Pulse Detection Algorithms for Use in SETI@home

Eric J. Korpela, Eric M. Heien, Dan Werthimer
Space Sciences Laboratory
University of California, Berkeley
http://setiathome.ssl.berkeley.edu/
korpela@ssl.berkeley.edu

Abstract

We describe the pulse detection algorithms used by the most recent versions of SETI@home.

The first is a modified folding algorithm with an optimal threshold function. It is designed to efficiently search for repeating pulses over a 15+ octave period range with constant false alarm rate per period searched. This algorithm is potentially useful for other applications where detection of faint pulsed emission is desired (i.e. pulsar searches).

The second is a triplet detector, which searches for sets of three evenly spaced pulses that exceed a power threshold. It is useful for detecting pulsed emission that has total duration less than the searched exposure.

We describe the first results of the use of these algorithms in SETI@home and discuss the pulsed radio frequency interference (RFI) environment of the Arecibo observatory.

The Folding Algorithm

The folding algorithm divides the data into chunks of duration equal to the period being searched and co-adds them in order to improve signal to noise ratio. The folding algorithm used in SETI@home is a departure from the standard fast folding algorithm (D.H. Staelin, 1969 Proc. IEEE, 57, 724). On modern computer systems a CPU cache miss can be more expensive than a floating-point addition. The SETI@home folding algorithm has been designed to reduce cache misses at the cost of some additional floating-point math.

The SETI@home folding algorithm described schematically below. The implementation details vary slightly from this description. The SETI@home client passes the folding algorithm an array of measured power (at a constant frequency) of length (N) equivalent to the integration time on a spot on the sky. The folding algorithm divides this chunk into three equal parts and co-adds the data (period=N/3). The algorithm searches the coadded data for any signals above a dynamically computed threshold. The coadded data is further divided into two, and again coadded (period=N/6) and searched for above threshold events. This is process is repeated until a period of two samples is reached.

The algorithm then returns to the original data array, and again divides the data into three, this time with the endpoints of the divided arrays shifted to achieve a period=(N-1)/3, and the folding process is repeated. The periods searched by the SETI@home folding algorithm are:

$$\frac{N}{3\times 2^n} \text{ to } \frac{N}{4\times 2^n} \text{ in period steps of } \frac{1}{3\times 2^n} \text{ with } n = 0\text{–}\log_2(\tfrac{N}{3}) - 1$$
$$\frac{N}{4\times 2^n} \text{ to } \frac{N}{5\times 2^n} \text{ in period steps of } \frac{1}{4\times 2^n} \text{ with } n = 0\text{–}\log_2(\tfrac{N}{4}) - 1$$
$$\frac{N}{5\times 2^n} \text{ to } \frac{N}{6\times 2^n} \text{ in period steps of } \frac{1}{5\times 2^n} \text{ with } n = 0\text{–}\log_2(\tfrac{N}{5}) - 1$$

In principle further periods missed in this search could be examined. The primary benefit of this increased sensitivity to pulses at these periods with duty cycles that are small compared to the duration of a single sample. However, the SETI@home client contains a filter which removes any signal with a bandwidth greater than 2 kHz or equivalently duration of less than 0.5 ms.

The Folding Algorithm Threshold Function

One design goal of the SETI@home pulse finding algorithm was to provide a nearly constant false alarm rate per period searched. This allows the algorithm to be as sensitive as possible over the full period range without providing a large number of spurious detections.

The distribution of random noise in the data provided to the folding algorithm by the SETI@home client program is exponential. As the data is coadded, the distribution becomes more nearly Gaussian. However, because of the enormous quantity of data and the large number of periods being searched, the thresholds must be set at high levels (i.e. large multiples of the standard deviation). Unfortunately the high order moments of the distribution never become close enough to Gaussian to use Gaussian statistics to approximate the probability of exceeding a threshold. To calculate the probability of exceeding a threshold we must use more fundamental statistics.

Given data n samples of data from an exponential distribution with unit variance, the probability of the sum exceeding a threshold T is

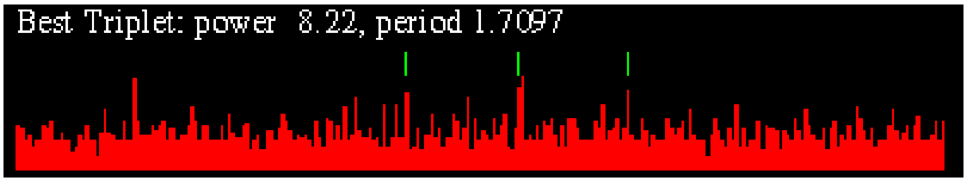
$$P(>T) = Q(n, T)$$

where Q is the complementary incomplete gamma function

$$Q(n, T) = \frac{1}{\Gamma(n)} \int_T^\infty e^{-t} t^{n-1} dt$$

In a folded array of length $m = ceil(period)$, the probability of this threshold being exceeded is $P_m = mQ(n, T)$. Since we desire a constant probability of a hit due to noise in the folded array, the SETI@home client chooses a constant P_m and solves for the data threshold $T = Q^{-1}(n, \frac{P_m}{m})$ based upon current values of n and m.

Inverting the incomplete gamma function is a costly calculation. To avoid repeating this calculation unnecessarily, the SETI@home client caches previous calculations in a hashed lookup table.



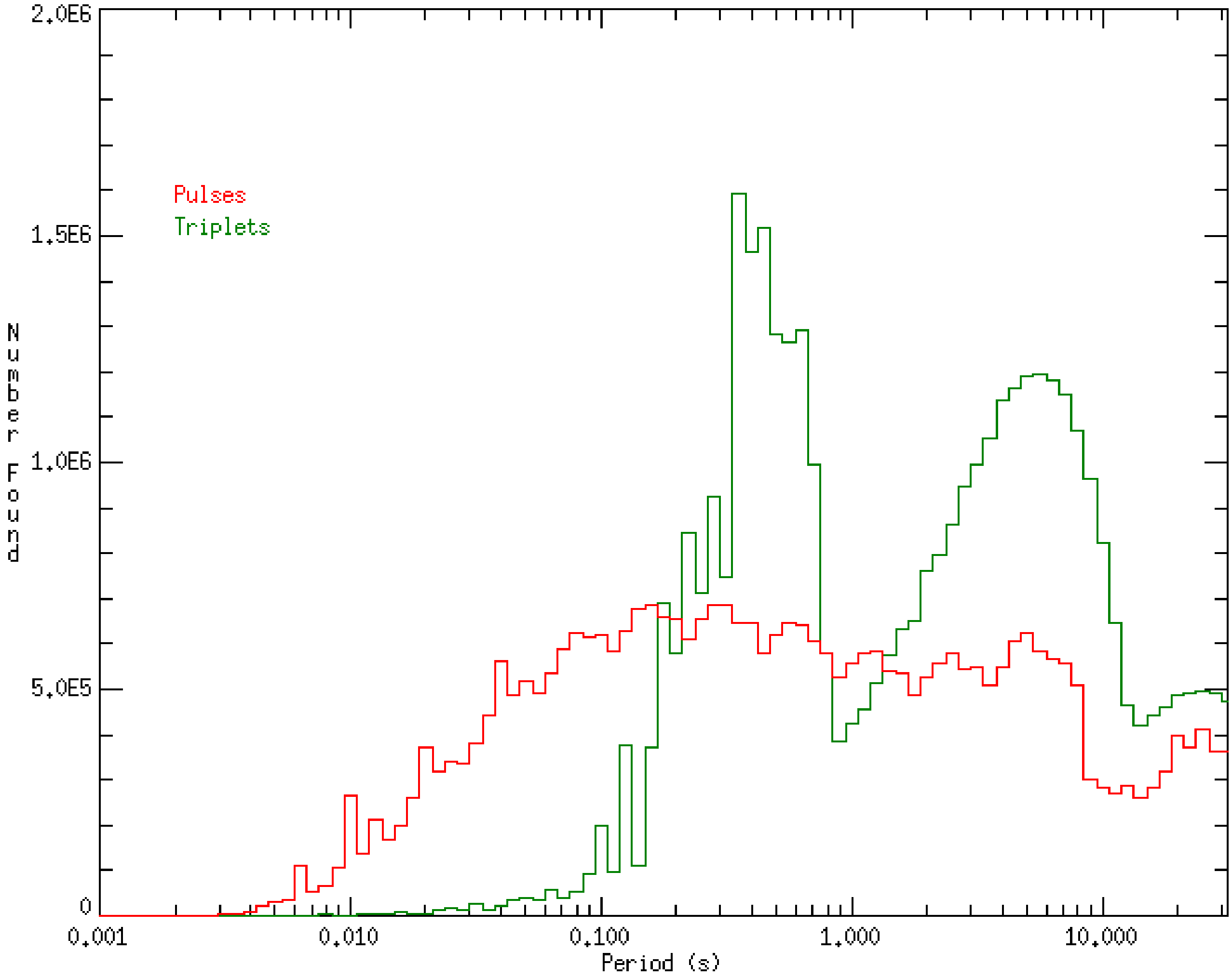
The Triplet Algorithm

When compared with the folding algorithm, the triplet algorithm is simple in design and implementation. A triplet is defined to be three evenly spaced signals above a constant threshold with periods between signals when the power drops below threshold. The illustration above shows a typical example.

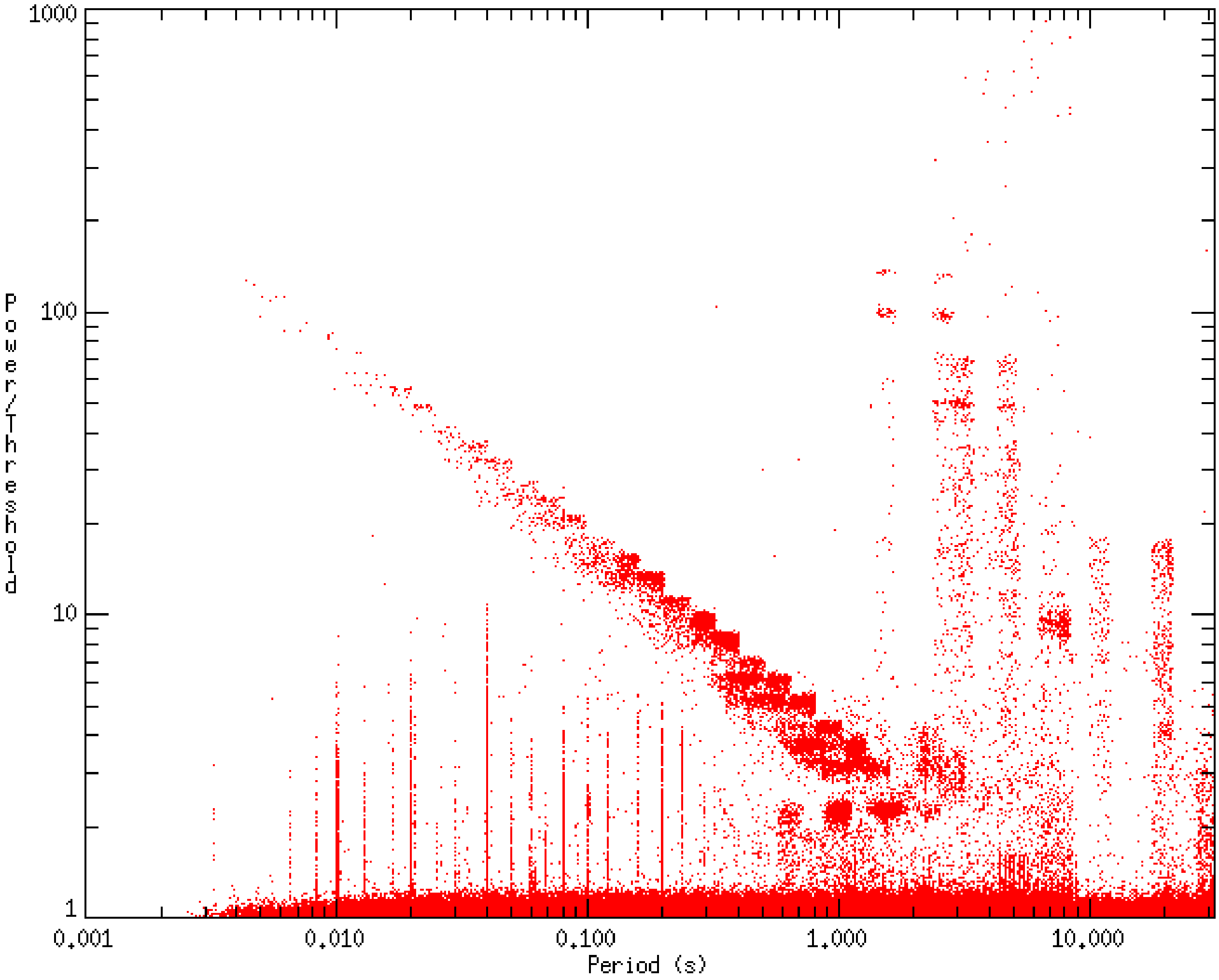
The benefits of the triplet algorithm are its minimal CPU use, its sensitivity to pulsed signals that are broadcast for a duration that is short compared to the integration time. The disadvantages are its limitation to half integer periods, its relative insensitivity to short period signals, and its tendency to report triplets in continuous RFI signals.

First Results

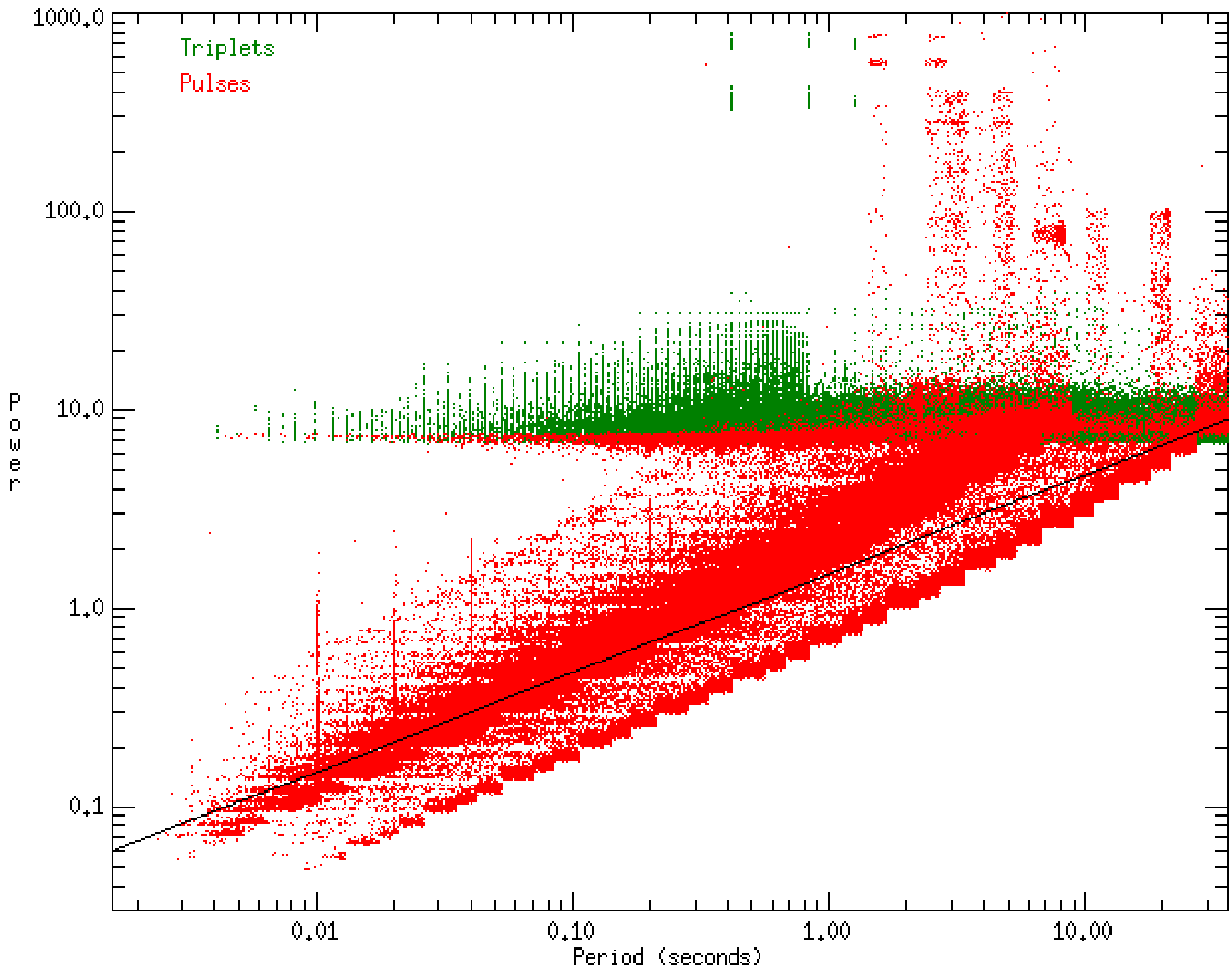
Recent versions of SETI@home (version 3.0 and above) have included the pulse detection algorithms described here. The thresholds of each algorithm have been chosen such that the probability of a hit due to random noise in a work unit (10 kHz bandwidth, 107 seconds duration) is about 50%. In SETI@home nomenclature we refer to a hit by the folding algorithm as a "pulse" and a hit by the triplet algorithm as a "triplet." As of Jan 6, 2000, SETI@home users have returned 3.7×10^7 pulses and 4.2×10^7 triplets.



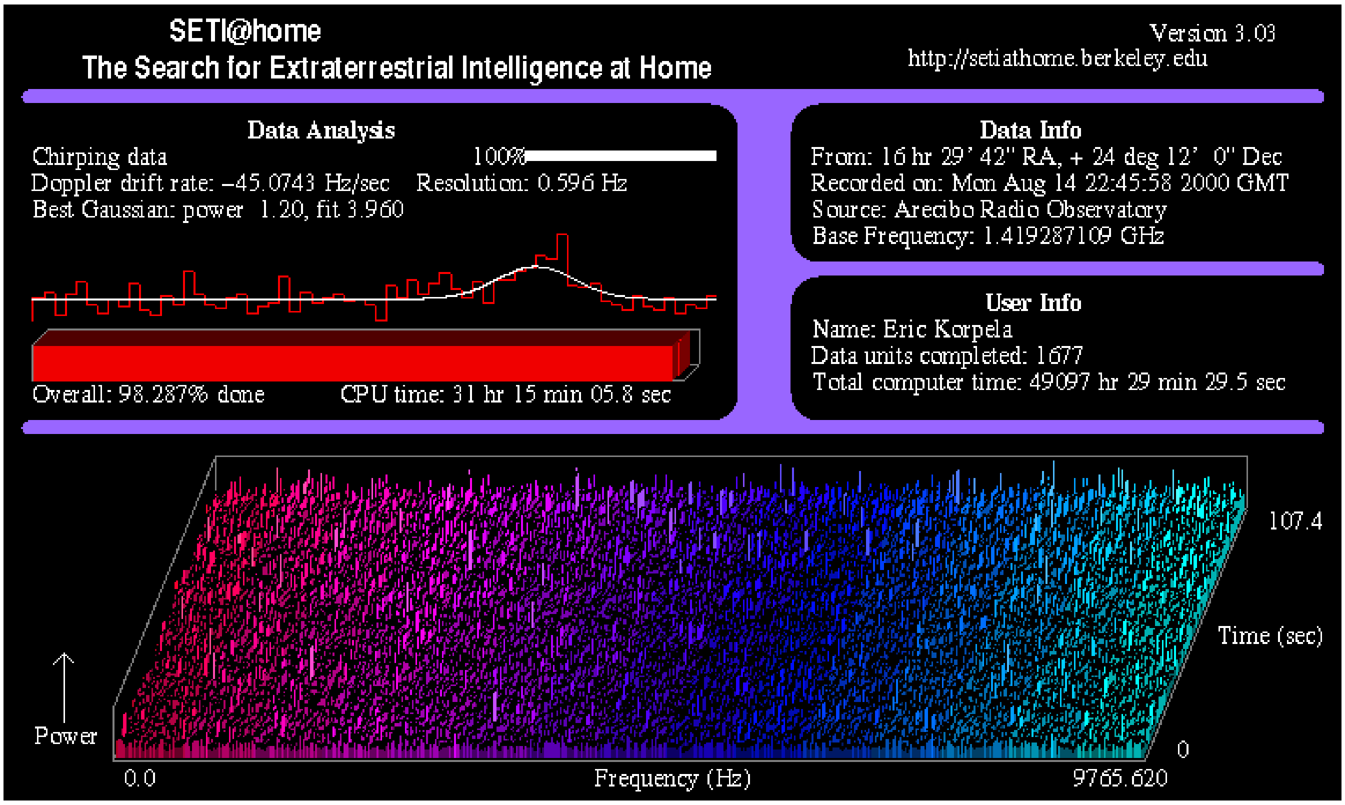
This figure shows the distribution of periods of the signals returned histogrammed with logarithmic (0.05 decade) bins. In pure random noise, each algorithm returns a constant hit rate per period searched. The distribution of periods searched is logarithmic with period for the folding algorithm and linear with period for the triplet algorithm. The distribution of triplet periods (green) is consistent with noise at periods below 70 ms and above 800 ms. The large number of signals between 70 and 800 ms is due to RFI. The distribution of periods for the folding algorithm (red) is nearly consistent with random noise, with some excess in the region between 70 and 800 ms. The fall-off at short periods is due to the distribution of sampling rate of the data passed to the algorithm. The scalloping of the distribution is an artifact of the algorithm. The fall-off above 7 seconds is due to the distribution of beam integration time in work units.



This figure shows the power distribution of a sample of 371 thousand pulses returned by the folding algorithm, shown as a ratio of the power to the detection threshold. Most of the pulses returned reside in the noise distribution near the lower edge. Vertical structures represent common periods found in RFI. Note the peaks near harmonics of 3.33 ms, 10 ms, 100 ms, 333 ms, 1 s, and other "human oriented" periods. The diagonal structure is an unidentified source of RFI at approximately constant power, but at a period that varies by orders of magnitude on timescales of several hours.



This figure shows a comparison of the period and power distribution of triplet (green) and pulses found by the folding algorithm (red). Here power is expressed as a ratio to the mean power in the input array. The faintest signals shown represent received pulse energies of about $1.8x10^{-26}$ J/m^2. Note the large number of RFI detections by the triplet algorithm with periods between 100ms and 1 s. The constant power-variable period RFI detected by the folding algorithm is also visible. The black line illustrates the danger of choosing a threshold to be a constant number of standard deviations in the folded array. If a threshold of this type were chosen to match the false alarm rate at long periods, it would exclude most of the short period signals. If the threshold were chosen to match the sensitivity at short period, the results would be overwhelmed by long period detections.



What is SETI@home?

SETI@home is currently the worlds largest distributed computing project, involving nearly 2.7 million volunteers in over 200 countries around the world. Radio telescope data from the Arecibo radio observatory covering a 2.5 MHz wide band centered on 1420 MHz is divided into work units of 10 kHz bandwidth and 107 second duration. These work units are distributed over the Internet to the volunteers who run the SETI@home client program. On Windows and Macintosh platforms this client presents itself as a screensaver (see above) which analyzes the data when the computer is not otherwise in use. A text based version is available for most UNIX based platforms. This client can be run in the background at low priority.

The client searches for narrow band signals over 14 octaves of bandwidth between 0.075 and 1221 Hz. It performs this search at 24291 Doppler drift rates between -50 Hz/s and +50 Hz/s. It performs Gaussian fitting to detect faint signals that match the profile expected as the telescope beam moves past an object on the sky. And it searches for pulsed signals of the type described here. The entire process typically requires 4 trillion floating point operations to fully examine a single work unit.

SETI@home volunteers have collectively donated 523,000 CPU years for a total of 6.4×10^{20} floating point operations. The results returned include 1.56 billion potential signals which are being processed through data quality checkers and RFI removal algorithms.

For more information about SETI@home see Korpela et al. (2001, *Computing in Science & Engineering*, **v3n1**, Scientific Programming) or visit *http://setiathome.ssl.berkeley.edu/*.