THE BIOPHARMACEUTICAL RESEARCH AND DEVELOPMENT PROCESS

POTENTIAL NEW MEDICINES

PHASE I  PHASE II  PHASE III  PHASE IV

IND SUBMITTED  NDA/BLA SUBMITTED  FDA APPROVAL

TENS  HUNDREDS  THOUSANDS

NUMBER OF VOLUNTEERS

1 FDA-APPROVED MEDICINE

$2.6 BILLION

Source: Pharmaceutical Research and Manufacturers of America (http://phrma.org)
The Biopharmaceutical Research and Development Process

<table>
<thead>
<tr>
<th>BASIC RESEARCH</th>
<th>DRUG DISCOVERY</th>
<th>PRE-CLINICAL</th>
<th>CLINICAL TRIALS</th>
<th>FDA REVIEW</th>
<th>POST-APPROVAL RESEARCH &amp; MONITORING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PHASE I</td>
<td>PHASE II</td>
<td>PHASE III</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PHASE IV</td>
</tr>
</tbody>
</table>

**Potential New Medicines**

Number of Volunteers:
- TENS
- HUNDREDS
- THOUSANDS

**Source:** Pharmaceutical Research and Manufacturers of America (http://phrma.org)

*If you stop failing so often you massively reduce the cost of drug development.*

— Sir Andrew Witty, CEO, GlaxoSmithKline

$2.6 BILION

1 FDA-APPROVED MEDICINE
THE BIOPHARMACEUTICAL RESEARCH AND DEVELOPMENT PROCESS

BASIS RESEARCH

PHASE I

IND SUBMITTED

TENS

PRE-Clinical

PHASE II

HUNDREDS

CLINICAL TRIALS

PHASE III

THOUSANDS

FDA REVIEW

FDA APPROVAL

POST-APPROVAL RESEARCH & MONITORING

PHASE IV

NUMBER OF VOLUNTEERS

POTENTIAL NEW MEDICINES

If you stop failing so often you massively reduce the cost of drug development.

— Sir Andrew Witty
CEO, GlaxoSmithKline

1 FDA-APPROVED MEDICINE

$2.6 BILLION

Source: Pharmaceutical Research and Manufacturers of America (http://phrma.org)
1. Does the compound do what you want it to?
2. Does the compound **not** do what you **don’t** want it to?
3. Is what you want it to do the right thing?
Drug Discovery

Omics

Target Identification
Screening

Compounds

Hits

Leads

Clinical Candidates

Lead Identification

Lead Optimization

Cost
Computational Drug Discovery

Omics
- Target Identification

Virtual
- Screening

Modeling
- Lead Optimization

Compounds

Hits

Leads

Clinical Candidates

Cost
Structure Based Drug Design

- Pose Prediction
- Binding Discrimination
- Affinity Prediction

Virtual Screening

Lead Optimization
Structure Based Drug Design

- Pose Prediction
- Binding Discrimination
- Affinity Prediction

Virtual Screening

Lead Optimization
Drug Discovery Funnel

http://pharmit.csb.pitt.edu
Drug Discovery Funnel

\[ \text{gauss}_1(d) = \omega_{\text{gauss}_1} e^{-\left(\frac{d-0.5}{0.5}\right)^2} \]
\[ \text{gauss}_2(d) = \omega_{\text{gauss}_2} e^{-\left(\frac{d-3}{0.5}\right)^2} \]
\[ \text{repulsion}(d) = \begin{cases} 
\omega_{\text{repulsion}} d^2 & d < 0 \\
0 & d \geq 0 
\end{cases} \]

\[ \text{hydrophobic}(d) = \begin{cases} 
\omega_{\text{hydrophobic}} & d < 0.5 \\
0 & d > 1.5 \\
\omega_{\text{hydrophobic}}(1.5 - d) & \text{otherwise} 
\end{cases} \]

\[ \text{hbond}(d) = \begin{cases} 
\omega_{\text{hbond}} & d < -0.7 \\
0 & d > 0 \\
\omega_{\text{hbond}}\left(-\frac{1}{d} \right) & \text{otherwise} 
\end{cases} \]

Protein-Ligand Scoring

Model

- Pose Prediction
- Binding Discrimination
- Affinity Prediction
The universal approximation theorem states that, under reasonable assumptions, a feedforward neural network with a finite number of nodes can approximate any continuous function to within a given error over a bounded input domain.
Deep Learning
Deep Learning
Deep Learning

\[
\delta^l = \left((w^{l+1})^T \delta^{l+1}\right) \odot \sigma'(z^l)
\]

\[
\frac{\partial L}{\partial w^l_{jk}} = \alpha^{l-1}_k \delta^l_j \text{ and } \frac{\partial L}{\partial b^l_j} = \delta^l_j
\]
Convolutional Neural Networks

Convolution

Feature Maps

Fully Connected

Traditional NN

Dog: 0.99
Cat: 0.02

Convolution

weight 1
weight 2
weight 3

Fully-connected

weight 1
weight 2
weight 3
weight 4
weight 5
Convolutional Filters
Protein-Ligand Representation

(R,G,B) pixel
Protein-Ligand Representation

(R,G,B) pixel → (Carbon, Nitrogen, Oxygen,...) voxel

The only parameters for this representation are the choice of grid resolution, atom density, and atom types.
Why Grids?

Cons

- coordinate frame dependent
- pairwise interactions not explicit

Pros

- clear spatial relationships
- amazingly parallel
- easy to interpret
Data Augmentation

- **Pose AUC**
  - Iteration range: 0 to 2000
  - Graph showing change over iteration

- **Affinity RMSE**
  - Iteration range: 0 to 2000
  - Graph showing change over iteration
  - Colors: Not Augmented (orange), Augmented (green)
Data Augmentation

![Graphs showing Pose AUC and Affinity RMSE over iterations for Not Augmented, Augmented, Test, and Train datasets.](image)

- **Pose AUC**
  - Y-axis: 0 to 1.0
  - X-axis: 0 to 2000 iterations
- **Affinity RMSE**
  - Y-axis: 0 to 4.0
  - X-axis: 0 to 2000 iterations

The graphs illustrate the performance of data augmentation in improving the AUC and RMSE as the iteration count increases.
Pose Results

Redocked Pose

Percent Targets with Low RMSD Top Pose

Default2017  Default2018  HiRes Affinity  HiRes Pose  Vina
Pose Results

Crossdocketed Pose

Percent Targets with Low RMSD Top Pose

- Default2017
- Default2018
- HiRes Affinity
- HiRes Pose
- Vina
Affinity Results

HiRes Affinity

Default 2018

Vina

Spearman = 0.598, RMSE = 1.714

Spearman = 0.570, RMSE = 1.686

Spearman = 0.473, RMSE = 1.887
Affinity Results

HiRes Affinity

Spearman = 0.598, RMSE = 1.714

Default 2018

Spearman Correlation

Clustered

Random

Vina

Spearman = 0.473, RMSE = 1.887
But what is it learning?

- AliphaticCarbon
- NitrogenDonor
- OxygenDonorAcceptor
- OxygenAcceptor
Beyond Scoring

- 2x2x2 Max Pooling
- 3x3x3 Convolution
- 2x2x2 Max Pooling
- 3x3x3 Convolution
- 2x2x2 Max Pooling
- 3x3x3 Convolution

48x48x48x35 → 24x24x24x35 → 24x24x24x32 → 12x12x12x32 → 12x12x12x64 → 6x6x6x64 → 6x6x6x128

- Fully Connected
- Softmax+Logistic Loss
- Affinity
- Pose Score
Beyond Scoring

2x2x2 Max Pooling

3x3x3 Convolution

2x2x2 Max Pooling

3x3x3 Convolution

2x2x2 Max Pooling

3x3x3 Convolution

2x2x2 Max Pooling

3x3x3 Convolution

2x2x2 Max Pooling

3x3x3 Convolution

Fully Connected

Fully Connected

Softmax+Logistic Loss

Pseudo-Huber Loss

Affinity

Pose Score
Beyond Scoring

2x2x2 Max Pooling

3x3x3 Convolution

2x2x2 Max Pooling

3x3x3 Convolution

2x2x2 Max Pooling

3x3x3 Convolution

2x2x2 Max Pooling

3x3x3 Convolution

Fully Connected

Fully Connected

Fully Connected

Fully Connected

Pseudo-Huber Loss

Softmax+Logistic Loss

Affinity

Pose

Score

\[
\frac{\partial L}{\partial w_{jk}^{l}} = (\delta_{l}^{T} \delta_{l}^{T+1}) a_{j}^{T-1} \delta_{j}^{T} \quad \text{and} \quad \frac{\partial L}{\partial b_{j}^{T}} = \delta_{j}^{T}
\]
Beyond Scoring

https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html
Beyond Scoring

More Oxygen Here

Less Oxygen Here

\[ \frac{\partial L}{\partial A} = \sum_{i \in G_A} \frac{\partial L}{\partial G_i} \frac{\partial G_i}{\partial D} \frac{\partial D}{\partial A} \]
Beyond Scoring

More Oxygen Here

Less Oxygen Here

\[ \frac{\partial L}{\partial A} = \sum_{i \in G_A} \frac{\partial L}{\partial G_i} \frac{\partial G_i}{\partial D} \frac{\partial D}{\partial A} \]
Minimizing Low RMSD Poses

![Graph showing RMSD Change vs # Poses with two categories: Best and First Minimization. The graph indicates a decrease in RMSD Change with an increase in # Poses, with 'better' and 'worse' labels indicating the direction of change.]
Iterative Refinement

![Graph showing Iterative Refinement](image-url)
Iterative Refinement

RMSD Change

-4 -3 -2 -1 0 1 2 3 4

# Poses

Best
First Minimization
Second Iteration
Iterative Refinement

RMSD Change

# Poses

-4 -3 -2 -1 0 1 2 3 4

Best
First Minimization
Second Iteration
Third Iteration
Generative Modeling
Discriminative Model

Features $X$ $\rightarrow$ Prediction $y$
Generative Model

Features $X$
Generative Model

Features $\mathbf{X}$
Generative Model

$y \rightarrow X$
Generative Adversarial Networks

True Examples

Generator

Discriminator

Loss

Is this a real dog picture?
Generative Adversarial Networks

True Examples

Generator

Discriminator

Loss

Is this a real dog picture?
Generative Adversarial Networks

https://arxiv.org › stat
by IJ Goodfellow - 2014 - Cited by 4339 - Related articles
Jun 10, 2014 - Submission history. From: Ian Goodfellow [view email] [v1] Tue, 10 Jun 2014 18:58:17 GMT (1257kb,D). Which authors of this paper are ...
Generative Adversarial Networks

[See images of generated faces]

Generative Adversarial Networks

https://arxiv.org/abs/1406.2661

by IJ Goodfellow - 2014 - Cited by 4339 - Related articles


GMT (1257kb,D). Which authors of this paper are...
PROGRESSIVE GROWING OF GANs FOR IMPROVED QUALITY, STABILITY, AND VARIATION

Tero Karras
NVIDIA

Timo Aila
NVIDIA

Samuli Laine
NVIDIA

Jaakko Lehtinen
NVIDIA
Aalto University

https://youtu.be/G06dEcZ-QTg
PROGRESSIVE GROWING OF GANs FOR IMPROVED QUALITY, STABILITY, AND VARIATION

Tero Karras  
NVIDIA

Timo Aila  
NVIDIA

Samuli Laine  
NVIDIA

Jaakko Lehtinen  
NVIDIA
Aalto University

https://youtu.be/G06dEcZ-QTg
Generative Models

Generative models approximate a data distribution directly. They can map samples from one distribution (noise or input data) to realistic samples from an output distribution of interest.

noise sample

Generator

generated receptor & ligand grid
Autoencoding

Encoder

Generator

Latent Space

L2 Loss
Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules

Rafael Gómez-Bombarelli†#, Jennifer N. Wei‡#, David Duvenaud†#, José Miguel Hernández-Lobato§#, Benjamín Sánchez-Lengeling†, Dennis Sheberla‡, Jorge Aguilera-Iparraguirre†, Timothy D. Hirzel†, Ryan P. Adams¶, and Alán Aspuru-Guzik††
Context Encoding

http://people.eecs.berkeley.edu/~pathak/context_encoder/
Context Encoding

receptor grid

Generator

generated ligand grid
Receptor-Conditional Ligand-Variational Model
Receptor-Conditional Ligand-Variational Model

[Diagram showing a model with input and output nodes, labeled with 'L2 loss' and 'GAN loss'.]
Receptor-Conditional Ligand-Variational Model

GAN loss

Discriminator
n_levels = 3
conv_per_level = 3
n_filters = 32
width_factor = 2
n_latent = 1024
Autoencoding Examples
Autoencoding Examples

2AVO
Autoencoding Examples
Autoencoding Examples

4PYX
Autoencoding Examples

1LBF
Autoencoding Examples
Atom Fitting

\[ a^* = \arg \min_{a} \| d - D(a) \|^2_2 + \lambda E(a) \]
Atom Fitting

\[ a^* = \arg \min_a \| d - D(a) \|^2 + \lambda E(a) \]
Conditioning on the Receptor
Conditioning on the Receptor
Interpolating

Two atom toy system

protein 2Å ligand

University of Pittsburgh Computational and Systems Biology
Iterpolating

Two atom toy system

protein ligand 2Å
LALRNN
Removing the third dimension
Chomsky Hierarchy

http://www.cs.appstate.edu/~dap/classes/2490/chapter11print.html
Grammars

Balanced Parentheses

S → ε
S → (S)
S → SS

Palindromes

S → ε
S → aSa
S → bSb

Arithmetic

E ::= id
| num
| E + E
| E * E
| ( E )

(aa)
(babbab)
(abbaabba)

3 + 4 * 5
(3 + 4) * 5
<table>
<thead>
<tr>
<th>Section</th>
<th>Formal Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>ATOMS</td>
</tr>
<tr>
<td>3.1.5</td>
<td>ORGANIC SUBSET ATOMS</td>
</tr>
<tr>
<td>3.5</td>
<td>BRACKET ATOMS</td>
</tr>
<tr>
<td>3.1.3</td>
<td>CHIRALITY</td>
</tr>
<tr>
<td>3.1.2</td>
<td>HYDROGENS</td>
</tr>
<tr>
<td>3.1.7</td>
<td>ATOM CLASS</td>
</tr>
<tr>
<td>3.2, 3.9</td>
<td>BONDS AND CHAINS</td>
</tr>
<tr>
<td>3.4</td>
<td>RINGBOND</td>
</tr>
<tr>
<td>3.3</td>
<td>BRANCHED_ATOM</td>
</tr>
<tr>
<td>3.7</td>
<td>DOT</td>
</tr>
<tr>
<td>3.10</td>
<td>SMILES STRINGS</td>
</tr>
</tbody>
</table>

**SMILES**

\[
\text{NH}_2\]

clccccclN
Push Down Automata

Balanced Parentheses

If input is ), what’s on stack top doesn’t matter and a ( is pushed to stack.

If input is ) and ( is on stack top, then ( is popped and nothing is pushed to stack.
Bottom Up Parsing

A PDA can be implemented with a parse table

<table>
<thead>
<tr>
<th>state</th>
<th>action</th>
<th>goto</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$ident$</td>
<td>$+$</td>
</tr>
<tr>
<td>0</td>
<td>$s3$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$a$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$s4$</td>
<td>$r2$</td>
</tr>
<tr>
<td>3</td>
<td>$r3$</td>
<td>$r3$</td>
</tr>
<tr>
<td>4</td>
<td>$s3$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>$r1$</td>
</tr>
</tbody>
</table>

```
while(true)
  s = state on top of stack
  a = current input token
  if(action[s][a] == sN)        shift
    push N
    a = next input token
  else if(action[s][a] == rR)   reduce
    remove rhs of rule R from stack
    X = lhs of rule R
    N = state on top of stack
    push goto[N][X]
  else if(action[s][a] == a)    accept :-)
    return success
  else                      error
    return failure
```

$S \rightarrow E\$
$E \rightarrow T + E$
$E \rightarrow T$
$T \rightarrow identifier$

$x + y\$

states != rules
The NN Part

Implement every state as its own neural network that calculates a function of the input in the context of the parse (encoder) or outputs a syntactically correct string according to the rules of the grammar (generator).
Implement every state as its own neural network that calculates a function of the input in the context of the parse (encoder) or outputs a syntactically correct string according to the rules of the grammar (generator).
Does it work???

\[ \text{NH}_2 \]

\[
\begin{align*}
\text{c1ccccc1N} & \quad \text{GGGAGAAUUGUCCC} \\
& \quad (((((\ldots))) ))
\end{align*}
\]

\[
\begin{align*}
S & \rightarrow . \\
S & \rightarrow () \\
S & \rightarrow (S) \\
S & \rightarrow S(S) \\
S & \rightarrow S.
\end{align*}
\]

\[
\begin{array}{|c|c|c|c|}
\hline
\text{state} & . & ( & | & S \\
\hline
0 & s6 & s1 & | & g5 \\
1 & s6 & s1 & s7 & g4 \\
2 & s6 & s1 & s11 & g3 \\
3 & s10 & s2 & s9 & | \\
4 & s10 & s2 & s8 & | \\
5 & s10 & s2 & | & | \\
6 & \text{Reduce } S \rightarrow . & | & | \\
7 & \text{Reduce } S \rightarrow () & | & | \\
8 & \text{Reduce } S \rightarrow (S) & | & | \\
9 & \text{Reduce } S \rightarrow S(S) & | & | \\
10 & \text{Reduce } S \rightarrow S. & | & | \\
11 & \text{Reduce } S \rightarrow S() & | & | \\
12 & \text{END} & | & | \\
\hline
\end{array}
\]
LALRNN vs GRU

GRU Encoder → Latent Space (100) → GRU/LALRNN Decoder

Graph showing the comparison between GRU RNN and LALRNN for Batch MSE Loss over Training Iteration.
Acknowledgements

Matt Ragoza

Jocelyn Sunseri  Paul Francoeur

Department of Computational and Systems Biology
OUR FIELD HAS BEEN STRUGGLING WITH THIS PROBLEM FOR YEARS.

STRUGGLE NO MORE! I'M HERE TO SOLVE IT WITH ALGORITHMS! DEEP LEARNING!

SIX MONTHS LATER: WOW THIS PROBLEM IS REALLY HARD. YOU DON'T SAY.