

Supporting Information for **Multiple-Replica Strategies for Free-Energy Calculations in NAMD: Multiple-Walker Adaptive Biasing Force and Walker Selection Rules**

Jeffrey Comer^{1,2}, James C. Phillips³, Klaus Schulten^{4,3}, and Christophe Chipot^{*1,3,4}

¹Laboratoire International Associé Centre National de la Recherche Scientifique et University of Illinois at Urbana-Champaign, Unité Mixte de Recherche n°7565, Université de Lorraine, B.P. 70239, 54506 Vandœuvre-lès-Nancy cedex, France

²Institute of Computational Comparative Medicine and Nanotechnology Innovation Center of Kansas State, Department of Anatomy and Physiology, Kansas State University, Manhattan, Kansas 66506

³Theoretical and Computational Biophysics Group, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, 405 North Mathews Avenue, Urbana, Illinois 61801

⁴Department of Physics, University of Illinois at Urbana-Champaign, 1110 West Green Street, Urbana, Illinois 61801

Tcl implementation of the procedure `resampleWalkers` used by the procedure `selectionRules` listed in the main text.

```
proc resampleWalkers {wList} {  
    set num [llength $wList]  
  
    # Normalize the list of weights.  
    set wSum 0.0  
    foreach w $wList {  
        set wSum [expr {$wSum + $w}]  
    }  
    set weightList {}  
    foreach w $wList {  
        lappend weightList [expr {$w/$wSum}]  
    }  
}
```

*E-mail: chipot@ks.uiuc.edu

```

}

# Get the number of clones for each walker.
set wbar(0) [lindex $weightList 0]
set u [expr {rand()}]
set copyNumList [list [expr { int($num*$wbar(0)+$u) }] ]
for {set r 1} {$r < $num} {incr r} {
    set r0 [expr {$r-1}]
    set wbar($r) [expr {$wbar($r0) + [lindex $weightList $r]}]
    lappend copyNumList [expr {int($num*$wbar($r)+$u)-int($num*$wbar($r0)+$u)}]
}
return $copyNumList
}

```

Tcl implementation of the procedure `minExchanges` used by the procedure `selectionRules` listed in the main text.

```

# Determine the minimal exchanges that must be made to resample.
proc minExchanges {copyNumList} {
    # Make a list of exchanges.
    set cloneZeroList {}
    set cloneMultList {}
    set r 0
    foreach cloneNum $copyNumList {
        if {$cloneNum == 0} { lappend cloneZeroList $r }
        if {$cloneNum > 1} { lappend cloneMultList $r }
        incr r
    }

    # Is nothing cloned?
    if {[llength $cloneZeroList] == 0} {
        return {}
    }

    # Walkers cloned multiple times are copied to
    # walkers cloned zero times.
    # Make the list of exchanges 'srcDestList'.
    set srcDestList {}
}

```

```
set zeroInd 0
foreach mult $cloneMultList {
    # We get one clone just by doing nothing.
    set extraNum [expr {[lindex $copyNumList $mult]-1}]
    for {set j 0} {$j < $extraNum} {incr j} {
        set dest [lindex $cloneZeroList $zeroInd]
        set srcDestList [concat $srcDestList $mult $dest]
        incr zeroInd
    }
}

return $srcDestList
}
```