Analyzing High Entropy Alloy Data Using Citrination

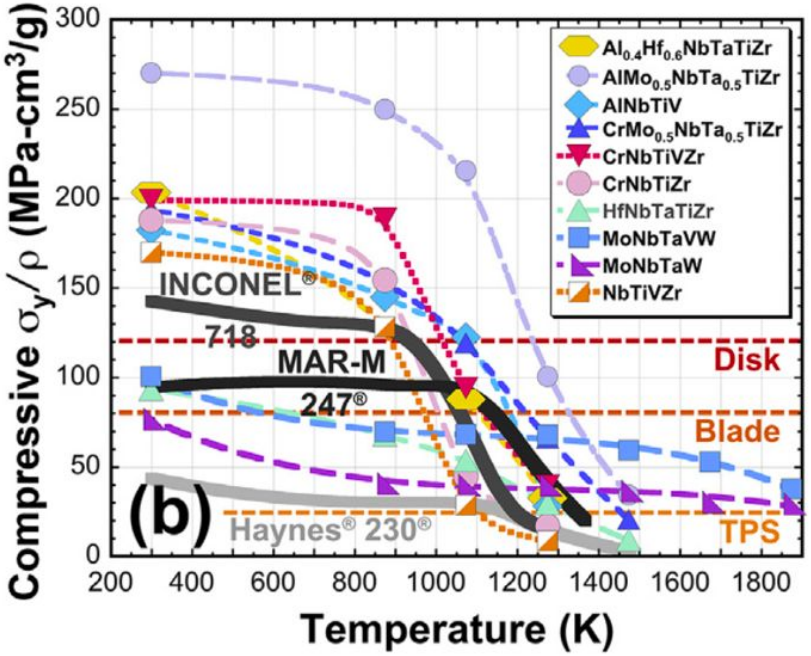
Undergraduate Researchers: Alec Saville & Nick Lipski
Graduate Researcher: Francisco Coury
Research Advisor: Amy Clarke

What is a High Entropy Alloy (HEA)?

- High Entropy Alloys
  - MPEAs, CCAs, etc.
- New class of materials
  - No principal element
- Examples
  - HfNbTaTi - BCC
  - $Al_{0.3}CoCrFeNi$ - FCC

Current Research Focus

- Applications in extreme environments
- Tailorable microstructure and properties
  - Multiple possible compositions
- High-throughput property prediction
  - Accelerated alloy design

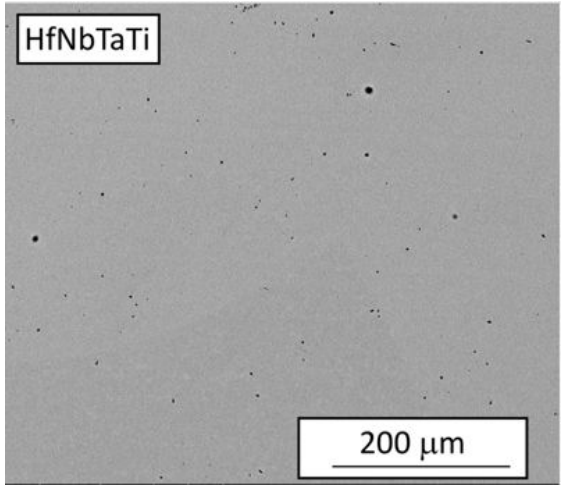


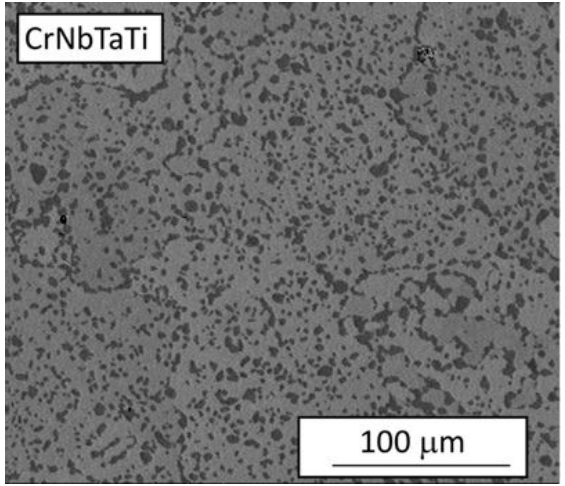
D.B. Miracle and O.N. Senkov, Acta Mater. **122**, 448 (2017).

The Challenge

- Predicting mechanical properties
  - Yield Strength
- Complex phase formation
  - 6 dimensional phase diagrams required
- Thermodynamic simulations are best effort currently
- Possible alternative prediction method
  - Citrination

$$\Omega = \frac{T_m^{mix} \Delta S_{mix}}{|\Delta H_{mix}|} \quad \delta = 100\sqrt{\sum_{i=1}^{n} c_i (1 - r_i / \bar{r})^2}$$

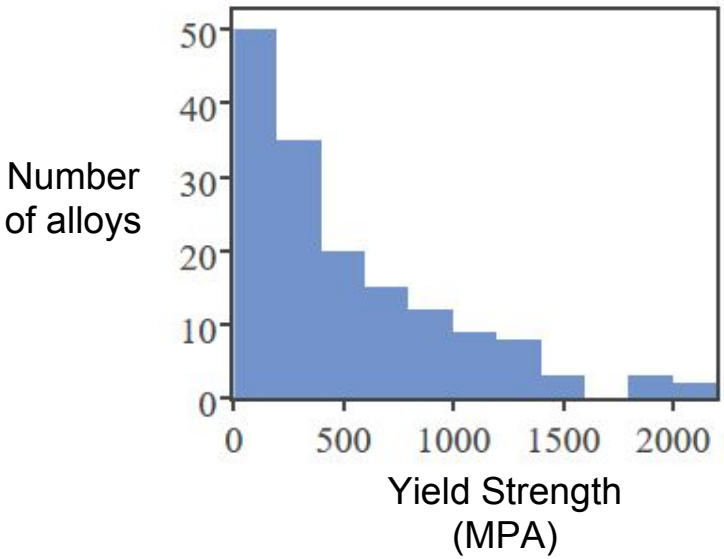




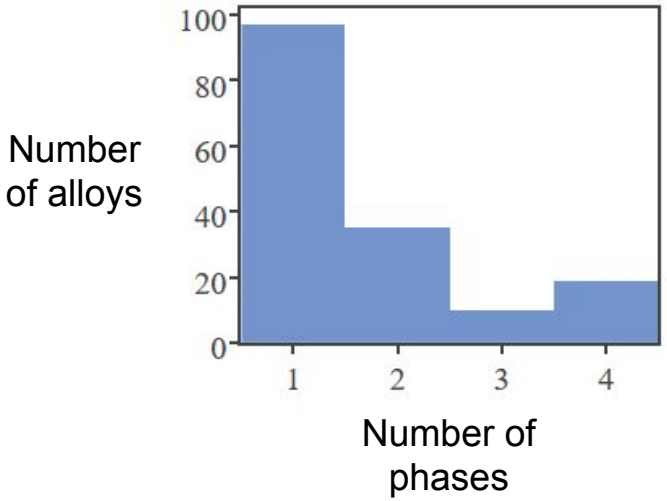
Working with Citrination

- Predicting empirical thermodynamic parameters (Takeuchi 2005)
  - Python script
  - Experimental data
- Creating training data set from published papers
  - Miracle et. al 2017
  - Citrination Data Generator
  - 50 HEAs included so far
- Models for phase equilibria
  - Predicts yield strength, but not elongation

```python
def calc_enthalpy_of_mixing(chemical_formula):

    comp = composition.Composition(chemical_formula)

    elements = []
    for element in comp.elements:
        elements.append(str(element))

    binary_enthalpies = {}

    for c in itertools.combinations(elements, 2):

        binary = "".join(c)
        binary_enthalpy = query_for_property("Enthalpy of mixing", binary)
        binary_enthalpies[binary] = binary_enthalpy

    enthalpy_of_mixing = 0

    for k, v in binary_enthalpies.items():
        binary_comp = composition.Composition(k)
        e1 = binary_comp.elements[0]
        e2 = binary_comp.elements[1]
        contribution = float(v)*float(comp.get_atomic_fraction(e1))*float(comp.get_atomic_fraction(e2))
        enthalpy_of_mixing += contribution

    enthalpy_of_mixing = round(4*enthalpy_of_mixing, 3)

    return(enthalpy_of_mixing)


def calc_entropy_of_mixing(chemical_formula):

    comp = composition.Composition(chemical_formula)
    entropy_of_mixing = 0

    for element in comp:
        ele_fraction = comp.get_atomic_fraction(element)
        contribution = float(ele_fraction)*math.log(float(ele_fraction))
        entropy_of_mixing += contribution
```



Error Metric	Value
Root mean squared error (MPa) (0.0 for a perfect model)	212.38942306025044
Non-dimensional model error (0.0 for a perfect model)	0.4557020204273737
Uncertainty calibration: fraction of actual values within the prediction error bars (0.68 is perfectly calibrated)	0.5732484076433121
Uncertainty calibration: root mean square of standardized errors (1.0 is perfectly calibrated)	1.3144560765766122



Error Metric	Value
Root mean squared error (0.0 for a perfect model)	0.9039119357169356
Non-dimensional model error (0.0 for a perfect model)	0.884651161916204
Uncertainty calibration: fraction of actual values within the prediction error bars (0.68 is perfectly calibrated)	0.5838509316770186
Uncertainty calibration: root mean square of standardized errors (1.0 is perfectly calibrated)	2.2764148216709836

Future with Citrination

- Machine Learning Algorithm still work in progress
  - Many more papers to gain information from
  - 250 HEA's in references
    - 70 extrapolated to date
- Identify new trends or candidates
  - Move away from empirical approach

Our Takeaway

- Machine learning has potential to greatly accelerate alloy design
  - Not just HEAs
- Utilize in future for investigative and predictive analyses
  - Graduate work and industry
- Continue to update and expand HEA database

Thanks Citrination for working with us!