NERDS: NETWORK RECONSTRUCTION VIA DYNAMIC SYSTEMS

JAMES HENDERSON

1. Quickstart

This section provides a brief overview of how to use the functions in `NeRDS` to infer a network from time-series data by fitting a nonparametric additive model to the estimated time derivatives. We begin by describing the data format and then show the function calls necessary to carry out the three stages of the algorithm: (1) normalization and smoothing, (2) estimating an additive model, and (3) estimating the coupling.

1.1. Data Format.

`NeRDS` is designed for time-series data that can be thought of as noisy observations of the system trajectories in a dynamic system. We will use the Mouse Embryonic Stem Cell (MESC) data provided with the package for illustrative purposes.

The time-series data should be organized as an $n \times d$ matrix corresponding to n observations of d components. Later function calls also require a vector of length n containing corresponding observation times.

```
> require(NeRDS,'Rlib')
> data('MESC',package='NeRDS')
> Y <- MESC$Y[[1]]
> tm <- MESC$tm
> n <- nrow(Y)
> d <- ncol(Y)
```

When the data consist of R multiple experiments on the same system, they should be organized as a list with each entry an $n \times d$ matrix corresponding to an experiment.

```
> Y <- MESC$Y
> R <- length(Y)
> n <- dim(Y[[1]])[1]
> d <- dim(Y[[1]])[2]
```

1.2. Normalize and Smooth.

The data should be normalized to ensure all components are on the same scale, then smoothed in order to provide functional estimates of the trajectories and time derivatives.

```
> Y <- normData(Y)
> sm <- smooth(Y,tm)
```

Use `smooth.intersect` to stack the estimated derivatives and trajectories into a single matrix for each.

```
> smth <- smooth.intersect(sm)
```

Date: August 17, 2014.

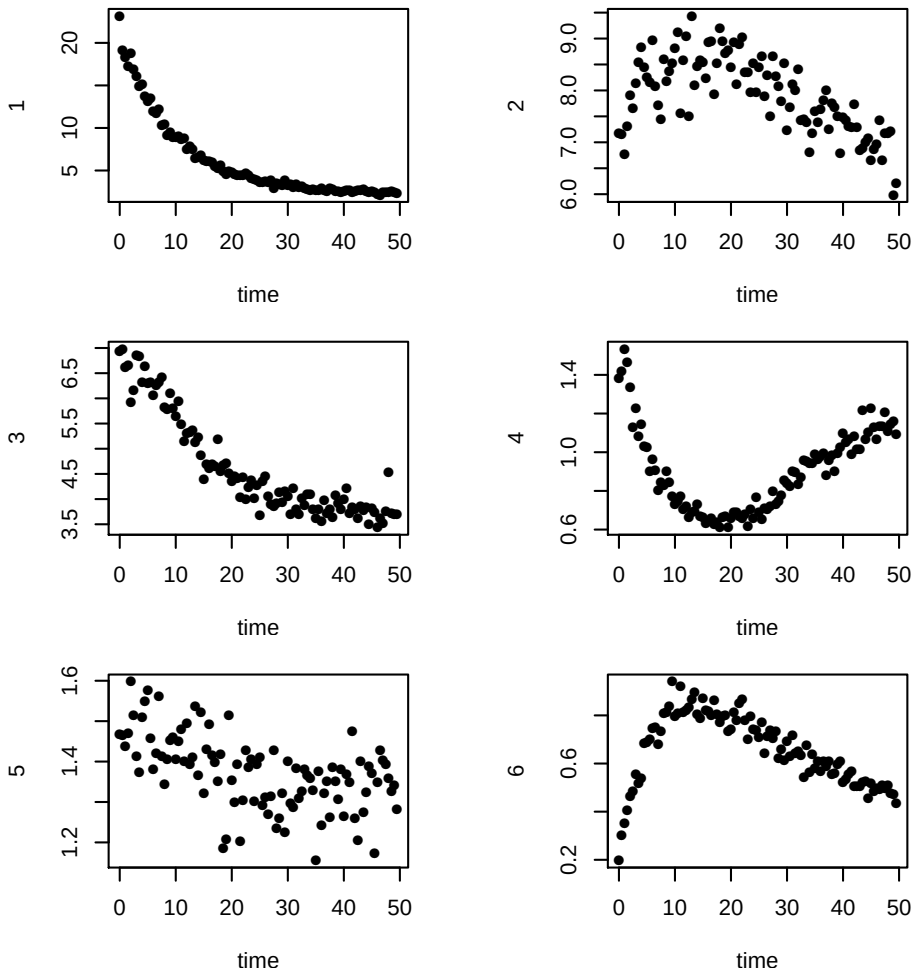

FIGURE 1. Raw time-series from the first experiment in the MESC data.

1.3. **Fit an Additive Model to Each Derivative.** For fixed values of the smoothing parameter `labmda1` and sparsity parameter `lambda2` an additive model can be fit to the derivative of component i using a call to `backfit.i`.

```
> fits <- list()
> lambda1 <- 5e-3
> lambda2 <- 0
> for(i in 1:d){
+     xdot <- smth$Xdot.hat[,i]
+     fits[[i]] <- backfit.i(xdot,smth$X.hat,i,lambda1,lambda2,-.1,1.1)
+ }
```

Model did not converge, i = 6

The function `grid.search.i` may be useful for selecting the two tuning parameters above.

1.4. **Estimate the Coupling.** Having fitted additive models to each of the time derivatives, it remains to compute the coupling metrics to guage the strength of each potential edge.

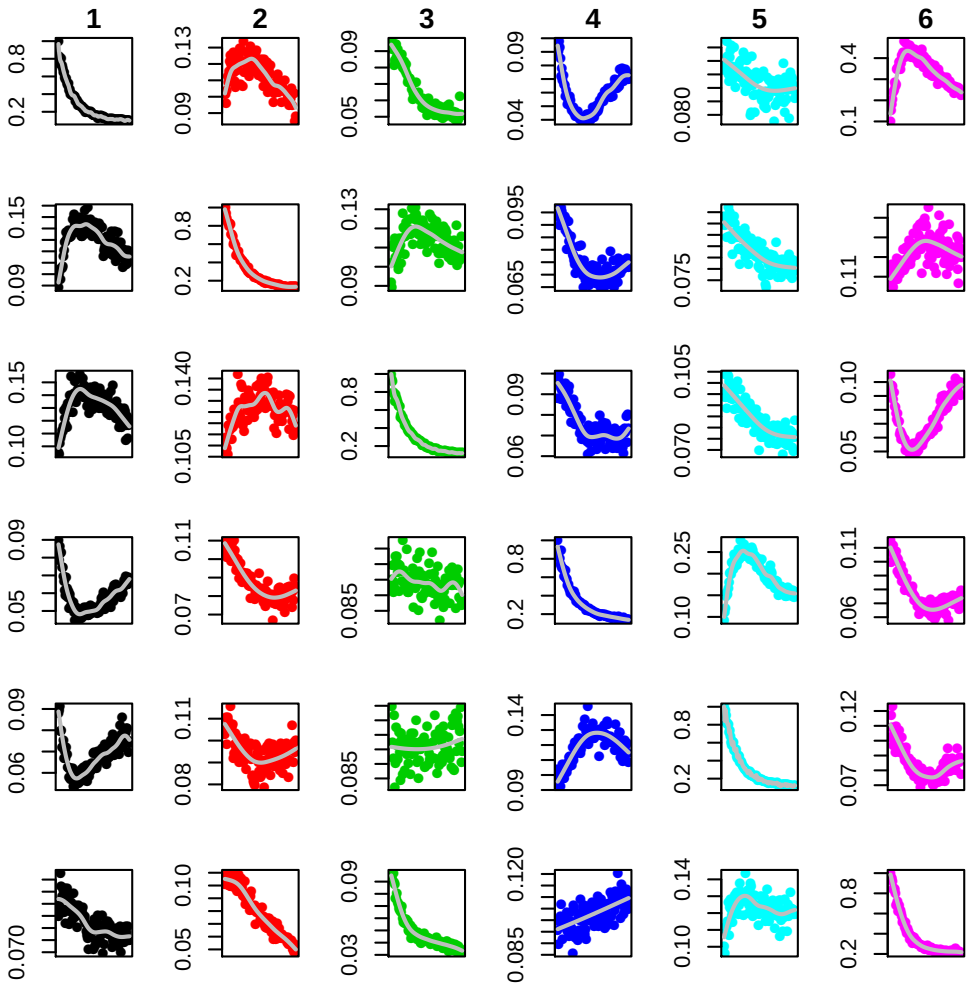

FIGURE 2. Normalized observations and smoothing estimates of the trajectories from the six experiments in the MESC data.

```
> rho <- coupling(fits,smth$X.hat)
> round(rho,2)
```

Potential edges can then be ranked by their coupling scores.