Stack-Slide LAL Code

- P. Brady
- T. Creighton

	What we promised	What we've got
✓	Preconditioning: LowPassFilter()	Generalized package tdfilters

✓	Applying corrections:
	resampling time series
	sliding power spectra
	summing power spectra
		CreateResampleRules()
		ApplyResampleRules()
		ApplySlideRules()
		SumSpectra()

✗	Metric formalism: efficiently sampling the parameter space	Not started

Package tdfilters

- Compute zeros, poles, gain of transfer function for a given frequency response:
  - currently implemented for low-, high-pass filters, but easily generalized.

- Convert any zeros-poles-gain transfer function to a time-domain filter.

- Apply time-domain filters pointwise to data in realtime, or to data vectors, in an efficient manner.

Package stackslide

- CreateResampleRules()
  - Generates a list of decimation corrections from a TEMPO-style barycentring model.

- ApplyResampleRules()
  - Decimates and resamples a time series according to the above rules.

- ApplySlideRules()
  - Slides a set of power spectra according to a frequency model.

- SumSpectra()
  - Adds a corrected set of power spectra and searches for peaks.

What's next?

- Basically ~1 month behind on deliverables, due mostly to startup in coding to LAL standard.

- Hopefully can make up for this, and be back on track by beginning of summer.

- Next: parameter-space metric and parameter selection.

Mid-June: Assemble components and begin testing on simulated data.

Mid-September: Continue testing on prototype/engineering data.

Mid-November: Broad-area searches.

31 February, 2001: Testing complete.