Notes on MCP (Model Context Protocol)

From Khaled Al'Shama

I would like to follow up on the exciting idea of integrating BrAPI endpoints with advanced LLM applications using the Model Context Protocol (MCP). MCP simplifies how APIs interact with LLMs like ChatGPT, as well as locally hosted models such as LLaMA 4 or DeepSeek-R1, enabling seamless, context-rich interactions that can significantly enhance the utility and accessibility of our endpoints.

Large Language Models (LLMs) such as ChatGPT are limited to their training data and cannot directly access external systems. MCP addresses this by providing a standardized interface for LLMs to interact dynamically with APIs, allowing them to retrieve real-time data and perform actionable tasks beyond their static knowledge.

Here is a minimal Python example demonstrating how easily we can wrap a BrAPI endpoint using FastMCP:

```python
from fastmcp import FastMCP
import requests

# Initialize the MCP server
mcp = FastMCP("BrAPI MCP Server")

# Define the tool to fetch trial information
@mcp.tool()
def get_trial_info(trial_id: str) -> dict:
     """
    Retrieve trial information from BrAPI given a trial ID.
    """
     url = f"https://myserver.org/brapi/v2/trials/{trial_id}"
     response = requests.get(url)
    response.raise_for_status()
     return response.json()

# Run the MCP server
if __name__ == "__main__":
     mcp.run()
```

This straightforward approach empowers LLM users and developers by allowing models to dynamically invoke BrAPI tools, greatly expanding our APIs' versatility and accessibility.

For a quick, practical introduction to MCP, I highly recommend this short online course:

MCP: Build Rich-Context AI Apps with Anthropic

I believe exploring this integration can significantly enrich our ongoing community/hackathon discussions regarding AI/ML integration with BrAPI

ChatGPT

You said:

How can BrAPI and AI/ML be used together?

ChatGPT said:

Great question! BrAPI (Breeding API) and AI/ML (Artificial Intelligence / Machine Learning) can complement each other powerfully in plant breeding and agricultural research. Here's how they can be used together:

What is BrAPI?

BrAPI is an open standard API designed to facilitate the exchange of plant breeding data — like genotypes, phenotypes, trials, germplasm, and more — in a standardized way. It helps breeders and researchers share and access large, diverse datasets across platforms and organizations efficiently.

What is AI/ML?

AI and ML involve algorithms that learn patterns from data to make predictions, classifications, or optimizations, often with minimal human intervention once trained.

How BrAPI and AI/ML can work together

1. Data Integration and Accessibility

- BrAPI standardizes and organizes data from multiple sources such as breeding trials, phenotypic measurements, and genotypic data.
- This structured data is ideal input for AI/ML models, which often require large volumes of clean, well-structured datasets.
- AI/ML can then analyze these integrated datasets to extract insights and patterns that breeders wouldn't see manually.

2. Predictive Modeling and Genomic Selection

- AI/ML models can use the genotypic and phenotypic data retrieved via BrAPI to predict traits of interest (e.g., yield, drought tolerance).
- This is especially useful for genomic selection, where breeders predict breeding values and select the best candidates early, saving time and resources.

3. Trait Discovery and Feature Selection

- Machine learning techniques can analyze BrAPI data to identify novel trait associations or markers.
- It can also select the most relevant features from complex, multi-dimensional data to improve prediction accuracy.

4. Decision Support Systems

- AI models trained on BrAPI data can power decision support tools for breeders, recommending crosses or optimal trial designs.
- For example, suggest which germplasm to cross to maximize a desired trait based on historical data.

5. Real-time Phenotyping and Monitoring

- When combined with IoT devices and sensor data (possibly accessible via BrAPI), AI can analyze real-time phenotyping data for stress detection, disease outbreaks, or growth monitoring.

Summary

- BrAPI = standardized access to breeding datasets
- AI/ML = advanced data analysis, prediction, and decision-making

Together, they create a powerful ecosystem for accelerating plant breeding by enabling better data-driven decisions, speeding up variety development, and improving crop resilience.

If you want, I can give you examples of specific AI/ML methods commonly used with BrAPI data or tools that implement both. Would you like that?

You said:

yes please

ChatGPT said:

Awesome! Here are some specific AI/ML methods commonly applied to breeding data accessible via BrAPI, along with examples of tools and workflows that combine both:

---

1. Genomic Selection / Genomic Prediction

- What it is:
  Predicting complex traits (yield, disease resistance, quality) using genomic marker data.
- Common AI/ML methods:
  - Ridge regression BLUP (RR-BLUP)
  - Random Forests

    - Gradient Boosting Machines (e.g., XGBoost)
    - Deep Neural Networks
    - Support Vector Machines (SVMs)
- How BrAPI helps:
  BrAPI endpoints provide standardized access to genotype and phenotype datasets needed to train these models.
- Example Tools:
    - G2P (Genotype to Phenotype) pipelines like those in R packages (e.g., `rrBLUP`, `BGLR`) combined with BrAPI clients.
    - Platforms like Breeding Insight or Integrated Breeding Platform (IBP) use BrAPI and genomic prediction modules.

---

2. Phenotype Image Analysis with Deep Learning

- What it is:
  Using AI models (CNNs) to analyze images from phenotyping platforms or field sensors to extract traits like disease severity or plant height.
- How BrAPI helps:
  BrAPI can serve as the backbone for integrating phenotypic data and metadata alongside image files or references, enabling seamless data management and linking images with genotypic info.
- Example Tools:
    - Deep learning frameworks (TensorFlow, PyTorch) for trait scoring on images.
    - Phenotyping platforms that expose data via BrAPI for downstream AI analysis.

---

3. Feature Selection and Trait Association

- What it is:
  Using ML algorithms to identify which genetic markers or environmental variables most influence traits.
- Common AI/ML methods:
    - LASSO regression
    - Random forests for variable importance

    - SHAP values (explainable AI) to interpret model outputs
- How BrAPI helps:
  It standardizes access to diverse datasets (genotype, phenotype, environment) facilitating multi-factor analyses.

---

4. Cross Prediction and Breeding Simulation

- What it is:
  Predicting the performance of potential crosses or simulating breeding cycles using AI.
- Methods:
    - Predictive modeling with genotype data
    - Optimization algorithms for cross selection
- How BrAPI helps:
  Provides up-to-date germplasm info, pedigree data, and trial results to feed into simulations.
- Example:
    - Software tools that use BrAPI to pull parental data and run AI-based cross prediction.

---

5. Decision Support Tools

- What it is:
  Combining ML predictions with user-friendly dashboards or apps that help breeders plan trials or select varieties.
- How BrAPI helps:
  Acts as the data layer allowing these tools to query current datasets programmatically.
- Example:
    - Platforms like Breedbase integrate BrAPI and machine learning models with UI for breeder decision-making.

Summary Table

Use Case	AI/ML Methods	Role of BrAPI	Example Tools
Genomic Selection	RR-BLUP, Random Forest, DNN	Access genotypic & phenotypic data	rrBLUP, BGLR, Breeding Insight
Image-based Phenotyping	CNNs (Convolutional Neural Nets)	Link images & metadata	TensorFlow models + BrAPI
Feature Selection	LASSO, Random Forest, SHAP	Integrate multi-source data	Custom ML pipelines
Cross Prediction	Predictive models, optimization	Pull pedigree and germplasm data	Breeding simulation software
Decision Support	Various ML + UI layers	Real-time querying of datasets	Breedbase, IBP

If you'd like, I can help draft an example workflow or sample code snippet showing how you might pull data from BrAPI and use it for training a model — just let me know!

You said:

how would you improve the BrAPI data model?

ChatGPT said:

Improving the BrAPI data model involves balancing standardization, flexibility, and extensibility to meet the evolving needs of modern breeding and ag-tech workflows — especially when integrating with AI/ML. Here are some well-grounded suggestions to improve BrAPI's data model:

🔧 1. Better Support for High-Throughput Phenotyping (HTP) and Sensor Data

Why?

HTP platforms (drones, imaging, sensors) are increasingly common, but BrAPI's current data model has limited support for complex, time-series, or image-rich data.

Improvements:

- Introduce a time-series phenotypic observation model, capturing data across time and space.
- Add standardized linking for image, video, and sensor stream metadata.
- Include fields for sensor calibration, platform metadata, and environmental context (light, soil, weather at time of capture).

🔗 2. Enhanced Provenance and Metadata Tracking

Why?

In AI/ML, data lineage is critical for reproducibility and model validation.

Improvements:

- Strengthen data provenance fields — capture:
    - Data source and method (e.g. manual scoring, LiDAR, hyperspectral).
    - Observer/device ID.
    - Preprocessing steps.
- Add versioning and data audit trails so that users can track how data has changed over time.

🧬 3. Expanded Support for Multi-Omics Data

Why?

Breeding programs are moving beyond genotypes to include transcriptomics, proteomics, metabolomics, etc.

Improvements:

- Extend the genotype model to support multi-omics assays.
- Use modular schemas or linked ontologies to describe:
    - Assay types
    - Sample prep methods
    - Raw vs. processed values

🤖 4. ML/AI Readiness Enhancements

Why?

AI/ML thrives on structured, contextual, and interoperable data.

Improvements:

- Ensure all numeric fields specify units, value types, and ranges to prevent ambiguity during model training.
- Add ML-friendly annotations, such as:
    - Trait heritability estimates
    - Data quality flags
    - Feature importance scores (optional)
- Provide sample-ready data export formats (e.g., directly export phenotype matrix + genotype matrix in CSV/Parquet).

🌍 5. Geospatial and Environmental Context Improvements

Why?

Environmental factors are often crucial for accurate trait modeling (G×E interactions).

Improvements:

- Standardize geospatial metadata (e.g., GPS resolution, spatial uncertainty).
- Add support for integrated weather/soil/environmental data layers, possibly via external links (e.g., to APIs from AgMIP, NASA, FAO).
- Store geoJSON for field trials or plot maps.

📦 6. Improved Ontology and Trait Management

Why?

AI models require consistent, well-defined traits and variables.

Improvements:

- Enforce stronger linkage to controlled vocabularies and ontologies (e.g., Crop Ontology, Planteome).
- Add tooling or endpoints for ontology reconciliation, alias mapping, and trait harmonization.
- Enable tagging of traits with data modality labels (e.g., numeric, ordinal, image-derived).

🧩 7. Plug-in / Extension Mechanism

Why?

Breeding workflows vary across crops and institutions — custom fields are common.

Improvements:

- Provide an official extension schema (e.g., `customFields`, `extensions`) with namespacing.
- Define how clients/servers can negotiate optional fields or support domain-specific needs.

Summary: Top Priorities

Area	Improvement Focus
Phenotyping Data	Time-series, images, sensor metadata
Metadata & Provenance	Data source, method, version tracking
Multi-Omics Integration	Omics assay models, standardized format
AI/ML Readiness	Units, value types, quality flags, model-ready exports
Geospatial & Environmental	GPS resolution, external data links, G×E modeling support

Ontology & Trait Consistency	Stronger trait harmonization, ontology reconciliation
Schema Extensibility	Modular, namespaced extensions for custom use cases

---

Would you like me to mock up a JSON example of an improved BrAPI phenotype observation with AI/ML-friendly metadata?

You said:

yes please

ChatGPT said:

Great! Below is a mock JSON example of an improved BrAPI phenotype observation that includes enhancements for AI/ML use, sensor metadata, time-series support, and ontological clarity.

📝 Scenario: A drone captures canopy height at different time points in a maize field. The observation is linked to a controlled trait ontology, includes GPS and sensor metadata, and is marked with a quality flag.

✅ Enhanced BrAPI Phenotype Observation (JSON)

json

Copy

Edit

```
{ "observationDbId": "obs12345", "observationUnitDbId": "plot789",
"observationVariableDbId": "trait:canopy_height_v1",
"observationVariableName": "Canopy height (drone-derived)",
"traitOntologyReference": { "ontologyName": "CropOntology", "ontologyId":
"CO_123:0000456", "traitClass": "Plant Height", "dataType": "Numeric",
```

```
"unit": "cm" }, "value": 172.3, "valueType": "numeric", "valueUnit": "cm",
"timestamp": "2025-06-02T10:15:00Z", "season": "2025",
"observationTimeSeries": { "timePoint": "25_DAS", "timePointDescription":
"25 days after sowing" }, "geoLocation": { "latitude": 41.235,
"longitude": -93.612, "altitude": 315.2, "spatialAccuracy": "±1.5m" },
"method": { "methodDbId": "method_drone_rgb_v1", "methodName":
"Drone-based RGB canopy analysis", "description": "Processed RGB images to
estimate canopy height using stereo reconstruction" }, "sensorMetadata": {
"sensorType": "Drone RGB Camera", "model": "DJI Phantom 4 Pro",
"sensorResolution": "5472x3648", "flightHeightMeters": 15.0,
"captureAngle": "Nadir", "weatherAtCapture": { "temperatureC": 28.4,
"windSpeedKph": 5.2, "cloudCoverPercent": 10 } }, "dataQuality": {
"qualityFlag": "PASS", "confidenceScore": 0.95, "validationNotes":
"Validated against ground-truth plot heights within ±5cm." },
"observationSource": { "collectedBy": "FieldBot01", "dataCuratedBy":
"j.doe@cropinst.org", "lastUpdated": "2025-06-02T16:00:00Z" },
"extensions": { "mlFeatureEngineeringHint": { "group":
"image_derived_traits", "standardizeForModeling": true,
"featureImportanceRank": 3 } } }
```

🔍 Highlights of the Improvements

Feature	Purpose
`traitOntologyReference`	Ensures machine-readable trait identity and units
`sensorMetadata`	Critical for reproducibility and data interpretation
`observationTimeSeries`	Captures time-series nature of observation
`dataQuality`	Helps filter or weight data in AI/ML pipelines

`extensions.mlFeatureEngineeringHint`	Optional ML guidance for downstream tools

✅ Benefits for AI/ML

- Structured, standardized data makes preprocessing minimal.
- Units, value types, and trait identity are machine-readable.
- Time, space, and method metadata improve context modeling (e.g., G×E).
- Quality metrics and flags let models selectively trust inputs.

Would you like a version of this compatible with a real BrAPI client or something you can load into a tool like Breedbase or a Jupyter notebook?

You said:

can you do the same thing for the Variant model in BrAPI

ChatGPT said:

Absolutely! Here's a mock JSON for an improved BrAPI Variant model, designed to be AI/ML-ready, multi-omics extensible, and bioinformatically rich — useful for genomic prediction, GWAS, and ML pipelines.

🧬 Scenario: A variant detected in a maize inbred line via whole-genome sequencing, linked to ontology terms, annotated with its functional impact, and enriched with population frequency, quality, and ML-relevant metadata.

✅ Enhanced BrAPI Variant (JSON)

json

Copy

Edit

```
{ "variantDbId": "var98765", "variantName": "SNP_chr3_1849275_TG",
"referenceDbId": "refGenome_v5", "referenceName": "Maize_B73_v5",
"referenceSetDbId": "refset_maize_v5", "chromosome": "3", "start":
1849275, "end": 1849275, "referenceBases": "T", "alternateBases": ["G"],
"variantType": "SNP", "callSetCount": 423, "variantSetDbId":
"variantset_2025_wgs_maize", "contextSequence": "CTGACTTT[T/G]ACGAACT",
"info": { "qualityScore": 97.8, "depth": 44, "alleleFrequency": 0.37,
"minorAllele": "G", "minorAlleleFreq": 0.37, "heterozygosity": 0.21,
"functionalEffect": { "effect": "missense_variant", "impact": "MODERATE",
"geneId": "Zm00001d027438", "transcriptId": "Zm00001d027438_T001",
"codonChange": "ATC>GTC", "aminoAcidChange": "Ile>Val", "ontology": {
"source": "SO", "term": "SO:0001583", "label": "missense_variant" } },
"traitAssociations": [ { "traitName": "Grain yield", "ontologyId":
"CO_123:0000654", "associationType": "GWAS", "pValue": 3.2e-06,
"effectSize": 0.28 } ], "mlFeatureEngineeringHint": {
"includeInPolygenicScore": true, "featureImportanceRank": 42,
"encodingRecommended": "one-hot" } }, "provenance": { "discoveryMethod":
"Whole Genome Sequencing", "pipelineVersion": "GATK v4.4",
"variantCaller": "HaplotypeCaller", "callDate": "2025-05-30T13:45:00Z" },
"source": { "createdBy": "j.smith@genomicslab.org", "curatedBy":
"curator@breedbase.org", "lastUpdated": "2025-06-03T09:15:00Z" } }
```

🔍 Key Enhancements Over Standard BrAPI Variant

Section	Enhancement
`contextSequence`	Flanking bases for AI models using sequence context
`functionalEffect`	Detailed bioinformatics annotation, linked to gene/transcript + ontology
`traitAssociations`	Useful for GWAS/G2P ML pipelines — links variant to trait relevance
`mlFeatureEngineeringHint`	Optional ML guidance: importance rank, encoding style, inclusion advice
`provenance`	Ensures reproducibility, version tracking, and trust in data
`info.qualityScore`, `depth`, `alleleFrequency`	Quality and frequency fields directly usable in ML filtering or weighting

---

🧠 Benefits for AI/ML Workflows

- ML-readiness: Encoding guidance and importance scores aid automated feature selection.
- Bioinformatics integration: Functional effects and annotations can feed into interpretable models.
- Trait linking: Helps with supervised learning, QTL prioritization, and multi-trait optimization.
- Contextual depth: Improves biological signal modeling, especially with models that integrate sequence or functional context.

Would you like this format aligned to an actual BrAPI server's schema (e.g. Breedbase's variant loader), or a version exportable to Python/pandas for AI modeling?

Attach

Search

Reason

Voice