Integrating Machine Learning with a Genetic Algorithm for Materials Exploration

Joseph Kern

Agenda

• What genetic algorithms are
• What polyga is
• What you need to design a genetic algorithm
• Fitness functions
• Running polyga
The genetic algorithm is inspired by nature.
For materials design, we use it to solve the inverse design problem.

Targets:
- Glass transition temperature ($T_g$) $> 500$ K,
- Dielectric Constant (Dk) (100 Hz) $> 4$,
- Band gap ($E_g$) $> 5$ eV,
- Charge Injection Barrier (EIB) $> 3$ eV,
- Cohesive Energy Density (COE) $< 80 \, \text{cal/cm}^3$
In polyga, we split parents at the center and recombine them.

DNA (~3300 blocks)

Polymer

ML-Based Property Prediction
For materials design, we need three things before we can run a genetic algorithm...

1. DNA
2. DNA compilation algorithm
3. Property predictors
   1. Features
I won’t go into the full details of DNA collection because it’s domain specific.
The DNA compilation algorithm is also something you need to design.

Property predictors will be models you use to predict the properties of new materials.

\[ \text{[*]NCC(=O)C1CCC(C=O)CC1OS(=O)c1ccc(S(=O)(=O)O[=O])c1} \]

- \( T_g \): 505.95 K
- \( E_g \): 5.18 eV
- \( \varepsilon \): 4.60
Once we have those three things, we can program some criteria for influencing evolution.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>High dielectric breakdown field</td>
<td>Large bandgap</td>
</tr>
<tr>
<td>strength</td>
<td>Large electron charge injection barrier</td>
</tr>
<tr>
<td></td>
<td>Low cohesive energy density</td>
</tr>
<tr>
<td>High thermal stability</td>
<td>High glass transition temperature</td>
</tr>
<tr>
<td>Large energy density per volume</td>
<td>Large dielectric constant</td>
</tr>
</tbody>
</table>

Targets:
- Glass transition temperature \((T_g)\) > 500 K,
- Dielectric Constant \((Dk)\) (100 Hz) > 4,
- Band gap \((E_g)\) > 5 eV,
- Charge Injection Barrier \((\Phi_e)\) > 3 eV,
- Cohesive Energy Density \((e_{coh})\) < \(80 \text{ cal/cm}^3\)

Clamping fitness function

\[
y_{ij} = \begin{cases} 
-\max(x_{ij}, \text{target}_j), & \text{goal}_j = \text{less than target}_j \\
\min(x_{ij}, \text{target}_j), & \text{goal}_j = \text{greater than target}_j
\end{cases}
\]

Fitness Score = \(\sum \minmax(y_{ij})\)

Successful!
Using a clamping fitness function, averages for target properties get closer to their goals with each generation.

\[
\text{score} = 0.2\times t\_g + 0.2\times e\_g + 0.2\times c\_ed + 0.2\times e\_ib + 0.2\times d\_k
\]
A single run can find many successfully polymers, but diversity is usually limited.
We can create multiple islands with different targets and have migration occur between them to try and achieve increase diversity.

Island Fitness Functions:
1. Maximize $T_g$
2. Maximize $E_g$
3. Maximize $\epsilon$
4. MinMaxScaled linear combination of all properties (Clamp)

Using migration, the diversity of the chemical space explored was increased, but the number of successful polymers was not.
At this point, we can run polyga...

• Crossover
• Mutation
  • Mutation rate
  • Selected randomly from DNA
• Selection
  • Based on fitness functions
  • Based on user desire
polyga is pip installable!

https://github.com/Ramprasad-Group/polyga

polyga 1.0.12

pip install polyga

Released: Oct 28, 2021
Polymer Genetic Algorithm

By Joseph D Kern

Generalized genetic algorithm designed for materials discovery.

Launch Tool

Version v1.0.12 - published on 05 Nov 2021

doi:10.21981/VMQG-HM41

cite this

Open source: license | download

View All Supporting Documents

5 users, detailed usage
0 Citation(s)
0 questions (Ask a question)
0 review(s)
0 wish(es) (New Wish)

Watch resource

When watching a resource, you will be notified of changes made. You may stop watching at any time.