Distilling Free-Form Natural Laws from Experimental Data

Michael Schmidt[1] and Hod Lipson[2,3]*

For centuries, scientists have attempted to identify and document analytical laws that underlie physical phenomena in nature. Despite the prevalence of computing power, the process of finding natural laws and their corresponding equations has resisted automation. A key challenge to finding analytic relations automatically is defining algorithmically what makes a correlation in observed data important and insightful. We propose a principle for the identification of nontriviality. We demonstrated this approach by automatically searching motion-tracking data captured from various physical systems, ranging from simple harmonic oscillators to chaotic double-pendula. Without any prior knowledge about physics, kinematics, or geometry, the algorithm discovered Hamiltonians, Lagrangians, and other laws of geometric and momentum conservation. The discovery rate accelerated as laws found for simpler systems were used to bootstrap explanations for more complex systems, gradually uncovering the "alphabet" used to describe those systems.

Mathematical symmetries and invariants underlie nearly all physical laws in nature (*1*), suggesting that the search for many natural laws is inseparably a search for conserved quantities and invariant equations (*2*, *3*). Automated techniques for generating, collecting, and storing data from scientific measurements have become increasingly precise and powerful, but automated processes for distilling this data into knowledge in the form of analytical natural laws have not kept pace. Thus, there is a pressing practical need (*4*, *5*) for improved forms of scientific data mining (*6*, *7*).

The most prohibitive obstacle to overcome in order to search for conservation laws computationally is finding meaningful and nontrivial invariants. There exist an infinite number of identities that are numerically invariant but have no connection to the natural physics or dynamics of the system. We introduce a principle for identifying only the useful analytical relations that are related to the system dynamics. We then demonstrate how a search algorithm based on this principle identifies meaningful analytical links in data captured from various physical systems (Fig. 1).

Our goal is to find natural relations where they exist, with minimal restrictions on their analytical form (i.e., free-form). Many methods exist for modeling scientific data: Some use fixed-form parametric models derived from expert knowledge, and others use numerical models (such as neural networks) aimed at prediction. Still others have explored restricted model spaces using greedy monomial search (*8*, *9*). Alternatively, we seek the principal unconstrained analytical expression that explains symbolically precise conserved relations, thus helping distill data into scientific knowledge.

Symbolic regression (*10*) is an established method based on evolutionary computation (*11*) for searching the space of mathematical expressions while minimizing various error metrics [see section S4 in the supporting online material (SOM)]. Unlike traditional linear and nonlinear regression methods that fit parameters to an equation of a given form, symbolic regression searches both the parameters and the form of equations simultaneously (see SOM section S6). Initial expressions are formed by randomly combining mathematical building blocks such as algebraic operators {+, –, ÷, ×}, analytical functions (for example, sine and cosine), constants, and state variables. New equations are formed by recombining previous equations and probabilistically varying their subexpressions. The algorithm retains equations that model the experimental data better than others and abandons unpromising solutions. After equations reach a desired level of accuracy, the algorithm terminates, returning a set of equations that are most likely to correspond to the intrinsic mechanisms underlying the observed system.

Although symbolic regression is typically used to find explicit (*12*–*14*) and differential equations (*15*), this method cannot readily find conservation laws or invariant equations. Rather than trying to model a specific signal, we are trying to detect any underlying physical law that the system obeys, which may or may not be constant (e.g., a Lagrangian).

A particular challenge is requiring the law to be a function of the system's state while avoiding trivial or meaningless relations. For any system over the state space x, there are infinitely many trivial equations over x that satisfy a conserved quantity, such as $\sin^2(x_1) + \cos^2(x_1)$ or $x_1 + 4.56 - x_2 x_1 / x_2$. Additionally, there are infinitely many arbitrarily close trivial conservations, such as $4.56 + 1/(100 + x_1^2)$. To distinguish good conservation law candidates from poor ones, we need a more robust principle than simply invariance alone.

The identification of nontrivial relations is a major challenge, even for human scientists: Many published invariant quantities have turned out to be coincidental (*16*). The mere appearance of a conserved value is insufficient for a conservation

[1]Computational Biology, Cornell University, Ithaca, NY 14853, USA. [2]School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA. [3]Computing and Information Science, Cornell University, Ithaca, NY 14853, USA.

*To whom correspondence should be addressed. E-mail: hod.lipson@cornell.edu

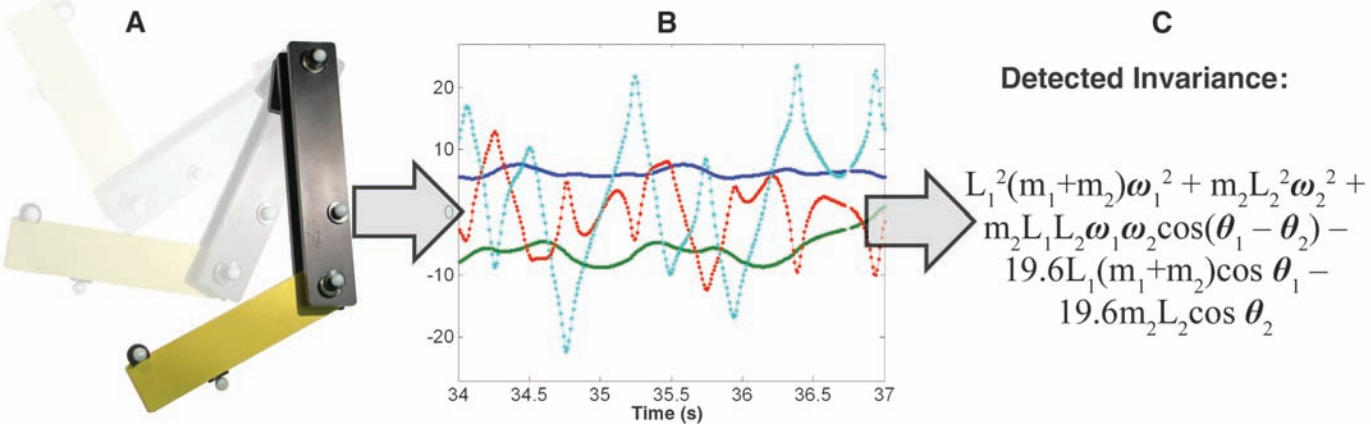

Fig. 1. Mining physical systems. We captured the angles and angular velocities of a chaotic double-pendulum (**A**) over time using motion tracking (**B**), then we automatically searched for equations that describe a single natural law relating these variables. Without any prior knowledge about physics or geometry, the algorithm found the conservation law (**C**), which turns out to be the double pendulum's Hamiltonian. Actual pendulum, data, and results are shown.

law. The key insight into identifying nontrivial conservation laws computationally is that the candidate equations should predict connections between dynamics of subcomponents of the system. More precisely, the conservation equation should be able to predict connections among derivatives of groups of variables over time, relations that we can also readily calculate from new experimental data.

One instance of such a metric is the partial derivatives between pairs of variables (see SOM section S1). For example, in a two-dimensional system we could measure variables $x(t)$ and $y(t)$ over time. The system's partial derivatives estimated from time-series data would then be $x'/y' \approx \Delta x/\Delta y$ and $y'/x' \approx \Delta y/\Delta x$ (where x' and y' represent the time derivatives of x and y). Similarly, given a candidate conservation law equation $f(x,y)$, we can derive the same values through differentiation: $(\delta f/\delta y)/(\delta f/\delta x) \approx \delta x/\delta y$ and $(\delta f/\delta x)/(\delta f/\delta y) \approx \delta y/\delta x$. We can now compare $\Delta x/\Delta y$ values from the experimental data with $\delta x/\delta y$ values from a candidate conservation expression $f(x,y)$ to measure how well it predicts intrinsic relations in the system. In higher-dimensional systems, multiple variable pairings and higher-order derivatives yield a plethora of criteria to use. See SOM sections S2 and S3 for generalization to higher-dimensional systems. Using the partial-derivative pairs, we define a new type of search criteria for measuring how well a candidate analytical expression represents a nontrivial invariance over the experimental data.

An important consequence of the partial-derivative–pair measure is that it can also identify relations that represent other nontrivial identities of the system beyond invariants and conservation laws. For example, if the system is confined to a manifold, the manifold equation can also derive accurate partial-derivative pairs. Similarly, the partial-derivative pair can identify equations such as Lagrangian equations, the energy equivalent to the equation of motion in classical mechanics, which summarize the systems dynamics but are not invariant.

One can control, to an extent, the type of law that the system might find by choosing what variables to provide to the algorithm. For example, if we only provide position coordinates, the algorithm is forced to converge on a manifold equation of the system's state space. If we provide velocities, the algorithm is biased to find energy laws. If we additionally supply accelerations, the algorithm is biased to find force identities and equations of motion. However, given these or other types of variables, other or previously unknown analytical laws may exist.

We used an algorithm (Fig. 2) to search for analytical laws in data captured from several synthetic and physical systems using various sets of system variables. We present results for a number of physical experimental systems (see SOM section S7 for a study of synthetic systems, geometric symmetries, and manifolds). A video is available online (see SOM section S14).

We collected data from standard experimental systems typically used in undergraduate physics education: an air-track oscillator and a double pendulum (Fig. 3). We used motion-tracking software to record the devices' positions over time. We then numerically calculated velocities and accelerations (see SOM section S11). All data sets are available in SOM section S15.

Without any additional information, system models, or theoretical knowledge, the search with the partial-derivative–pairs criterion produced several analytical law expressions directly from these data. For each system, the algorithm outputs a short list of ~10 equations that have maximal accuracy found for different sizes (complexities) of equations (see SOM section S8). We then inspect this list manually to select the final equation. Often the list consists of varying approximations or elaborations on a particular law equation, but it can contain qualitatively different equations, as discussed below.

We experimented on two configurations of the air track: (i) two-spring single-mass and (ii) three-spring double-mass. Similarly, we collected time-series data from a pendulum and a double pendulum (Fig. 3) with the use of motion-tracking (SOM section S12).

The single-car air track is a harmonic oscillator with slight damping from the air and its two springs. With only minimal noise and damp-

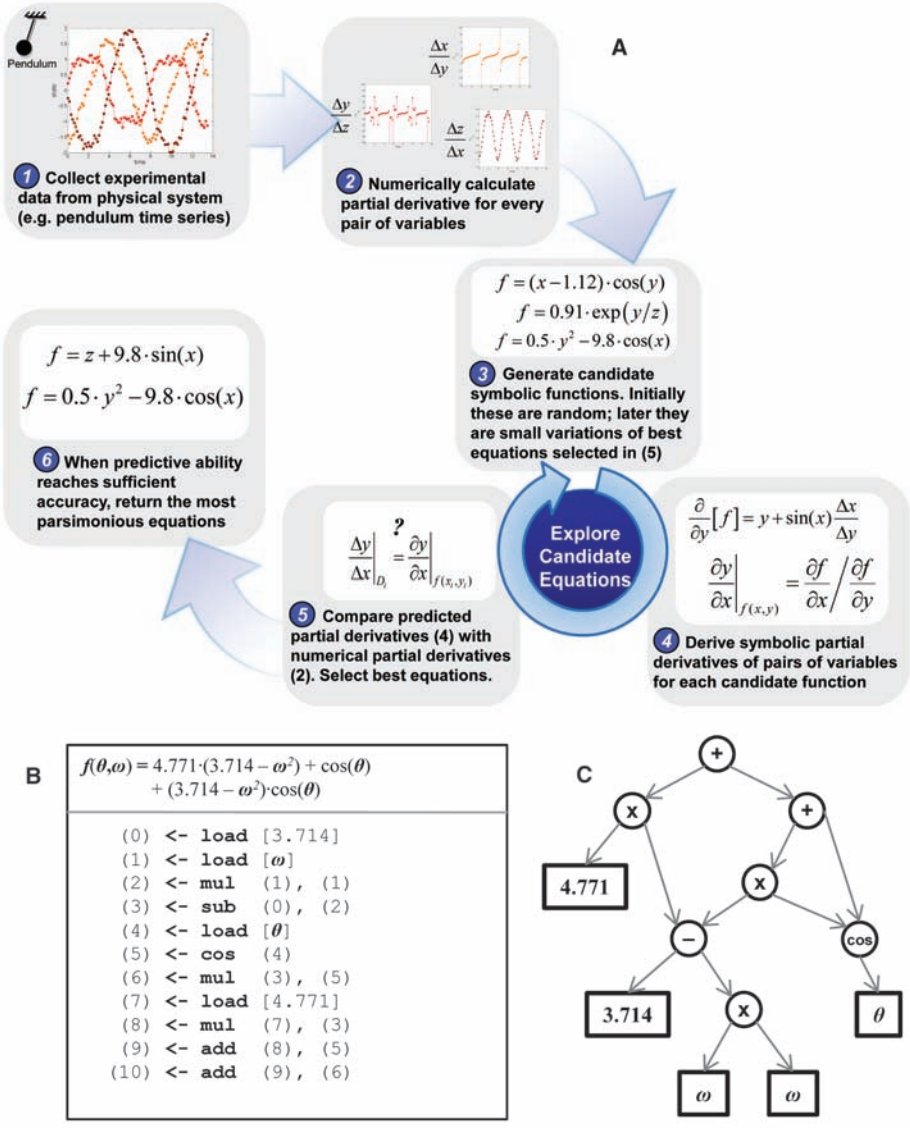

Fig. 2. Computational approach for detecting conservation laws from experimentally collected data. (**A**) First, calculate partial derivatives between variables from the data, then search for equations that may describe a physical invariance. To measure how well an equation describes an invariance, derive the same partial derivatives symbolically to compare with the data. (**B**) The representation of a symbolic equation in computer memory is a list of successive mathematical operations (see SOM section S6). (**C**) This list representation corresponds to a graph, where nodes represent mathematical building blocks and leaves represent parameters and system variables. Both (B) and (C) correspond to the same equation. The algorithm varies these structures to search the space of equations.

ing, it was the simplest physical system that we examined. The double-mass air track consisted of two coupled harmonic oscillators of different masses. There was considerable noise in this data set as a result of compression of the middle spring. The pendulum is a nonlinear oscillator that is masked by small-angle approximations. The double pendulum is a coupled nonlinear oscillator system that exhibits rich dynamics (*17*) and chaos at certain energies (*18*), making it challenging to model (*19*, *20*). Additionally, there is higher measurement noise and dampening errors due to higher velocities.

Given position and velocity data over time, the algorithm converged on the energy laws of each system (Hamiltonian and Lagrangian equations). Given acceleration data also, it produced the differential equation of motion corresponding to Newton's second law for the harmonic oscillator and pendulum systems. Given only position data for the pendulum, the algorithm converged on the equation of a circle, indicating that the pendulum is confined to a circle. The algorithm also produced several inexact expressions through small-angle approximations—for example, using x in place of $\sin(x)$ and $1 - x^2$ in place of $\cos(x)$ in the pendulum and double-pendulum systems.

An interesting approximate law for the double pendulum that emerged was conservation of angular momentum. Given only data measured while the pendulum was chaotic (at high energy), the algorithm fixated on this law. The conservation of momentum equation is simpler than other valid laws and is approximately correct for high velocities where gravity is negligible, as with the high-energy chaotic data set.

Similarly, given only data from low-velocity in-phase oscillations, the algorithm fixated on small-angle approximations and uncoupled energy terms. By combining the chaotic data with low-velocity in-phase oscillation data, the algorithm converged onto the precise energy laws after several hours of computation.

In the absence of appropriate building blocks, the algorithm developed approximations. For example, eliminating the sine and cosine operations from the set of equation building blocks caused the pendulum invariant to be expressed as $\omega^2 + k_1\theta^2 - k_2\theta^4$ (where θ is the pendulum's angle, ω is the angular velocity, and k_1 and k_2 are constants), thereby exploiting the 4th-order Taylor series expansion of the cosine function. Eliminating cosine but not sine drove the algorithm to converge on the equality $\cos(\theta) = \sin(\theta + \pi/2)$ or more complex equivalences (see SOM section S13).

Useful scientific theory is both predictive and parsimonious. Similarly, some equations may be more accurate but overfit the data, whereas others may be more parsimonious but oversimplify

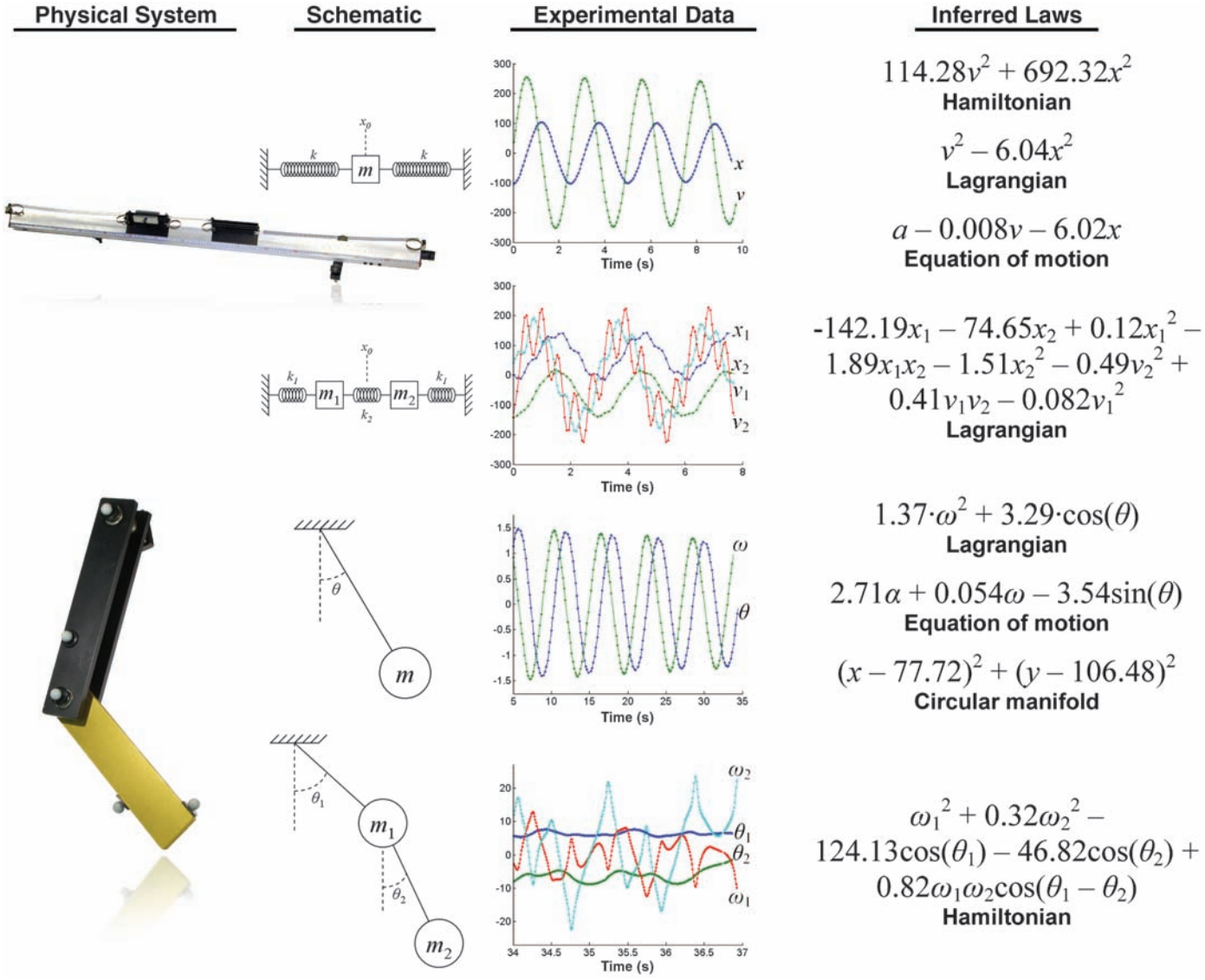

Fig. 3. Summary of laws inferred from experimental data collected from physical systems. Depending on the types of variables provided to the algorithm, it detects different types of laws. Given solely position information, the algorithm detects position manifolds; given velocities, the algorithm detects energy laws; given accelerations, it detects equations of motion and sum of forces laws (θ, angle; ω, angular velocity; α, angular acceleration).

(*21*, *22*); the right balance is difficult to specify in advance. Instead of producing a single result, the algorithm produces a small set of final candidate analytical expressions on the accuracy-parsimony Pareto front (see SOM section S8), which represents the optimal solutions as they vary over equation complexity and the maximum predictive ability. Parsimony is measured as the inverse of the number of terms in the expression, whereas the predictive accuracy is measured as error on withheld experimental data used only for validation.

The Pareto front for the double pendulum (Fig. 4A) reveals a few particularly simple equations that predict the partial-derivative pairs accurately. Predictive accuracy was measured by cross-validation with the partial-derivative–pairs criterion (see SOM section S2). The Pareto front tends to contain a cliff where predictive ability jumps rapidly at some minimum complexity. Predictive ability then improves only marginally with more complex equations (Fig. 4A). The conservation of angular momentum equation lies on the Pareto front, though it is inexact. The double pendulum's Hamiltonian lies at the point representing the simplest equation with the largest increase in predictive ability. In all of our experiments, the solution at this point has been an exact theoretical law (see SOM section S7 for additional systems).

In the worst case, the time to converge on the law equations depends exponentially on the complexity of the law expression itself and roughly quadratically on the system dimensionality (the number of variable pairings) (Fig. 4B). The algorithm's search is readily parallelizable, as many candidate functions need to be evaluated simultaneously. In a 32-core implementation, the time required ranged from a few minutes for the harmonic oscillator to 30 hours for the double pendulum. The impact of noise also couples with these factors (see SOM section S9). For comparison, the simulated double-mass air-track and simulated double-pendulum data sets (where measurements are noiseless) took ~1/10th of the computational effort to analyze. A summary of performance versus noise level is provided in SOM section S9.

Though the algorithm can present equations corresponding to physical laws in their mathematical form, we are still faced with the challenge of justifying and giving words to their meaning. One difficulty is that we cannot know with certainty the units of bulk constants in the law expressions (for example, combinations of masses, lengths, etc. embodied in the system). Second, the equation may model something that is inherently difficult to observe directly, such as total energy. Requiring equations to maintain consistent physical units still leaves room for ambiguity.

A more systematic approach to parsing the coefficients is to analyze multiple data sets from the same systems, albeit with different configurations and parameters. To demonstrate this approach, we used several virtual double pendula with randomly chosen masses and lengths to generate several synthetic data sets. We fit the free coefficients of the automatically discovered model to each data set and then invoked the equation search algorithm again to seek a relation between the coefficients and parameter sets. After arbitrarily defining $k_1 = 1$, the algorithm identified that $k_2 = m_2L_2^2/(m_1L_1^2 + m_2L_1^2)$, $k_3 = 2m_2L_2/(m_1L_1 + m_2L_1)$, $k_3 = 19.6/L_1$, and $k_4 = 19.6m_2L_2/(m_2L_1^2 + m_1L_1^2)$, where 19.6 is the only absolute constant (over all parameter variations) whose units are necessarily meters per square seconds (see SOM section S5). In the above expressions, m is mass and L is length. A similar approach can be used to identify coefficients that vary slowly over time (for example, because of damping, creeping, or ecological drift).

Computational systems such as this could play a role in modeling high-dimensional and complex phenomena (*23*, *24*) that currently stress the reach of expert-driven research. A key challenge is scaling to higher complexity. To accomplish this, scientists leverage knowledge from simpler systems to explain more complex systems. Can an algorithm do this as well?

One method to use prior knowledge is seeding the equation search by initializing the algorithm's initial set of candidate expressions with terms from equations from simpler systems. For example, the single-pendulum and the double-harmonic oscillator equations provide clues to the laws governing the more complex double pendulum. We shuffled terms of the simpler systems (for example, exchanging velocity symbols with double-pendulum velocity variables) and randomized parameters to generate many inexact initial expressions. This seeding approach does not constrain the equation search, but it biases the search to reuse terms from previous laws.

Bootstrapping the double-pendulum search in this fashion reduced the search time by nearly an order of magnitude, from 30 to 40 hours of computation to 7 to 8 hours (Fig. 4B). On the basis of this result, we conjecture that bootstrapping may be critical for detecting laws in higher-order systems that are veiled in complexity.

A statistical analysis of the subexpression frequency and complexity across populations of various physical systems revealed that the terms that are both frequently used and complex tend to be more physically meaningful, such as trigonometric terms representing potential energies, squared velocities representing kinetic energies, or linear force combinations (see SOM section S10). These terms may make up an "emergent alphabet" for describing a range of systems, which could accelerate their modeling and simplify their conceptual understanding.

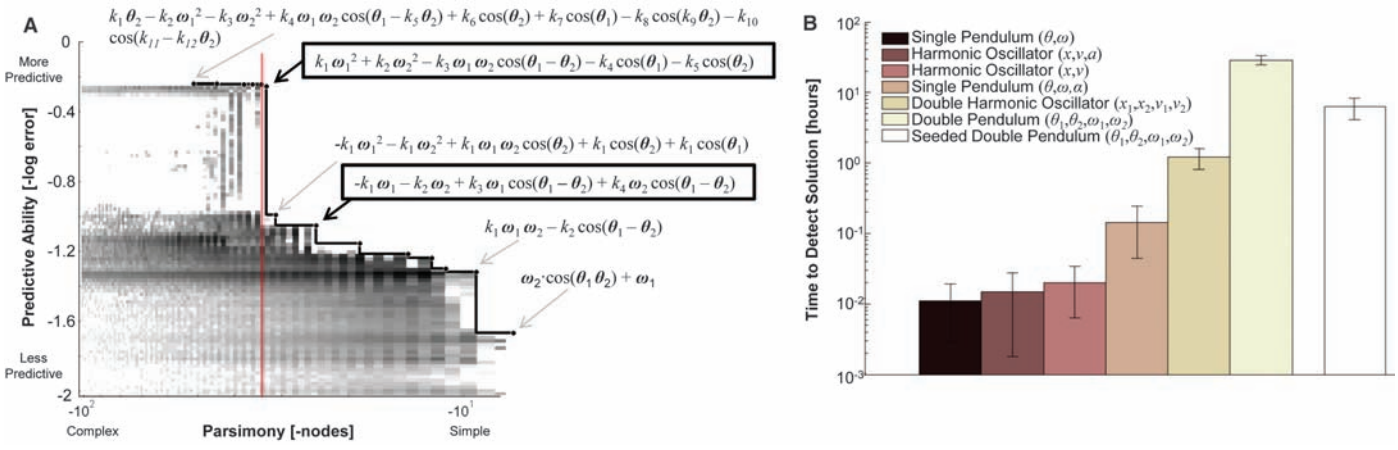

Fig. 4. Parsimony versus accuracy and computation time. (**A**) Pareto front (solid black curve) for physical laws of the double pendulum and the frequency of sampling during the law equation search (grayscale). The equation at the cliff corresponds to the exact energy conservation law of the double pendulum (highlighted in the figure). A second momentum conservation law that we encountered is also highlighted. (**B**) Computation time required to detect different physical laws for several systems. The computation time increases with the dimensionality, law equation complexity, and noise. A notable exception is the bootstrapped double pendulum, where reuse of terms from simpler systems helped reduce computational cost by almost an order of magnitude, suggesting a mechanism for scaling higher complexities.

We have demonstrated the discovery of physical laws, from scratch, directly from experimentally captured data with the use of a computational search. We used the presented approach to detect nonlinear energy conservation laws, Newtonian force laws, geometric invariants, and system manifolds in various synthetic and physically implemented systems without prior knowledge about physics, kinematics, or geometry. The concise analytical expressions that we found are amenable to human interpretation and help to reveal the physics underlying the observed phenomenon. Many applications exist for this approach, in fields ranging from systems biology to cosmology, where theoretical gaps exist despite abundance in data.

Might this process diminish the role of future scientists? Quite the contrary: Scientists may use processes such as this to help focus on interesting phenomena more rapidly and to interpret their meaning.

References and Notes

1. P. W. Anderson, *Science* **177**, 393 (1972).
2. E. Noether, *Nachr. d. König Gesellsch. d. Wiss. zu Göttingen, Math-Phys. Klasse* 235 (1918).
3. J. Hanc, S. Tuleja, M. Hancova, *Am. J. Phys.* **72**, 428 (2004).
4. D. Clery, D. Voss, *Science* **308**, 809 (2005).
5. A. Szalay, J. Gray, *Nature* **440**, 413 (2006).
6. R. E. Valdés-Pérez, *Commun. Assoc. Comput. Mach.* **42**, 37 (1999).
7. R. D. King *et al.*, *Nature* **427**, 247 (2004).
8. P. Langley, *Cogn. Sci.* **5**, 31 (1981).
9. R. M. Jones, P. Langley, *Comput. Intell.* **21**, 480 (2005).
10. J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. (MIT Press, Cambridge, MA, 1992).
11. S. Forrest, *Science* **261**, 872 (1993).
12. J. Duffy, J. Engle-Warnick, *Evolutionary Computation in Economics and Finance* **100**, 61 (2002).
13. F. Cyril, B. Alberto, in *2007 IEEE Congress on Evolutionary Computation*, S. Dipti, W. Lipo, Eds. (IEEE Press, Singapore, 2007), pp. 23–30.
14. B. Elena, B. Andrei, L. Henri, in *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC '05)* (IEEE Press, 2005), pp. 321–324.
15. J. Bongard, H. Lipson, *Proc. Natl. Acad. Sci. U.S.A.* **104**, 9943 (2007).
16. S. Nee, N. Colegrave, S. A. West, A. Grafen, *Science* **309**, 1236 (2005).
17. P. Jäckel, T. Mullin, *Proc. R. Soc. London Ser. A* **454**, 3257 (1998).
18. T. Shinbrot, C. Grebogi, J. Wisdom, J. A. Yorke, *Am. J. Phys.* **60**, 491 (1992).
19. Y. Liang, B. Feeny, *Nonlinear Dyn.* **52**, 181 (2008).
20. M. Mor, A. Wolf, O. Gottlieb, in *Proceedings of the 21st ASME Biennial Conference on Mechanical Vibration and Noise* (ASME Press, Las Vegas, NV, 2007), pp. 1–8.
21. P. Gregory, R. Denis, F. Cyril, in *Evolution Artificielle, 6th International Conference*, vol. 2936, L. Pierre, C. Pierre, F. Cyril, L. Evelyne, S. Marc, Eds. (Springer, Marseilles, France, 2003), pp. 267–277.
22. E. D. De Jong, J. B. Pollack, in *Genetic Programming and Evolvable Machines*, vol. 4 (Springer, Berlin, 2003), pp. 211–233.
23. S. H. Strogatz, *Nature* **410**, 268 (2001).
24. P. A. Marquet, *Nature* **418**, 723 (2002).

25. This research was supported in part by Integrative Graduate Education and Research Traineeship program in nonlinear systems, a U.S. NSF graduate research fellowship, and NSF Creative-IT grant 0757478 and CAREER grant 0547376. We thank M. Kurman for editorial consultation and substantive editing of the manuscript.

Supporting Online Material

www.sciencemag.org/cgi/content/full/324/5923/81/DC1
Materials and Methods
SOM Text
Figs. S1 to S7
Tables S1 to S3
References
Movie S1
Data Sets S1 to S15

15 September 2008; accepted 19 February 2009
10.1126/science.1165893

The Automation of Science

Ross D. King,[1]* Jem Rowland,[1] Stephen G. Oliver,[2] Michael Young,[3] Wayne Aubrey,[1] Emma Byrne,[1] Maria Liakata,[1] Magdalena Markham,[1] Pınar Pir,[2] Larisa N. Soldatova,[1] Andrew Sparkes,[1] Kenneth E. Whelan,[1] Amanda Clare[1]

The basis of science is the hypothetico-deductive method and the recording of experiments in sufficient detail to enable reproducibility. We report the development of Robot Scientist "Adam," which advances the automation of both. Adam has autonomously generated functional genomics hypotheses about the yeast *Saccharomyces cerevisiae* and experimentally tested these hypotheses by using laboratory automation. We have confirmed Adam's conclusions through manual experiments. To describe Adam's research, we have developed an ontology and logical language. The resulting formalization involves over 10,000 different research units in a nested treelike structure, 10 levels deep, that relates the 6.6 million biomass measurements to their logical description. This formalization describes how a machine contributed to scientific knowledge.

Computers are playing an ever-greater role in the scientific process (*1*). Their use to control the execution of experiments contributes to a vast expansion in the production of scientific data (*2*). This growth in scientific data, in turn, requires the increased use of computers for analysis and modeling. The use of computers is also changing the way that science is described and reported. Scientific knowledge is best expressed in formal logical languages (*3*). Only formal languages provide sufficient semantic clarity to ensure reproducibility and the free exchange of scientific knowledge. Despite the advantages of logic, most scientific knowledge is expressed only in natural languages. This is now changing through developments such as the Semantic Web (*4*) and ontologies (*5*).

A natural extension of the trend to ever-greater computer involvement in science is the concept of a robot scientist (*6*). This is a physically implemented laboratory automation system that exploits techniques from the field of artificial intelligence (*7*–*9*) to execute cycles of scientific experimentation. A robot scientist automatically originates hypotheses to explain observations, devises experiments to test these hypotheses, physically runs the experiments by using laboratory robotics, interprets the results, and then repeats the cycle.

High-throughput laboratory automation is transforming biology and revealing vast amounts of new scientific knowledge (*10*). Nevertheless, existing high-throughput methods are currently inadequate for areas such as systems biology. This is because, even though very large numbers of experiments can be executed, each individual experiment cannot be designed to test a hypothesis about a model. Robot scientists have the potential to overcome this fundamental limitation.

The complexity of biological systems necessitates the recording of experimental metadata in as much detail as possible. Acquiring these metadata has often proved problematic. With robot scientists, comprehensive metadata are produced as a natural by-product of the way they work. Because the experiments are conceived and executed automatically by computer, it is possible to completely capture and digitally curate all aspects of the scientific process (*11*, *12*).

To demonstrate that the robot scientist methodology can be both automated and be made effective enough to contribute to scientific knowledge, we have developed Robot Scientist "Adam" (*13*) (Fig. 1). Adam's hardware is fully automated such that it only requires a technician to periodically add laboratory consumables and to remove waste. It is designed to automate the high-throughput execution of individually designed microbial batch growth experiments in microtiter plates (*14*). Adam measures growth curves (phenotypes) of selected microbial strains (genotypes) growing in defined media (environments). Growth of cell cultures can be easily measured in high-throughput, and growth curves are sensitive to changes in genotype and environment.

We applied Adam to the identification of genes encoding orphan enzymes in *Saccharomyces cerevisiae*: enzymes catalyzing biochemical reactions thought to occur in yeast, but for which the encoding gene(s) are not known (*15*). To set up Adam for this application required (i) a comprehensive logical model encoding knowledge of *S. cerevisiae* metabolism [~1200 open

[1]Department of Computer Science, Aberystwyth University, SY23 3DB, UK. [2]Cambridge Systems Biology Centre, Department of Biochemistry, University of Cambridge, Sanger Building, 80 Tennis Court Road, Cambridge CB2 1GA, UK. [3]Institute of Biological, Environmental and Rural Sciences, Aberystwyth University, SY23 3DD, UK.

*To whom correspondence should be addressed. E-mail: rdk@aber.ac.uk