Supplementary Information

Construction of a DFT-Derived Thermodynamic Database

We employed density functional theory to generate a large amount of thermodynamic data with which to train our data mining models. In particular, we calculated roughly 15,000 compounds from the Inorganic Crystal Structure Database (ICSD) and used these results to infer chemical interactions between elements. Other groups have performed such high-throughput calculations on materials from the ICSD,[1-3] and have queried the outputs for interesting technological materials.[4, 5] We also perform some limited chemical potential fitting in cases where we mix DFT and DFT+*U* calculations,[6] where elements undergo finite-temperature phase changes, or where elements crystallize in molecular configurations for which semi-local DFT performs particularly poorly. In short, our general approach is similar to that employed by these other groups, and indeed much of our work occurred in parallel with theirs.

In the present paper, however, we analyze our DFT calculation outputs in a substantially different way than has been done previously. We desire the capability to investigate arbitrary, never-before-studied chemistries in a very computationally efficient fashion, such that we may search a combinatorial explosion of possible compositions for new materials. To do so, we build a predictive model that gives us insight into the chemical behavior of any combinations of elements. Our model requires neither knowledge of crystal structure nor any other experimental inputs. The model is based on combining a metallurgical heuristic with a machine learning algorithm, as discussed in the main body of our manuscript. The model is approximately six orders of magnitude faster than DFT for studying a particular composition, yet provides similar thermodynamic predictive accuracy.

In order to produce examples of phase stability with which to train our model, we discretize all of the binary phase diagrams generated by our DFT calculations. In particular, we divide the convex hulls of each of the C(89 elements, 2) = 3,916 binary systems in our study into 19 discrete formation energy data points. We give an example of this discretization procedure for the Si-V system in Figure S1.

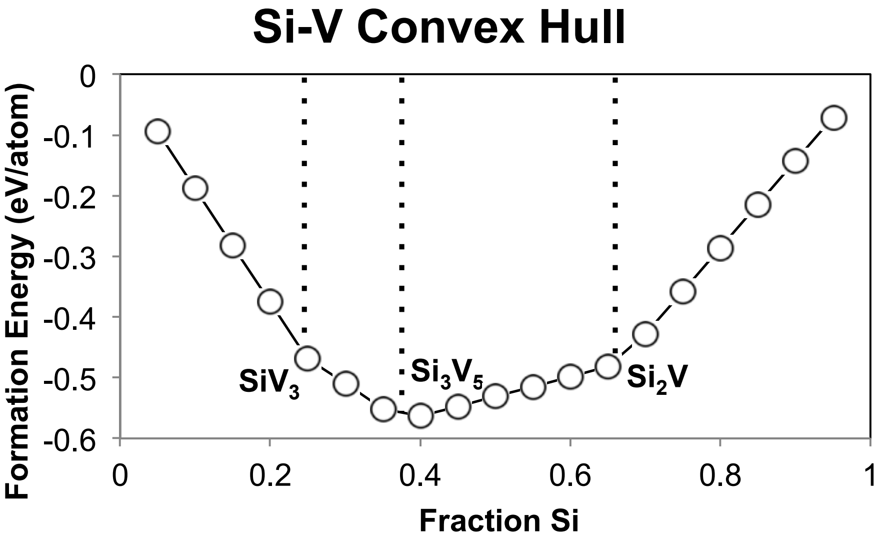

Figure S1. An example binary convex hull constructed from DFT calculations of stable compounds (in this case, SiV_3, Si_3V_5, and Si_2V), illustrating discrete points at which we sample formation energies.

Construction of a Metallurgical Heuristic

One component of our stability screening model is a simple metallurgical heuristic,[7, 8] which constructs a ternary compound's formation energy from a composition-weighted average of corresponding binary compound formation energies. We illustrate this averaging procedure schematically in Figure S2.

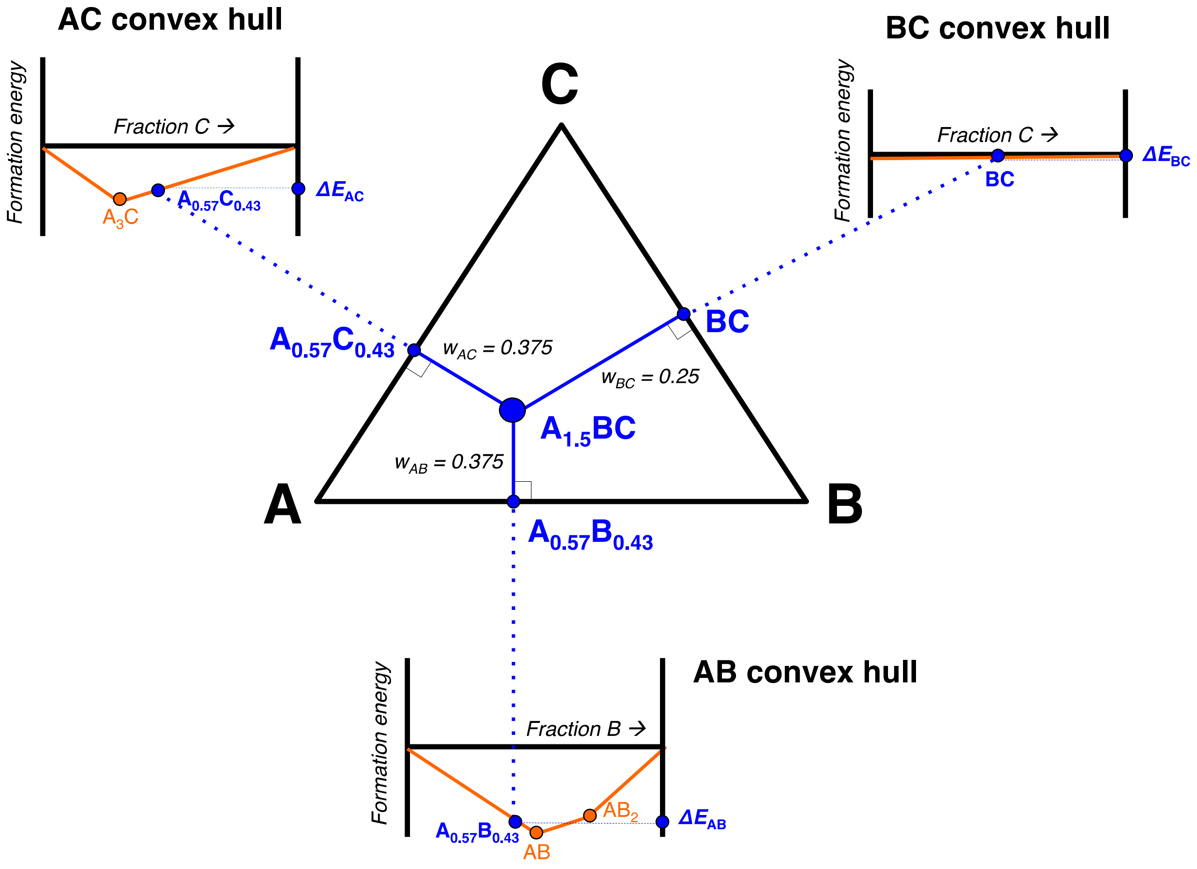

Figure S2. A schematic illustration of our metallurgical heuristic, which averages constituent binary formation energies to produce a prediction for a ternary formation energy. In this case, the formation energy of a ternary compound $A_{1.5}BC$ is given by $\Delta E_{tern} = 0.375 * \Delta E_{AC} + 0.375 * \Delta E_{AB} + 0.25 * \Delta E_{BC}$.

In Figure S2, we wish to construct the formation energy of a ternary composition $A_{1.5}BC$ based on known formation energies in the binary systems AC, AB, and BC. In this hypothetical case, AC exhibits a single stable compound at A_3C, AB has two stable compounds at AB and AB_2, and BC is a phase-separating system containing no stable compounds. In all three binary systems, the stable compounds (or lack thereof) define a convex hull between the two constituent elements, and hence specify the formation energy of all intermediate compositions between pure component 1 and pure component 2.

From the point representing $A_{1.5}BC$ on the ternary A-B-C phase diagram, we construct perpendiculars to the AC, AB, and BC edges of the ternary diagram. The resulting intersection points correspond to compositions on the binary AC, AB, and BC convex hulls. We obtain formation energies from each of these three points and make a prediction for $A_{1.5}BC$ based on a weighted average of the three binary-derived points. The weights are simply inversely related to the distance between $A_{1.5}BC$ and each binary hull point. Thus, in this example, because $A_{1.5}BC$ is more distant from the pure BC edge of the ternary diagram, the BC weight is smaller than the AB and AC weights.

Training Data for the Machine Learning Model

The total dataset for training the ML model consists of 83,728 rows and 130 columns. The rows correspond to 9,324 stable ternary compounds and 74,404 discretized points on binary A-B phase diagrams. The ML task at hand is to use 129 descriptive attributes (*none* of which are DFT calculation outputs) to predict the value of the target (130th) column, which is the given composition's formation energy with respect to the elements (i.e., a DFT calculation output).

To explain the origin of the descriptive attributes, which are readily obtained analytically for any given composition, we will take the example of iron oxide, FeO. Most of the 129 descriptive attributes simply encode the particular composition under consideration. In particular, 112 of these attributes give a compound's percentage elemental composition. For FeO, only two of these attributes are nonzero: Fe_frac = 0.5 and O_frac = 0.5. The other 110, from Ac_frac, Ag_frac, … to Zn_frac, Zr_frac are identically zero.

The other 17 descriptive attributes are heuristic quantities that we developed using chemical intuition to boost the accuracy of the resulting ML model. They are described below, and their numerical values are reported for the FeO example:

- **Average atomic mass**: Composition-weighted average of the atomic masses of the elements in the compound. *Value for FeO: 0.5 x 55.845 + 0.5 x 15.999 = 35.92*.
- **Average column on periodic table**: Composition-weighted average of the columns of the elements in the compound. *Value for FeO: 0.5 x 8 + 0.5 x 16 = 12.0*.
- **Average row on the periodic table**: Composition-weighted average of the rows of the elements in the compound. *Value for FeO: 0.5 x 4 + 0.5 x 2 = 3.0*.
- **Maximum difference in atomic number**: Largest atomic number in the composition less the smallest. *Value for FeO: 26 – 8 = 18*.
- **Average atomic number**: Composition-weighted average of the atomic numbers of the elements in the compound. *Value for FeO: 0.5 x 26 + 0.5 x 8 = 17.0*.
- **Maximum difference in atomic radii**: Largest atomic radius in the composition less the smallest (in pm). *Value for FeO: 140 – 60 = 80*.
- **Average atomic radius**: Composition-weighted average of the atomic radii of the elements in the compound. *Value for FeO: 0.5 x 140 + 0.5 x 60 = 100.0*.
- **Maximum difference in electronegativity**: Largest electronegativity in the composition less the smallest. *Value for FeO: 3.44 – 1.83 = 1.61*.

- **Average electronegativity**: Composition-weighted average of the electronegativities of the elements in the compound. *Value for FeO: 0.5 x 3.44 + 0.5 x 1.83 = 2.635*.
- **Average number of *s* valence electrons**: Composition-weighted average of the number of *s* valence electrons associated with the elements in the compound. *Value for FeO: 0.5 x 4 + 0.5 x 2 = 3.0*.
- **Average number of *p* valence electrons**: Analogous to above, but for *p* electrons. *Value for FeO: 0.5 x 0 + 0.5 x 4 = 2.0*.
- **Average number of *d* valence electrons**: Analogous to above, but for *d* electrons. *Value for FeO: 0.5 x 6 + 0.5 x 0 = 3.0*.
- **Average number of *f* valence electrons**: Analogous to above, but for *f* electrons. *Value for FeO: 0.5 x 0 + 0.5 x 0 = 0.0*.
- ***s* fraction of valence electrons**: Composition-weighted fraction of all valence electrons in the compound that represent *s* states. *Value for FeO: 3.0 / (3.0 + 2.0 + 3.0 + 0.0) = 0.375*.
- ***p* fraction of valence electrons**: Analogous to above, but for *p* electrons. *Value for FeO: 2.0 / (3.0 + 2.0 + 3.0 + 0.0) = 0.25.*
- ***d* fraction of valence electrons**: Analogous to above, but for *d* electrons. *Value for FeO: 3.0 / (3.0 + 2.0 + 3.0 + 0.0) = 0.375.*
- ***f* fraction of valence electrons**: Analogous to above, but for *f* electrons. *Value for FeO: 0.0 / (3.0 + 2.0 + 3.0 + 0.0) = 0.0.*

Predictive Modeling of Formation Energy

We use the rotation forest ensembling technique[9] with reduced error pruning trees as the underlying regression model to predict formation energy. Here we first briefly describe ensemble modeling in general, then the specific techniques used in our model.

Ensemble learning is a popular technique that essentially combines multiple models into a single model with the goal of improving accuracy and robustness as compared to constituent models. A great deal of research has gone into designing ensemble models based on the same predictive modeling technique used on different data subsets and/or feature subsets.[9, 10] Figure S3 depicts the general approach of ensemble learning, wherein the labeled data (with known formation energy) is used to construct multiple distinct models using subsets of original labeled data, which are merged to obtain the final trained ensemble model. For testing, the unlabeled data (with unknown formation energy) is passed through the ensemble model to generate the final predictions. Different ensemble methods can vary in terms of how they permute the training data to construct subsets for individual models, and also in the way they combine the predictions from the individual models to generate the final predictions. Polikar[11] presents a good review of ensemble methods.

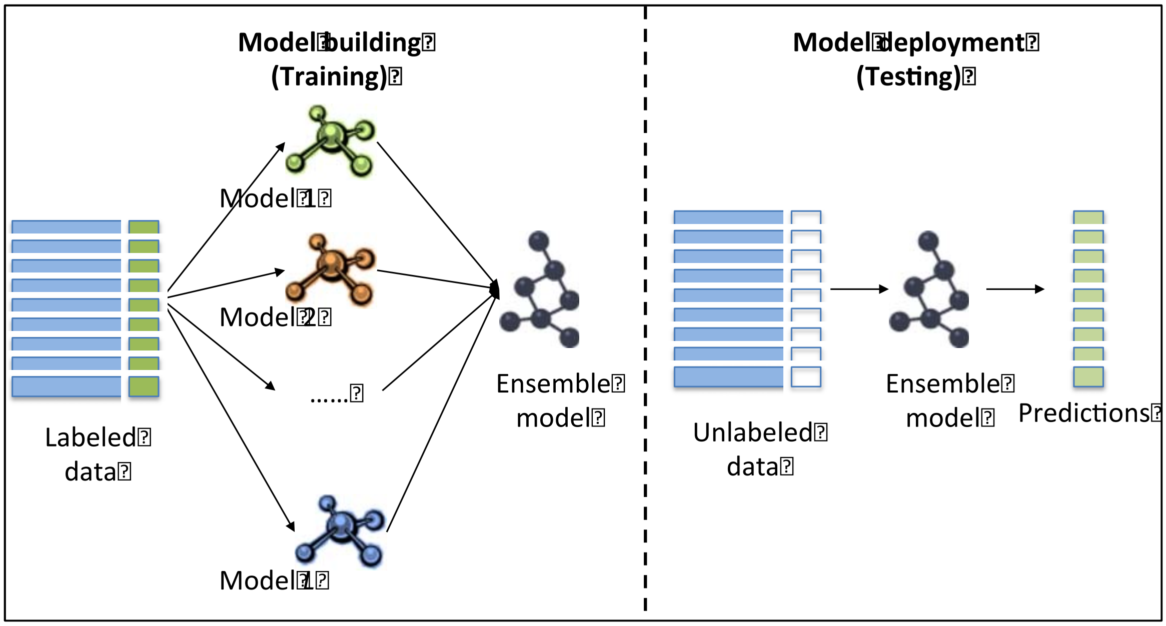

Figure S3. General schematic of ensemble predictive modeling.

The rotation forest approach uses feature extraction to create ensembles of either classification or regression learners. Here, we summarize the more complete discussion of Rodriguez and Kuncheva.[9] The training data for the base classifier is created by applying Principal Component Analysis (PCA) to *K* subsets of the feature set, followed by *K* axis rotations to form the new features for the base learner. Figure S4 depicts the rotation forest ensembling technique. The feature set *F* is split into *K* random subsets, and for each subset, a bootstrap sample is drawn with a smaller sample size, to which PCA is applied to obtain the principal component coefficients. The PCA vectors are arranged into a sparse rotation matrix, which is subsequently rearranged to match the order of features in *F*. Let the resulting rotation matrix be denoted by R_i^a. The base classification/regression modeling technique is then used to learn a model using $[XR_i^a\ Y]$ as the input training data, which is essentially a rotated version of the entire training data *X. Y* is the vector of class labels. This procedure is repeated *L* times with different random feature splits to generate *L* models. For testing, an unlabeled data instance x is processed as follows: the *L* rotated versions of *x* (i.e. xR_i^a, for $0 \le i \le L$) are used to get the individual predictions from the *L* models, which are averaged to get the final prediction for *x*.

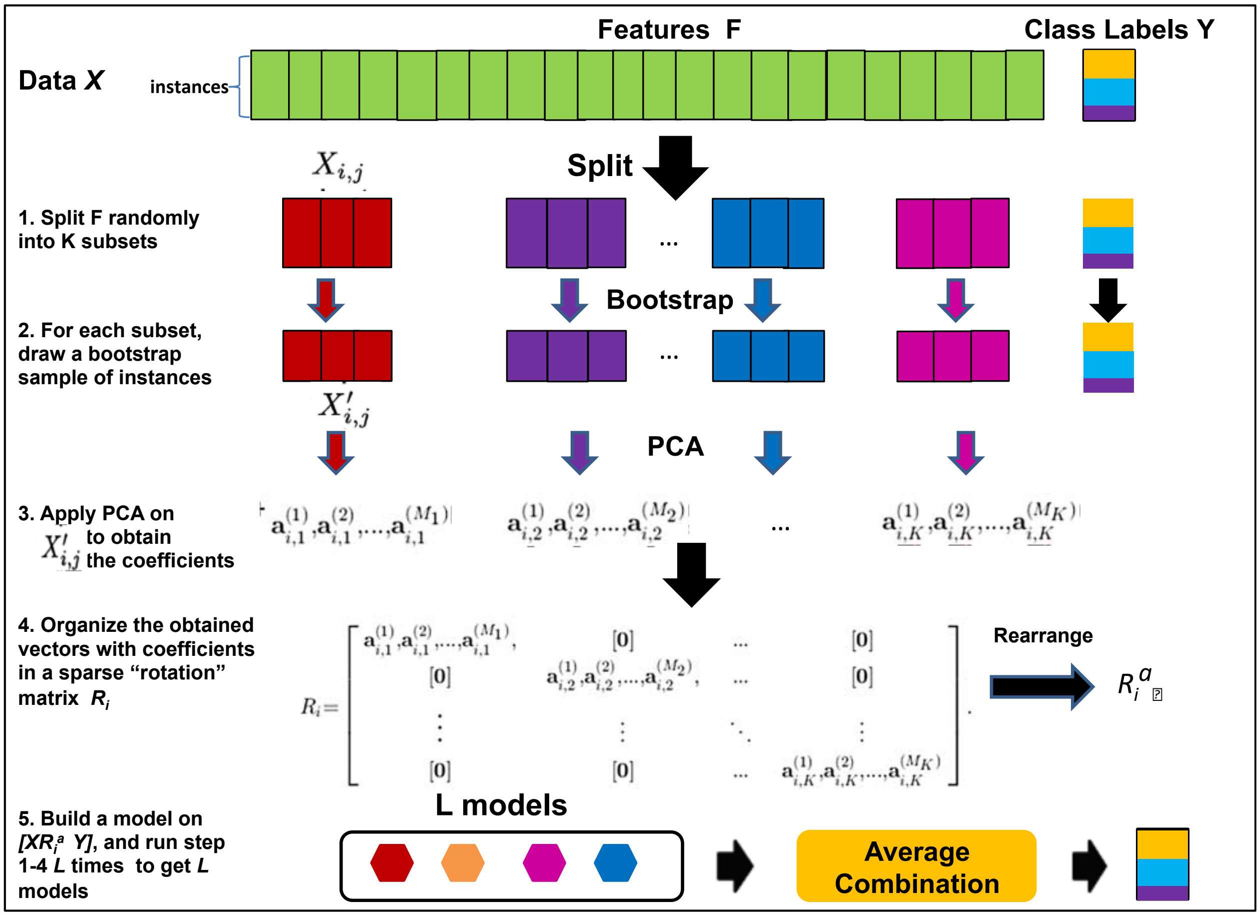
Features F
Class Labels Y
Data X
instances
$X_{i,j}$
Split
1. Split F randomly into K subsets
...
Bootstrap
2. For each subset, draw a bootstrap sample of instances
$X'_{i,j}$
PCA
3. Apply PCA on $X'_{i,j}$ to obtain the coefficients
$\mathbf{a}_{i,1}^{(1)},\mathbf{a}_{i,1}^{(2)},\ldots,\mathbf{a}_{i,1}^{(M_1)}$
$\mathbf{a}_{i,2}^{(1)},\mathbf{a}_{i,2}^{(2)},\ldots,\mathbf{a}_{i,2}^{(M_2)}$
$\mathbf{a}_{i,K}^{(1)},\mathbf{a}_{i,K}^{(2)},\ldots,\mathbf{a}_{i,K}^{(M_K)}$
4. Organize the obtained vectors with coefficients in a sparse "rotation" matrix R_i
$R_i = \begin{bmatrix} \mathbf{a}_{i,1}^{(1)},\mathbf{a}_{i,1}^{(2)},\ldots,\mathbf{a}_{i,1}^{(M_1)} & [0] & \cdots & [0] \\ [0] & \mathbf{a}_{i,2}^{(1)},\mathbf{a}_{i,2}^{(2)},\ldots,\mathbf{a}_{i,2}^{(M_2)} & \cdots & [0] \\ \vdots & \vdots & \ddots & \vdots \\ [0] & [0] & \cdots & \mathbf{a}_{i,K}^{(1)},\mathbf{a}_{i,K}^{(2)},\ldots,\mathbf{a}_{i,K}^{(M_K)} \end{bmatrix}$
Rearrange
R_i^a
5. Build a model on $[XR_i^a\ Y]$, and run step 1-4 L times to get L models
L models
Average Combination

Figure S4. Block diagram of the rotation forest ensembling technique.

Reduced Error Pruning Tree (REPTree)[12] is an efficient decision tree learner. A decision tree consists of internal nodes denoting the different attributes and the branches denoting the possible values of the attributes, while the leaf nodes indicate the final predicted value of the target variable. In general, a decision tree construction begins at the top of the tree (root node) with all of the data. At each node, splits are made according to an information gain criterion, which splits the data into corresponding branches. Computation on remaining nodes continues in the same manner until one of the stopping criteria is met, which include maximum tree depth, minimum number of instances in a leaf node, or minimum variance in a node. Such decision trees can be susceptible to over-fitting on the training data, for which REPTree uses reduced-error pruning. Part of the training data is withheld from decision tree construction as a pruning set and is subsequently used for pruning. At each internal node in the tree, it is possible to establish the error rate by propagating the errors upwards from the leaf nodes. This is compared to the error rate if that internal node was replaced by a leaf node with the average value of the target attribute in that node. If it results in a reduction of error, then the subtree below the node can be potentially pruned. This calculation is done for all the nodes, and one with highest reduced-error rate is pruned. This procedure is repeated on the resulting tree until there is no possibility of further error reduction. All error rates are computed using the pruning set.

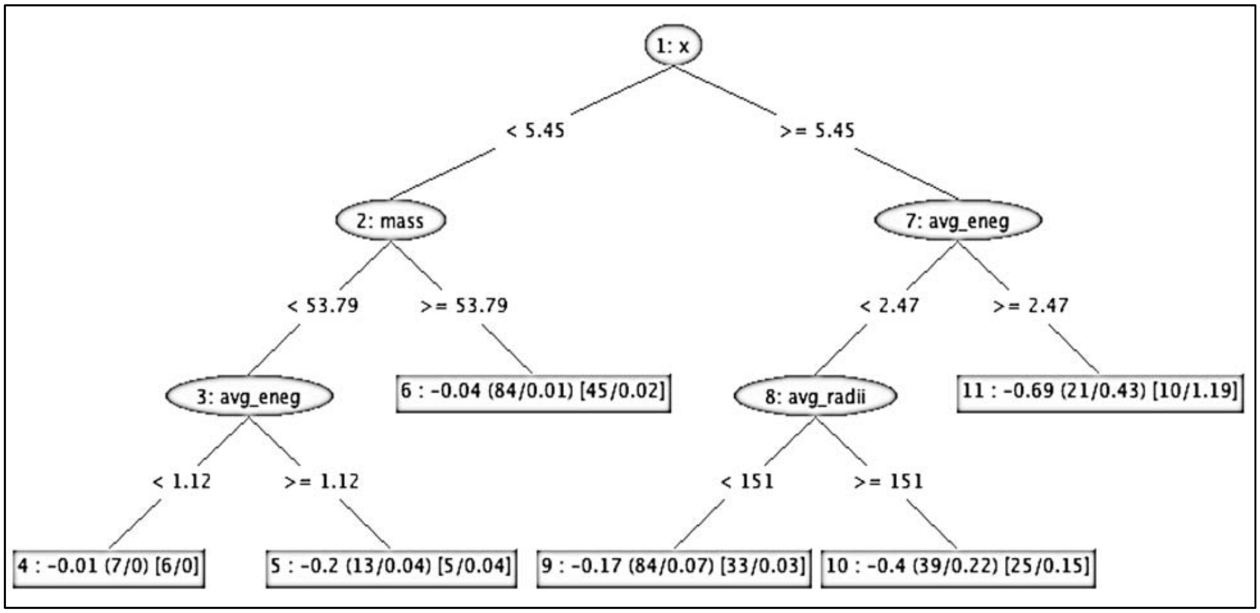

Figure S5. Illustrative representation of the REPTree on a small subset of our formation energy data.

Figure S5 shows an illustrative REPTree constructed on a small subset of the formation energy prediction database, consisting of only 372 instances and 6 predictor attributes. The internal nodes are depicted as elliptical and leaf nodes as rectangular. The internal nodes contain the splitting attribute, and the branches show the splitting condition. Each leaf node contains the predicted formation energy value for instances following the path up to that leaf node, along with the statistics in brackets based on the training set. The numbers in the first bracket correspond to the part of training data that was used to build the decision tree; the first number denotes the coverage, i.e., the number of instances that follow the path up to this leaf node, and the second number represents the error rate or average offset of the predicted value as compared to the actual value. The numbers in the second bracket denote the same thing for the part of training data that was used for pruning.

Model Evaluation

Ten-fold cross validation was used to evaluate model performance. The formation energy prediction database was randomly divided into 10 segments; nine segments were used for building the model and the remaining segment was used to test the model. This procedure is repeated 10 times with different test segments. In this way, all the instances in the dataset are tested exactly once using a model that did not see that instance while training.

The criteria that are employed for evaluation of models' predictive performances are the coefficient of correlation (*R*), Mean Absolute Error (*MAE*), and Root-Mean-Squared Error (*RMSE*) between the actual and predicted values. The definitions of these evaluation criteria are as follows:

$$R = \frac{\sum_{i=1}^{n}(y_i^t - \overline{y^t})(y_i^p - \overline{y^p})}{\sqrt{\sum_{i=1}^{n}(y_i^t - \overline{y^t})^2}\sqrt{\sum_{i=1}^{n}(y_i^p - \overline{y^p})^2}} \tag{1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^t - y_i^p}{y_i^t}\right| \tag{2}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i^t - y_i^p)^2}{n}} \tag{3}$$

Where y_i^t and y_i^p are the target and predicted formation energies, respectively, $\overline{y_i^t}$ and $\overline{y_i^p}$ are the mean of the target and predicted formation energies corresponding to *n* patterns. *R* is a measure of correlation between the predicted and the measured values and therefore, determines accuracy of the fitting model (higher *R* equates to higher accuracy). The *MAE* and *RMSE* are error measures with smaller errors indicative of better prediction accuracy. The WEKA data mining toolkit[13] was used for model construction and evaluation.

[1] A. Jain, G. Hautier, C. J. Moore, S. P. Ong, C. C. Fischer, T. Mueller, K. A. Persson, and G. Ceder, Comp Mater Sci **50**, 2295 (2011).

[2] S. Curtarolo, et al., Comp Mater Sci **58**, 227 (2012).

[3] S. Curtarolo, G. L. W. Hart, M. Buongiorno-Nardelli, N. Mingo, S. Sanvito, and O. Levy, Nat Mater **12**, 191 (2013).

[4] K. S. Yang, W. Setyawan, S. D. Wang, M. Buongiorno-Nardelli, and S. Curtarolo, Nat Mater **11**, 614 (2012).

[5] H. Chen, et al., Chem Mater **24**, 2009 (2012).

[6] A. Jain, G. Hautier, S. Ong, C. Moore, C. C. Fischer, K. Persson, and G. Ceder, Phys Rev B **84**, 045115 (2011).

[7] M. Hillert, *Phase Equilibria, Phase Diagrams and Phase Transformations: Their Thermodynamic Basis* (Cambridge University Press, Cambridge, 2008).

[8] Y. M. Muggianu, M. Gambino, and J. P. Bros, J Chim Phys Pcb **72**, 83 (1975).

[9] J. J. Rodriguez and L. I. Kuncheva, Ieee T Pattern Anal **28**, 1619 (2006).

[10] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, Ieee T Pattern Anal **20**, 226 (1998).

[11] R. Polikar, IEEE Circuits and Systems Magazine **6**, 21 (2006).

[12] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques* (Morgan Kaufmann, 2005).

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, ACM SIGKDD Explorations Newsletter **11**, 10 (2009).