

Using FFTW in this package

Karim J. Rahim

August 19, 2013

1 Overview

The multitaper package can be made to take advantage of the FFTW algorithm [1]. This is done by overwriting the R calls to `fft` and `mvfft` with calls to `fftw` and `mvfftw` provided in the package `fftwtools`. In my experience FFTW is only faster than R's default, Singleton's mixed radix FFT [2], when using large data sets.

2 Suggested Use with FFTWTOOLS

The following will overwrite the functions `fft` and `mvfft` to use FFTW, I suggest experimenting with setting the size at which `fftw` and `mvfftw` is called. I found the settings effective for on my Linux laptop by simulation. I increased the length of a random input dataset size in powers of two and compared the execution speed of `fft` to `fftw`.

```
> require("fftwtools")
> useFFTWGE <- 2^17 ## about 130 thousand values with padding.
> useMVFFTWGE <- 2^16 ## Based on two tests; one of 5 columns and one of 10 columns of data
> fft <- function(z, inverse = FALSE) {
+   if(length(z) >= useFFTWGE) {
+     fftwtools::fftw(z, inverse=inverse)
+   } else {
+     stats::fft(z, inverse=inverse)
+   }
+ }
> ## do the same for now
> mvfft <- function(z, inverse=FALSE) {
+   if(dim(z)[1] >= useMVFFTWGE) {
+     fftwtools::mvfftw(z, inverse=inverse)
+   } else {
+     stats::mvfft(z, inverse=inverse)
+   }
+ }
>
```

The following code will restore R's default functionality.

```
> rm(fft, mvfft)
```

3 Additional notes

If you are interested in additional improvement, there is a package called **FFTW** which allows one to specify plans, see [1], and the package **fftwtools** allows one perform an **FFT** on real data and not have the complex conjugate returned—thus saving memory allocation time.

References

- [1] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
- [2] R.C. Singleton. Mixed Radix Fast Fourier Transforms. In IEEE Acoustics, Speech, and Signal Processing Society. Digital Signal Processing Committee, editors, *Programs for digital signal processing*. IEEE Press, 1979.