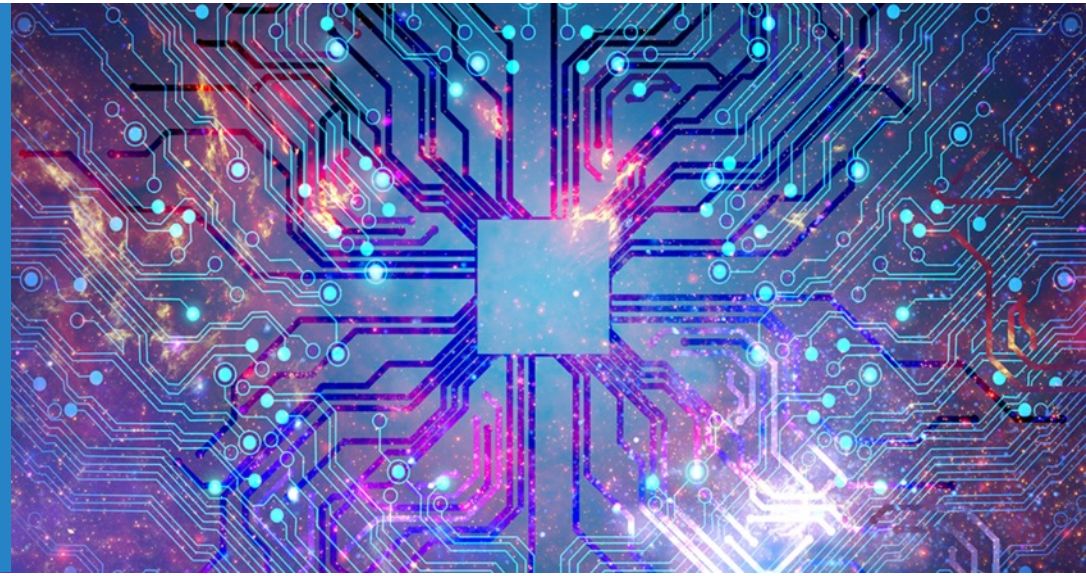
Integrating generative AI with automation and simulations for biological systems design



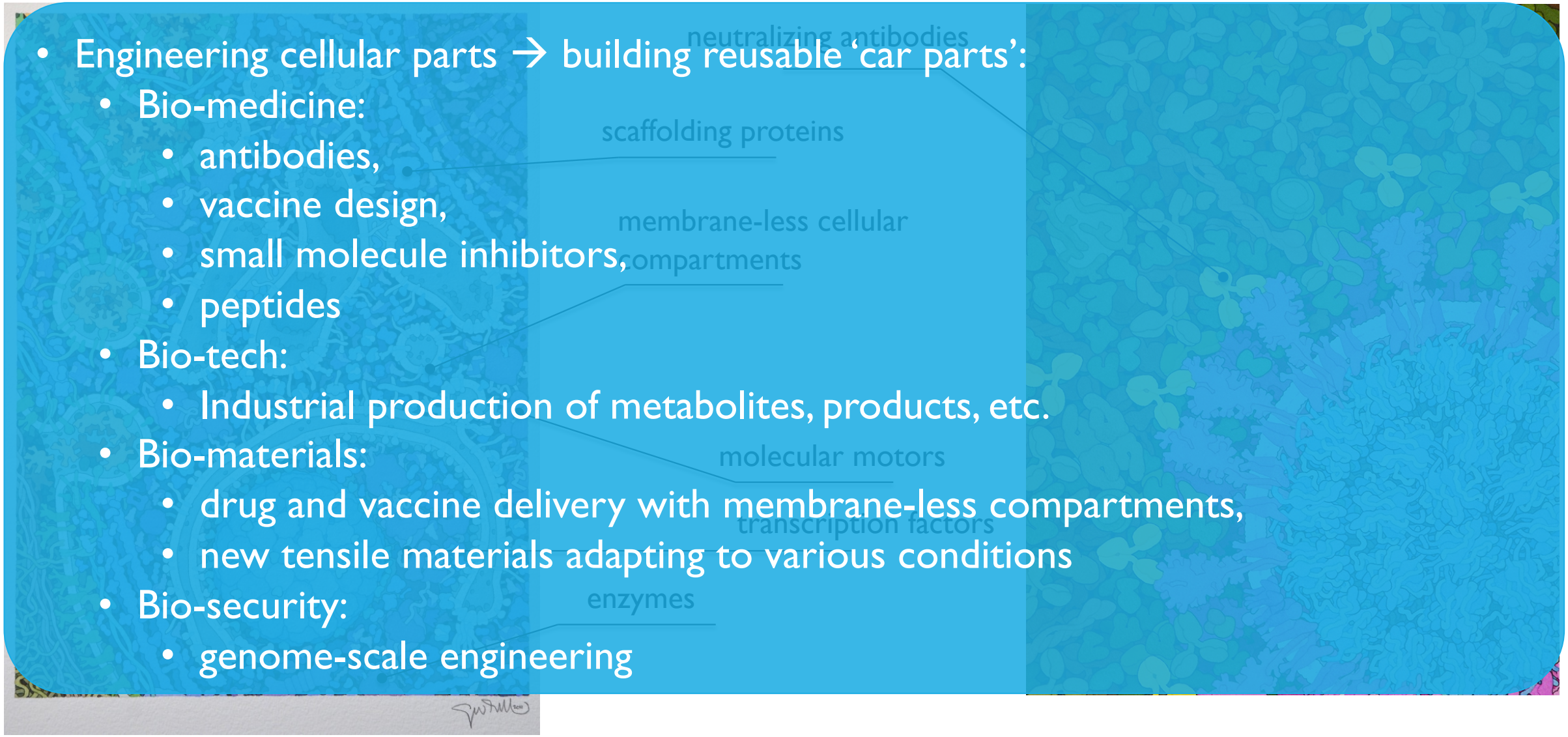
Arvind Ramanathan/ ramanathana@anl.gov

Argonne National Laboratory/ University of Chicago Consortium for Advanced Science and Engineering (CASE)

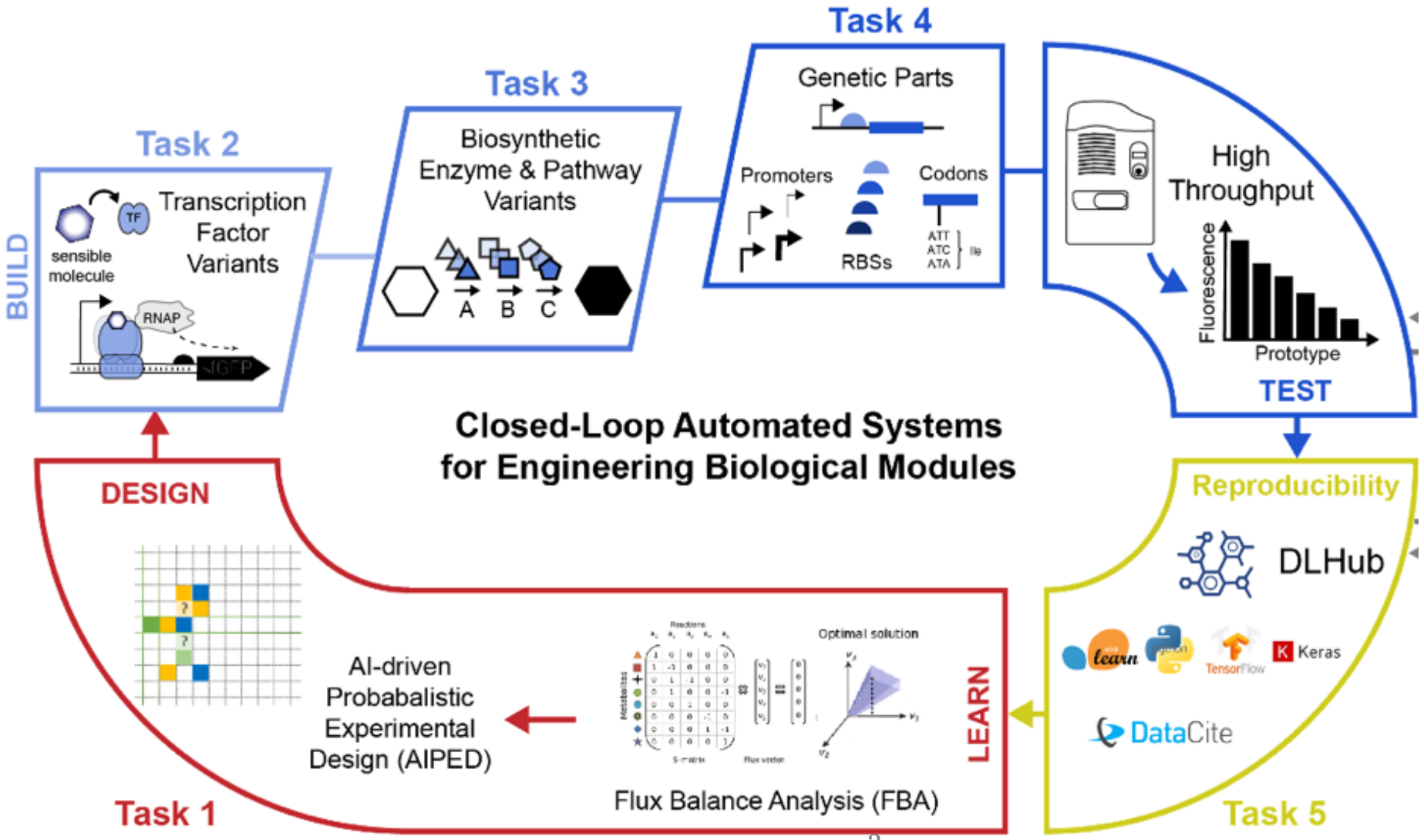
Northwestern-Argonne Institute for Science and Engineering (NAISE)

Autonomous robotic platform for designing "cellular parts"

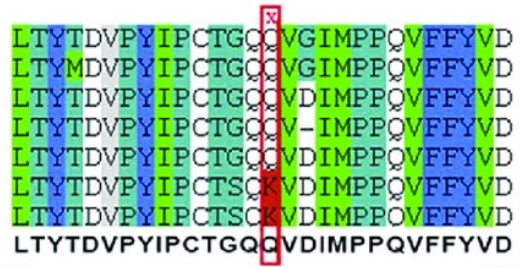
- Engineering cellular parts → building reusable 'car parts':
  - Bio-medicine:
    - antibodies,
    - vaccine design,
    - small molecule inhibitors,
    - peptides
  - Bio-tech:
    - Industrial production of metabolites, products, etc.
  - Bio-materials:
    - drug and vaccine delivery with membrane-less compartments,
    - new tensile materials adapting to various conditions
  - Bio-security:
    - genome-scale engineering

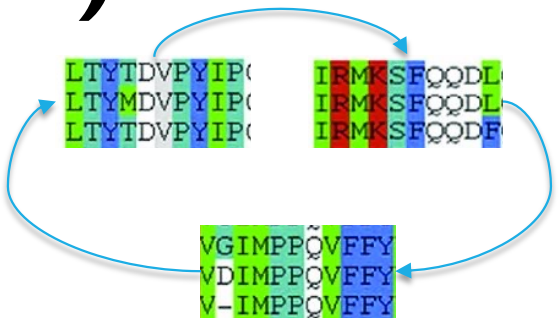


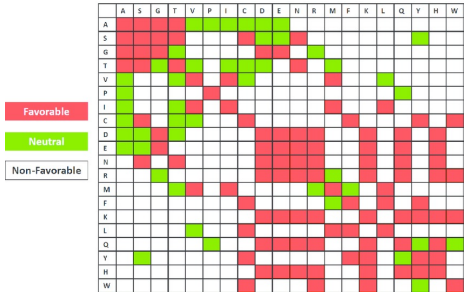
An integrative platform for engineering biological "parts"



Current paradigm of designing cellular parts is not scalable (and not sustainable)

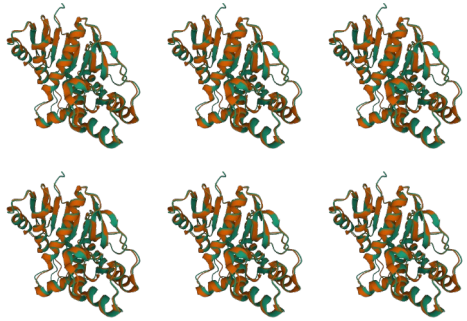




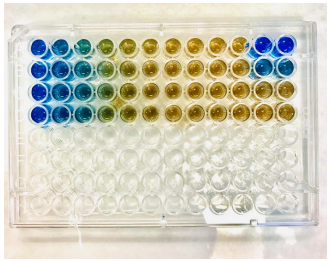


Parent sequences → Scaffold / active site identification → Combinatorial libraries → Score substitutions

- Only a negligible fraction ($1/10^{117}$ maximum) of the available sequence space can be searched
- Complicated further by combinatorial optimization involved in pathway designs



Selected sequences ← High-throughput protein screens/ assays ← Ensemble of ranked variants

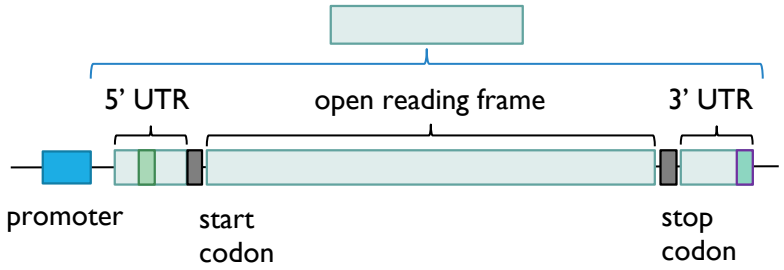


Outline (What this talk is about?)

- Learning representations for complex biological datasets
  - foundation models for genomes
  - genome-scale language models
- Scaling foundation models for genomic-scale data + generative models:
  - individual gene / protein level (malate dehydrogenase/ MDH as an example)
  - whole genome level (SARS-CoV-2 as an example)
- Embodied agents as scientific assistants for biological discovery:
  - autonomous laboratories / self-driving laboratories
  - teaching robots to write biological protocols
  - applications to antimicrobial discovery
- Future work/ perspectives

Biological information and hierarchy

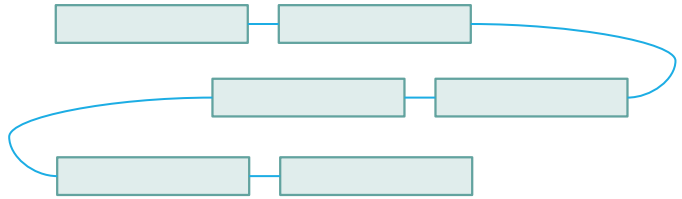
Hierarchical information representation for '-omics' data



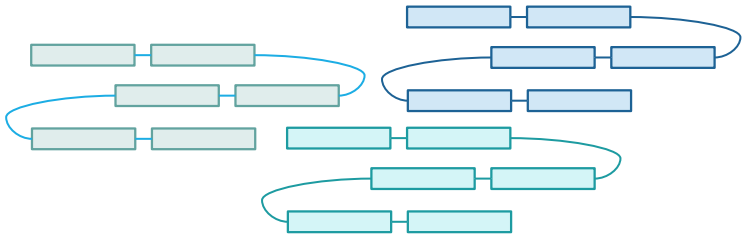
Enzyme/transcription factor sequences - codon level tokenization, GPT models

gene/ gene product (aka protein)

Open reading frames – codon level tokenization, Reformer model

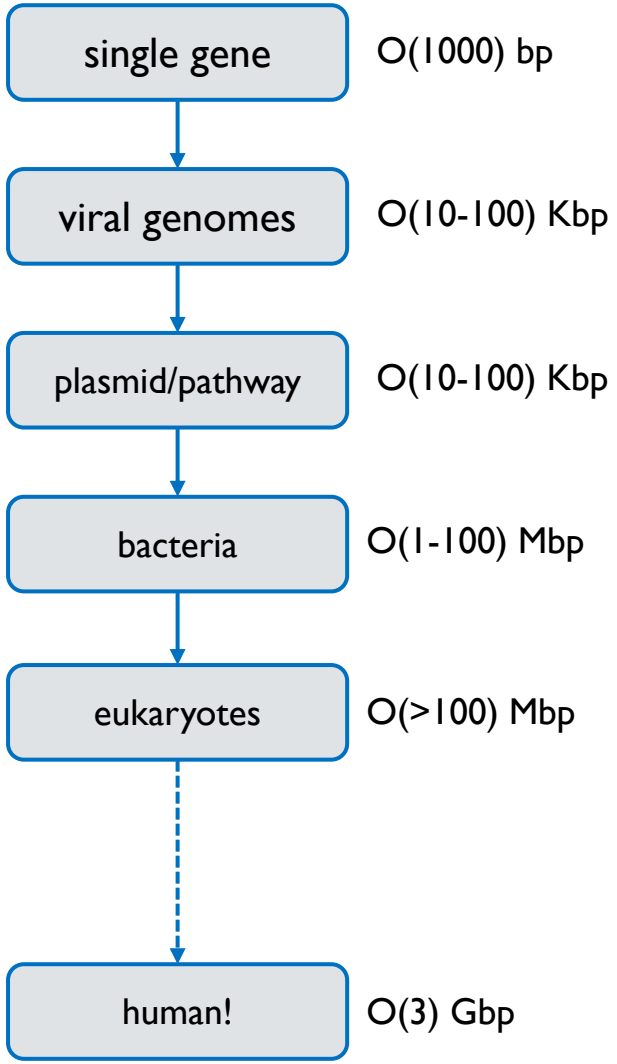


collection of genes (either as "contigs" or ORFs)

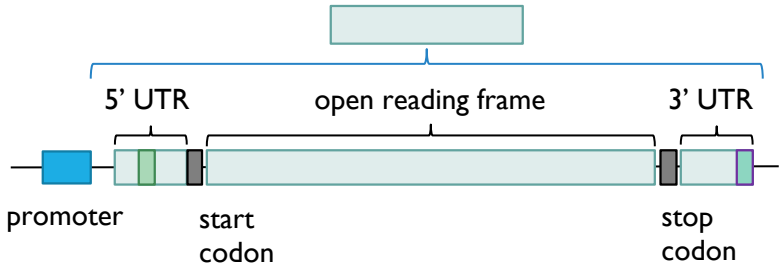


Full genome sequences - BPE Encoding, cannot currently generate full genomes

entire genomes



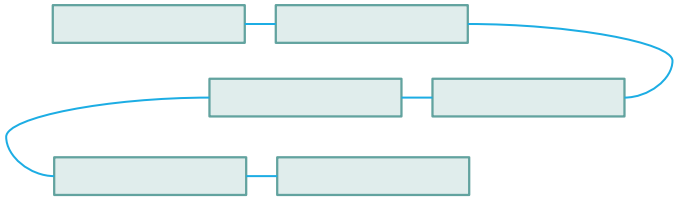
Genome-scale language models (GenSLM)



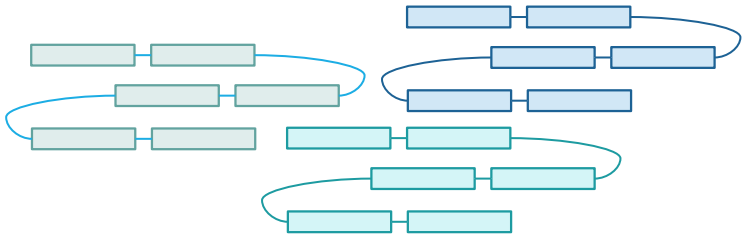
gene/ gene product (aka protein)

Enzyme/transcription factor sequences - codon level tokenization, GPT models

Open reading frames – codon level tokenization, Reformer model



collection of genes (either as "contigs" or ORFs)

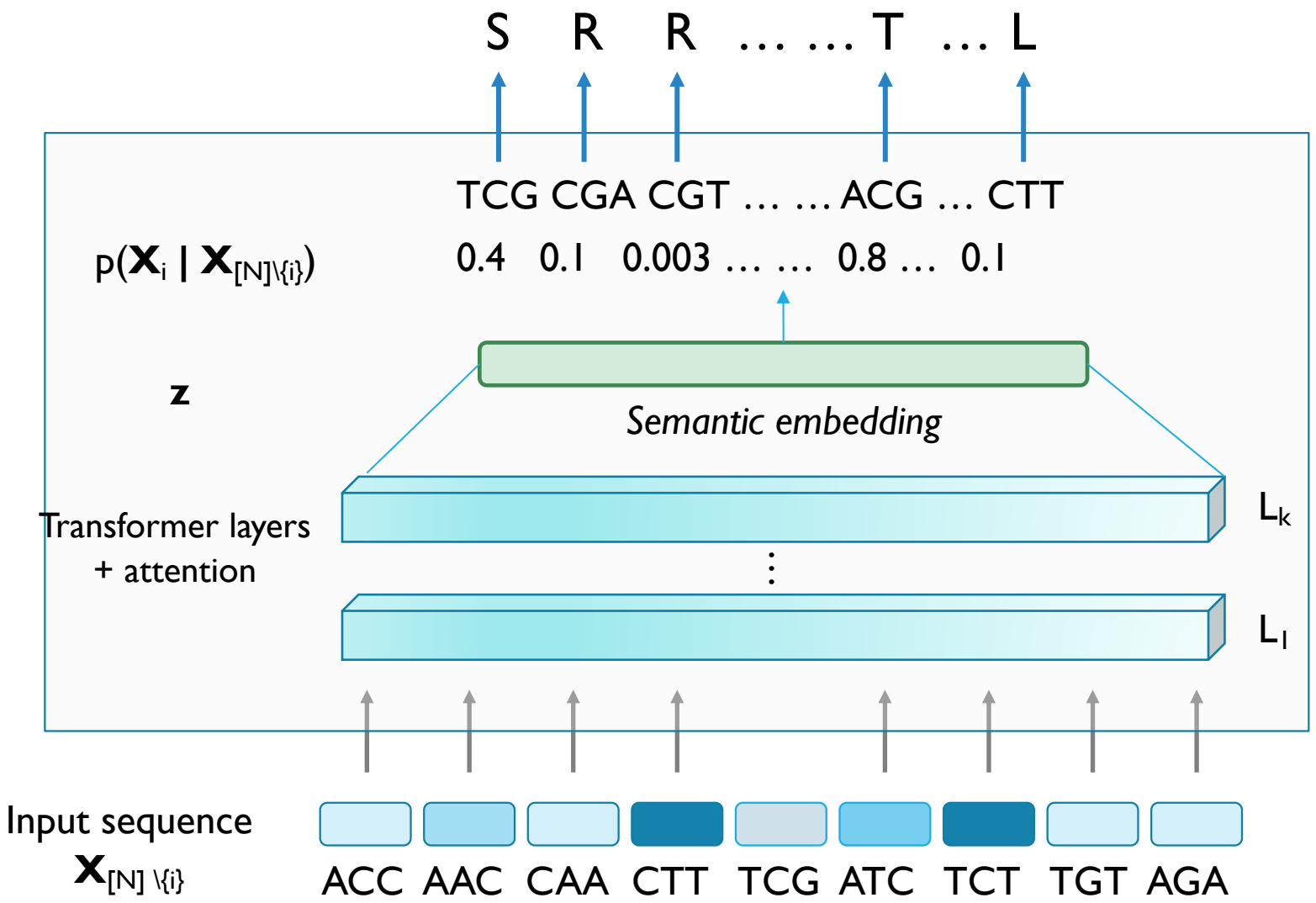


entire genomes

Full genome sequences - BPE Encoding, cannot currently generate full genomes

- Go beyond traditional k-mer models:
  - variable length issues
- At each level of hierarchy maintain information learned at the lower levels (gene → collection/cluster → full genomes)
- Scale at each level but "tie" it together with stable diffusion models

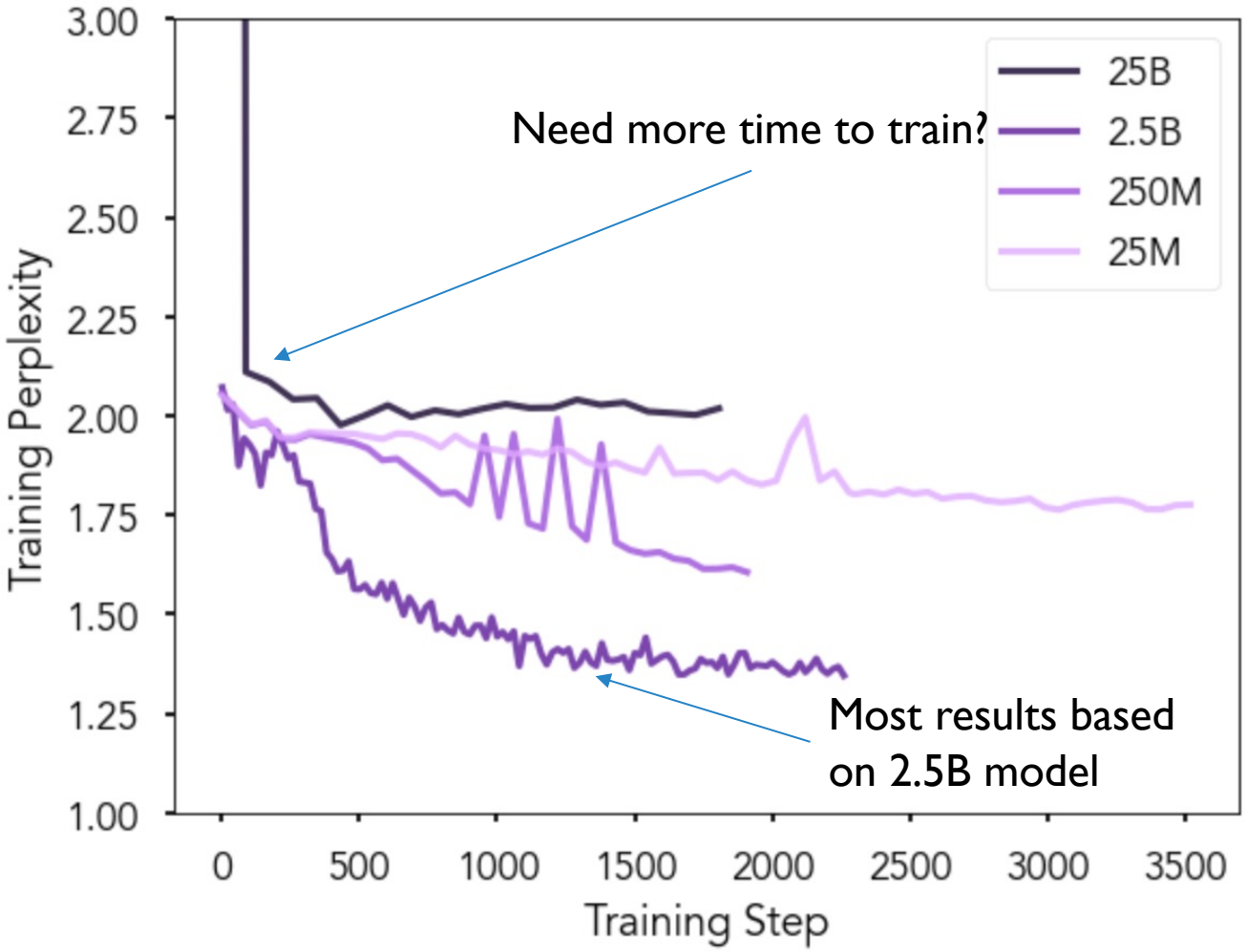
Genome-scale Language Models (GenSLMs)



Model	Seq. length	#Parameters	Dataset
GenSLM-Foundation	2048	25M, 250M, 2.5B, **25B**	110M
GenSLM	10240	25M, 250M, 2.5B, **25B**	1.5M
GenSLM-Diffusion	10240	2.5B	1.5M

- Scaling LLMs with 25B parameters:
  - O (L^2) complexity in the attention computation
  - overcome communication overheads, parameters, checkpointing
- Variation within SARS-CoV-2 sequences can be small (< 1% overall variation)
  - Need foundation model to accommodate diversity
- **One of the largest foundation model trained on raw nucleotide sequences**

GenSLMs achieve state-of-the-art perplexity even with shorter training cycles

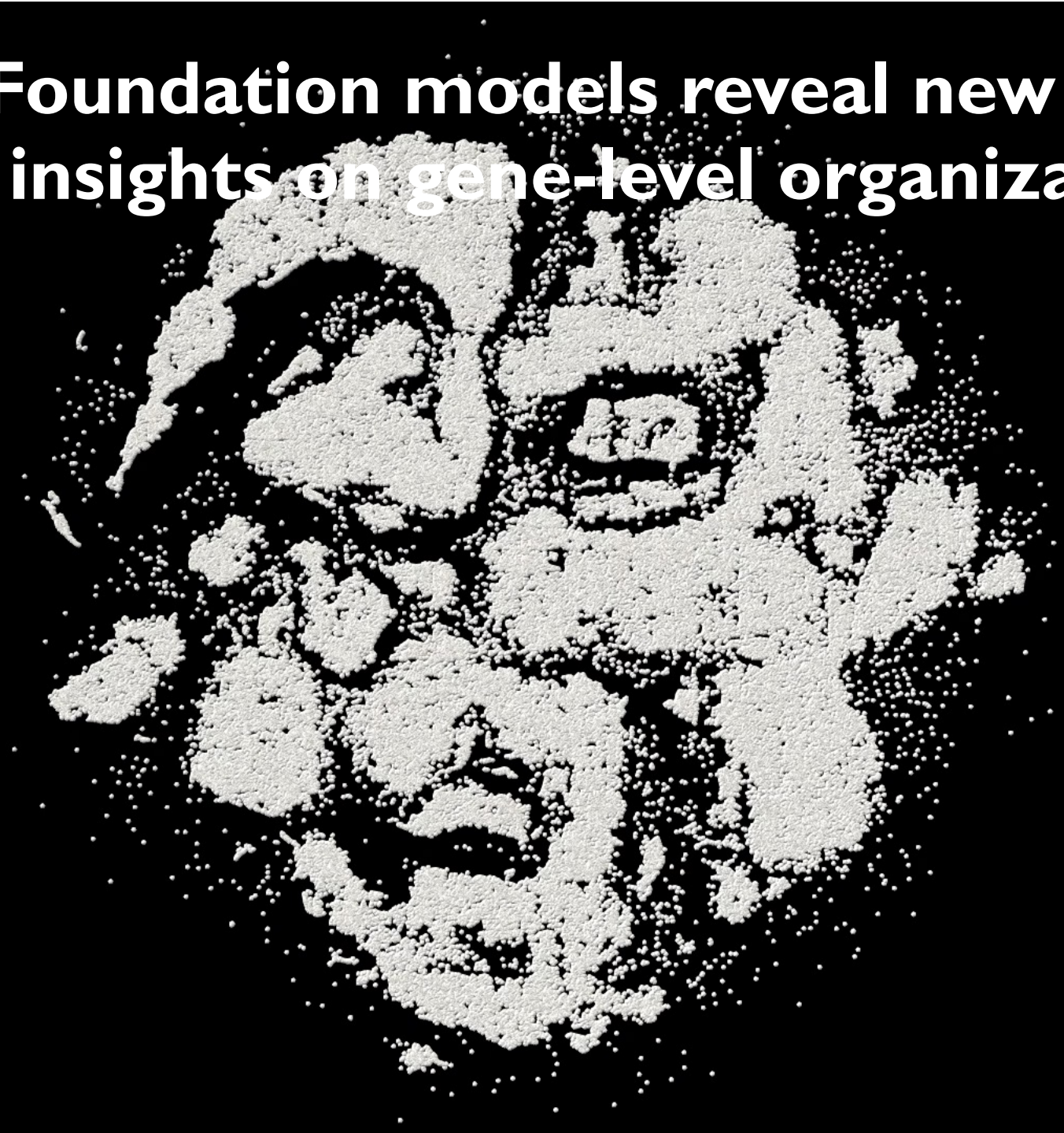


- Perplexity measures the number of guesses required by the LLM to predict the token of interest
  - Perplexity of 1 implies perfect model[1]

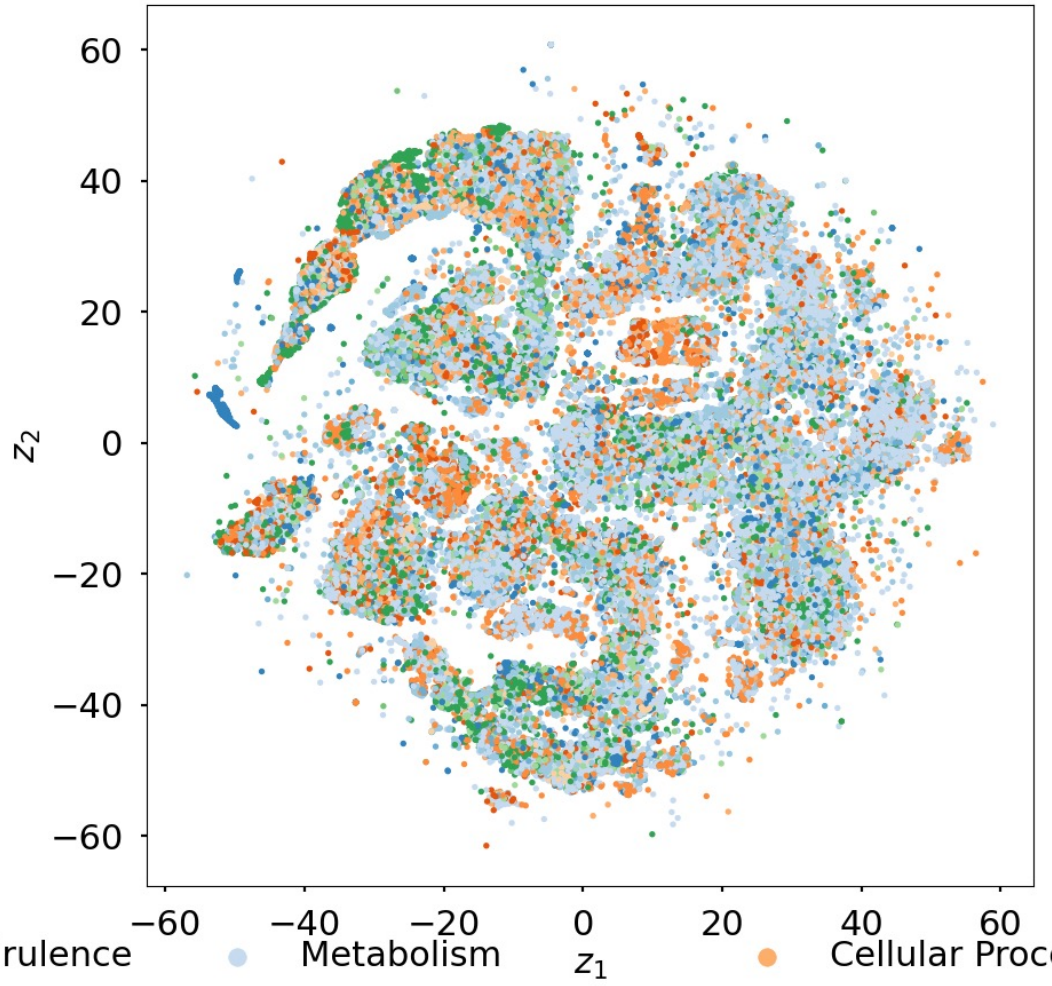
- As trainable parameters increase, model perplexity reduces
  - **Challenge**: 25B model includes model sharding and the training time available on GPUs imposing limitations
  - **Solution**: Cerebras CS-2 wafer-scale cluster enables training

1. Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling laws for neural language models." arXiv preprint arXiv:2001.08361(2020).

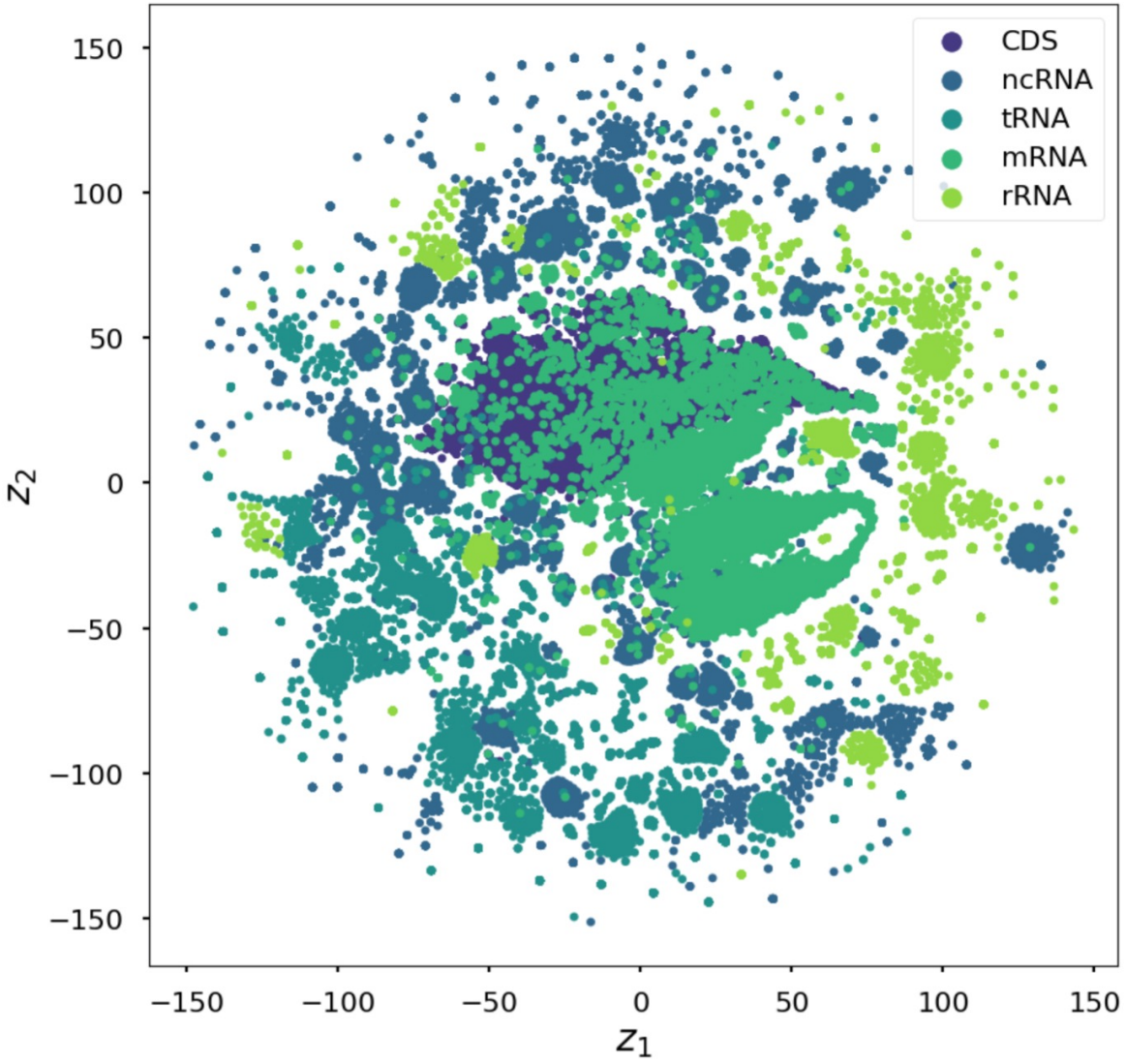
GenSLM Foundation models reveal new biological insights on gene-level organization

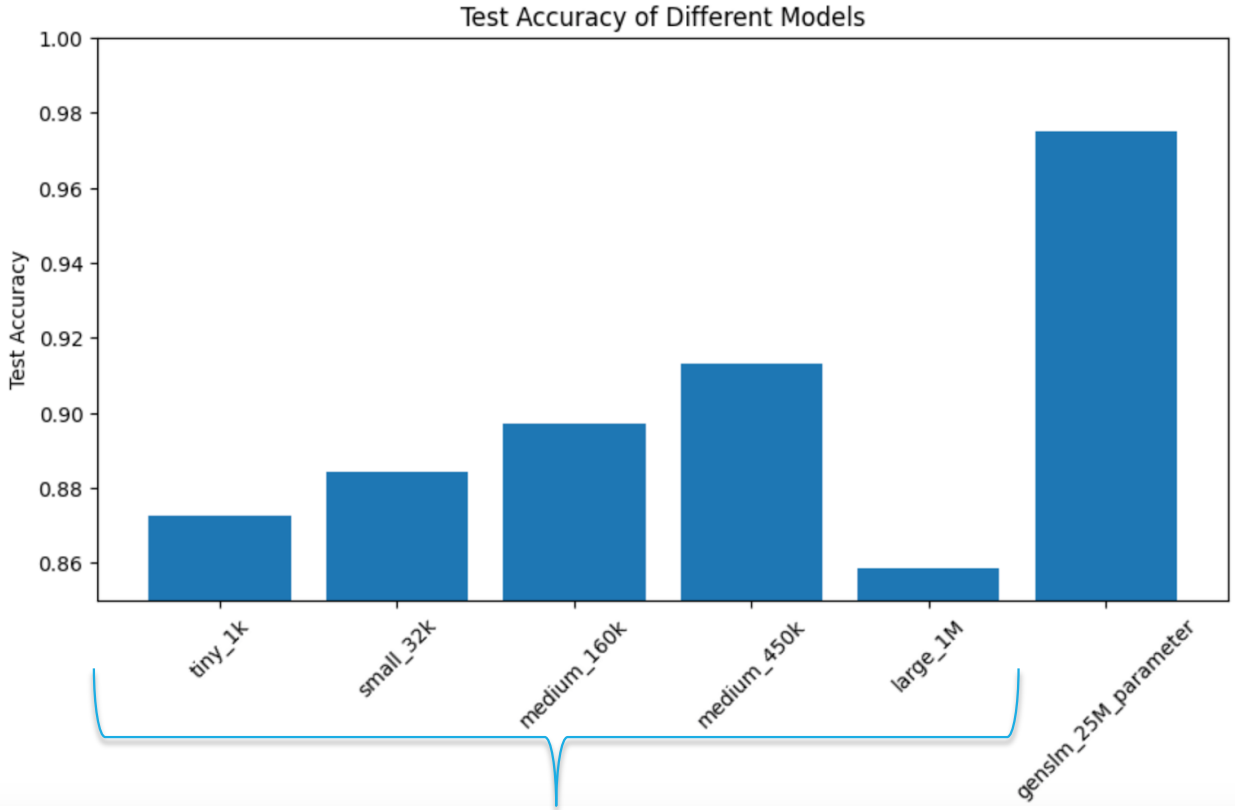


GenSLMs also reveal function level organization of bacterial genes



GenSLMs can also distinguish coding and non-coding sequences …





HyenaDNA (small-large models)

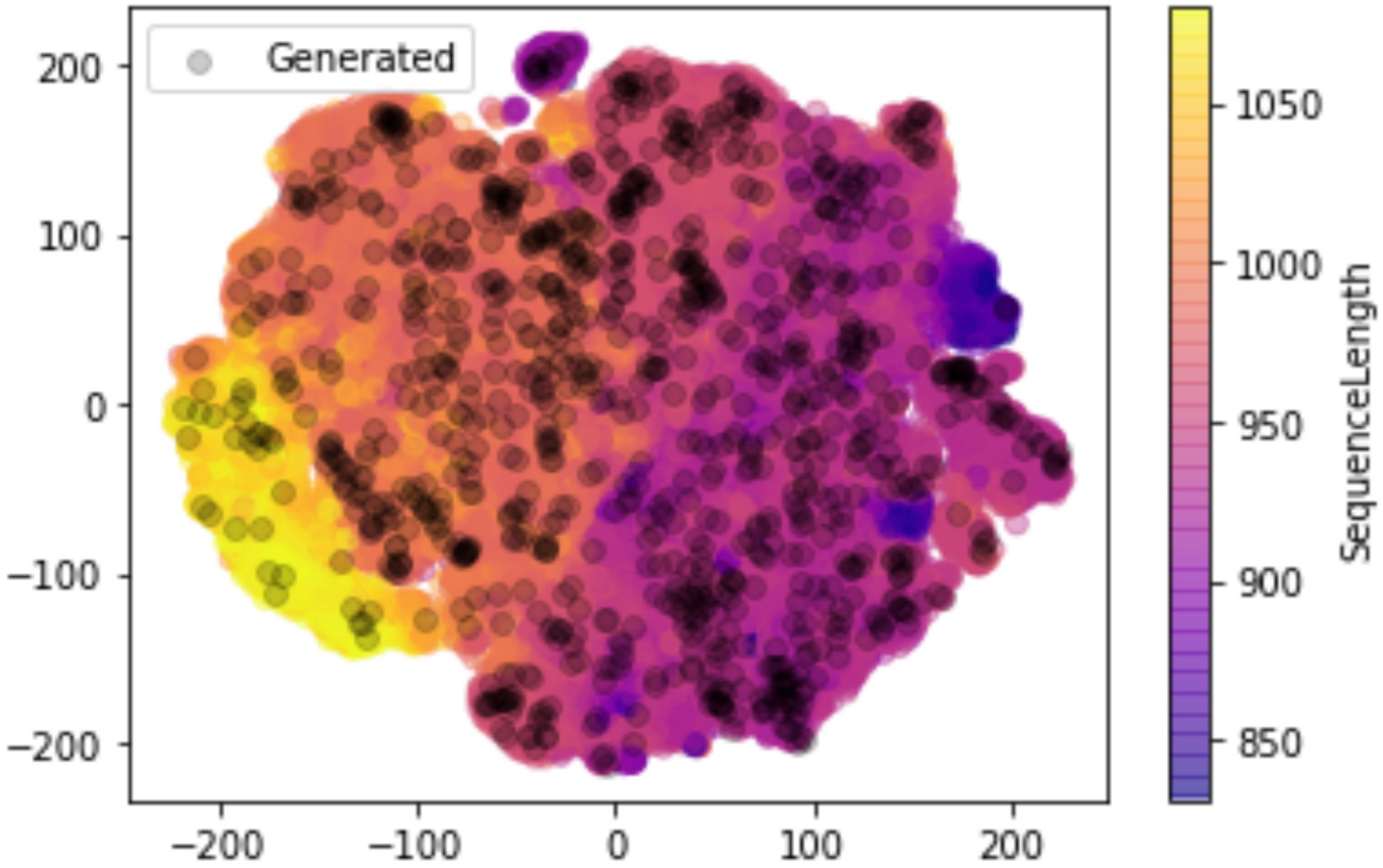
Infrastructure of GenSLM Foundation Models


```python
import torch
import numpy as np
from torch.utils.data import DataLoader
from genslm import GenSLM, SequenceDataset

model = GenSLM("genslm_25M_patric", model_cache_dir="/content/gdrive/MyDrive")
model.eval()

# Input data is a list of gene sequences
sequences = [
    "ATGAAAGTAACCGTTGTTGGAGCAGGTGCAGTTGGTGCAAGTTGCGCAGAATATATTGCA",
    "ATTAAAGATTTCGCATCTGAAGTTGTTTTGTTAGACATTAAAGAAGGTTATGCCGAAGGT",
]

dataset = SequenceDataset(sequences, model.seq_length, model.tokenizer)
dataloader = DataLoader(dataset)

# Compute averaged-embeddings for each input sequence
embeddings = []
with torch.no_grad():
    for batch in dataloader:
        outputs = model(batch["input_ids"], batch["attention_mask"], output_hidden_states=True)
        # outputs.hidden_states shape: (layers, batch_size, sequence_length, hidden_size)
        emb = outputs.hidden_states[0].detach().cpu().numpy()
        # Compute average over sequence length
        emb = np.mean(emb, axis=1)
        embeddings.append(emb)

# Concatenate embeddings into an array of shape (num_sequences, hidden_size)
embeddings = np.concatenate(embeddings)
embeddings.shape
>>> (2, 512)
```

speed

Generative modeling with GenSLMs

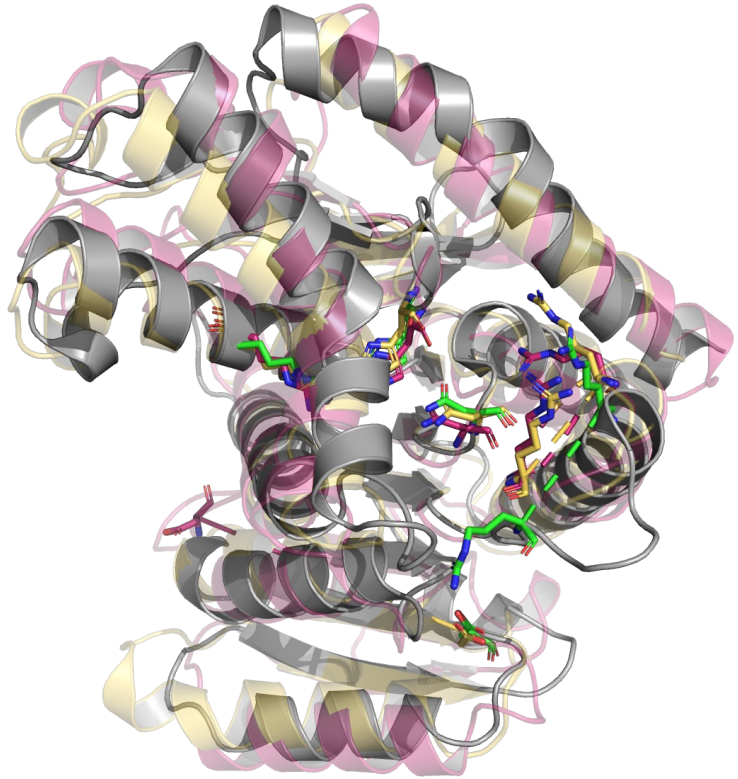
GenSLMs are accurate enough to generate gene sequences…

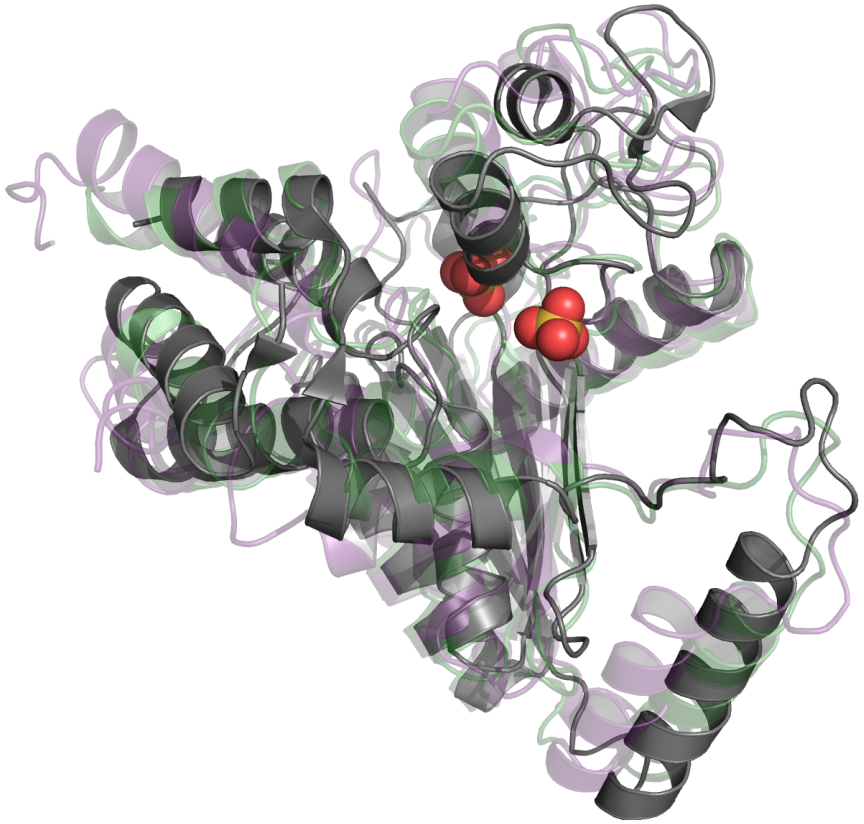


UMAP embeddings of generated sequences agree with learned embeddings using GPT-2

generated sequences 'fill' evolutionary gaps

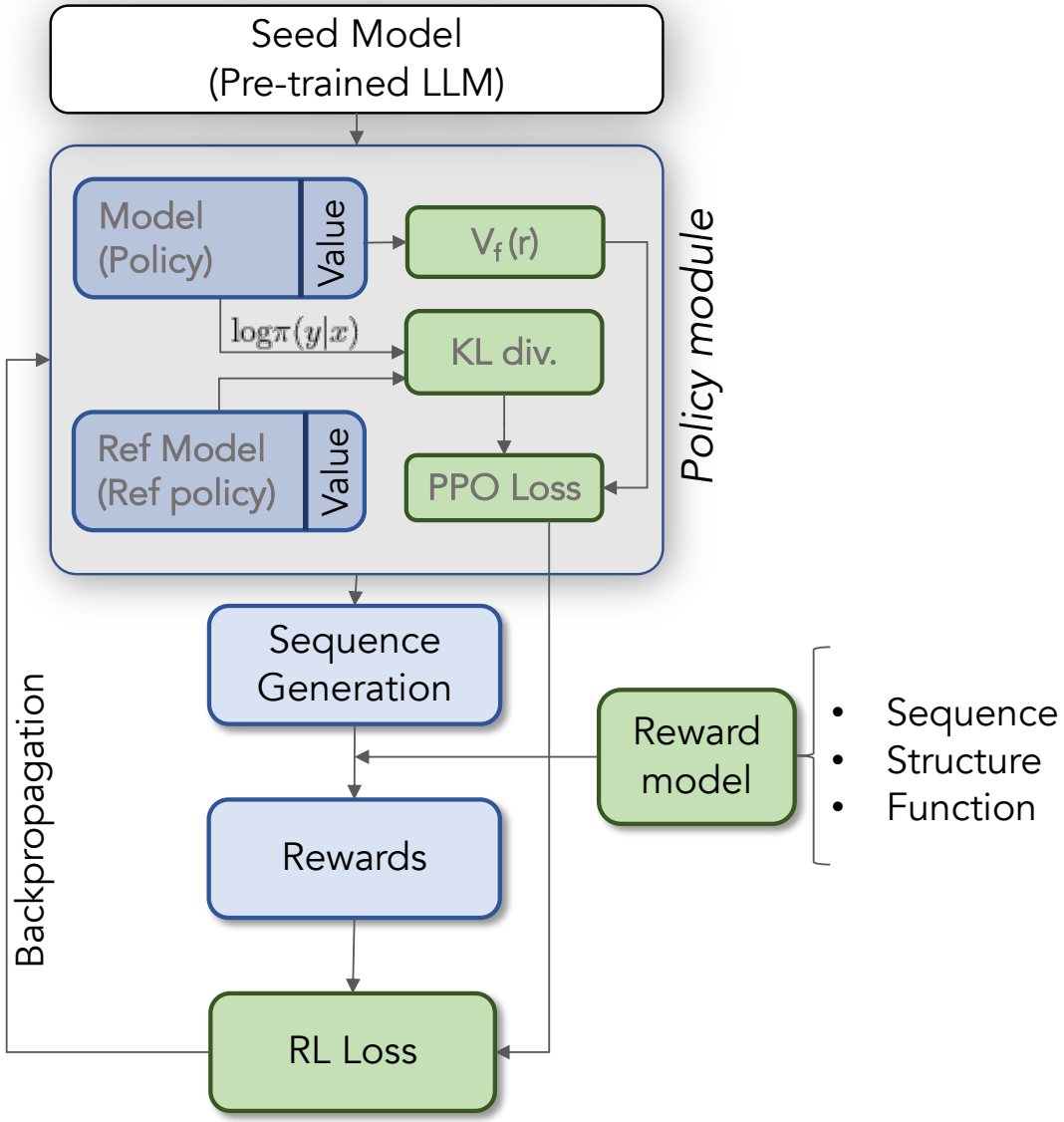
Proteins share MDH similarity at key sites as predicted with OpenFold





GenSLMs learn the two distinct isoforms for MDH and within each isoform we find conservation of key residues and placement of binding sites

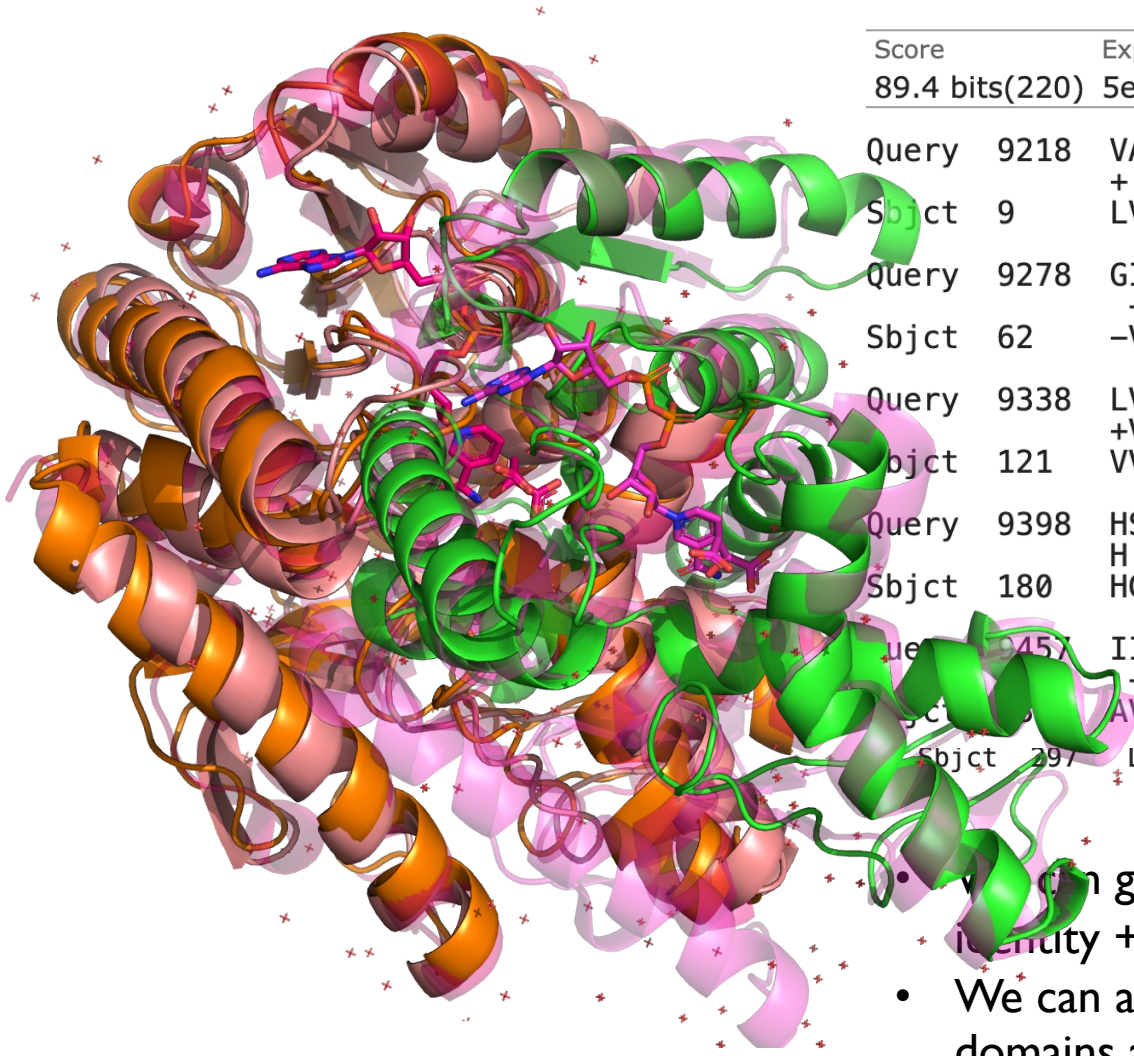
Designing enzymes by incorporating experimental feedback (aka ChatGPT for protein design)



- Need general framework that enables generative design of proteins by incorporating experimental feedback
- Genome-scale language models (GenSLMs)[1] provide a means to incorporate generative modeling for gene sequences:
  - complementary to protein language models
- Rewards for the model:
  - intrinsic – sequence specific (e.g., GC content for environmental adaptation)
  - extrinsic – functional annotation/ enzyme activity measured via experimentation

[1] M. Zvyagin, et al, Genome-scale language models map the evolutionary trajectories of SARS-CoV-2 (SC'22 Gordon Bell Prize)

Multi-objective RL for generative design allows greater sequence diversity across MDH sequences



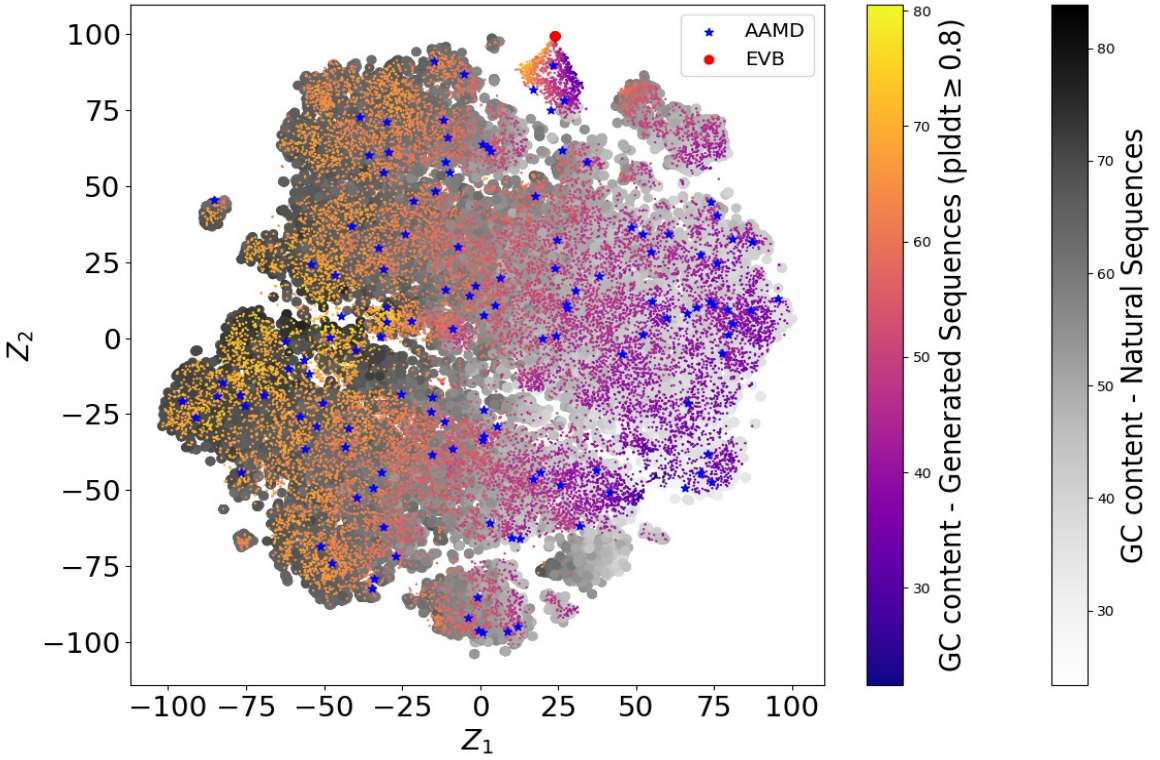
Range 17: 9 to 244 GenPept Graphics ▼ Next Match ▲ Previous Match ▲ First Match

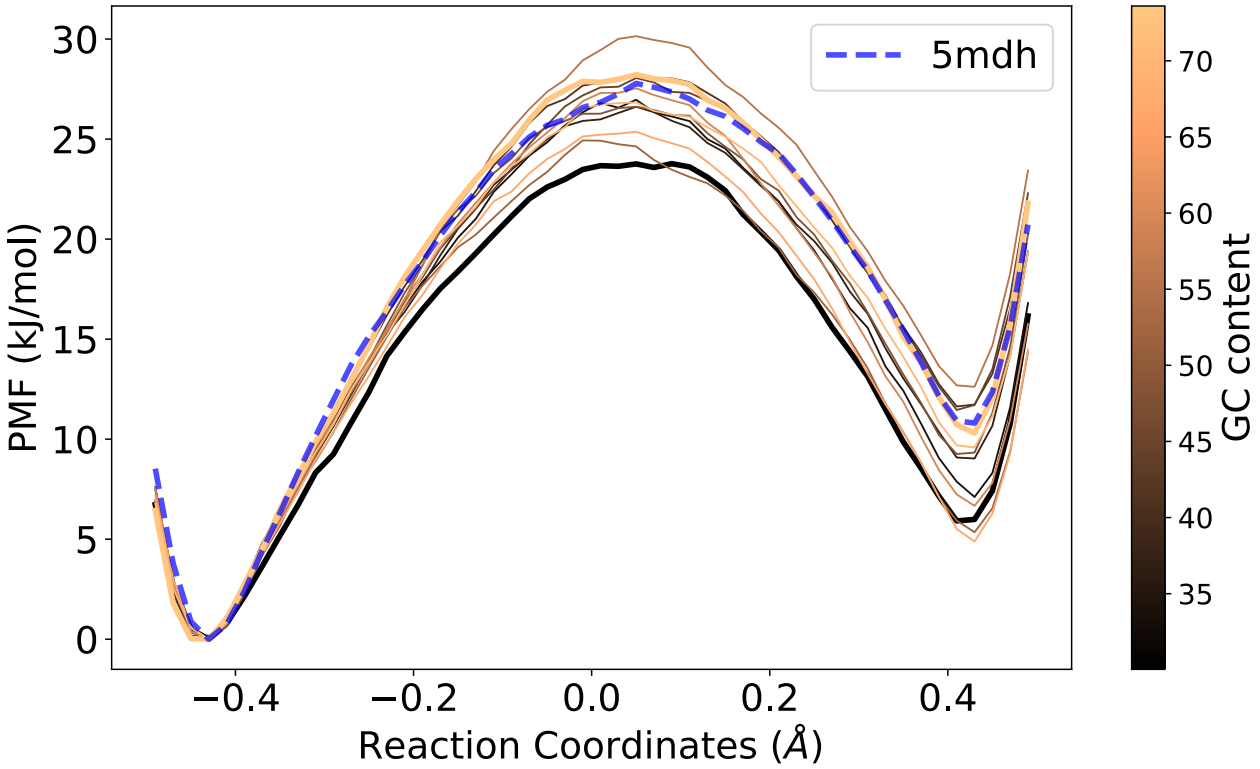
Score	Expect	Method	Identities	Positives	Gaps
89.4 bits(220)	5e-14	Compositional matrix adjust.	72/247(29%)	115/247(46%)	12/247(4%)

```
Query  9218  VAVTGAAGQIGYSLLFRIASGSMFGPDQPVVLHLIEIEPALPALQGVVMELEDCAFPLLK  9277
             + + GA G IG ++   + S  M G    + L+ I      P + GV  E++ CAFP +
Sbjct  9     LVIVGAGGMIGSNM---VQSALMLGLTPNICLYDI----FEPGVHGVFDEIQQCAFPGVN  61

Query  9278  GIVPTASLEEGFKGVNWALLVGSVPRKAGMERKELLGINGKIFVGQGKAIAANAAKDVRI  9337
              +  T + EE F G  + +  G  PRK GM R++LL  N KI    G  I     +   +
Sbjct  62    -VTYTVNPEEAFTGAKYIISSGGAPRKEGMTREDLLKGNCKIAAEFGDNIKKYCPEVEHV  120

Query  9338  LVVGNPCNTNCLIAMNNAADVPRDRWFAMTRLDENRAKAQLAKKAGVDVTTVTNMTIWGN  9397
             +V+ NP +   L A+ ++   P ++  LD  R +  LA + GV    VT    +G
Sbjct  121   VVIFNPADVTALTALIHSGLKP-NQLTSLAALDSTRLQQALALEFGVQQDKVTGAHTYGG  179

Query  9398  HSATQYPDFYNAHINGRPANEV-IHDEAWLKGDFITTVQQRGAAIIKARGLSSAASAANA  9456
             H            ++G+P  E+ + DE W +    T   Q G+ IIK RG SS  S A
Sbjct  180   HGEQMAVFASQVKVDGKPLAEMGLSDERWEEIKHHTV--QGGSNIIKLRGRSSFQSPAYN  237

Que  9457  IIDTVKS  9463
           +  +++
Sbjct  238   AVKMIEA  244

Sbjct  297   LAKSYEHLCKMRDEIVELGIVPPVAEWKEMNPNL  330
```

- We can generate new sequences with varying degrees of sequence identity + positive matches
- We can also generate minimal sequences that have functional domains and can function as a productive enzyme

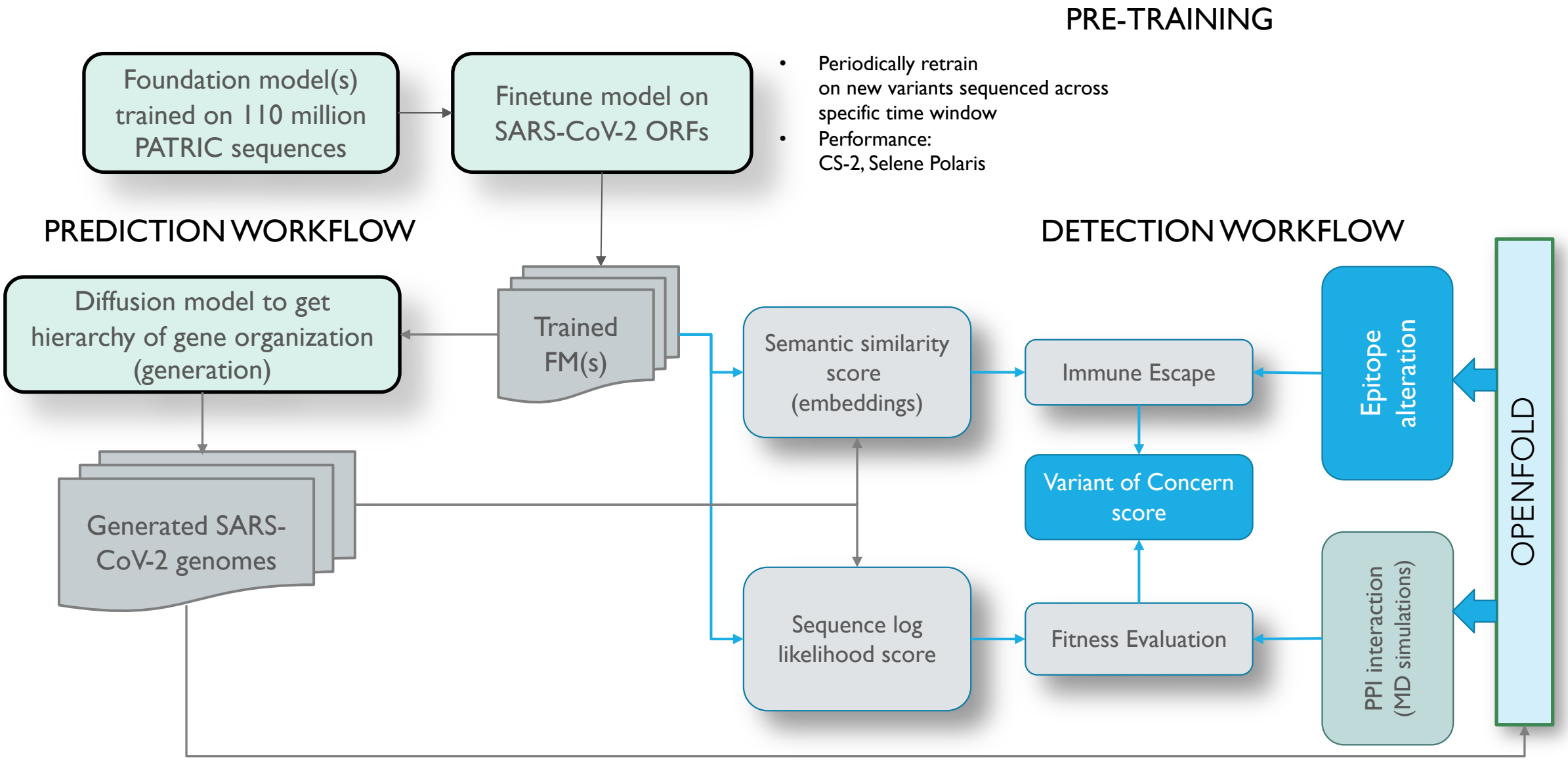
Generative models can sample novel sequences with better activation energy for MDH



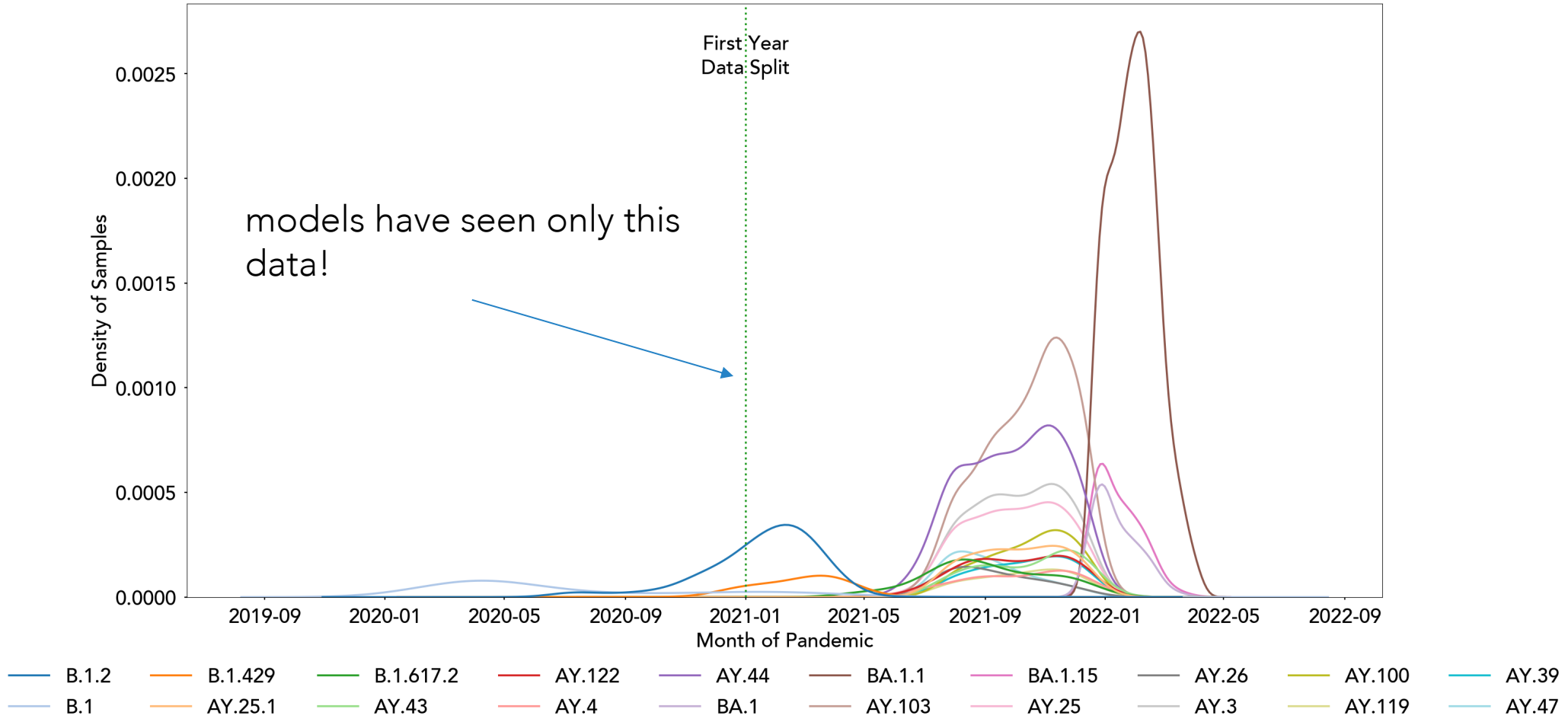


- Exploring even top 1% (1,000 variants x 20 simulation windows = 2,000 simulations) from the embedding space using simulations can overcome the limits on nodes (for a single iteration of RL-based finetuning)
- Labeling productive designs and ranking → large compute requirements across multiple computing sites/ facilities

Using foundation models to predict SARS-CoV-2 evolution



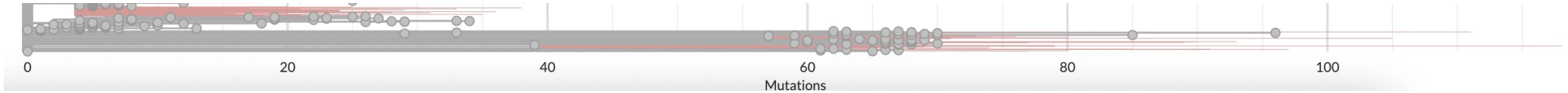
GenSLMs finetuned on SARS-CoV-2 genomes can distinguish variants



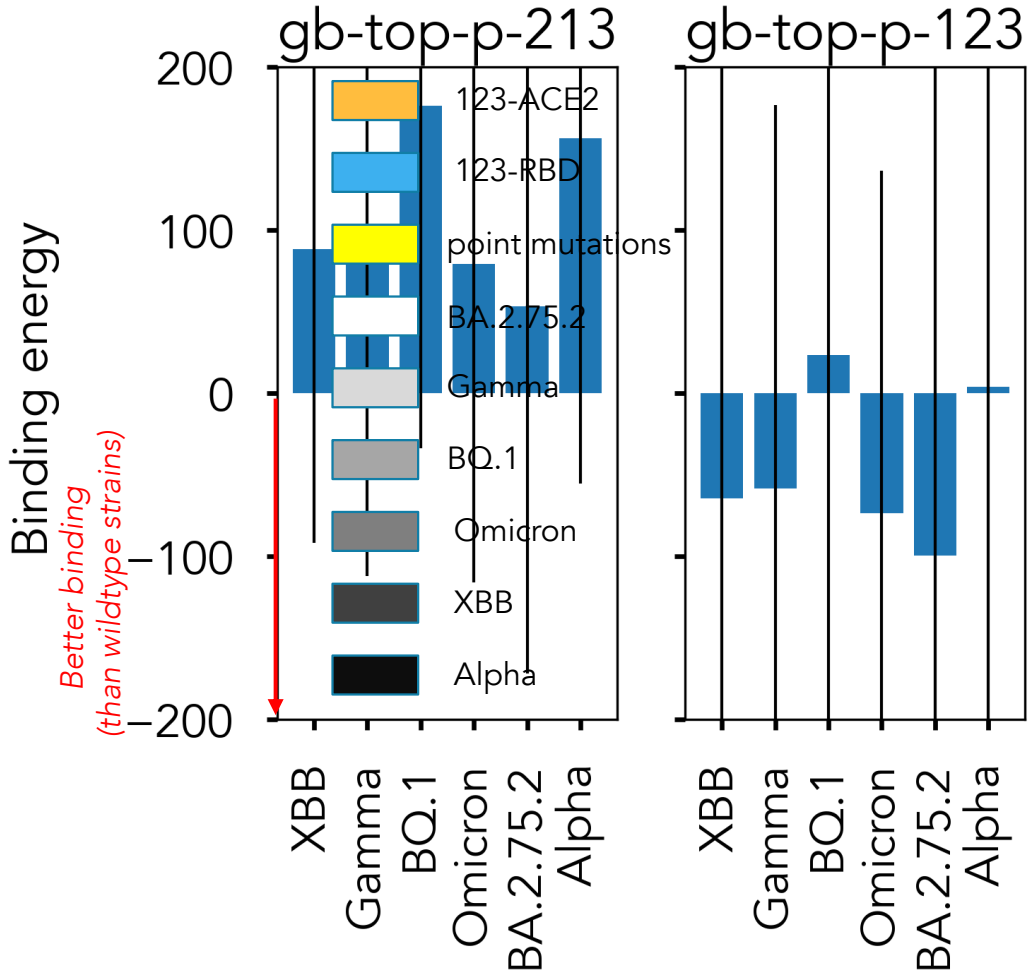
GenSLMs finetuned on SARS-CoV-2 genomes can distinguish variants

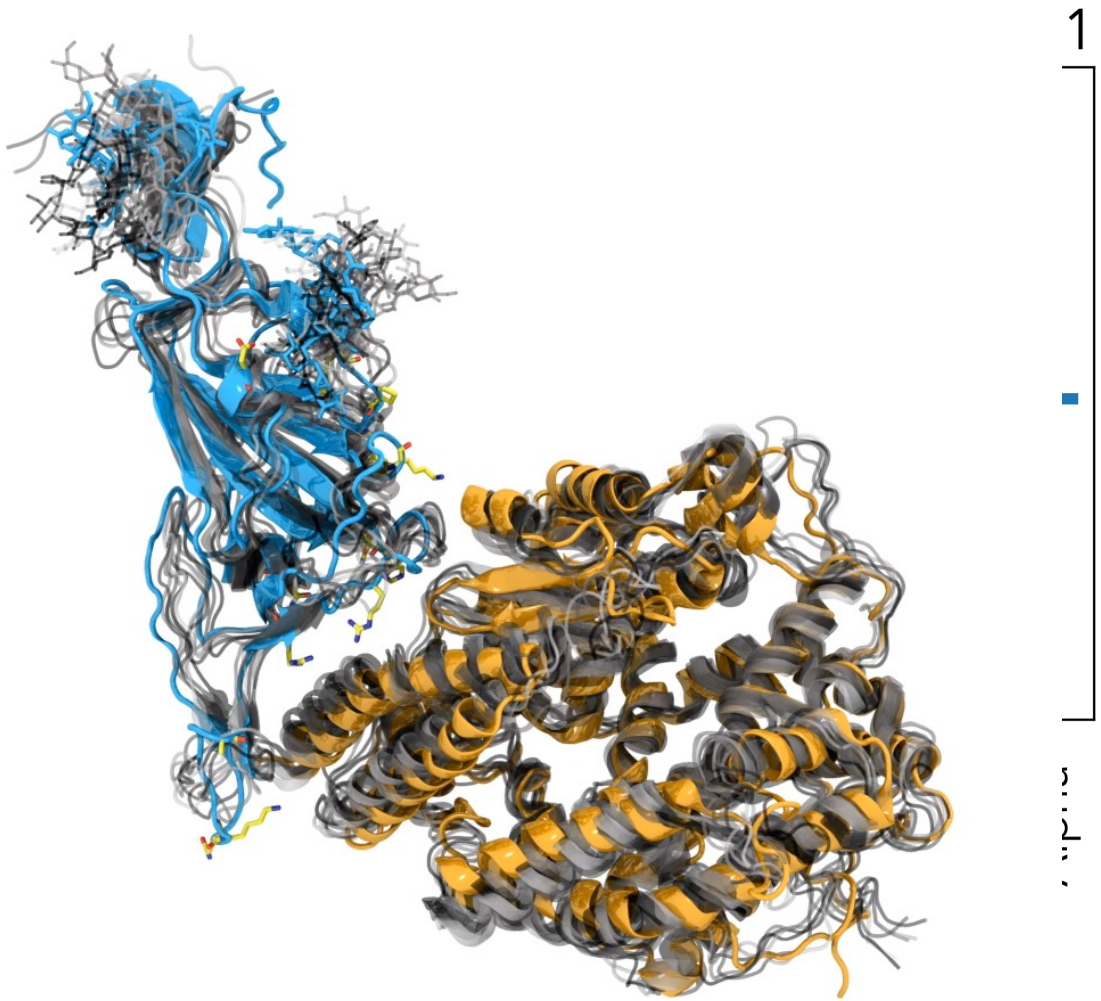
We can generate synthetic sequences that look like SARS-CoV-2

	Genome_ID	Of_Interest	Predicted_Variant	Distance_to_Reference	Neighbors_win_X	K_Neighbors_Variant_Dictionary
212	top-p0-9-0212	True	B.1	19.0	538	{'B.1': 16, 'B.1.206': 3, 'B.1.596': 1}
295	top-p0-9-0295	True	B.1	28.0	274	{'B.1': 16, 'B.1.206': 3, 'B.1.596': 1}
313	top-p0-9-0313	True	B.1	19.0	538	{'B.1': 16, 'B.1.206': 3, 'B.1.596': 1}
349	top-p0-9-0349	True	omicron	76.0	298	{'omicron': 20}
398	top-p0-9-0398	True	B.1	28.0	274	{'B.1': 16, 'B.1.206': 3, 'B.1.596': 1}
416	top-p0-9-0416	True	B.1.1.7	56.0	67	{'B.1.1.7': 17, 'B.1.1': 2, 'None': 1}
438	top-p0-9-0438	True	B.1.1.7	49.0	71	{'B.1.1.7': 13, 'B.1.1': 5, 'None': 2}
540	top-p0-9-0540	True	omicron	76.0	298	{'omicron': 20}
544	top-p0-9-0544	True	B.1.1.7	56.0	67	{'B.1.1.7': 17, 'B.1.1': 2, 'None': 1}
715	top-p0-9-0715	True	B.1.1.7	49.0	71	{'B.1.1.7': 13, 'B.1.1': 5, 'None': 2}
807	top-p0-9-0807	True	B.1	10.0	650	{'B.1': 16, 'B.1.206': 3, 'B.1.596': 1}

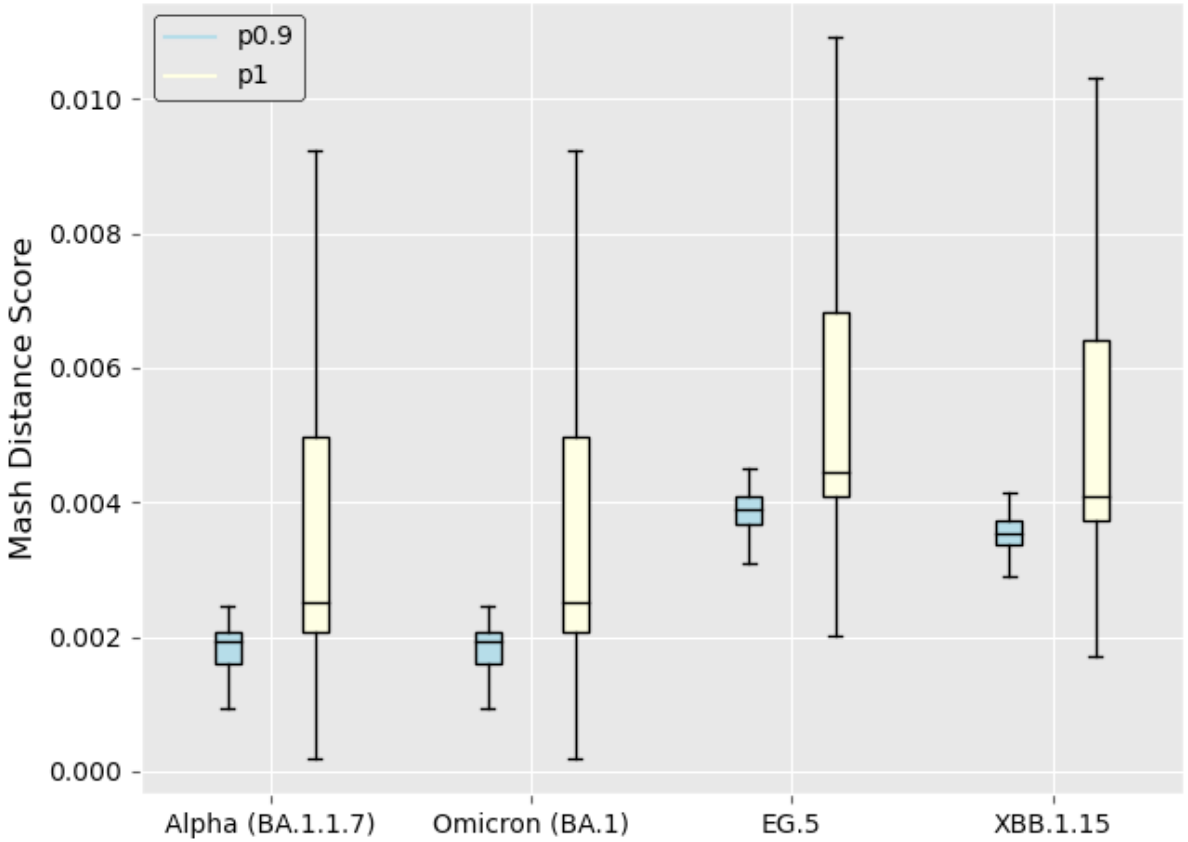


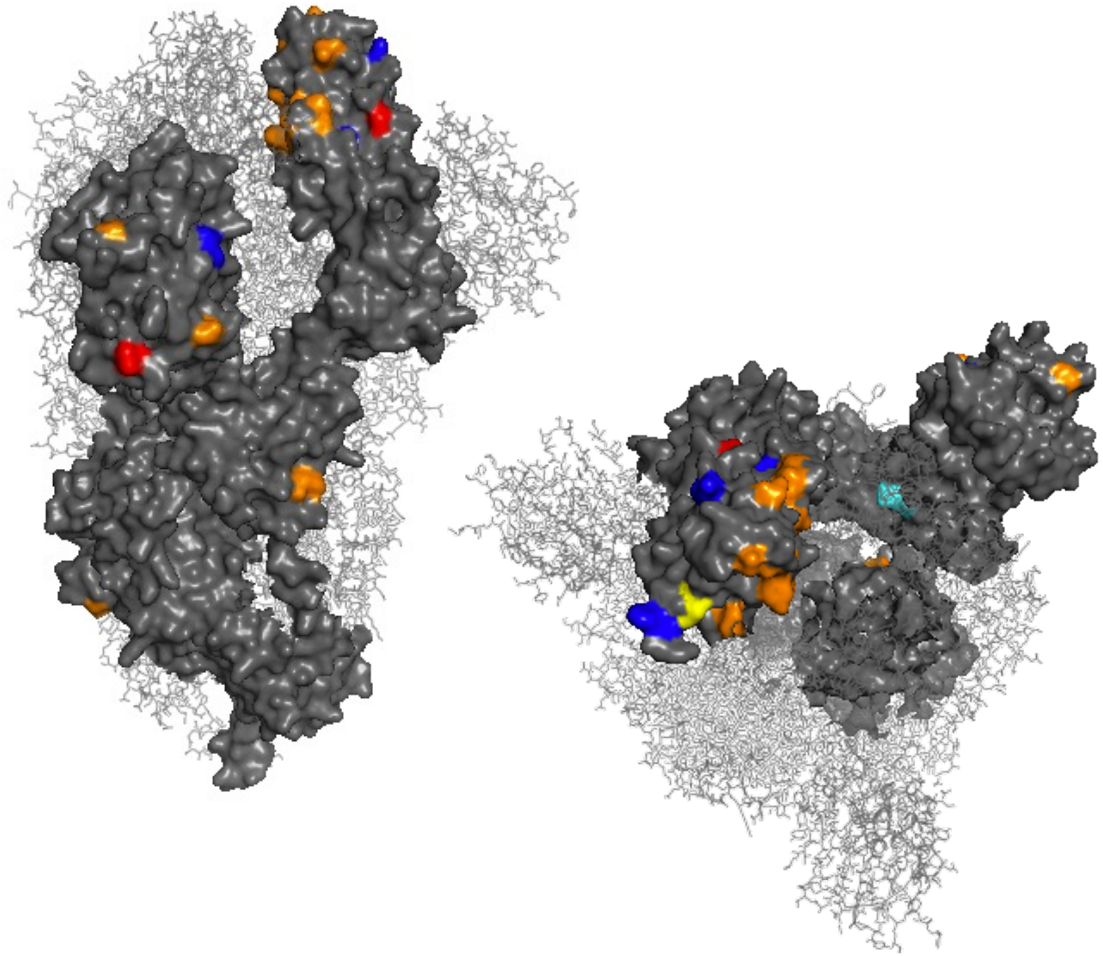
A generated variant is evolutionarily close to BQ.1!





Our models also present characteristics of EG.5 …

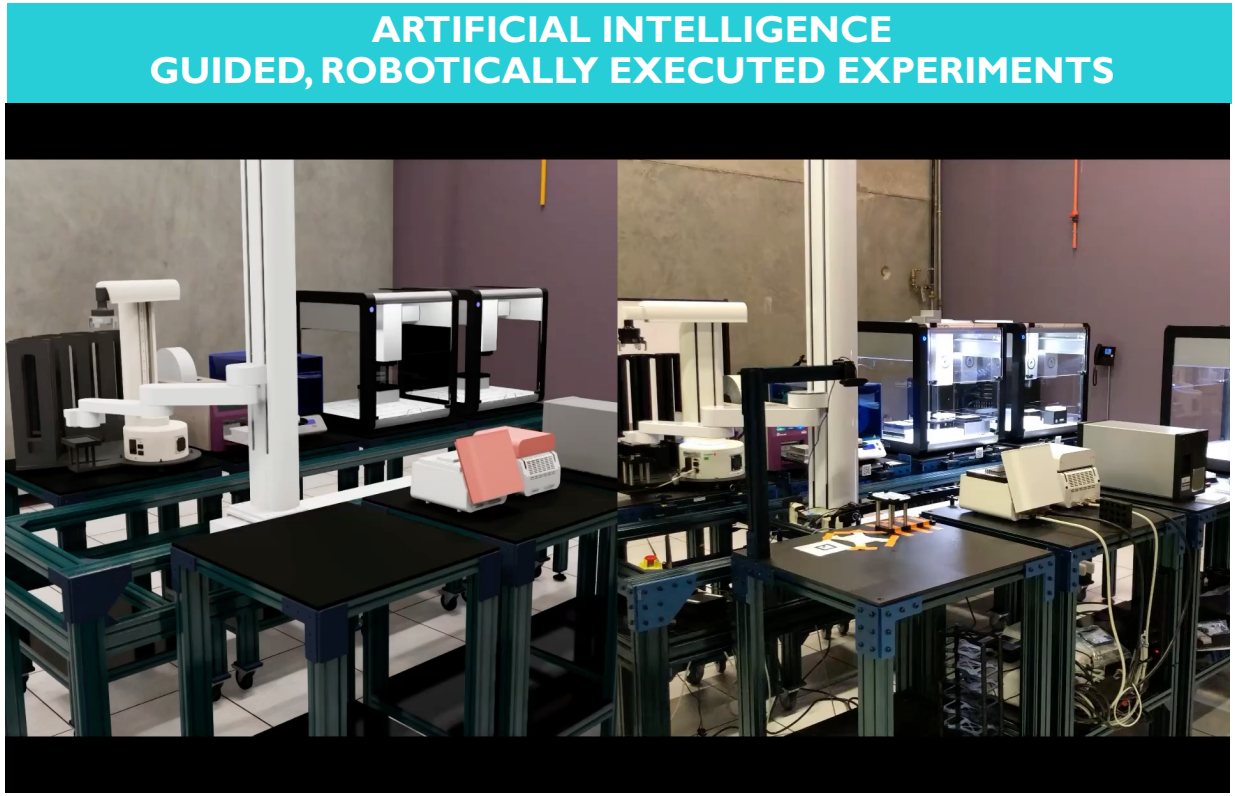




Building embodied agents as scientific assistants…

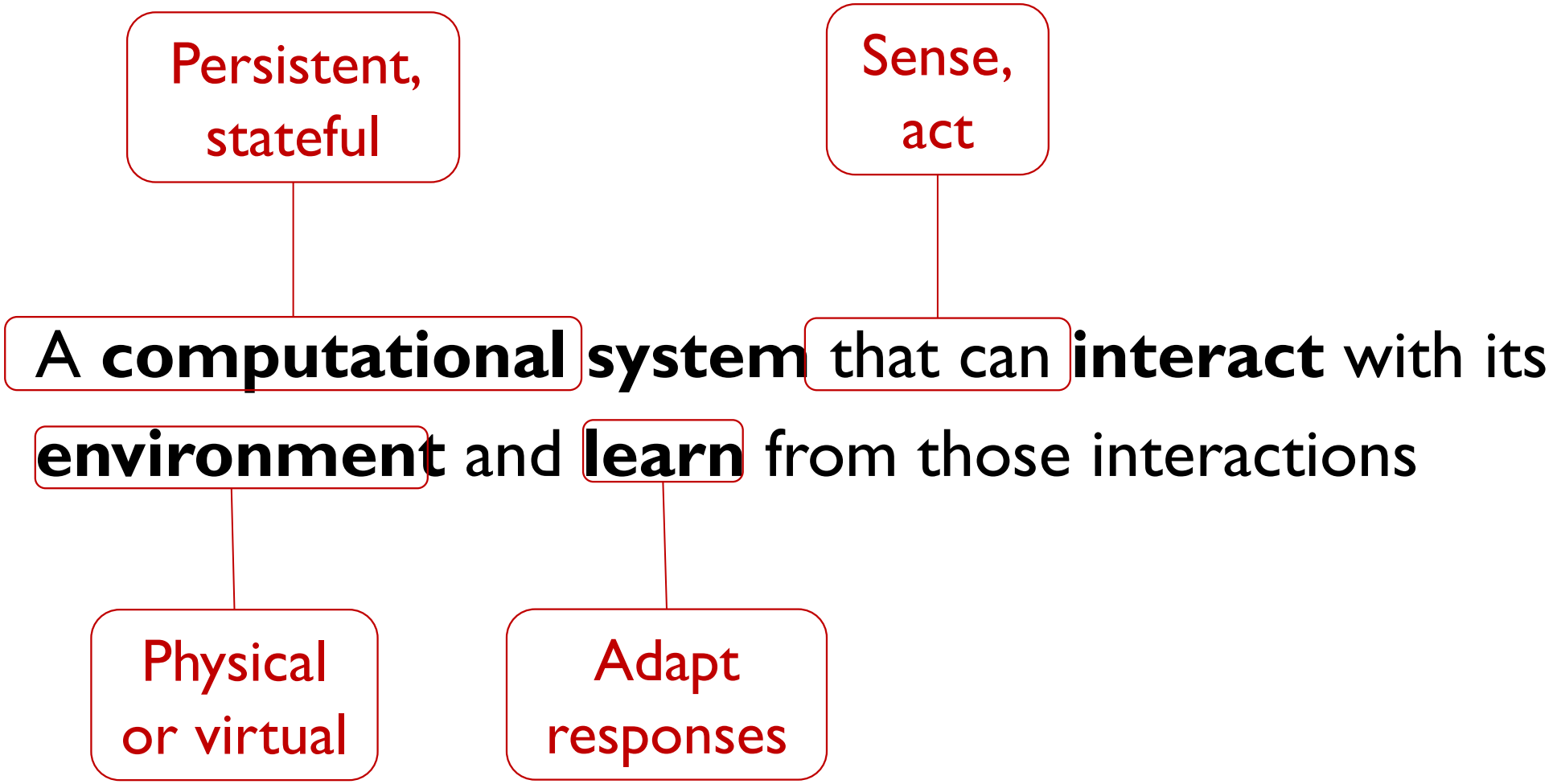
Autonomous Discovery @Argonne

- **The vision**
  - A system that starts with a high-level description of a hypothesis and autonomously carries out computational and experimental workflows to confirm or reject that hypothesis
  - **Use of AI in robotics and simulations to close the loop on planning, execution, and analysis of experiments**
- Builds on
  - **AI approaches to planning** (multiple steps), and integration of results, causality, etc.
  - **Machine learning/simulation** to design and predict exp properties and outcomes
  - **Automation of experimental protocols** (robotic steps and workflows)
  - **Active Learning or RL** for selection of next experimental targets, etc.



https://github.com/anl-sdl/
https://www.cs.uchicago.edu/~rorymb/

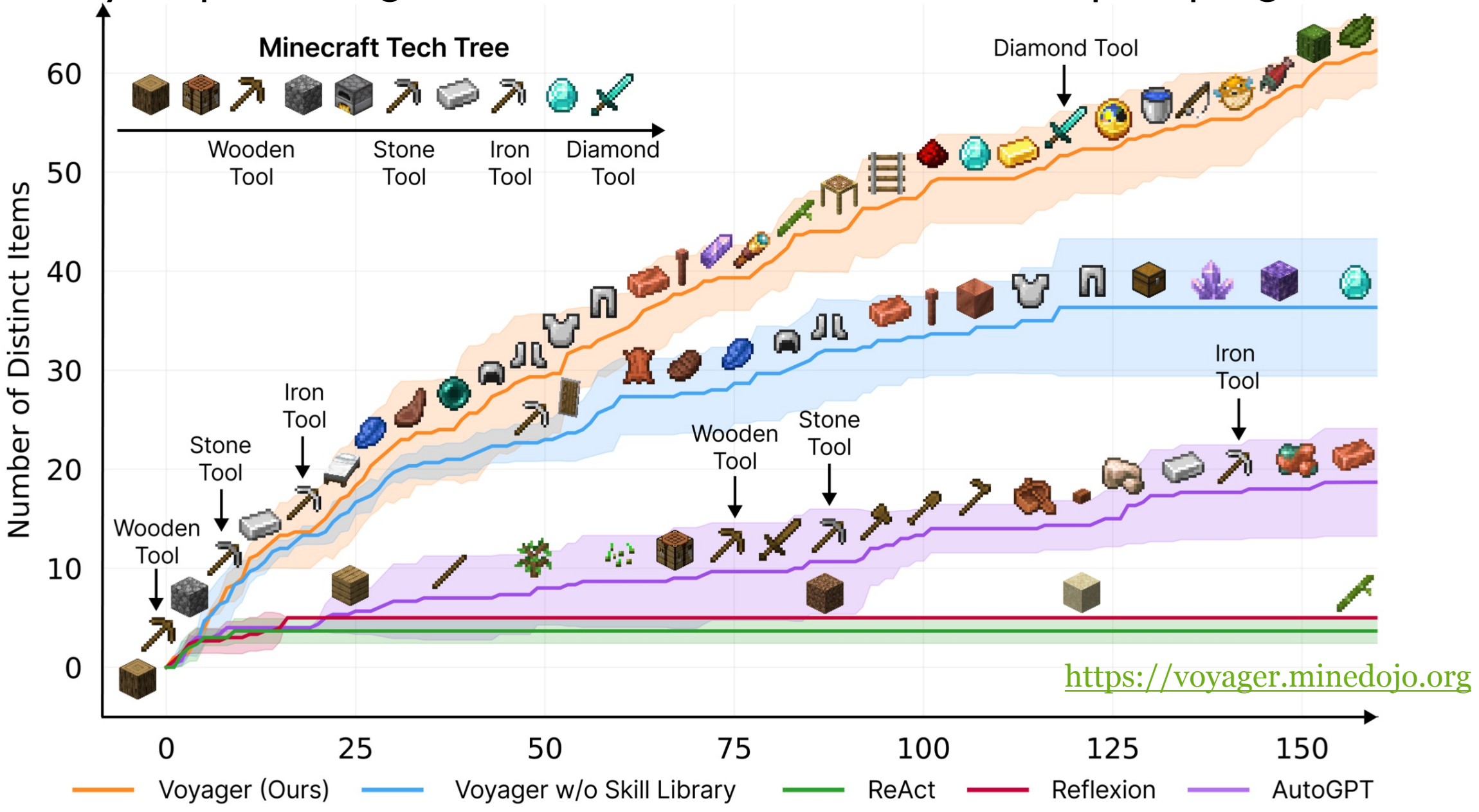
Embodied agent



Courtesy: Ian Foster, ANL/ UChicago


https://voyager.minedojo.or

VOYAGER discovers new Minecraft items and skills continually by self-driven exploration, significantly outperforming SOTA. X-axis denotes the number of prompting iterations.



https://voyager.minedojo.org

Generated by GPT-4, based on goal "discover as many diverse things as possible"

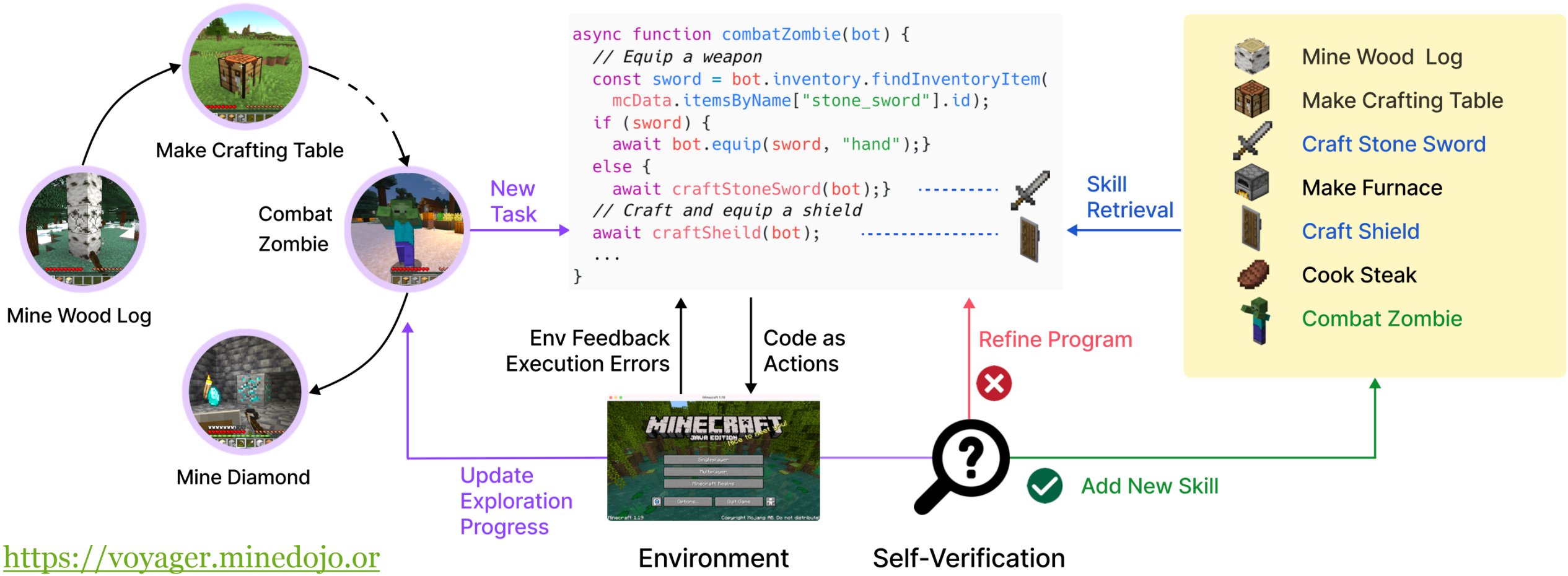
Generate executable code for embodied control

Store and retrieve complex behaviors

Automatic Curriculum

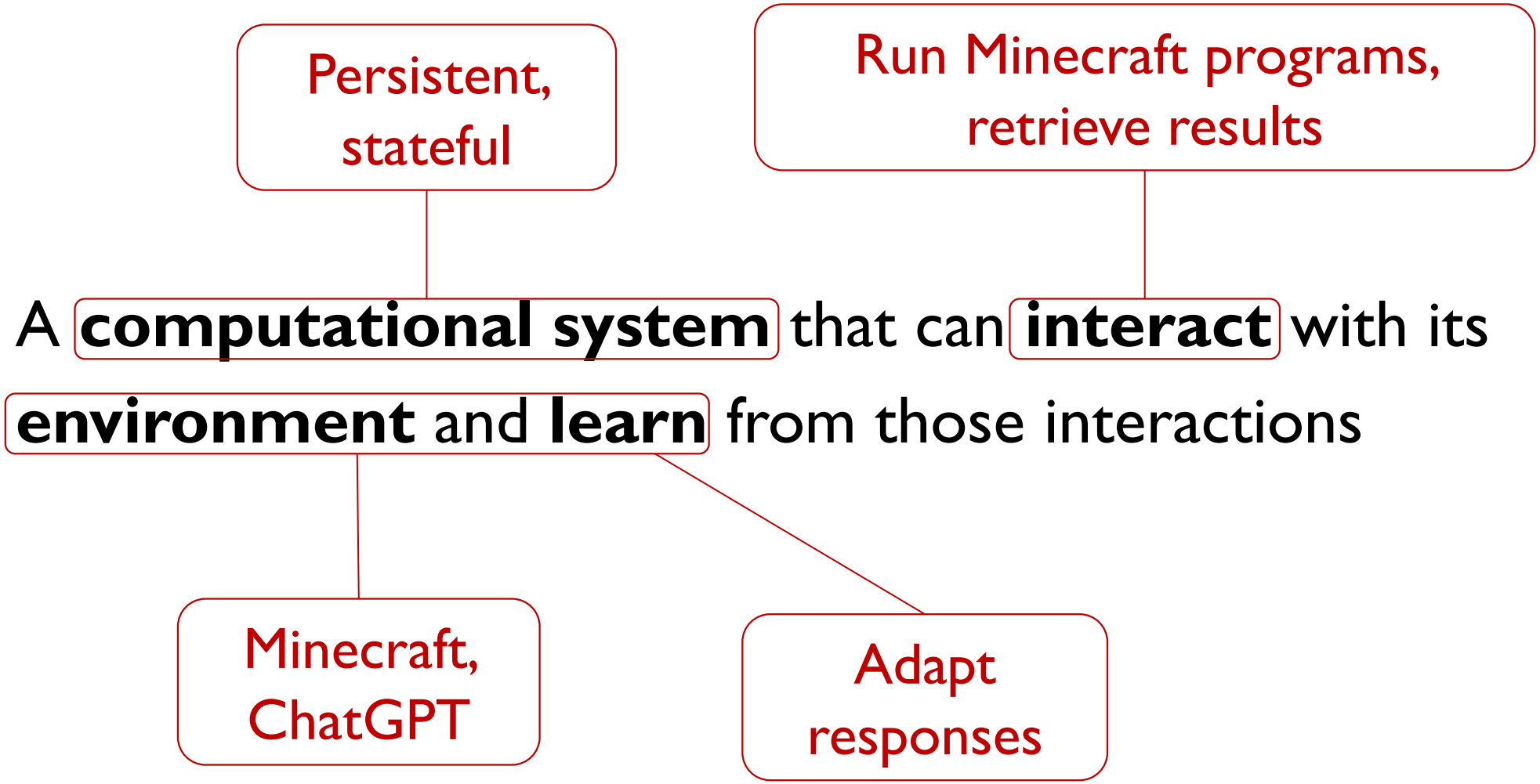
Iterative Prompting Mechanism

Skill Library



https://voyager.minedojo.or

Embodied agent: Voyager – that generates new Minecraft games/ programs



Courtesy: Ian Foster, ANL/ UChicago

A scientific assistant

- Configure and run computational simulations
- Configure and run physical experiments
- Collect, organize, curate data
- Search the literature for data, protocols, etc.
- Formulate hypotheses
- Define protocols to test hypotheses
- Diagnose problems with experiments and simulations
- …

- Many skills, often requiring specialized knowledge
- **Ability to interact with many resources in many places**

Courtesy: Ian Foster, ANL/ UChicago

Building embodied scientific agents is fraught with challenges

1) **Act on resources** regardless of location and interface

→ Widely deployed **local agents**

provide a global footprint for actions

Friction: Varying interfaces, behaviors; reliability; security

2) Execute remote actions **reliably**

→ Cloud-hosted **managed research acceleration**

services buffer against inevitable failures

Friction: Failures, scalability, usability

3) Manage who is **trusted** to perform what actions, where and when

→ **Distributed authentication with delegation**

enables secure management of privileges

Friction: Varying credentials, authentication protocols, authorization policies; need to act on behalf of others

Courtesy: Ian Foster, ANL/ UChicago

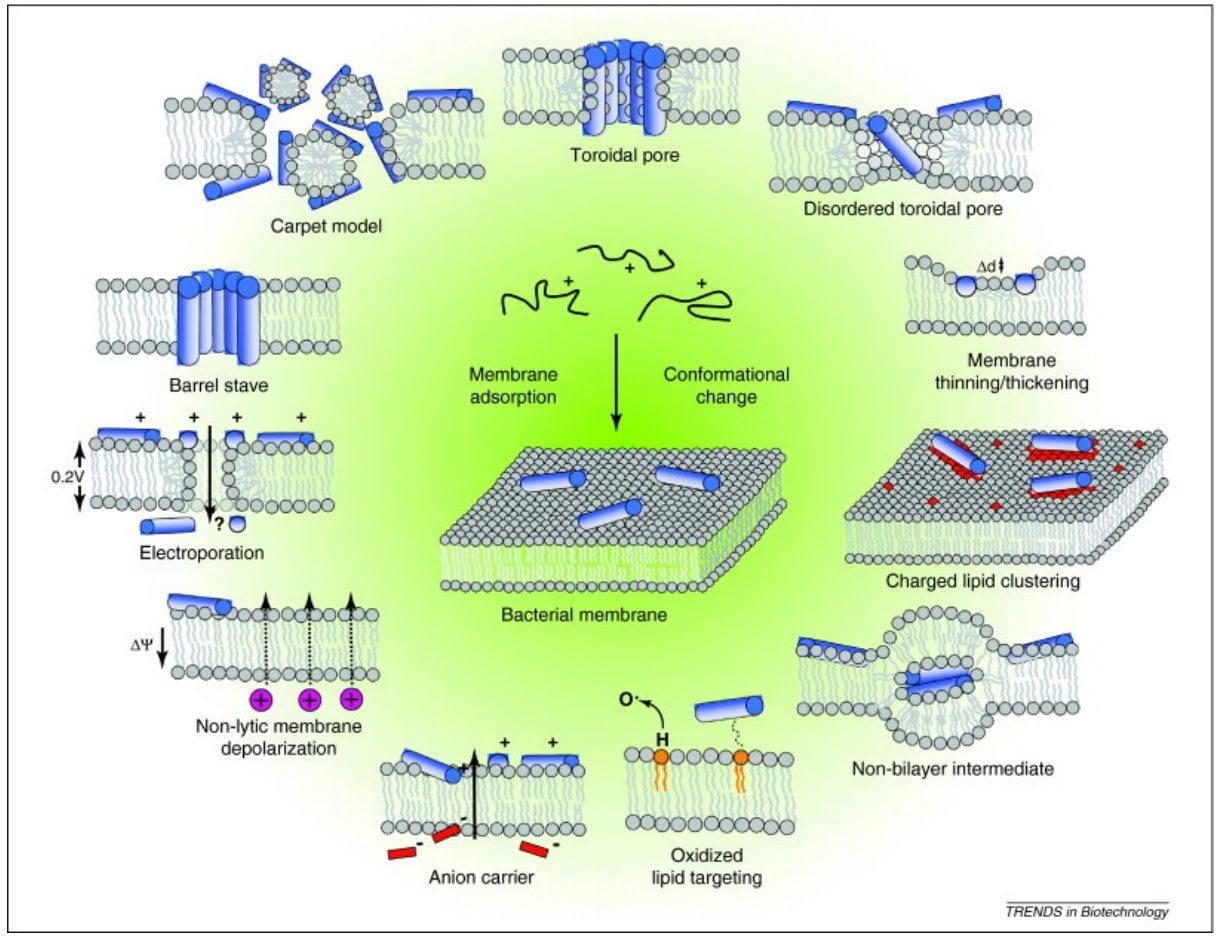
Design of antimicrobial peptides

An antimicrobial peptide (AMP) is a short (typically 12 to 50 amino acid) molecule that can target and kill viruses, bacteria, fungi, and other pathogens

Challenge: Design an AMP that can kill specified bacterial strains without harming host cells

With 20 possible amino acids, there are $20^{20} = 10^{26}$ AMPs of length 20

A rational design approach might combine knowledge of bacterial cell membrane composition and structure, AMP molecular and structural properties, host cell membrane characteristics and intracellular pathways—knowledge that may be gained by database/literature search, simulation, experiment



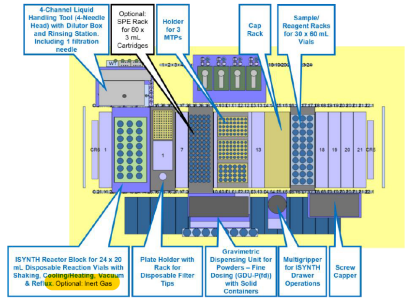
L. T. Nguyen, E.F. Haney, H.J Vogel, The expanding scope of antimicrobial structures and their modes of action, Trends in Biotechnology, 20 (9): 464-472

Automated synthesis and screening platform for antimicrobial peptides design

1

2

Throughput: ~96 peptides/day



Amide Coupling Screening – ISYNTH REACTSCREEN (SWING based)

Peptide synthesis (**PSE**)

3

Throughput: 7 strains x 96 peptides/ day + (16-24 hours)

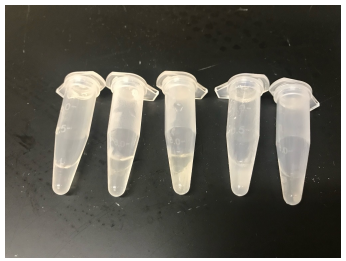


Plate incubator

Hudson robotic arm set up for liquid handling

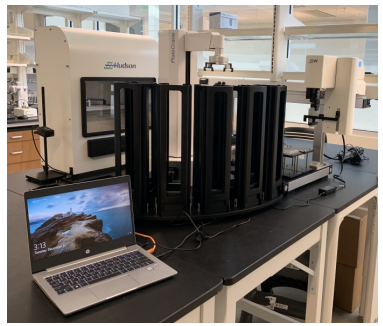
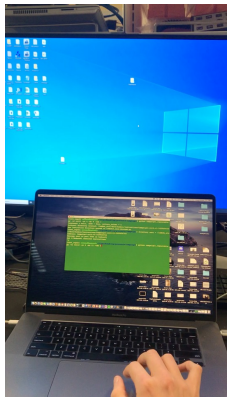
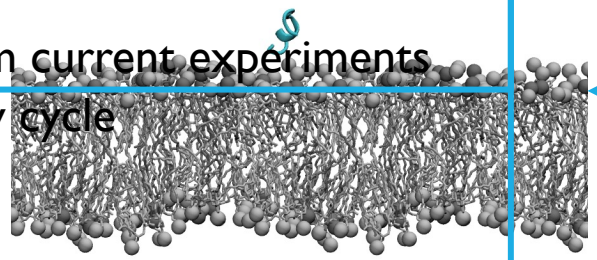


Plate reader



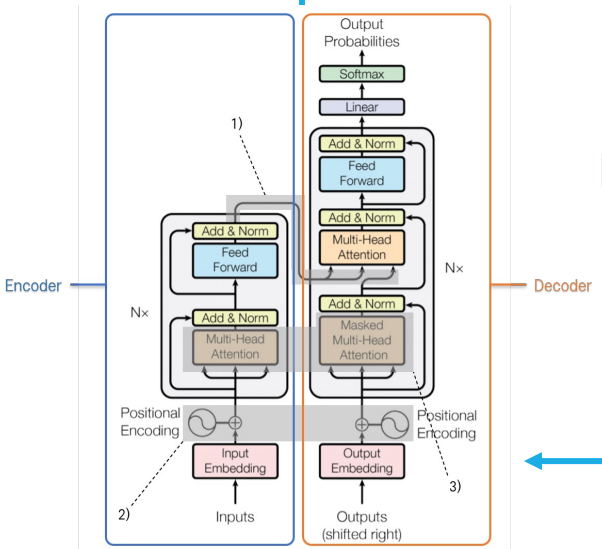
Peptide screening across ~30 Hope College E. coli strains + NIH/CDC ESKAPE collection (**BIO**)

AI models learn successful features from current experiments and constrain generation of AMPs every cycle



Simulations capture behaviors of successful (and unsuccessful) experiments and derive features for constraining AI models

1A

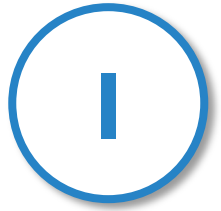


Large-language models for AMP generation (**ALCF**)

Throughput: O(1000s) per hour

Simulations + simulation surrogates (**ALCF**) *Throughput: O(10s) per day*

Some AMP questions and tasks



Q1: Peptide XXX is a known antimicrobial. What is its most likely mechanism of action?"

Q2: Peptide YYY..UUU shows activity against strain G of E. coli. What is the main mechanism of action?

Q3: Can mutation H to YYH…UUUU still act as an effective antimicrobial?

Q4: Pathway P is implicated in action of peptide YYY…UUU as a modulator in strain G of E. coli. What is the likely mode of action that enables this peptide on Pathway P?

Q5: How similar is P to other sub-systems in other organisms?

Task: Define protocols to validate proposed answers to Q1, Q2, Q3, Q4, Q5

Task: Run these experimental protocols in a self-driving laboratory

Courtesy: Ian Foster, ANL/ UChicago

A look at what our scientific assistant has to ”skill” with

PubMed ChatGPT

- Retrieve abstracts **A** from PubMed that reference specified **peptide**
- Use ChatGPT to build hypotheses by using retrieval-augmented generation: e.g.: “Given **A**, which organism is **{peptide}** acting on?”

Protein BLAST protein ▸ protein AlphaFold

- Protein BLAST
- Set up simulations for integrative runs
- Query datasets + assimilate similarity, etc.
- Run AlphaFold + other actions:

BV-BRC NCBI

- Query BV-BRC datasets
- Assimilate similarity scores, etc.
- Enable set up of interactions (e.g., AlphaFold)

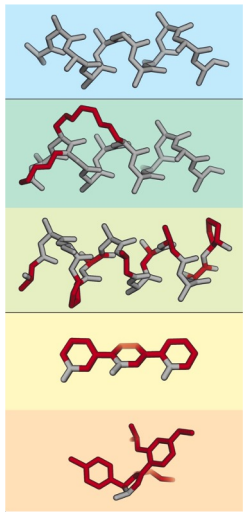
Invoke individual agents, which query databases, retrieve data, run simulations, run experiments, etc.

BV-BRC = Bacterial and Viral Bioinformatics Resource Center

Courtesy: Ian Foster, ANL/ UChicago

Link with self-driving laboratory

Set of peptides as input



PubMed ChatGPT Protein BLAST AlphaFold BV-BRC NCBI

Query PubMed for ChatGPT feedstock

Align proteins, predict structure, rank results

Evaluate structures and filter results

Agents run on HPC/AI resources

PMC Agent

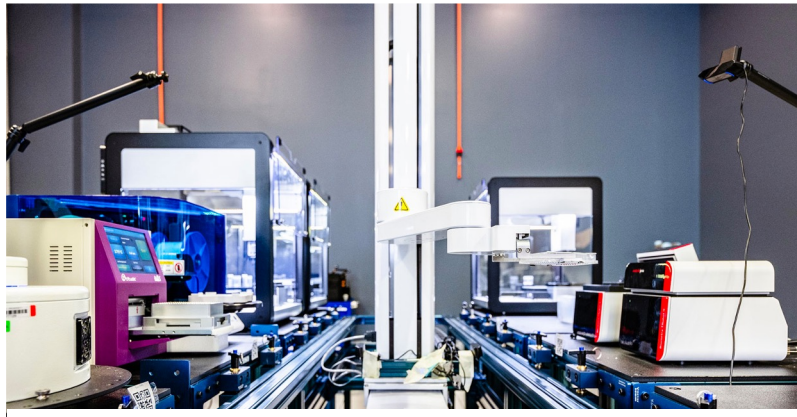
UniProt Agent

BC-BRC Agent

Candidates for experimental evaluation

2

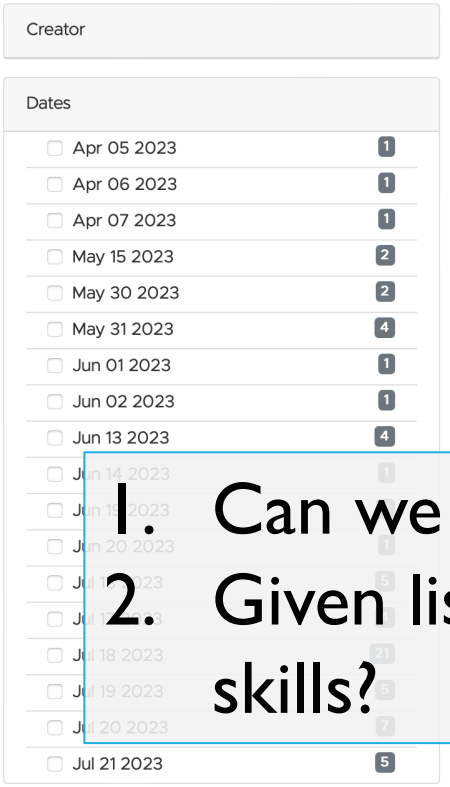
3



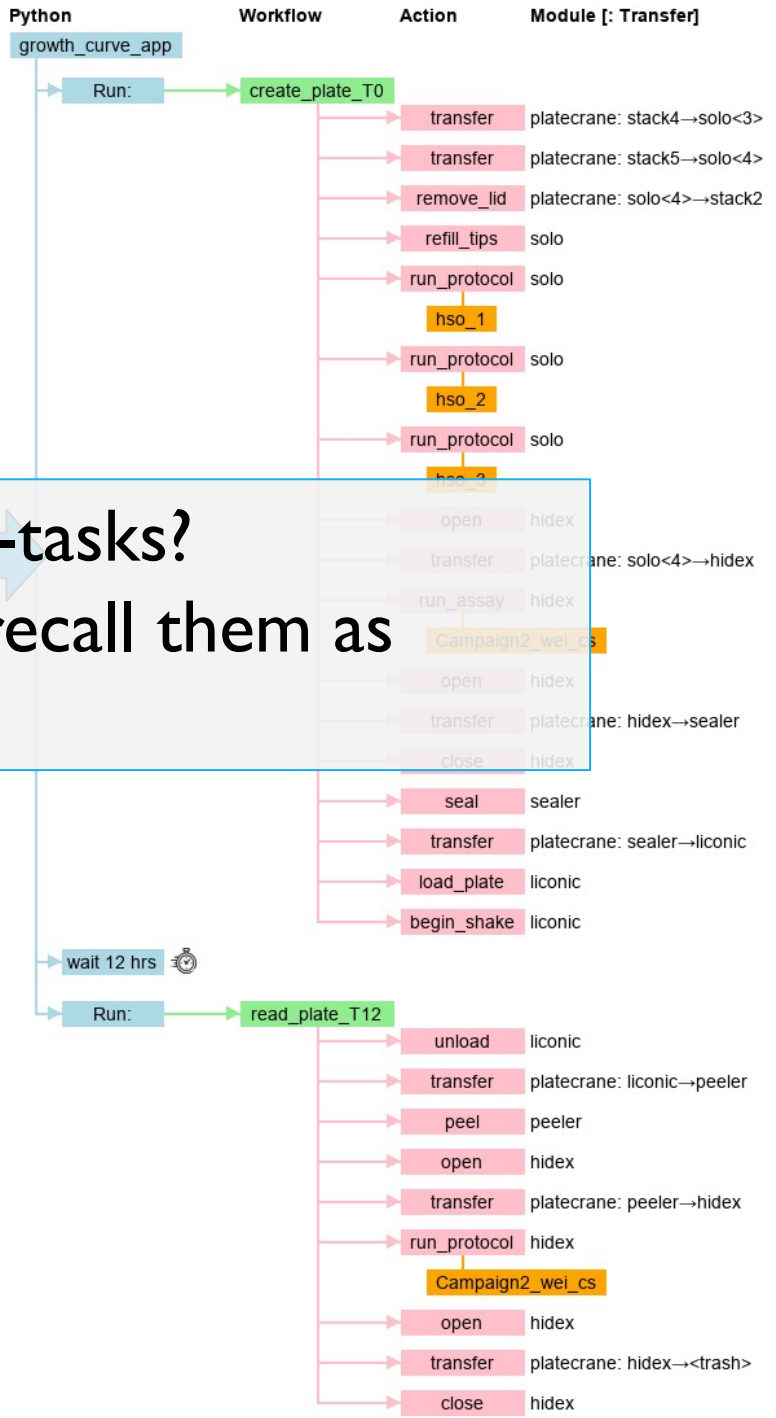
Self-driving lab performs experiments

Courtesy: Ian Foster, ANL/ UChicago

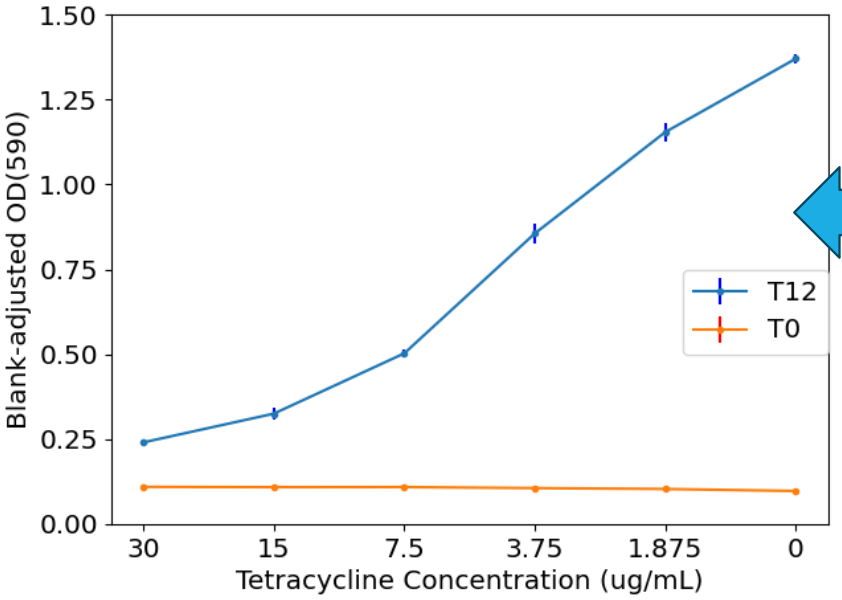
Growth assay application



List of datasets, one per experiment, on data portal.



1. Can we translate lab protocols into a list of lab sub-tasks?
2. Given list of sub-tasks, can we solve each task and recall them as skills?



Results from experiment in which tetracycline solution at varying concentrations was added to E. coli. Y-axis = blank-adjusted optical density at 590nm at start of experiment (T0) and 12 hours after start (T12). Results show mean plus error bars from four identical runs.

Embodied Agent for Automated Lab Code Generation

```
##################### Performing Task 1... ####################
Reasoning: Based on the information provided, it seems like the next logical step
 would be to prepare the master mix for the PCR reaction. This involves combining
 various reagents in specific volumes to create the master mix solution.

Task: Prepare the master mix for the PCR reaction.
```

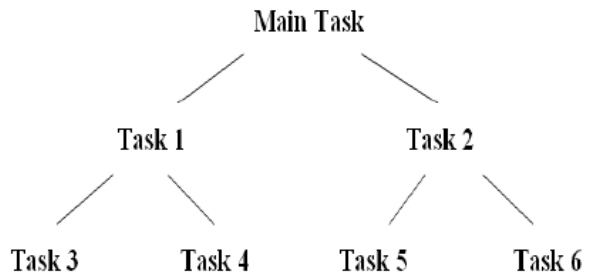
Agent

Memory

Task Prompt

Task Decomp.

Goal Tracking

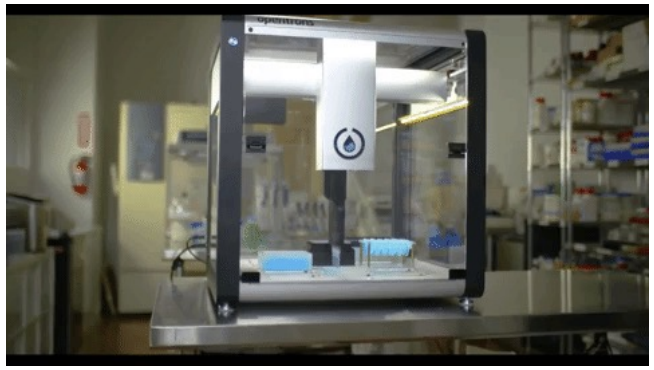


Candidate Code

```
Useful Programs:

def PCR_Master_Mix(labware_info, protocolContext):
    */
    Input: labware_info --> json-str
    Pass in a variable labware_info that contains labware information and
    quantities used

    Output: function call that creates master mix DNA and assigns to appropriate locations
    */

Human:
labware_info = {"number_of_samples":96,
"right_pipette":"flex_8channel_1000",
"left_pipette":"flex_8channel_1000","mastermix_volume":18,"DNA_volume":2}
```

Code Action

Execution Error

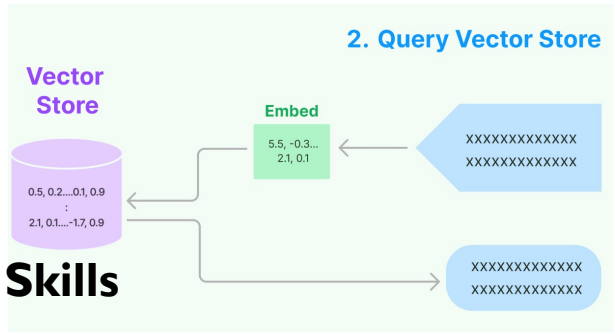


Verify Code

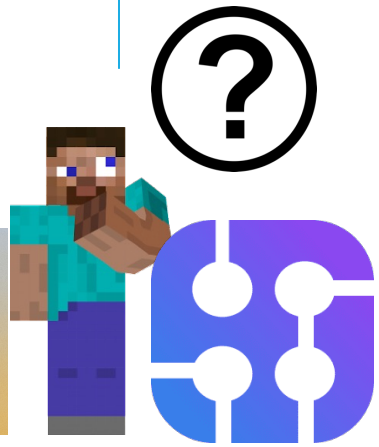
Refine Code

Retrieve

Memory of Tasks



Add Code Skill



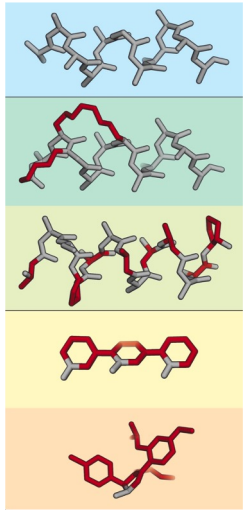
Planning Demo

Code Generation

```
(finalvenv) (base) dhcp-10-105-24-11:Opentrons_Code_Generator BrianHsu$ p
```

Feedback to define additional experiments

Set of peptides as input



PubMed ChatGPT Protein BLAST AlphaFold BV-BRC NCBI

Query PubMed for ChatGPT feedstock

Align proteins, predict structure, rank results

Evaluate structures and filter results

Agents run on HPC/AI resources

PMC Agent

UniProt Agent

BC-BRC Agent

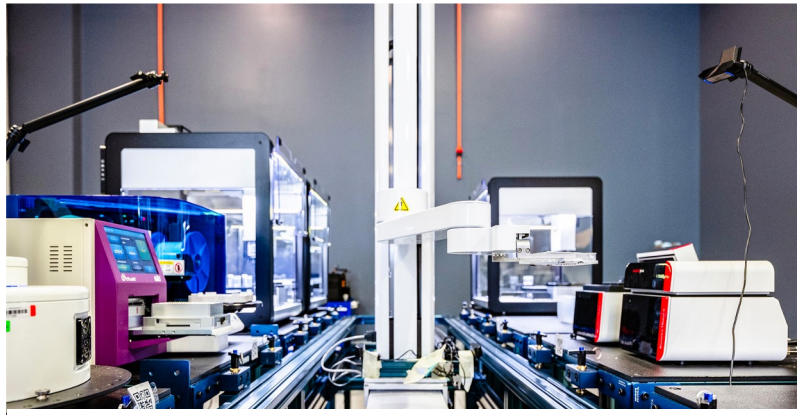
Candidates for experimental evaluation

2

?

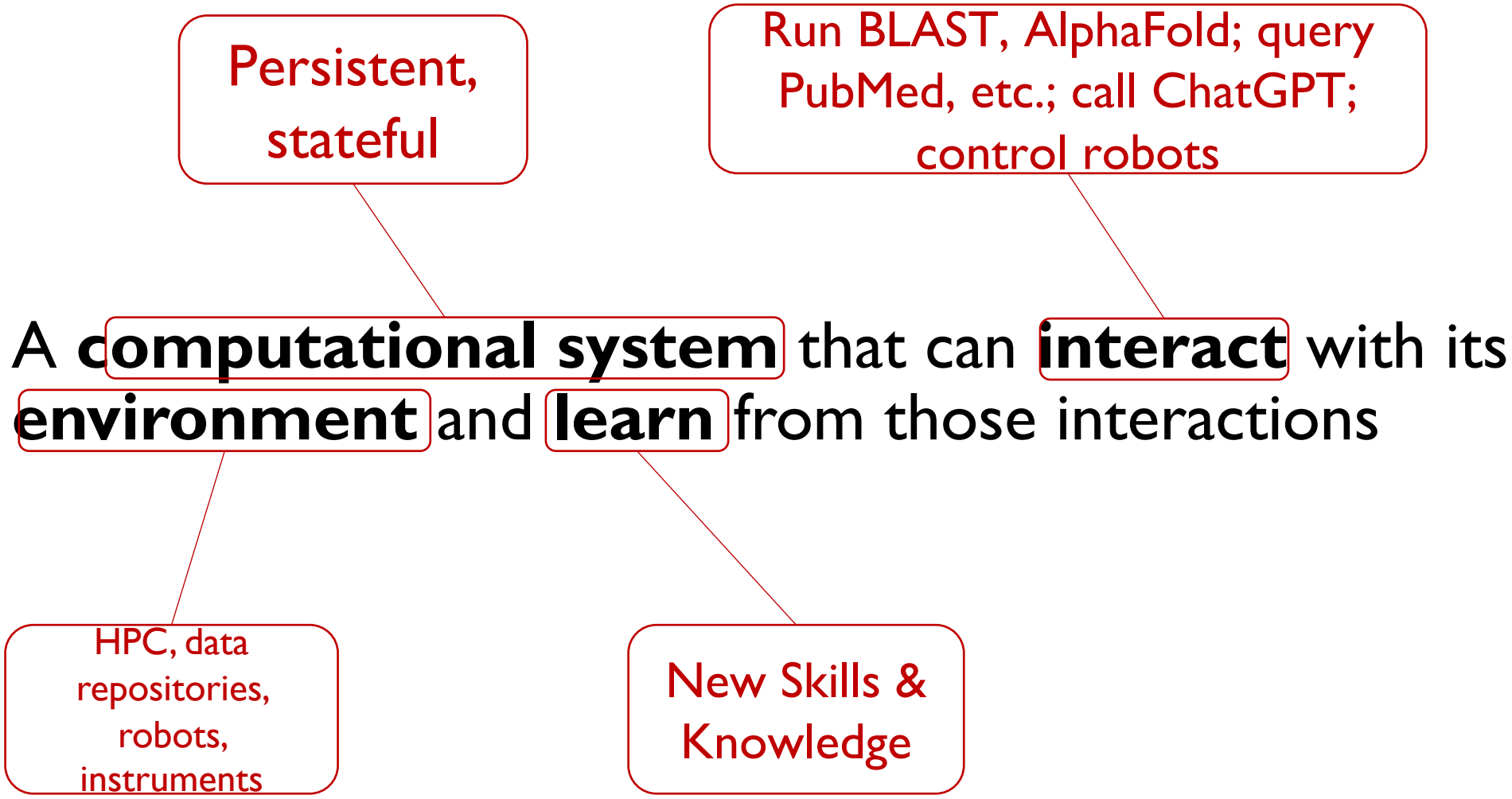
3

Generate additional experiments?



Self-driving lab performs experiments

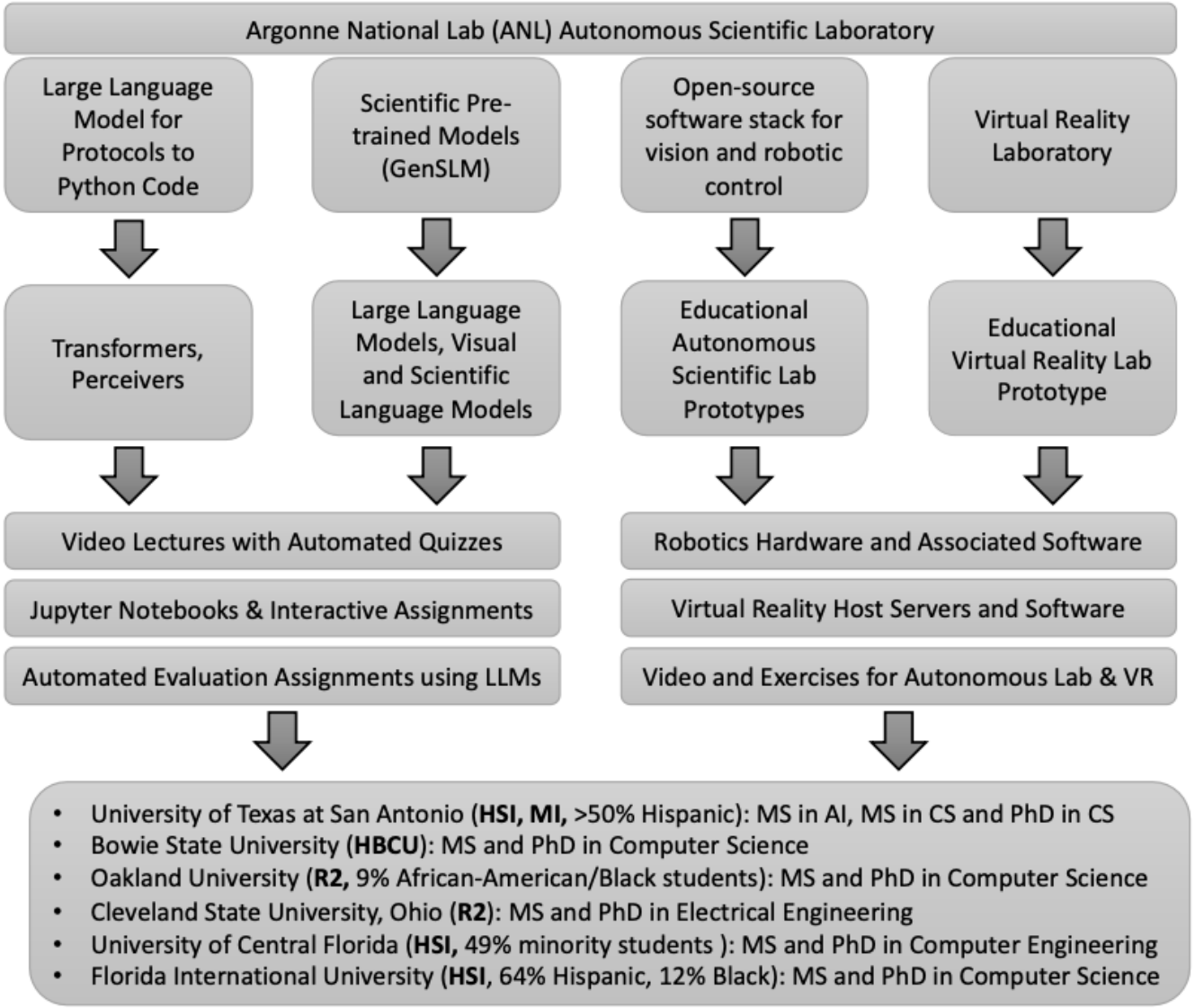
Embodied Agent: AMP Designer…



Courtesy: Ian Foster, ANL/ UChicago

Scaling out the simulation for “smart science factory”

Training the next gen workforce with autonomous laboratories …



Existing Capabilities at ANL

Training of faculty by ANL researchers

AI Software and Hardware at R2/HSI Universities

Outcomes of the MEDAL project

Target Population of Students


J. Greg Paul
UChicago/ANL
Joe Insley
ANL
(NVIDIA)
Romero
(NVIDIA)
UChicago/ANL
Max
ANL
Natalia Vassilieva
ANL
Cindy Bohorquez
Vitor Mateevitsi
ANL
Janet Knowles
ANL
Silvio Rizzi
ANL
Defne Ozgulbas
UIUC
Rick Stevens
Uchicago/ANL
Thomas Br
Uchicago/ANL
Rafael Vescovi
Ramanathan
Lee Liming
Vishwanath
ANL
Lukas Lacinski
ANL
Dana
Lin
ANL
Michael E. Papka
UIC/ANL
Bharat
UIC/NIU
Doga Ozgulbas
ANL
Valerie Hayot
ANL
Abraham Stroka
ANL
Manik Shukla
ANL
Hen
ANL
Casey Stone
ANL
James
J. Davis
ANL
Michael Irvin
ANL
Zhen
Xie
(NVIDIA)
ANL
Weili Nie
(NVIDIA/ Harvard)

Acknowledgements

Funding

- DOE- National Virtual Biotechnology Laboratory (NVBL)
- Exascale Computing Project Cancer Deep Learning Environment (CANDLE)
- Exascale Workflows Project (ExaWorks)
- DOE Codesign for multimodal AI
- NSF MRI: Multi-modal imaging
- DOE-MEDAL (RENEW) project for workforce training

Computing Time

- Argonne Leadership Computing (Theta/ Theta-GPU/ AI-testbed)
- Cerebras/Nvidia
- NERSC

Colleagues

- Richard Scheuermann
- James Olds
- Wesley Scott
- Anda Trifan
- Ashka Shah
- Ozan Gokdemir
- Mike Tynes

Data/ Code/ Models

- https://github.com/ramanathanlab/genslm
- Access to model weights will also be available via API

Questions/Comments

ramanathana@anl.gov