

PRISMS Center

Center for PRedictive Integrated Structural Materials Science

CASM Tutorial

Brian Puchala, Anton Van der Ven

John C. Thomas, Anirudh Natarajan, John Goiri, Min-Hua Chen,
Max Radin, Jonathon Bechtel, Elizabeth Decolvenaere, Anna Belak,
Naga Sri Harsha Gunda, Sanjeev Kolli, Sesha Sai Behara



PRISMS Workshop 2021

Mon. Aug 10, 2021

Website: prisms-center.org



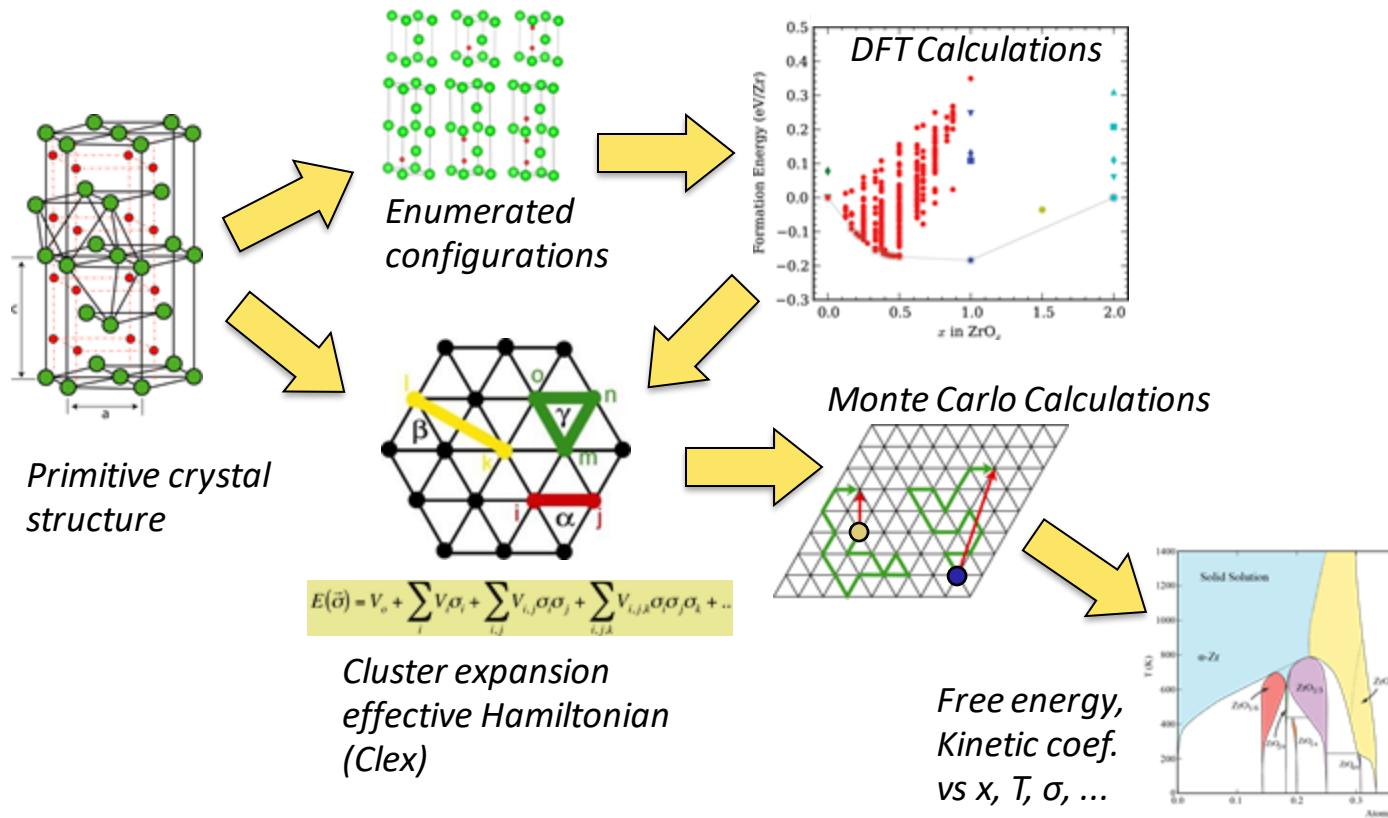
U.S. DEPARTMENT OF
ENERGY

Office of
Science

Center for PRedictive Integrated
Structural Materials Science

PRISMS

CASM Project Workflow



CASM Tutorial Outline

Day 1:

- Cluster expansion formulation
- Defining the "prim"
- Enumeration
- Occupation basis set construction

Day 2:

- DFT Calculation
- Fitting coefficients
- Monte Carlo
- Free energy integration and phase diagram construction

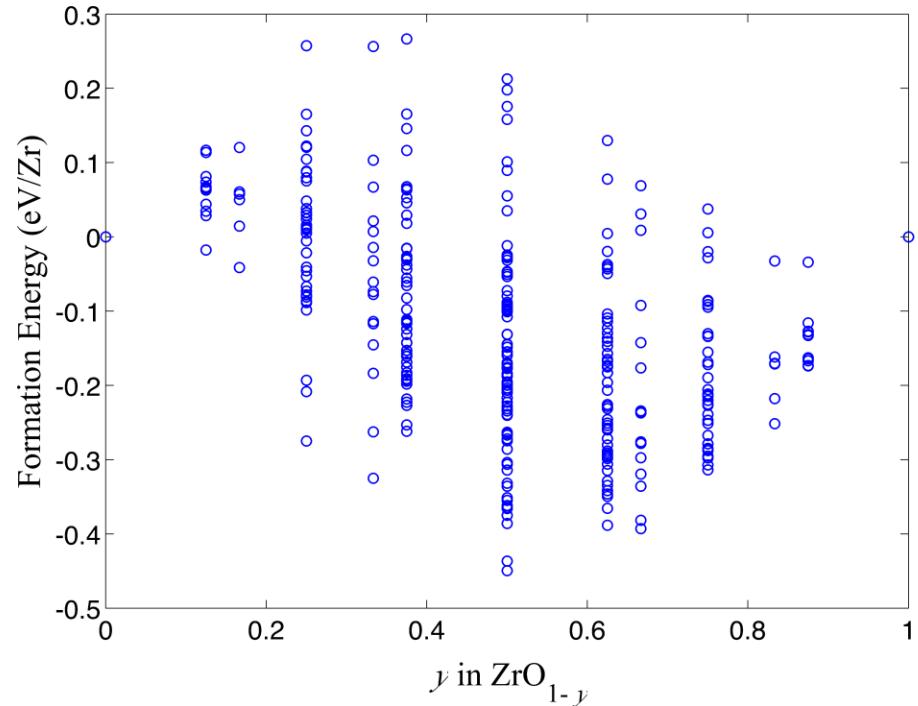
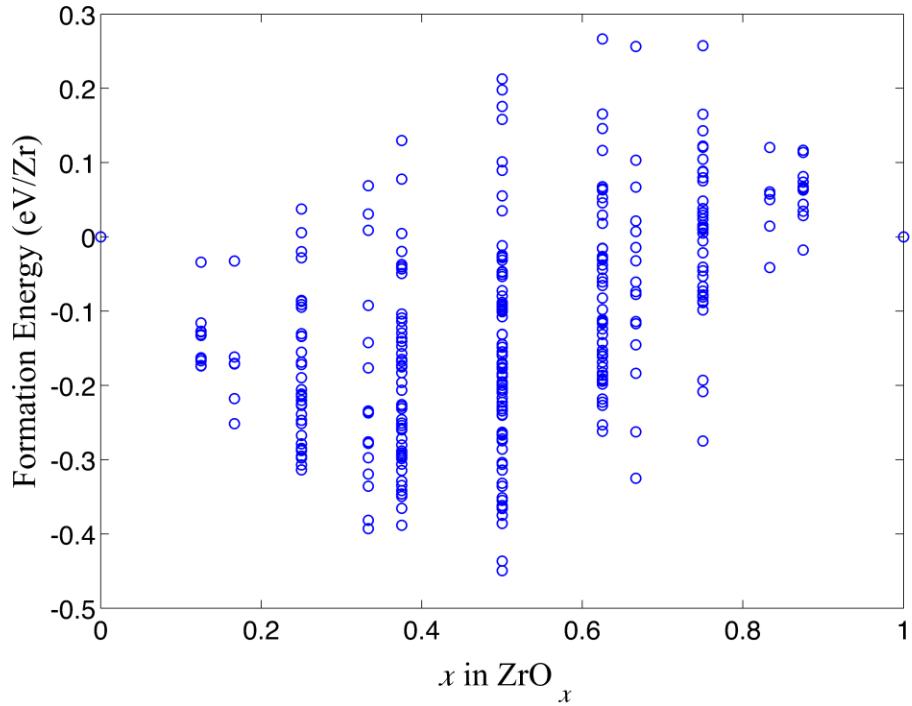
CASM Composition metrics

Each configuration has a *composition*, and there are various ways to measure it:

- Atomic fraction: `atom_frac(Zr)`
 - Excludes "Va" species
- Number per unit cell: `comp_n(Zr)`
- Parametric composition: `comp(a)`, `comp(b)`, etc.
 - Independent axes only:
 - Binary has 1 axis, Ternary has 2 axes, etc.
 - User choice of axes

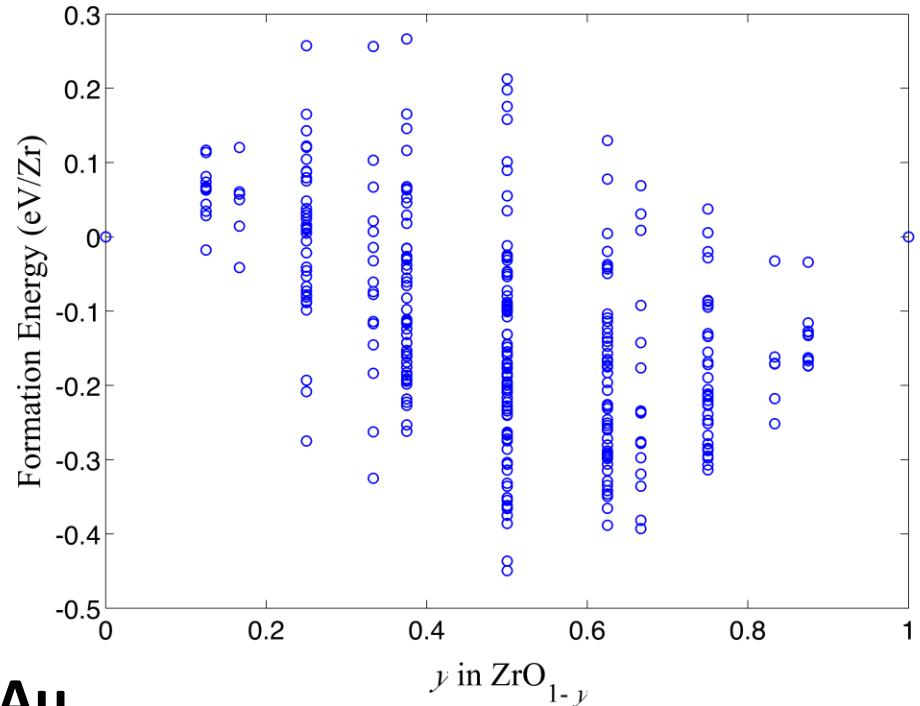
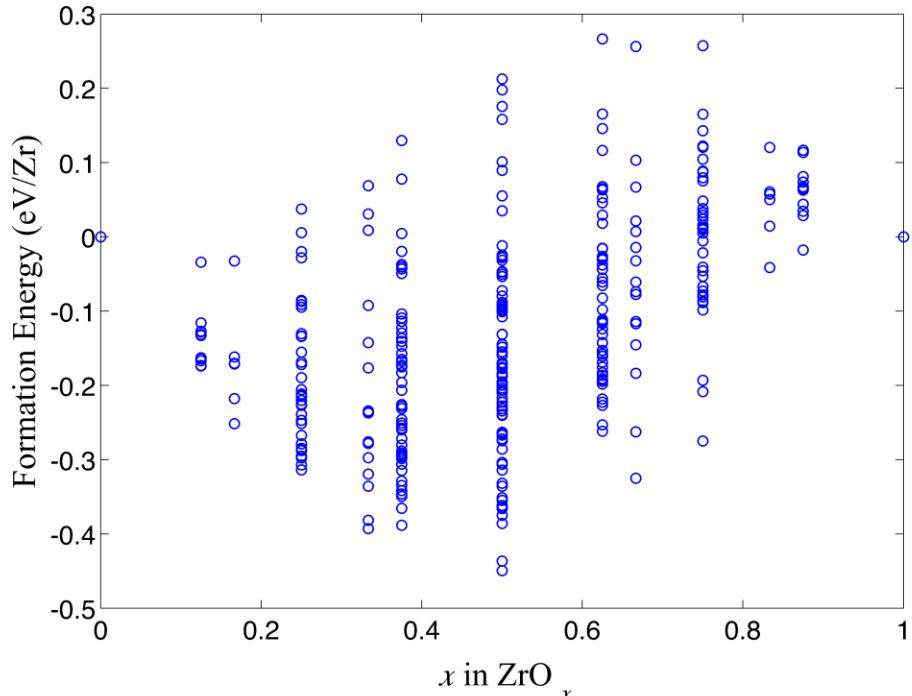
Specifying composition axes

Binary Compound:

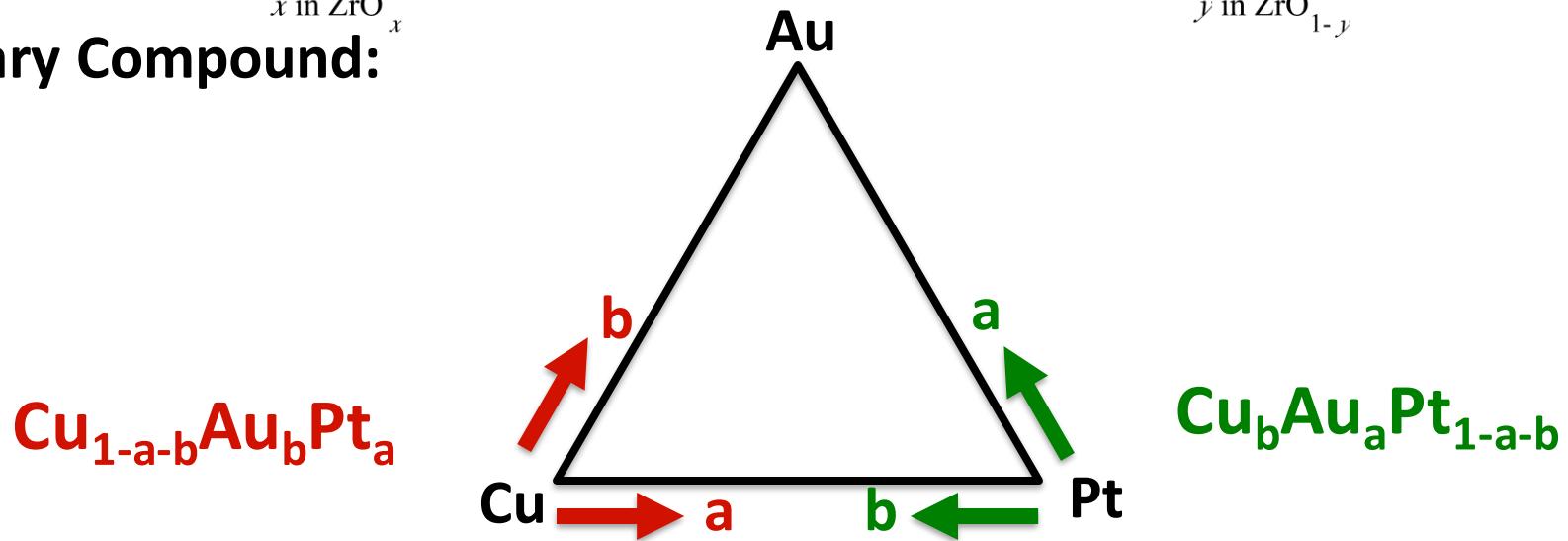


Specifying composition axes

Binary Compound:



Ternary Compound:



Specifying composition axes

```
[Adamant:~/Documents/Zr0$ casm composition -d
-- Construct: CASM Project --
from: "/Users/johnct/Documents/Zr0"

-- Load project data --
read: "/Users/johnct/Documents/Zr0/.casm/composition_axes.json"
```

Standard composition axes:

KEY	ORIGIN	a	GENERAL FORMULA
---	---	---	---
0	Zr(2)Va(2)	Zr(2)O(2)	Zr(2)Va(2-2a)O(2a)
1	Zr(2)O(2)	Zr(2)Va(2)	Zr(2)Va(2a)O(2-2a)

← Zr₂O_{2a}[Va]_{2(1-a)} ← Zr₂O_{2(1-a)}[Va]_{2a}

Please use 'casm composition --select' to choose your composition axes.

```
Adamant:~/Documents/Zr0$ █
```

NOTE: prim.json determines formula unit (2 Zr)

CASM Tutorial Outline

Day 1:

- Cluster expansion formulation
- Defining the "prim"
- Enumeration
- Occupation basis set construction

Day 2:

- DFT Calculation
- Fitting coefficients
- Monte Carlo
- Free energy integration and phase diagram construction

DFT calculations

Provide DFT code input files and parameters

- Pseudopotentials, k-point density, energy cut-off, and settings files

Use ‘casm select’ to identify a selection of configurations to calculate

`casm calc --setup -c my_selection`

- Setup input files for selected configurations

`casm calc --submit -c my_selection`

- Automates job submission and tracking
- Uses `prisms-jobs` package

Support for VASP in CASM 1.X

- QuantumEspresso update coming soon

CASM structure file

CASM

First-principles statistical mechanical software for the study of multi-component crystalline solids

[GitHub](#)

[PRISMS Center](#)

[Van der Ven group](#)

Example 1) Structure with occupation and energy

```
{  
    "atom_coords" : [  
        [ 0.00000000000, 0.00000000000, 0.00000000000 ],  
        [ 1.754750223661, -1.754750223661, 0.00000000000 ],  
        [ 1.754750223661, 0.00000000000, 1.754750223661 ],  
        [ 1.754750223661, 1.754750223661, 0.00000000000 ]  
    ],  
    "atom_type" : [ "C", "B", "B", "A" ],  
    "coordinate_mode" : "Cartesian",  
    "global_properties" : {  
        "energy" : {  
            "value" : 17.003  
        }  
    },  
    "lattice" : [  
        [ 1.754750223661, 0.00000000000, -1.754750223661 ],  
        [ 1.754750223661, 3.509500447322, 1.754750223661 ],  
        [ 1.754750223661, -3.509500447322, 1.754750223661 ]  
    ]  
}
```

On this page

Description

[Project files](#)

JSON Attributes List

JSON Attributes Description

[Structure JSON object](#)

Examples

[Example 1\) Structure with occupation and energy](#)

[Example 2\) Structure with occupation, displacement, and strain](#)

CASM structure file

Generate JSON structure file from configurations:

- `casm query -k 'structure' -j`

From VASP calculations in `training_data`:

- `casm calc --report`

Using ASE (Atomic Simulation Environment) as a converter:

- `casm convert...`

Relaxations and Mapping/Import

casm update:

Read all new property files, and check relaxation

lattice_deformation (LD):

Degree of homogeneous strain (per atom)

basis_deformation (BD):

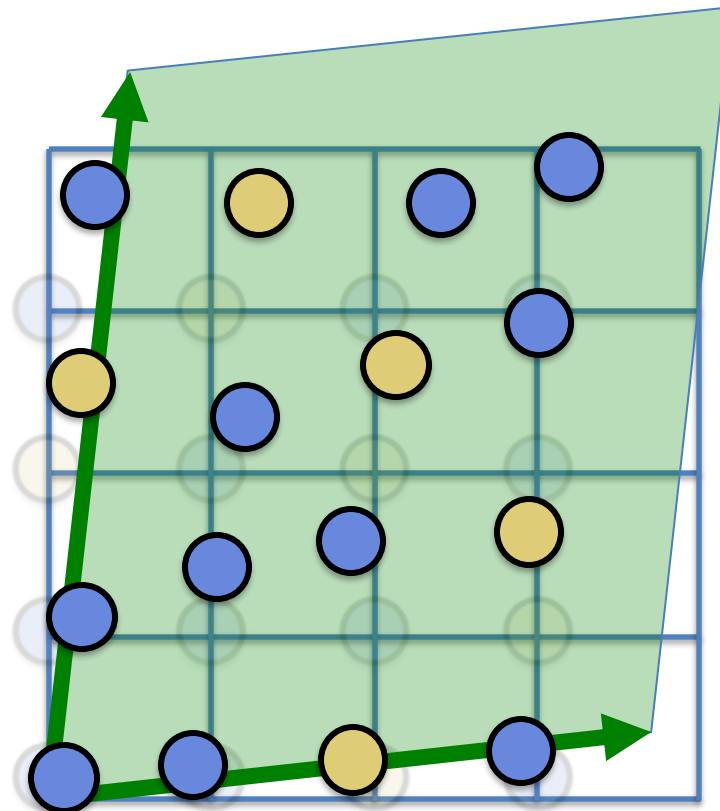
Mean square displacement

total deformation cost:

$$w * LD + (1-w) * BD, \quad 0 < w < 1$$

casm import:

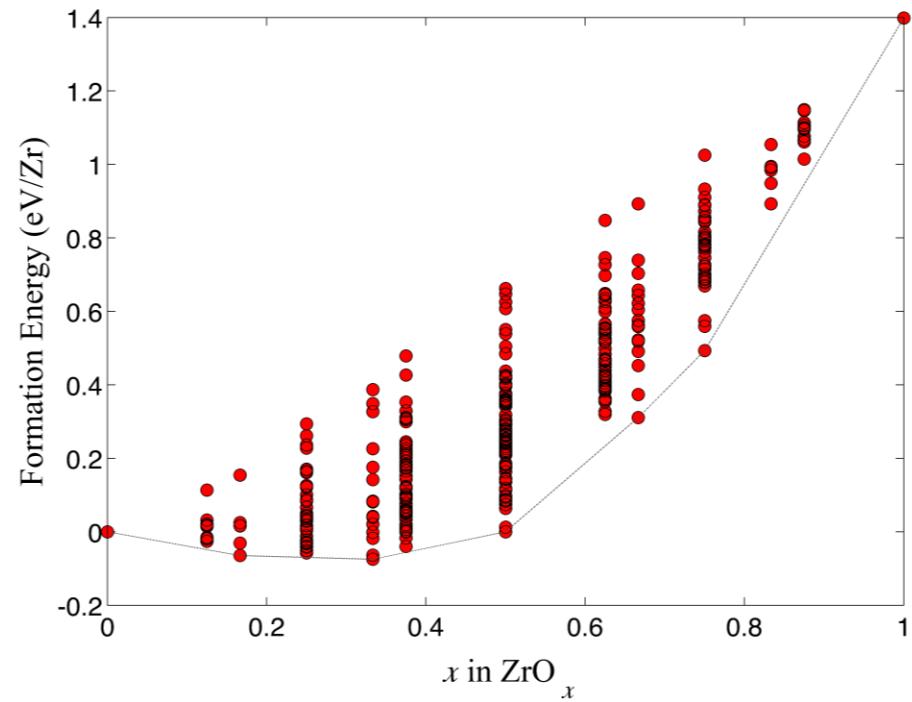
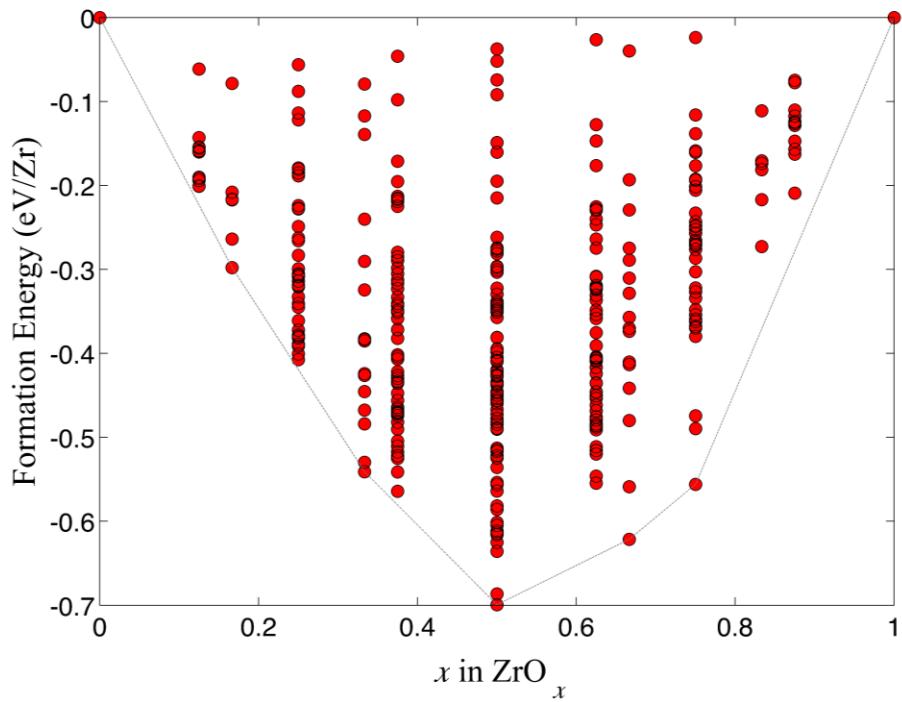
Use same algorithms as casm update to add ideal or relaxed configurations to project (even if not enumerated)



Very large relaxations may change the configuration. Conflicts can be resolved using either min-energy or min-deformation rule

Setting energy reference

Chemical energy can only be measured relative to reference compounds



`casm ref --set-auto` will attempt to set energy references as near to extreme stoichiometries as possible (left case)

CASM Tutorial Outline

Day 1:

- Cluster expansion formulation
- Defining the "prim"
- Enumeration
- Occupation basis set construction

Day 2:

- DFT Calculation
- Fitting coefficients
- Monte Carlo
- Free energy integration and phase diagram construction

$$\frac{E(\vec{\sigma})}{N} = \frac{V_0}{N} + \sum_{\alpha} m_{\alpha} V_{\alpha} \left(\frac{1}{Nm_{\alpha}} \sum_{\delta \in \Omega_{\alpha}} \phi_{\delta}(\vec{\sigma}) \right)$$

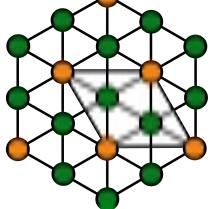
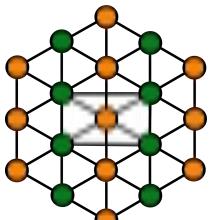
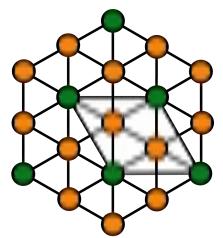
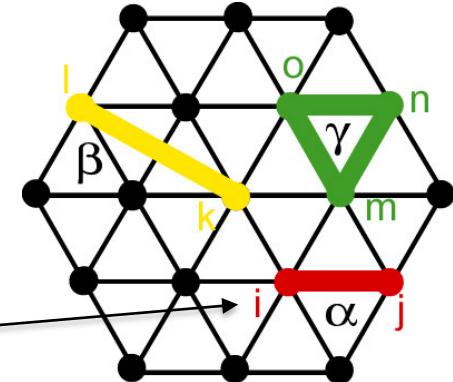
\downarrow

$\Gamma_{\alpha}(\vec{\sigma})$ Correlation function

$$e(\vec{\sigma}) = v_0 + \sum_{\alpha} v_{\alpha} \Gamma_{\alpha}(\vec{\sigma})$$

Cluster

\downarrow



$$\begin{pmatrix} e(\vec{\sigma}_1) \\ \vdots \\ e(\vec{\sigma}_L) \\ \vdots \\ e(\vec{\sigma}_M) \end{pmatrix} = \begin{pmatrix} 1 & \Gamma_{\alpha}(\vec{\sigma}_1) & \Gamma_{\beta}(\vec{\sigma}_1) & \Gamma_{\gamma}(\vec{\sigma}_1) & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \Gamma_{\alpha}(\vec{\sigma}_L) & \Gamma_{\beta}(\vec{\sigma}_L) & \Gamma_{\gamma}(\vec{\sigma}_L) & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \Gamma_{\alpha}(\vec{\sigma}_M) & \Gamma_{\beta}(\vec{\sigma}_M) & \Gamma_{\gamma}(\vec{\sigma}_M) & \vdots & \vdots \end{pmatrix} \begin{pmatrix} v_o \\ v_{\alpha} \\ v_{\beta} \\ v_{\gamma} \\ \vdots \\ \vdots \end{pmatrix}$$

$$\vec{Y} = X\vec{V}$$

Query calculation results

$$\vec{Y} = X\vec{V}$$

```
[\$ casm query -c CALCULATED -k 'comp formation_energy'
```

#	configname	selected	comp(a)	formation_energy
	SCEL1_1_1_1_0_0_0/0	1	0.00000000	0.00000000
	SCEL1_1_1_1_0_0_0/1	1	0.50000000	0.04906895
	SCEL1_1_1_1_0_0_0/2	1	1.00000000	0.00000000
	SCEL2_2_1_1_0_0_1/0	1	0.25000000	0.02808172
	SCEL2_2_1_1_0_0_1/1	1	0.50000000	0.03510646
	SCEL2_2_1_1_0_0_1/2	1	0.75000000	0.03163656
	SCEL2_2_1_1_0_1_1/1	1	0.50000000	0.03484921
	SCEL3_3_1_1_0_0_2/4	1	0.66666667	0.03913948
	SCEL3_3_1_1_0_0_2/5	1	0.33333333	0.03425718
	SCEL3_3_1_1_0_0_2/6	1	0.50000000	0.03925148
	SCEL3_3_1_1_0_0_2/7	1	0.66666667	0.03100968
	SCEL3_3_1_1_0_0_2/8	1	0.83333333	0.02288649
	SCEL3_3_1_1_0_2_1/0	1	0.16666667	0.01736991
	SCEL3_3_1_1_0_2_1/2	1	0.33333333	0.02552916
	SCEL3_3_1_1_0_2_1/3	1	0.50000000	0.03191005
	SCEL3_3_1_1_0_2_1/5	1	0.33333333	0.02587874
	SCEL3_3_1_1_0_2_1/6	1	0.50000000	0.03231296
	SCEL3_3_1_1_0_2_1/7	1	0.66666667	0.02772067
	SCEL3_3_1_1_0_2_1/8	1	0.83333333	0.02022512
	SCEL3_3_1_1_0_2_1/9	1	0.66666667	0.02819053
	SCEL3_3_1_1_0_2_2/1	1	0.33333333	0.03785139
	SCEL3_3_1_1_0_2_2/10	1	0.50000000	0.04466163

Query correlations

$$\begin{pmatrix} e(\vec{\sigma}_1) \\ \vdots \\ e(\vec{\sigma}_L) \\ \vdots \\ e(\vec{\sigma}_M) \end{pmatrix} = \begin{pmatrix} 1 & \Gamma_\alpha(\vec{\sigma}_1) & \Gamma_\beta(\vec{\sigma}_1) & \Gamma_\gamma(\vec{\sigma}_1) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_\alpha(\vec{\sigma}_L) & \Gamma_\beta(\vec{\sigma}_L) & \Gamma_\gamma(\vec{\sigma}_L) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_\alpha(\vec{\sigma}_M) & \Gamma_\beta(\vec{\sigma}_M) & \Gamma_\gamma(\vec{\sigma}_M) & \dots \end{pmatrix} \begin{pmatrix} v_o \\ v_\alpha \\ v_\beta \\ v_\gamma \\ \vdots \end{pmatrix}$$

[\$ casm query -k 'corr(0:5)'
– will print correlations

$$\vec{Y} = X\vec{V}$$

#	configname	selected	corr(1)	corr(2)	corr(3)	corr(4)	corr(5)
	SCEL1_1_1_1_0_0_0/0	1	-1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
	SCEL1_1_1_1_0_0_0/1	1	0.00000000	-1.00000000	1.00000000	-1.00000000	1.00000000
	SCEL1_1_1_1_0_0_0/2	1	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
	SCEL2_2_1_1_0_0_1/0	1	-0.50000000	0.00000000	0.33333333	0.00000000	1.00000000
	SCEL2_2_1_1_0_0_1/1	1	0.00000000	0.00000000	-0.33333333	0.00000000	1.00000000
	SCEL2_2_1_1_0_0_1/2	1	0.50000000	0.00000000	0.33333333	0.00000000	1.00000000
	SCEL2_2_1_1_0_1_1/1	1	0.00000000	-0.50000000	0.00000000	0.50000000	-1.00000000
	SCEL3_3_1_1_0_0_2/4	1	0.33333333	-0.33333333	0.55555556	-0.33333333	0.77777778
	SCEL3_3_1_1_0_0_2/5	1	-0.33333333	-0.33333333	0.11111111	0.11111111	0.55555556
	SCEL3_3_1_1_0_0_2/6	1	0.00000000	-0.33333333	0.11111111	-0.11111111	0.55555556
	SCEL3_3_1_1_0_0_2/7	1	0.33333333	0.33333333	0.11111111	0.11111111	0.55555556
	SCEL3_3_1_1_0_0_2/8	1	0.66666667	0.33333333	0.55555556	0.33333333	0.77777778
	SCEL3_3_1_1_0_2_1/0	1	-0.66666667	0.33333333	0.44444444	0.33333333	0.55555556
	SCEL3_3_1_1_0_2_1/2	1	-0.33333333	0.00000000	-0.11111111	0.22222222	0.11111111
	SCEL3_3_1_1_0_2_1/3	1	0.00000000	-0.33333333	-0.11111111	0.11111111	0.11111111
	SCEL3_3_1_1_0_2_1/5	1	-0.33333333	0.33333333	-0.11111111	-0.11111111	0.11111111

casm learn

- Built atop Python machine-learning package 'scikit-learn'

scikit-learn: <http://scikit-learn.org>
deap: <http://deap.readthedocs.io>

casm learn: Problem definition

Solve $\vec{Y} = X\vec{V}$ for \vec{V}

- X : matrix of correlations
- \vec{Y} : list of energies (DFT formation energies)
- \vec{V} : unknown expansion coefficients

$$\begin{pmatrix} e(\vec{\sigma}_1) \\ \vdots \\ e(\vec{\sigma}_L) \\ \vdots \\ e(\vec{\sigma}_M) \end{pmatrix} = \begin{pmatrix} 1 & \Gamma_\alpha(\vec{\sigma}_1) & \Gamma_\beta(\vec{\sigma}_1) & \Gamma_\gamma(\vec{\sigma}_1) & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_\alpha(\vec{\sigma}_L) & \Gamma_\beta(\vec{\sigma}_L) & \Gamma_\gamma(\vec{\sigma}_L) & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_\alpha(\vec{\sigma}_M) & \Gamma_\beta(\vec{\sigma}_M) & \Gamma_\gamma(\vec{\sigma}_M) & \vdots & \vdots \end{pmatrix} \begin{pmatrix} v_o \\ v_\alpha \\ v_\beta \\ v_\gamma \\ \vdots \end{pmatrix}$$

casm learn: Feature selection methods

- Some estimators (such as LASSO, Ridge Regression) automatically select features (casm learn option ‘SelectFromModel’)
- In other cases, specify algorithm to optimize CV score ('GeneticAlgorithm', 'RFE' (recursive feature elimination, etc.)

casm learn: Specification file

```
"problem_specs": {  
    "data": {  
        "y": "formation_energy",  
        "X": "corr",  
        "kwargs": null,  
        "type": "selection",  
        "filename": "train"  
    },  
    "cv": {  
        "method": "KFold",  
        "kwargs": {  
            "n_splits": 10,  
            "shuffle": true  
        }  
    },  
    "weight": {  
        "method": "wHullDist",  
        "kwargs": {  
            "A": 0.0,  
            "B": 1.0,  
            "hull_selection":  
            "CALCULATED",  
            "kT": 0.01  
        }  
    }  
}
```

Name of a 'casm select' output file, or "CALCULATED", or "MASTER"

scikit-learn class name and arguments, documented on scikit-learn.org

weighting method name and keyword arguments

Detailed descriptions of options can be found via
'casm-learn –settings-format'

casm learn: Specification file

```
"estimator": {  
    "method": "LinearRegression"  
},  
"feature_selection": {  
    "method": "GeneticAlgorithm",  
    "kwargs": {  
        "constraints_kwargs": {  
            "fix_on": [],  
            "n_features_min": 10,  
            "fix_off": [],  
            "n_features_max": "all"  
        },  
        "cxUniformProb": 0.5,  
        "mutFlipBitProb": 0.02,  
        "evolve_params_kwargs": {  
            "n_population": 50,  
            "n_generation": 10,  
            "n_repetition": 20,  
            "n_halloffame": 50,  
            "n_features_init": 10  
        },  
        "selTournamentSize": 3  
    }  
},
```

scikit-learn class name and keyword arguments, documented on scikit-learn.org

scikit-learn class name or casm.learn.evolve class name and keyword arguments

Detailed descriptions of options can be found via
'casm-learn –settings-format'

casm learn: Execution

- Specify dataset:

```
cd fitting
```

```
casm select -c CALCULATED -o train
```

- Run 'casm learn':

```
casm learn -s direct_fit.json
```

- Every run: stores results in 'direct_fit_halloffame.pkl'

casm learn: Execution

- Estimator and feature selection methods can change for subsequent runs
 - Individual results with best CV scores stored in hall of fame
- View results:

```
casm learn -s direct_fit.json --hall
```
- Select cluster expansion:

```
casm learn -s direct_fit.json --select 0
```

CASM Tutorial Outline

Day 1:

- Cluster expansion formulation
- Defining the "prim"
- Enumeration
- Occupation basis set construction

Day 2:

- DFT Calculation
- Fitting coefficients
- Monte Carlo
- Free energy integration and phase diagram construction

Grand canonical ensemble

Constant T, μ_{Li}

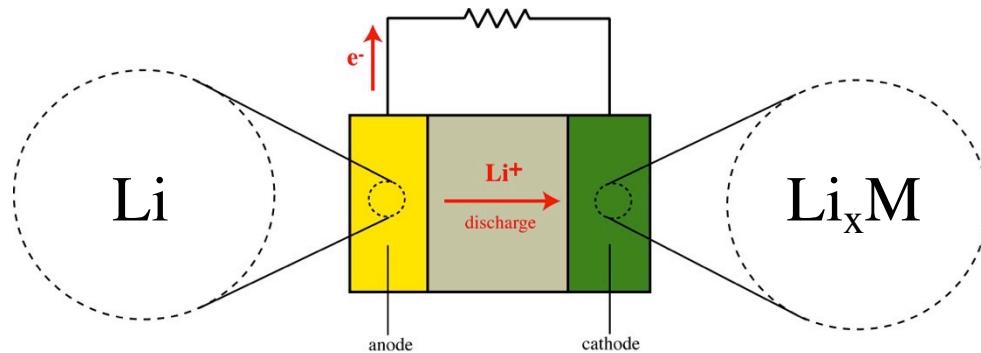
Controlling μ_{Li} instead of N_{Li}

by controlling the voltage

Characteristic potential

$$\Phi = F - \mu_{\text{Li}} N_{\text{Li}}$$

$$Z = \sum_{\vec{\sigma}} \exp\left(-\frac{E(\vec{\sigma}) - \mu_{\text{Li}} N_{\text{Li}}}{k_B T}\right) \longrightarrow \Phi = -k_B T \ln Z$$



$$P(\vec{\sigma}) = \frac{1}{Z} \exp\left(-\frac{E(\vec{\sigma}) - \mu_{\text{Li}} N_{\text{Li}}}{k_B T}\right)$$

Thermodynamic averages

Canonical Ensemble
Constant T, N_{Li}

Internal energy

$$\bar{E} = \sum_{\vec{\sigma}} P(\vec{\sigma}) E(\vec{\sigma})$$

Grand Canonical Ensemble
Constant T, μ_{Li}

Grand canonical energy

$$\bar{\Omega} = \sum_{\vec{\sigma}} P(\vec{\sigma}) \Omega(\vec{\sigma}) \quad \Omega = E - \mu_{\text{Li}} N_{\text{Li}}$$

Average number of Li

$$\bar{N}_{\text{Li}} = \sum_{\vec{\sigma}} P(\vec{\sigma}) N_{\text{Li}}(\vec{\sigma})$$

Importance sampling

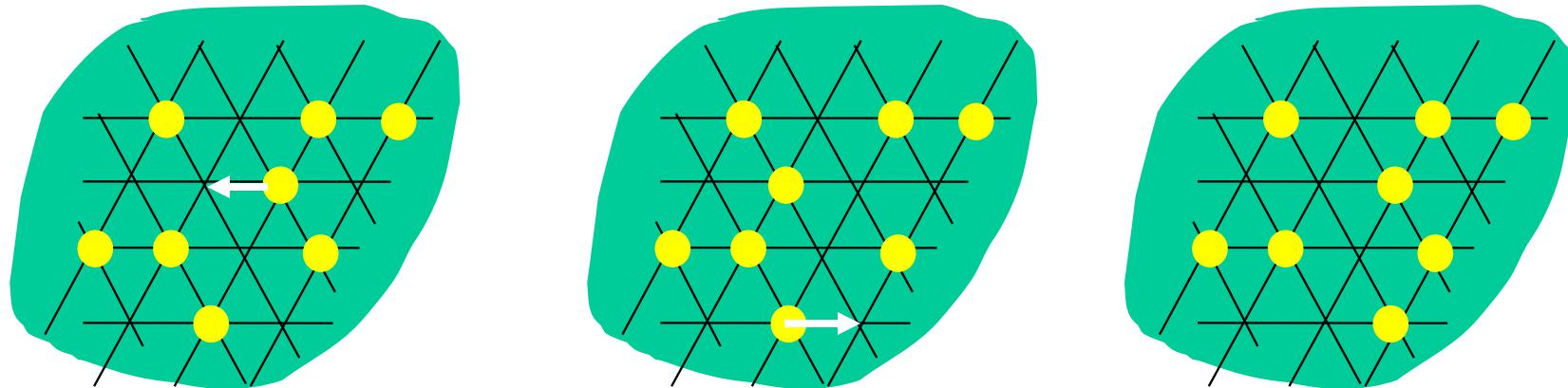
To numerically average thermodynamic quantities,
you don't want to randomly sample $\vec{\sigma}$ from the 2^N possibilities

Instead you want to sample configurations $\vec{\sigma}$
with their probability of occurrence

$$P(\vec{\sigma}) = \frac{1}{Z} \exp\left(-\frac{E(\vec{\sigma}) - \mu_{Li} N_{Li}}{k_B T}\right)$$

$$Z = \sum_{\vec{\sigma}} \exp\left(-\frac{E(\vec{\sigma}) - \mu_{Li} N_{Li}}{k_B T}\right)$$

Monte Carlo Algorithm



A Monte Carlo Algorithm

1. Start with some configuration
2. Choose perturbation of the system
3. Compute energy for that perturbation

4. If $\Delta E < 0$ \rightarrow accept perturbation
 If $\Delta E > 0$ \rightarrow accept perturbation if $\exp\left(\frac{-\Delta E}{kT}\right) > \xi$ a random number
5. Choose next perturbation

Property will be average over these states

Monte Carlo output

(Grand Canonical)

Thermodynamic variables controlled in Monte Carlo (i.e. input)

$$T, \mu_{Li}$$

Output from Monte Carlo

$$\Omega = E - \mu_{Li} N_{Li}$$

$$\overline{\Omega}(T, \mu_{Li})$$

$$C_{\mu_{Li}} = \frac{\overline{\Omega^2} - \overline{\Omega}^2}{k_B T^2}$$

$$\chi_{Li} = \frac{\overline{N_{Li}^2} - \overline{N}_{Li}^2}{k_B T}$$

Free energy integration

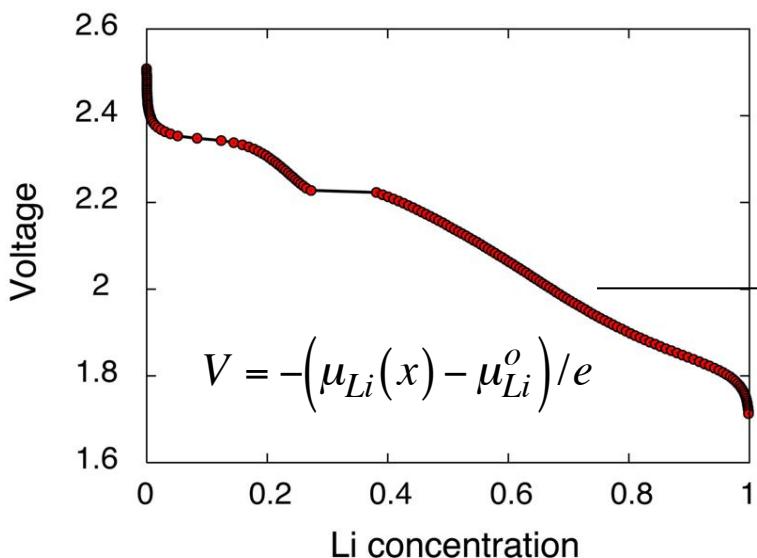
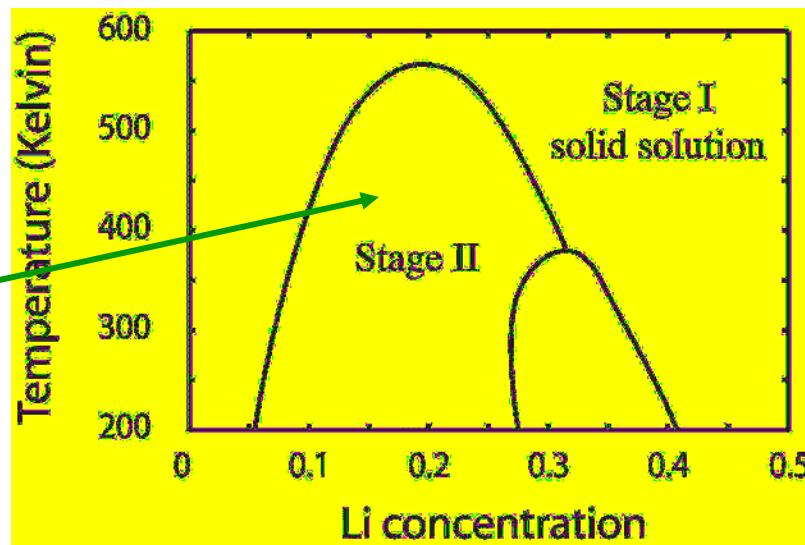
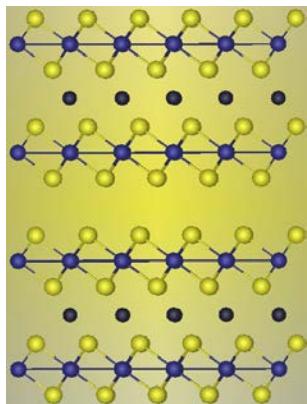
$$g(x_{Li}, T) = g(x_{Li}^o, T) + \int_{x_{Li}^o}^{x_{Li}} \mu_{Li} dx_{Li}$$

$$\beta \Phi(\mu_{Li}, T) = \beta_o \Phi(\mu_{Li}^o, T) + \int_{\beta_o}^{\beta} \Omega d\beta$$

$$\Phi = F - \mu_{Li} N_{Li}$$

$$\Omega = E - \mu_{Li} N_{Li}$$

Li_xTiS_2 : thermodynamic properties (grand canonical monte carlo)



Free energy integration

$$g(x_{\text{Li}}, T) = g(x_{\text{Li}}^o, T) + \int_{x_{\text{Li}}^o}^{x_{\text{Li}}} \mu_{\text{Li}} dx_{\text{Li}}$$

CASM Workflow

Monte Carlo

$$Z = \sum_{\vec{\sigma}} \exp\left(-\frac{E(\vec{\sigma})}{k_B T}\right)$$

- Evaluate effective Hamiltonian Monte Carlo for ensemble averages
 - Traverse grid of parameters (μ , T , etc.) in various directions
- Integrate to determine free energies

$$F = -k_B T \ln(Z)$$

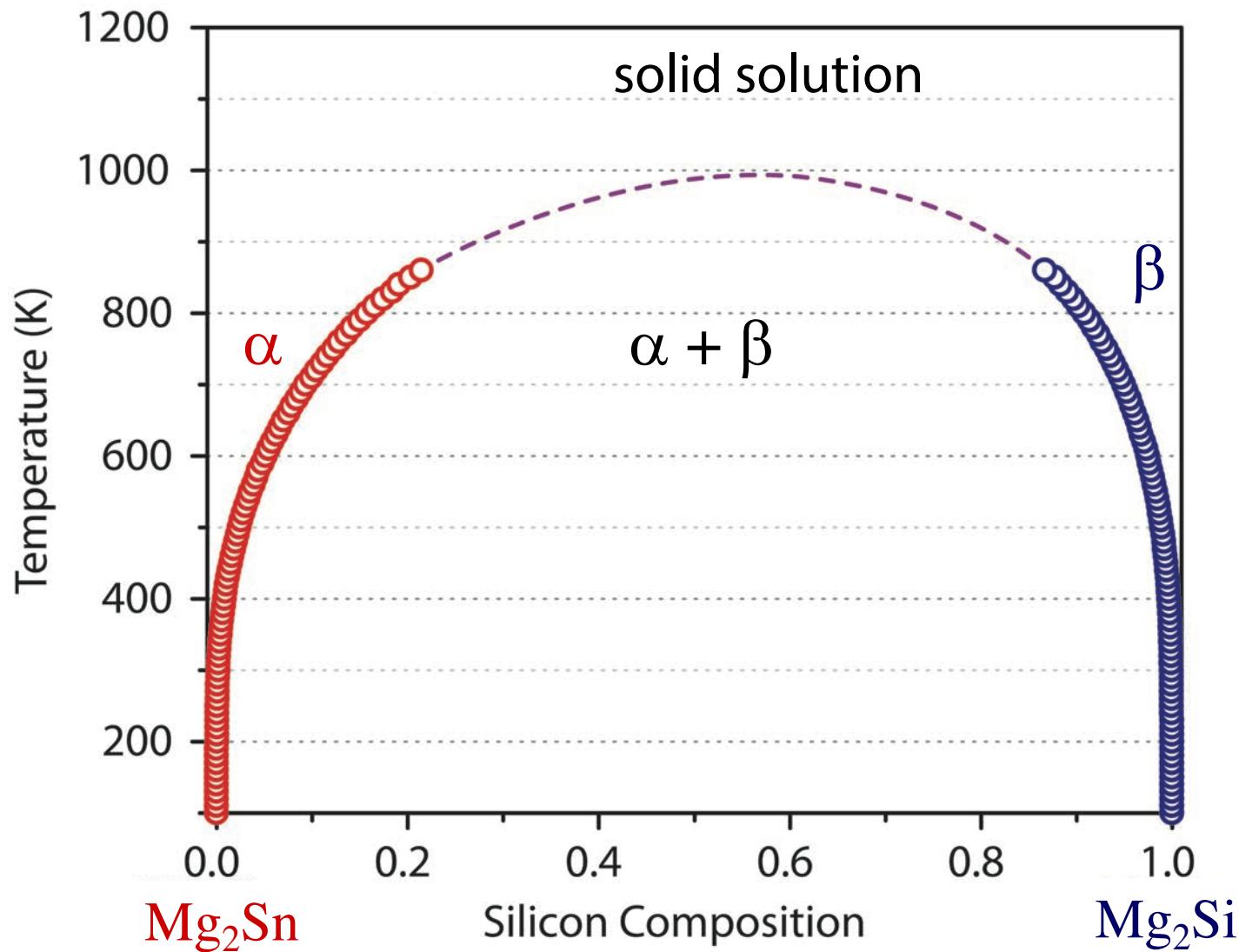
Monte Carlo

- See 'casm monte -h' and 'casm format --monte' for help setting up an input file.
- The input file lets you specify:
 - Monte Carlo ensemble:
 - “grand_canonical” for semi-grand canonical ensemble
 - “canonical” for fixed-composition (canonical) ensemble
 - Pathway in thermodynamic parameter space, with dependent or independent runs

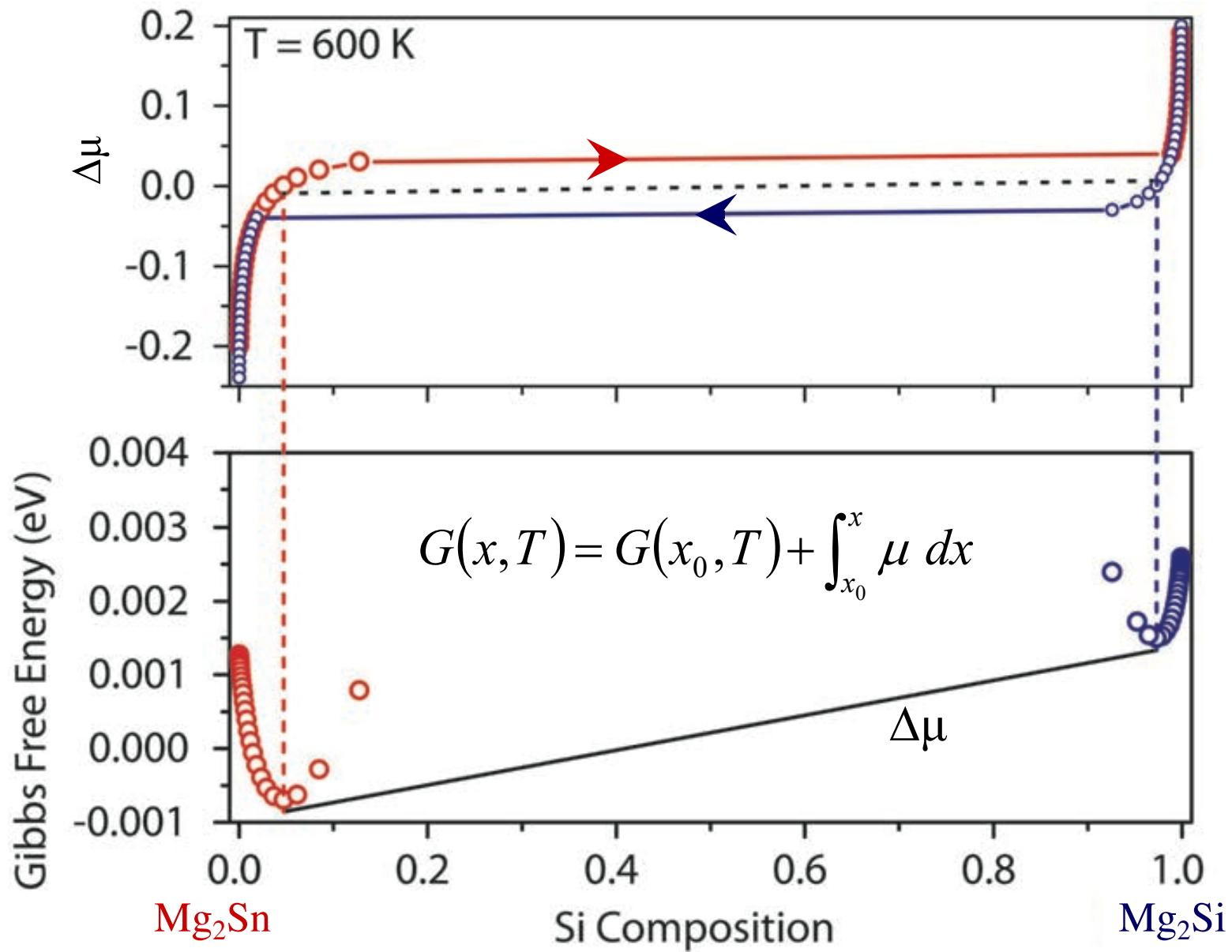
Monte Carlo

- See 'casm monte -h' and 'casm format --monte' for help setting up an input file.
- The input file lets you specify:
 - Quantities to sample & convergence criteria
 - The cluster expansion to use
 - Supercell and initial configuration
 - Enable saving configurations encountered during Monte Carlo into your configuration list

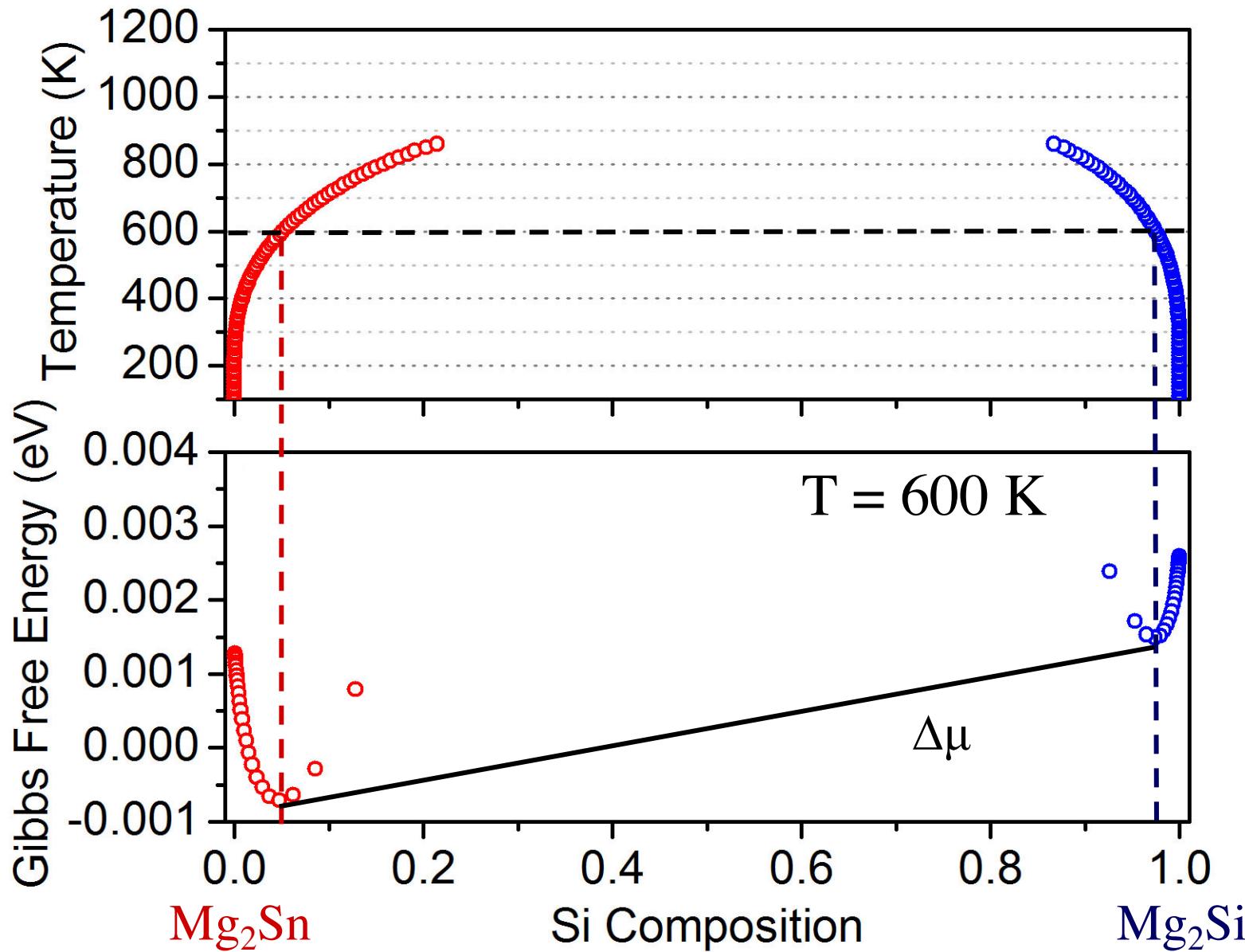
Example: miscibility gap



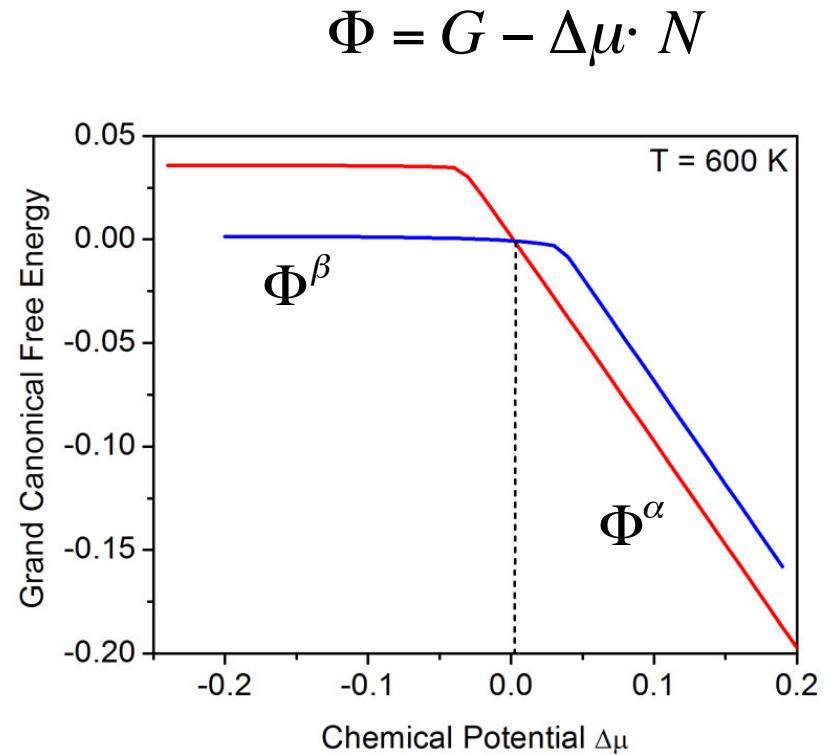
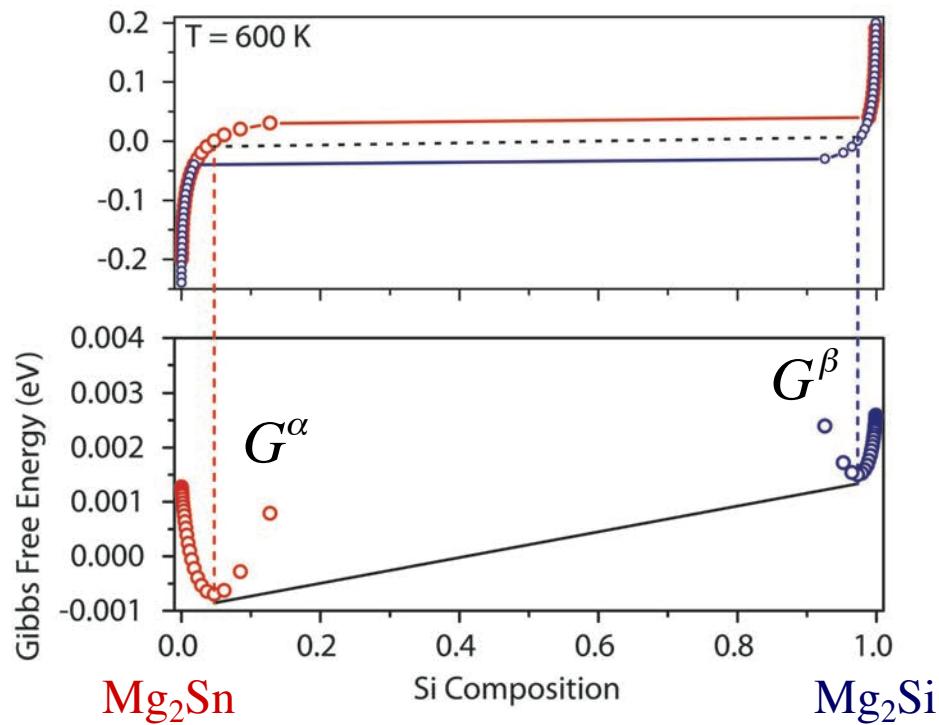
Chemical Potential and Free Energy



Drawing the Phase Diagram

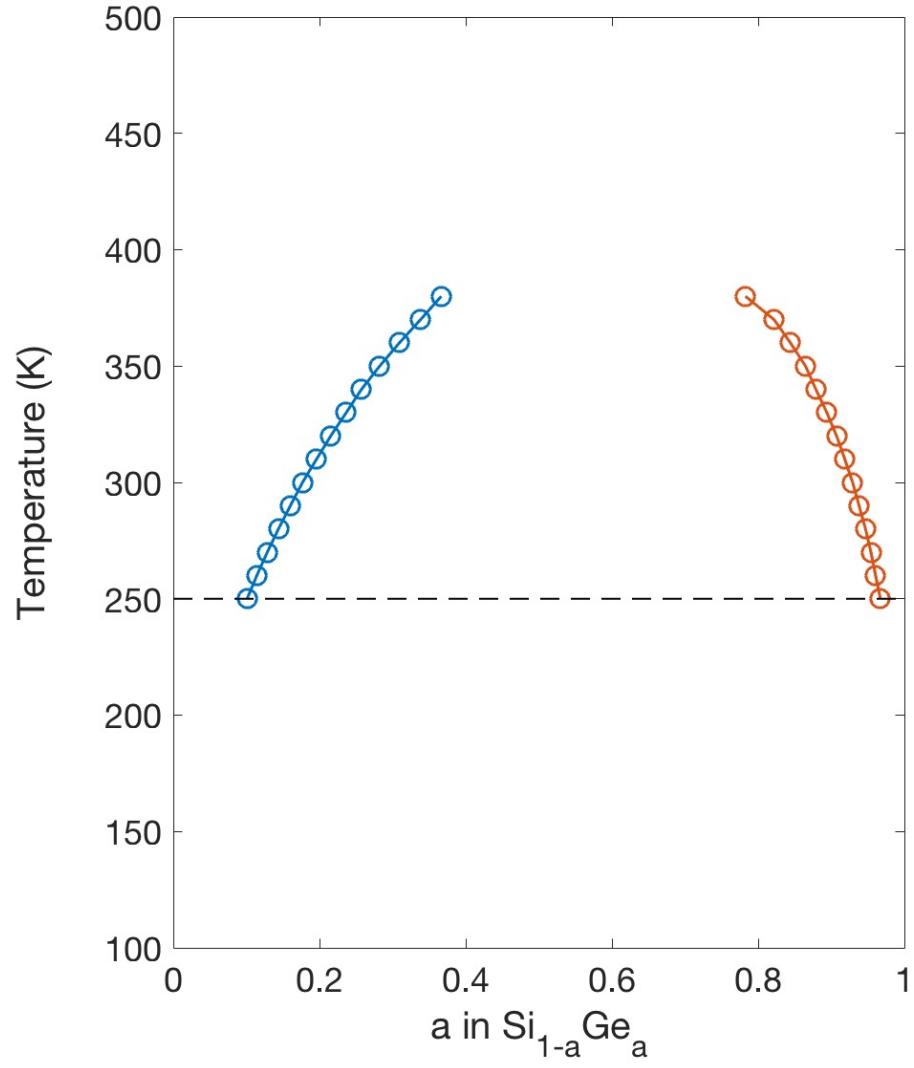
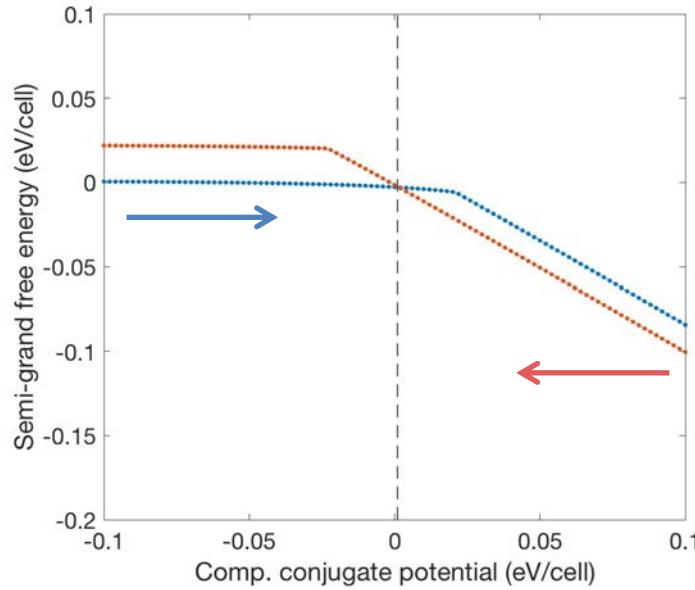
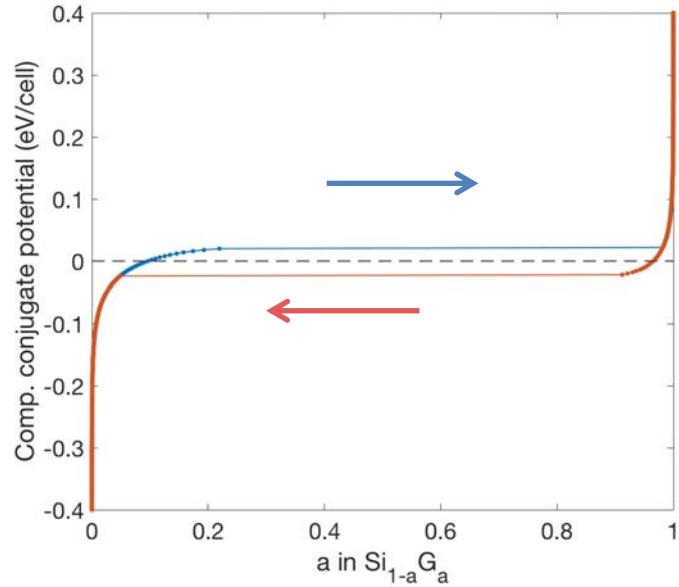


In terms of grand canonical free energies

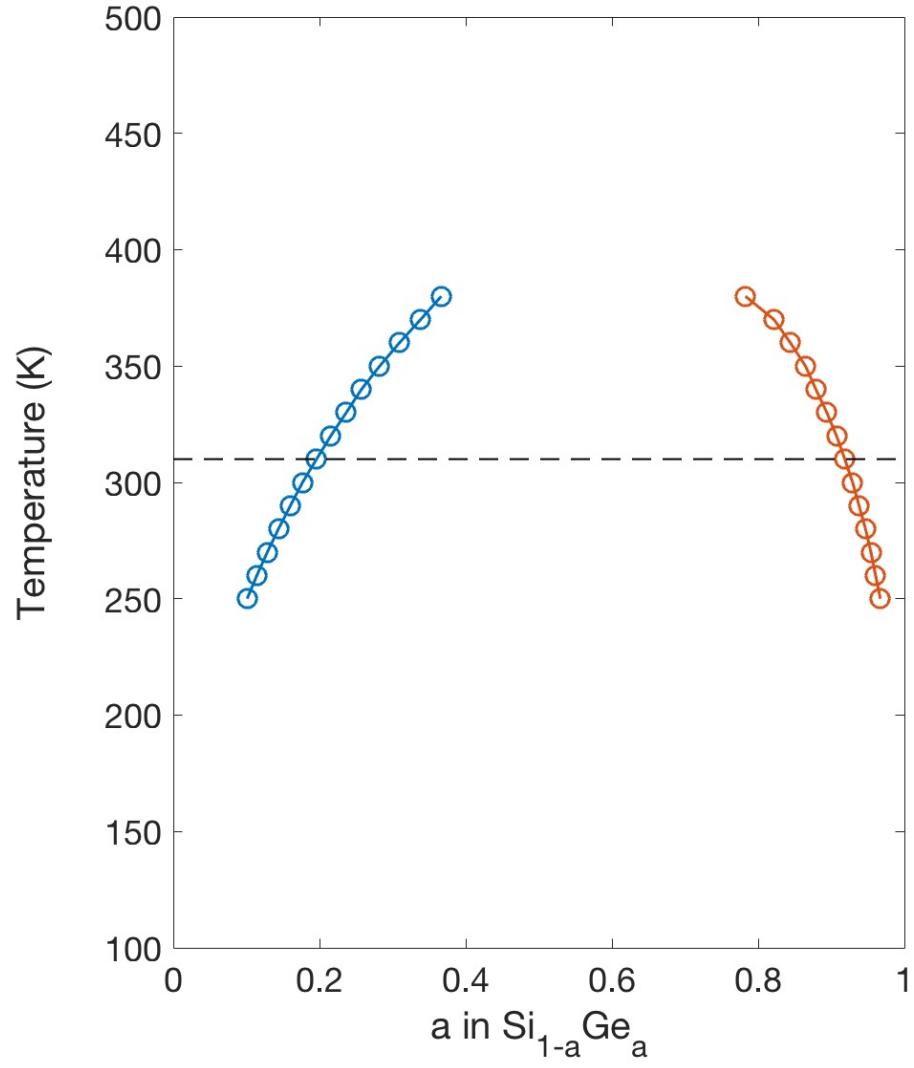
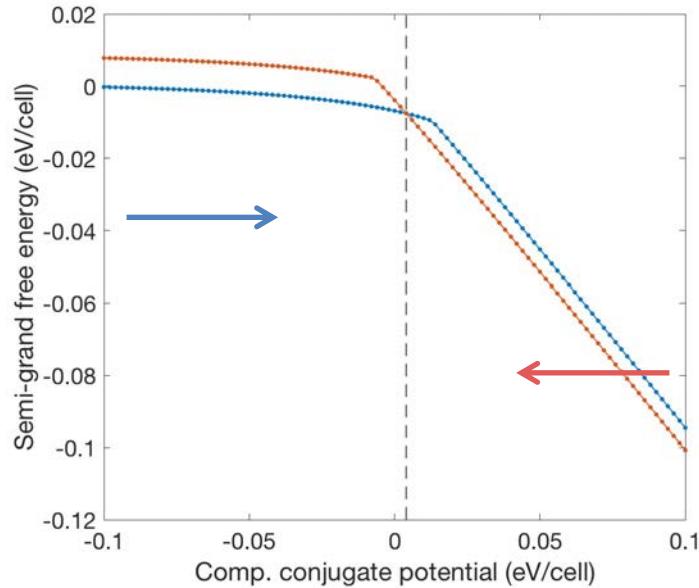
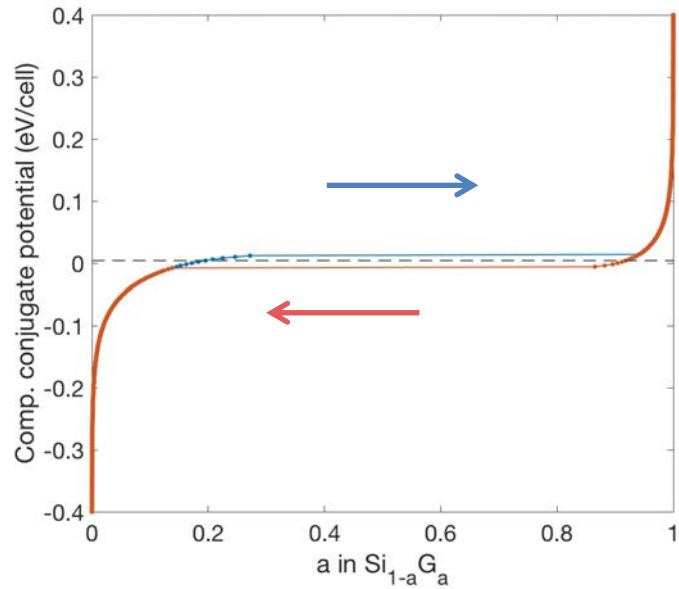


$$\Phi = G - \Delta\mu \cdot N$$

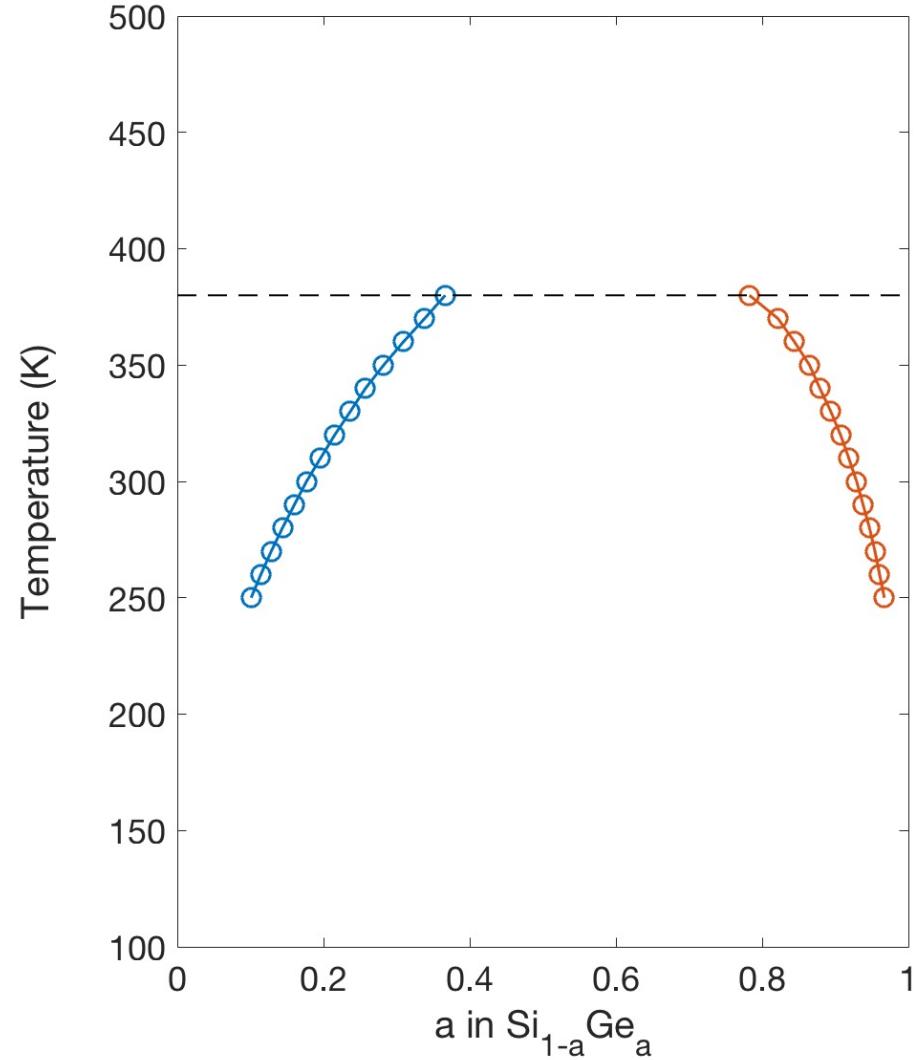
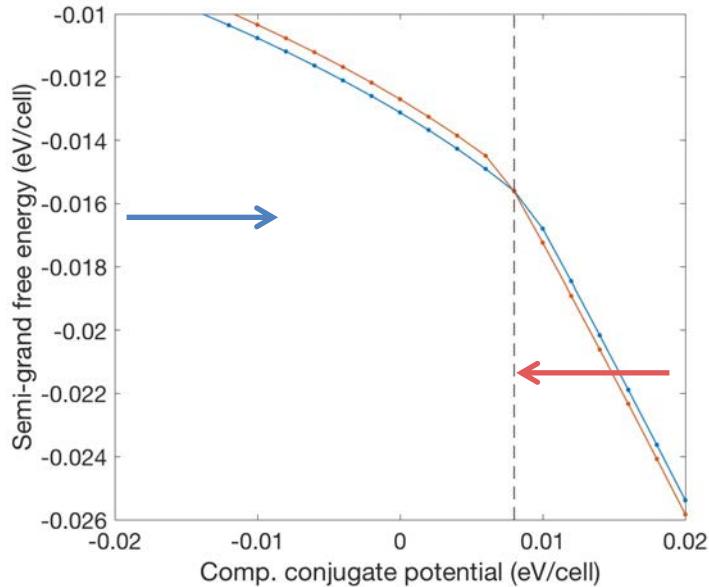
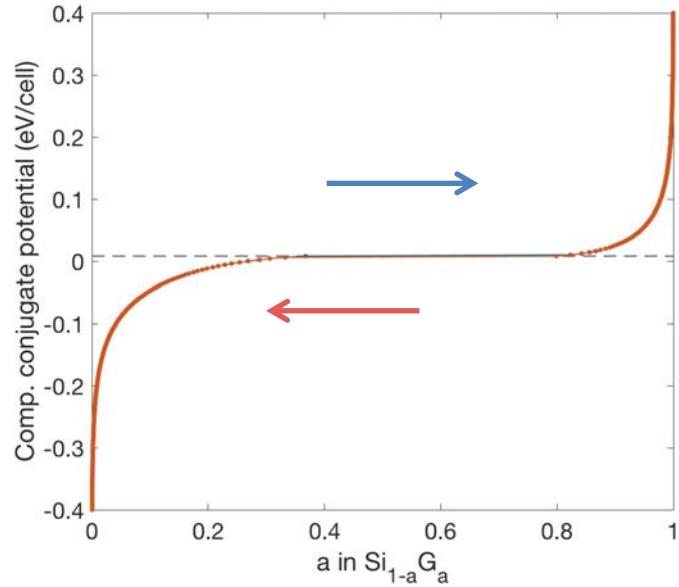
Phase diagram construction

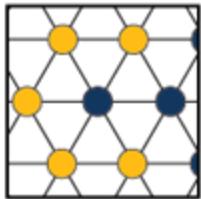


Phase diagram construction



Phase diagram construction

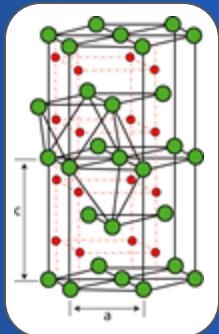




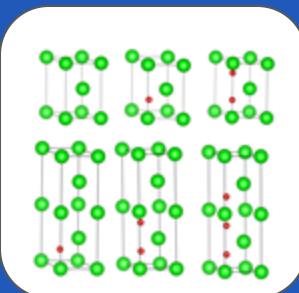
CASM

A Clusters Approach to Statistical Mechanics

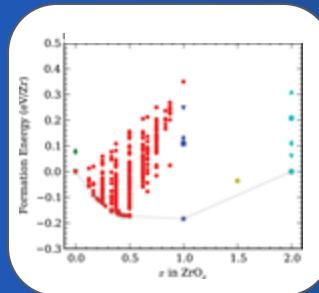
prisms-center.github.io/CASMcode_docs/



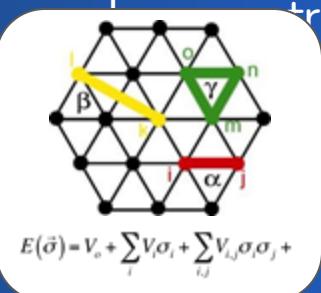
Crystal
structure
specification



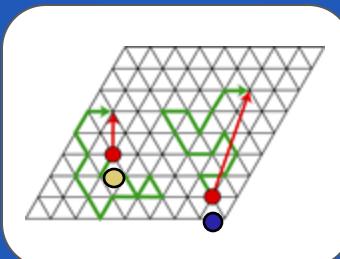
Structure
enumeration



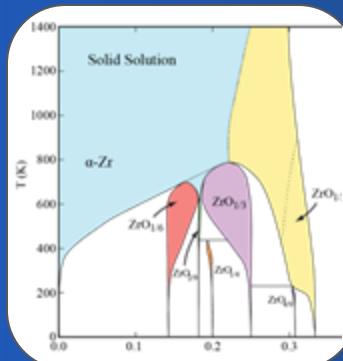
Ab initio
calculations



Effective
Hamiltonian
construction
and



Monte Carlo
calculations



Thermodyna
mic and
kinetic