェア（"夕顔 MPlot"、"夕顔 M2Plot+"、"DetectMap"）
 - 非弾性散乱用可視化ソフトウェア（"VisualContM"、"D4MatSlicer"）

1.3.3. 動作環境

空蟬は、Python 3.6 以降（64bit）が動作する以下（Table 1）に示す OS の上でのみ動作する。

Table 1 Available operation systems for Utsusemi

| OS | Architecture | ディストリビューションなど |
|---------|----------------|---|
| Linux | x86_64, arm64 | Ubuntu 18.04, 20.04, 22.04 / CentOS 7.x, 8.x / AlmaLinux 8 |
| Windows | x86_64 | Windows 10, 11 |
| macOS | x86_64 (Intel) | 10.14(Mojave), 10.15(Catalina), 11.6(Big Sur), 12.4(Monterey) |
| macOS | Apple Silicon | 12.4(Monterey) |

またインストールを行う際の PC は、少なくとも 8GB 以上のメモリ容量、1GB のストレージの空き容量が必要である。データ処理時の要求メモリは、典型的な BL の 1 つの Run を処理するのに 0.5GB 程度、複数同時に処理する場合はその倍数程度が必要となる。特殊な多次元空間データ処理アプリケーション（D4Mat2Slicer など）では、最低でも 32GB は必要となる。

1.3.4. 主な機能

- MLF のイベントデータのヒストグラム化機能
 - DAQ ミドルウェアが出力するイベントデータをヒストグラム化
 - ヒストグラムの横軸単位：Time-of-Flight、波長、運動量遷移（momentum transfer）、エネルギー、エネルギー遷移、d 値
 - フィルタリングの使用：時間分解、外部 TTL シグナルを使用したデータ分別
- 汎用的なコマンドシーケンス実行ソフトウェア（SequenceEditor）
 - スクリプトはコマンドラインでも実行可能
- 汎用的可視化ソフトウェア
 - 夕顔を用いたプロット
 - 1 次元プロット（MPlot）
 - 2 次元プロット（M2Plot+）
 - 検出器マップ表示プロット（DetectMap）
- 中性子散乱測定 of データ処理・可視化ソフトウェア
 - 簡単な補正
 - 検出器効率補正、立体角補正、規格化、非弾性専用補正（ki/kf）など
 - 単結晶非弾性散乱測定の可視化（VisualContM）
 - 単結晶多次元測定の可視化（D4MatSlicer、D4Mat2Slicer）

1.3.5. 対応する中性子検出用モジュール

様々な検出器からのシグナルは、それぞれ専用のハードウェア機器（モジュール）で収集され、DAQ ミドルウェアを通すことでイベントデータ化される。MLF のイベントデータはモジュールごとに少しフォーマットが異なるため、その対応状況を Table 2 に示す。

Table 2 Available DAQ electronics modules on Utsusemi

| 対応検出器 | モジュール名 | モジュールの説明 |
|------------------------------|-------------------------|--|
| PSD（一次元位置敏感型検出器） | NEUNET | 1 次元型の検出器用のモジュール |
| 窒素モニター | GATENET | 0 次元型の検出器用のモジュール |
| 汎用シグナル | TrigNET | シグナルによって同時に測定されている中性子イベントを分別し、個別に解析するために使用 |
| ガンマ線検出器 | APV8008 | |
| 1 次元シンチレーションカウンタ 2 次元系検出器 | Readout, ReadoutGate | 2 次元系検出器（WLSF、MWPC、RPMT）にも対応 |

1.3.6. 導入されている MLF のビームライン

2022 年度現在で、空蟬が導入されているビームライン（BL）は Table 3 のようになっている。

Table 3 Beam lines which imported Utsusemi

| BL | 名前 | コード | 使用形態 |
|------|--------|-----|----------------|
| BL01 | 四季 | SIK | 空蟬 |
| BL02 | DNA | DNA | 空蟬+DNA 用拡張 |
| BL11 | PLANET | HPN | 空蟬+HPN 用ソフトウェア |
| BL14 | アマテラス | AMR | 空蟬 |
| BL15 | 大観 | SAS | 空蟬+SAS 用拡張 |
| BL17 | 写楽 | NVR | 空蟬 |
| BL19 | 匠 | ENG | 空蟬+ENG 用ソフトウェア |
| BL21 | NOVA | NVA | 空蟬（一部） |

1.3.7. 開発環境

開発環境は Table 4 のようになっている。

Table 4 Environment for Development

| 項目 | 使用 |
|--------------------|---|
| 開発言語 | C++, Python3 (3.6, 3.8) |
| メイン開発環境 | Linux (Ubuntu 20.04) |
| 利用外部ライブラリ (C++) | 万葉ライブラリ環境(Boost, HDF5, NeXus, MiniXML, gsl) |
| 利用外部ライブラリ (Python) | numpy, scipy, matplotlib, PySide2(Qt), pyqtgraph など |
| 動作確認環境 | Ubuntu linux 16.04(64bit), 18.04, 20.04 / CentOS 7.x, 8.x / Windows 10 (64bit) / macOS (10.14, 10.15, 11.6, 12.5) |

1.3.8. ライセンス

空蟬のライセンスは GPLv3⁵⁾に従うものとする。

2. Ubuntu 20.04/22.04 へのインストール (debian package 版)

2.1. 概要

本章では、空蟬環境を Ubuntu 20.04/22.04 に対し Debian バイナリパッケージでインストールする方法を示す。

なお、2022 年 11 月現在で、対応している MLF におけるビームラインは Table 5 の通りである。

Table 5 Available beam lines for Utsusemi debian packages

| BL(code) | 注意点 | 状況 |
|-----------|--------------|------|
| BL01(SIK) | 特になし | 対応済み |
| BL02(DNA) | QENSfit は含まず | 対応済み |
| BL14(AMR) | 特になし | 対応済み |
| BL11(HPN) | Teuse は含まず | 対応済み |
| BL15(SAS) | Dview は含まず | 対応済み |

2.2. 動作環境

本章で使用するパッケージの対象 OS は Table 6 の通りである。

Table 6 Available Ubuntu Linux version

| Ubuntu Linux | Architecture |
|--------------|----------------|
| 20.04 LTS | x86_64 (amd64) |
| 22.04 LTS | |

他の OS (CentOS など) にインストールしたい場合は、CentOS 7 へのインストールマニュアルなどを参照のこと。

2.2.1. 事前の確認事項

Ubuntu Linux は目的の PC に既に導入されていることを想定している。また極力 OS で提供されているパッケージを利用するため、その OS に対応したインストール用レポジトリにアクセスできる、つまり PC がネットに接続された状態にしておくこと。インストールするためには root 権限が必要である。

2.2.1.1. 古い空蟬 (スクリプト版) の削除

空蟬は、デフォルト設定では、下記の場所にインストールされている。

```
/opt/mlfsoft/
```

このフォルダがない場合 (空蟬をインストールした経験がない) 本節を無視して良い。

なお、過去に Debian パッケージによるインストールを行っていた場合は本節を行わずに次に進むこと。次節以降のインストール作業により、自動的に上書き (更新) される。どのようにインストールされたかわからない場合は以下のコマンドで確認すること。

```
$ sudo dpkg -l mlf-manyo1
```

Debian パッケージによってインストールされている場合は以下のように表示される。


```
$ sudo dpkg -l mlf-manyo1
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-
aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                Version                Architecture Description
+++-=====
=====>
ii  mlf-manyo1:amd64 0.3.4.210713-ubu20 amd64      Manyo Library
```

Debian パッケージでインストールされていない場合は下記のように表示される。

```
$ sudo dpkg -l mlf-manyo1
[sudo] password for hoge:
dpkg-query: no packages found matching mlf-manyo1
```

上記のように目的の Ubuntu に、Debian パッケージでインストールされていなかった場合（スクリプトでインストールするタイプ、すなわち“python3 Install.py”コマンドでインストールした場合）、下記のようにフォルダを削除する。

```
$ sudo rm -rf /opt/mlfsoft
```

2.3. 手順の概要

主な手順としては、以下の通りである。

- インストール
 - Ubuntu のレポジトリ情報の更新
 - インストール用スクリプトの実行、もしくは各種パッケージ（mlf-manyo1 パッケージ、mlf-utsusemi-with-ugao パッケージなど）のインストール
- 初期設定

2.4. インストール

2.4.1. Ubuntu のレポジトリ情報の更新

Ubuntu のレポジトリ情報を最新にするために、以下のコマンドを実行する。

```
$ sudo apt update
```

このステップ以降、インターネットへの接続が必要であるが、このコマンドがエラーなく実行できていれば問題ない。そのほか Web ブラウザなどで確認することもできる。実際のネットワークの設定や、確認方法は割愛する。

2.4.2. 簡易版

最も簡単な方法はインストール用スクリプト（installUtsusemi_FromDebianPackages.sh）をダウンロードし、実行することである。具体的には下記のように行う。

```
$ wget
https://cdn.mlf.plus/ce/Utsusemi4/Downloads/installUtsusemi_FromDebianP
ackages.sh
$ bash ./installUtsusemi_FromDebianPackages.sh
[sudo] password for xxx:
```


これにより、Ubuntu のバージョン (20.04/22.04) に応じたパッケージファイルの最新版が自動的にダウンロードされ、インストールされる。パッケージファイルは、上記のスクリプトを動作したフォルダ内にダウンロードされ、スクリプト内で root 権限 (sudo) にて実行される。インストール後にこれらのスクリプトやパッケージファイルは削除して良い。

なお、このスクリプトによりインストールされるパッケージは Table 7 の通りである。

Table 7 List of available debian packages

| パッケージ名 | インストール対象 | 必要なパッケージ | BL |
|------------------------|----------------------|------------------------|---------|
| mlf-manyo1 | 万葉ライブラリ | | 全 BL 共通 |
| mlf-utsusemi-with-ugao | 空蟬一式 (夕顔も含む) | mlf-manyo1 | 全 BL 共通 |
| mlf-utsusemi-sas | BL15 (大観) データ処理用コード | mlf-utsusemi-with-ugao | BL15 |
| mlf-utsusemi-dna | BL02 (DNA) データ処理用コード | mlf-utsusemi-with-ugao | BL02 |

2.4.3. 別のインストール方法 (手動)

できるだけ不要なパッケージをインストールしたくない場合、使用したいビームライン環境に必要なパッケージのみダウンロードして直接インストールすることも可能である (Table 8)。なおダウンロードは、空蟬インストールポータルから可能である。

Table 8 Required packages for each beamline

| Beamline | Packages |
|------------------|--|
| BL01, BL11, BL14 | mlf-manyo1, mlf-utsusemi-with-ugao |
| BL02 | mlf-manyo1, mlf-utsusemi-with-ugao, mlf-utsusemi-dna |
| BL15 | mlf-manyo1, mlf-utsusemi-with-ugao, mlf-utsusemi-sas |

ダウンロードするパッケージファイルの名前は、以下のように、パッケージ名にバージョン、ディストリブション (ubu20 や ubu22) やアーキテクチャ (amd64) などが付いた名前となる (Table 9)。

Table 9 Example names of actual debian packages

| ファイル名の例 |
|---|
| mlf-manyo1_0.3.4.210713-ubu20_amd64.deb |
| mlf-utsusemi-with-ugao_4.0.210713-ubu20_amd64.deb |
| mlf-utsusemi-sas_4.0.210713-ubu20_amd64.deb |
| mlf-utsusemi-dna_4.0.210713-ubu20_amd64.deb |

ファイル名に含まれる 0.3.4 (mlf-manyo1) および 4.0 (mlf-utsusemi-with-ugao) はリリース番号と呼ばれ、比較的大きなバージョンアップがあった場合に新たに付与される。210713 はソースコードの最終変更日でリリース番号とは独立にコードのアップデートとともに増加する。ubu20 はインストール先の Ubuntu のバージョン、amd64 は CPU のタイプ (アーキテクチャ) を示す。

2.4.3.1. 万葉ライブラリと空蟬環境 (+夕顔)

パッケージ名は以下の二つである。

- mlf-manyo1
- mlf-utsusemi-with-ugao

この二つは、全てのビームライン (BL01、BL02、BL11、BL14、BL15) で必要である。

ダウンロードしたファイル(例:“mlf-manyo1_0.3.4.210713-ubu20_amd64.deb”、“mlf-utsusemi-with-ugao_4.0.210713-ubu20_amd64.deb”)を以下のように指定して実行する。

まず、万葉ライブラリをインストールする。

```
$ sudo apt install ./mlf-manyo1_0.3.4.210713-ubu20_amd64.deb
```

ここでファイルの指定として、フォルダなども含めたパス（フルパス）が必要である。上記は同じファイルと同じフォルダにいたので、“./”で代用できる。下記のようにファイル名だけ書いてもエラーになるので注意すること。

```
$ sudo apt install mlf-manyo1_0.3.4.210713-ubu20_amd64.deb
ERROR : Unable to locate package
```

なお、実行時のログは以下のようなものである。

```
$ sudo apt install ./mlf-manyo1_0.3.4.210713-ubu20_amd64.deb
[sudo] password for hoge:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'mlf-manyo1' instead of
'/media/psf/Home/Ubuntu20.04/mlf-manyo1_0.3.4.210713-ubu20_amd64.deb'
The following additional packages will be installed:
  libaec0 libboost-program-options1.71.0 libboost-serialization1.71.0
  libgfortran5 libgsl23 libgslcblas0 libhdf4-0 libhdf5-103
  libnexus1 libsz2
Suggested packages:
  gsl-ref-psdoc | gsl-doc-pdf | gsl-doc-info | gsl-ref-html libhdf4-doc
  libhdf4-dev hdf4-tools
The following NEW packages will be installed:
  libaec0 libboost-program-options1.71.0 libboost-serialization1.71.0
  libgfortran5 libgsl23 libgslcblas0 libhdf4-0 libhdf5-103
  libnexus1 libsz2 mlf-manyo1
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,050 kB/5,722 kB of archives.
After this operation, 27.1 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

途中でインストール継続確認のメッセージが表示される。インストールに支障がなければ、“y”を入力する。こののちインストールが始まる。特にエラーが起きていなければインストールは終了している。なお、apt コマンドのオプションに“-y”を加えておけば継続確認をされることはない。

```
$ sudo apt install -y ./mlf-manyo1_0.3.4.210713-ubu20_amd64.deb
```

次に空蟬環境（+タ顔）のインストールを下記のように実行する。

```
$ sudo apt install ./mlf-utsusemi-with-ugao_4.0.210713-ubu20_amd64.deb
```

実行時のログは以下のようなものである。

```
$ sudo apt install ./mlf-utsusemi-with-ugao_4.0.210713-ubu20_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'mlf-utsusemi-with-ugao' instead of
'/media/psf/Home/Ubuntu20.04/mlf-utsusemi-with-ugao_4.0.210210-
ubu20_amd64.deb'
The following additional packages will be installed:
  blt fonts-lyx javascript-common libblas3 libdouble-conversion3 libjs-
jquery libjs-jquery-ui liblapack3

(中略)

python3-pyside2.qtgui python3-pyside2.qtwidggets python3-scipy python3-
tk qt5-gtk-platformtheme qttranslations5-l10n
tk8.6-blt2.5 ttf-bitstream-vera
0 upgraded, 40 newly installed, 0 to remove and 0 not upgraded.
Need to get 43.2 MB/63.7 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

途中でインストール継続確認のメッセージが表示される。インストールに支障がなければ、「y」を入力する。こののちインストールが始まる。特にエラーが起きていなければ無事にインストールは終了している。

2.4.3.2. BL15 環境のインストール

BL15 は別途専用パッケージがあるので、それをダウンロードし（例：“mlf-utsusemi-sas_4.0.210713-ubu20_amd64.deb”）、同様に指定して実行する。

```
$ sudo apt install -y ./mlf-utsusemi-sas_4.0.210713-ubu20_amd64.deb
```

2.4.3.3. BL02 環境のインストール

BL02 は別途専用パッケージがあるので、それをダウンロードし（例：“mlf-utsusemi-dna_4.0.210713-ubu20_amd64.deb”）、同様に指定して実行する。

```
$ sudo apt install -y ./mlf-utsusemi-dna_4.0.210713-ubu20_amd64.deb
```

また、BL02 の環境では、ホームフォルダに以下のようなフォルダ（“ana/DNA”）を作成して必要なファイルなどをコピーしておく必要がある。

```
$ mkdir -p ~/ana/DNA
$ cp -r /opt/mlfsoft/python-utsusemi/utsusemi/DNA/environ-home/ana/*
~/ana/DNA
```

2.5. 初期設定

空蟬における初期設定は、

1. 必要なデータフォルダの作成
2. データ処理用フォルダの作成
3. 環境変数の設定

である。

なお、デフォルトのインストール先は、Table 10 のようになっている。

Table 10 Default path of MLF software installation

| デフォルト場所 | 説明 |
|---------------|---|
| /opt/mlfsoft/ | 万葉ライブラリや空蟬など、MLF で開発されたソフトウェアがインストールされる場所 |

2.5.1. データフォルダの作成

データフォルダとは RAW データを格納する場所である。

デフォルトの場所（既存の環境変数設定ファイルでの設定）は Table 11 のようになる。

Table 11 Default path of data folder

| デフォルト場所 | 説明 |
|---------|--|
| /data | 各分光器のイベントデータディレクトリが置かれる場所。この場所に解析に必要な各装置のデータを置くためのフォルダである SIK/ AMR/ などが作成される。 |

この場所は、後述する環境設定アプリ内で設定できる *DATA_DIR* 設定、もしくは環境変数 (*UTSUSEMI_DATA_DIR*) にて指定されている必要がある。

データフォルダは使用する BL のコードを使用して下記のように作成しておく。

```
$ sudo mkdir -p /data/SIK
$ sudo mkdir -p /data/AMR
$ sudo mkdir -p /data/HPN
$ sudo mkdir -p /data/SAS
$ sudo mkdir -p /data/DNA
```

デフォルトの場所は “/data” でなくても良いので、例えば下記のようにホームフォルダに作成して、環境設定アプリの *DATA_DIR* 設定で指定しても良い。

```
$ mkdir -p ~/data/SIK
```

2.5.2. データ処理用フォルダの作成

空蟬の動作には、一部のデータ処理機能が作成する一時ファイルやユーザー専用のパラメータファイルなどを保存するためのフォルダが予め作成されている必要がある。

通常はユーザーのホームフォルダに “ana” というフォルダを作成し、その下に “xml” 及び “tmp” フォルダを作成する。

```
$ mkdir -p ~/ana/xml
$ mkdir -p ~/ana/tmp
```

この “ana” フォルダのある場所（上記の例ではホームフォルダ）は、後述する環境設定アプリ内での *USR_DIR* 設定、もしくは環境変数 *UTSUSEMI_USR_DIR* にて指定しておく必要がある。

2.5.3. 環境設定

環境設定とは空蟬を動作させるのに必要な環境変数を設定するもので、下記のコマンドを実行するだけである（コマンド実行のパスに空蟬の実行ファイル場所を追加）。

```
$ export PATH=/opt/mlfsoft/python-utsusemi/utsusemi/bin/:${PATH}
```

この設定により、下記 (Table 12) の空蟬アプリケーションが実行できるようになる。

Table 12 Utsusemi applications on Ubuntu Linux

| Utsusemi アプリケーション | 説明 |
|-------------------|--------------------------|
| EnvironSetting | 【初期設定】 空蟬の各種設定を行う |
| Sequencer, Ana | SequenceEditor を起動する |
| M2PlotPlus | M2Plot を起動する |
| MPlot | MPlot を起動する |
| D4Mat | 【非弾性】 D4MatSlicer を起動する |
| D4Mat2 | 【非弾性】 D4Mat2Slicer を起動する |

なお、インストール直後は、必ず一度 EnvironSetting を実行して、次節の空蟬環境設定を行うこと。

上記の PAHT の設定を行って初期設定を終わらせれば上記のアプリケーションが起動し、そのアプリケーション上で処理が可能となる。

ただし、PATH の設定だけではターミナル上では空蟬の内部コマンドやそれらを用いたスクリプトの実行はできない。ターミナル上でそれらを行いたい場合は、下記のコマンドを実行する。

```
source /opt/mlfsoft/python-utsusemi/utsusemi/bin/_loadenv
```

これらの設定を常に有効にするには、例えば ~/.bashrc などに下記を追加しておけば良い。

```
# Utsusemi4 environments
if [ -f /opt/mlfsoft/python-utsusemi/utsusemi/bin/_loadenv ]; then
    export PATH=${PATH}:/opt/mlfsoft/python-utsusemi/utsusemi/bin
    # スクリプトをターミナルで実行する場合、以下が必要
    source /opt/mlfsoft/python-utsusemi/utsusemi/bin/_loadenv
fi
```

2.5.3.1. 空蟬環境設定

空蟬環境設定はインストール直後に必ず最低一度は実行して設定を済ませる必要がある。未設定だと空蟬アプリケーションが起動しない。インストールされたアプリ EnvironSetting を以下のように起動して行う。

```
$ EnvironSetting
```

このアプリケーションで設定できる内容は以下の通りである。

- 使用する装置環境 (BL01、BL11、BL14、BL15)
- データフォルダとデータ処理用フォルダ
- ログを残すかどうか
- GUI アプリのデフォルトフォントサイズ
- 高速化のための並列計算の数
- デバッグモードのオン・オフ

よって、違う装置 (ビームライン) のデータを解析する場合は、この環境設定で装置環境を切り替える必要がある。

環境設定の項目は Table 13 の通りである。

Table 13 Environment variables for Utsusemi

| 設定 (環境変数) | 役割 |
|-------------------|---|
| Instrument | 解析に使用するデータの装置の選択を行う。これにより、使用する RAW データフォルダ、使用できるコマンドなどが変化する。 |
| Log quiet | 解析コマンドのログを減らす場合に使用する。普段は YES で良い。 |
| Font Size | GUI アプリのフォントサイズを定義する。通常は 12 で良いだろう。大きくしたり小さくしたりはできるが、GUI パーツ (ボタンなど) やレイアウトが崩れる可能性があるので注意。 |
| Debug mode | 空蟬のデバッグモード。普段は NO (チェックボックスオフ) で良い。 |
| Number of threads | 一部の解析コマンドが使用するマルチスレッドの最大値を設定する。(通常は実行する PC のコアの数程度を設定すると良い) |
| DATA_DIR | イベントデータなど RAW データを格納するフォルダ。(前項の例では /data や ~/data) |
| USR_DIR | 空蟬が作成する一時ファイルが保存されユーザー専用のパラメータファイルが置かれるフォルダである。通常このフォルダの下に ana フォルダが置かれる。(前項の例ではホームフォルダ ~/) なお指定したフォルダが存在しない場合は作成可能であれば作成される。 |

2.5.4. (旧) 環境変数の設定

空蟬を使用する方法として環境変数自体を直接設定する方法もある。ただし手動で環境変数を変更するので、Linux に慣れたユーザー以外は非推奨である。

空蟬を動作させるのに必要な環境変数を手動で設定する。それらは BL ごとに異なり、雛形となるファイルはそれぞれ Table 14 に示す場所にある。

Table 14 Path to the environment setting file for each BL

| 対象 BL | 設定ファイル |
|---------------|--|
| BL01 (四季) | /opt/mlfsoft/python-utsusemi/utsusemi/SIK/bashrc.py3.SIK |
| BL02 (DNA) | /opt/mlfsoft/python-utsusemi/utsusemi/DNA/bashrc.py3.SIK |
| BL10 (PLANET) | /opt/mlfsoft/python-utsusemi/utsusemi/HPN/bashrc.py3.HPN |
| BL14 (アマテラス) | /opt/mlfsoft/python-utsusemi/utsusemi/AMR/bashrc.py3.AMR |
| BL15 (大観) | /opt/mlfsoft/python-utsusemi/utsusemi/SAS/bashrc.py3.SAS |

これらの雛形ファイルにおける、空蟬用環境変数は Table 15 のようになる。

Table 15 Environment variables for Utsusemi

| 環境変数 | デフォルト値 | 役割、意味 |
|--------------------|-------------------------------|---------------------------|
| UTSUSEMI_INST_CODE | <inst-code> | 装置コード (SIK,AMR など) |
| UTSUSEMI_DATA_DIR | "/data" | データフォルダの指定 |
| UTSUSEMI_USR_DIR | \${UTSUSEMI_USR_DIR-\${HOME}} | ユーザーのフォルダ (デフォルトはホームフォルダ) |
| UTSUSEMI_LOG_QUIET | n | データ処理実行時のログ表示。"Y"だとかなり減る。 |
| UTSUSEMI_MULTH | 8 | データ処理実行時のマルチプロセッシングの数 |
| UTSUSEMI_DEBUGMODE | n | デバッグ用フラグ (デフォルト n で良い) |

これらのファイルに書かれている環境変数の設定を有効にするには、以下のようなコマンドを実行する。

```
$ source /opt/mlfsoft/python-utsusemi/utsusemi/SIK/bashrc.py3.SIK
=====
Welcome to Utsusemi Environment for SIK
              ver 4.0
=====
Ana   : Sequencer
M2Plot: M2PlotPlus
D4mat : D4MatSlicer
D4mat2: D4Mat2Slicer
=====
```

ただ、このやり方では扱いにくいので、下記のような設定を、例えば”~/.bashrc”に追記するとよい。

```
# Utsusemi4 environments
# For BL01 (SIK)
if [ -f /opt/mlfsoft/python-utsusemi/utsusemi/SIK/bashrc.py3.SIK ];
then
    alias SIK="source /opt/mlfsoft/python-
utsusemi/utsusemi/SIK/bashrc.py3.SIK"
fi

# For BL02 (DNA)
if [ -f /opt/mlfsoft/python-utsusemi/utsusemi/DNA/bashrc.py3.DNA ];
then
    alias DNA="source /opt/mlfsoft/python-
utsusemi/utsusemi/DNA/bashrc.py3.DNA"
fi

# For BL11 (HPN)
if [ -f /opt/mlfsoft/python-utsusemi/utsusemi/HPN/bashrc.py3.HPN ];
then
    alias HPN="source /opt/mlfsoft/python-
utsusemi/utsusemi/HPN/bashrc.py3.HPN"
fi

# For BL14 (AMR)
if [ -f /opt/mlfsoft/python-utsusemi/utsusemi/AMR/bashrc.py3.AMR ];
then
    alias AMR="source /opt/mlfsoft/python-
utsusemi/utsusemi/AMR/bashrc.py3.AMR"
fi

# For BL15 (SAS)
if [ -f /opt/mlfsoft/python-utsusemi/utsusemi/SAS/bashrc.py3.SAS ];
then
    alias SAS="source /opt/mlfsoft/python-
utsusemi/utsusemi/SAS/bashrc.py3.SAS"
fi
```

これを”~/.bashrc”に追記したのち、有効にするために以下のようにコマンドを打つ。

```
$ source ~/.bashrc
```

もしくは、ターミナルを閉じて再度開くと有効となっている。

環境が有効になったかどうかは、それぞれの分光器の環境へ移るためのキーワードを打ち込む。キーワードは各分光器の略称である (Table 16)。

Table 16 Key codes to set environment

| 装置 | キーワード |
|--------------|-------|
| 四季 (BL01) | SIK |
| DNA (BL02) | DNA |
| プラネット (BL11) | HPN |
| アマテラス (BL14) | AMR |
| 大観 (BL15) | SAS |

例えば四季の場合なら以下のように表示される。

```
$ SIK
=====
Welcome to Utsusemi Environment for SIK
                        ver 4.0
=====
Ana   : Sequencer
M2Plot: M2PlotPlus
D4mat : D4MatSlicer
D4mat2: D4Mat2Slicer
=====
$
```

BL01 (SIK) の環境への移行が完了したことがわかるメッセージが表示される。

なお、自分の環境で個別に環境変数を設定したい場合は、それぞれの `bashrc.py3.XXX` ファイルをコピーして改変すれば良いだろう。特にデータファイルの置き場所を変更した場合、例えばホームフォルダ以下に設置する場合などには、`UTSUSEMI_DATA_DIR` の設定を変更した `bashrc.py3.XXX` を準備し、`.bashrc` 内で設定を有効にするようにすれば良いだろう。

2.6. 空蟬の実行

初期設定が正しく終わっていれば、ターミナルから空蟬アプリケーション、もしくは空蟬コマンドが実行できる。例えば空蟬の基本 GUI である `Sequencer` の起動は以下になる。

```
$ Sequencer
```

もしくは

```
$ Ana
```

そのほか前節で示したような、`M2Plot`、`D4mat`、`D4mat2` といったアプリケーションも同様に起動できる。

2.6.1. 空蟬のバージョン確認

空蟬のバージョンの確認方法を示す。バグなどの報告や、アップデートする際の目安となる。前節の空蟬環境の有効化を済ませたのちに下記のコマンドを入力する。


```
$ python3
>>> import utsusemi
>>> utsusemi.__version__
'4.0.220825'
```

もしくは

```
$ python3
>>> import UtsusemiInfo
-----
                Utsusemi Info
-----
        version : 4.0 (rev 220825)
    Contact person : Yasuhiro Inamura
        e-mail : yasuihiro.inamura@j-parc.jp
-----

Manyo>>>
```

リリース番号とバージョン番号が表示される。何か空蟬で問題が発生した場合はこの情報と一緒に連絡すること。もしくはパッケージ自体のバージョンでも良い。

```
$ sudo apt show mlf-utsusemi-with-ugao
[sudo] password for hoge:
Package: mlf-utsusemi-with-ugao
Version: 4.0.210816-ubu20
Status: install ok installed
(後略)
```

2.7. アンインストール

アンインストールを行う場合は下記のコマンドを実行すれば良い。順番はインストールの時の逆順で、mlf-utsusemi-dna、 mlf-utsusemi-sas、 mlf-utsusemi-with-ugao、最後に mlf-manyo1 である。

```
$ sudo apt remove mlf-utsusemi-dna
$ sudo apt remove mlf-utsusemi-sas
$ sudo apt remove mlf-utsusemi-with-ugao
$ sudo apt remove mlf-manyo1
```

なお、まだいくつかの関連するフォルダがあるので、必要に応じて削除する。
インストール時に作成したデータフォルダの削除は下記のように行うことができる。

```
$ rm -rf /data
```

また、一時ファイル保存フォルダは下記のように削除可能である。

```
$ rm -rf ~/ana
```

2.8. 問題が発生した場合

インストール時にエラーや予期せぬ状態になった場合、使用した OS とインストール方法を確認して手順に問題がないようなら、担当者（著者）へ状況報告をお願いしたい。

インストール後の動作に問題が発生した場合は、前節に示した空蟬のバージョンを確認し、発生した事象を担当者（著者）まで報告していただきたい。

2.8.1. その他の情報

万葉ライブラリパッケージ（mlf-manyo1）に必須のパッケージは Table 17 の通りである。

Table 17 Package names required for Manyo library installation

| パッケージ名（バージョンは省略） |
|--------------------------|
| libboost-system |
| libboost-filesystem |
| libboost-serialization |
| libboost-program-options |
| libgsl |
| libhdf5 |
| libnexus |
| libssl |

空蟬環境パッケージ（mlf-utsusemi-with-ugao）に必須のパッケージは Table 18 の通りである。

Table 18 Package names required for Utsusemi installation

| Ubuntu 20.04 用パッケージ名 (バージョンは省略) |
|------------------------------------|
| mlf-manyo1 |
| python3-matplotlib |
| python3-scipy |
| python3-pyside2.qtc core |
| python3-pyside2.q tgui |
| python3-pyside2.q twidgets |

3. Windows 用バイナリ版インストール

3.1. 概要

本章では空蟬を Windows 環境にバイナリパッケージでインストールする方法を示す。

なお、2023 年 4 月現在で、対応している MLF におけるビームラインは Table 19 の通りである。

Table 19 Available beam lines for Utsusemi on Windows

| BL(code) | 注意点 | 状況 |
|-----------|------------|------|
| BL01(SIK) | 特になし | 対応済み |
| BL14(AMR) | 特になし | 対応済み |
| BL15(SAS) | Dview は含まず | 対応済み |

3.2. 動作環境

本章で動作を保証する Windows は Table 20 の通りである。

Table 20 Available Windows version

| Windows | Architecture |
|---------|----------------|
| 10, 11 | x86_64 (amd64) |

これら以外の環境（32bit 版も含め）は今後もサポートする予定はない。

3.2.1. 事前の確認事項

アカウント名に関して注意すべき点がある。インストールする Windows の管理者ユーザーのアカウント名が英数字でなければならない（Table 21）。既に日本語を利用していた場合、新たに英数字のアカウント名でユーザーを作成し、そちらでログインしてからインストールする必要がある。新たにアカウントを作成する方法などは省略する。

Table 21 Account name examples

| アカウント名の条件 | 良い例 | 悪い例 |
|-----------|---------|-----|
| 英数字であること | Inamura | 稲村 |

3.3. 手順の概要

Windows 版におけるインストール手順は以下のような流れとなる。

1. 環境構築（Minimum タイプのみ必要）
2. インストーラによるインストール（All-in-one、Minimum タイプ）
3. 環境設定

3.4. インストール

3.4.1. インストーラのタイプ

Windows 版には、以下（Table 22）の 2 種類のインストーラがあるので、環境に合わせて利用すること。

Table 22 Type of installation

| タイプ | 説明 | ファイル名の例 |
|------------|--|--|
| All-in-one | Python 含め全て同梱されている。 (推奨) | Utsusemi.4.0.230403.x86_64.msi |
| Minimum | Python 環境はない最小限のインストーラ。Python3.9.x のみ対応。Python の環境は自分で構築する必要がある。 中・上級者向け。 | Utsusemi_minimum.4.0.230403.x86_64.msi |

3.4.2. 空蟬本体のインストール

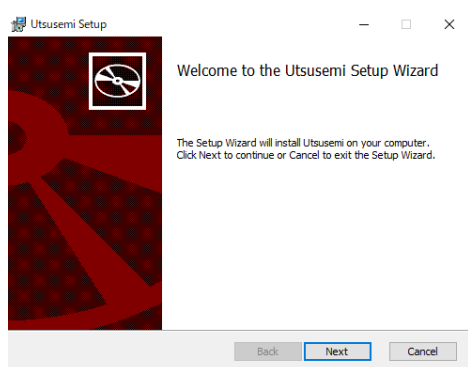
ここでは、All-in-one タイプのインストーラでのインストールを示す。なお、Minimum タイプにはあらかじめ Python 環境が必要であるが、要件は次節にて示す。

空蟬パッケージインストーラをダブルクリックしてインストールを行う。

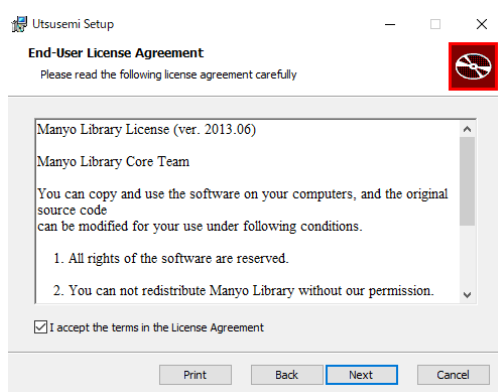


なお、既に以前のバージョンの空蟬が入っている場合、一旦以前の空蟬をアンインストールしてから行うこと。

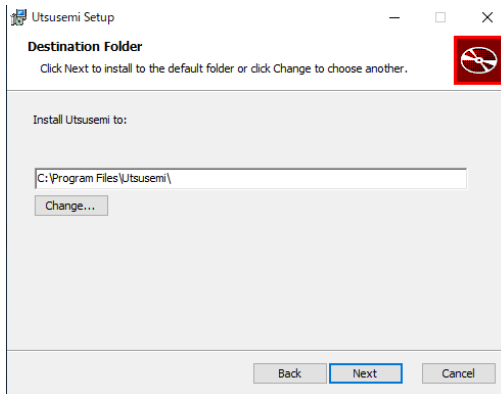
1. インストールウィザードが現れるので、以降は基本的に「Next」で進めて行く。



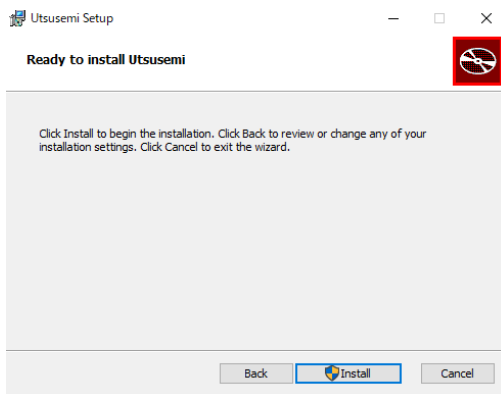
2. 使用許諾書の表示が表示されるので、同意して「Next」ボタンを押す。



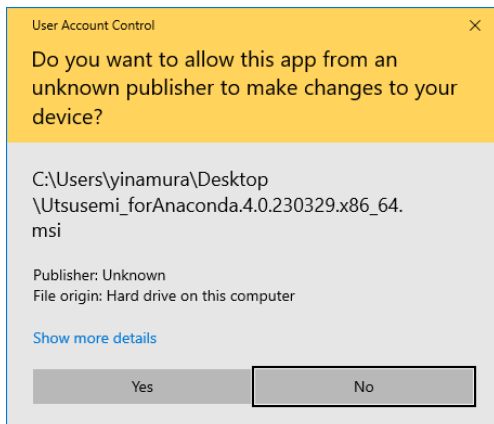
3. インストール先の確認が出る。よければ「Next」ボタンを押す。（デフォルトでは "C:\Program Files\Utsusemi\"）



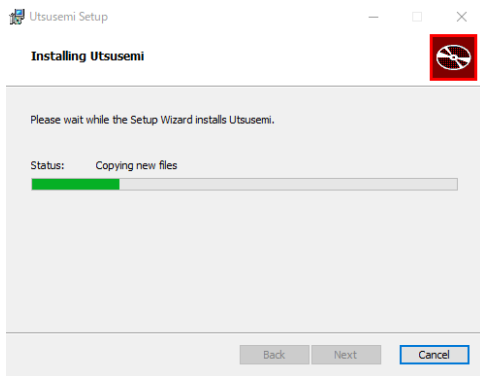
4. インストール準備完了の表示が出るので、「Install」ボタンを押す。



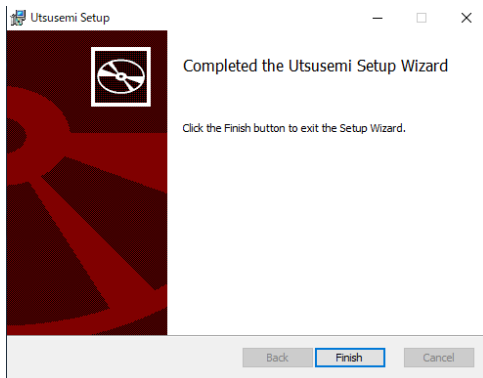
5. 途中でユーザーのアカウント制御のダイアログが出た場合、「Yes」を押す。



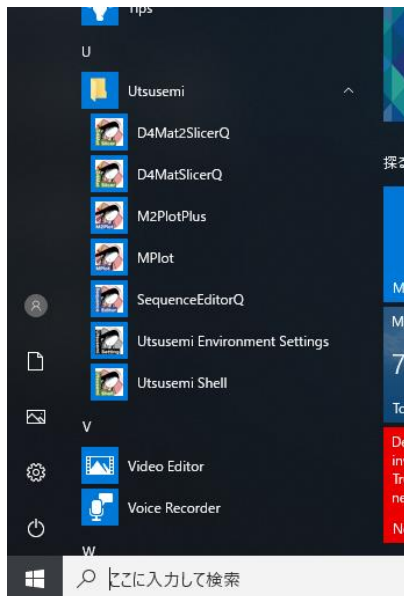
6. おおよそインストールに2～3分程度かかる。



7. 以下のようにダイアログが出ると正常に終了している。「Finish」ボタンを押して終了する。



8. スタートメニューからインストールされた Utsusemi ソフトウェアを確認できる。



3.4.2.1. Minimum タイプの場合

Minimum タイプは、Python3.9（2023 年 4 月現在）の環境にのみ対応している。

インストール前に必要なことは以下であるが、詳細は述べない（本章末尾の Appendix 参照のこと）。

- Python3.9 及び、必要なパッケージのインストール
- コマンドプロンプトから Python3.9 が実行でき、パッケージの import もできるようにしておく

なお、Python と必要なパッケージを以下（Table 23）に示す。

Table 23 list of the packages required by a minimum BinaryInstaller for win

| | |
|----------|-----------------------------------|
| Python | 3.9.x |
| Packages | numpy, scipy, matplotlib, PySide6 |

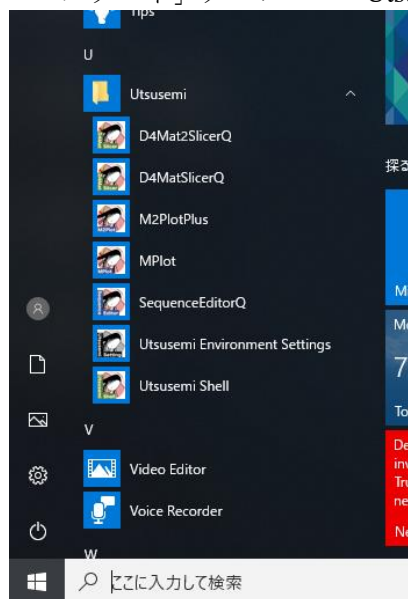
結局のところ必要となる環境は、起動したコマンドプロンプト内で Python3.9 インタプリタから下記のようなコマンドを実行してエラーが出ない環境である。

```
C:\¥Users¥hoge>python
Python 3.9.16 (main, Mar 8 2023, 10:39:24) [MSC v.1916 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license" for more information
>>> import numpy
>>> import scipy
>>> import matplotlib
>>> import PySide6
```

3.4.3. インストールされるアプリケーション

空蟬ソフトウェア群は、インストール時に指定した場所（C:\¥¥Program Files¥Utsusemi¥）にある。また、アプリケーションへのショートカットが、以下の場所に作成される。

「スタート」メニュー >> Utsusemi



ここには Table 24 に示すプログラムが準備されている。詳細は後で述べる。

Table 24 List of Utsusemi Applications installed

| Utsusemi アプリケーション | 説明 |
|-------------------------------|------------------------------|
| D4MatSlicer | D4MatSlicer を起動する。 |
| D4Mat2Slicer | D4Mat2Slicer を起動する。 |
| M2PlotPlus | M2Plot を起動する。 |
| MPlot | MPlot を起動する。 |
| SequenceEditorQ | SequenceEditor を起動する。 |
| Utsusemi Environment Settings | 【環境設定】 空蟬のフォルダ環境変数を設定する。 |
| Utsusemi Shell | Utsusemi 環境が設定されたターミナルを起動する。 |

3.5. 初期設定

空蟬における初期設定は、

1. 必要なフォルダの作成（データフォルダ、データ処理用フォルダ）
 2. 環境変数の設定
- である。

3.5.1. 必要なフォルダの作成

空蟬 Windows 版では、まず解析用データの置き場所と一時ファイル保存場所をフォルダとして準備する必要がある（Table 25）。

Table 25 Path to folders required by Utsusemi

| 設定（環境変数名） | 役割 |
|-----------------------|--|
| データフォルダ (DATA_DIR) | イベントデータなど RAW データを格納するフォルダ。 (Linux 版では"/data") |
| 解析用フォルダ (USR_DIR) | 空蟬が作成する一時ファイルの保存先となる ana/tmp などのフォルダを置く場所である。(Linux 版では"\$HOME") |

この二つのフォルダの場所は任意であるが、例えば Table 26 に示すような場所に作成する。

Table 26 Example of path to folders required by Utsusemi

| 設定 | デフォルトフォルダ |
|---------|------------------------------------|
| データフォルダ | C:\ユーザー\ユーザー名\ドキュメント\Utsusemi\DATA |
| 解析用フォルダ | C:\ユーザー\ユーザー名\ドキュメント\Utsusemi\HOME |



3.5.2. 環境設定

次に空蟬の環境の設定を Utsusemi Environment Settings で行う。なお、この作業はインストール後に一度は必ず行う必要がある。

このアプリケーションで設定できる内容は以下の通りである。

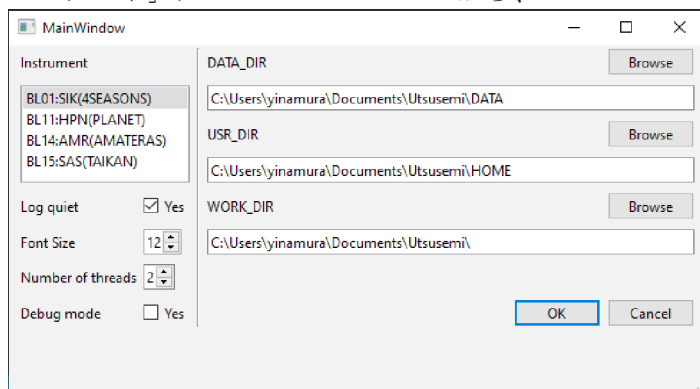
- 使用する装置環境（BL01、BL11、BL14、BL15）
- データフォルダと解析フォルダの設定
- ログを残すかどうか

- 高速化のための並列計算の数
- デバッグモードのオン・オフ

よって、違う装置（ビームライン）のデータを解析する場合は、この環境設定で装置環境を切り替える必要がある。

3.5.2.1. 環境設定の起動

「スタート」メニュー → Utsusemi から、Utsusemi Environment Settings を起動する。



3.5.2.2. 設定項目

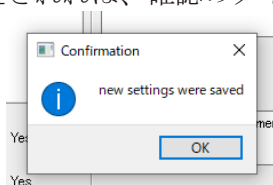
Utsusemi Environment Settings で設定できるものは Table 27 のものである。

Table 27 Environment variables for Utsusemi on Windows

| 設定（環境変数） | 役割 |
|-------------------|---|
| Instrument | 解析に使用するデータの装置の選択を行う。これにより、使用する RAW データフォルダ、使用できるコマンドなどが変化する。 |
| Log quiet | 解析コマンドのログを減らす場合に使用する。普段は YES で良い。なお macOS での空蟬の実行ログは通常あらわには表示されない。すべてのログは /tmp/mlfsoft.log に書き込まれる。 |
| Font Size | GUI アプリのフォントサイズを定義する。通常は 12 で良いだろう。大きくしたり小さくしたりはできるが、GUI パーツ（ボタンなど）やレイアウトが崩れる可能性があるので注意。 |
| Debug mode | 空蟬のデバッグモードの設定である。普段は NO（チェックボックスオフ）で良い。 |
| Number of threads | 一部の解析コマンドが使用するマルチスレッドの最大値を設定する。（通常は実行する PC のコアの数程度を設定すると良い） |
| DATA_DIR | イベントデータなど RAW データを格納するフォルダ。（Linux 版では"/data"） |
| USR_DIR | 空蟬が作成する一時ファイルの保存先となる ana/tmp などのフォルダを置く場所である。（Linux 版では"\$HOME"） |

3.5.2.3. 設定後

環境設定での設定項目の変更後、その設定を反映するために OK ボタンを押す。正しく設定されれば、確認のダイアログが表示される。



このダイアログが出ずにエラーのダイアログが出た場合は、特に指定した *DATA_DIR* と *USR_DIR* のフォルダがあるかどうかを確認すること。この設定により、*USER_DIR* の下に

- *ana* フォルダ
- その下に *tmp* フォルダ、および *xml* フォルダ

がなければ自動的に作成される。



ここで *ana* フォルダ、*tmp* や *xml* フォルダが見つからないようなら設定を見直すか、自分で作成しておくこと。これらのフォルダがない場合空蟬は正しく動作しない可能性があるので注意すること。

一方 *DATA_DIR* で設定した場所には、次節にて述べるルールでデータ（イベントデータなど）を配置する。

3.5.3. データの配置ルール

環境変数 *DATA_DIR* で指定したフォルダに、以下のようなルールで配置する。Table 28 を参照すること。

1. 装置コードと同じ名前のフォルダを作成する。（“SIK”, “AMR”, “SAS”など）
2. 装置コードのフォルダに、その装置での Run No のついたデータのフォルダを置く。（“SIK012345_20150301”など）

Table 28 Examples of beam line RAW data folder

| BL | 装置コード | 配置例 |
|-------------|-------|-----------------------------|
| BL01 四季 | SIK | DATA/SIK/SIK012345_20160401 |
| BL11 PLANET | HPN | DATA/HPN/HPN012345_20160401 |
| BL14 アマテラス | AMR | DATA/AMR/AMR012345_20160401 |
| BL15 大観 | SAS | DATA/SAS/SAS012345_20160401 |
| BL19 匠 | ENG | DATA/ENG/ENG012345_20160401 |

3.6. 空蟬の実行

環境設定が終了し解析するデータの配置も終わっていれば、アプリケーションから解析が行えるようになる。

空蟬のインストールにより、以下のアプリケーションが「スタート」メニューから使用できる。簡単な説明を行う。

Utsusemi Environment Settings

【環境設定】空蟬の環境変数を設定する。詳細は前述の「環境設定」以下を参照のこと。

MPlot, M2PlotPlus

空蟬で使用されるデータ可視化ソフトウェアである。それぞれ、1次元プロット、2次元プロットである。詳細は空蟬ユーザーマニュアルを参照のこと。Linux 版と同じ。

D4MatSlicer, D4Mat2Slicer

D4MatSlicer、および D4Mat2Slicer を起動する。あとは Linux 版と同じ。

SequenceEditorQ

SequenceEditor を起動する。

Utsusemi Shell

空蟬環境で、Shell（コマンドライン）を起動する。この上で環境設定にて指定したビームラインに応じたコマンドを呼び出すことができる。よって、自作のスクリプトを走らせることもできる。

解析に使用するアプリケーションの詳細な説明は、今後発行予定の空蟬ユーザーズマニュアルを参照のこと。

3.6.1. コマンドプロンプトからの使用

アプリケーションを利用せずに、例えば自分で作成した Python スクリプトを実行することも可能である。特に現場で作成したスクリプトや新しい解析のアイデアを形にしたコードなどを自由に実行するためには、コマンドプロンプト上で空蟬環境を利用する必要がある。その方法を以下に示す。

1. Utsusemi Shell を起動する。
2. 空蟬の設定が有効な状態でコマンドプロンプトが立ち上がるので、そのまま Python を起動するか、他のインタプリタ（例えば iPython など）を利用する。
3. 実行フォルダを変更しフォルダの移動、インタプリタの起動、スクリプトの実行を行うこともできる。

3.6.2. バージョンの確認

インストールされた空蟬のバージョンを確認するには、以下の方法がある。バグなどの報告や、アップデートする際の目安となる。インストーラファイル名には、数字が 2 種類ある。リリース番号は比較的大きな仕様の変更（機能追加）の時に付与される。同じリリース番号でも軽微な更新の場合は最終アップデート日付のみ変更される。Table 29 のようなルールになる。

Table 29 About Release number

| 例 | Utsusemi.X.Y.Z.yymmdd |
|--------|-----------------------|
| X.Y.Z | リリース番号 |
| yymmdd | 最終アップデート日付 |

一方、Utsusemi Shell を起動し、以下のようなコマンドを実行することでも情報を得られる。

```
>>> import utsusemi
>>> utsusemi.__version__
'4.0.0'
```

もしくは、以下のようなコマンドでも可能である。

```
>>> import UtsusemiInfo
-----
                Utsusemi Info
-----
version : X.Y.Z
Contact person : Yasuhiro Inamura
e-mail : xxxxx.yyyyy@j-parc.jp
-----
```

ファイル名と同様、X.Y.Z がリリース番号である。

```

C:\Program Files\Utsusemi\python>
Welcome to Utsusemi Environment for SIR
=====
C:\Program Files\Utsusemi\python>
Python 3.7.2 (tags/v0.7.2:9a3ff0492, Dec 23 2018, 23:09:28) [MSI v.1916 64 bit (AMD64)] on win32
Type 'help', 'copyright', 'credits' or 'license' for more information.
>>> report utsusemi
>>> utsusemi --version
4.0.0
>>>
>>> report Utsusemi Info
=====
Utsusemi Info
=====
version : 4.0.0
Contact person : Yasuhiro Inamura
e-mail : yasuhiro.inamura@parc.jp
=====
Margo>>>

```

3.7. アンインストール

空蟬のアンインストールを行う場合、Windows デフォルトのアンインストール（スタートメニュー → コントロールパネル → プログラム → プログラムのアンインストール）や、空蟬パッケージインストーラを起動しアンインストールを選択する。

ただし、完全には消去されず、幾つかフォルダやファイルが残される。それらは手動で消去して良い。

`C:\Program Files\Utsusemi`（フォルダ）

また、幾つかの設定ファイルがフォルダに入って残されている。次回インストールした時に初期設定として呼び出されるが、消去しても良い。

`C:\Users\ユーザーネーム\AppData\Roaming\Utsusemi`（フォルダ）

3.8. 問題が発生した場合

問題が発生する場合、以下の二つの場面が考えられる。

3.8.1. インストール直後の問題

インストールしたのにアプリが起動しない、といったインストール直後に発生している問題の大部分は、インストールに失敗しているか、環境設定が適切になされていないか、のどちらかの原因が濃厚である。一度アンインストールし、再度インストール、環境設定を行うこと。それでもエラーなどが出る場合、その状況を担当者（著者）に伝えること。

過去によくあったのが、

- 空蟬の条件（Python 3.7.x かつ 64bit）に合致していない Python がインストールされている。
- Python が 2 種類以上インストールされている。

などである。なお、2 種類インストールされていた場合、大抵どちらかが壊れている可能性が高く、正しくアンインストールできないことがある。その場合は以下のような対応を行うこと。

1. アンインストールできる Python は、アンインストールする。
2. 壊れてアンインストールできない方は、そのバージョンをインストールした Python インストーラを起動し、インストールではなく「修復」を選択し、正しくインストールされた状態にする。
3. 改めて修復した Python をアンインストールする。
4. 最後に正しい Python をインストールする。

3.8.2. コマンド実行中もしくは可視化ソフトウェア実行中の問題

想定とは異なる事象、例えばデータ処理が進まない、GUI のソフトウェアが想定される動作をしない（ボタンが効かない）などあった場合、環境と空蟬のバージョンおよび症状とターミナルに表示されたログを担当者（著者）に送付すること。

3.8.2.1. 症状の報告について

症状に関しては、

- 何をしようとして
- どのコマンド（ソフトウェア）を使用して
- 何をしたら
- どうなった

の情報を担当者（著者）に報告・送付すること。

3.9. Appendix

3.9.1. Minimum install 用環境作成例

3.9.1.1. Python.org 利用

Python.org (<https://www.python.org/>) の Python インストーラを利用して構築する例を簡便に示す。

- Python.org の Downloads (<https://www.python.org/downloads/>)から、python3.9.13（3.9 系列で Windows バイナリ配布されている最新版）をダウンロードし、インストールする。
- インストールダイアログで Add Python to PATH にチェックを入れて環境変数 PATH に設定されるようにしておく。
- Python のインストール終了後、Command Prompt アプリから、pip で numpy, scipy, matplotlib, PySide6 をインストールする。

```
python -m pip install scipy
python -m pip install matplotlib
python -m pip install PySide6
```

- Minimum Installer で空蟬をインストールする（これからは All-in-one と同じ）

3.9.1.2. Anaconda 利用

Anaconda を利用して環境を構築する例を簡便に示す。ただし Anaconda で Python 環境を作ると一部のパッケージ（PySide6）が conda で入れられず pip で入れることになるので、両者が混在（conda + pip）することで問題が発生する可能性もあるので注意すること。

最新の Anaconda では Python3.10 が使用されるので、Python3.9 を使用する環境を作成する必要がある。その過程も含めて示す。

- Anaconda をインストールする。
- Python3.9 環境を作成する。
 1. "Anaconda Navigator"を起動する。
 2. 左の欄にある"Environments"を選ぶ。
 3. "base(root)"や"Anaconda3"などが並んでいるが、下にある"Create"ボタンで新しい環境（本節では以降、"Python3.9"）を作成、その際に目的の Python のバージョン 3.9

を選択する。

- Python3.9 環境の有効化
 1. "Anaconda prompt"起動する。
 2. `conda info -e` を実行し、新しい環境（環境名 "Python3.9"）があることを確認する。
 3. `conda active Python3.9` を実行し、新しい環境に切り替える。
- パッケージのインストール
 1. "Anaconda Navigator"の"Environments"から新しい環境"Python3.9"を選択する。
 2. 左の欄を用いて、 "numpy", "scipy", "matplotlib" をインストールする。
 3. ただし、"PySide6"は存在しないので、別途 `pip` をインストールし、適当なコマンドプロンプト上で、`python -m pip install PySide6` を実行した。
- コマンドプロンプトから上記の Python3.9 環境を利用できるようにする。
 1. `path` を通すために、環境の場所を確認する（通常 ホームフォルダの下の "Anaconda3¥envs"の中にある `C:\Users\hoge\Anaconda3\envs\Python3.9` ）。
 2. 環境変数 `PATH` に、確認した Python3.9 環境のパスや、その下のパスを以下のように追加する。

```
C:¥Users¥hoge¥Anaconda3¥envs¥Python3.9¥  
C:¥Users¥hoge¥Anaconda3¥envs¥Python3.9¥Scripts  
C:¥Users¥hoge¥Anaconda3¥envs¥Python3.9¥Library¥bin
```

4. macOS (Big Sur および Monterey) 用バイナリ版インストール

4.1. 概要

本章では 2020 年から発売が開始された macOS である Big Sur(11.x)および Monterey(12.x)上で動作する空蟬のバイナリ版のインストールについて述べる。

なお、インストール用のバイナリファイルは、従来の Intel 版と、新しい Apple Silicon チップ (M1、M2 chip) 版のそれぞれで異なることに注意すること。

2022 年 11 月現在で、対応している MLF におけるビームラインは Table 30 の通りである。

Table 30 Available beam lines for Utsusemi on Windows

| BL(code) | 注意点 | 状況 |
|-----------|------------|------|
| BL01(SIK) | 特になし | 対応済み |
| BL14(AMR) | 特になし | 対応済み |
| BL15(SAS) | Dview は含まず | 対応済み |

4.2. 動作環境

以下の macOS でのみ動作検証している (Table 31)。

- Intel 用 macOS 11.3 (Big Sur) および 12.4 (Monterey)
- Apple silicon (M1) 用 macOS 12.3 (Monterey)

Table 31 Confirmed macOS versions

| macOS | 11.x(Big Sur) | 12.x(Monterey) |
|------------------------|---------------|----------------|
| Intel | Enable | Enable |
| Apple Silicon (M1, M2) | Enable | Not verified |

4.3. 手順の概要

空蟬 macOS 版のインストーラは、macOS でよく使用されるパッケージファイル (ファイル名の拡張子が.pkg) として配布される。このファイルをダブルクリックすることでアプリケーションフォルダに Utsusemi フォルダがインストールされる。

実際の手順は以下のようになる。

- macOS セキュリティの設定
- 空蟬のインストール

4.4. インストール

4.4.1. macOS セキュリティの設定

残念ながら空蟬は macOS から見て安全が保障されたソフトウェア (AppStore など配布されるような公式のアプリケーション) ではないため、実行するためには予め macOS のセキュリティの設定を行う必要がある。

注意点としては、セキュリティ設定前に、既に空蟬をインストールしていた (アプリケーションフォルダに Utsusemi フォルダがある) 場合、予め Utsusemi フォルダを削除しておく必要があることである。

1. Finder の「アプリケーション」→「ユーティリティ」の中の「ターミナル」を起動する。
2. ターミナルの中で以下のコマンドを打つ。パスワードを聞かれるので、ログインする時のパスワードを打ち込む。

```
$ sudo spctl --master-disable
```

3. macOS のシステム環境設定を開き、「セキュリティとプライバシー」設定の「一般」タブを開く。
4. 先のコマンドにより、これまで隠されていた「ダウンロードしたアプリケーションの実行許可」に「すべてのアプリケーションを許可」項目が表示されるので、それを選択する。



5. 上図の「すべてのアプリケーションを許可」項目が見えない場合は、Mac 本体を再起動すると「すべてのアプリケーションを許可」が選択できるようになるので、設定する。

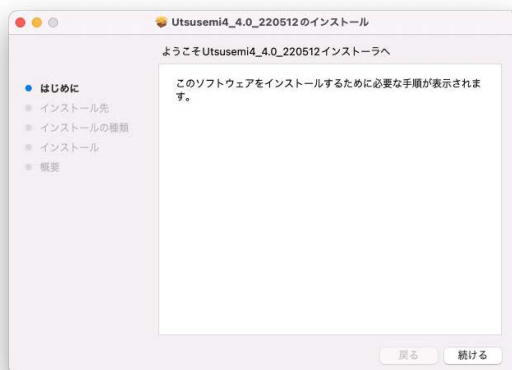
このセキュリティ設定がなされていないと、空蟬アプリの起動時に全く何も反応がないので注意すること。

4.4.2. 空蟬のインストール

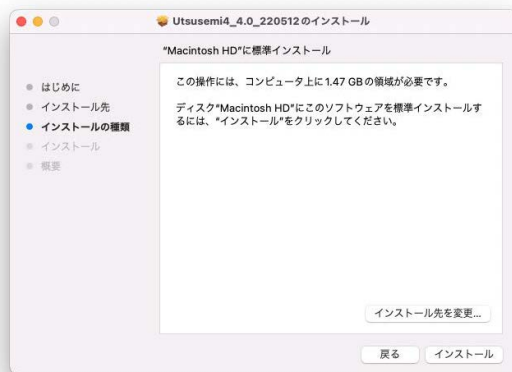
1. インストーラをダブルクリックする。



2. 後は通常のインストールダイアログが現れるので「続ける」を押す。



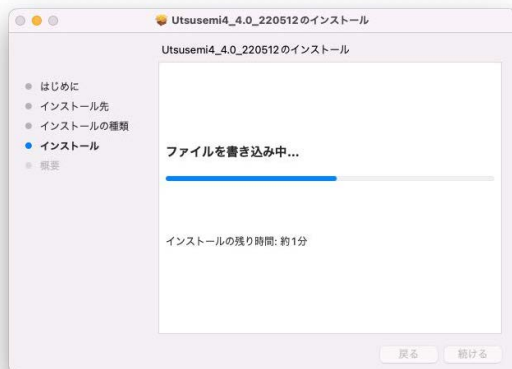
3. インストール先を変更せずに「インストール」を押す（インストール先は「アプリケーション」フォルダ）。



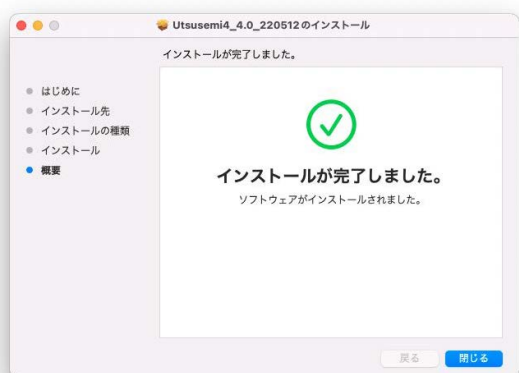
4. パスワードを入力する。



5. インストールが開始される。

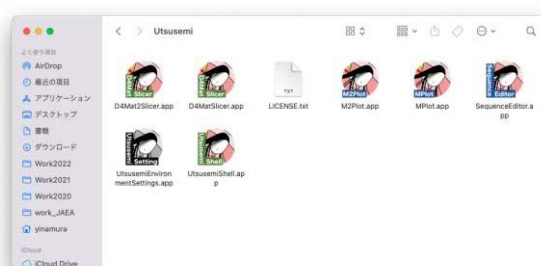


6. インストールが完了するので、「閉じる」を押す。



4.4.2.1. インストールされるアプリケーション

空蟬ソフトウェア群は、「アプリケーションフォルダ」の「Utsusemi」フォルダ内にある。もしくは LaunchPad からアクセスできる。インストールにより、以下のプログラムが準備される。



ここには Table 32 に示すプログラムが準備されている。詳細は後で述べる。

Table 32 List of Utsusemi applications

| Utsusemi アプリケーション | 説明 |
|-----------------------------|------------------------------|
| D4MatSlicer | D4MatSlicer を起動する。 |
| D4Mat2Slicer | D4Mat2Slicer を起動する。 |
| M2PlotPlus | M2Plot を起動する。 |
| MPlot | MPlot を起動する。 |
| SequenceEditor | SequenceEditor を起動する。 |
| UtsusemiEnvironmentSettings | 【環境設定】空蟬のフォルダ環境変数を設定する。 |
| UtsusemiShell | Utsusemi 環境が設定されたターミナルを起動する。 |

4.5. 初期設定

空蟬における初期設定は、

1. 必要なフォルダの作成（データフォルダ、データ処理用フォルダ）
 2. 環境変数の設定
- である。

4.5.1. 必要なフォルダの作成

空蟬 Mac 版では、まず解析用データの置き場所と一時ファイル保存場所をフォルダとして準備する必要がある（Table 33）。

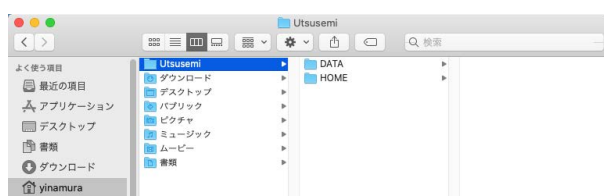
Table 33 Path to folders required by Utsusemi

| 設定（環境変数名） | 役割 |
|-----------------------|--|
| データフォルダ (DATA_DIR) | イベントデータなど RAW データを格納するフォルダである。 (Linux 版では"/data") |
| 解析用フォルダ (USR_DIR) | 空蟬が作成する一時ファイルの保存先となる ana/tmp などのフォルダを置く場所である。(Linux 版では"\$HOME") |

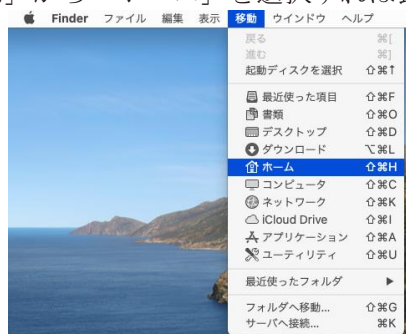
この二つのフォルダは本来どこにあっても良いが、macOS Catalina (10.15) 以降のセキュリティ強化の結果、特にデータフォルダをデスクトップや書類フォルダ、共有フォルダに置くとアクセスできないことがある。したがって空蟬の macOS 版ではデフォルトの場所はホームフォルダ直下 (Users/<user name>/) としている。環境設定 Utsusemi Environment Setting（詳細は次節）の初回起動時にこの場所に自動的に作成される（Table 34）。この場所で不都合がある場合は、別途用意すること。

Table 34 Example of path to folders required by Utsusemi

| 必要なフォルダ | デフォルトの場所（インストール時に自動生成） |
|---------|------------------------|
| データフォルダ | ホームフォルダ/Utsusemi/DATA |
| 解析用フォルダ | ホームフォルダ/Utsusemi/HOME |

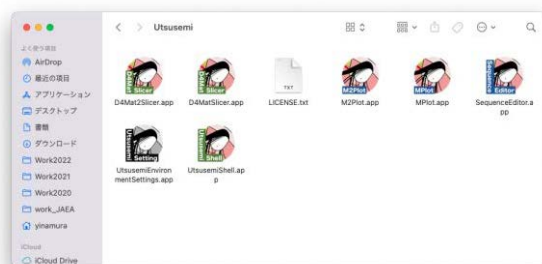


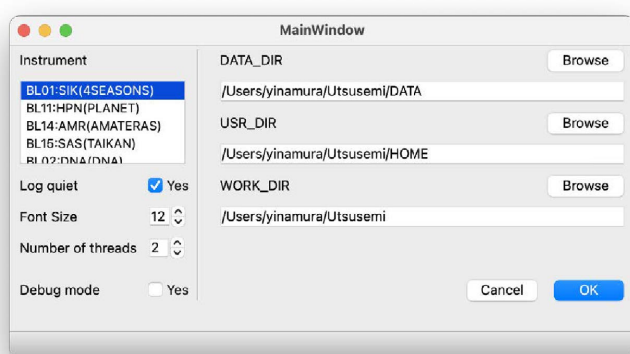
ホームフォルダの場所がよくわからない場合は、下記のようにデスクトップのメニュー「移動」から「ホーム」を選択すれば良い。



4.5.2. 環境設定

次に空蟬の環境の設定を Utsusemi Environment Settings で行う。「アプリケーション」フォルダ → Utsusemi もしくは LaunchPad から UtsusemiEnvironmentSettings を起動する。





なお、この作業は、インストール後に一度は必ず行う必要がある。

このアプリケーションで設定できる内容は以下の通りである。

- 使用する装置環境（BL01、BL11、BL14、BL15）
- データフォルダと解析フォルダの設定
- ログを残すかどうか
- GUI アプリのデフォルトフォントサイズ
- 高速化のための並列計算の数
- デバッグモードのオン・オフ

よって、違う装置（ビームライン）のデータを解析する場合は、この環境設定で装置環境を切り替える必要がある。

4.5.2.1. 設定項目

UtsusemiEnvironmentSettings で設定できるものは以下（Table 35）のものである。

Table 35 Environment Variables for Utsusemi on macOS after bigsur

| 設定（環境変数） | 役割 |
|-------------------|---|
| Instrument | 解析に使用するデータの装置の選択を行う。これにより、使用する RAW データフォルダ、使用できるコマンドなどが変化する。 |
| Log quiet | 解析コマンドのログを減らす場合に使用する。普段は YES で良い。なお macOS での空蟬の実行ログは通常あらわには表示されない。すべてのログは /tmp/mlfsoft.log に書き込まれる。 |
| Font Size | GUI アプリのフォントサイズを定義する。通常は 12 で良いだろう。大きくしたり小さくしたりはできるが、GUI パーツ（ボタンなど）やレイアウトが崩れる可能性があるので注意。 |
| Debug mode | 空蟬のデバッグモード。普段は NO（チェックボックスオフ）で良い。 |
| Number of threads | 一部の解析コマンドが使用するマルチスレッドの最大値を設定する（通常は実行する PC のコアの数程度を設定すると良い） |
| DATA_DIR | イベントデータなど RAW データを格納するフォルダ。（Linux 版では "/data"） |
| USR_DIR | 空蟬が作成する一時ファイルの保存先となる ana/tmp などのフォルダを置く場所である。（Linux 版では "\$HOME"） |

4.5.2.2. 設定後

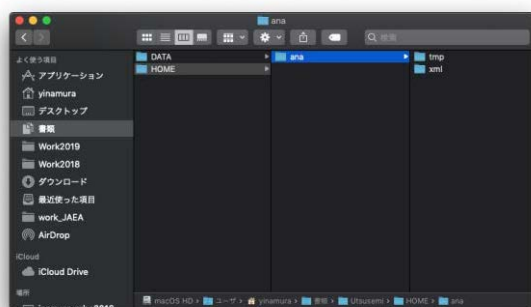
設定項目を反映するために「OK」ボタンを押す。正しく設定されれば、確認のダイアログが表示される。



このダイアログが出ずにエラーのダイアログが出た場合は、特に指定した *DATA_DIR* と *USR_DIR* のフォルダがあるかどうかを確認すること。この設定により、*USER_DIR* の下に

- *ana* フォルダ
- この下に *tmp* フォルダ、および *xml* フォルダ

がなければ自動的に作成される。



なお、*ana* フォルダや *tmp* フォルダが見つからないようなら設定を見直すか、自分で作成しておくこと。これらのフォルダがない場合空蟬は正しく動作しない可能性がある。

一方、*DATA_DIR* で設定した場所には、次節にて述べるルールで、データ（イベントデータなど）を配置する。

4.5.2.3. データの配置ルール

環境変数 *DATA_DIR* で指定したフォルダに、以下のようなルールで配置する。Table 36 を参照のこと。

1. 装置コードと同じ名前のフォルダを作成する。四季（BL01）なら"SIK"、アマテラス（BL14）なら"AMR"である。
2. 装置コードのフォルダに、その装置での Run No の付いたデータのフォルダを置く。四季（BL01）なら、例えば"SIK012345_20150301"などである。

Table 36 Examples of beam line RAW data folder

| BL | 装置コード | 配置例 |
|-------------|-------|-----------------------------|
| BL01 四季 | SIK | DATA/SIK/SIK012345_20160401 |
| BL11 PLANET | HPN | DATA/HPN/HPN012345_20160401 |
| BL14 アマテラス | AMR | DATA/AMR/AMR012345_20160401 |
| BL15 大観 | SAS | DATA/SAS/SAS012345_20160401 |
| BL19 匠 | ENG | DATA/ENG/ENG012345_20160401 |

4.6. 空蟬の実行

環境設定が終了し、解析するデータの配置も終わったなら、アプリケーションやターミナルから解析が行えるようになる。

4.6.1. アプリから起動

空蟬のインストールにより、以下のアプリケーションも「アプリケーションフォルダ / Utsusemi フォルダ」もしくは Launchpad から起動できるようになる。

SequenceEditor

空蟬でデータ処理作業のメインとなるアプリである SequenceEditor を起動する。

UtsusemiEnvironmentSettings

【環境設定】空蟬の環境変数を設定する。詳細は前述の「環境設定」以下を参照のこと。

MPlot, M2PlotPlus

空蟬で使用されるデータ可視化ソフトウェアである。それぞれ、1次元プロット、2次元プロットである。詳細は空蟬ユーザーマニュアルを参照のこと。Linux 版と同じ。

D4MatSlicer, D4Mat2Slicer

D4MatSlicer、および D4Mat2Slicer を起動する。あとは Linux 版と同じ。

UtsusemiShell

空蟬環境で、Shell（コマンドライン）を起動する。この上で環境設定にて指定したビームラインに応じたコマンドを呼び出すことができる。よって、自作のスクリプトを走らせることも可能である。

解析に使用するアプリケーションの詳細な説明は、今後発行予定の空蟬ユーザーズマニュアルを参照のこと。なお、執筆時点（2023 年 10 月）でいくつかのアプリを単独起動させるとファイル読み込み（ヒストグラム化処理や D4Mat データ）でエラーが発生することが判明している。よって SequenceEditor や D4mat、D4mat2 に関してはアプリ起動時にターミナルが必ず開く仕様となっている。この連動したターミナルを閉じると GUI アプリも落ちるので注意すること。

4.6.2. UtsusemiShell から起動

Linux での処理に慣れている場合、空蟬の全機能を使うために UtsusemiShell から各種ソフトウェアを起動することもできる。空蟬環境でターミナルが立ち上がるので、ここで各種空蟬のコマンドや機能を利用することが可能となる。スクリプトを走らせたり、GUI を起動したりできる。

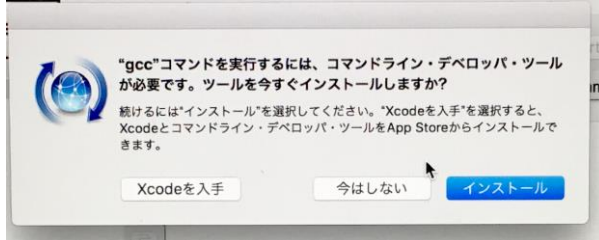


```
=====
Welcome to Utsusemi Environment for SIK
=====
Ana    : Sequencer
M2Plot: M2PlotPlus
D4mat  : D4MatSlicer
D4mat2: D4Mat2Slicer
=====
hoge@localhost %
```

ここで、Ana と打てば SequenceEditor が起動する。他にも M2Plot や非弾性散乱装置であれば D4mat や D4mat2 など使用可能である。

4.6.3. 起動時の注意

何らかのアプリケーションを実行した際に下記のようなダイアログが開くことがある。



この場合、「インストール」を選択すると、インターネットに接続していれば自動的に最小限必要なソフトウェアがダウンロードされる。「キャンセル」を選択もできるが、次回以降も同じダイアログが開くことになる。なお「Xcodeを入手」を選ぶと、Mac や iPhone のアプリを作成するキットソフトウェア一式が導入され膨大な時間がかかるので気をつけること。

4.6.4. ターミナルからの使用

アプリケーションを利用せずに、例えば自分で作成した Python スクリプトを実行することも可能である。特に現場で作成したスクリプト、新しい解析のアイデアを形にしたものなどを自由に実行するためには、

- A. UtsusemiShell を起動して利用する。
- B. ターミナルアプリ上で、空蟬環境を利用する。

のやり方がある。A. UtsusemiShell に関しては前項で少し触れたが、これを用いるとシンプルな設定でターミナルが起動する。先に示した SequenceEditor の起動だけではなく、いくつかのアプリもここから起動することも可能である。

既に自分でターミナルの設定を変更していてその環境で空蟬環境も利用する場合は、B の MacOS のターミナル上で空蟬環境を利用する必要がある。その方法を以下に示す。

1. ターミナルを起動する。
2. 以下のコマンドを実行する。

```
$ source /Applications/Utsusemi/.opt/bin/_loadenv
```

3. これで直接万葉ライブラリや空蟬の関数を呼び出すことができる。

```
$ python3
Python 3.9.12 (main, Apr 18 2022, 11:57:03)
[Clang 13.1.6 (clang-1316.0.21.2.3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import Manyo
Manyo>>>
```

4.6.5. バージョンの確認

インストールされた空蟬のバージョンを確認するには、以下の方法がある。バグなどの報告や、アップデートする際の目安となる。

インストーラファイルの名前に含まれている数字で判断する場合、Table 37 に示すルールになる。

Table 37 About release number

| 例 | Utsusemi_X.Y.Z_nnnn |
|-------|---------------------|
| X.Y.Z | リリース番号 |
| nnnn | 最終アップデート日付 |

一方、Utsusemi Shell を起動し、以下のようなコマンドを実行することでも情報を得られる。

```
>>> import utsusemi
>>> utsusemi.__version__
'4.0.0'
```

もしくは、以下のようなコマンドでも可能である。

```
>>> import UtsusemiInfo

-----
                Utsusemi Info
-----

version : X.Y.Z (rev nnnnnn)
Contact person : Yasuhiro Inamura
      e-mail : xxxxx.yyyyy@j-parc.jp
-----
```

ファイル名と同様、X.Y.Z がリリース番号、nnnn が最終アップデート日付である。

4.7. アンインストール

アンインストールを行う場合は、アプリケーションフォルダ内の、Utsusemi フォルダをゴミ箱へ捨てるだけである。ただし完全には消去されず、幾つかフォルダやファイルが残される。これらも消去したい場合は手動で行うこと。

DATA DIR フォルダ

USR DIR フォルダ

また、幾つかの設定ファイルが残されている。次回インストールした時に初期設定として呼び出されるが、消去しても良い。

ホームフォルダ/Library/Utsusemi フォルダ

```
$ rm -rf $HOME/Library/Utsusemi
```

ただし、これは不可視フォルダなので、通常目にはしない。

4.8. 問題が発生した場合

想定とは異なる事象、例えばデータ処理が進まない、GUI のソフトウェアが想定される動作をしない（ボタンが効かない）などあった場合、空蟬のバージョンと症状、ログファイルを担当者（著者）に送付すること。

4.8.1. 症状の情報について

症状に関しては、

- 何をしようとして
- どのコマンド（ソフトウェア）を使用して
- 何をしたら
- どうなった

の情報を整理すること。

4.8.2. ログファイルについて

macOS 版では、データ処理中のログは、ターミナル経由で起動するアプリ (SequenceEditor、D4mat、D4mat2) ではターミナル上に表示されるので、それをコピーする。

一方それ以外のアプリでは/tmp/mlfsoft.log に書き込まれるので、このログファイルを取り出すこと。ただし、通常のファインダからはアクセスできないので、ターミナルから

```
$ cp /tmp/mlfsoft.log ~/Desktop
```

として、デスクトップにコピーするか、ファインダのメニュー「移動」から「フォルダの移動...」を選んで出てきたダイアログに、/tmp/と打ち込めばこのフォルダにアクセスできるようになる。

5. macOS (Catalina、Mojave および High Sierra) 用バイナリ版インストール

5.1. 概要

本章では 2022 年 12 月現在サポートが終了（セキュリティアップデートが終了）となっている古い macOS 上で動作する空蟬のバイナリ版のインストールについて述べる。

2022 年 11 月現在で、対応している MLF におけるビームラインは Table 38 の通りである。

Table 38 Available beam lines for Utsusemi on Windows

| BL(code) | 注意点 | 状況 |
|-----------|------------|------|
| BL01(SIK) | 特になし | 対応済み |
| BL14(AMR) | 特になし | 対応済み |
| BL15(SAS) | Dview は含まず | 対応済み |

5.2. 動作環境

基本的に以下の macOS でのみ動作する。

- 10.15 (macOS Catalina)
- 10.14 (macOS Mojave)
- 10.13 (macOS High Sierra)

5.3. 手順の概要

空蟬のインストーラは、macOS でよく使用されるパッケージファイル（ファイル名の拡張子が .pkg）として配布される。このファイルをダブルクリックすることで「アプリケーション」フォルダに Utsusemi フォルダがインストールされる。

実際の手順は以下のようになる。

- macOS セキュリティの設定
- 空蟬インストール

5.4. インストール

5.4.1. macOS セキュリティの設定

残念ながら空蟬は macOS から見て安全が保障されたソフトウェア（AppStore など配布されるような公式のアプリケーション）ではないため、実行するためには予め macOS のセキュリティの設定を行う必要がある。なお、セキュリティ設定前に、既に空蟬をインストールしていた（アプリケーションフォルダに Utsusemi フォルダがある）場合、予め Utsusemi フォルダを削除しておくこと。

1. Finder の「アプリケーション」→「ユーティリティ」の中の「ターミナル」を起動する。
2. ターミナルの中で以下のコマンドを打つ。パスワードを聞かれるので、ログインする時のパスワードを打ち込む。

```
$ sudo spctl --master-disable
```

3. macOS のシステム環境設定を開き、「セキュリティとプライバシー」設定の「一般」タブを開く。

4. 先のコマンドにより、これまで隠されていた「ダウンロードしたアプリケーションの実行許可」に「すべてのアプリケーションを許可」項目が表示されるので、それを選択する。

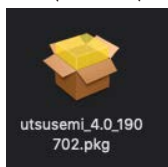


5. 上図の「すべてのアプリケーションを許可」項目が見えない場合は、Mac 本体を再起動すると「すべてのアプリケーションを許可」が選択できるようになるので、設定する。

このセキュリティ設定がなされていないと、空蟬アプリの起動時に全く何も反応がないので注意すること。

5.4.2. 空蟬のインストール

1. インストーラをダブルクリックする。



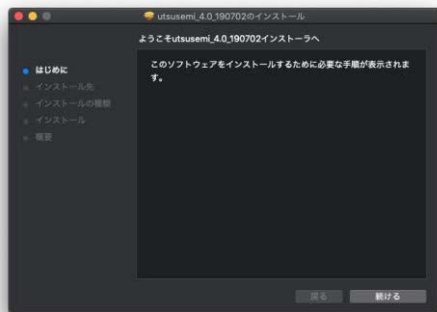
ダブルクリックした時に下記のような「開発元が未確認のため開けません」とダイアログが出た場合は、インストーラを右クリック、もしくは Control キーを押しながらアイコンをクリックして現れるショートカットメニューから「開く」を選択する。



すると、下記のような開いて良いかを尋ねるダイアログが現れるので「開く」を選択する。



2. 後は通常のインストールダイアログが現れるので「続ける」を押す。



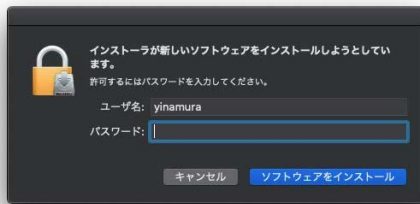
3. 場合によってはインストール先を聞かれることもあるので選択して「続ける」を押す。



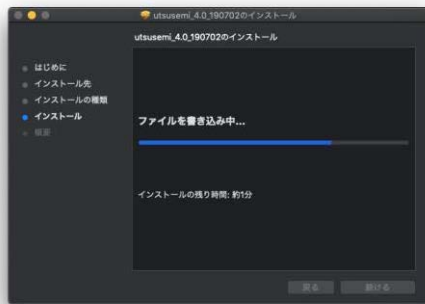
4. インストール先を変更せずに「インストール」を押す（インストール先は「アプリケーション」フォルダ）。



5. パスワードを入力する。



6. インストールが開始される。

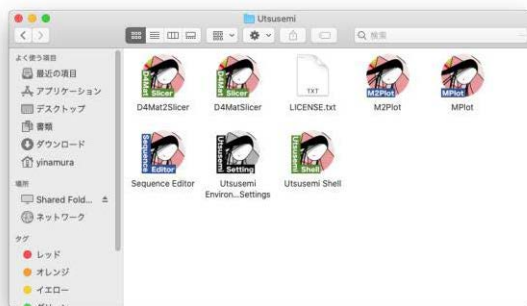


7. インストールが完了するので、「閉じる」を押す。



5.4.2.1. インストールされるアプリケーション

空蟬ソフトウェア群は、「アプリケーションフォルダ」の「Utsusemi」フォルダ内にある。もしくは LaunchPad からアクセスできる。インストールにより、以下のプログラムが準備される。



ここには Table 39 に示すプログラムが準備されている。詳細は後で述べる。

Table 39 Installed Utsusemi applications

| Utsusemi アプリケーション | 説明 |
|-------------------------------|------------------------------|
| D4MatSlicer | D4MatSlicer を起動する。 |
| D4Mat2Slicer | D4Mat2Slicer を起動する。 |
| M2PlotPlus | M2Plot を起動する。 |
| MPlot | MPlot を起動する。 |
| SequenceEditorQ | SequenceEditor を起動する。 |
| Utsusemi Environment Settings | 【環境設定】空蟬のフォルダ環境変数を設定する。 |
| Utsusemi Shell | Utsusemi 環境が設定されたターミナルを起動する。 |

5.5. 初期設定

空蟬における初期設定は、

1. 必要なフォルダの作成（データフォルダ、データ処理用フォルダ）
 2. 環境変数の設定
- である。

5.5.1. 必要なフォルダの作成

空蟬 Mac 版では、まず解析用データの置き場所と一時ファイル保存場所をフォルダとして準備する必要がある（Table 40）。

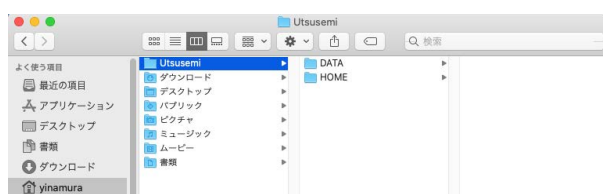
Table 40 Path to folders required by Utsusemi

| 設定（環境変数名） | 役割 |
|-----------------------|--|
| データフォルダ (DATA_DIR) | イベントデータなど RAW データを格納するフォルダ。 (Linux 版では"/data") |
| 解析用フォルダ (USR_DIR) | 空蟬が作成する一時ファイルを保存する ana/tmp などのフォルダを置く場所である。(Linux 版では"\$HOME") |

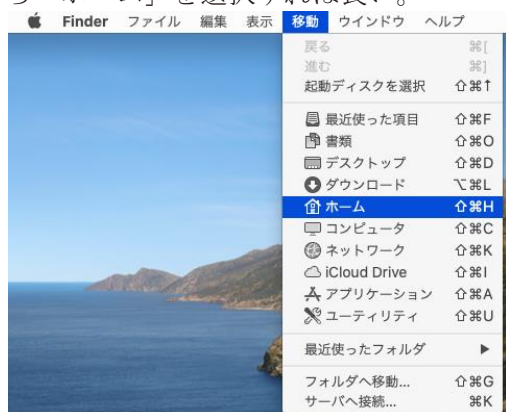
この二つのフォルダは本来どこにあっても良いが、macOS Catalina (10.15) 以降のセキュリティ強化の結果、特にデータフォルダをデスクトップや書類フォルダ、共有フォルダに置くとアクセスできないことがある。したがって空蟬の macOS 版ではデフォルトの場所はホームフォルダ直下 (/Users/<user name>/) としている（Table 41）。環境設定 Utsusemi Environment Setting（詳細は次節）の初回起動時にこの場所に自動的に作成される。この場所で不都合がある場合は、別途用意すること。

Table 41 Example of path to folders required by Utsusemi

| 必要なフォルダ | デフォルトの場所（インストール時に自動生成） |
|---------|------------------------|
| データフォルダ | ホームフォルダ/Utsusemi/DATA |
| 解析用フォルダ | ホームフォルダ/Utsusemi/HOME |

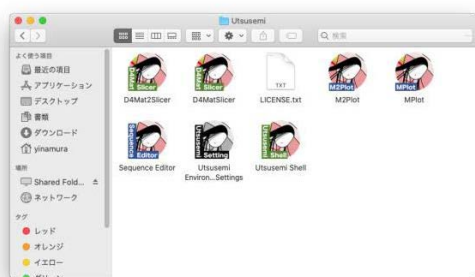


ホームフォルダの場所がわからない場合は、下記のようにデスクトップのメニュー「移動」から「ホーム」を選択すれば良い。

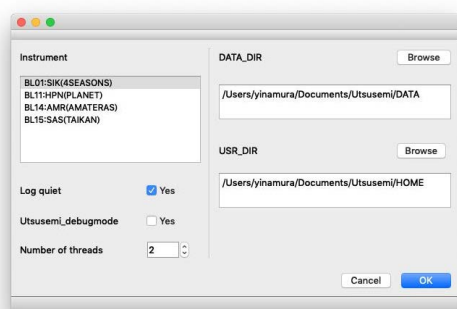


5.5.2. 環境設定

次に空蟬の環境の設定を Utsusemi Environment Settings で行う。「アプリケーション」フォルダ → Utsusemi もしくは LaunchPad から



Utsusemi Environment Settings を起動する。なお、この作業はインストール後に一度は必ず行う必要がある。



このアプリケーションで設定できる内容は以下の通りである。

- 使用する装置環境 (BL01、BL11、BL14、BL15)
- データフォルダと解析フォルダの設定
- ログを残すかどうか
- 高速化のための並列計算の数
- デバッグモードのオン・オフ

よって、違う装置 (ビームライン) のデータを解析する場合は、この環境設定で装置環境を切り替える必要がある。

5.5.2.1. 設定項目

Table 42 に示す項目が設定できる。

Table 42 Environment Variables for Utsusemi on macOS before 10.15 (macOS Catalina)

| 設定 (環境変数) | 役割 |
|-------------------|--|
| Instrument | 解析に使用するデータの装置の選択を行う。これにより、使用する RAW データフォルダ、使用できるコマンドなどが変化する。 |
| Log quiet | 解析コマンドのログを減らす場合に使用する。普段は YES で良い。なお macOS での空蟬の実行ログは通常あらわには表示されない。すべてのログは/tmp/mlfsoft.log に書き込まれる。 |
| Debug mode | 空蟬のデバッグモード。普段は NO (チェックボックスオフ) で良い。 |
| Number of threads | 一部の解析コマンドが使用するマルチスレッドの最大値を設定する。(通常は実行する PC のコアの数程度を設定すると良い) |
| DATA_DIR | イベントデータなど RAW データを格納するフォルダ。(Linux 版では "/data") |
| USR_DIR | 空蟬が作成する一時ファイルの保存先となる ana/tmp などのフォルダを置く場所である。(Linux 版では "\$HOME") |

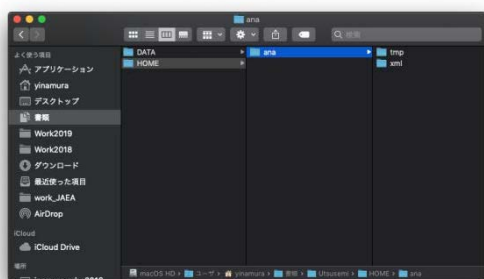
5.5.2.2. 設定後

設定項目を反映するために OK ボタンを押す。正しく設定されれば、確認のダイアログが表示される。



このダイアログが出ずにエラーのダイアログが出た場合は、特に指定した DATA_DIR と USR_DIR のフォルダがあるかどうかを確認すること。この設定により、USER_DIR の下に

- ana フォルダ
 - この下に tmp フォルダ、および xml フォルダ
- がなければ自動的に作成される。



なお、ana フォルダや tmp フォルダが見つからないようなら設定を見直すか、自分で作成しておくこと。これらのフォルダがない場合空蟬は正しく動作しない可能性があるので注意すること。

一方、*DATA_DIR* で設定した場所には、次節にて述べるルールで、データ（イベントデータなど）を配置する。

5.5.2.3. データの配置ルール

環境変数 *DATA_DIR* で指定したフォルダに、以下のようなルールで配置する。Table 43 を参照すること。

1. 装置コードと同じ名前のフォルダを作成する。（"SIK"、"AMR"、"SAS"など）
2. 装置コードのフォルダに、その装置での Run No の付いたデータのフォルダを置く。
("SIK012345_20150301"など)

Table 43 Examples of beam line RAW data folder

| BL | 装置コード | 配置例 |
|-------------|-------|-----------------------------|
| BL01 四季 | SIK | DATA/SIK/SIK012345_20160401 |
| BL11 PLANET | HPN | DATA/HPN/HPN012345_20160401 |
| BL14 アマテラス | AMR | DATA/AMR/AMR012345_20160401 |
| BL15 大観 | SAS | DATA/SAS/SAS012345_20160401 |
| BL19 匠 | ENG | DATA/ENG/ENG012345_20160401 |

5.6. 空蟬の実行

環境設定が終了し、解析するデータの配置も終わったなら、アプリケーションやターミナルから解析が行えるようになる。

5.6.1. アプリから起動

空蟬のインストールにより、以下のアプリケーションが「アプリケーションフォルダ/Utsusemi フォルダ」もしくは Launchpad から起動できるようになる。

Utsusemi Environment Settings

【環境設定】空蟬の環境変数を設定する。詳細は前述の「環境設定」以下を参照のこと。

MPlot, M2PlotPlus

空蟬で使用されるデータ可視化ソフトウェアである。それぞれ、1次元プロット、2次元プロットである。詳細は空蟬ユーザーマニュアルを参照のこと。Linux 版と同じ。

D4MatSlicer, D4Mat2Slicer

D4MatSlicer、および D4Mat2Slicer を起動する。あとは Linux 版と同じ。

SequenceEditorQ

SequenceEditor を起動する。

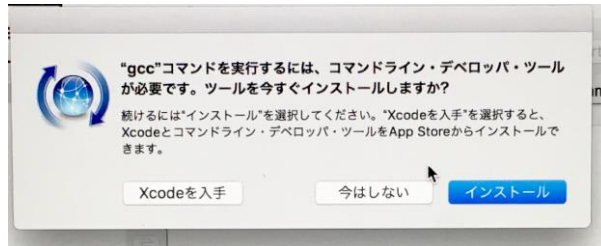
Utsusemi Shell

空蟬環境で、Shell（コマンドライン）を起動する。この上で環境設定にて指定したビームラインに応じたコマンドを呼び出すことができる。よって、自作のスクリプトを走らせることもできる。

解析に使用するアプリケーションの詳細な説明は、今後発行予定の空蟬ユーザーズマニュアルを参照のこと。

5.6.1.1. 起動時の注意

何らかのアプリケーションを実行した場合、下記のようなダイアログが開くことがある。



この場合、「インストール」を選択すると、インターネットに接続していれば自動的に最小限必要なソフトウェアがダウンロードされる。「キャンセル」を選択もできるが、次回以降も同じダイアログが開くことになる。なお「Xcode を入手」を選ぶと、Mac や iPhone のアプリを作成するキットソフトウェア一式が導入され膨大な時間がかかるので気をつけること。

5.6.2. ターミナルからの使用

アプリケーションを利用せずに、例えば自分で作成した Python スクリプトを実行することも可能である。特に現場で作成したスクリプト、新しい解析のアイデアを形にしたものなどを自由に実行するためには、

- A. Utsusemi Shell を起動して利用する
- B. ターミナルアプリ上で、空蟬環境を利用する

のやり方がある。Utsusemi Shell を用いると、シンプルな設定でターミナルが起動する。既に自分でターミナルの設定を変更していてその環境で空蟬環境も利用する場合は、B の MacOS のターミナル上で空蟬環境を利用する必要がある。その方法を以下に示す。

1. ターミナルを起動する。
2. 以下のコマンドを実行する。

```
$ source /Applications/Utsusemi/.opt/bin/_loadenv
```

3. これで直接万葉ライブラリや空蟬の関数を呼び出すことができる。

```
$ python3
Python 3.7.2 (default, Mar 5 2019, 20:00:04)
[Clang 6.0.1 (tags/RELEASE_601/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import Manyo
Manyo>>>
```

5.6.3. バージョンの確認

インストールされた空蟬のバージョンを確認するには、以下の方法がある。バグなどの報告や、アップデートする際の目安となる。インストーラファイルの名前に含まれている数字で判断する場合、Table 44 のようなルールになる。

Table 44 About Release number

| 例 | Utsusemi_X.Y.Z_nnnn |
|-------|---------------------|
| X.Y.Z | リリース番号 |
| nnnn | 最終アップデート日付 |

一方、Utsusemi Shell を起動し、以下のようなコマンドを実行することでも情報を得られる。

```
>>> import utsusemi
>>> utsusemi.__version__
'4.0.0'
```

もしくは、以下のようなコマンドでも可能である。

```
>>> import UtsusemiInfo

-----
                Utsusemi Info
-----

version : X.Y.Z (rev nnnnnn)
Contact person : Yasuhiro Inamura
e-mail : xxxxx.yyyyyy@j-parc.jp
-----
```

ファイル名と同様、X.Y.Z がリリース番号、nnnn が最終アップデート日付である。

5.7. アンインストール

アンインストールを行う場合は、アプリケーションフォルダ内の、Utsusemi フォルダをゴミ箱へ捨てるだけである。

ただし、完全には消去されず、幾つかフォルダやファイルが残される。これらも消去したい場合は手動で行うこと。

- DATA_DIR フォルダ
- USR_DIR フォルダ

また、幾つかの設定ファイルが残されている。次回インストールした時に初期設定として呼び出されるが、消去しても良い。

- ホームフォルダ/Library/Utsusemi フォルダ

```
$ rm -rf $HOME/Library/Utsusemi
```

ただし、これは不可視フォルダなので、通常目にはしない。

5.8. 問題が発生した場合

想定とは異なる事象、例えばデータ処理が進まない、GUI のソフトウェアが想定される動作をしない（ボタンが効かない）などあった場合、症状とログファイルを担当者（著者）に送付すること。

5.8.1. 症状の情報について

症状に関しては、

- 何をしようとして
- どのコマンド（ソフトウェア）を使用して
- 何をしたら
- どうなった

の情報を整理すること。

5.8.2. ログファイルについて

macOS 版では、データ処理中のログは全て/tmp/mlfsoft.log に書き込まれるので、このログファイルを取り出すこと。ただし、通常のコマンドからはアクセスできないので、ターミナルから

```
$ cp /tmp/mlfsoft.log ~/Desktop
```

として、デスクトップにコピーするか、ファインダのメニュー「移動」から「フォルダの移動...」を選んで出てきたダイアログに、`/tmp/`と打ち込めばこのフォルダにアクセスできるようになる。

6. Ubuntu 18.04 LTS および 20.04 LTS への汎用インストール

6.1. 概要

本章では比較的様々なバージョンの Ubuntu Linux へ空蟬をインストールために作成された汎用的なスクリプトを利用する方法を示す。このスクリプトにより、本来インストールに必要なソースコードのコンパイル、必要なライブラリの別途インストールなどを Ubuntu Linux のバージョンに依存せず同一の方法で行うことができる。

なお、2022 年 11 月現在で、対応している MLF におけるビームラインは Table 45 の通りである。

Table 45 Available Beam lines for script installation for Ubuntu

| BL(code) | 注意点 |
|-----------|--------------|
| BL01(SIK) | 特になし |
| BL02(DNA) | QENSfit は含まず |
| BL14(AMR) | 特になし |
| BL11(HPN) | Teuse は含まず |
| BL15(SAS) | Dview は含まず |

6.2. 動作環境

本章で使用するスクリプトによるインストールの対象 OS は Table 46 に示す通りである。

Table 46 Available Ubuntu Linux version for general installation script

| Ubuntu Linux | Architecture |
|--------------|-----------------------|
| 20.04 LTS | x86_64 (amd64), arm64 |
| 18.04 LTS | |

6.2.1. 事前の確認事項

Ubuntu は目的の PC に既に導入されていることを想定している。また極力 OS に付属するパッケージを利用するため、その OS に対応したインストール用レポジトリにアクセスできる、つまり PC がネットに接続された状態にしておくこと。

他の OS (CentOS など) にインストールしたい場合は、別項のマニュアルを参照のこと。

また、インストール途中でインターネットへの接続を行うので、インターネットに接続していることを確認する。Web ブラウザなどで確認する程度でよい。ネットワークの設定や、確認方法は割愛する。

6.3. 手順の概要

主な手順としては、以下の通りである。

- 1) インストールファイルのダウンロード
- 2) 圧縮ファイルの展開
- 3) インストールスクリプトの実行

以降の手順では、圧縮ファイルはユーザーのホームディレクトリに存在するものとする。

6.4. インストール

6.4.1. ファイルのダウンロード

Ubuntu に空蟬環境をコンパイル・インストールするためのスクリプト、およびソースコードは一つの圧縮ファイルで存在する。

| | |
|---------|---|
| ファイル名の例 | Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS.tar.gz |
|---------|---|

ファイル名の 4.0 はリリース番号と呼ばれ、比較的大きなバージョンアップがあった場合に新たに付与される。210104 はソースコードの最終変更日でリリース番号とは独立にコードのアップデートとともに増加する。実際の入手方法は、空蟬のポータルサイト (<https://mlfinfo.jp/groups/comp/ja/utsusemi.html>) の「インストール」項目を参照すること。

6.4.2. 圧縮ファイルの展開

```
$ tar xzvf Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS.tar.gz
... (ログ)
```

結果として、Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS という名前のディレクトリが作成される。以下のように確認できる。

```
$ ls -F
Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS/
Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS.tar.gz
```

なお、Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS ディレクトリには以下のファイルやディレクトリが存在する。

```
$ ls -F Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS/
Install.py src/
```

6.4.3. スクリプトの実行

まず、先ほど展開したディレクトリへ移動する。

```
$ cd Utsusemi4_4.0_210104_Ubuntu_1604_2004LTS
```

インストールスクリプトを下記のように Python3 で実行する。

```
$ python3 Install.py
```

すると、以下のようなメインメニューが現れる。

```

$ python3 Install.py

=====
Welcome to Utsusemi software installer (ver.C0.3.8)
=====

-----
Install Directories
MLF software root (Manyo, Utsusemi): /opt/mlfsoft
User's home directory : /home/yinamura
RAW data Directory : /data
-----

-----
Utsusemi Install Main Menu for Debian or Ubuntu
-----

0... Install ALL modules for an instrument

        note: this will DELETE all of modules installed before.

-----
1... Required Ubuntu Packages
2... Utsusemi software (Manyo-lib and common code)
3... Utsusemi software for each BL

7... Change Install Directories

9... Uninstall

others.. exit
-----
>> Which ?

```

ここで、行いたい作業の数字を打ち込む。

なお、メニューの頭に表示されているインストール先となるディレクトリは、メニュー番号 7 で変更可能である。初期状態では、MLF 計算環境公式の場所である。ユーザーのホームディレクトリにインストールしたい場合は、番号 7 を選ぶこと。メニューの項目は Table 47 の通りである。

Table 47 Menu number and meanings on Install.py

| 番号 | 動作 |
|----|--|
| 0 | 全てのインストールを自動で行う。全く最初のインストールの時や、新しいバージョンにすべて置き換えるとき等に使用する。これは以下にあるメニュー番号 1,2,3,4 を順次実行することと同等である。 |
| 1 | 空蟬環境をインストールするために必要な Ubuntu のパッケージを自動でインストールする。Ubuntu 自身の機能を用いるため、インターネットに接続している必要がある。 |
| 2 | 万葉ライブラリと空蟬の自動インストールを行う。この過程ではソースコードからのコンパイルが入るため、時間がかかる。 |
| 3 | 空蟬の装置依存コードのインストールと設定を自動で行う。装置ごとに設定が異なるため、インストール時に装置を選択する必要があるが、別の装置の環境をインストールしたいときに使用する。 |
| 7 | インストール先を変更する。（詳細は後述） |
| 9 | インストールされた空蟬環境を取り除くのに使用する。 |

例として、メニュー番号 0 を選び、すべてをインストールする。

```
>> Which ? 0 [リターン]
```

この後さらに、万葉ライブラリのマルチスレッド対応のために動作させる最大スレッドの数を入力する。通常はその PC のコア数でよいだろう。

```
Please input the number of threads, which is required to use
Manyo-lib with multi-threads.
```

```
>> How many threads? [>=1, default=1]
```

これらの質問に答えると、実際のコンパイルやインストール作業に入る。その際、システム管理者の権限 (root) が必要となるので、以下のようにパスワードが聞かれることがある。そのユーザーのパスワードを打ち込むこと。

```
[sudo] password for hoge:
```

これらの後、自動的にコンパイルやインストール、必要なディレクトリの作成、コードのコピー、環境の設定などが行われる。大量のログが流れる。大まかな順序は、メニューにあった項目の順である。具体的に何をやっているか知りたい場合は 4 章以降を参照のこと。

特に問題がなければ、以下のように表示される。


```

-----
Install of
- manyo-core
- manyo-utsusemi
- python-utsusemi
- python-utsusemi-SIK
- python-utsusemi-AMR
- python-utsusemi-HPN
- python-utsusemi-SAS
- manyo-SAS
- manyo-TKtools
- python-utsusemi-TKtools
were completed.
-----

```

```

-----
Utsusemi Base Environment Setting has been completed.
-----

```

```

-----
Utsusemi Environment Setting for SIK has been completed.
-----

```

```

-----
Utsusemi Environment Setting for AMR has been completed.
-----

```

```

-----
Utsusemi Environment Setting for HPN has been completed.
-----

```

```

-----
Utsusemi Environment Setting for SAS has been completed.
-----

```

```

-----
Utsusemi Environment Setting for TKtools has been completed.
-----

```

```

Utsusemi Default Install was completed.

```

```

>> Back to Main Menu? [Y/n]

```

インストールが終了すると以下のように聞かれるので、終わるなら「n」を入力すること。
リターンだけや「y」を入力すると、先のメインメニューへ戻る。

```

>> Back to Main Menu? [Y/n]

```

6.4.3.1. メニュー番号2を選んだ場合

メニュー番号2を選ぶと万葉ライブラリのインストールを行う。

選択後に、万葉ライブラリのマルチスレッド対応のために動作させる最大スレッドの数を
入力する。通常はそのPCのコア数でよいだろう。

```
Please input the number of threads, which is required to use
Manyo-lib with multi-threads.
```

```
>> How many threads? [>=1, default=1]
```

これらの質問に答えると、実際のコンパイルやインストール作業に入る。その際、システム管理者の権限 (root) が必要となるので、以下のようにパスワードが聞かれることがある。そのユーザーのパスワードを打ち込むこと。

```
[sudo] password for hoge:
```

6.4.3.2. メニュー番号 3 を選んだ場合

このメニュー番号を選ぶと BL ごとに個別にインストールを行うことができる。選択後に、メニュー 2 を実行したか聞かれるので、確認すること。

```
>> Have you installed ManyoLib and Utsusemi Base system ( menu number
2 ) [y/N]?
```

続けて以下のようにインストールするビームラインの選択が行われる。

```
-----
Select Beam Lines you install
-----
[ 1] BL01 4SEASONS (SIK)
[ 2] BL02 DNA (DNA)
[14] BL14 AMATERAS (AMR)
[11] BL11 PLANET (HPN)
[15] BL15 Taikan (SAS)
[TK] TKtools
[-1] Exit menu
-----

>> Which Package?
```

必要な数字を打ち込んでリターンする。四季 (BL01) なら 1 を、アマテラス (BL14) なら 1 を入力する。-1 であれば、このメニューから抜ける。

6.4.4. インストール先のディレクトリについて

6.4.4.1. デフォルトのインストール先

デフォルトのインストール先は、Table 48 のようになっている。

Table 48 Default Installation path

| 表現 | デフォルト | 説明 |
|-----------------------------------|------------------|--|
| MLF (Utsusemi) software directory | /opt/mlfsoft/ | 万葉ライブラリ、空蟬などをインストールする場所。 |
| RAW data directory | /data | 各分光器のイベントデータディレクトリが置かれる場所。ここに各装置のデータを置くためのフォルダ (SIK/や AMR/) を作成。 |
| Users' home directory | /home/XXXXXX (~) | 一時保管用の解析用パラメータファイル等が置かれるディレクトリ (ana/xml、ana/tmp) が置かれる場所。 |

以上の場所は、MLF の標準に近いものとなっている。

ただし、インストール先が/opt/mlfsoft など他のユーザーにも影響がないとも限らない場所であるため、個人のディレクトリにインストールも可能となっている。その変更はインストールメニュー番号 7 で行うことができる。

6.4.4.2. 個人のディレクトリへのインストール

個人のディレクトリにインストールする際のデフォルトは Table 49 のようになっている。

Table 49 Default Installation path for private area

| | 場所 |
|-----------------------------------|-------------------|
| MLF (Utsusemi) software directory | ~/mlfsoft |
| RAW data directory | ~/mlfsoft/RAWDATA |
| Users' home directory | ~/ |

この場合、ソフトウェアのアンインストールは、ほぼ~/mlfsoft ディレクトリを削除するだけでほぼ終了する。

インストールメニュー番号 7 を選ぶと以下のメニューが現れる。

```
-----
Install Directories Menu
-----
1..... Change Directories to private area
2..... Return Directories to MLF default
3..... Load Previous Install Directories
4..... Show Present Setting

0..... Exit
>> Which?
```

ここで 1 を入力すると、以下のようにデフォルト設定の個人用インストール先が示される。

```
=====
                Set your install paths
=====
1. Path to install Utsusemi software
   --> /home/yinamura/mlfsoft

2. Path to Event data directory
   --> /home/yinamura/mlfsoft/RAWDATA

3. Path to user directory (for ana/xml or exp/xml)
   --> /home/yinamura
=====
>> Which do you change ? [1 to 3. 0 to exit]
```

さらに変更を加えたいならその番号を入力する。変更が不要であれば 0 を入力すると一つ前の「Install Directories Menu」に戻るの、そこでさらに 0 を入力するとメインメニューへ戻る。なお、元の MLF 計算環境準拠の設定に戻る場合は「Install Directories Menu」で番号 2 を選ぶとよい。

メインメニューへ戻ると、メニュー上部にある Install Directories の情報が変更されているはずである。

6.5. 初期設定

空蟬における初期設定は、環境変数を設定することである。それを「空蟬環境を有効にする」と呼ぶ。

6.5.1. 空蟬環境の有効化

必要なソフトウェアはすべてインストールされている。空蟬環境を実行するための設定（~/bashrc に書かれている）を有効にするために、以下のようにコマンドを打つ。

```
$ source ~/.bashrc
```

もしくは、ターミナルを閉じて、再度開くと有効となっている。

環境が有効になったかどうかは、それぞれの分光器の環境へ移るためのキーワードを打ち込む。キーワードは、各分光器の略称である（Table 50）。

Table 50 Keywords for Utsusemi environment setting

| 装置 | キーワード |
|-------------|-------|
| 四季（BL01） | SIK |
| DNA（BL02） | DNA |
| アマテラス（BL14） | AMR |
| 大観（BL15） | SAS |

例えば四季の場合なら以下のように表示される。

```
$ SIK
=====
Welcome to Utsusemi Environment for SIK
=====
Ana : Sequencer
D4mat : D4MatSlicer
Inv : MethodAssistant
=====
$
```

BL01 環境への移行が完了したことがわかるメッセージが表示される。

ホームフォルダ直下の”bashrc”ファイルには、上記のキーワードごとに参照する bash スクリプトが存在し、その中で Table 51 のような空蟬用環境変数が指定されている。環境に応じて書き換えること。

Table 51 Environment variables for Utsusemi

| 環境変数 | デフォルト値 | 役割、意味 |
|--------------------|-------------------------------|---------------------------|
| UTSUSEMI_INST_CODE | <inst-code> | 装置コード。（SIK,AMR など） |
| UTSUSEMI_DATA_DIR | "/data" | データフォルダの指定。 |
| UTSUSEMI_USR_DIR | \${UTSUSEMI_USR_DIR-\${HOME}} | ユーザーのフォルダ。（デフォルトはホームフォルダ） |
| UTSUSEMI_LOG_QUIET | n | データ処理実行時のログ表示。"Y"だとかなり減る。 |
| UTSUSEMI_MULTH | 8 | データ処理実行時のマルチプロセッシングの数。 |
| UTSUSEMI_DEBUGMODE | n | デバッグ用フラグ。（デフォルト n で良い） |

6.6. 空蟬の実行

初期設定が終わっていれば、ターミナルから空蟬のコマンドが実行できるはずである。例えば空蟬の基本 GUI である **Sequencer** の起動は以下のようになる。

```
$ Ana
```

6.6.1. 空蟬のバージョン確認

空蟬のバージョンの確認方法を示す。バグなどの報告や、アップデートする際の目安となる。前節の空蟬環境の有効化を済ませたのちに下記のコマンドを入力する。

```
$ python3
>>> import utsusemi
>>> utsusemi.__version__
'4.0.220825'
```

もしくは以下のように入力するとリリース番号とバージョン番号が表示される。

```
$ python3
>>> import UtsusemiInfo
-----
                Utsusemi Info
-----
version : 4.0 (rev 220825)
Contact person : Yasuhiro Inamura
  e-mail : yasuihiro.inamura@j-parc.jp
-----

Manyo>>>
```

6.7. アンインストール

アンインストールを行う手順としては、インストールスクリプトを起動し、メニューから以下を選択する。

```
9..... Uninstall
```

なお、まだいくつかフォルダ（データフォルダ、データ処理用フォルダ）が残っているので下記のように削除する。

```
$ rm -r ~/ana
$ sudo rm -r /data
```

6.8. 問題が発生した場合

インストール時にエラーや予期せぬ状態になった場合、使用した OS とインストール方法を確認して手順に問題がないようなら、担当者（著者）へ状況報告をお願いしたい。

インストール後の動作に問題が発生した場合は、前節に示した空蟬のバージョンを確認し、発生した事象を担当者（著者）まで報告していただきたい。

7. CentOS 7 へのインストール

7.1. 概要

本章では CentOS 7.9 における Python3 環境上に空蟬をインストールする手順の例を示す。

7.2. 動作環境

本章では CentOS 7.9 にて構築しているが、RedHat7 系列なら同じ手順でインストール可能である。

ここでは空蟬のインストールの前に必要な環境の構築手段を示す。

7.2.1. レポジトリ登録

空蟬のインストールに必要となる gcc など開発系パッケージや、各種ライブラリのパッケージを導入するために新たなレポジトリの登録が必要である (Table 52)。

Table 52 External CentOS7 repositories required for Utsusemi installation

| 作業 | 内容 |
|----------------|---------------------------|
| CentOS 開発環境の構築 | 開発ツールのグループパッケージ導入 |
| 追加のレポジトリ | epel レポジトリの追加 |
| その他必要なツール類 | cmake, doxygen, swig3 の導入 |

よって、以下のコマンドで上記の導入を行う。

```
$ su -
# yum groupinstall base 'Development tools' --
  setopt=group_package_types=mandatory,default,optional
# yum install epel-release
# yum install cmake doxygen
```

また、C++コードを Python3 から呼び出すためのラッパーツールである SWIG を用いているが、CentOS に入っているバージョン (swig 2.0.x) では Python3 での万葉ライブラリにおいていくつか不具合が生じる (ElementContainer 同士の割り算ができない) ため、SWIG3 (swig 3.0.x) に置き換える必要がある。

```
$ su -
# yum erase swig
# yum install swig3
```

7.2.2. 必要な外部ライブラリのインストール

万葉ライブラリのコンパイルや、空蟬可視化コードを動作させるのに必要な外部ライブラリ及び、CentOS7 におけるパッケージの情報を下記の Table 53 に示す。

Python3 関連で必要となるライブラリは、未だデフォルトのパッケージでは提供されていない (Python3 用 matplotlib など) ために、pip を用いてインストールすることに注意。

Table 53 External Shared Library required for Utsusemi environment (CentOS7)

| 対象ソフトウェア | ライブラリ | 必要なバージョン | CenOS 7.9 で使用可能なパッケージ |
|-----------------------|-----------------------------|---------------|-------------------------------|
| 万葉ライブラリ関連 (C++コード) | Python3 | 3.6 以降 | python3-devel |
| | Mini-XML | 2.7 以降 3.0 未満 | (不使用：理由は後述) |
| | HDF5 | 1.8.x | hdf5-devel |
| | NeXus | 4.2.x 以降 | (存在せず) |
| | Gnu Scientific Library(GSL) | 指定なし | gsl-devel |
| | Boost C++ Library | 1.46.x 以降 | boost-devel |
| | Open SSL | 指定なし | openssl-devel |
| Python 関連 (GUI 関連) | PySide | 1.2.x | python3-pip + pyside2 |
| | Numpy | 1.6.x 以降 | python3-pip + numpy(pip) |
| | Scipy | 0.10.x 以降 | python3-pip + scipy(pip) |
| | matplotlib | 1.0.x 以降 | python3-pip + matplotlib(pip) |

以上のうち、パッケージで存在するライブラリを導入する。

```
$ su -
# yum install python3-devel hdf5-devel gsl-devel boost-devel openssl-
devel
# yum install python36-pip
# pip3 install -U pip
# python3 -m pip install pyside2
# python3 -m pip install numpy matplotlib scipy
```

mini-XML はパッケージがあるが NeXus と不整合があるためにソースからコンパイルする必要がある。また NeXus はパッケージが存在しない。これらに関しては、次のインストール手順で示す。

7.2.3. インストール先の作成

本マニュアルでは、以下をインストール先とする。

```
/opt/mlfsoft/
```

すなわちこのフォルダの下に以下のような万葉ライブラリや空蟬パッケージのフォルダが置かれる。

```
/opt/mlfsoft/manyo
/opt/mlfsoft/python-utsusemi
/opt/mlfsoft/uGao
```

インストール作業を簡略化するために、このフォルダの所有者を作業者のアカウント（本マニュアルでは hoge）とする。

```
$ su -
# mkdir -p /opt/mlfsoft/
# chown hoge.hoge -R /opt/mlfsoft
```

万葉ライブラリに必要となるライブラリ（本マニュアルでは別途ビルドする NeXus ライブラリ）は、万葉ライブラリのローカルフォルダにインストールする。

7.2.4. mini-xml のインストール

mini-xml は、epel のパッケージで存在するものの、API が大きく変わってしまったバージョンである 3.0 が提供されている。3.0 では万葉ライブラリのコンパイルが不可能であることを確認している。そこで、少し前のバージョンのソースコードをダウンロードする。

- ダウンロード先: <https://github.com/michaelsweet/mxml/releases>
- バージョン: v2.7 以上 (2.8~2.12 のどれでも良い)

コンパイル手順は以下の通り。

```
$ tar xzvf mxml-2.8.tar.gz
$ cd mxml-2.8
$ ./configure --prefix=/opt/mlfsoft/manyo
$ make install
```

7.2.5. NeXus のインストール

NeXus のインストールにはコードをダウンロードする必要がある。

- ダウンロード先: <https://github.com/nexusformat/code/releases>
- ファイル: v4.4.3.tar.gz（最新版でも大丈夫と思われる）

コンパイル手順は下記の通り。インストール先は/opt/mlfsoft/manyo を指定する。ただし、CentOS では 64bit のライブラリは lib64 というフォルダに入れられるため、万葉ライブラリでのデフォルトである lib にコピーする。また include/nexus の中身も include に移す。

```
$ tar xzvf code-4.4.3.tar.gz
$ cd code-4.4.3
$ mkdir build
$ cd build
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/opt/mlfsoft/manyo ..
$ make
$ make install
$ mkdir /opt/mlfsoft/manyo/lib
$ cp /opt/mlfsoft/manyo/lib64/* /opt/mlfsoft/manyo/lib
$ cp /opt/mlfsoft/manyo/include/nexus/* /opt/mlfsoft/manyo/include/
```

7.2.6. PySide、numpy、scipy、matplotlib の動作確認

```
$ python
>>> import numpy
>>> import scipy
>>> import PySide
```

7.3. 手順の概要

主な手順は以下の通りである。

1. ソースコードの準備
2. 万葉ライブラリ コアコード
3. 夕顔コード
4. 空蟬 基本コード
5. 空蟬 装置専用コード

7.4. インストール

7.4.1. ソースコードの準備

ソースコードは、**subversion** や **github** など管理されているが、Ubuntu Linux 版のインストールパッケージにまとまって入っているのでそれを利用する。

空蟬ポータルサイトの「ダウンロード」から空蟬の UbuntuLinux 用をダウンロードする。
これを展開した **src** ディレクトリにあるので、それを利用する。

```
$ tar xzvf Utsusemi_4.0_201112_Ubuntu_1604_2004LTS_PY3.tar.gz
$ cd Utsusemi_4.0_201112_Ubuntu_1604_2004LTS_PY3
$ ls -F
Install.py src/
$ ls -F src/
    environ-DNA/          manyo-DNA/          manyo-SAS/
manyo-core/              manyo-utsusemi/      python-ugao/
version.txt
lib-levmar/              manyo-LevmarTools/  manyo-TKtools/
manyo-emaki/             python-emaki/        python-utsusemi/
```

7.4.2. 万葉ライブラリコアコードのインストール

万葉ライブラリコアコードのインストールは以下のように行う。これにより C++コードがコンパイルされ、ライブラリがインストールされる。

```
$ cd manyo-core
$ make clean;
$ python do_make_Manyo.py --manyo-path=/opt/mlfsoft/manyo --dist-
name=RH7 install
```

7.4.2.1. C++コードのコンパイルスクリプト

上記のように、万葉ライブラリの C++コードおよび後述する空蟬の C++コードをコンパイルする場合、コードと同じフォルダに入っているスクリプト（**do_make_Manyo.py** や **do_make_Utsusemi.py**）を利用してインストールを行う。

書式は以下のようになり、そのオプションなどは Table 54 に示す。

```
$ python do_make_Manyo.py <command> [<option> <option> ...]
$ python do_make_Utsusemi.py <command> [<option> <option> ...]
```

Table 54 Command List of `do_make_Manyo.py` and `do_makeUtsusemi.py` (CentOS7)

| <code>do_make_Manyo.py / do_make_Utsusemi.py <command> [<option> <option> ...]</code> | | |
|---|-----------------------------------|---|
| command | make | コンパイルを行う。 |
| | install | コンパイル+インストールを行う。 |
| | clean | make clean を行う。 |
| | man | help を作成するスクリプトを実行する。 |
| | depend | コンパイル時に必要な情報を収集する。（普段は使用しない） |
| option | --help | ヘルプ表示。 |
| | --manyo-path=<path> | MANYO_PATH を指定する。（通常は /opt/mlfsoft/manyo） |
| | --dist-name=<name> | Linux ディストリビューションを指定する。（"RH6" "RH7" "Ubuntu"） |
| | --pyver=n.m | 使用する Python のバージョンを指定する。省略時はスクリプト実行に使用した Python のバージョン。 |
| | --numth=n | マルチスレッド処理の数を指定する。 |
| | --without-para | コンパイルの平行処理を行わない。指定しない場合、高速に処理されるがメモリを大量に消費（数 GB）するので注意。 |
| | --debug | デバッグモードを指定する。 |
| | --ver=<version> | バージョンの指定。（表記の問題なので不要） |
| | --without-depend | コンパイル時に必要な情報を収集する手順を省く。これにより 高速化するが必ず一度は実施する必要があるので使用しないこと。 |
| | --other-modules=<mod>[,<mod>,...] | 他のモジュールに依存する場合に使用する。<mod> はモジュール名。 |

7.4.3. タ顔コードのインストール

タ顔は Python コードのみで構成されている。インストール先のフォルダにコピーするだけである。

```
$ cp -r python-ugao /opt/mlfsoft/
$ ls -F /opt/mlfsoft
manyo/ python-ugao/
$ ls -F /opt/mlfsoft/python-ugao
uGao/
```

7.4.4. 空蟬基本コードのインストール

空蟬基本コードは、以下の手順となる。

1. C++（万葉ライブラリ）コード
2. Python コード

まず、空蟬の万葉ライブラリ用コードを以下のようにインストールする。C++コードがコンパイルされ、ライブラリがインストールされる。

```
$ cd manyo-utsusemi
$ make clean;
$ python do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-name=RH7 install
```

次に空蟬の Python コードをインストールする。実際には所定のフォルダをコピーするだけである。

```
$ cp -r python-utsusemi /opt/mlfsoft
$ ls -F /opt/mlfsoft
manyo/ python-ugao/ python-utsusemi/
$ ls -F /opt/mlfsoft/python-utsusemi
utsusemi/
$ ls -F /opt/mlfsoft/python-utsusemi/utsusemi
AMR/ DNA/ ENG/ HPN/ SAS/ SIK/ Tktools/ __init__.py ana/ facade/
scripts/ vis/
```

それぞれの装置における固有の関数が必要な場合がある。固有の関数を機能させるためには空蟬基本コードを拡張する装置固有コードが必要である (Table 55) が、上記のコピーでそれらは既にコピーされている。

Table 55 Installation paths of python codes specialized to beam lines (CentOS7)

| BL | 万葉ライブラリ用コード | Python コードインストール先 |
|------|-------------------|------------------------------|
| BL01 | なし | python-utsusemi/utsusemi/SIK |
| BL02 | manyo-DNA (詳細は後述) | python-utsusemi/utsusemi/DNA |
| BL11 | なし | python-utsusemi/utsusemi/HPN |
| BL14 | なし | python-utsusemi/utsusemi/AMR |
| BL15 | manyo-SAS (詳細は後述) | python-utsusemi/utsusemi/SAS |
| BL17 | BL15 と同じ (詳細は後述) | python-utsusemi/utsusemi/VNR |
| BL19 | manyo-Emaki | (不要) |

ただし、いくつかの装置は下記のように、万葉ライブラリ用の C++コードを別途インストールする必要がある。それぞれの BL ごとのインストール方法は後述する。

最後に、空蟬の一部の機能は一時ファイルを作成するため、そのためのフォルダを作成する必要がある。

通常はユーザーのホームフォルダに ana というフォルダを作成し、その下に xml 及び tmp フォルダを作成する。

```
$ mkdir -p ~/ana/xml
$ mkdir -p ~/ana/tmp
```

この ana フォルダのある場所 (上記の例ではホームフォルダ) は、予め環境変数 `UTSUSEMI_USR_DIR` にて指定しておく必要がある (設定方法は環境設定の項で述べる)。

7.4.5. BL01、BL11 および BL14 の場合

これらのビームライン用のコードは python コードだけであるので、特に作業は生じない。

7.4.6. BL15 と BL17 の場合

BL15 は、専用の万葉ライブラリ用のコード (C++) のコンパイル・インストールが必要である。

BL17 は基本 BL15 の環境を利用するので、BL15 のインストールを行った後であれば使用可能である。

7.4.6.1. 万葉ライブラリ用コード

空蟬の万葉ライブラリコードと同じである。インストール先などのパラメータも一致させること。

```
$ cd manyo-SAS
$ make clean
$ python do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-
name=RH7 install
```

空蟬の Python コードは既にインストール（コピー）されているので特に作業はない。

7.4.7. BL02 の場合

BL02 においては、独自の計算用関数（C++）が導入されているので、それらに必要なライブラリを個別にインストールする必要がある（Table 56）。

Table 56 External Shared Libraries for BL02 environment (CentOS7)

| 対象ソフトウェア | 必要な C++ライブラリ | 必要なバージョン | CentOS 7.9 で使用可能なパッケージ |
|-----------------------|--------------|----------|------------------------|
| 万葉ライブラリ関連 (C++コード) | blas | 指定なし | blas-devel |
| | lapack | 指定なし | lapack-devel |
| | levmar | 2.5 | (存在せず) |

7.4.7.1. 万葉ライブラリ用 C++コード

levmar コードのコンパイルに必要な、blas および lapack のパッケージインストールを行う。

```
$ su -
# yum install lapack-devel blas-devel
# exit
```

続けて levmar ライブラリをコンパイルし、出来上がった lib ファイルを万葉ライブラリと同じ場所へコピーする。

```
$ cd Utsusemi-DNA/lib-levmar/2.5
$ make
$ cp liblevmar.* /opt/mlfsoft/manyo/lib
$ cp *.h /opt/mlfsoft/manyo/include/
```

さらに levmar ライブラリを使用する DNA 用関数群（万葉ライブラリの一部として）をコンパイルする。

```
$ cd ../../manyo-LevmarTools/
$ python3 do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-
name=RH7 install
$ cd ../manyo
```

最後に DNA 用の C++コードをコンパイルする。

```
$ python3 do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-
name=RH7 install
```

なお、前述のように Python コードのインストールは既に行っているため不要である。

最後に DNA 専用の環境設定を行う。DNA 用のコマンドなどを\$(HOME)/ana/DNA としてコピー、もしくはリンクを作成する。

```
$ mkdir -p ~/ana/xml
$ mkdir -p ~/ana/tmp
$ cp -r $(pwd)/environ-home/ana ~/ana/DNA
もしくは
$ ln -s $(pwd)/environ-home/ana ~/ana/DNA
```

7.4.8. BL19 の場合

BL19 は、専用の万葉ライブラリ用のコード (C++) のコンパイル・インストールが必要である。

7.4.8.1. 万葉ライブラリ用コード

BL19 の万葉ライブラリ用コードのインストールは空蟬基本コードの場合と同じである。インストール先などのパラメータも一致させること。

```
$ cd manyo-emaki
$ make clean
$ python do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-
name=RH7 install
```

Python コードは特に作業はない。

7.5. 環境設定

7.5.1. 環境変数を設定

空蟬の環境変数設定は、装置ごとに異なる。よって使用する装置に応じて設定を行う必要がある。基本的な設定はそれぞれの装置の **Python** コードのフォルダに存在する **bashrc.py3.XXX** (XXX は装置コード) に収められている。

```
/opt/mlfsoft/python-utsusemi/SIK/bashrc.py3.SIK
/opt/mlfsoft/python-utsusemi/SAS/bashrc.py3.SAS
など

# Utsusemi software environment
# (2021.02.08)

# Utsusemi Environment
export UTSUSEMI_INST_CODE=SIK
export UTSUSEMI_DATA_DIR=/data
export UTSUSEMI_USR_DIR=${UTSUSEMI_USR_DIR-${HOME}}
export UTSUSEMI_WORK_DIR=${UTSUSEMI_USR_DIR}
export UTSUSEMI_LOG_QUIET=n
export UTSUSEMI_MULTH=8
export UTSUSEMI_DEBUGMODE=n

# Path to MLF Software
MLFSOFT_PATH=/opt/mlfsoft

...
```

仮にインストール先が **/opt/mlfsoft** でない場合は、このファイルの **MLFSOFT_PATH** 環境変数の設定を変えること。

これらの設定を有効にするには、空蟬を使用するユーザーのホームフォルダ直下の環境設定用ファイル.bashrc に bashrc.py3.XXX をそのまま追記するか、以下のように source コマンドで空蟬の環境設定用ファイルを実行する。

```
$ source /opt/mlfsoft/python-utsusemi/XXX/bashrc.py3.XXX
```

推奨する方法は、.bashrc に以下のような設定を、それぞれの装置ごとに追加することである。

```
# For Utsusemi environment SIK
if [ -f /opt/mlfsoft/python-utsusemi/SIK/bashrc.py3.SIK ]; then
    alias SIK="source /opt/mlfsoft/python-utsusemi/SIK/bashrc.py3.SIK"
fi

# For Utsusemi environment SAS
if [ -f /opt/mlfsoft/python-utsusemi/SAS/bashrc.py3.SAS ]; then
    alias SAS="source /opt/mlfsoft/python-utsusemi/SAS/bashrc.py3.SAS"
fi
```

この設定では装置コード"SIK"と"SAS"の空蟬パッケージがインストールされたことを意味する。またこれが設定されている場合、以下のように装置コードを入力することで、それぞれの装置の環境に切り替えることが可能となる。

BL01 の場合：

```
$ SIK
```

BL15 の場合：

```
$ SAS
```

7.5.2. 環境変数について

空蟬の動作に関わる環境変数は Table 57 の通りである。

Table 57 Environment variables for Utsusemi (CentOS7)

| 環境変数 | 説明 | 値 |
|--------------------|---|--------|
| UTSUSEMI_INST_CODE | 対象となる装置コード。 | 装置コード |
| UTSUSEMI_DATA_DIR | Raw Data の置き場所。この下に装置コードのフォルダ ("SIK","AMR")、それらの下に RunNo ごとのデータフォルダが並ぶ。 | フォルダパス |
| UTSUSEMI_USR_DIR | ユーザーのベースフォルダ。いくつかの設定ファイルやテンポラリファイルが作成される。 | フォルダパス |
| UTSUSEMI_WORK_DIR | 作業フォルダ。（一部のアプリのみ使用） | フォルダパス |
| UTSUSEMI_LOG_QUIET | 処理中のログを減らす。 | y/n |
| UTSUSEMI_MULTH | マルチスレッドで動作する処理のスレッド数を指定する。（仮に大きな値を入れても CPU の個数よりは大きくならない） | 正の整数 |
| UTSUSEMI_DEBUGMODE | デバッグモードを指定する。 | y/n |

7.6. 空蟬の実行

初期設定が終わっていれば、ターミナルから空蟬のコマンドが実行できるはずである。例えば空蟬の基本 GUI である **Sequencer** の起動は以下のようになる。

```
$ Ana
```

7.6.1. 空蟬のバージョン確認

前節の空蟬環境の有効化を済ませたのちに下記のコマンドを入力する。

```
$ python3
>>> import utsusemi
>>> utsusemi.__version__
'4.0.220825'
```

もしくは

```
$ python3
>>> import UtsusemiInfo
-----
                        Utsusemi Info
-----
      version : 4.0 (rev 220825)
  Contact person : Yasuhiro Inamura
      e-mail : yasuhiko.inamura@j-parc.jp
-----
Manyo>>>
```

ソースコードのリリース番号とバージョン番号が表示される。

7.7. アンインストール

手順としては、インストール先のフォルダを消去する。

```
$ rm -r /opt/mlfsoft
```

なお、まだいくつかフォルダ（データフォルダ、データ処理用フォルダ）が残っているので下記のように削除する。

```
$ rm -r ~/ana
$ sudo rm -r /data
```

7.8. 問題が発生した場合

インストール時にエラーや予期せぬ状態になった場合、使用した OS とインストール方法を確認して手順に問題がないようなら、担当者（著者）へ状況報告をお願いしたい。

インストール後の動作に問題が発生した場合は、前節に示した空蟬のバージョンを確認し、発生した事象を担当者（著者）まで報告していただきたい。

8. AlmaLinux 8 および RockyLinux 8 へのインストール

8.1. 概要

本章では既にサポートが終了した CentOS 8 の代替手段として開発された AlmaLinux 8 及び RockyLinux 8 上に空蟬用の環境を準備し空蟬をインストールする手順の例を示す。両方とも同じ手順である。

8.2. 動作環境

本章では AlmaLinux 8 および RockyLinux 8 にて構築しているが、他の RedHat8 系列なら同じ手順でインストール可能であろう。

ここでは空蟬のインストールの前に必要な環境の構築手段を示す。

8.2.1. レポジトリ登録

空蟬のインストールに必要となる gcc など開発系パッケージや、各種ライブラリのパッケージを導入するために新たなレポジトリの登録が必要である (Table 58)。

Table 58 External AlmaLinux8 and RockyLinux8 repositories required before installation

| 作業 | 内容 |
|------------|-------------------------|
| 開発環境の構築 | 開発ツールのグループパッケージ導入 |
| 追加のレポジトリ | epel および PowerTools を追加 |
| その他必要なツール類 | swig および cmake 導入 |

よって、以下のように上記の導入を行う。

```
$ sudo yum groupinstall base 'Development tools' --
setopt=group_package_types=mandatory,default,optional
$ sudo yum install epel-release
```

なお、レポジトリ PowerTools の追加には、それぞれ以下のファイルを編集する必要がある。

| Distribution | File path |
|--------------|--|
| AlmaLinux | /etc/yum.repos.d/almalinux-powertools.repo |
| RockyLinux | /etc/yum.repos.d/Rocky-PowerTools.repo |

このファイルの以下の行

```
enabled=0
```

を

```
enabled=1
```

に書き換える。

swig や cmake は既にインストールされているはずであるが、以下のようにインストールする。

```
$ sudo yum install swig
$ sudo yum install cmake
```

最後に、レポジトリのアップデートを行っておく (epel と PowerTools のため)


```
$ sudo yum update
```

8.2.2. 必要な外部ライブラリのインストール

万葉ライブラリのコンパイルや、空蟬可視化コードを動作させるのに必要な外部ライブラリ及び、AlmaLinux8 や RockyLinux8 におけるパッケージの情報を下記の Table 59 に示す。

Python3 関連で必要となるライブラリは、未だデフォルトのパッケージでは提供されていない (PySide2 など) もしくは少し古いバージョンのために、pip3 (python3 用 pip) を用いてインストールする方法を示す。

Table 59 External Shared Library required for Utsusemi environment (RH8 series)

| 対象ソフトウェア | ライブラリ | 必要なバージョン | AlmaLinux8 および RockyLinux8 のパッケージ |
|-----------------------|------------------------|---------------|--|
| 万葉ライブラリ関連 (C++コード) | Python3 | 3.6 | python3-devel |
| | Mini-XML | 2.7 以降 3.0 未満 | (不使用：理由は後述) |
| | HDF5 | 1.8.x | hdf5-devel |
| | NeXus | 4.2.x 以降 | (存在せず) |
| | Gnu Scientific Library | 指定なし | gsl-devel |
| | Boost C++ Library | 1.46.x 以降 | boost-devel |
| | Open SSL | 指定なし | openssl-devel |
| Python 関連 (GUI 関連) | Numpy | 1.6.x 以降 | python3-numpy |
| | Scipy | 0.10.x 以降 | python3-scipy |
| | matplotlib | 1.0.x 以降 | python3-matplotlib |
| | PySide | 1.2.x | python3-pip (pip3) + pyside2(on pip3) |

以上のうち、パッケージで存在する C++コード用ライブラリを導入する。

```
$ sudo yum install python3-devel
$ sudo yum install hdf5-devel
$ sudo yum install gsl-devel
$ sudo yum install boost-devel
$ sudo yum install openssl-devel
```

残りの mini-XML はパッケージがあるが NeXus と不整合があるためにソースからコンパイルする必要がある。また NeXus はパッケージが存在しない。これらに関しては、次のインストール手順で示す。

続けて Python のパッケージ (主に可視化、GUI 関連) を導入する。すべて pip のパッケージを用いるが、インストールのやり方は自由である (以下は一例)

```
$ sudo yum install python3-numpy python3-scipy python3-matplotlib
$ sudo yum install python3-pip
$ sudo pip3 install -U pip
$ python3 -m pip install --user pyside2
```

8.2.3. インストール先の作成

本マニュアルでは、以下をインストール先とする。

```
/opt/mlfsoft/
```

すなわちこのフォルダの下に以下のような万葉ライブラリや空蟬パッケージのフォルダが置かれる。

```
/opt/mlfsoft/manyo
/opt/mlfsoft/python-utsusemi
/opt/mlfsoft/python-ugao
```

インストール作業を簡略化するために、このフォルダの所有者を作業者のアカウント（本マニュアルでは hoge）とする。

```
$ su -
# mkdir -p /opt/mlfsoft/
# chown hoge.hoge -R /opt/mlfsoft
```

万葉ライブラリに必要となるライブラリ（本マニュアルでは別途ビルドする NeXus ライブラリ）は、万葉ライブラリのローカルフォルダにインストールする。

8.2.4. mini-xml のインストール

mini-xml は、epel のパッケージで存在するものの、API が大きく変わってしまったバージョンである 3.0 が提供されている。3.0 では万葉ライブラリのコンパイルが不可能であることを確認している。そこで、少し前のバージョンのソースコードをダウンロードする。

- ダウンロード先: <https://github.com/michaelsweet/mxml/releases>
- バージョン: v2.7 以上（2.8～2.12 のどれでも良い）

コンパイル手順は以下の通り。

```
$ tar xzvf mxml-2.8.tar.gz
$ cd mxml-2.8
$ ./configure --prefix=/opt/mlfsoft/manyo
$ make install
```

8.2.5. NeXus のインストール

NeXus のインストールにはコードをダウンロードする必要がある。

- ダウンロード先: <https://github.com/nexusformat/code/releases>
- ファイル: v4.4.3.tar.gz（最新版でも大丈夫と思われる）

コンパイル手順は下記の通り。インストール先は/opt/mlfsoft/manyo を指定する。ただし、RedHat8 系列では 64bit のライブラリは lib64 というフォルダに入れられるため、万葉ライブラリでのデフォルトである lib にコピーする。また include/nexus の中身も include に移す。

```
$ tar xzvf code-4.4.3.tar.gz
$ cd code-4.4.3
$ mkdir build
$ cd build
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/opt/mlfsoft/manyo ..
$ make
$ make install
$ mkdir /opt/mlfsoft/manyo/lib
$ cp /opt/mlfsoft/manyo/lib64/* /opt/mlfsoft/manyo/lib
$ cp /opt/mlfsoft/manyo/include/nexus/* /opt/mlfsoft/manyo/include/
```

8.2.6. PySide、numpy、scipy および matplotlib の動作確認

```
$ python3
>>> import numpy
>>> import scipy
>>> import PySide
```

8.3. 手順の概要

主な手順は以下の通りである。

1. ソースコードの準備
2. 万葉ライブラリ コアコード
3. タ顔コード
4. 空蟬 基本コード
5. 空蟬 装置専用コード

8.4. インストール

8.4.1. ソースコードの準備

ソースコードは、**subversion** や **github** など管理されているが、Ubuntu Linux 版のインストールパッケージにまとまって入っているのでそれを利用する。

空蟬ポータルサイトの「インストール」項目のリンク先から、空蟬の UbuntuLinux 用をダウンロードする。

これを展開した **src** ディレクトリにあるので、それを利用する。

```
$ tar xzvf Utsusemi_4.0_201112_Ubuntu_1604_2004LTS_PY3.tar.gz
$ cd Utsusemi_4.0_201112_Ubuntu_1604_2004LTS_PY3
$ ls -F
Install.py src/
$ ls -F src/
    environ-DNA/          manyo-DNA/          manyo-SAS/
manyo-core/              manyo-utsusemi/      python-ugao/
version.txt
lib-levmar/              manyo-LevmarTools/   manyo-TKtools/
manyo-emaki/             python-emaki/        python-utsusemi/
```

8.4.2. 万葉ライブラリコアコードのインストール

万葉ライブラリコアコードを以下のようにインストールする。C++コードがコンパイルされ、ライブラリがインストールされる。なお、AlmaLinux8 と RockyLinux8 であっても、引数の **-dist-name** の値は **RH7** で良い。

```
$ cd manyo-core
$ make clean;
$ python do_make_Manyo.py --manyo-path=/opt/mlfsoft/manyo --dist-name=RH7 install
```

8.4.2.1. コンパイルスクリプトについて

上記のように、万葉ライブラリの C++コード、及び後述する空蟬の C++コードをコンパイルする場合、コードと同じフォルダに入っているスクリプト (**do_make_Manyo.py** や **do_make_Utsusemi.py**) を利用してインストールを行う。書式は以下の通り。

```
$ python do_make_Manyo.py <command> [<option> <option> ...]
$ python do_make_Utsusemi.py <command> [<option> <option> ...]
```

パラメータは Table 60 を参照すること。

Table 60 Command List of `do make Manyo.py` and `do make Utsusemi.py` (RH8 series)

| <code>do make Manyo.py / do make Utsusemi.py <command> [<option> <option> ...]</code> | | |
|---|-----------------------------------|---|
| command | make | コンパイルを行う。 |
| | install | コンパイル+インストールを行う。 |
| | clean | make clean を行う。 |
| | man | help を作成するスクリプトを実行する。 |
| | depend | コンパイル時に必要な情報を収集する。（普段は使用しない） |
| option | --help | ヘルプを表示する。 |
| | --manyo-path=<path> | MANYO_PATH を指定する。（通常は /opt/mlfsoft/manyo） |
| | --dist-name=<name> | Linux ディストリビューションを指定する。（"RH6" "RH7" "Ubuntu"） |
| | --pyver=n.m | 使用する Python のバージョンを指定する。省略時はスクリプト実行に使用した Python のバージョンとなる。 |
| | --numth=n | マルチスレッド処理の数を指定する。 |
| | --without-para | コンパイルの平行処理を行わない。指定しない場合、高速に処理されるがメモリを大量に消費（数 GB）するので注意。 |
| | --debug | デバッグモードを指定する。 |
| | --ver=<version> | バージョンを指定する。（表記の問題なので不要） |
| | --without-depend | コンパイル時に必要な情報を収集する手順を省く。これにより 高速化するが必ず一度は実施する必要があるので使用しないこと。 |
| | --other-modules=<mod>[,<mod>,...] | 他のモジュールに依存する場合に使用する。<mod> はモジュール名。 |

8.4.3. タ顔コードのインストール

タ顔は Python コードのみで構成されている。インストール先のフォルダにコピーするだけである。

```
$ cp -r python-ugao /opt/mlfsoft/
$ ls -F /opt/mlfsoft
manyo/ python-ugao/
$ ls -F /opt/mlfsoft/python-ugao
uGao/
```

8.4.4. 空蟬基本コードのインストール

空蟬基本コードは、以下の手順となる。

1. C++（万葉ライブラリ）コード
2. Python コード

まず、空蟬の万葉ライブラリ用コードを以下のようにインストールする。C++コードがコンパイルされ、ライブラリがインストールされる。万葉ライブラリの時と同様に、引数の `-dist-name` の値は RH7 で良い。

```
$ cd manyo-utsusemi
$ make clean;
$ python do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-
name=RH7 install
```

次に空蟬の Python コードをインストールする。実際には所定のフォルダをコピーするだけである。

```
$ cp -r python-utsusemi /opt/mlfsoft
$ ls -F /opt/mlfsoft
manyo/ python-ugao/ python-utsusemi/
$ ls -F /opt/mlfsoft/python-utsusemi
utsusemi/
$ ls -F /opt/mlfsoft/python-utsusemi/utsusemi
AMR/ DNA/ ENG/ HPN/ SAS/ SIK/ TKtools/ __init__.py ana/ facade/
scripts/ vis/
```

それぞれの装置における固有の関数が必要な場合がある。固有の関数を機能させるためには空蟬基本コードを拡張する装置固有コードが必要であるが、上記のコピーでそれらは既にコピーされている (Table 61)。

Table 61 Installation paths of python codes specialized to beam lines (RH8 series)

| BL | 万葉ライブラリ用コード | Python コードインストール先 |
|------|-------------------|------------------------------|
| BL01 | なし | python-utsusemi/utsusemi/SIK |
| BL02 | manyo-DNA (詳細は後述) | python-utsusemi/utsusemi/DNA |
| BL11 | なし | python-utsusemi/utsusemi/HPN |
| BL14 | なし | python-utsusemi/utsusemi/AMR |
| BL15 | manyo-SAS (詳細は後述) | python-utsusemi/utsusemi/SAS |
| BL17 | BL15 と同じ (詳細は後述) | python-utsusemi/utsusemi/VNR |
| BL19 | manyo-Emaki | (不要) |

ただし、いくつかの装置は下記のように、万葉ライブラリ用の C++コードを別途インストールする必要があるが、それぞれの BL ごとのインストール方法は後述する。

最後に、空蟬の一部の機能は一時ファイルを作成するため、そのためのフォルダを作成する必要がある。

通常はユーザーのホームフォルダに ana というフォルダを作成し、その下に xml 及び tmp フォルダを作成する。

```
$ mkdir -p ~/ana/xml
$ mkdir -p ~/ana/tmp
```

この ana フォルダのある場所 (上記の例ではホームフォルダ) は、環境変数 `UTSUSEMI_USR_DIR` にて指定しておく必要がある (設定方法は環境設定の項で述べる)。

8.4.5. BL01、BL11、BL14 の場合

これらのビームライン用のコードは、python コードだけであるので、特に作業は生じない。

8.4.6. BL15 と BL17 の場合

BL15 は、専用の万葉ライブラリ用のコード (C++) のコンパイル・インストールが必要である。

BL17 は基本 BL15 の環境を利用するので、BL15 のインストールを行った後であれば使用可能である。

8.4.6.1. 万葉ライブラリ用コード

空蟬の万葉ライブラリコードと同じである。インストール先などのパラメータも一致させること。

```
$ cd manyo-SAS
$ make clean
$ python do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-name=RH7 install
```

空蟬の Python コードは既にインストール（コピー）されているので特に作業はない。

8.4.7. BL02 の場合

BL02 においては、独自の計算用関数（C++）が導入されているので、それらに必要なライブラリを個別にインストールする必要がある（Table 62）。

Table 62 External shared libraries for BL02 environment (RH8 series)

| 対象ソフトウェア | 必要な C++ ライブラリ | 必要なバージョン | CenOS 7.9 で使用可能なパッケージ |
|-----------------------|------------------|----------|-----------------------|
| 万葉ライブラリ関連 (C++コード) | blas | 指定なし | blas-devel |
| | lapack | 指定なし | lapack-devel |
| | levmar | 2.5 | (存在せず) |

8.4.7.1. 万葉ライブラリ用 C++コード

levmar コードのコンパイルに必要な、blas および lapack のパッケージインストールを行う。

```
$ su -
# yum install lapack-devel blas-devel
# exit
```

続けて levmar ライブラリをコンパイルし、出来上がった lib ファイルを万葉ライブラリと同じ場所へコピーする。

```
$ cd Utsusemi-DNA/lib-levmar/2.5
$ make
$ cp liblevmar.* /opt/mlfsoft/manyo/lib
$ cp *.h /opt/mlfsoft/manyo/include/
```

さらに levmar ライブラリを使用する DNA 用関数群（万葉ライブラリの一部として）をコンパイルする。

```
$ cd ../../manyo-LevmarTools/
$ python3 do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-name=RH7 install
$ cd ../manyo
```

最後に DNA 用の C++コードをコンパイルする。

```
$ python3 do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-name=RH7 install
```

なお、前述のように Python コードのインストールは既に行っているため不要である。

最後に DNA 専用の環境設定を行う。DNA 用のコマンドなどを\$(HOME)/ana/DNA としてコピー、もしくはリンクを作成する。

```
$ mkdir -p ~/ana/xml
$ mkdir -p ~/ana/tmp
$ cp -r $(pwd)/environ-home/ana ~/ana/DNA
もしくは
$ ln -s $(pwd)/environ-home/ana ~/ana/DNA
```

8.4.7.2. BL19 の場合

BL19 は、専用の万葉ライブラリ用のコード (C++) のコンパイル・インストールが必要である。

8.4.7.3. 万葉ライブラリ用コード

BL19 の万葉ライブラリ用コードのインストールは空蟬の場合と同じである。インストール先などのパラメータも一致させること。

```
$ cd manyo-emaki
$ make clean
$ python do_make_Utsusemi.py --manyo-path=/opt/mlfsoft/manyo --dist-name=RH7 install
```

Python コードは特に作業はない。

8.5. 環境設定

8.5.1. 環境変数を設定

空蟬の環境変数設定は、装置ごとに異なる。よって使用する装置に応じて設定を行う必要がある。基本的な設定はそれぞれの装置の Python コードのフォルダに存在する bashrc.py3.XXX (XXX は装置コード) に収められている。

```
/opt/mlfsoft/python-utsusemi/SIK/bashrc.py3.SIK
/opt/mlfsoft/python-utsusemi/SAS/bashrc.py3.SAS
など

# Utsusemi software environment
# (2021.02.08)

# Utsusemi Environment
export UTSUSEMI_INST_CODE=SIK
export UTSUSEMI_DATA_DIR=/data
export UTSUSEMI_USR_DIR=${UTSUSEMI_USR_DIR-${HOME}}
export UTSUSEMI_WORK_DIR=${UTSUSEMI_USR_DIR}
export UTSUSEMI_LOG_QUIET=n
export UTSUSEMI_MULTH=8
export UTSUSEMI_DEBUGMODE=n

# Path to MLF Software
MLFSOFT_PATH=/opt/mlfsoft

...
```

仮にインストール先が/opt/mlfsoft でない場合は、このファイルの *MLFSOFT_PATH* 環境変数の設定を変えること。

これらの設定を有効にするには、空蟬を使用するユーザーのホームフォルダ直下の環境設定用ファイル *.bashrc* に *bashrc.py3.XXX* をそのまま追記するか、以下のように *source* コマンドで空蟬の環境設定用ファイルを実行する。

```
$ source /opt/mlfsoft/python-utsusemi/XXX/bashrc.py3.XXX
```

推奨する方法は、*.bashrc* に以下のような設定を、それぞれの装置ごとに追加することである。

```
# For Utsusemi environment SIK
if [ -f /opt/mlfsoft/python-utsusemi/SIK/bashrc.py3.SIK ]; then
    alias SIK="source /opt/mlfsoft/python-utsusemi/SIK/bashrc.py3.SIK"
fi

# For Utsusemi environment SAS
if [ -f /opt/mlfsoft/python-utsusemi/SAS/bashrc.py3.SAS ]; then
    alias SAS="source /opt/mlfsoft/python-utsusemi/SAS/bashrc.py3.SAS"
fi
```

この設定では装置コード"SIK"と"SAS"の空蟬パッケージがインストールされたことを意味する。またこれが設定されている場合、以下のように装置コードを入力することで、それぞれの装置の環境に切り替えることが可能となる。

BL01 の場合

```
$ SIK
```

BL15 の場合

```
$ SAS
```

8.5.2. 環境変数について

空蟬の動作に関わる環境変数は Table 63 の通りである。

Table 63 Environment variables for Utsusemi (RH8 series)

| 環境変数 | 説明 | 値 |
|--------------------|---|--------|
| UTSUSEMI_INST_CODE | 対象となる装置コード。 | 装置コード |
| UTSUSEMI_DATA_DIR | Raw Data の置き場所。この下に装置コードのフォルダ ("SIK","AMR")、それらの下に RunNo ごとのデータフォルダが並ぶ。 | フォルダパス |
| UTSUSEMI_USR_DIR | ユーザーのベースフォルダ。いくつかの設定ファイルやテンポラリファイルが作成される。 | フォルダパス |
| UTSUSEMI_WORK_DIR | 作業フォルダ。（一部のアプリのみ使用） | フォルダパス |
| UTSUSEMI_LOG_QUIET | 処理中のログを減らす。 | y/n |
| UTSUSEMI_MULTH | マルチスレッドで動作する処理のスレッド数を指定する。（仮に大きな値を入れても CPU の個数よりは大きくならない） | 正の整数 |
| UTSUSEMI_DEBUGMODE | デバッグモードを指定する。デフォルトは n。 | y/n |

8.6. 空蟬の実行

初期設定が終わっていれば、ターミナルから空蟬のコマンドが実行できるはずである。例えば空蟬の基本 GUI である **Sequencer** の起動は以下のようになる。

```
$ Ana
```

8.6.1. 空蟬のバージョン確認

前節の空蟬環境の有効化を済ませたのちに下記のコマンドを入力する。

```
$ python3
>>> import utsusemi
>>> utsusemi.__version__
'4.0.220825'
```

もしくは

```
$ python3
>>> import UtsusemiInfo
-----
                Utsusemi Info
-----

        version : 4.0 (rev 220825)
Contact person : Yasuhiro Inamura
        e-mail  : yasuhiko.inamura@j-parc.jp
-----

Manyo>>>
```

ソースコードのリリース番号とバージョン番号が表示される。

8.7. アンインストール

手順としては、インストール先のフォルダを消去する。

```
$ rm -r /opt/mlfsoft
```

なお、まだいくつかフォルダ（データフォルダ、データ処理用フォルダ）が残っているので下記のように削除する。

```
$ rm -r ~/ana
$ sudo rm -r /data
```

8.8. 問題が発生した場合

インストール時にエラーや予期せぬ状態になった場合、使用した OS とインストール方法を確認して手順に問題がないようなら、担当者（著者）へ状況報告をお願いしたい。

インストール後の動作に問題が発生した場合は、前節に示した空蟬のバージョンを確認し、発生した事象を担当者（著者）まで報告していただきたい。

参考文献

- 1) Y. Inamura, T. Nakatani, J. Suzuki, T. Otomo, Development status of software 'Utsusemi' for Chopper Spectrometers at MLF, J-PARC. J. Phys. Soc. Japan., vol. 82, no. Supple.A, 2013, SA031, 9p.
- 2) J-PARC MLF, 稲村泰弘, 空蟬ポータルサイト, <https://mlfinfo.jp/groups/comp/ja/utsusemi.html> (参照 : 2023 年 11 月 1 日) .
- 3) K.Nakayoshi, Y.Yasu, E.Inoue, H.Sendai, M.Tanaka, S.Satoh, S.Muto, N.Kaneko, T.Otomo, T.Nakatani and T.Uchida, Development of a data acquisition sub-system using DAQ-Middleware, Nucl. Instrum. Methods Phys. Res. A, vol. 600, no. 1, 2009, pp.173-175.
- 4) J. Suzuki, T. Nakatani, T. Ohhara, Y. Inamura, M. Yonemura, T. Morishima, T. Aoyagi, A. Manabe, T. Otomo, Object-oriented data analysis framework for neutron scattering experiments, Nucl. Instrum. Methods Phys. Res. A, vol. 600, issue 1, 2009, pp.123-125.
- 5) Free Software Foundation, GPLv3, <https://www.gnu.org/licenses/gpl-3.0.html> (参照 : 2023 年 11 月 1 日)

