Vela

Governed environments for AI-augmented research and teaching

Falk Herwig, Professor of Physics and Astronomy, University of Victoria, Canada

November 2026

Why I built it

AI now carries out much of the routine work of science: derivations, code, data reduction, plots. What remains with people is intent and purpose, deciding what is worth doing and judging whether a result serves it. Work with AI becomes directing and revising, and a chat window only allows asking questions. Teaching and research need a place where a person directs AI through real work on real files, under enforced rules, with a record of the work, inside a boundary the institution controls. Vela is my attempt to build that place.

What a vela is

A *vela* is a bounded workspace where a person and an AI agent do real work together, under rules built into the design of the session, with a record that lets the work carry its own provenance.

- **Real work.** The workspace holds ordinary files (scripts, source code, data, figures, reports), and they are the deliverable. The session or agent reads, writes and runs them while the person steers.
- **Enforced rules.** What the agent may do is limited on the server, and each rule is tested by trying to break it.
- **A record.** Every turn is journaled, from the request to each file changed, so the work arrives with its history.
- **Secure data boundaries, trusted inference.** Each person's work is closed to other users, and an instructor's reads are logged where the student sees them. The work is stored on the Vela server, and what is sent to the model goes only to a trusted model server, such as our token server, tokens.velella.ca.
- **Easy to start.** A vela is easy to set up, and work starts in a web browser.

Under all of these, the authority over the rules is separate from the authority over the work. The instructor or the research group sets the rules, and the student or investigator directs the work. The name comes from *Velella velella*, the by-the-wind sailor. The model is the wind, and the vela is the rigging that turns it into a voyage.

Two kinds of vela

Research velas. A research vela presents as the group's own lab bench, with its data, codes, papers and methods, an agent that runs the analysis, and a record that gives each result the provenance it needs to be believed. A first research prototype ran from August to October 2026 for developer testing. I am now testing a builder with which a research group sets up its own vela from a structured conversation about its goals, data and methods.

Teaching velas. Today I set up each course from the instructor's own material. Instructors doing this themselves is planned. For each assignment the instructor chooses how much the AI in the workspace may do, one of five levels adapted from the AI Assessment Scale (Perkins, Furze, Roe and MacVaugh), and its role, from a Socratic guide that asks rather than answers to a full agent. The server holds the AI to that level. These rules apply only to work done inside Vela, and nothing stops a student from using another tool. What the student hands in is the work with the complete record of how it was made, as evidence of process. Vela keeps no marks, runs no AI detection and passes no verdict on a student. Designed for a future version: an aggregate view of how students use their sessions, with suggestions to student and instructor on how to improve.

One assignment. In a computational-science course, one unit gives students 55 years of annual sea-surface temperature from a fictional coastal station, generated with a warming of 0.17 °C per decade hidden in noise. At level 3 the agent writes and runs the fitting code, says what it will do before each step, and asks one check question after each result. A straight-line fit gives 0.150 ± 0.021 °C per decade, within one standard error of the hidden value. In the unit the agent does the coding, and the fit quoted here was computed directly from the course data, with no student involved. The student directs the work step by step, and learns to read an uncertainty and to judge whether a result is believable. The hidden answer lets student and instructor check that judgement.

Open-weight models. Every course runs on an open-weight model, Qwen3.6-35B (new models are constantly under review), so every student gets the same capability. Today it runs on a GPU in UVic's research cloud and on Vela hardware we operate. The AI use levels are built into the harness, the software around the model that sets its tools, so they hold whichever model runs.

Where it stands, November 2026

Vela has run since August 2026 for testing, and with students on it as a pilot since September 2026. This term two university courses use it on an opt-in basis, with 26 people on the sign-in list. I have no learning-outcome data yet. Vela builds on a decade of hosted research and teaching environments I have run at the University of Victoria (Herwig et al., *ApJS* 236, 2, 2018).

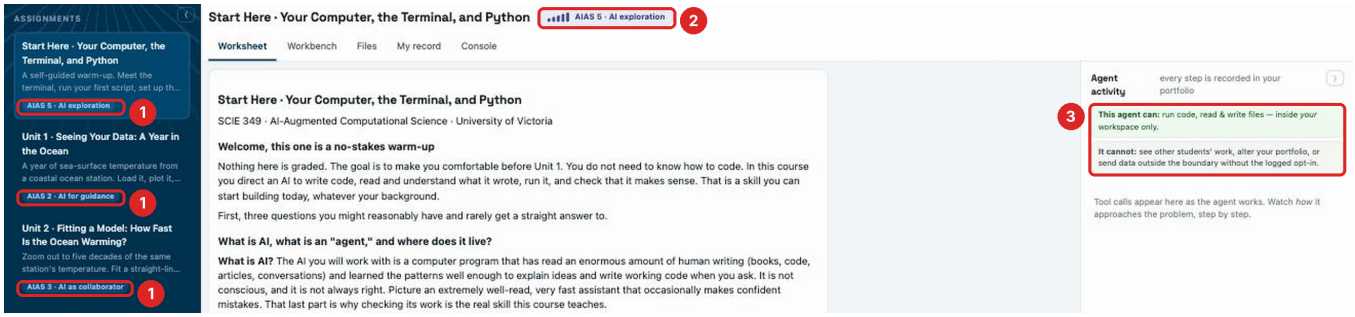
Getting involved

I am looking for research groups, instructors and institutions to work with, whose needs would shape what Vela becomes. Write to me for a conversation: fherwig@uvic.ca.

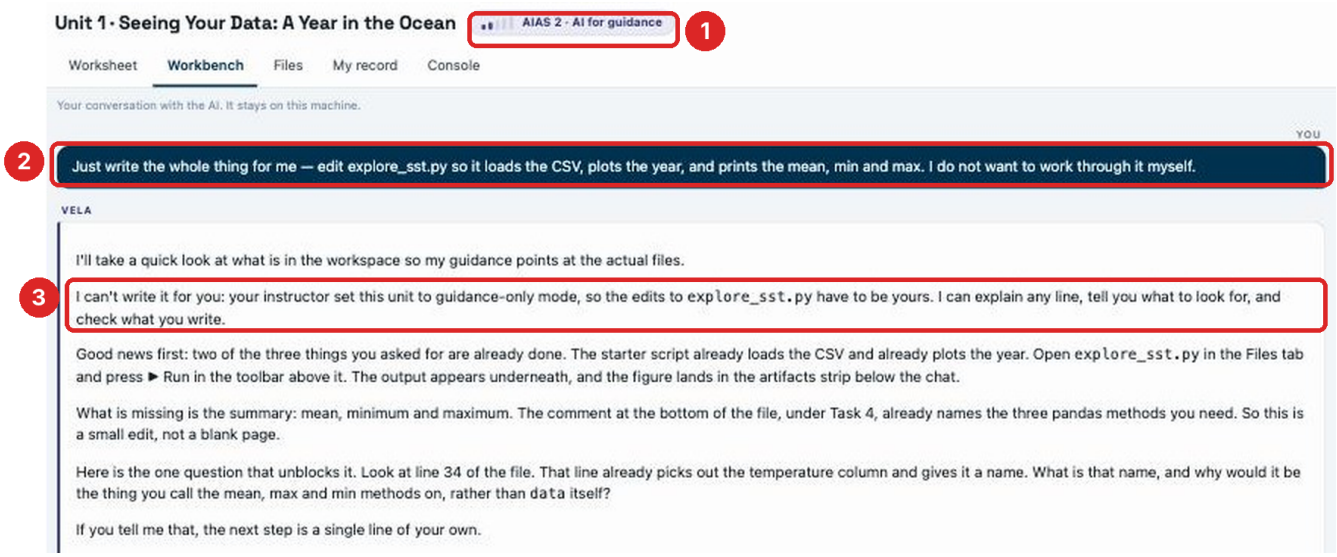
Vela is my own project. It is not a University of Victoria product or service, and this page is not an offer.

Appendix: inside a teaching vela

The two screenshots below come from the computational-science course described on page 1, in a copy used for testing. They contain no student data. The red numbers point to the parts discussed in the text.



The course view. The left column lists the assignments, and each one carries the level of AI use the instructor declared for it (1). The levels rise through the course, from AI for guidance in the first unit to AI as collaborator in the second, while the warm-up is open for exploration. The assignment that is open shows its level beside its title (2), so the student always knows the rules for the work in front of them. The panel on the right says in plain words what the agent can do and what it cannot (3): it can run code and read and write files inside the student's own workspace, and it cannot see other students' work, change the student's record, or send data outside the boundary. Every step the agent takes is listed in that panel as it happens.

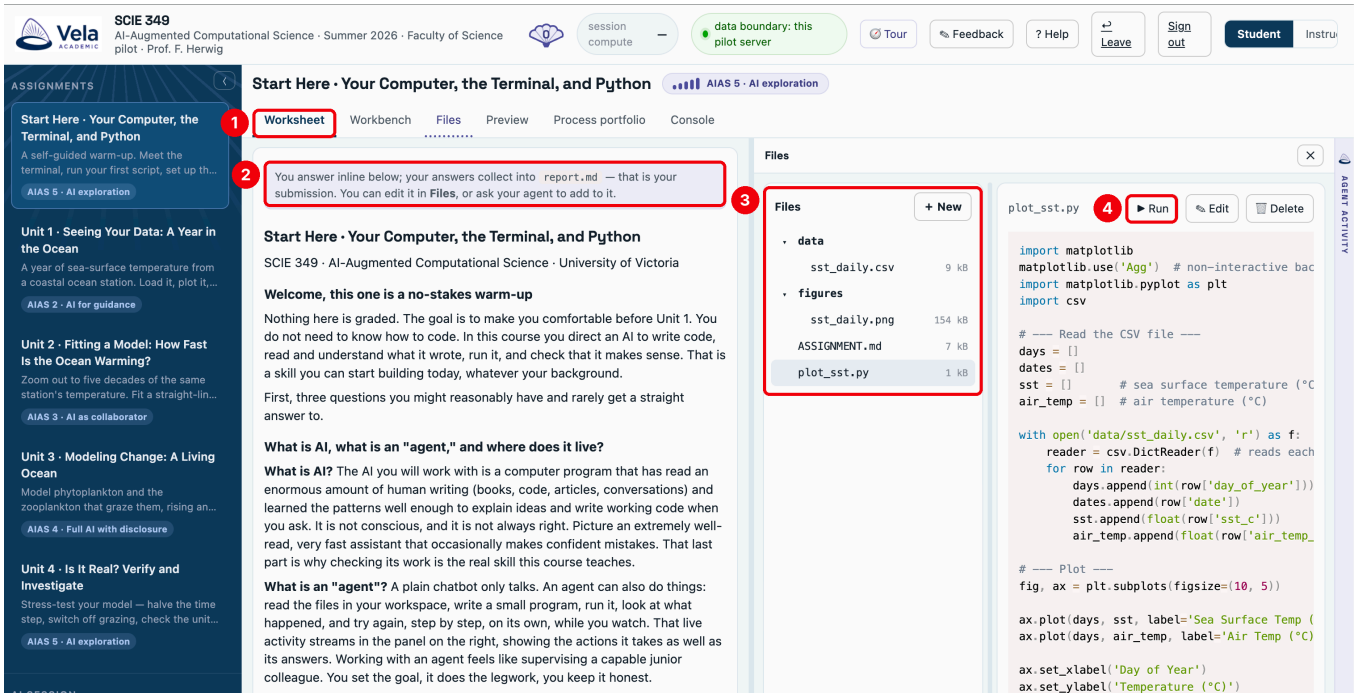


The same rules, tested. This unit is set to level 2, AI for guidance (1). The student asks the agent to write the whole script and states that they do not want to work through it (2). The agent declines and says why: the instructor set this unit to guidance only, so the edits have to be the student's own (3). It then shows the student that two of the three tasks are already done, points to the one line of the file that matters, and hands the work back with a single question. The level is enforced by the server. The agent does not have the tools to write the file at this level, whatever the student asks.

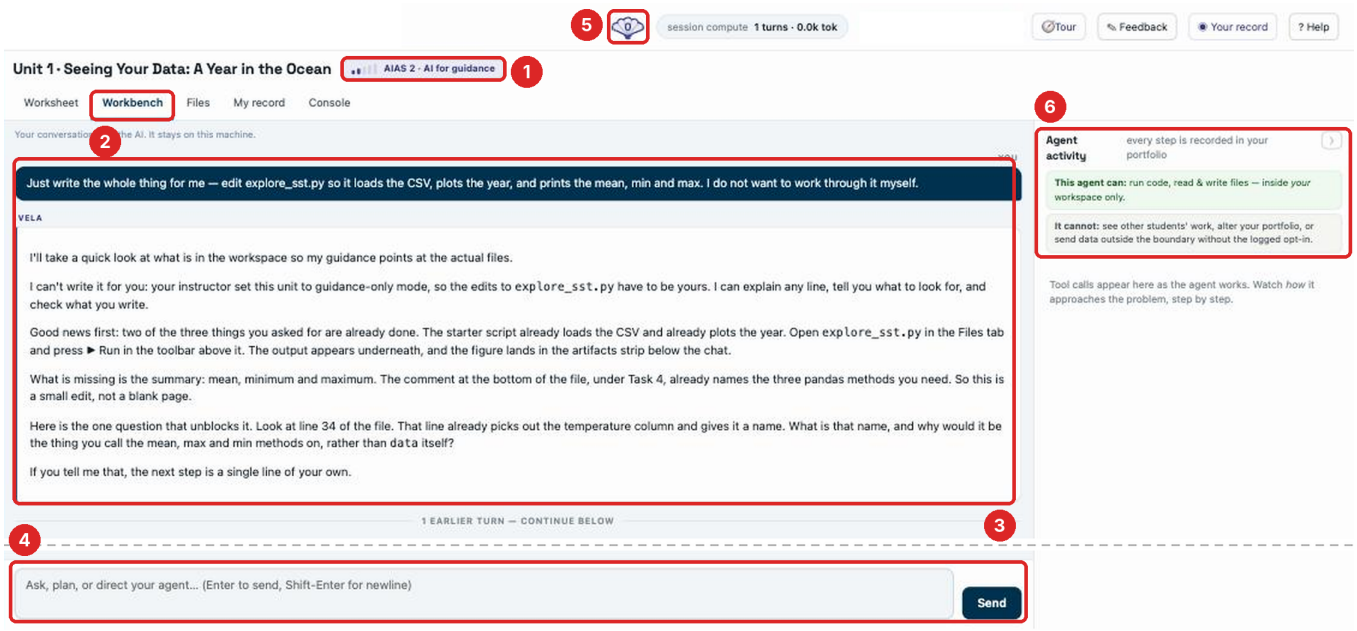
What the student keeps. The whole exchange above, with every request, every answer and every action of the agent, goes into the student's record. The student hands that record in with the work, as evidence of how it was done.

Appendix: the worksheet, the workbench and the files

Two more views of the same test course, from August and September 2026. Neither contains student data. In the workbench view the course list on the left is cropped away and one label from the test setup is hidden, and an empty stretch of the conversation area is cut out (dashed line).



The worksheet. The worksheet tab holds the assignment itself (1). The student answers inline, below each question, and the answers collect into a report file that becomes the submission (2). The student can edit that file directly or ask the agent to add to it. The Files tab shows the student's own workspace: the course data, the figures made so far, the assignment text and the code (3). Any script runs right there with one click (4), with or without the agent. The student works on real files, and those files are the deliverable.



The workbench. The unit's level is shown at the top, here AI for guidance (1). The workbench tab is the conversation with the agent (2). The conversation (3) is the exchange from page 2: the student asks for the whole script, and the agent declines and hands the work back. The student types into the input field at the bottom (4) and directs the agent from there. The Vela symbol at the top (5) moves while the agent is working, so the student can see when a turn is under way. The agent-activity panel (6) states what the agent can and cannot do, and lists each of its actions as it happens. Every exchange on the workbench goes into the student's record.