



BeePop+

An Agent-Based Honey
Bee Colony Model for
Pesticide Risk Assessment

Jeff Minucci and Tom Purucker

U.S. EPA, Office of Pesticide Programs
Environmental Fate and Effects Division

1st ICPPR Symposium on Bee Effect Models

March 17, 2026

Disclaimer: The views expressed in this presentation are those of the author(s) and do not necessarily represent the views or policies of the U.S. EPA.

Why Do We Need Colony Models?

The US EPA uses a **tiered pollinator risk assessment framework** that progresses from conservative assessments of toxicity to individual bees (Tier I) through refined semi-field and field studies (Tiers II-III)

Tier I (individual)

Individual-level effects;
toxicity studies on
individual bees



Image credit: eurofins

Tier II (semi-field)

Colony-level effects;
tunnel, colony feeding
studies



Image credit: Univ. of NE

Tier III (full-field)

Colony-level effects;
actual crops, typical use
patterns, natural foraging



Image credit: Penn State Univ.

Why Do We Need Colony Models?

Higher-tier studies (semi-field, field) have increased realism but high cost and complexity, can be difficult to interpret. **Weather, pathogens and high inter-colony variability** often reduce these studies' power to detect adverse effects of pesticides

Weather



Pathogens



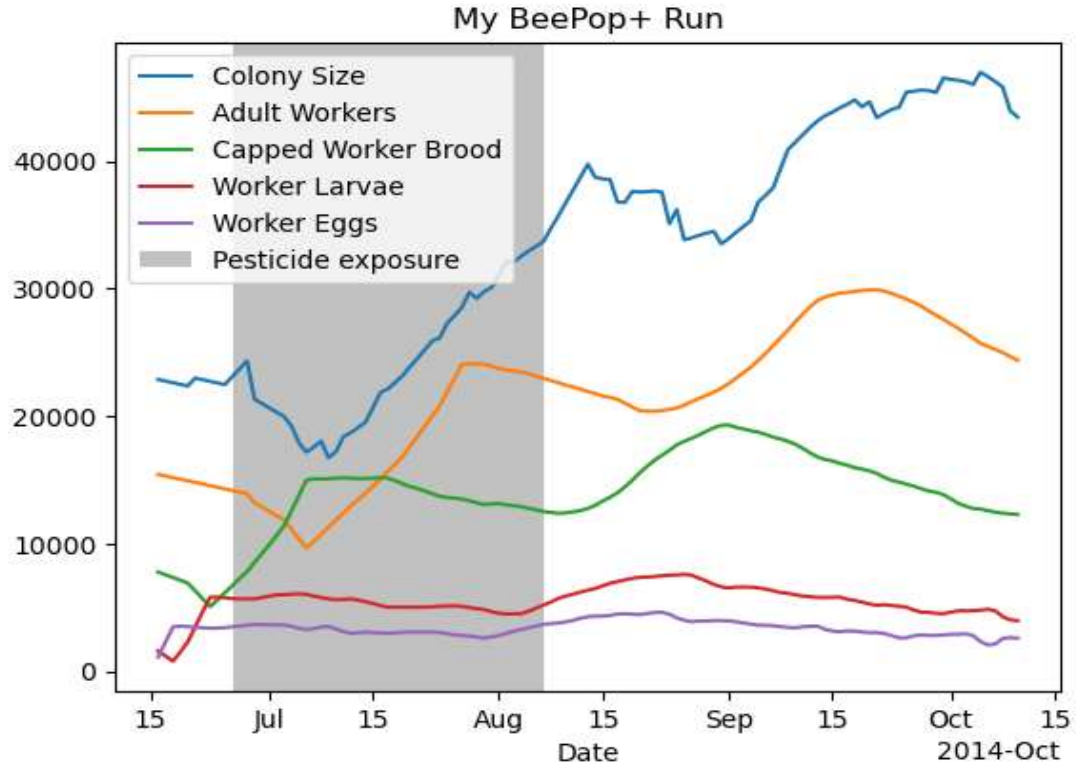
High inter-colony variability



Why Do We Need Colony Models?

Colony models have the potential to extrapolate Tier 1 toxicity data to colony-level outcomes under various exposure scenarios

They can be used as an **ecologically-grounded higher-tier model** to inform the ecological risk assessment and risk management process



BeePop+: An Agent-Based Colony Model



Developed by EPA and USDA as an extension of the USDA BeePop/VarroaPop model (DeGrandi-Hoffman et al., 1989, 2004), adding pesticide exposure and effects modules (Garber et al., 2022)

BeePop+ General Characteristics



Agent-based model (ABM) that simulates cohorts of bees (e.g., queen, workers, brood, foragers) interacting with each other and their environment

Simulates realistic colony dynamics over time (queen egg-laying, brood development, pesticide-driven mortality, pathogens, foraging behavior, food consumption) with weather, seasonal cycles

Includes multiple stressors which can be simulated together – pesticide exposure + parasitic *Varroa* mite populations + effects

Python based model (ported from original C++ code)

Simulating the Colony Lifecycle

Queen: individual agent whose egg-laying rate varies according to

- Queen strength (max. egg laying rate)
- Season (temperature + photoperiod)
- Number of foragers and in-hive nurse bees
- Sperm reserves (impacts drone production)

Workers/Drones: Each day's eggs form a cohort that progresses through its life stages on a set timeline

Eggs

3 days



Larvae

5 days



Pupae

13 days



Adults

variable lifespan



In-hive workers

Foragers

Foraging and Resource Dynamics

- Foraging Model**
- Non-spatial:** No landscape model; resources assumed to be non-limiting and uniformly available within foraging range
 - Fixed trip parameters:** Each forager makes a set number of pollen and nectar trips daily, collecting a set load per trip
 - Weather limited:** Temperature 12–43°C, low wind (≤ 8.94 m/s), and low rainfall (≤ 0.197 mm)
 - Task switching:** When no larvae present, only nectar trips
- Food consumption**
- Age and caste-specific:** Individual daily pollen and nectar needs differ by developmental stage, adult age group, and caste

Foraging and Resource Dynamics

Colony food balance: Daily intake from foragers minus daily consumption by all colony members determines net change in pollen and nectar stores

Daily collection (g) = Active Foragers × Trips/day × Load/trip

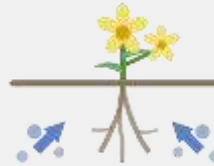
Daily consumption (g) = Σ (individuals in age/caste group × daily need)

Four Pesticide Application Modes



Foliar spray

- Spray application during bloom
- Residues decline over time based on foliar dissipation rate parameter
- Contact exposure to foragers from plant surfaces



Soil application

- Pesticide applied to soil taken up through plant roots into pollen and nectar
- Residue levels depend on active ingredient properties and soil characteristics



Seed treatment

- Coating on seeds is absorbed systemically as the plant grows
- Modeled as constant residues in pollen and nectar throughout exposure period



Direct residue specification

- User provides a daily time series of pollen and nectar residues directly
- Maximum flexibility for field-measured data, fate and transport model output

Modeling Pesticide Exposure

Pesticide exposure is modeled as a chain from environmental residues through foraging, food storage, and consumption to individual daily doses for each caste and age cohort



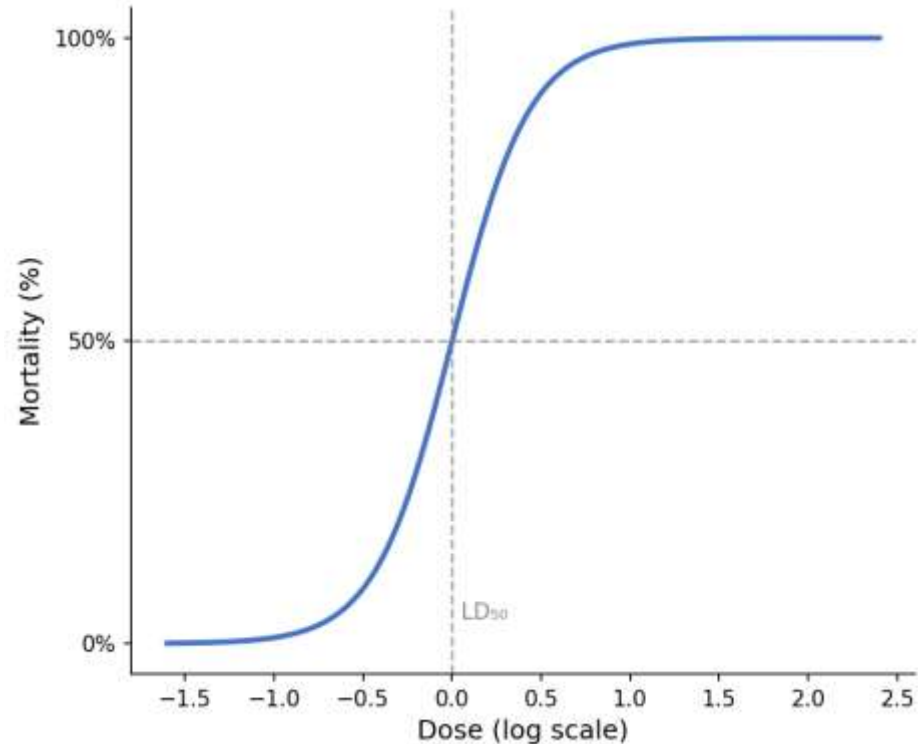
From Dose to Effects

Log-logistic dose-response:

Percent of bees dying per cohort based on LD_{50} and slope

Stage-specific toxicity: Separate LD_{50} and slope values adults, larvae

Acute effects only: Each simulation day, mortality is based on daily dose. Surviving bees are unaffected by future exposures unless the dose is larger.



Running BeePop+: Primary Inputs

Simulation Dates Start and end dates for the simulation period

Weather Data Daily temperature, wind speed, rainfall

Colony Initial Conditions Number of adult workers, brood, stored pollen and nectar, queen strength, *Varroa* mite infestation rates (optional)

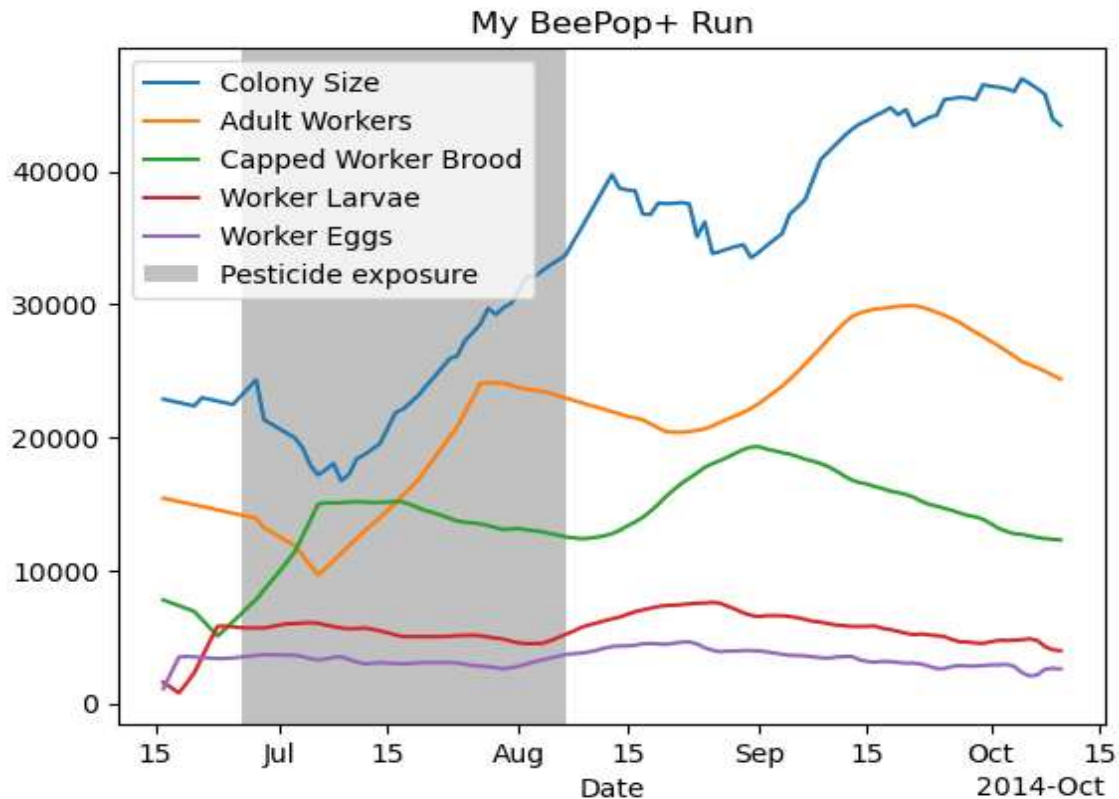
Toxicity Parameters LD₅₀ and slope for larvae & adults (from standard Tier I lab studies)

Exposure Data & Parameters Application rate & timing, or daily concentrations in pollen and nectar; BeeREX standard exposure assumptions

Running BeePop+: Output

Time series output
(pandas DataFrame)

44 output columns (daily)
including colony size, adult workers, foragers, brood, food stores, pesticide residues in food stores, and mortality

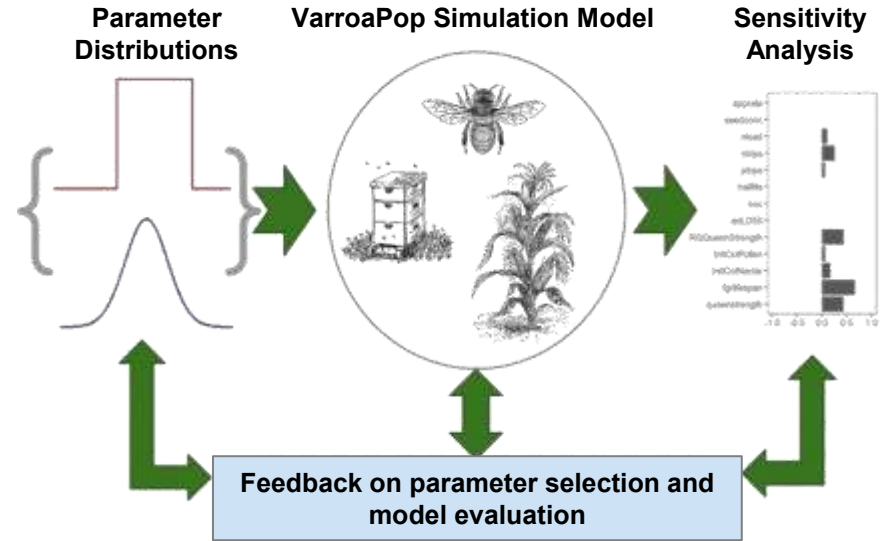


BeePop+ Parameters and Customization

A global sensitivity analysis

identified model parameters with high influence on colony outcomes under various scenarios (Kuan et al., 2018):

- **Queen strength** (max. egg-laying rate): primary driver of colony size across all scenarios
- **Forager lifespan:** longer-lived foragers sustain resource collection and colony growth



BeePop+ Parameters and Customization

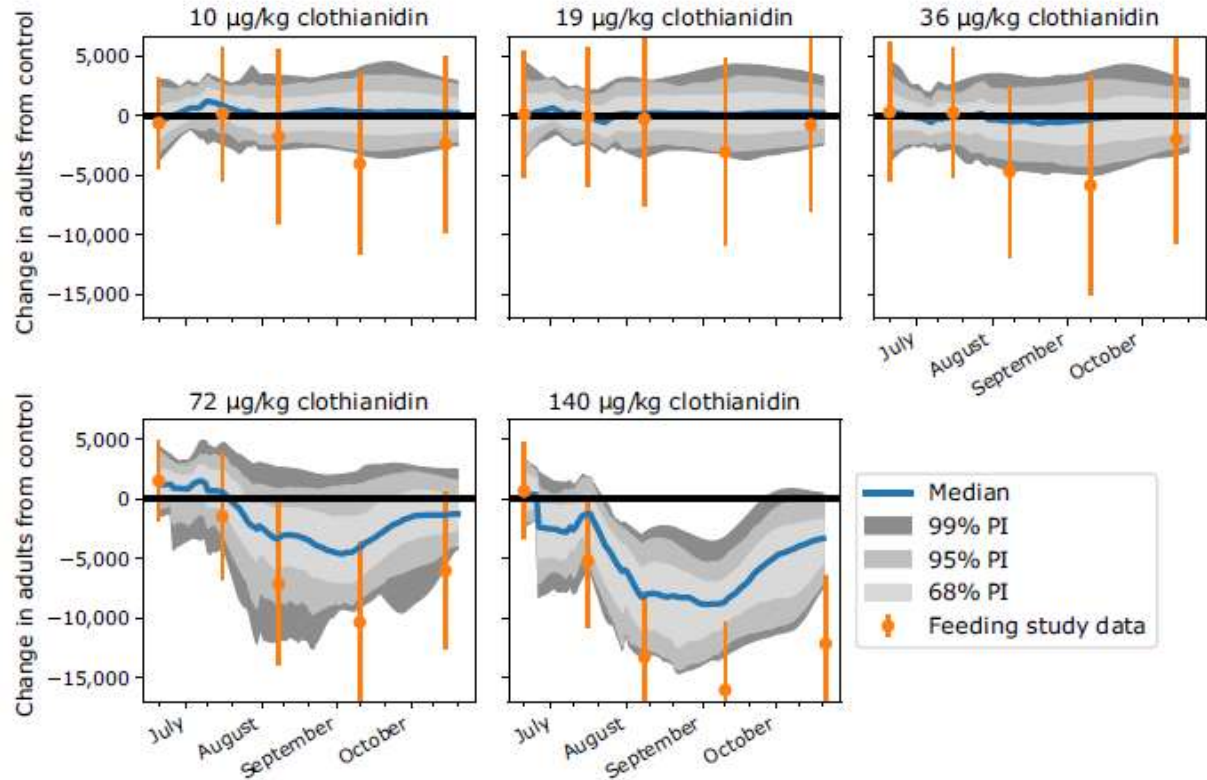
111 parameters are available for model fine tuning and customization, but most can be left at their default values:

- Foraging behavior (6)
- Food consumption rates (16)
- Requeening behavior (7)
- *Varroa* mite population dynamics (12)
- *Varroa* mite treatments (5)
- Supplemental nectar feeding (8)
- Mortality rates during lifestage transitions (8)

Validating with Colony Feeding Studies

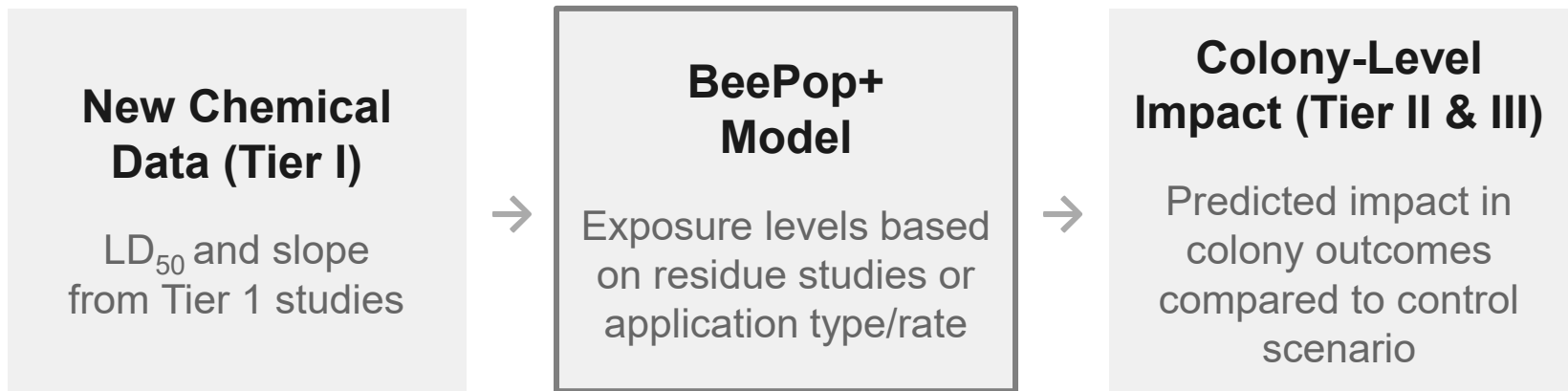
Proof of concept study (2021) demonstrated that **BeePop+** could be calibrated to fit colony feeding study data for **clothianidin**

On-going research is comparing model output to colony feeding study data across **12 different pesticides**



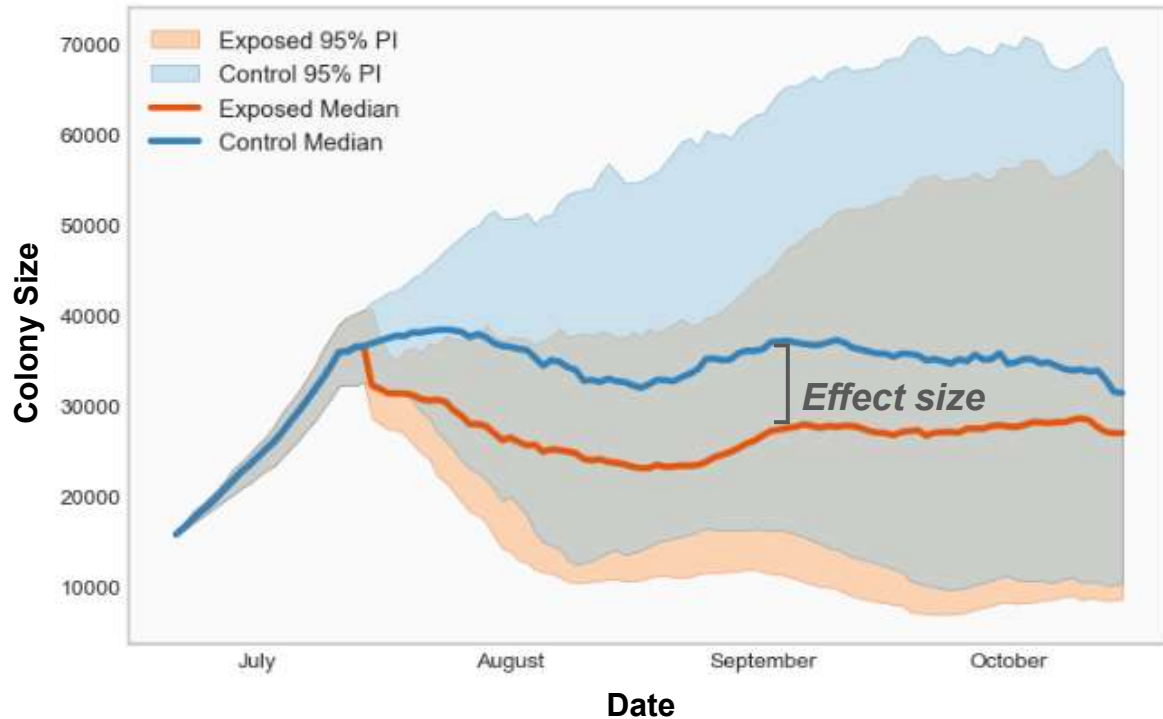
BeePop+ as a Regulatory Tool

BeePop+ is being developed as a **risk assessment model** that sits between Tier I and Tier II — augmenting the risk assessment process with mechanistic, colony-level predictions



BeePop+ as a Regulatory Tool

Control vs Exposed: Colony Size



If BeePop+ predicts no significant impacts to the colony, we can potentially eliminate need to call in higher tier semi-field studies

If BeePop+ *does* predict significant impacts, predictions could be used to inform ecological risk assessment and risk management

pybeepop+: Running BeePop+ in Python

```
beepop = PyBeePop()

# Define inputs and file paths
START_DATE = '06/16/2014'
END_DATE = '10/10/2014'
run_parameters = {"ICWorkerAdults": 23000, "ICWorkerBrood": 8000, "SimStart": START_DATE,
                  "SimEnd": END_DATE, "IPollenTrips": 8, "INectarTrips": 17}
weather = os.path.abspath('example_data/cedar_grove_NC_weather.txt')
residue_file = os.path.abspath('example_data/example_residue_file.txt')

# Load inputs into BeePop+
#beepop.beepop.set_latitude(55)
beepop.load_weather(weather)
beepop.load_residue_file(residue_file)
beepop.set_parameters(run_parameters)

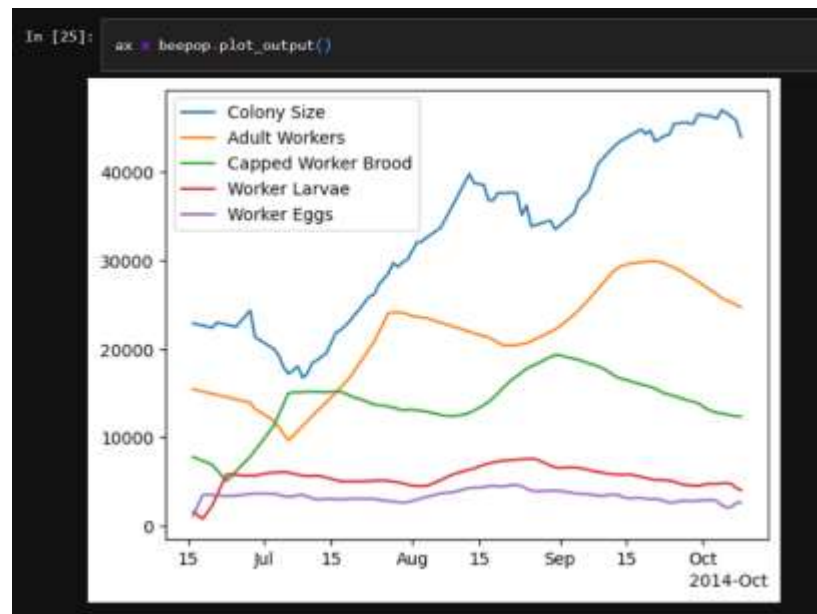
[4]: ✓ 0.0s Python
```

Run the model and print the results DataFrame

```
results = beepop.run_model()
print(results)

[5]: ✓ 0.0s Python
```

	Date	Colony Size	Adult Drones	Adult Workers	Foragers	\
0	Initial	23000	0	15561	7439	
1	06/16/2014	22874	0	15435	7439	
2	06/17/2014	22748	0	15309	7439	
3	06/18/2014	22622	0	15183	7439	
4	06/19/2014	22496	0	15057	7439	



<https://github.com/USEPA/pybeepop>

pybeepop+: Running BeePop+ in Python

Extensive documentation is also available on the model website:

<https://usepa.github.io/pybeepop/>



pybeepop+ Documentation

🌟 Introduction & Quick Start

Overview, installation, and usage instructions for pybeepop+.

[Read the Introduction →](#)

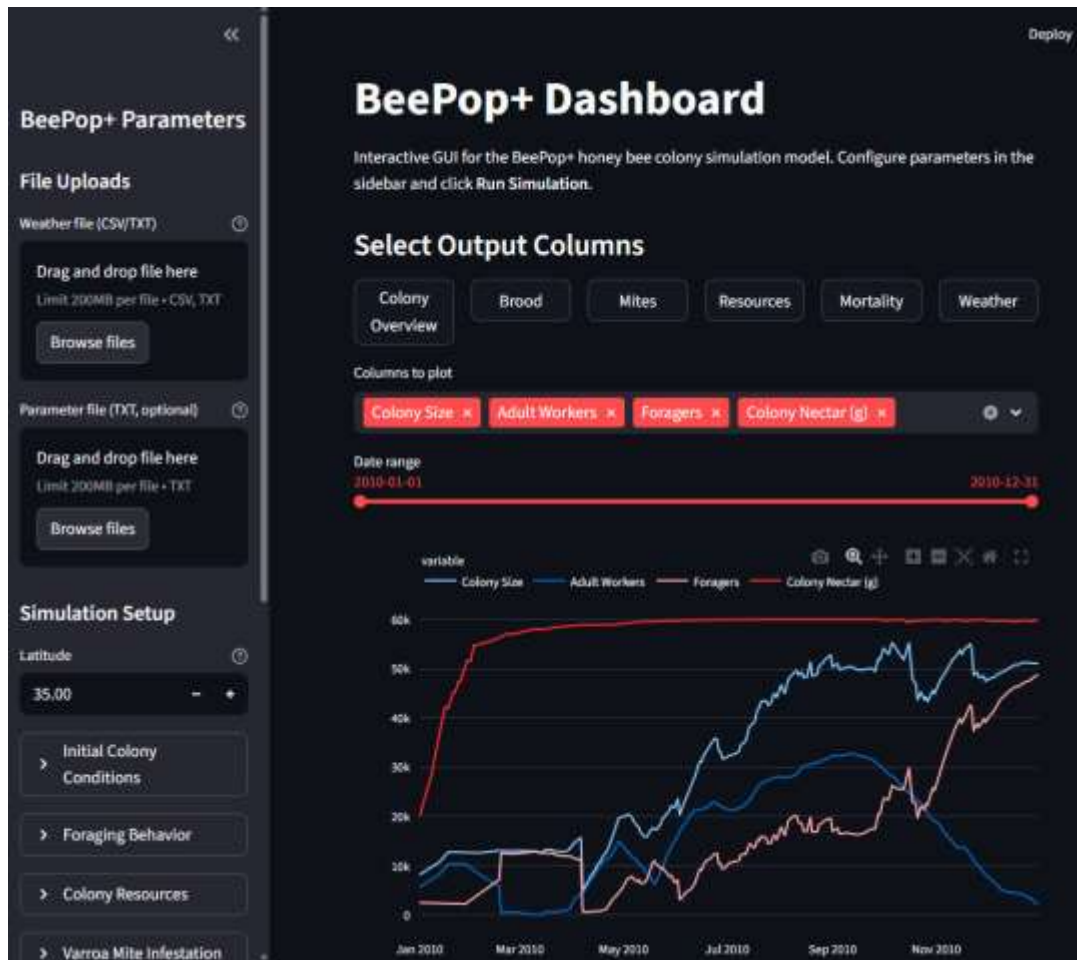
🐝 BeePop+ Model Parameters

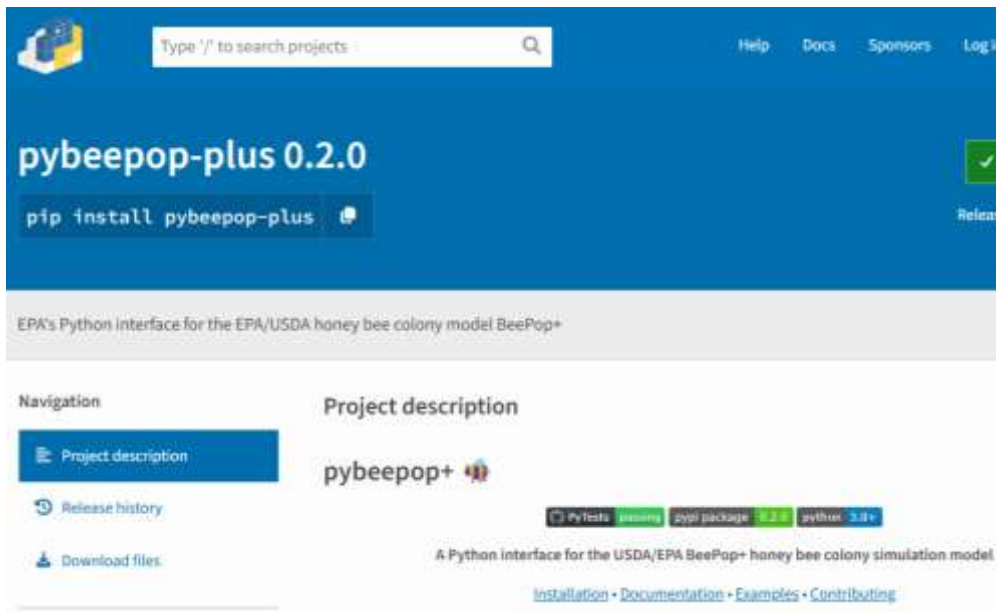
Browse the full list of BeePop+ model parameters, descriptions, and default values.

[View Parameter List →](#)

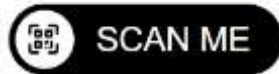
Coming soon: BeePop+ Graphical Interface

**Graphical user
interface** for
BeePop+ is in
development and
expected to release
in late 2026





Thank you!



To install in Python:

pip install pybeepop-plus

Jeff Minucci

minucci.jeffrey@epa.gov

Office of Pesticide Programs

Environmental Fate and Effects Division



Tom Purucker

purucker.tom@epa.gov

Office of Pesticide Programs

Environmental Fate and Effects Division