BrAPI Success Stories

Authors

- **Peter "BrapMan" Selby** ✉
  ⓘD [0000-0001-7151-4445](#) · ⓞ [BrapiCoordinatorSelby](#)
  Cornell University · Funded by NIFA-DSFAS 2022-67021-37024

- **Trevor "Cool Kid" Rife**
  ⓘD [0000-0002-5974-6523](#) · ⓞ [trife](#)
  Clemson University

- **Sebastian "Baz" Raubach**
  ⓘD [0000-0001-5659-247X](#) · ⓞ [sebastian-raubach](#)
  The James Hutton Institute

- **Iain Milne**
  ⓘD [0000-0002-4126-0859](#) · ⓞ [imilne](#)
  The James Hutton Institute

- **Becky Smith**
  ⓘD [0000-0002-8968-3383](#) · ⓞ [Batbaby91](#)
  The James Hutton Institute

- **Isabelle Alic**
  ⓘD [0000-0002-8961-6068](#) · ⓞ [Isabelle-inrae](#)
  INRAE

✉ — Correspondence possible via [GitHub Issues](#) or email to Peter "BrapMan" Selby <ps664@cornell.edu>.

Abstract

The Breeding API (BrAPI) project is an effort to enable interoperability among plant breeding databases. BrAPI is a standardized RESTful web service API specification for communicating plant breeding data. This community driven standard is free to be used by anyone interested in plant breeding data management. This manuscript describes updates and outlook for the current version of BrAPI.

Introduction

Plant and animal breeding is an incredibly important part of today's society. Almost every country in the world has some kind of breeding program supporting the agricultural community to produce bigger, better, healthier, more sustainable crops. Modern breeding techniques require large amounts of high quality data to be effective. In the digital age, that breeding data is being collected, managed, and analyzed with computer software. Interoperability between breeding software tools, systems, and databases can substantially increase the efficiency of a breeding program. The ability to share tools gives each program a boost in computational power. The ability to share data means everyone has access to larger, more complete, datasets and get build more accurate computational models and produce more accurate predictions.

The Breeding API (BrAPI) project is an effort to enable interoperability among breeding tools, systems, and databases. BrAPI is a standardized Representational State Transfer (REST), web service, Application Programming Interface (API), specification for breeding and related agricultural data. [1] By using the BrAPI standard, breeding software can more easily become interoperable, allowing groups to more easily share data and software tools.

How it works

An Application Programming Interface (API) is a technical connection between two pieces of software. Just as a Graphical User Interface (GUI) or a Command Line Interface (CLI) allows a human user to interact with a piece of software, an API allows one software application to interact with another. A GUI or CLI might allow a user to input data, read data, and start processes within an application. An API allows one piece of software (sometimes called a client, user agent, or service consumer) to programmatically input data, read data, and start process within another piece of software (sometimes called a server or service provider).

A Representational State Transfer (REST) web service is a type of API commonly used in today's modern web infrastructure. REST is a technical architecture that describes the stateless transmission of data between applications. Typically, REST systems are implemented using the standard HTTP protocol that most of the modern internet is built upon. REST implementations also generally use JavaScript Object Notation (JSON) to represent the data being transferred. Both HTTP and JSON are programming language agnostic, very stable, and very flexible. This means BrAPI can be implemented in almost any piece of software, and can solve a wide range of use cases.

Data repositories and service providers can choose to represent their data as a BrAPI compatible API. By mapping the internal data structures to the standard models, data repositories can easily expose data to the outside world. Similarly, they can accept new data from external sources and automatically map the new data into the existing database. Client application developers can take advantage of this standardization by building tools that can easily integrate with all other BrAPI compatible data repositories. Visualization, reporting, analytics, data collection, and quality control

tools can be built once and shared with other organizations following the standards. As the number of BrAPI compatible databases, tools, and organizations grows, so does the value added by implementing the standard into a given application.

Project Updates

Over its lifetime, the BrAPI project has grown and changed substantially. The latest stable version of the specification (v2.1) looks vastly different from the original version (v1.0) released in 2017. The total size of the specification has almost quadrupled in that time, going from 51 endpoints documented in v1.0 to 201 endpoints documented in v2.1. Because of this growth, the specification documents were reorganized into four modules: BrAPI-Core, BrAPI-Germplasm, BrAPI-Genotyping, and BrAPI-Phenotyping. Figure 1 shows a simplified domain map of the whole BrAPI v2.1 data model, divided into the organizational modules. The early versions of the specification focused on read-only phenotype data, with a small consideration to the other domains. Now the specification has a full representation of most of the major concepts applicable to the breeding process. The new specification is also internally consistent, easier to navigate, and allows for read, write, and update capabilities. None of those qualities were a guarantee for the earlier versions.
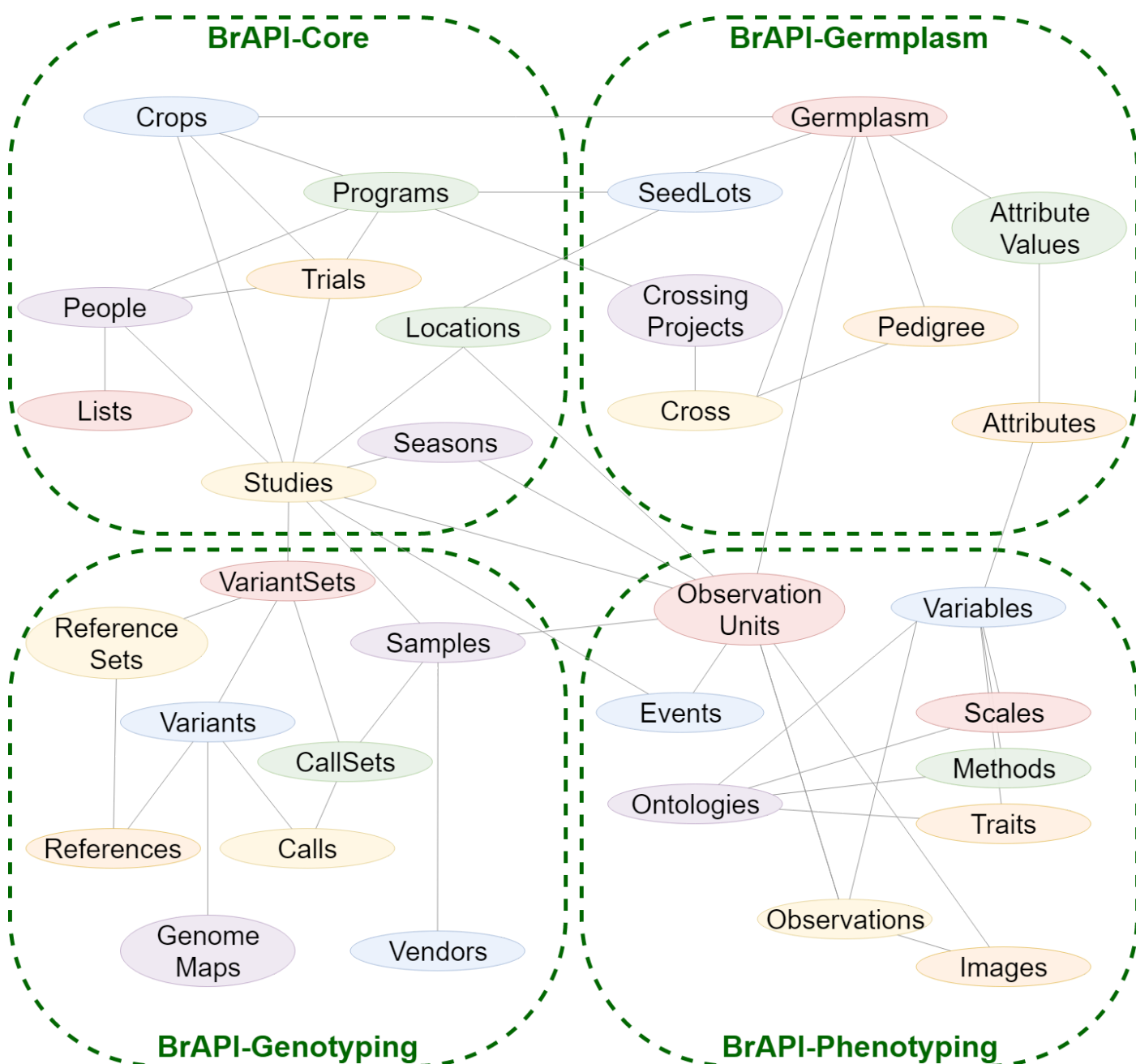
Figure 1: A simplified domain map of the whole BrAPI data model, divided into organizational modules. A more detailed Entity Relationship Diagram (ERD) is available on brapi.org.

As the specification has matured, so have the tools, services, and libraries available to the community to work with the specification. Every version of the specification is now released with a change log to guide developers upgrading from a previous version, an Entity Relationship Diagram (ERD) to describe the whole data model visually, and a JSON Schema version of the model to be used in some automated development efforts. For groups who are using Java, Java Script, Python, R, or Drupal, there are community maintained libraries available that contain full BrAPI implementations ready to be added to some existing code. The BrAPI Test Server and the BRAVA validation tool are both still available to the community for testing purposes, and they have been maintained to support every version of the specification. Finally, the three new resources list pages on brapi.org advertise the other BrAPI compatible software available in the community. The BrAPPs list displays the 10 standalone, plug-and-play, applications available to the community. The servers list displays the 27 registered public data servers, their current status, and a form for registering additional servers. The compatible software list shows the 31 software applications that are BrAPI compliant, and again, a form for registering additional applications. The process for registering new items for these lists is completely voluntary, so the totals represent a lower bound for the number of BrAPPs, data repositories, and applications available in the the community.

Community Growth

The international BrAPI Community consists of the software developers, breeders, and related scientists working on BrAPI related projects and data sources. This community is what sustains the BrAPI project, building implementations, maintaining development tools, and providing input to enhance the specification. As the project has grown, so has the community. The BrAPI project started in June 2014 with less than ten people coming together to discuss the idea. Over the next nine years, the community has grown to between 200 and 250 members. The community mailing list has 208 members, and the BrAPI Slack workspace has 234 members. The project leadership uses the mailing list to broadcast newsletters, announcements, and updates to the community. The BrAPI Slack workspace allows members of the community to discuss specific topics and collaborate directly with each other.

The BrAPI Hackathons are a major staple of the BrAPI community. Twice a year, the community gathers to discuss the specification and collaborate on BrAPI related projects. This time is very valuable to the community; for some organizations, the hackathon is the only time during the year when they have time to work on anything related to BrAPI. During the COVID-19 pandemic, virtual hackathons took the place of in person events. While the virtual hackathons do not provide the same level of face-to-face time that is crucial to collaborative work, they did allow for more attendees to gather and share their opinions. The typical virtual hackathon has about twice as many registered attendees compared to an in-person hackathon. However, attendees have reported much more productive work time during the in-person events. As a compromise, going forward, the community leadership has decided to have one in-person hackathon and one virtual hackathon each year, to balance the advantages of both.

As the project has matured, a formal project leadership structure became increasingly important. As described in Figure 2, the project governance is divided into two groups. The Project Management team is responsible for the day to day operations of the project. The PI and Co-PI are responsible for the project funds, and responsible for hiring the BrAPI Project Coordinator who is paid by the project funds. The Advisory Board is a group of elected officials representing the community. The board is responsible for long term planning of the project, as well as quick decision making on behalf of the community. The two groups meet quarterly to report on progress and stay synchronized.
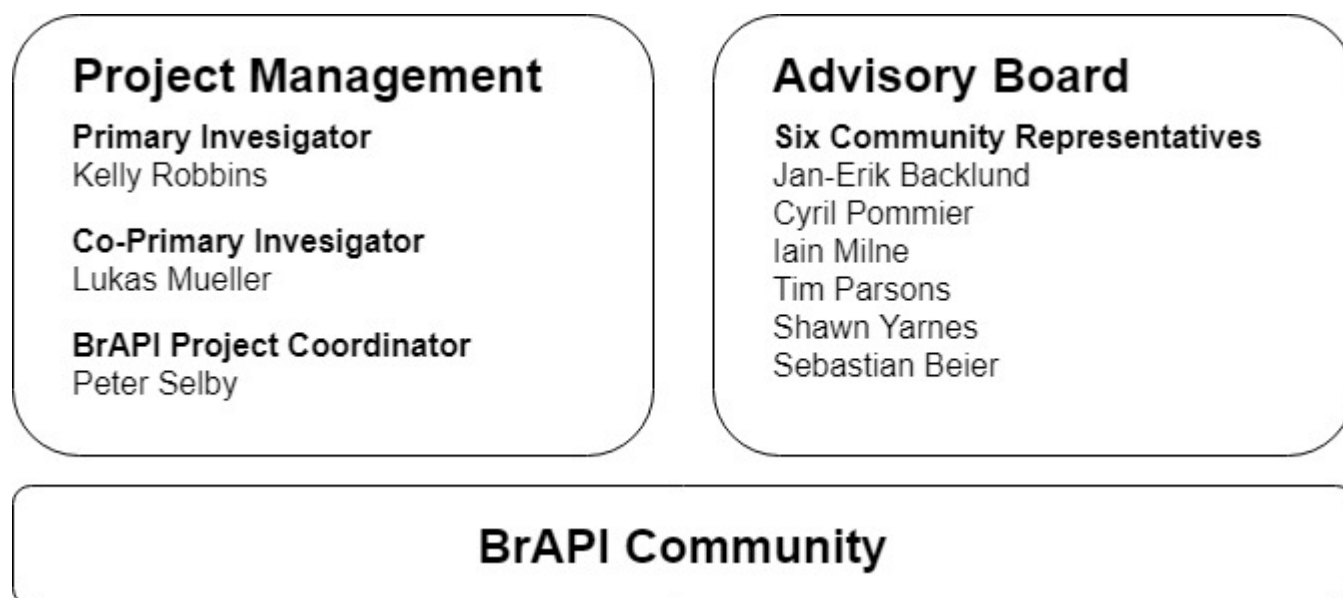
Figure 2: The formal governance structure of the BrAPI Project Community

Success Stories

6+/- success stories highlighting BrAPI usefulness in breeding cycle. Perhaps reference the original BrAPI paper where possible use cases were proposed.

Data Collection (Field Book, Climmob, other Phenoapps, etc)

- General use case description(s)
- Specific tool examples
- Alternate solutions/ why is it better with BrAPI
- future related use cases, areas to improve

Data Management (DeltaBreed, BMS, Breedbase, BIMS, Germinate, PHIS, etc)

- General use case description(s)
- Specific tool examples
- Alternate solutions/ why is it better with BrAPI
- future related use cases, areas to improve

Federated Data Management Infrastructures (AGENT, INCREASING, EURISCO, DataPLANT, NFDI4BioDiversity, FAIRAgro)

- General use case description(s) - AgrosystemIntegration of germplasm collections in context of data trustee models among private economy and public research, integration of ex-situ genbanks (EU H2020 projects AGENT, INCREASING), integrated agrosystems and plant research infrastructure)
- Specific tool examples - BraPI endponts for AGENT, IPK-Genbank, MIAPPE ISA-TAB2BRAPI service
- Alternate solutions/ why is it better with BrAPI - Schema.ORG lightweight meta data harvesting, ARCs as collaborative data decoration, API and publication pipeline
- future related use cases, areas to improve - LIMS to BrAPI proxies

Data visualization (Flapjack, Helium, Tassel, etc)

- General use case description(s)
- Specific tool examples
- Alternate solutions/ why is it better with BrAPI
- future related use cases, areas to improve

Analytics (QBMS, Mr Bean, etc)

- General use case description(s)
- Specific tool examples
- Alternate solutions/ why is it better with BrAPI
- future related use cases, areas to improve

brapi sync

- General use case description(s)
- Specific tool examples
- Alternate solutions/ why is it better with BrAPI
- future related use cases, areas to improve

Sample Submission (DArT) and Genotype DBs (MGIS, GIGWA)

- General use case description(s)
- Specific tool examples
- Alternate solutions/ why is it better with BrAPI
- future related use cases, areas to improve

Data Portal (FAIDARE, Phenospex?, etc)

- General use case description(s)
- Specific tool examples
- Alternate solutions/ why is it better with BrAPI
- future related use cases, areas to improve

Discussion

BrAPI for Breeders and Scientists

The BrAPI technical specification document is meant to be read and used by software developers. However, the purpose of the specification, and the community around it, is to make things faster, easier, and cheaper for the breeders and scientists working to make the world a better place. BrAPI offers a convenient path to automation and data integration for software tools in the breeding domain. All of the example use cases described above can be achieved with manual effort, moving and editing data files by hand. When the basic structure and flow of data becomes automated, breeders and scientists can spend more time focussing on the science and doing what they do best. For many, the ultimate goal is the development of a digital ecosystem: a collection of software tools and applications that can all work together seamlessly. In this digital ecosystem, data is collected digitally from the beginning, reducing as much human error as possible. The data is checked by quality control and stored automatically, then can be sent to any internal tool or external lab for further analysis with just the click of a button. This idea might sound too good to be true, but as more tools start sharing a universal data standard, automating data flow becomes easier, and the community gets closer to total interoperability.

Looking Ahead

The BrAPI specification will continue to grow, enabling more use cases and new types of data.

Conclusions and Impact

- High level summary of the project/consortium
- BrAPI is fitting into this gap, it doesn't need to fit these other gaps
- Call to action - Join us!

References

1. **BrAPI—an application programming interface for plant breeding applications**
Peter Selby, Rafael Abbeloos, Jan Erik Backlund, Martin Basterrechea Salido, Guillaume Bauchet, Omar E Benites-Alfaro, Clay Birkett, Viana C Calaminos, Pierre Carceller, Guillaume Cornut, …
Bioinformatics (2019-03-23) https://doi.org/gjgxxr
DOI: 10.1093/bioinformatics/btz190 · PMID: 30903186 · PMCID: PMC6792114