PathMovers

```
class MyPathMover(paths.PathMover):
    def __init__(self, ensemble, my_option):
        # addition ensembles possible
        super(MyPathMover, self).__init__()
        self.ensemble = ensemble
        self.my_option = my_option

    def _called_ensembles(self):
        # ensemble for input samples to __call__
        return [self.ensemble]

    def _get_in_ensembles(self):
        # input ensembles
        return [self.ensemble]

    # _get_out_ensembles defaults to _get_in_ensembles

    def __call__(self, sample_for_ensemble):
        # additional samples possible
        # do the stuff to make trial samples
        details = {'my_detail': 'value'}
        return [trial_sample], details
```

Move Scheme and Strategy

A mover for every ensemble:
lots of movers!

Plus, we want the flexibility to change aspects of the move scheme: options for movers, which ensembles are involved, etc