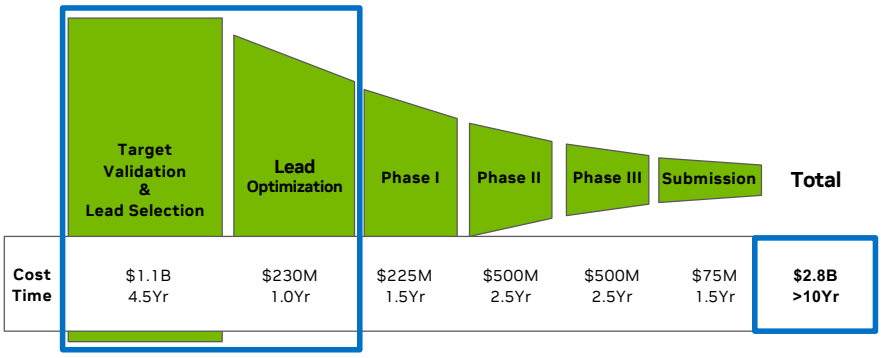
Thank you to the organizers for the opportunity to present, and I 'd like to thank you all for making it on this Friday morning. Fun fact – my first PyData was exactly 10 years ago. It was fun to browse the old schedule online and reflect upon how the field and the python ecosystem have changed.

Motivation: Drug Development is a Long and Expensive Process

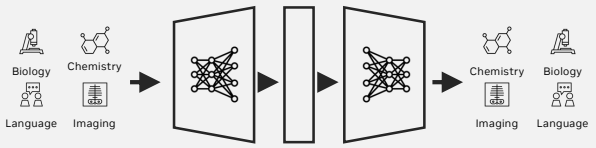


$2.8B and >10 Years to Bring a Drug to Market

Original source: *Developability assessment as an early de-risking tool for biopharmaceutical development*, J. Zurdo, 2013, DOI: 10.4155/pbp.13.3

Language Models are Revolutionizing Discovery

- Information from biomedical literature
- Prediction of chemical reactions
- Biomolecular property prediction
- Structure prediction and docking



BIOMEDICAL NLP
Learn all of PubMed

GENERATIVE CHEMISTRY
Novel Drug Candidates

PROTEIN STRUCTURE
Predict 3D Structures

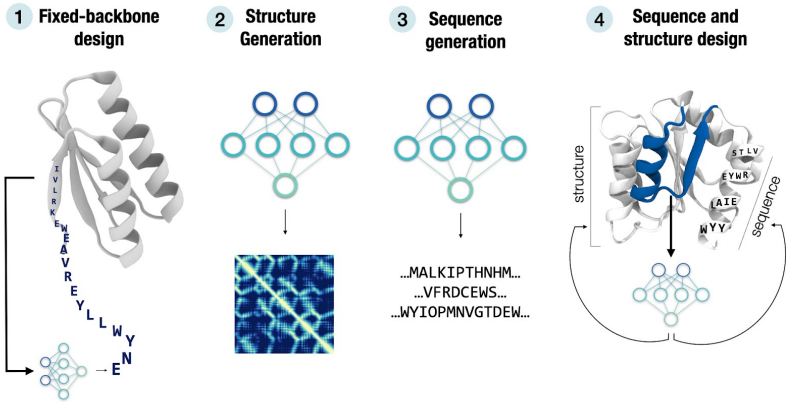
VIRTUAL SCREENING
Docking and Pose Prediction

One advantage of going last is that previous speakers have discussed your introductory material for you. As Andrew White noted yesterday – language is a powerful way to represent biology, which is probably not a coincidence given that we as scientists must use language to describe biology. Jacqui Cole also covered the topic for materials science in the third keynote yesterday. (Kevin Jablonka)

The utility of language models in a variety of synthesis related tasks was discussed by Daniel Probst, Samuel Genheden, and Steven Bennett.

Other notable applications exist as well – such as prediction of properties and providing protein features for structure models. May even be possible to use sequence directly for structure.

From Sequence to 3D and Back Again



Ferruz, N. *et al.*. *bioRxiv* 2022.08.31.505981 (2022)

Focus on proteins and how deep learning is applied
Sequence of amino acids – form coils and flat regions – secondary structure, fold up into a three dimensional conformation
Predict which amino acids are close
Generate 3d
Design both sequence and thre dimensional structure

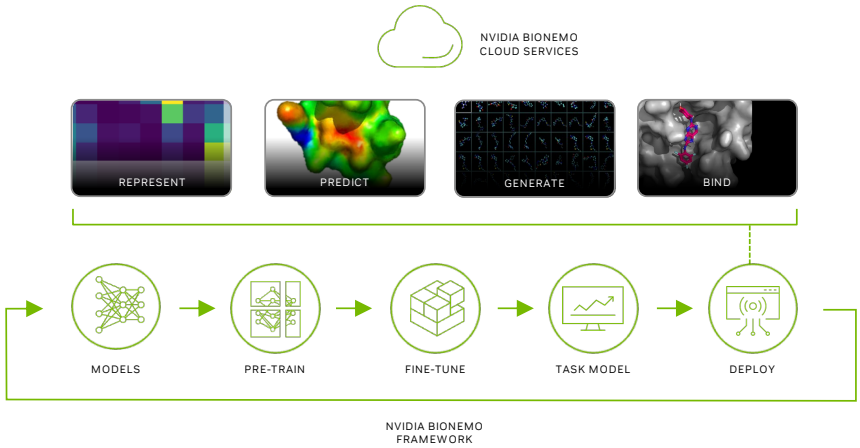
Outline

- Overview of BioNeMo: Inference Service and Training Framework
- MolMIM: Development of a Small Molecule Foundation Model for Generation
- Career Progression and Lessons from the Field

Let's begin by discussing BioNeMo.

NVIDIA BioNeMo

AI Tools, Frameworks, and Applications for Drug Discovery



NVIDIA

absci ALCHEMAB THERAPEUTICS AMGEN astellas AstraZeneca BROAD INSTITUTE Deloitte. Flagship Pioneering Innophore

Insilico Medicine InstaDeep MAYO CLINIC Meta Mila OpenFold Relation ROSTLAB VYASA

Bit of a marketing slide, but I think it does a nice r the framework and cloud services relate to each other.

There are two parts to BioNeMo – a training fran ice service. Long term objective is training-as-service. Models are developed or trained elsewhere with data and model parallelism as provided by the framework, then fine-tuned and specialized for tasks, then deployed to the inference service, if desired.

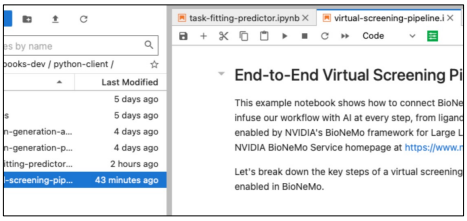
That's the ideal. However, as the situation exists right now the connection between framework and service isn't fully baked..

However, external models can also be deployed to the inference service, and that's where we started. Will cover inference service first, and then the training framework.

NVIDIA

Multiple Interfaces to a BioNeMo Model in the Inference Service

Interactive UI and Jupyter Workflows



API and Python Client

```python
import requests

ngc_token="<<NGC TOKEN>>"
headers = { "Authorization" : f"Bearer {ngc_token}" }

try:
    response =
requests.post("https://api.stg.bionemo.ngc.nvidia.com/v1/protein-
sequence/protgpt2/generate",
        headers=headers,
        json={
        "max_length":150,
```

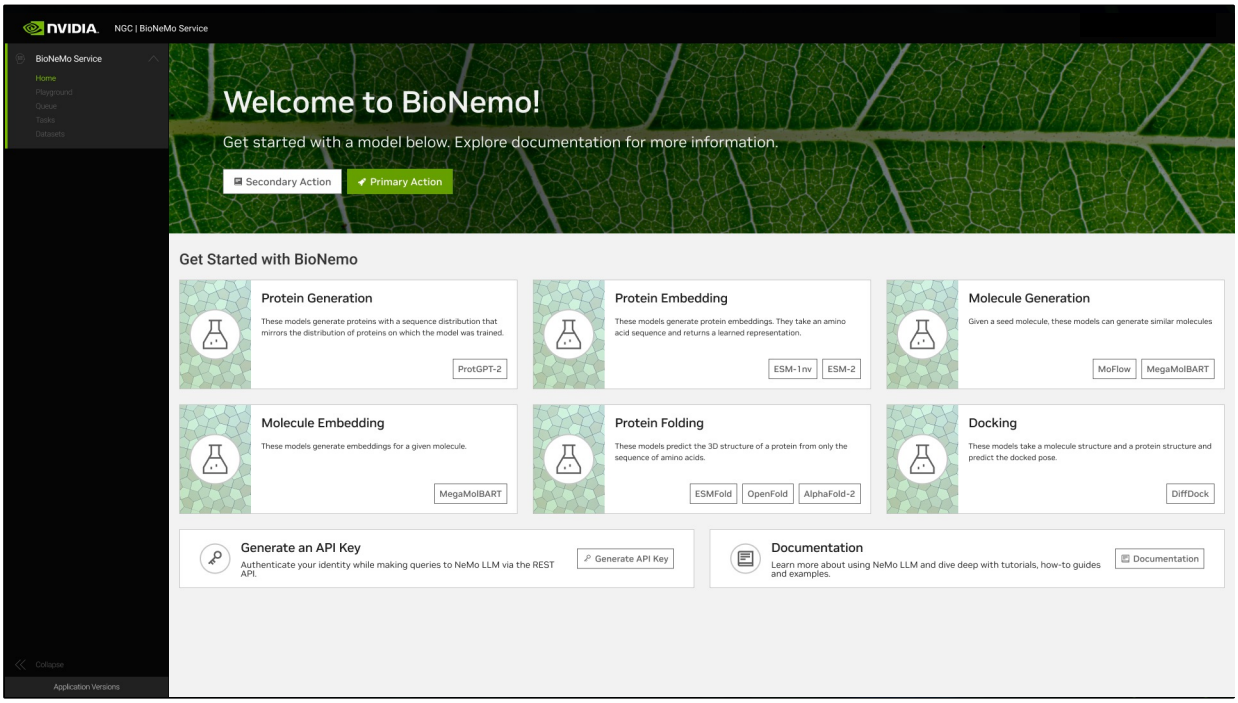
```python
from bionemo.api import BionemoClient

# Create a client instance
api = BionemoClient()  # NGC_API_KEY env var is read
used.

# Generate novel proteins
novel_proteins = api.protgpt2_sync(max_length=200, nu
# Fold the first protein
folded_protein = api.openfold_sync(novel_proteins["ge
```

Add API and Python interface. Can use standard python libraries such as requests, or there is a client. So you can pip install BioNeMoClient.

The web interface for each model has an interactive UI for outputs. Jupyter notebooks exist to demonstrate more advanced workflows



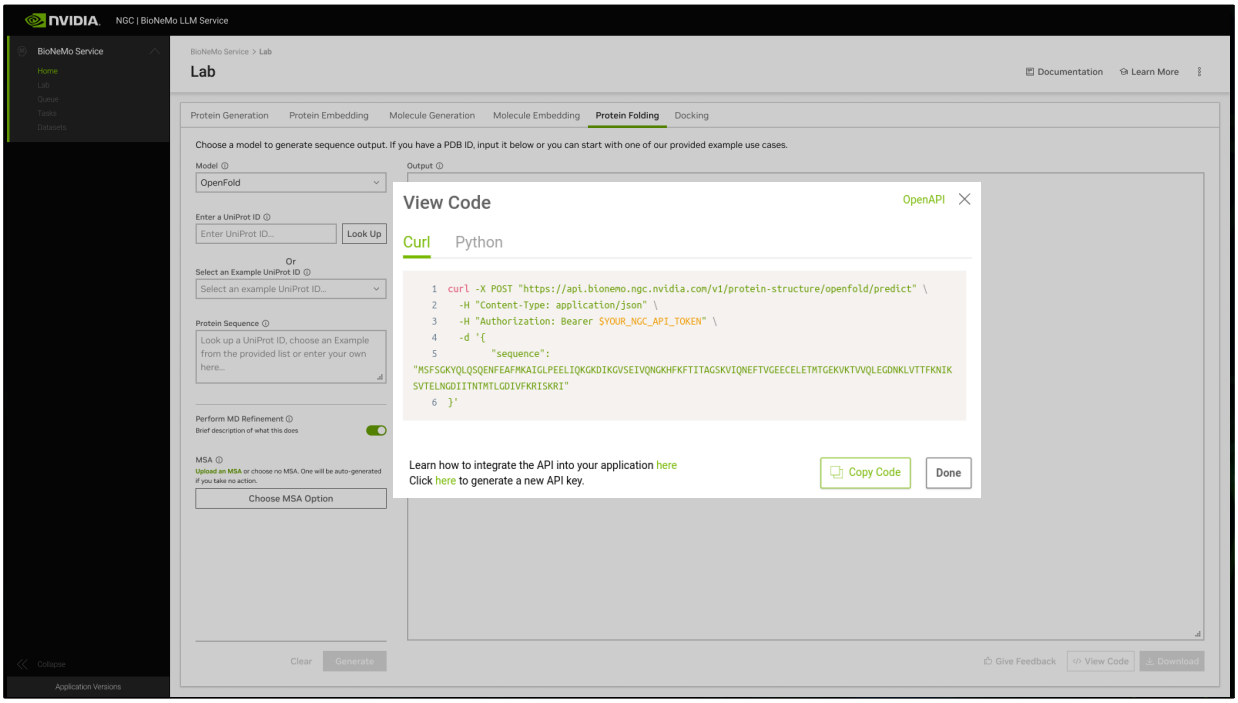
Of course needed to build interface to use this, user is scientist who doesn't want to deploy own models, or a software vendor.
I decided not to throw caution to the wind and attempt a live demonstration here, but I'd like to walk through some stills of highlights of a few of the models. This is the landing page for BioNeMo, lists all the models, documentation, and how to get an API key

NVIDIA

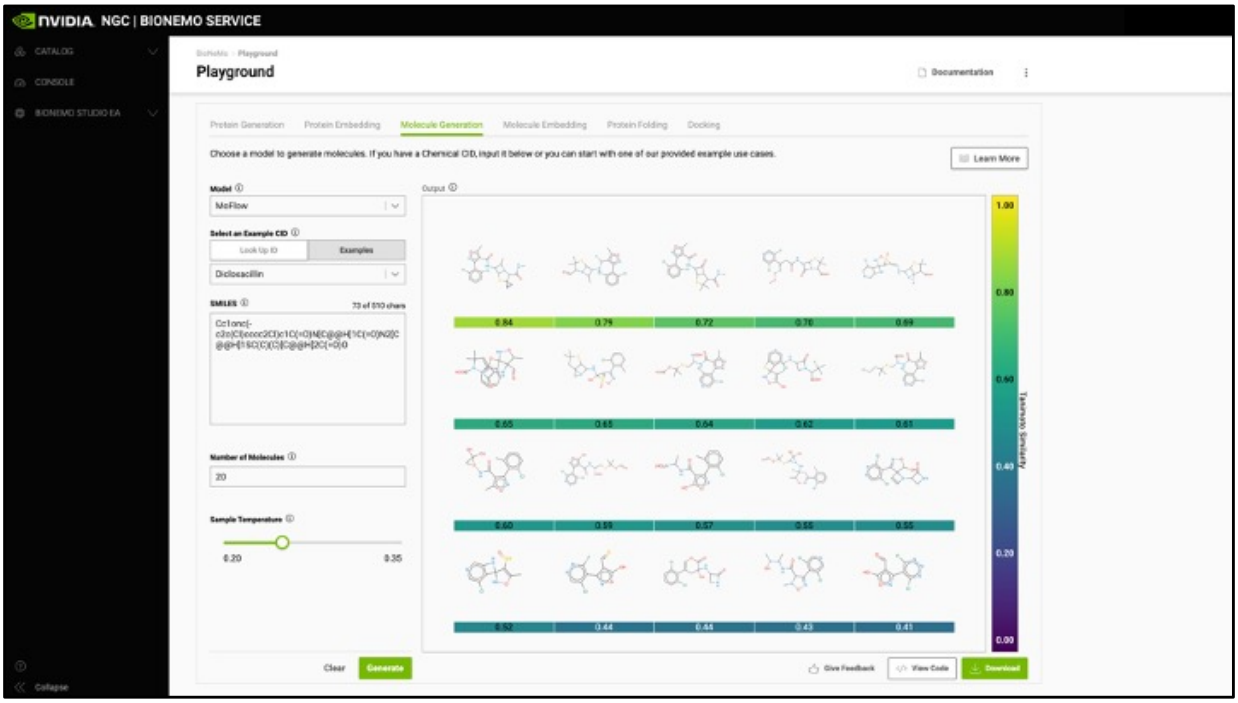
Here is an example of a protein – transferase for small peptide (linaridin) from Streptomyces
Used OpenFold, however dropdown can be used to select alternatively AlphaFold2 or ESMFold. Protein is entered as sequence or PDB ID. Can optionally provide own MSA. Also option to perform relaxation step although not shown here.

Structure generated in an interactive GUI – PyMol like. Perform annotations.

Images can be downloaded, as can PDB of prediction



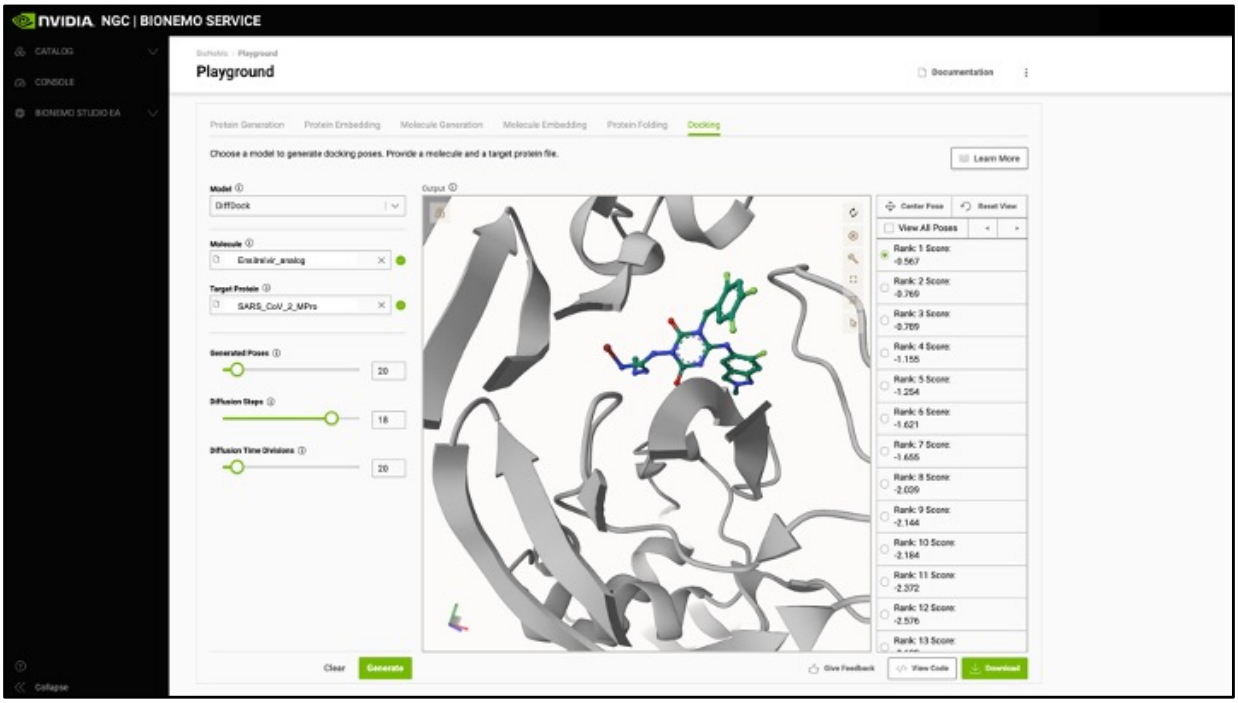
For each model prediction, the code is generated that will perform the prediction from the command line and from python. Can be used in scripts to automate calls to the API. Alteranatively can also use python client.



Here is an example of 20 molecules generated from MoFlow with standard sampling temperature, starting from Di-cloxacillin. Can enter molecules as SMILES or with ChEMBL ID.

Tanimoto similarity plotted below. Can download generated molecules.

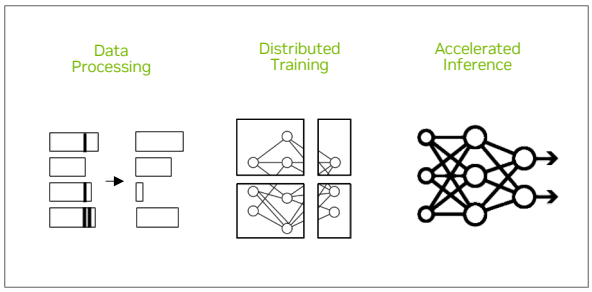
Antibiotic for staphylococcal infections.



DiffDock example. Performed blind docking a potential antiviral compound in SARS CoV main protease.

Upload the protein as a PDB file, and the molecule as an SDF file. Standard parameters for DiffDock, including the number of generated poses. On the right are the poses listed by confidence score, can select one or more to show. Image can be downloaded, or can download a PDB file with protein structure and all poses.

BioNeMo Framework Overview



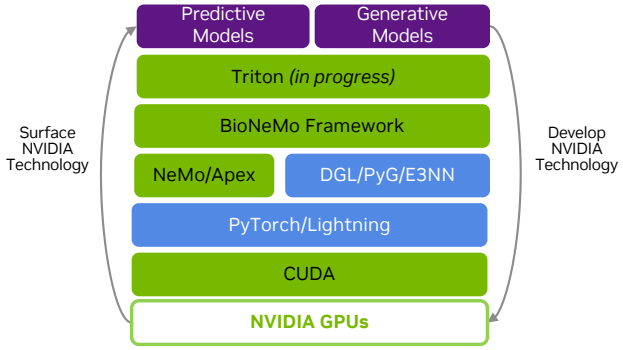
- Includes dataset processing, training, fine tuning, and example downstream tasks
- Support for multi-GPU and multi-node training
- Data parallelism, and three types of model parallelism
- Currently three LLM models for cheminformatics and protein applications – more models and model types coming soon

Now let's briefly touch on BioNeMo framework. The framework provides functionality to download and process datasets and perform pre-training. Fine-tuning capabilities are also provided so that models can be tuned for specific purposes. Example downstream tasks exist, recognize that this will usually be performed on internal data, so these are examples.

For inference, provide gRPC class and example notebook. Working on more automated deployment functionality.

Thre LLM models are provided – MegaMolBART for cheinformatics, and ESM1, ProtT5 for protein sequences. Additional models are in development including ESM-2, several nucleic acid models, and MolMIM, which will be covered in next section. Also working on several equivariant models – EquiDock, OpenFold, and DiffDock, discussed at the end.

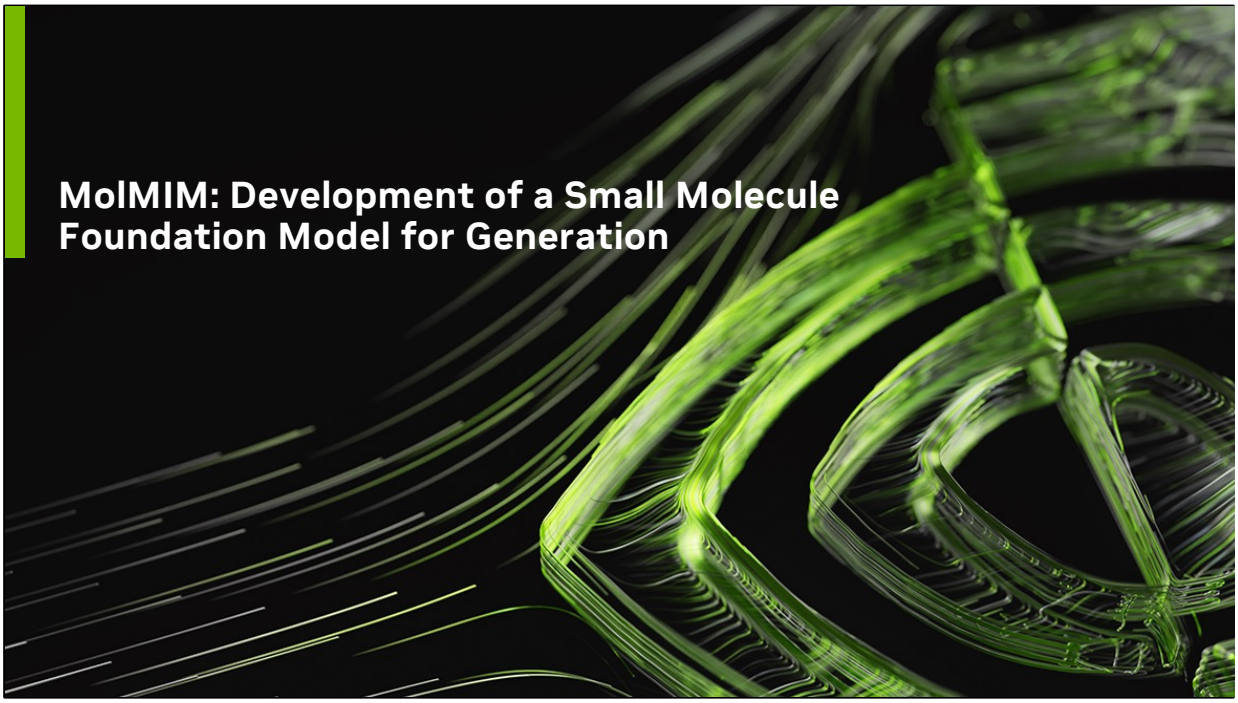
BioNeMo Framework Technology Stack



- Based on NVIDIA NeMo, which is a library for development and training of LLMs
- Automated deployment with Triton is in progress
- Surface and develop new software and hardware technology

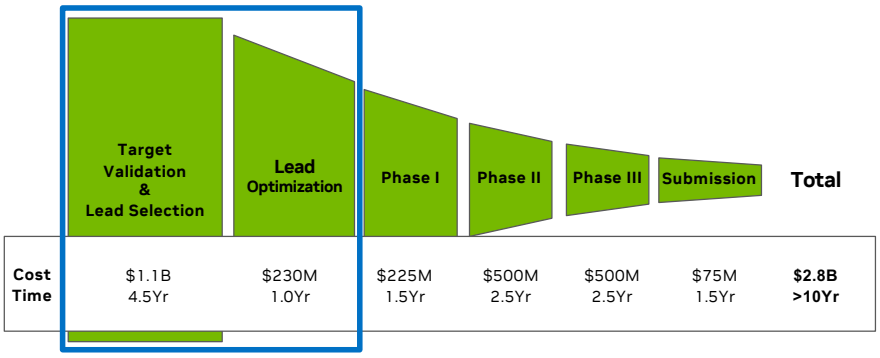
Technology stack diagram demonstrates one of the initial perspectives I provided on BioNeMo. It provides a way to surface hardware and software technology. See example of that in DiffDock section at the end. For now, this means specifically GPUs. In the future, that may also include CPU technology with Grace-Hopper configuraitons.

Conversely, challenges faced during development can be used to drive the development of new software and potentially even hardware changes.



So that was an overview of BioNeMo framework and inference service. Now let's discuss some of the team's R&D work. Will begin with small molecule language models.

Motivation: Drug Development is a Long and Expensive Process



$2.8B and >10 Years to Bring a Drug to Market

Original source: *Developability assessment as an early de-risking tool for biopharmaceutical development*, J. Zurdo, 2013, DOI: 10.4155/pbp.13.3

NVIDIA

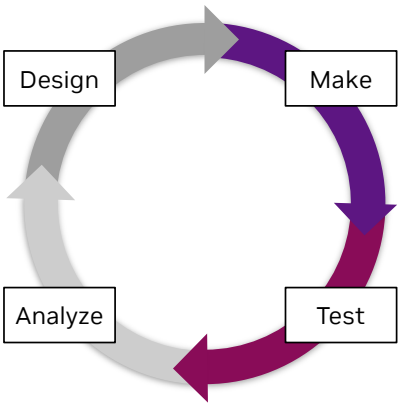
Thank organizers, honored to keynote. Sat in your chair 10 years ago
Three topics I'd like to cover today – product for which I'm the tech lead, called BioNeMo
Sample of the model development my team does
How I ended up in this role and a few lessons I've learned along the way, mostly the hard way

NVIDIA

Lead Discovery: Three Years for Design-Make-Test-Analyze Cycle

Hit Compound

Known or experimentally determined
Weakly active
Target unselective
Toxicity risk
Low metabolic stability



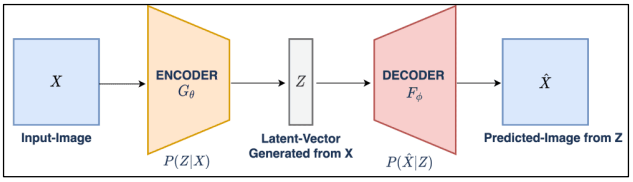
Candidate Drug

Highly potent
Effective for *in vivo* models
Metabolically stable
No toxicity issues

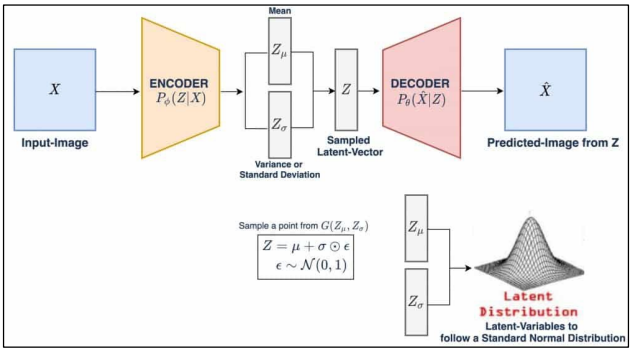
Multiple of DMTA cycles at 4-6 weeks/cycle
Transition between multiple labs

Autoencoder Models in a Nutshell

Autoencoder



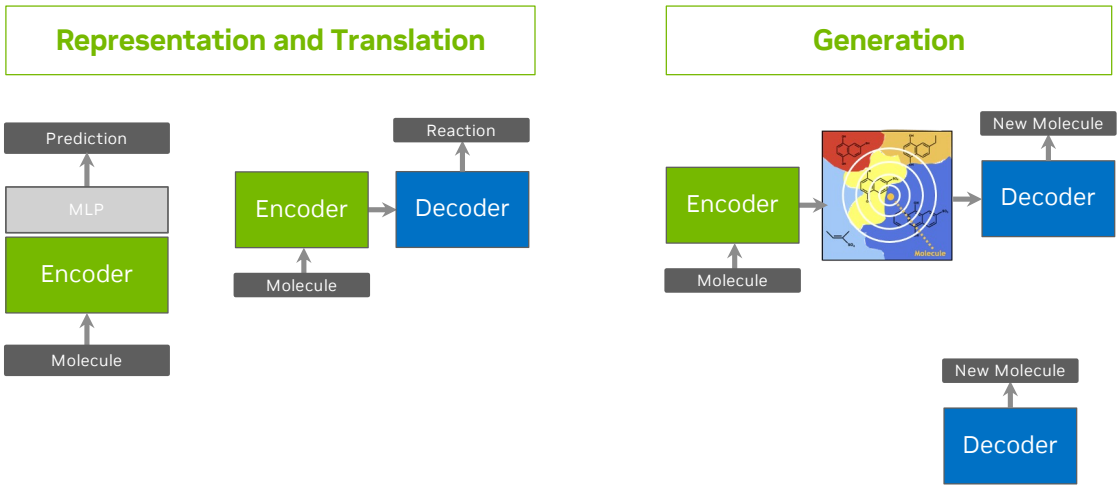
Variational Autoencoder (VAE)



Also works with sequences -- seq2seq models

Quick level set there are a variety of backgrounds present today, input data image – like PCA

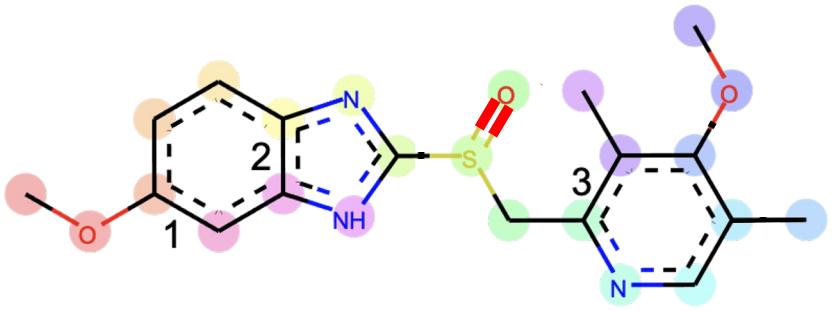
Cheminformatics Foundation Model Objectives



One of the long term objectives of a paired framework and inference service is to support development of foundation models. Ideally these models would support both representation and translation tasks (for seq2seq models). Can include things like property prediction. An example of a translation task would be retrosynthesis prediction.

For generation, this could be from some means of latent space maninpulation (conditional generation, RL, random sampling) and also de novo generation.

SMILES: a Natural Language Representation of Small Molecules



COc1ccc2n c(S (=O) Cc3ncc(C) c(OC)c3C) [nH]c2c

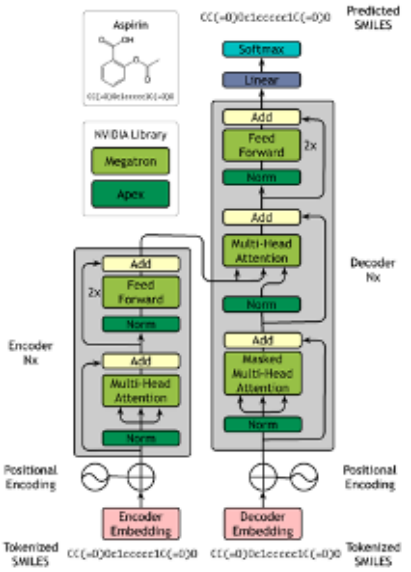
NVIDIA

Micght be wondering how we do this
Simplified molecular line input system

NVIDIA

MegaMolBART Molecule Representations

- MegaMolBART is a sequence-to-sequence developed in collaboration with AstraZeneca
- Based on BART NLP model
- Trained on 1.5B small molecules in SMILES format
- Useful for representation and sequence translation tasks
- Not well suited for generation tasks -- lacks an organized and uniformly shaped latent space



Chemformer publication: Irwin, R., *et al*, Mach. Learn.: Sci. Technol. 3 (2022). 22

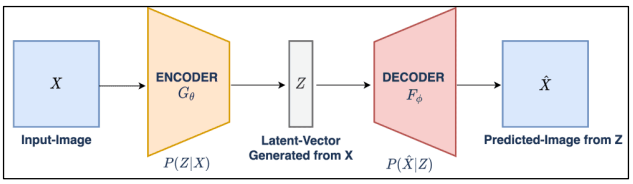
The first cheminformatics model that we developed was called MegaMolBART. It was developed with a collaboration with AstraZeneca and is based on a model developed by AstraZeneca, called Chemformer, as mentioned by Samuel Genheden. (see publication). Was ported to NVIDIA's Megatron framework, which is designed for development, training, and inference of transformer models at large scale. It is a BART model, as the name implies and was trained on 1.5B molecules from ZINC15.

MegaMolBART is useful for representations and sequence2sequence translation tasks. However, not designed for generative tasks – does not have an organized latent space. Additionally, there are other challenges. The encoder output varies with sequence length, so must use pooling to handle this, and then unclear how to broadcast this dimension. Decoding speed increase non-linearly with

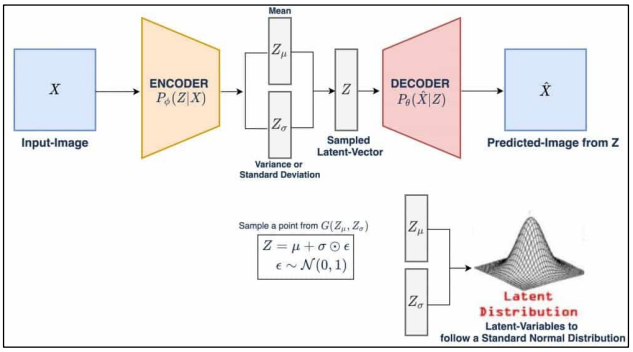
molecule length.

Autoencoder Models in a Nutshell

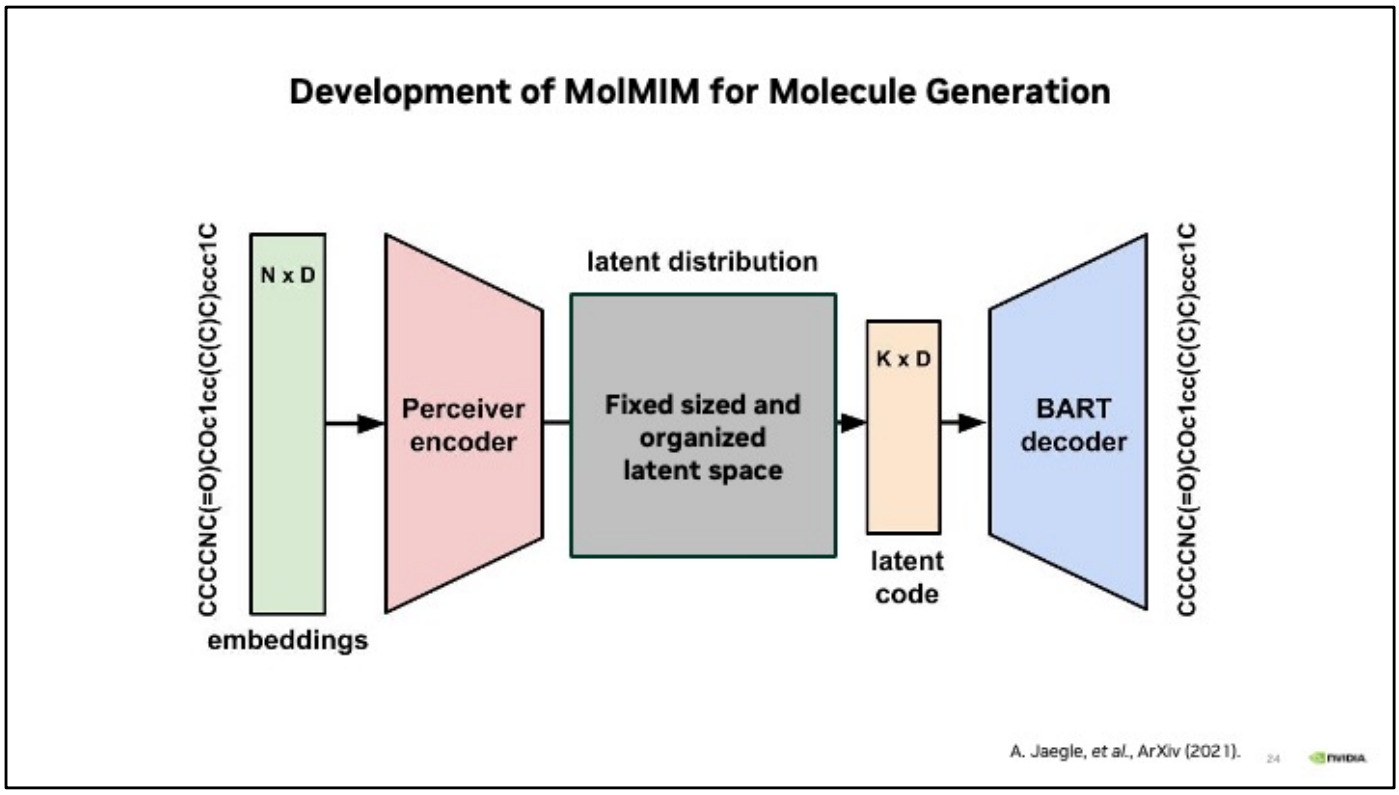
Autoencoder



Variational Autoencoder (VAE)



Wanted to design something that had a structured latent space – slight variantion



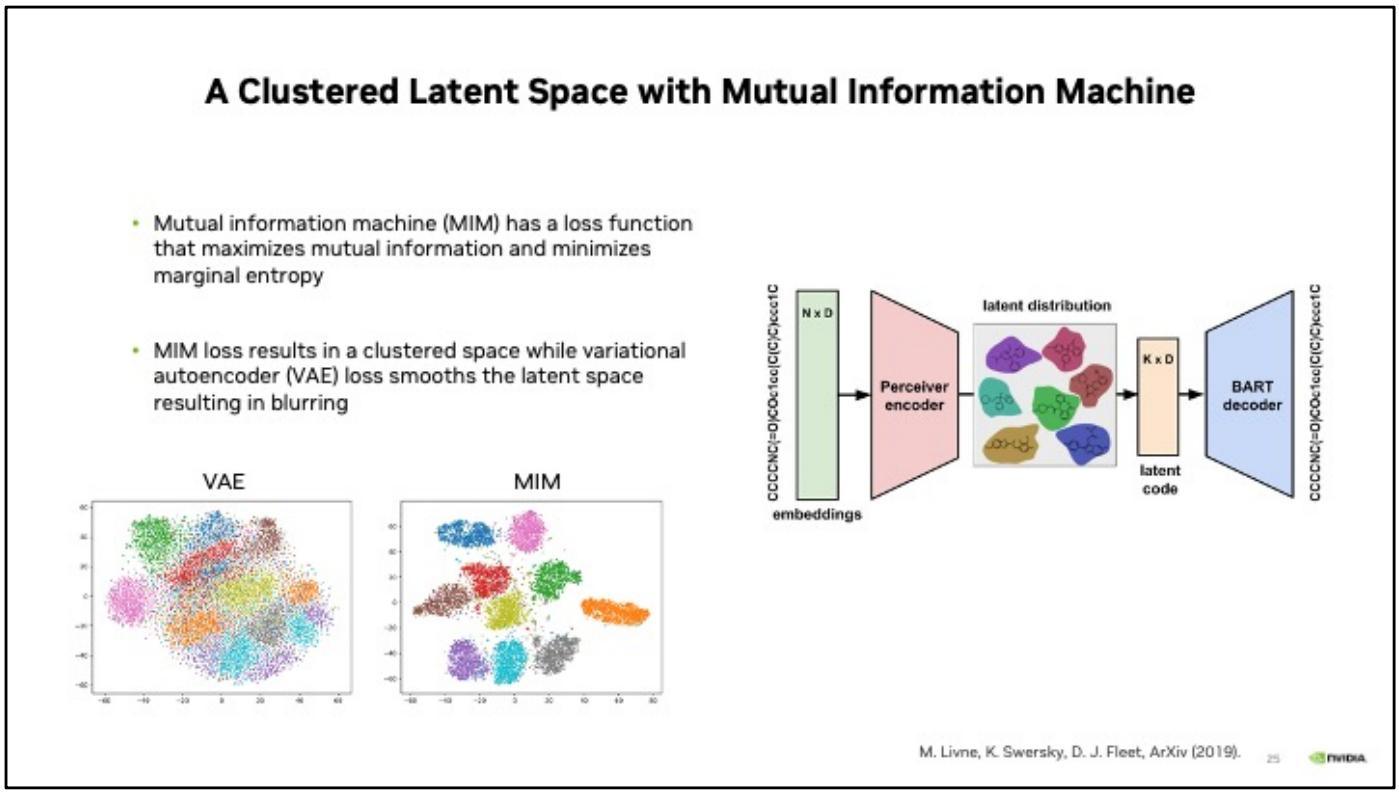
As one does, we borrowed from existing model architectures to address the problem of variable sequence length. Used a Perceiver model, which contains a cross attention mechanism that creates a fixed size latent space – effectively a form of dimensionality reduction.

Fixed size dimension is k, and has improved runtime complexity relative to the transformer. Created a PerceiverBART and trained on half the data as MegaMolBART.

k – Perceiver dimension
S = sequence
D = hidden size

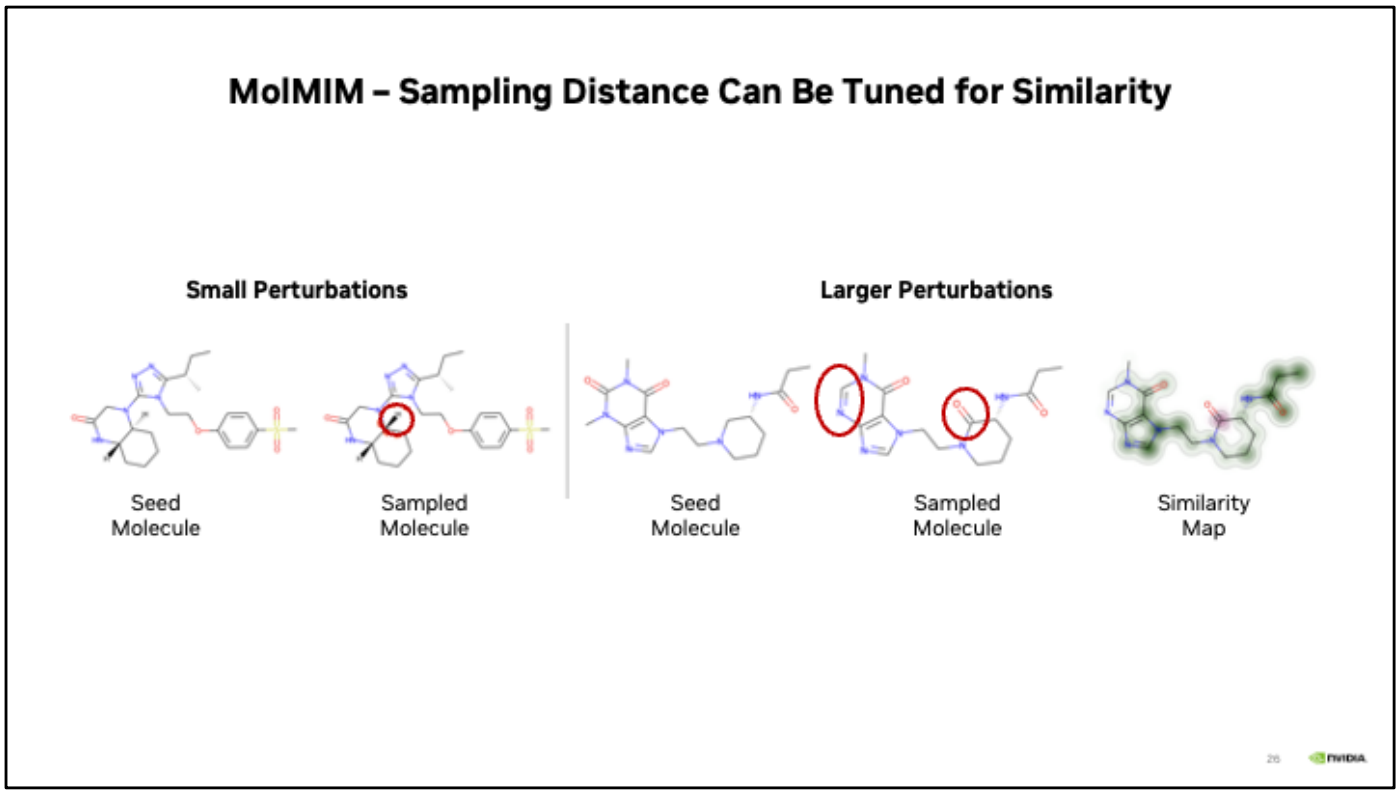
What's shown on the left just the encoder, there is an analogous dimensionality change in decoder).
Described in a series of papers from Google.
Byte Array = tokenized molecule
S = sequence/token length
C = hidden size

Latent Array = randomly initialized and learned
D = hidden size
N = k → fixed dimension



However, PerceiverBART doesn't solve problem of having an organized latent space. Solved that using a language model that one of the ML researchers on my team developed during his postdoc – called mutual information machine. Has a loss function that maximized mutual information and minimized marginal entropy.

Otherwise same architecture as a VAE. Differences in loss functions can be observed on right. Dimensionality reduction performed on images and colored by class. Can see how VAE blurs class boundaries while MIM results in clean boundaries.

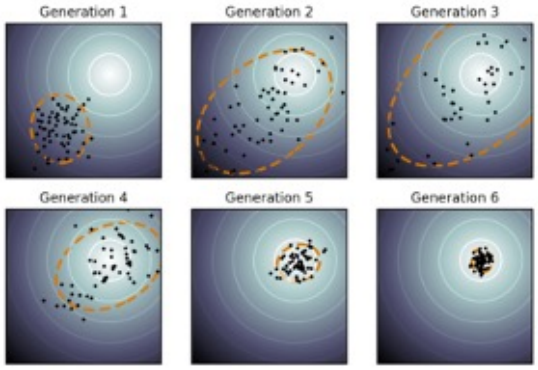


Example from a qualitative examination of latent space. Very small distances result in negligible changes, such as inversion of stereo chemistry on left.

Larger changes show removal of methyl group and two carbonyls

Measuring the Controllability of MolMIM Generation

- **Hypothesis:** having a structured latent space will improve performance of property guided optimization
- Chose covariance matrix adaptation (CMA-ES), which is a zeroth order optimization method
- CMA-ES is non-parametric and uses only a single scoring function per sample



N. Hansen, A. Ostermeier, *Evol. Comput.* 9, 159–195 (2001). 27 NVIDIA

Hypothesis that organized latent space would be controllable

Es – evulaitonary strategy

Multi-Objective Property Optimization

- Performed multi-objective optimization to jointly optimize two molecule properties (QED, SA) and binding to two protein targets (JNK3, GSK4β)
- Novelty is proportion of molecules with similarity metric (0.0 – 1.0) less than ≤ 0.4 relative to any other molecule
- Diversity is average similarity across all compounds
- MolMIM is competitive for success and diversity, but novelty has room for improvement

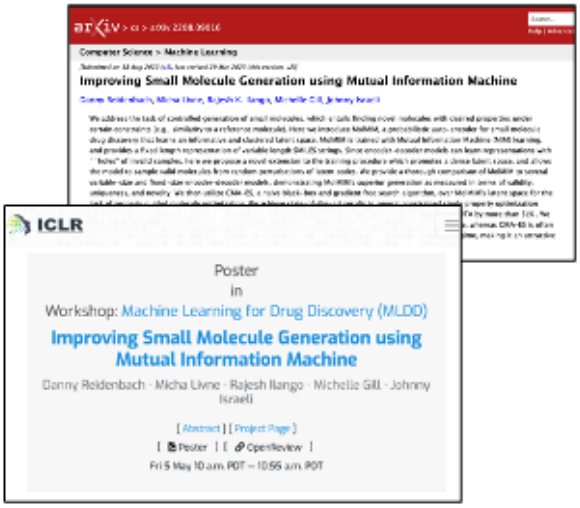
Model	QED + SA + JNK3 + GSK4β		
	Success (%)	Novelty (%)	Diversity
RationaleRL	74.8	56.1	0.621
MARS	92.3	82.4	0.719
JANUS	**100**	32.6	**0.821**
FaST	**100**	**100**	0.716
MolMIM (R)	97.5	71.1	0.791
MolMIM (A)	96.6	63.3	0.807
MolMIM (E)	98.3	55.1	0.767
MolMIM (E)†	99.2	54.8	0.772

Results above solid bar as in B. Chen, X. Fu, R. Barzilay, T. Jaakkola, ArXiv (2021)
QED, SA, JNK3, and GSK4β oracles from Therapeutic Data Commons

Optimization types.

JANUS model – genetic algorithm that mutates selfies. Models all trained with some notion of chemical properties or task specific reinforcement learning.

MolMIM: Research to Productization



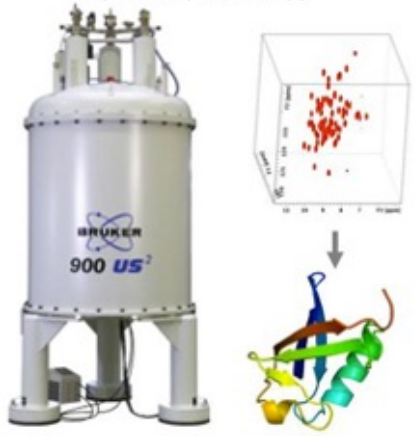
- Integration of MolMIM model into BioNeMo inference service
- Productionize model architecture and training framework
- Accelerated inference
- Improving encoder representations

MolMIM paper is on archive, and we presented a poster at ICLR
Ongoing work
Benchmarks -- service

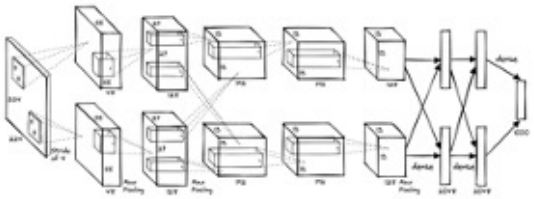
So that was an overview of BioNeMo framework and inference service. Now let's discuss some of the team's R&D work. Will begin with small molecule language models.

From Structural Biologist to Data Scientist

Postdoctoral Research: Enzyme Dynamics by NMR Spectroscopy



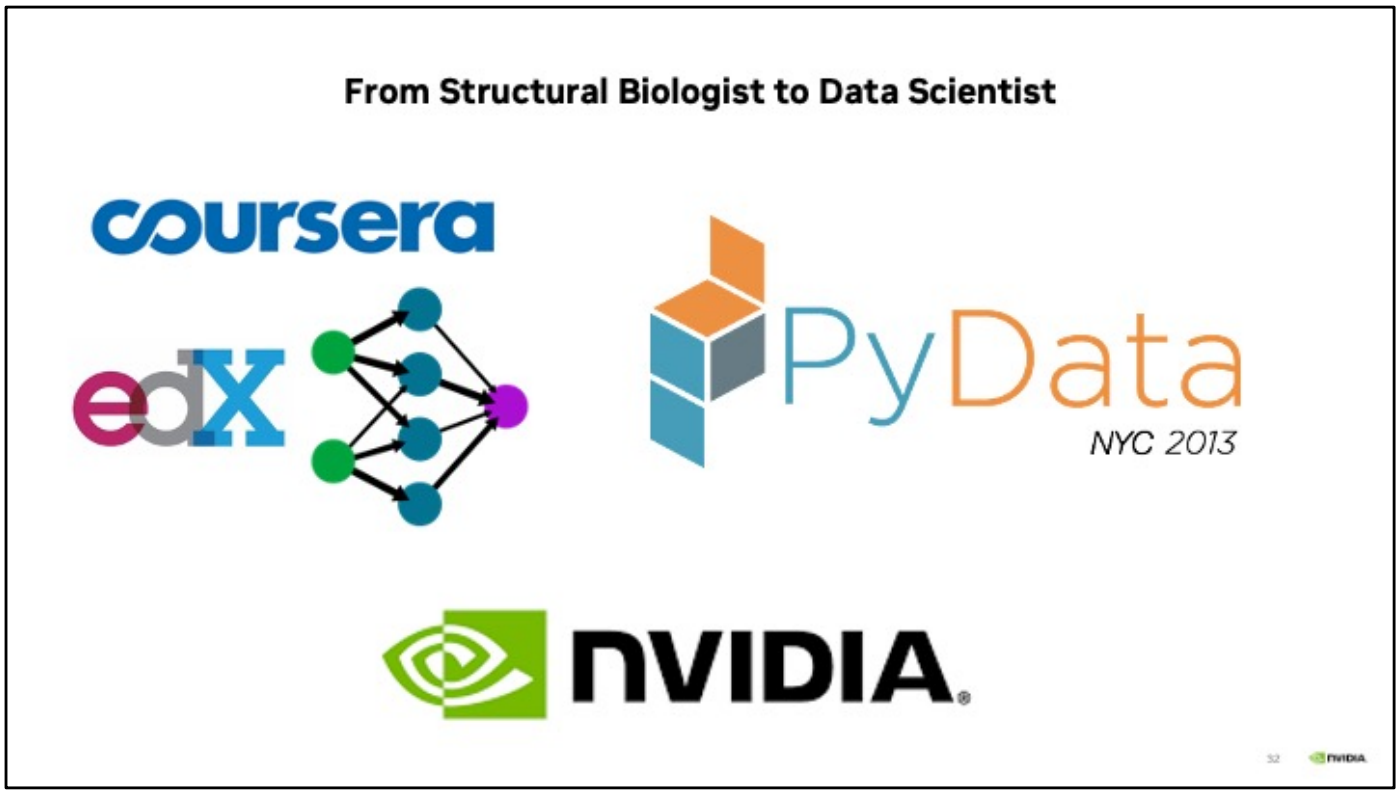
AlexNet Won ImageNet Challenge in 2012



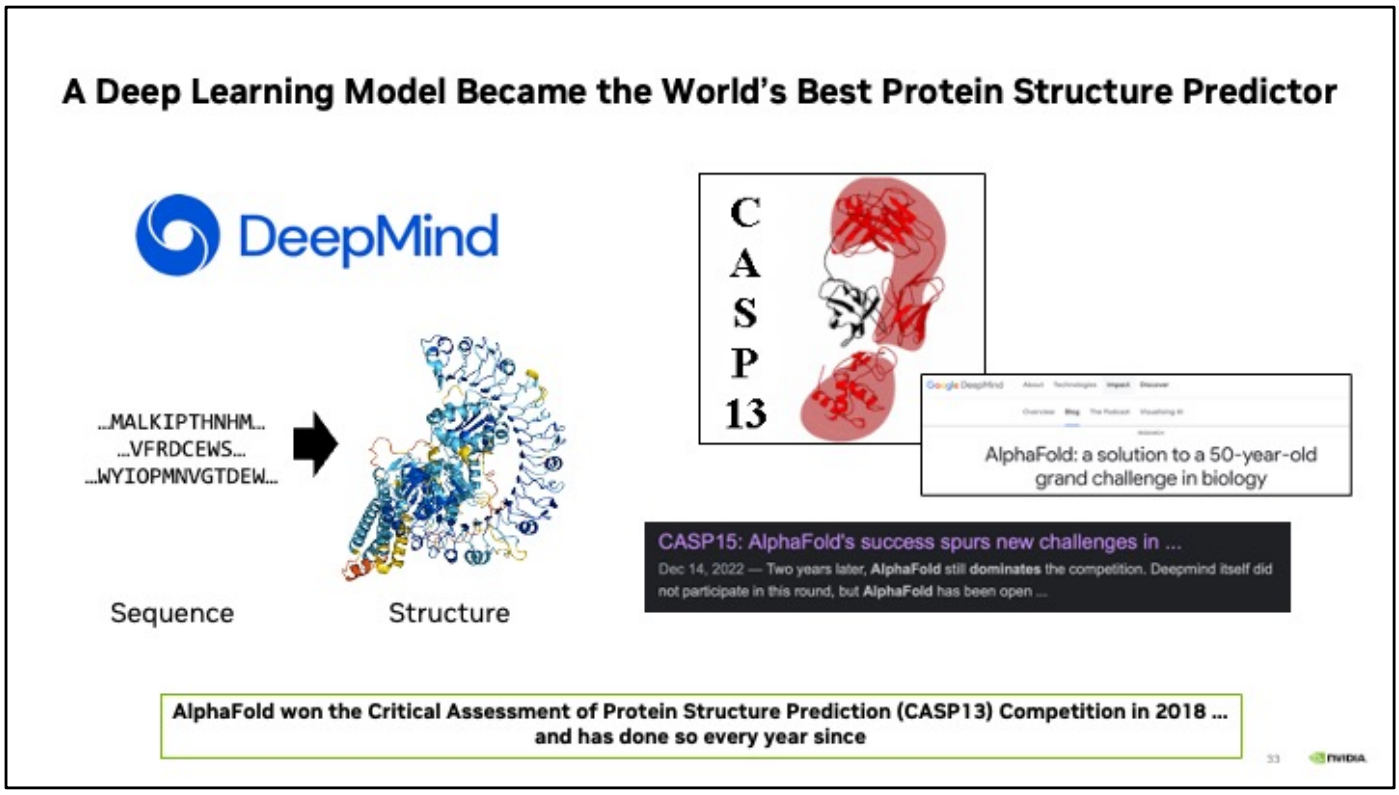
AlexNet didn't just win; it dominated. AlexNet was unlike the other competitors. This new model demonstrated unparalleled performance on the largest image dataset of the time, ImageNet. This event made AlexNet the first widely acknowledged, successful application of deep learning.

Don't miss the bigger picture: Machine learning will have an impact on every industry.

The story of how I got here has a number of twists and turns. Some by my own design, some by chance. Rewind 10 years ago – finishing postdoctoral research, read about this deep learning achievements



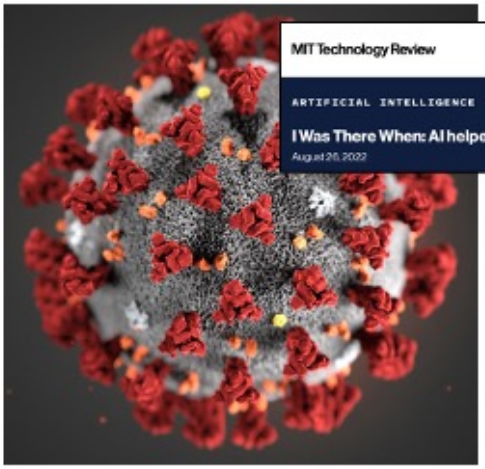
Considering such a change was stressful – wasn't sure if I could stay close to life sciences
Decided to treat the transition like I would anything else new that I needed to learn -- studied
Networked – went to data science conferences
Ended up at NVIDIA



Watershed moment for structural biology

AlphaFold not a panacea for drug design – opportunities for improvement

AI and the Race for a COVID-19 Vaccine

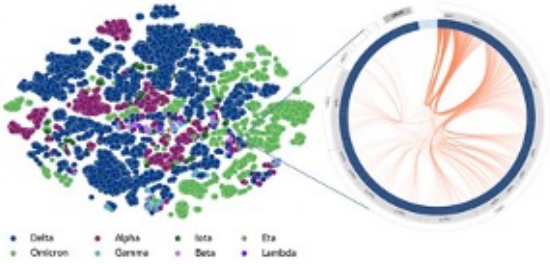


MIT Technology Review

ARTIFICIAL INTELLIGENCE

I Was There When: AI helped create a vaccine

August 26, 2022



Genome-scale language models (GenSLMs) discover distinct evolutionary patterns in SARS-CoV-2

Argonne

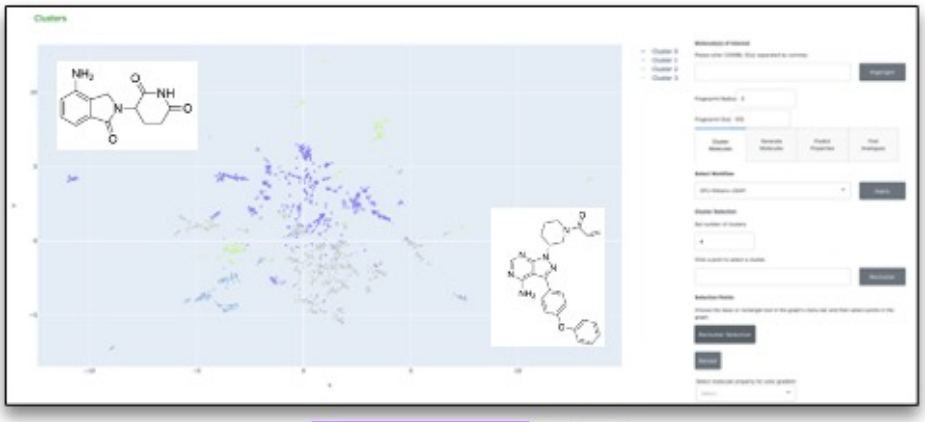
Argonne researchers win Gordon Bell Special Prize for adapting language models to track virus variants

Groundbreaking research focuses on understanding genomic sequences to catch more deadly variants of COVID-19.

Zvyagin, M. *et al.* Biorxiv 2022.10.10.511571 (2022) doi:10.1101/2022.10.10.511571.

NVIDIA

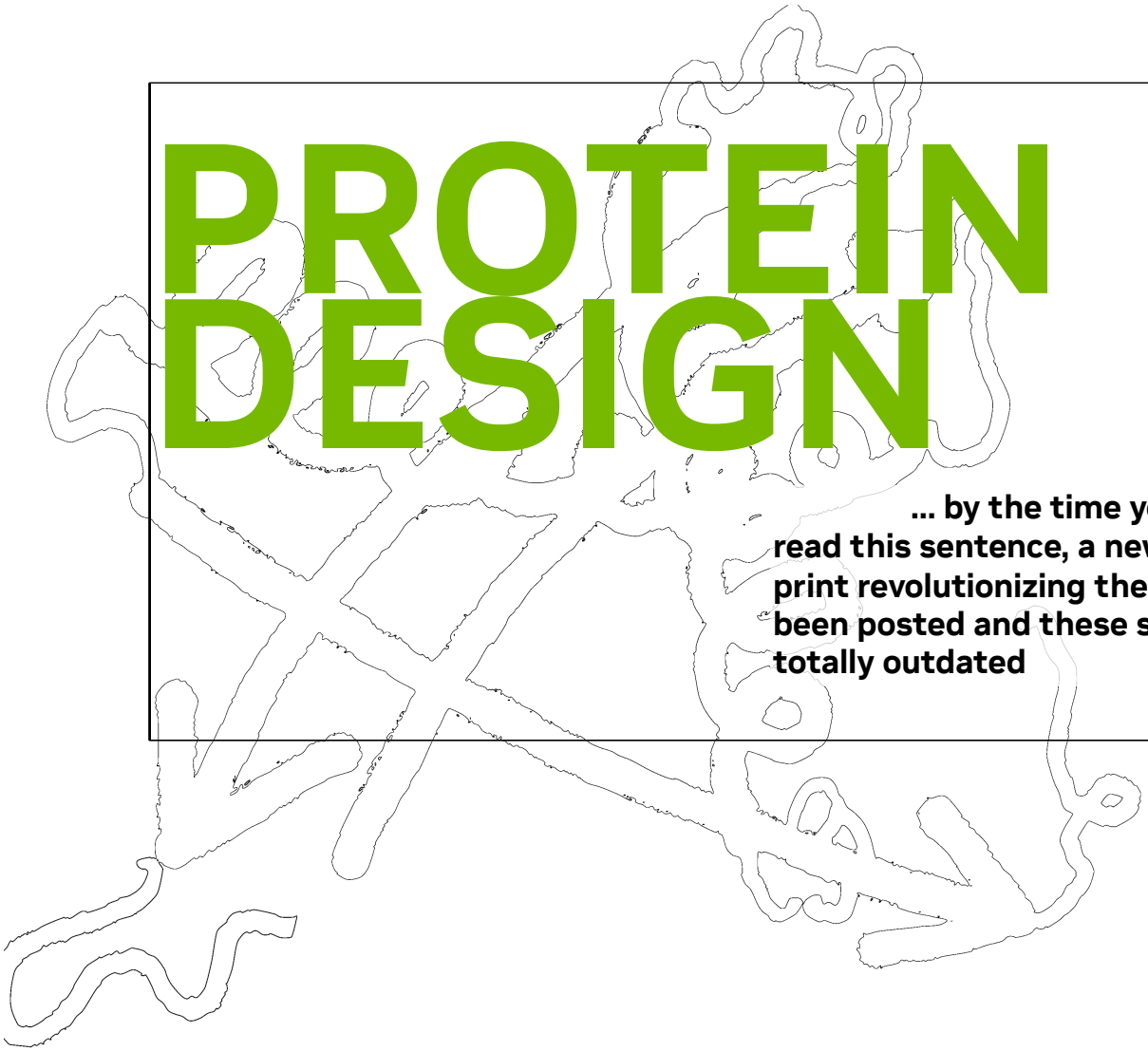
First Effort: Interface for Clustering and Visualization of Small Molecules



dask RAPIDS plotly | Dash

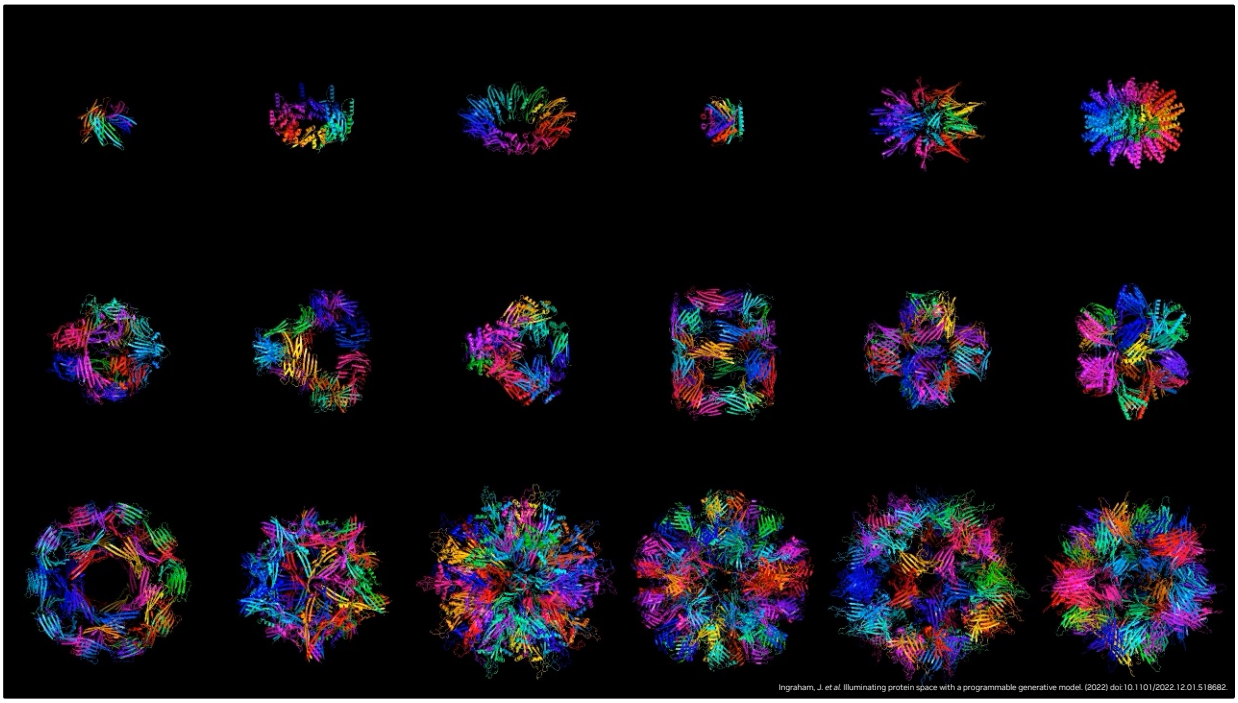
Deep learning is high risk. Ensure the project will succeed if deep learning fails.

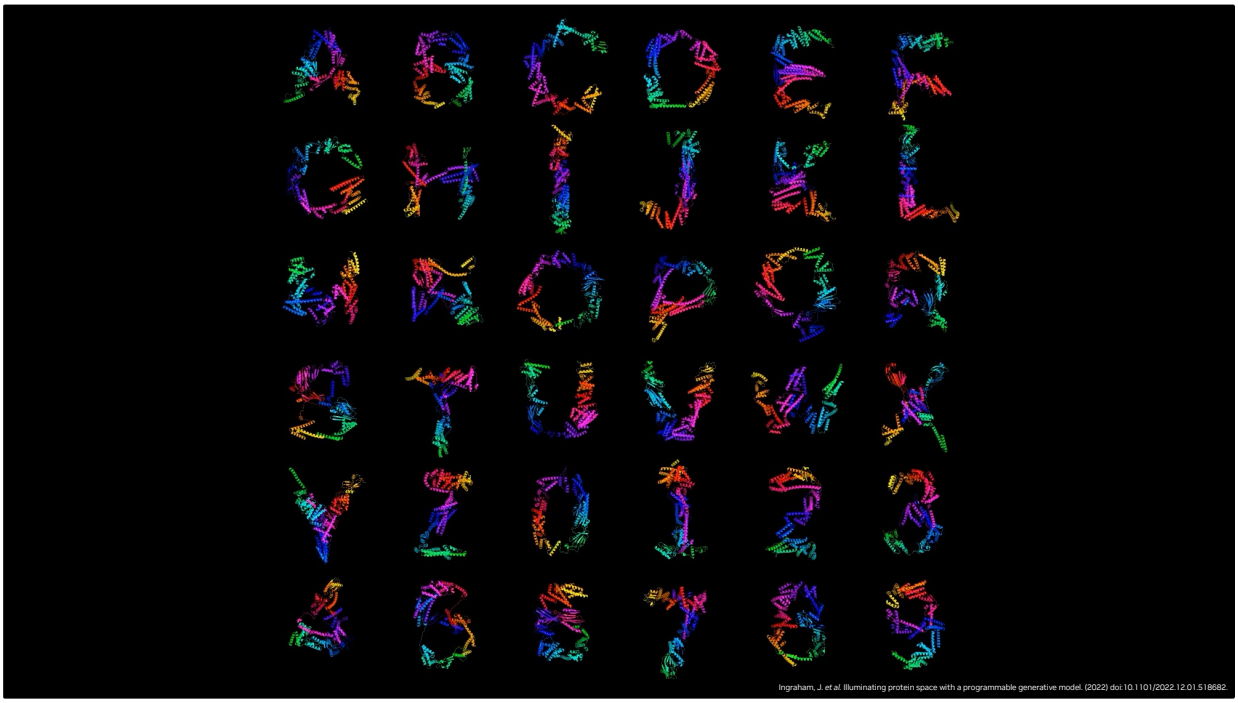
First effort – interface for performing ML on chemical databases and visualizing them. Dask and RAPIDS for ML on GPUs – multi-GPU and multi-node, Interface was built using plotly.
Didn't ship DL model to begin with, interface was useful when we were clustering.



PROTEIN DESIGN

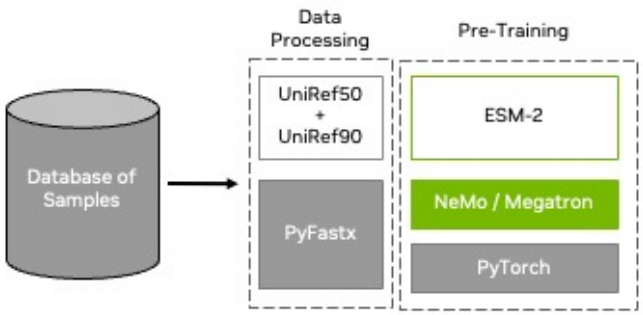
... by the time you've read this sentence, a new pre-print revolutionizing the field has been posted and these slides are totally outdated


Ingraham, J. et al. Illuminating protein space with a programmable generative model. (2022) doi:10.1101/2022.12.01.518682.



Ingraham, J. *et al.* Illuminating protein space with a programmable generative model. (2022) doi:10.1101/2022.12.01.518682.

Developing Deep Learning Models at Scale



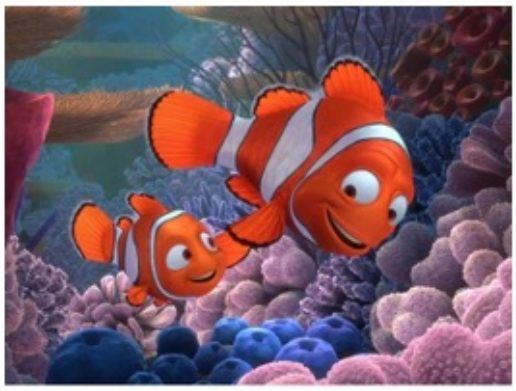
Model Size (Param)	Training Time (Days)	
	512 x V100s	512 x A100s
650M	8	~ 2X acceleration
3B	30	

Successes from calculated risks provide justification for growing a team.

Hypothesis

Es – evulaitonary strategy

Rapid Team Growth and Adventures in Management



Two Engineers

< Two Years

Over Thirty Engineers

Deep learning is hard, but growing and managing a team is the most challenging problem.

Hypothesis

Es – evulaitonary strategy

Conclusions

- BioNeMo is a framework and inference service for developing, training, deploying, and using deep learning models and tools for drug discovery
- MolMIM is a cheminformatics language model trained on SMILES with a structured latent space for molecule design
- Careers are long compared to the pace of machine learning advancement
- Capitalize on new opportunities and enjoy the ride!

BioNeMo Inference Service early access : https://www.nvidia.com/bionemo
BioNeMo Framework general access coming next week!

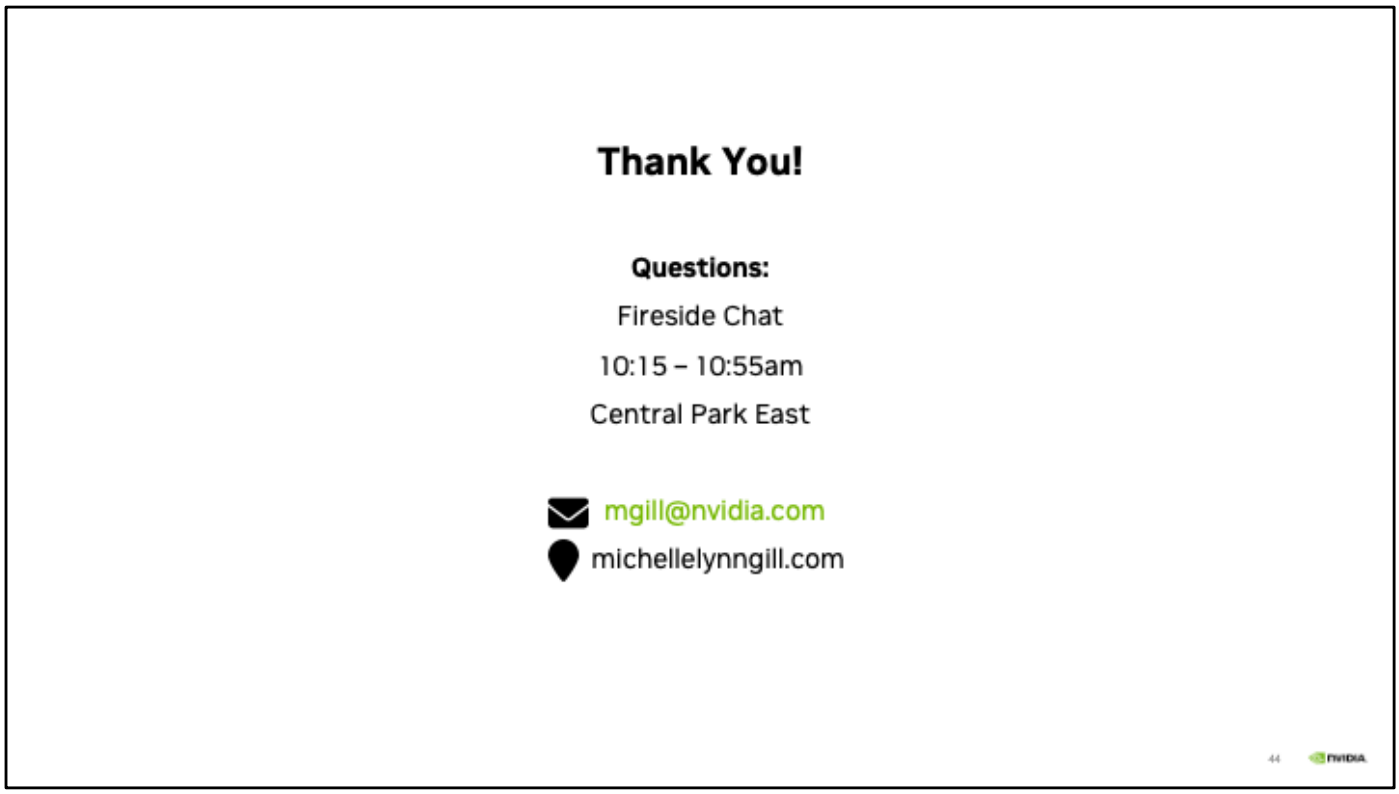
The BioNeMo Team

Johnny Israeli	Gagan Kaushik	Ohad Mosafi
	George Armstrong	Pablo Ribalta
Alireza Moradzadeh	Guoqing Zhou	Rajesh Ilango
Arkadiusz Nowaczynski	Han-Yi Chou	Sara Rabhi
Camir Ricketts	Jasleen Grewal	Simon Chu
Danny Reidenbach	Kevin Boyd	Srimukh Veccham
Dejun Lin	Maria Korshunova	Steven Kothen-Hill
Dorota Toczydlowska	Mario Geiger	Tomasz Grzegorzek
Emine Kucukbenli	Marta Stepniewska-Dziubinska	Timur Rvachov
Eric Dawson	Micha Livne	Yuxing Peng
Farhad Ramezanghorbani	Neha Tadimeti	Zachary McClure

43 NVIDIA

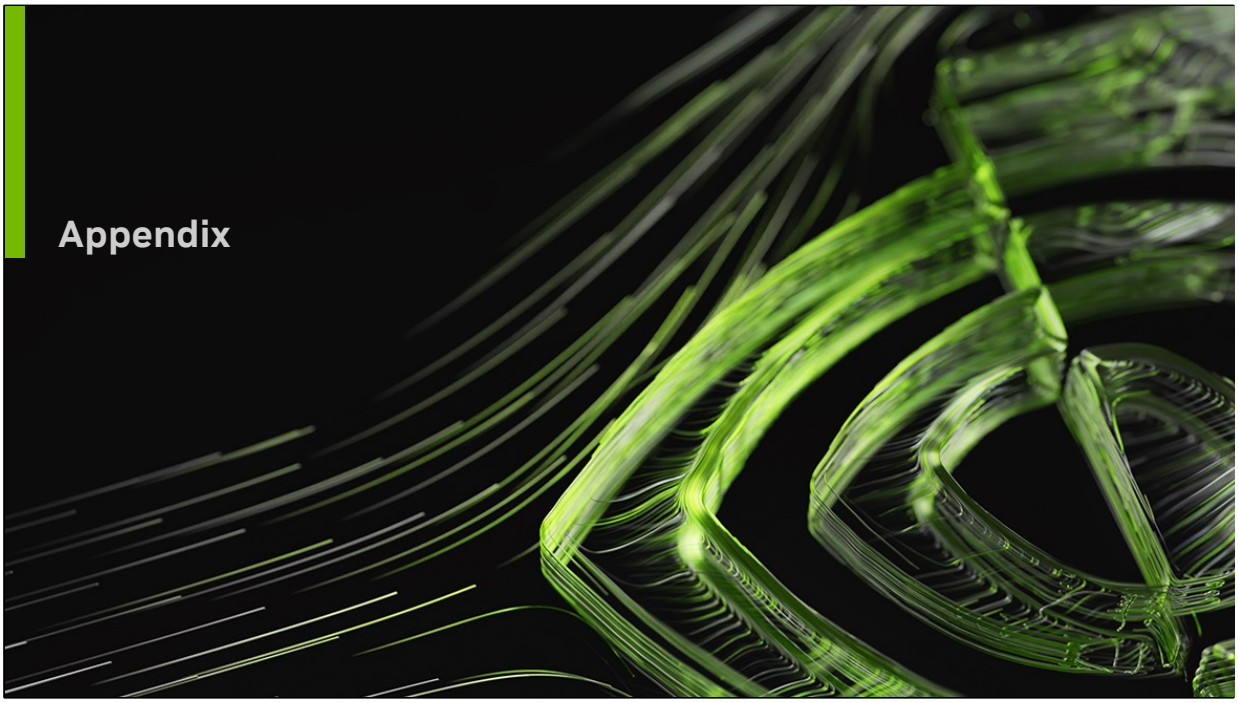
These are the individuals who work on BioNeMo and related work. There have also been a number outside the team who have helped at various points

18 months ago, BioNeMo team was my manager (Johnny Israeli), and engineer (Rajesh), and me.



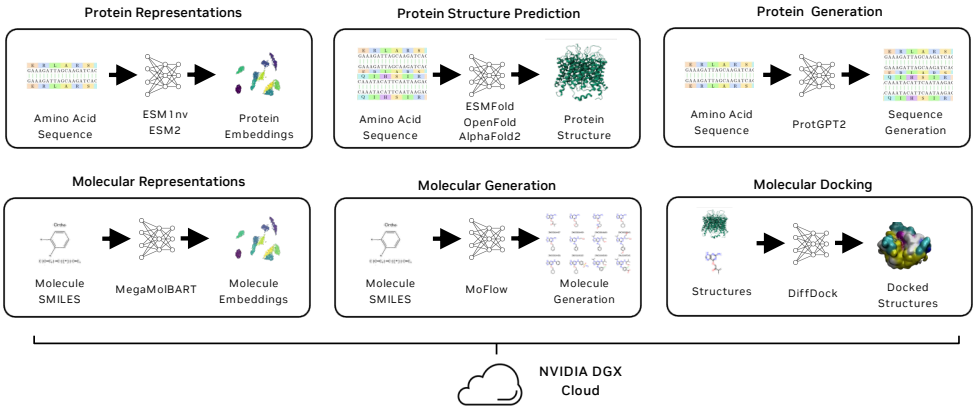
These are the individuals who work on BioNeMo and related work. There have also been a number outside the team who have helped at various points

18 months ago, BioNeMo team was my manager (Johnny Israeli), and engineer (Rajesh), and me.



Clara is the healthcare platform for AI and accelerated compute. Clara for Drug discovery includes GPU- accelerated and AI containers, platforms and tools.

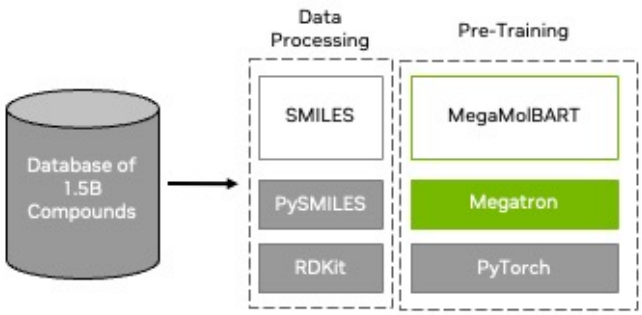
Nine Models in Inference Service for Drug Discovery Applications



Inference service has nine models , ESM 1 and ESM2 models for protein representations. For protein structure prediction, ESMFold, OpenFold, AlphaFold. Protein generation with ProtGPT2.

MegaMolBART can be used for molecular representations, MoFlow for molecular generation, and DiffDock for blind docking

Deep Learning at Scale



Attention Heads	Layers	Hidden Size	Feed Forward	Parameters
8	4	256	1024	10M
8	6	512	2048	45M
16	8	1024	4096	230M

47 NVIDIA

Hypothesis

Es – evulaitonary strategy

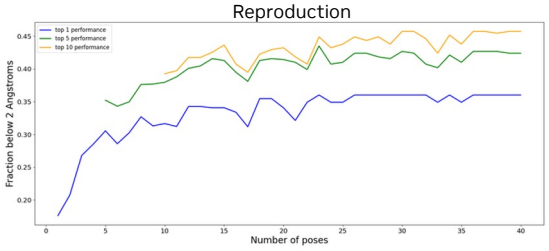
Life Cycle of a BioNeMo Model in the Inference Service

- Model checkpoints are accelerated using a variety of NVIDIA tools – standard tricks to custom CUDA kernels
- All quantitative and qualitative results are reproduced
- For DiffDock, the RMSD metrics were reproduced under a variety of different conditions

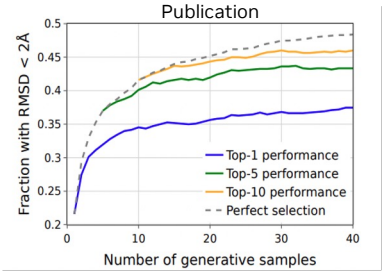
Method	Holo crystal proteins Top-1 RMSD %<2	Top-1 RMSD Med.	Top-5 RMSD %<2	Top-5 RMSD Med.
GNINA	22.9	7.7	32.9	4.5
SMINA	18.7	7.1	29.3	4.6
GLIDE	21.8	9.3		
EQUIBIND	5.5	6.2	-	-
TANKBIND	20.4	4.0	24.5	3.4
P2RANK+SMINA	20.4	6.9	33.2	4.4
P2RANK+GNINA	28.8	5.5	38.3	3.4
EQUIBIND+SMINA	23.2	6.5	38.6	3.4
EQUIBIND+GNINA	28.8	4.9	39.1	3.1
DIFFDOCK (40)	**38.2**	**3.3**	**44.7**	**2.40**

NV Trial #1	38.0
NV Trial #2	35.0
NV Trial #3	38.6
NV Trial #4	39.1
NV Trial #5	38.6

Reproduction



Publication



Like to cover how a model makes its way into the service, use DiffDock as an example. Model checkpoints are accelerated – these can be standard tricks or custom solutions, depends on model, impact of acceleration. Will cover this in last section.

For DiffDock, reproduced top-N RMSD under different conditions including different random seeds, and also with different numbers of generated poses

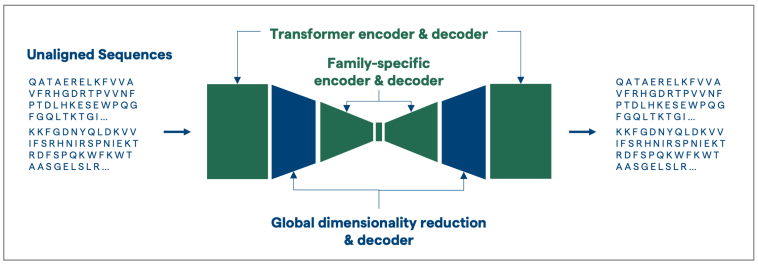
Proteins Generated from Evozyne's ProT-VAE Models

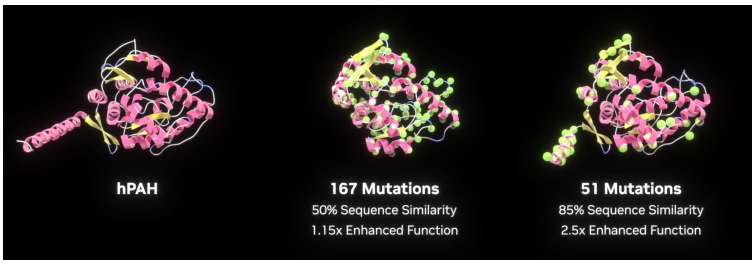
ProT-VAE: Protein Transformer Variational AutoEncoder for Functional Protein Design

Emre Sevgen[1†], Joshua Moller[1†], Adrian Lange[1], John Parker[1], Sean Quigley[1], Jeff Mayer[1], Poonam Srivastava[1], Sitaram Gayatri[1], David Hosfield[1], Maria Korshunova[2], Micha Livne[2], Michelle Gill[2], Rama Ranganathan[1], Anthony B. Costa[2*] and Andrew L. Ferguson[1*]

[1]Evozyne, Inc., 2430 N Halsted Street, Chicago, 60614, IL, USA.
[2]NVIDIA, 2788 San Tomas Expressway, Santa Clara, 95051, CA, USA.

*Corresponding author(s). E-mail(s): acosta@nvidia.com; andrew.ferguson@evozyne.com;
†These authors contributed equally to this work.

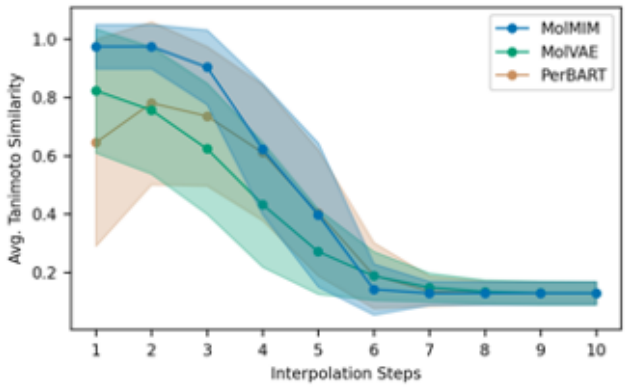




There is already a publication using BioNeMo framework for protein design. Evozyne used it to develop a VAE model for protein design., called ProtT-VAE.

Used to generate Src homology 3 (not shown(and phenylalinie hydroxylase enzymes (Phe—>Tyr). Two rounds of mutations, first 50% sequence similarity (to WT) and 1.15X function, second 85% similarity and 2.5X function.

Probing Latent Structure by Molecule Interpolation



- Pairwise interpolations performed at ten evenly spaced steps for 1,000 ZINC15 molecules
- Average Tanimoto similarity to first molecule was calculated at each step
- Molecules sampled from Perceiver BART and MolVAE have reduced similarity to start and a large degree of variability at early interpolation steps
- Molecules sampled from MolMIM are similar and have the smallest variance at early steps

Probed latent space quantitatively based on pairwise interpolation for 1k molecules. Made 10 evenly spaced steps.

PerBART and MoLVAE have poor starting similarity (reconstruction) as demonstrated by low tanimoto similarity. Also have large variance, MolMIM has high similarity and reduced variance

Remind that MolMIM not trained with similarity information

MolMIM – Performance on Seed Based Molecule Sampling

- Randomly sampled ten molecules for each of 20k molecules from test split
- Effective novelty is percentage of molecules that are valid, unique, not identical to seed, and novel
- Sampling radius empirically determined to maximize effective novelty
- CDDD used as baseline model – trained with molecular property loss
- Perceiver BART sampling speed improved relative to MegaMolBART
- MolVAE and MolMIM show significant improvements in validity and effective novelty

Model	Latent Dim	Validity (%)	Uniqueness (%)	Novelty (%)	Effective Novelty (%)	Test Runtime
MegaMolBART	Variable	75.0	84.8	94.2	51.1	8.7 hours
Perceiver BART	2048	71.8	94.9	94.6	59.1	38 min
MolVAE	2048	95.7	**100.0**	98.1	93.9	64 min
MolMIM	512	**98.7**	**100.0**	95.5	**94.2**	30 min
CDDD	512	84.5	98.9	**99.5**	82.2	12 hours[1]

[1]CDDD decoding speed limited by batch size.

R. Winter, *et. al.*, Chemical Science. 10, 1692–1701 (2019). 51

Randomly sampled ten molecules for each of 20K from test set.

Validity is percent of RDKIt-valid molecules, uniqueness based on sampled molecules, novelty is molecules not in training set – similar to Guacamol. Note effective novelty.

CDDD from robin winter – continuous data driven descriptors, trained with molecular property loss

Perceiver BART sampling speed improves because of MMB decoding longer sequences with invalid molecules. Adding loss to organize latent space improves

Single Property Optimization with CMA-ES

Model	QED (%)	Penalized logP	
	$\delta \geq 0.4$	$\delta \geq 0.4$	$\delta \geq 0.6$
AtomG2G	73.6	-	-
HeirG2G	76.9	-	-
DESMILES	77.8	-	-
QMO	92.8	7.71 ± 5.65	3.73 ± 2.85
MolGrow	-	8.34 ± 6.85	4.06 ± 5.61
GraphAF	-	8.21 ± 6.51	4.98 ± 6.49
GraphDF	-	9.19 ± 6.43	4.51 ± 5.80
CDGS	-	9.56 ± 6.33	5.10 ± 5.80
FaST	-	18.09 ± 8.72	8.98 ± 6.31
MolMIM	**94.6**	**28.45 ± 54.67**	**7.60 ± 23.62**
MolMIM		9.44 ± 4.12†	4.57 ± 3.87†

- Performed optimization of QED or penalized logP with query budget of 50,000 oracle calls per input molecule
- Success is % of molecules with QED ≥ 0.9 or penalized logP improvement while maintaining Tanimoto similarity $\delta \geq \{0.4, 0.6\}$
- MolMIM achieves the highest QED and logP success rates
- Penalized logP results impacted by known exploit where identical functional groups are repeatedly added

Results above solid bar as in B. Chen, X. Fu, R. Barzilay, T. Jaakkola, ArXiv (2021) and S. C. Hoffman, *et al*, Nat Mach Intell. 4, 21–31 (2022)
QED and logP oracles from Therapeutic Data Commons.
†logP improvement limited to ≤ 20

Test controllability of model with CMA-ES. Performed optimization of two properties -- QED / penalized logP. 50K calls per input molecule

QED >= 0.9, logP improvement while limited Tanimoto similarity

Penalized logP is water-octanol partition coefficient penalized by thetic accessibility and ring size

Single Property Optimization with CMA-ES

Model	QED (%)	Penalized logP	
	$\delta \geq 0.4$	$\delta \geq 0.4$	$\delta \geq 0.6$
AtomG2G	73.6	-	-
HeirG2G	76.9	-	-
DESMILES	77.8	-	-
QMO	92.8	7.71 ± 5.65	3.73 ± 2.85
MolGrow	-	8.34 ± 6.85	4.06 ± 5.61
GraphAF	-	8.21 ± 6.51	4.98 ± 6.49
GraphDF	-	9.19 ± 6.43	4.51 ± 5.80
CDGS	-	9.56 ± 6.33	5.10 ± 5.80
FaST	-	18.09 ± 8.72	8.98 ± 6.31
MolMIM	**94.6**	**28.45 ± 54.67**	**7.60 ± 23.62**
MolMIM		9.44 ± 4.12†	4.57 ± 3.87†

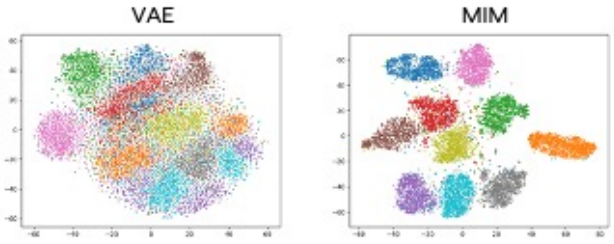
- Performed optimization of QED or penalized logP with query budget of 50,000 oracle calls per input molecule
- Success is % of molecules with QED ≥ 0.9 or penalized logP improvement while maintaining Tanimoto similarity $\delta \geq \{0.4, 0.6\}$
- MolMIM achieves the highest QED and logP success rates
- Penalized logP results impacted by known exploit where identical functional groups are repeatedly added
- Recall: MolMIM trained without chemical properties, activity, or fragment knowledge

Results above solid bar as in B. Chen, X. Fu, R. Barzilay, T. Jaakkola, ArXiv (2021) and S. C. Hoffman, *et al*, Nat Mach Intell. 4, 21–31 (2022)
QED and logP oracles from Therapeutic Data Commons.
†logP improvement limited to ≤ 20

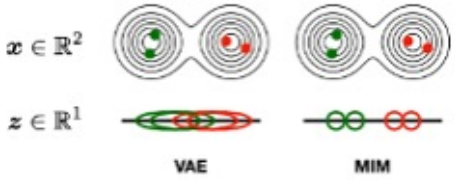
Compare to QMO – CDDD with controlled optimization applied to SMILES, includes activity prediction and tanimoto similiarity as pseudogradient
FaST is a VQ-VAE model trained on fragments with search policy trained on latent space.

A Clustered Latent Space with Mutual Information Machine

- Same architecture as VAE, but loss maximizes mutual information and minimizes marginal entropy
- MIM results in an informative and clustered latent space



$$\begin{aligned}\mathcal{L}_{\text{A-MIM}}(\boldsymbol{\theta}) &= \frac{1}{2}\big(CE\left(\mathcal{M}_S^q(\boldsymbol{x},\boldsymbol{z}),\, q_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\right) \\ &\quad + CE\left(\mathcal{M}_S^q(\boldsymbol{x},\boldsymbol{z}),\, p_{\boldsymbol{\theta}}\left(\boldsymbol{x},\boldsymbol{z}\right)\right)\big) \\ &\geq H_{\mathcal{M}_S^q}(\boldsymbol{x}) + H_{\mathcal{M}_S^q}(\boldsymbol{z}) - I_{\mathcal{M}_S^q}(\boldsymbol{x};\boldsymbol{z})\end{aligned}$$



M. Livne, K. Swersky, D. J. Fleet, ArXiv (2019).

Model	QED (%)	Penalized logP	
	$\delta \geq 0.4$	$\delta \geq 0.4$	$\delta \geq 0.6$
JT-VAE	8.8	1.03 ± 1.39	0.28 ± 0.79
GCPN	9.4	2.49 ± 1.30	0.79 ± 0.63
MolDQN	-	3.37 ± 1.62	1.86 ± 1.21
MMPA	32.9	-	-
VSeq2Seq	58.5	3.37 ± 1.75	2.33 ± 1.17
VJTNN+GAN	60.6	-	-
VJTNN	-	3.55 ± 1.67	2.33 ± 1.24
MoFlow	-	4.71 ± 4.55	2.10 ± 2.86
GA	-	5.93 ± 1.41	3.44 ± 1.09
AtomG2G	73.6	-	-
HeirG2G	76.9	-	-
DESMILES	77.8	-	-
QMO	92.8	7.71 ± 5.65	3.73 ± 2.85
MolGrow	-	8.34 ± 6.85	4.06 ± 5.61
GraphAF	-	8.21 ± 6.51	4.98 ± 6.49
GraphDF	-	9.19 ± 6.43	4.51 ± 5.80
CDGS	-	9.56 ± 6.33	5.10 ± 5.80
FaST	-	18.09 ± 8.72	8.98 ± 6.31
MolMIM	**94.6**	**28.45 ± 54.67**	**7.60 ± 23.62**
MolMIM		9.44 ± 4.12†	4.57 ± 3.87†

e Property Optimization

- Repeated QED and penalized logP optimization with query budget of 50,000 oracle calls per input molecule
- Success is % of molecules with QED ≥ 0.9 or penalized logP improvement while maintaining Tanimoto similarity $\delta \geq \{0.4, 0.6\}$
- MolMIM achieves the highest QED and logP success rates
- Penalized logP results impacted by known exploit where identical functional groups are repeatedly added
- MolMIM results were repeated with logP improvement limited

Results above solid bar as in B. Chen, X. Fu, R. Barzilay, T. Jaakkola, ArXiv (2021) and S. C. Hoffman, *et al*, Nat Mach Intell. 4, 21–31 (2022).
†logP improvement limited to ≤ 20

Perspective on BioNeMo

- Models have a finite lifespan, the value is in the learnings
- Developing and productizing internal research is useful for driving improvements to the platform
- Scalability and acceleration are differentiating factors
- Surface NVIDIA technologies, and use bottlenecks to drive the development software and hardware

I'd like to set the tone for this talk – this is probably the most important slide so if you only pay attention to this one, you're good.

I'll talk a lot about different models today. But models are far from the point . No disrespect intended to the developers of these models. But SOTA doesn't last long in a field that moves as quickly as this one. While we believe the models we've selected will prove useful, the long term value is what can be learned from them from and R&D standpoint.

Our team's own R&D is an important mechanism to drive improvements for the platform. There's nothing like dogfooding.

Scalability and acceleration are key to an accelerated computing and should be maximized. Finally, BioNeMo should surface NVIDIA technologies in ways that improve performance and benefit users. Bottlenecks and challenges that we encounter should be used to drive improvements to both software and hardware