

CASA, Vissage の開発

川崎 渉 (国立天文台)

概要

国立天文台アルマプロジェクト・コンピューティンググループでソフトウェア開発に従事している。ここでは、CASA および Vissage という 2 つのソフトウェア、およびそれらの開発作業について紹介したい。

1. CASA (Common Astronomy Software Applications)

1-1. CASA とは

CASA (<https://casa.nrao.edu>) は電波望遠鏡データ解析ソフトウェアで、干渉計 (ALMA および VLA) および単一鏡 (ALMA および野辺山 45m 鏡) データの解析を行うものである。2005 年より、米国国立電波天文台 (NRAO) を中心に国立天文台、欧州南天天文台 (ESO) などが協力して開発が進められているが、その原形は 1990 年代に開発された AIPS++ であり、そういった意味では既に 30 年という長い歴史を持つソフトであるとも言える。

1-2. ソースコード

技術的な側面に興味を持つ方が多いと思うので、まず CASA のソースコードについて紹介する。CASA は合計 200 万行以上、約 800MB という巨大なプログラムである。プログラミング言語としては、大雑把には、機能の低レベルな部分に関しては C++ (一部は前身である AIPS++ から引き継いだもの)、それらを用いる高レベル機能に関しては Python で実装されている。ユーザーから見た CASA は、IPython ベースのインタプリタ形式でコマンド入力するものと、Python モジュールとして提供されるものがあるが、いずれにせよ、ユーザーは Python を通じて CASA を使用することになる。なお、C++ コードを Python 側から呼び出すためのバインディングとして SWIG、タスク (CASA の高レベルコマンド) のインタフェース定義などの記述には XML が用いられるなどしている。

1-3. 開発チーム

2022 年 2 月現在、総勢 30 名 (NRAO が 20 名、ESO と国立天文台が 5 名ずつ) のチームで開発を進めている。国立天文台は主に単一鏡データ解析機能の開発を担当しており、アルマプロジェクト・コンピューティンググループ所属の 5 名 (中里剛、川崎渉、ミエル・ルノー (Miel Renaud)、林洋平、島田和彦) が開発を、また科学的な見地に立った単一鏡データ機能の開発項目選定やユーザーテストについては、東アジアアルマ地域センターの宮本祐介さんに担当して頂いている。

1-4. 開発作業

基本的に、いわゆるソフトウェア開発の V 字モデルに沿う形で開発が進められている。国立天文台が担当する単一鏡データ機能の場合、要求分析・要件定義・受け入れテスト (CASA 開発においては Validation と称する) といった上流部分は宮本さん、設計・開発・単体/結合テスト (CASA では Verification と称する) は上記 5 名の開発者が分担して進めている。これが CASA のリリース毎に繰

り返される。リリース間隔は昨秋まではおよそ半年毎、最近は間隔を短縮して 1 か月毎のリリースを目標としている。

実際の開発においては、NRAO が運用する様々なツール（ソースコード管理：Git、課題管理：JIRA、バージョン管理：Bitbucket、ビルド/テスト管理：Bamboo）を利用している。さらにこれに加えて、国立天文台の開発チーム内では Redmine を活用して相互にコードレビューを行い、コードの質の向上に努めている。開発が一段落すると Sphinx を用いてドキュメントの作成を行う。

CASA 開発チームは日米欧に分散していることもあって、コロナ禍以前からリモートで作業することには慣れている。開発者間の連絡・報告には電子メールや Confluence を、また Zoom を用いて週 1 回ミーティングを行う。さらに数年に一度程度、全ての開発者が一堂に会し、数日間議論を行う（昨年はやむなく Zoom 会議となった）。開発者間の共通言語は英語であるため、国立天文台の開発者だけで、別途、日本語による週例ミーティングを行い、さらなる意思の疎通を図っている。

1-5. ngCASA/CNGI : 次世代の CASA

最後に、現在、準備段階を経て開発作業が始まっている次世代の CASA についてふれておく。データ解析機能（ngCASA = Next Generation CASA）とより基本的な機能（CNGI = CASA Next Generation Infrastructure）で異なる名前が付けられているが、要は、クロック周波数の増加以上に多コア化を進める最近の CPU の動向と、ngVLA などをもたらすであろう将来的なさらなるデータサイズの増大に鑑み、従来の CASA においてほとんど行われてこなかった並列処理を本格的に導入するため大幅にコードを書き換えるという、野心的な計画である。既存のライブラリを最大限活用することによってソースコードの分量を減らし、維持管理を容易にしたいという狙いもある。現在 C++ で実装されているコードは Python で書き直し、Dask を用いて並列処理を行う（Numpy など、Python を用いた並列処理で処理速度を上げられるようなライブラリの活用が必須となる）。それに伴い、内部で用いるデータ形式 MeasurementSet も、従来の CASA table ベースから xarray ベースのものに再実装される。ユーザーにとってはともかく、開発者にとっては、前身である AIPS++以来 30 年にわたって C++で実装されてきたコードが Pythonに置き換えられるという、極めて大きな変化になる。元々、現在バージョン 6 である CASA の次の CASA7 から ngCASA/CNGI に移行する計画だったが、最近では、完成した部分から順次現在の CASA6 に取り入れていく方針に変化している。

2. Vissage (VISualisation Software for Astronomical Gigantic data cubEs)

私は作業時間の 90%を CASA の開発と関連業務に費やしているが、残りの 10%を使って FITS 画像ビューワ Vissage の開発を行っている。CASA とはいろいろな点で対照的な開発になっており、こちらについても紹介したい。

2-1. JVO による ALMA データ公開サービス

開発のきっかけは、国立天文台データセンターが運用するバーチャル天文台システム JVO (Japanese Virtual Observatory) において、PI による専有期間終了後に一般公開される ALMA の FITS データの閲覧・配布サービスの立ち上げだった。問題となったのはデータサイズが巨大（将来的には TB 以上を予想）、かつ多次元（空間方向 2 次元に加えて周波数方向や偏波の情報も含む 3 ないし 4 次元の

データキューブとなる)であることだった。我々は、巨大なデータをダウンロードさせるのではなく、(1) Web インタフェース上でユーザーが興味を持つ部分を迅速に見つけ、その部分だけを切り出してダウンロードし、その後、(2) ローカル環境においてより詳細に観察・解析する、という仕組みを提供することにした。前者については国立天文台データセンターの江口智士(当時)および Chris Zapart によりクイックルックシステム ALMAWebQL が開発され、既に JVO 上で稼働中である。後者の機能、即ち、ローカル環境にダウンロードした比較的小さな、しかし多次元のデータキューブを閲覧するためのソフトウェア Vissage を私が開発している。

2-2. 目的と機能

多次元の天文データを簡単に扱うことができ、しかも無料できちんとメンテナンスされているソフトウェアが現状ほぼ見当たらないという大きな問題を解消すべく開発を進めている。未完成の部分も多々あるが、既に以下のような機能を実装している。

(1) 多次元天文データの表示

空間 2 次元 + 周波数 1 次元の 3 次元データキューブの情報をディスプレイや紙といった 2 次元の面上で分かりやすく表現するにあたって、(電波)天文学者は以前より、周波数方向で積分やモーメント計算する(積分強度図・モーメント図)、周波数方向で分割して並べる(チャンネルマップ)、空間方向を 1 次元に限り周波数と合わせて 2 次元イメージとする(位置・速度図)など多彩な方法を駆使してきた。偏波や時間方向の情報も持つデータではさらに表示方法が多様化するの言うまでもない。今後、多波長データを用いた解析が盛んになるとともに、他波長の観測家や理論家といった、必ずしもデータキューブになじみがないユーザーが ALMA データキューブを扱う機会も増えるものと思われ、そのようなユーザーにとっても取扱いやすいものを提供することが必要だと考える。Vissage では、偏波を含む ALMA のデータキューブを簡単・直観的な操作で表示することができ、複数の異なる表示を同時に並べて閲覧することも可能である(図 1 参照)。また多波長に対応するべく、ALMA 以外のデータ表示のサポートを進めている。

(2) JVO をはじめとする天文データサービスとの連携

公開データを用いた研究スタイル(アーカイブ天文学)は理論・観測・数値シミュレーションに続く第 4 の天文学とも呼ばれ、今後さらに発展することが予想される。Vissage では、画面表示中の ALMA データを基にして JVO が提供する ALMA やすばるデータにアクセスする機能を実現した。

(3) 作図機能

FITS 画像を取り込んだ図(論文雑誌に掲載できる品質のもの)を簡単に作成できる機能を開発している。現在は単一の FITS 画像を eps 形式で出力するのみだが、複数画像の使用や文字・記号・図形などの書き込みなどにも対応し、より複雑な図を作成できるようにしたい。

(4) Python インタフェース

自作の Python スクリプトから Vissage を操作可能にするため、Vissage 用の Python モジュールを開発中である。用途としては、Python 上での他の解析ソフトやライブラリとの連携、Vissage を表示ツール・作図エンジンとして Python 側から制御することなどを想定している。GUI を画面表示することなく、バックグラウンド処理として Vissage に図を出力させることも可能である。

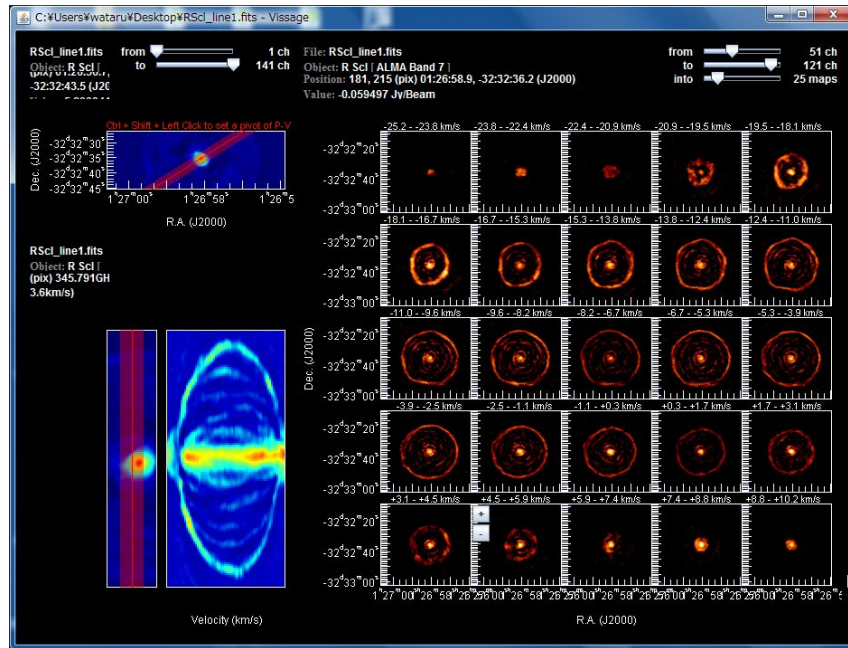


図 1 : Vissage の表示の一例。画面の左側 1/3 の領域に位置-速度図（左上は空間軸の選択を行うための積分強度図、左下がその空間軸を基にした位置-速度図）を、右側 2/3 の部分には同じデータキューブから作ったチャンネルマップを配置している。このような表示もマウスによる直観的な操作のみで実現可能である

この他、CASA など他の解析ソフトとの連携、多彩な形式での出力機能など、ALMA データキューブを用いた研究活動や多波長にまたがるアーカイブ天文学にとって痒い所に手が届くようなソフトウェアに育てていきたいと考えている。

2-3. 実装に関して

単一のパッケージで Linux, Mac, Windows な様々な環境で動作することを考え、Vissage 本体は Java で実装している (GUI は Swing)。なお、Vissage アイコンへの FITS ファイルのドラッグ&ドロップによる起動や使用メモリ量などのコントロールを実現するために用意した Windows 用のラッパー部分は C# で、Python 用のモジュールは (当然ながら) Python で実装している。並列処理に関しては未だ実現していないが、将来的には導入したい。

3. まとめ

以上、私が開発に従事している 2 つのソフトウェアの紹介をさせて頂いた。ソフトウェアとしては規模・開発形態・実装言語など全く異なるものであるが、これからの天文学研究に役立つソフトウェアにしたいという点は共通である。今後とも、機能の充実や品質の改善に努めていきたいと考えている。