Evolvix Mission Statement

Evolvix Exists to
Make Accurate Modeling Easy

Evolvix exists

to make it easy for biologists to accurately translate between
their verbal models of the messy real-world biology they study and
the contradiction-free math models understood by computers.
This enables simulating measurable real-life predictions, uncertainty
quantifications, and assessments of model credibility, thereby
expanding the thinking capabilities of biologists.

Evolvix

Evolvix exists for enabling biologists collaborating across space and time to reliably communicate their questions about the systems they study in a precise language. Its words, phrases and synonyms are to intuitively, unambiguously, and accurately map to rigorous math concepts, precisely link real biological observations to reliable math models, and hence enable computers to expand the thinking capabilities of biologists.

Evolvix translates the unchanging precision of reliable formal results into diverse dialects for diverse human cultures by using unambiguous words, synonyms and phrases easily understood by humans and by computers.

Evolvix facilitates automated rigorous analysis, processing and combining of complex models, highlighting results irrespective of outcome: expected or unexpected predictions, variations, surprises, limitations, ambiguities, misinterpretations, inconsistences, inaccuracies, errors and any other information that can help assess the credibility of models or the evidence that supports or rejects them. Evolvix keeps adding tools to help automate all analyses that can be defined precisely, maximizing reliable results accessible to decision-support systems, helping humans to choose responsibly in repeatable experiments as well as unique opportunities.

The Evolvix Social Contract

Evolvix is the first general programming language designed by biologists for biologists (as far as I know). Hence Evolvix carries an important social and scientific responsibility by shaping the initial impression of programming languages for many students and self-pronounced 'non-programmers'.

A concise and precise description for a growing knowledge base of biology that can be readily understood by humans and computers requires Evolvix to integrate new approaches with proven abilities for taming complexity across disciplines. New capabilities must never be allowed to break previously trusted capabilities out of respect to those who have invested time and energy into writing and assessing models relying on previously trusted capabilities.

Therefore Evolvix must fiercely guard against unnecessary complexity caused by poor design and disciplinary idiosyncrasies. Thus the trusted core of Evolvix must grow slowly, so it can keep evolving without choking on its own complexity. Experimental features are to be as easily introduced as mutations in a population, however before their use can be trusted they have to
(1) survive repeated rounds of rigorous, globally collaborative, open trans-disciplinary peer-reviews,
(2) adequately address all fair topical criticism of that feature from any feedback, invited professional experts in relevant fields, global citizen scientists, and beginners to ensure accuracy, stability and ease of use, (3) to ensure the overall simplicity of Evolvix, all relevant facts have to be reviewed by at least one 'complexity guard'. Each complexity guard must have trans-disciplinary training, have an excellent understanding of the overall requirements, design, development roadmap, and more. Each must process all relevant facts, pros, cons, trade-offs, design alternatives, opportunity costs, and all other relevant aspects. Each has to agree the proposed capability has stabilized on an excellent design, has matured into a well supported abstraction, and does not close other more important design opportunities elsewhere.

Evolvix aims to radically reduce the time students, outsiders and experimental biologists spend on learning to read and write its code, understand, write, and analyze new models, in a way that is easy to reproduce. Thus Evolvix aims to find the simplest and most expressive way of enabling to:

① develop, test, analyze, auto-improve, navigate, concurrently edit, share, combine, refactor, compare/create/merge: **versions of models, source code**
② import, export, document, check: **strange external data formats**
③ define, check, look-up, apply: **defaults for hiding code complexity**
④ document, find, navigate, analyze: **code, changes, errors**
⑤ store, organize, search, filter, extract, compare, evaluate, sum up, transform, post-process, analyze, annotate, test, sign, encrypt, backup, archive, import, export, exchange, secure delete: **results, provenance, reliability**
⑥ compute, estimate, visualize: **uncertainty, numeric errors, robust stats**
⑦ pretty-print, refactor, auto translate to readable forms: **code**
⑧ produce, navigate, analyze, evaluate, annotate, delete: **relevant beautiful overviews, plots, reports, logs, infos, docs**
⑨ do, reproduce, improve: **all above by all in the future, without re-implementing, chasing missing info or unclear semantics.**
⑩ Learning for life, how to document, use, improve, build: **reliable user-friendly decision support systems that facilitate responsible decisions, protect privacy and honor the equal dignity of all humans.**