A deep generative model for astronomical images of galaxies

Jeffrey Regier[1], Jon McAuliffe[1], Prabhat[2]

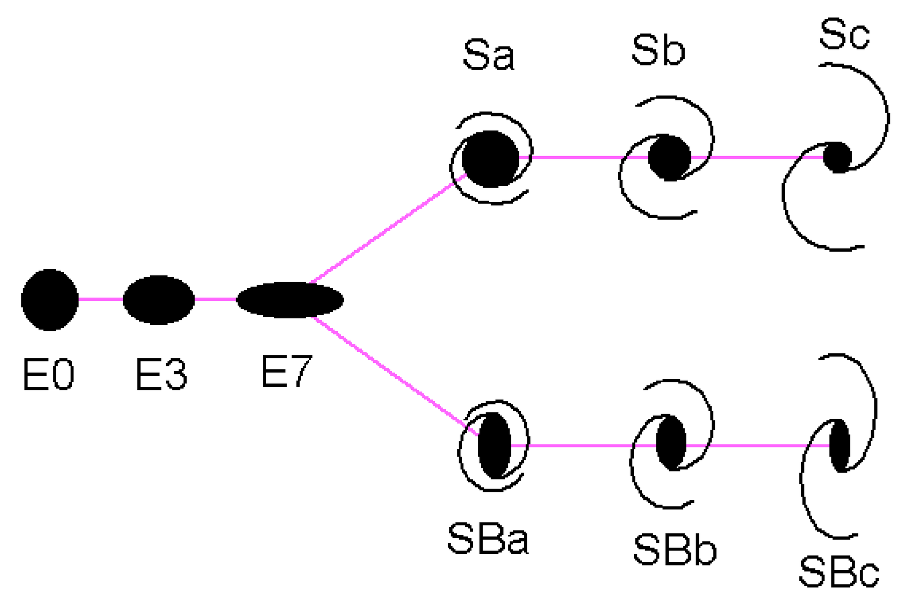
[1]Department of Statistics, University of California, Berkeley
[2]Lawrence Berkeley National Laboratory

Abstract

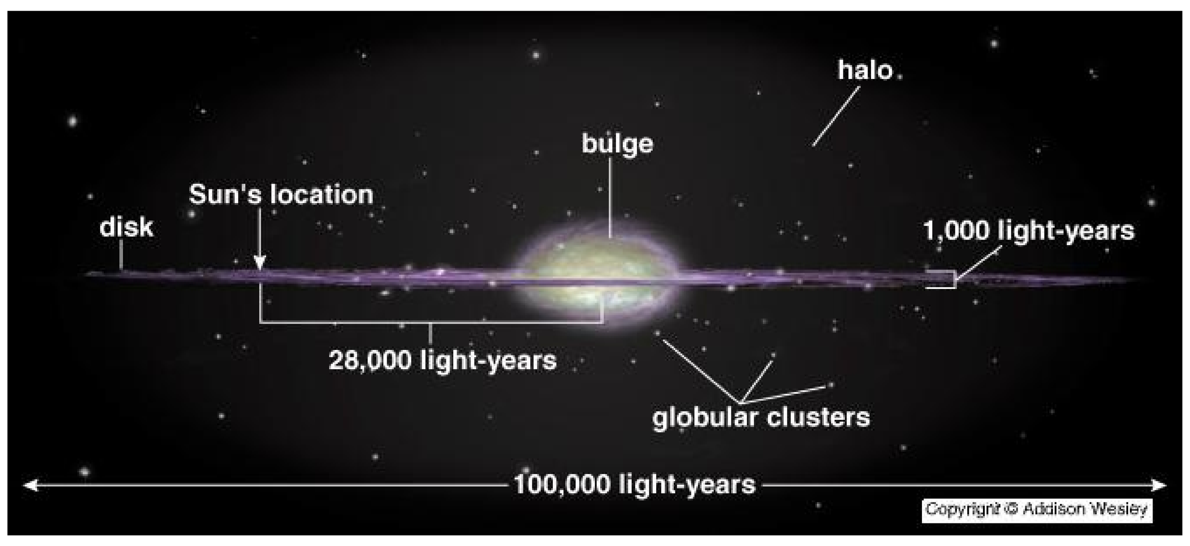
Simple parametric models suffice to describe many idealized galaxy shapes, but they severely misfit actual galaxies: they are not flexible enough. Nevertheless, fitting galaxies with simple parametric models is the current standard practice.

We propose a flexible Bayesian model for images of galaxies. For each image, a neural network maps the latent variables to the conditional distribution of pixels' intensities. We use variational inference to learn the parameters of the model from a large collection of training images. The proposed model fits held-out data more closely than the current standard practice and estimates model uncertainty.

Idealized galaxies



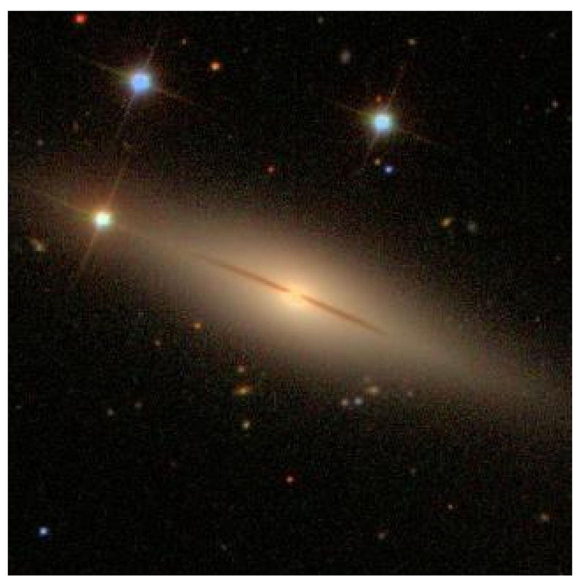
The Hubble "tuning fork" of galaxy morphology
Credit: Todd Thompson.



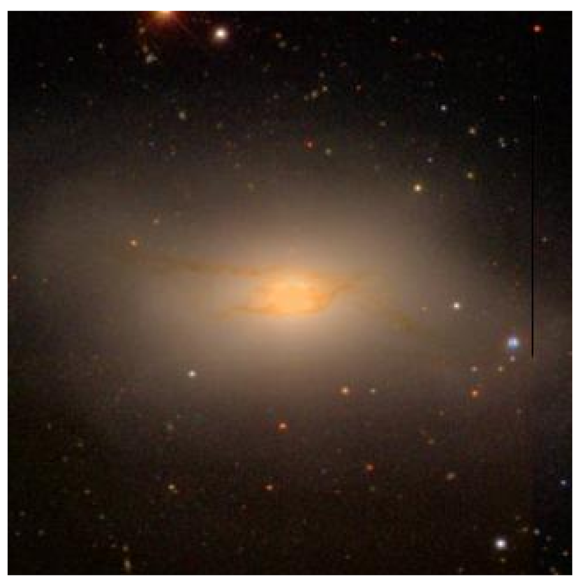
Milky Way galaxy
Credit: http://pics-about-space.com/milky-way-galaxy-halo

Actual galaxies

The Whirlpool galaxy—a classic spiral galaxy. credit: ESA Hubble / NASA.

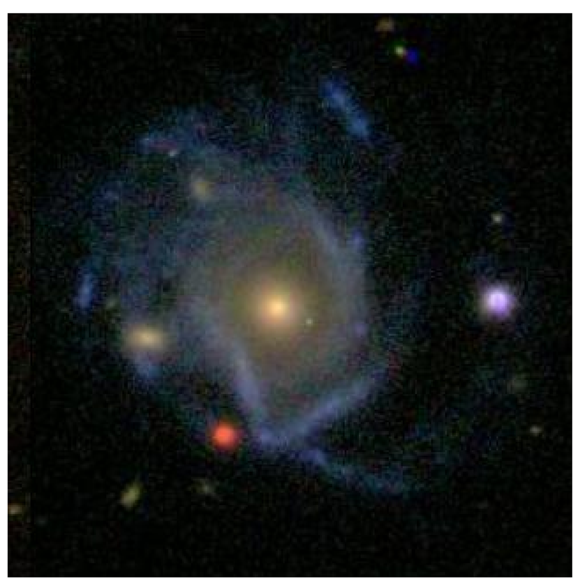


NGC 1032, an edge-on galaxy. credit: SDSS



NGC 4753, an elliptical galaxy with interesting dust filaments. credit: SDSS

An irregular galaxy. credit: NASA



NGC 60, a spiral galaxy with unusually distorted arms. credit: SDSS

The Sombrero galaxy, halfway between a spiral and an elliptical. credit: ESA Hubble / NASA

The model

For a particular image x of a galaxy, let z be a low-dimensional latent random vector, distributed as a multivariate standard normal. Given z, we model the observed intensities of the image's pixels $x = (x_1, \dots, x_m)$, as

$$x|z \sim \mathcal{N}(f_\mu(z), f_\sigma(z)).$$

We take the deterministic functions f_μ and f_σ to be neural networks that share some weights. We constrain f_σ to produce diagonal covariance matrices. As shorthand, let the neural network $f(z) := (f_\mu(z), f_\sigma(z))$.

Inference

Given an image's pixel intensities $x = (x_1, \dots, x_m)$, we aim to infer the posterior distribution of $z = (z_1, \dots, z_n)$. Unfortunately, integrating z out of the joint distribution (x, z) to compute the marginal likelihood of x is intractable due to the nonlinear form of f. Therefore, we turn to variational inference. Let the variational approximate posterior take the form

$$q(z|x) = \mathcal{N}(g_\mu(x), g_\sigma(x)),$$

where g_μ and g_σ are neural networks that map x to a mean vector and a diagonal covariance matrix, respectively. As shorthand, let neural network $g(x) := (g_\mu(x), g_\sigma(x))$. By the standard construction of the variational lower bound,

$$\begin{aligned}\log p(x) &\geq \log p(x) - D_{\mathrm{KL}}[q(z|x), p(z|x)] \\ &= \mathbb{E}_q[\log p(x|z)] - D_{\mathrm{KL}}[q(z|x), p(z)].\end{aligned}$$

Therefore, the distribution q that maximizes this lower bound minimizes $D_{\mathrm{KL}}[q(z|x), p(z|x)]$: this q is the best approximation of the specified form to the posterior. Let W_f and W_g be the weights of neural networks f and g, respectively. Maximizing over $W = (W_f, W_g)$ simultaneously finds the q that best approximates the posterior and the model p that assigns the highest probability to our data.

The normal-normal KL-divergence $D_{\mathrm{KL}}[q(z|x), p(z)]$ is closed form, but $\mathbb{E}_q[\log p(x|z)]$ is not. We can nonetheless efficiently compute unbiased estimates of its gradient, and therefore maximize the lower bound by stochastic gradient optimization.

We use the stochastic gradient based on the "reparameterization trick". Let $\epsilon \sim \mathcal{N}(0, I)$. Then

$$\begin{aligned}&\frac{\partial}{\partial W}\mathbb{E}_q[\log p(x|z)] \\ &= \mathbb{E}_\epsilon\left[\frac{\partial}{\partial W}\log p(x|z = g_\sigma(x)\epsilon + g_\mu(x))\right].\end{aligned}$$

Hence, for e sampled from ϵ,

$$\frac{\partial}{\partial W}\log p(x|z = g_\sigma(x)e + g_\mu(x))$$

is an unbiased estimate of the derivative of $\mathbb{E}_q[\log p(x|z)]$.

Dataset

- 43,444 galaxy images for training.
- Each image is cropped around one prominent galaxy.
- Each image is downscaled to 69 × 69 pixels.

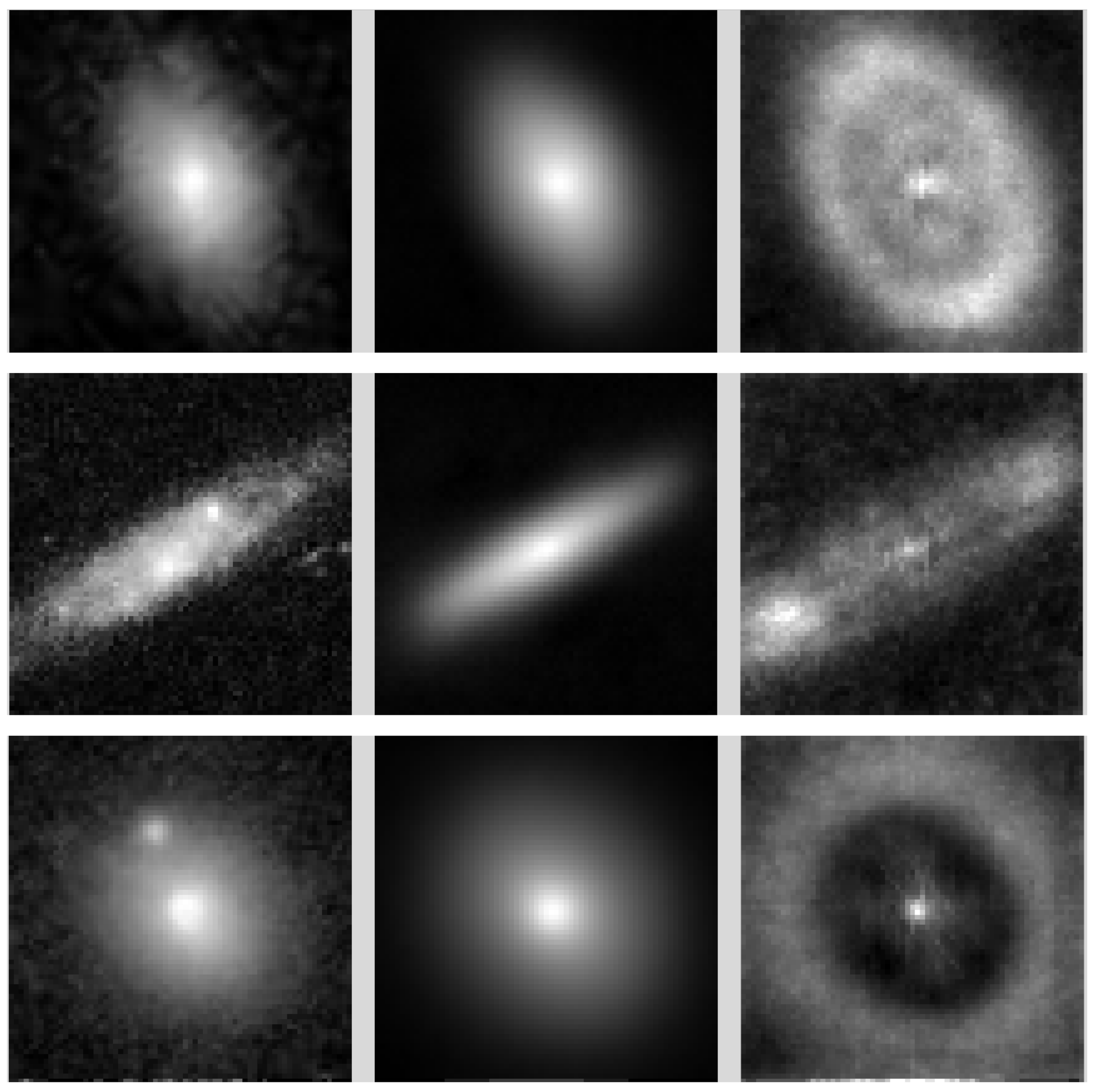
Implementation

- Mocha.jl—a neural network toolkit written in Julia, inspired by Caffe.
- New types of layers to compute the proposed loss function.
- f and g each have two hidden layers composed of 128 hidden nodes each, with rectified linear units.
- The parts corresponding to the output layers of f and g each use exponential nonlinearities to ensure that variances are strictly positive.
- z is 8 dimensional.
- On an Nvidia Tesla K20X GPU, the network performs roughly 200 iterations per second.

Quantitative results

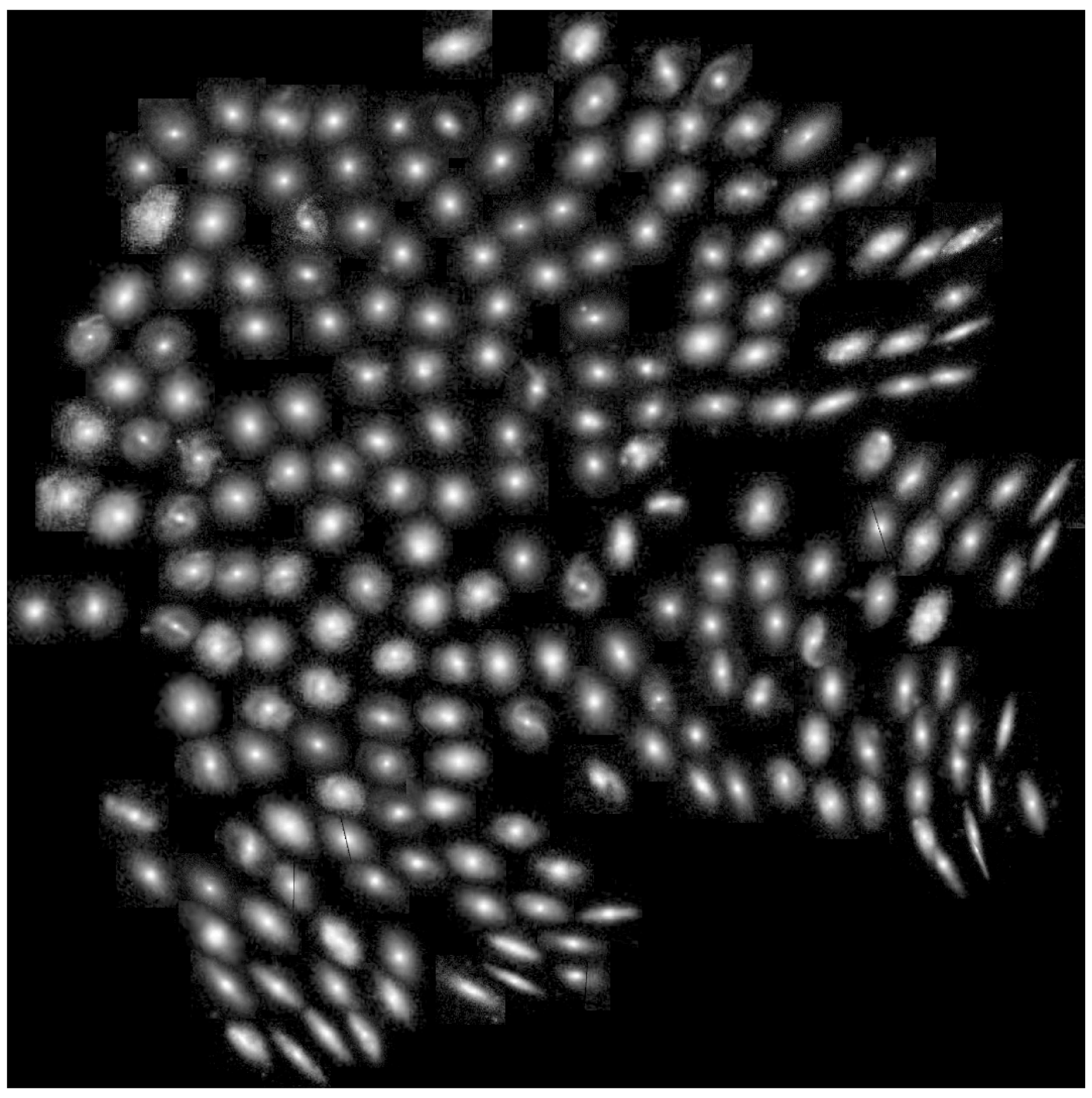
- Held-out dataset of 1000 images of galaxies.
- Current common practice: fit a scaled bivariate Gaussian density function to each imaged galaxy. We denote the fitted scaled density function, evaluated at each pixel, $\hat{x}$.
- On 97.1% of held-out images, $f_\mu(g_\mu(x))$ has lower mean squared error over pixels than $\hat{x}$.
- On 97.2% of held-out images, the pixels x have higher probability under $\mathcal{N}(f_\mu(g_\mu(x)), f_\sigma(g_\mu(x)))$ than under $\mathcal{N}(\hat{x}, \sigma I)$, for every $\sigma > 0$.

Sample conditional distributions



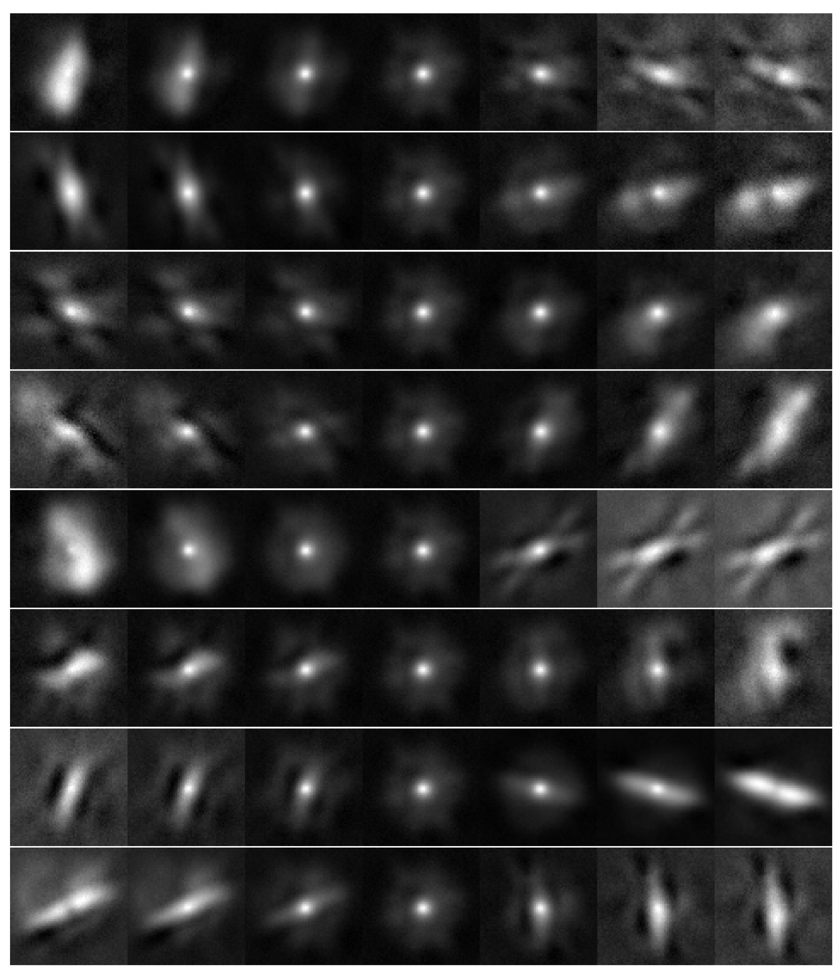
Each row corresponds to a different example from a test set. The left column shows the input x. The center column shows the output $f_\mu(z)$ for a z sampled from $\mathcal{N}(g_\mu(x), g_\sigma(x))$. The right column shows the output $f_\sigma(z)$ for the same z.

Stochastic neighbor embedding



Galaxies embedded in two dimensions based on the means of their variational distributions, $f_\mu(x)$.

Latent space



$f_\mu(z)$ for z values sampled according to a one-at-a-time experimental design. In each row, from left to right, one dimension of z is incremented by one standard deviation per column, while the other dimensions are fixed at zero. The center column in each row is $f_\mu(0, \dots, 0)$.

Acknowledgments

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

contact: jeff@stat.berkeley.edu