A micro PLUMED2.0 tutorial for ESPREsSo users

Davide Branduardi

What we will see

* How to get the code and the documentation
* How to start PLUMED 2.0 from ESPResSo
* How to monitor CVs
* How to bias CVs
* How to perform and analyze Metadynamics

What we will see

* How to get the code and the documentation
* How to start PLUMED 2.0 from ESPResSo
* How to monitor CVs
* How to bias CVs
* How to perform and analyze Metadynamics

Disclaimer 1: I am not an ESPResSo user!

What we will see

* How to get the code and the documentation
* How to start PLUMED 2.0 from ESPResSo
* How to monitor CVs
* How to bias CVs
* How to perform and analyze Metadynamics

Disclaimer 1: I am not an ESPResSo user!

Disclaimer 2: all that you will see is in a document which you can download from the web! (do not write, just follow)

What we will see

* How to get the code and the documentation
* How to start PLUMED 2.0 from ESPResSo
* How to monitor CVs
* How to bias CVs
* How to perform and analyze Metadynamics

Disclaimer 1: I am not an ESPResSo user!

Disclaimer 2: all that you will see is in a document which you can download from the web! (do not write, just follow)

I assume: linux shell+ ESPResSo input + gnuplot + vmd

Where you get the code

* Setup the environment (bash)

```
source
/home/davide/software/espresso/SETUP_FULL_ENVIRONMENT.sh
```

* Test it

```
plumed help
```

* The executable we will use is accessible on the workstations (copy it or link it):

```
/home/davide/software/espresso/Espresso
```

* Copy the exercises in your own directory for this tutorial

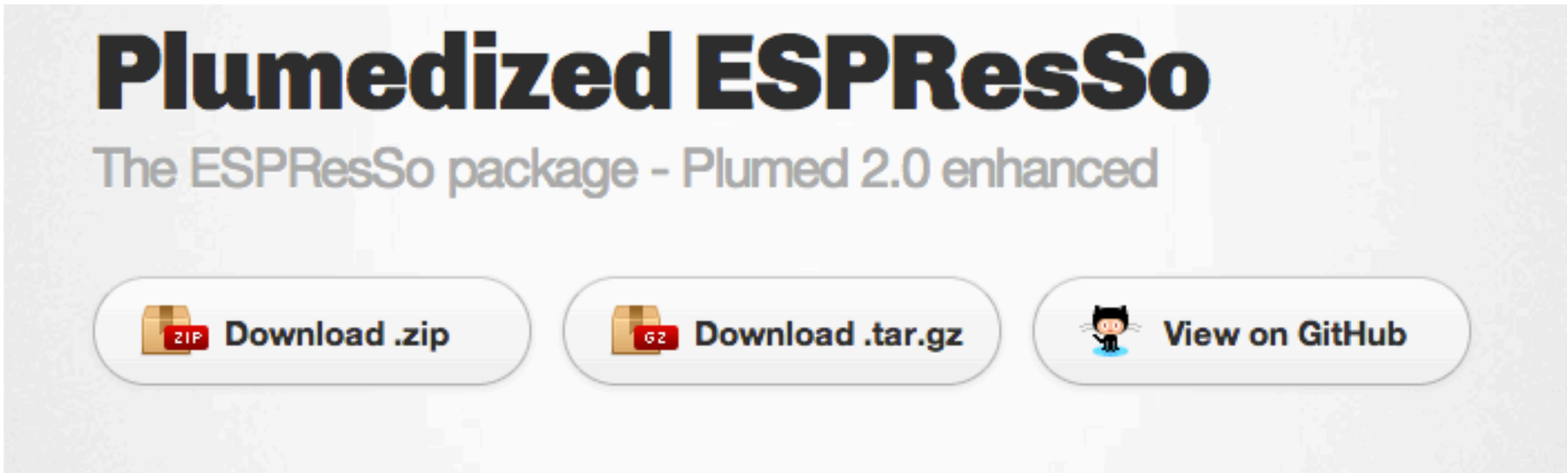
```
cp -r /home/davide/software/espresso/plumed_tests .
```

For the brave: get it and build it (not now)

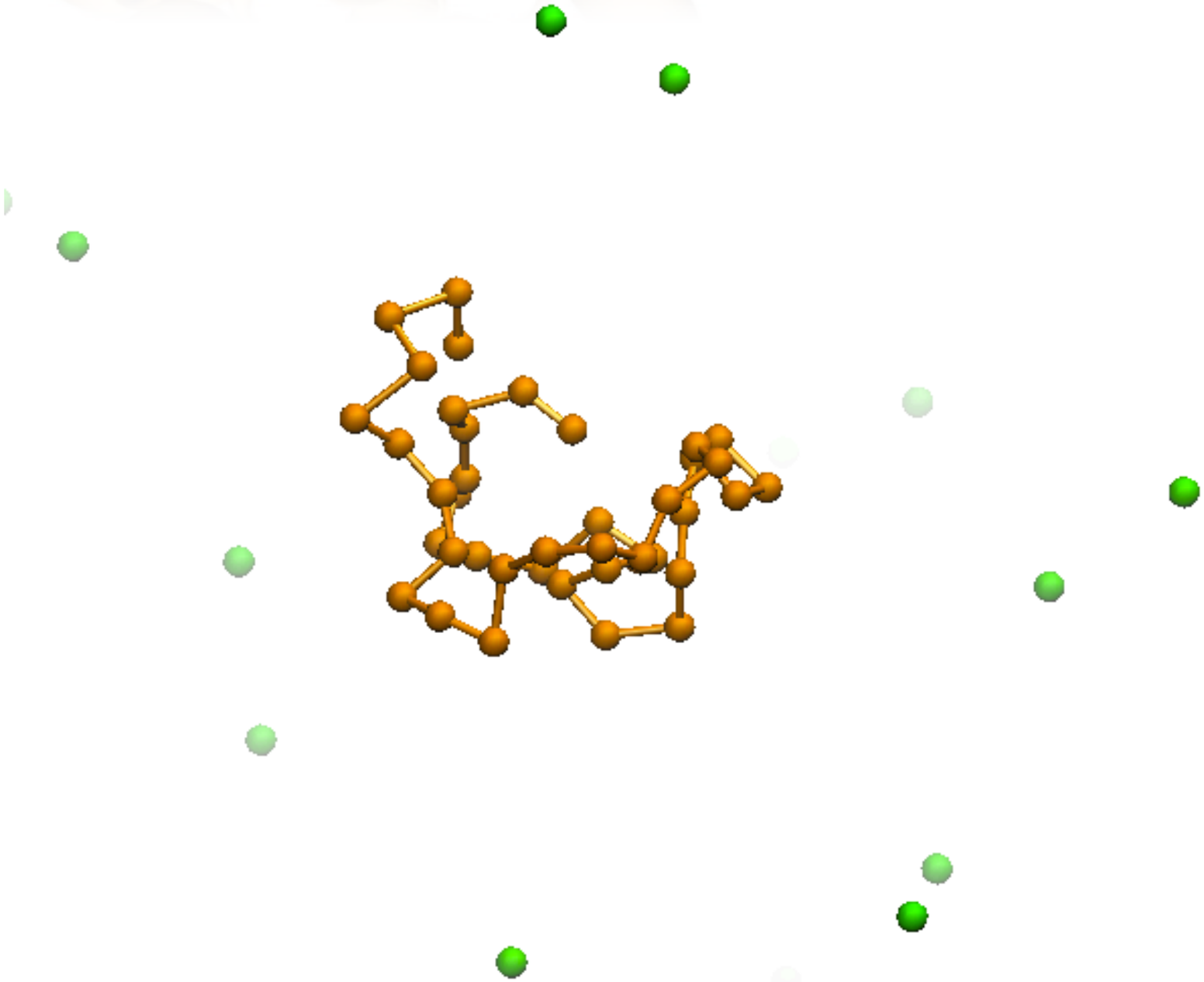
Get plumed: `https://github.com/plumed/plumed2`

Get libmatheval (opt): `http://www.gnu.org/software/libmatheval/`

Get ESPResSo (my own fork) along with this tutorial at:
`http://davidebr.github.io/espresso/`



The system: `tutorial.tcl`



1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

```
setmd plumedison 1    <- activate plumed
setmd plumedfile plumed.dat <- set the input (external file)
set plumed_input [open "plumed.dat" "w"]    <- the input
puts  $plumed_input "d1: DISTANCE ATOMS=1,40"
puts  $plumed_input "PRINT ARG=* STRIDE=50 FILE=COLVAR"
close $plumed_input
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

action
action

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

action
action

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

action
action

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

action
action

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

action
action

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

action
action

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* Open the `tutorial.tcl`

* You find the following

plumed.dat

action
action

```
d1: DISTANCE ATOMS=1,40
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

* Then in the loop plumed check what to do at every step

* Try it

```
/home/davide/software/espresso/Espresso tutorial.tcl
```

1_basic : how to start plumed

* What do you get? A banner on stdout and a COLVAR file

```
PLUMED: PLUMED compiled on Sep  3 2013 at 02:35:56
PLUMED: There is not yet a published paper describing this software.
PLUMED: If you use it in a publication please explicitly state
PLUMED: which version you are using and cite the previous paper [1]
PLUMED: For further information see the PLUMED web page at www.plumed-code.org
PLUMED: Molecular dynamics engine: espresso
PLUMED: Precision of reals: 8
PLUMED: Running over 1 node
PLUMED: Number of atoms: 80
PLUMED: File suffix:
PLUMED: FILE: plumed.dat
PLUMED: Action DISTANCE
PLUMED:   with label d1
PLUMED:   between atoms 1 40
PLUMED:   using periodic boundary conditions
PLUMED: Action PRINT
PLUMED:   with label @1
PLUMED:   with stride 50
PLUMED:   with arguments d1
PLUMED:   on file COLVAR
PLUMED:   with format  %f
PLUMED: END FILE: plumed.dat
PLUMED: Timestep: 0.010000
```

some features are parallel

the DISTANCE action you declared

the PRINT action you declared

1_basic : how to start plumed

* What do you get? A banner on stdout and a COLVAR file

COLVAR file:

```
#! FIELDS time d1
 0.500000 2.374363
 1.000000 2.551184
 1.500000 2.392192
 2.000000 2.292906
 2.500000 2.789281
 3.000000 2.782809
 3.500000 3.443960
 4.000000 4.355173
 4.500000 5.197304
 5.000000 5.761397
 5.500000 6.199342
 6.000000 6.726716
```

Header: tells what the file contains (here time and distance)

2_basic_on_off :reset plumed

* Open the `tutorial.tcl`

```
set i 0
while { $i<$n_cycle } {

set plumed_input [open "plumed.dat" "w"]
puts  $plumed_input "d1: DISTANCE ATOMS=1,40"
puts  $plumed_input "PRINT ARG=* STRIDE=10 FILE=COLVAR_$i"
close $plumed_input

    setmd plumedreset 1
    integrate $n_steps

    if { $vmd == "yes" } { imd positions }

    writevcf $f
    incr i
}
```

2_basic_on_off :reset plumed

* Open the `tutorial.tcl`

```
set i 0
while { $i<$n_cycle } {

set plumed_input [open "plumed.dat" "w"]
puts  $plumed_input "d1: DISTANCE ATOMS=1,40"
puts  $plumed_input "PRINT ARG=* STRIDE=10 FILE=COLVAR_$i"
close $plumed_input

    setmd plumedreset 1
    integrate $n_steps

    if { $vmd == "yes" } { imd positions }

    writevcf $f
    incr i
}
```

A new COLVAR at every round!

2_basic_on_off :reset plumed

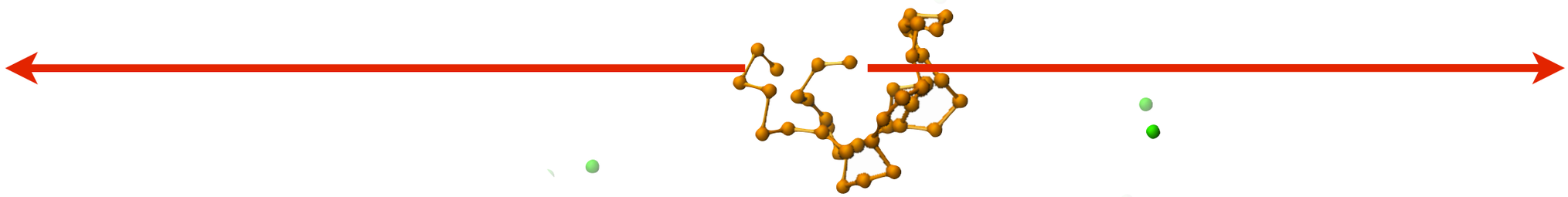
* Open the `tutorial.tcl`

```
set i 0
while { $i<$n_cycle } {

set plumed_input [open "plumed.dat" "w"]
puts  $plumed_input "d1: DISTANCE ATOMS=1,40"
puts  $plumed_input "PRINT ARG=* STRIDE=10 FILE=COLVAR_$i"
close $plumed_input

    setmd plumedreset 1
    integrate $n_steps

    if { $vmd == "yes" } { imd positions }

    writevcf $f
    incr i
}
```

A new COLVAR at every round!

Reread the input and reinitialize!

3_steer :steer the two ends of chain



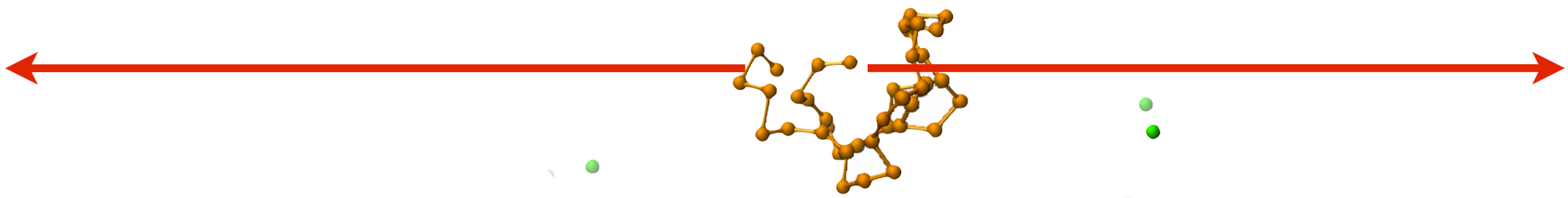
```
d1: DISTANCE ATOMS=1,40
# this hash is a comment
# a moving restraint using d1
MOVINGRESTRAINT ...
  ARG=d1
  STEP0=0       AT0=2.0   KAPPA0=5.0
  STEP1=20000   AT1=40.0  KAPPA1=5.0
  STEP2=40000   AT2=2.0   KAPPA2=5.0
... MOVINGRESTRAINT
PRINT ARG=* STRIDE=50 FILE=COLVAR
PRINT ARG=@1.d1_cntr,@1.d1_work STRIDE=50 FILE=WORK
```

give a recipe on how to steer the endpoints

$$H'(\mathbf{x},t) = H(\mathbf{x}) + \frac{k}{2}\left(\mathbf{s}(\mathbf{x}) - x_0 - vt\right)^2$$

$$v = (\texttt{AT1} - \texttt{AT0})/(\texttt{STEP1-STEP0})$$

3_steer :steer the two ends of chain



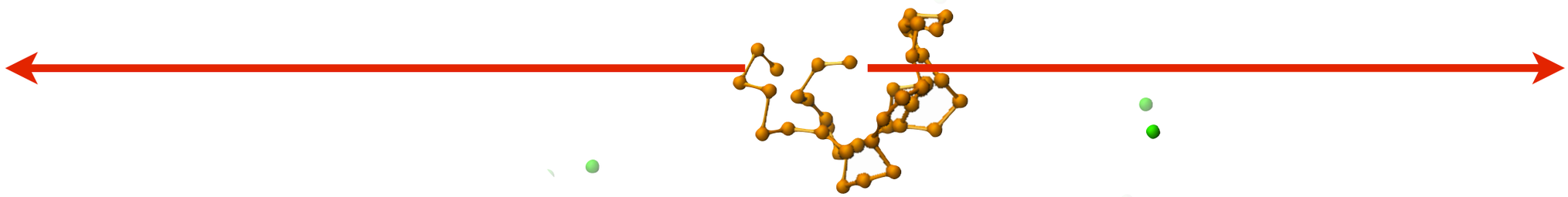
```
d1: DISTANCE ATOMS=1,40
# this hash is a comment
# a moving restraint using d1
MOVINGRESTRAINT ...
  ARG=d1
  STEP0=0      AT0=2.0   KAPPA0=5.0
  STEP1=20000  AT1=40.0  KAPPA1=5.0
  STEP2=40000  AT2=2.0   KAPPA2=5.0
... MOVINGRESTRAINT
PRINT ARG=* STRIDE=50 FILE=COLVAR
PRINT ARG=@1.d1_cntr,@1.d1_work STRIDE=50 FILE=WORK
```

give a recipe on how to steer the endpoints

$$H'(\mathbf{x},t) = H(\mathbf{x}) + \frac{k}{2}\left(\mathbf{s}(\mathbf{x}) - x_0 - vt\right)^2$$

$$v = (\texttt{AT1} - \texttt{AT0})/(\texttt{STEP1-STEP0})$$

3_steer :steer the two ends of chain



```
d1: DISTANCE ATOMS=1,40
# this hash is a comment
# a moving restraint using d1
MOVINGRESTRAINT ...
  ARG=d1
  STEP0=0       AT0=2.0   KAPPA0=5.0
  STEP1=20000   AT1=40.0  KAPPA1=5.0
  STEP2=40000   AT2=2.0   KAPPA2=5.0
... MOVINGRESTRAINT
PRINT ARG=* STRIDE=50 FILE=COLVAR
PRINT ARG=@1.d1_cntr,@1.d1_work STRIDE=50 FILE=WORK
```

give a recipe on how to steer the endpoints

$$H'(\mathbf{x},t) = H(\mathbf{x}) + \frac{k}{2}\left(\mathbf{s}(\mathbf{x}) - x_0 - vt\right)^2$$

$$v = (\mathtt{AT1} - \mathtt{AT0})/(\mathtt{STEP1}-\mathtt{STEP0})$$

selective printout of some components that each action has. How to know which are available? Read the output!

3_steer :steer the two ends of chain

```
PLUMED: Action MOVINGRESTRAINT
PLUMED:   with label @1
PLUMED:   with stride 1
PLUMED:   with arguments d1
PLUMED:   step0 0
PLUMED:   at 2.000000
PLUMED:   with force constant 5.000000
PLUMED:   step1 20000
PLUMED:   at 40.000000
PLUMED:   with force constant 5.000000
PLUMED:   step2 40000
PLUMED:   at 2.000000
PLUMED:   with force constant 5.000000
PLUMED:   added component to this action:  @1.bias
PLUMED:   added component to this action:  @1.force2
PLUMED:   added component to this action:  @1.d1_cntr
PLUMED:   added component to this action:  @1.d1_work
```

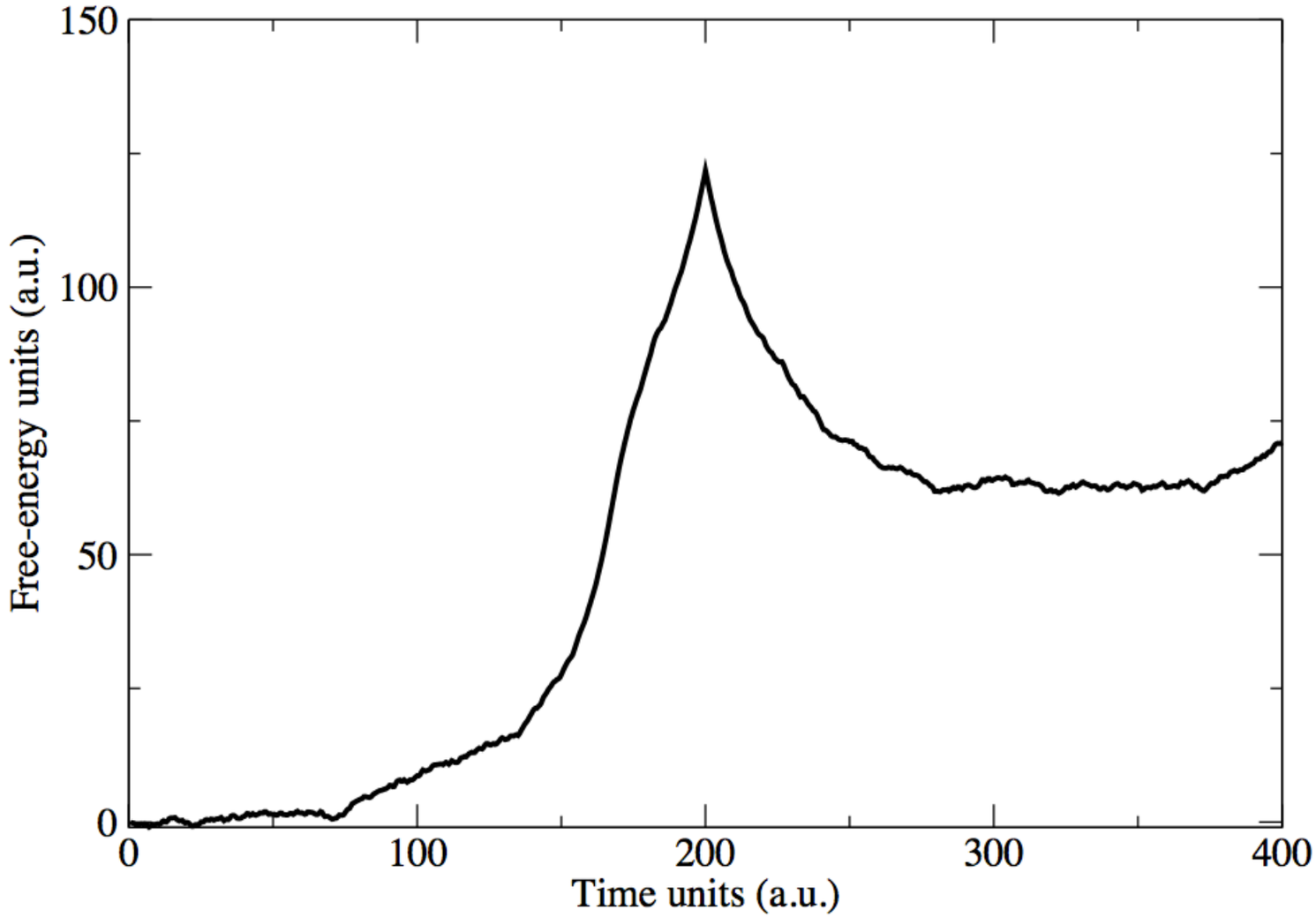
3_steer :steer the two ends of chain

```
PLUMED: Action MOVINGRESTRAINT
PLUMED:   with label @1
PLUMED:   with stride 1
PLUMED:   with arguments d1
PLUMED:   step0 0
PLUMED:   at 2.000000
PLUMED:   with force constant 5.000000
PLUMED:   step1 20000
PLUMED:   at 40.000000
PLUMED:   with force constant 5.000000
PLUMED:   step2 40000
PLUMED:   at 2.000000
PLUMED:   with force constant 5.000000
PLUMED:   added component to this action:  @1.bias
PLUMED:   added component to this action:  @1.force2
PLUMED:   added component to this action:  @1.d1_cntr
PLUMED:   added component to this action:  @1.d1_work
```

3_steer :steer the two ends of chain

$$W = \int_0^{t_s} dt \, \frac{\partial H'(\mathbf{x}, t)}{\partial t}$$

```
gnuplot> plot "WORK" u 1:3
```

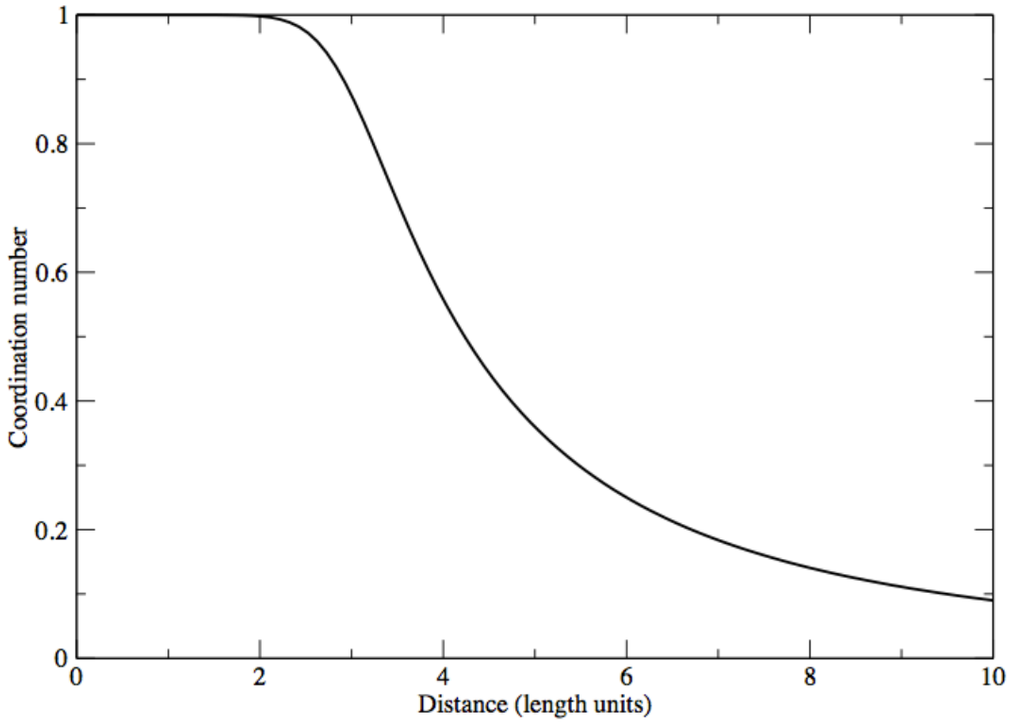


enjoy the movie: `vmd -e ../test.vmd`

4_coordination_number

* You can steer (or impose forces) along any of the provided collective variables. For example, you might want to see how particles collapse around the chain

$$C(\mathbf{x}) = \sum_{i}^{N_A} \sum_{j}^{N_B} \frac{1 - \left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{r_0} \right)^{nn}}{1 - \left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{r_0} \right)^{mm}}$$




```
gnuplot> r0=3;nn=14;mm=16
gnuplot> f(x)=(1-(x/r0)**nn)/(1-(x/r0)**mm)
gnuplot> set xr [0:10]
gnuplot> plot f(x)
```

4_coordination_number

```
d1: DISTANCE ATOMS=1,40
# a coordination number of the other particles
c1: COORDINATION GROUPA=1-40 GROUPB=41-80 R_0=3.0 NN=14 MM=16
# a moving restraint using d1: note that the kappa is kept if not specified
MOVINGRESTRAINT ...
  ARG=d1
  STEP0=0      AT0=2.0   KAPPA0=5.0
  STEP1=20000  AT1=40.0
... MOVINGRESTRAINT
# another moving restraint after the steer
 MOVINGRESTRAINT ...
  ARG=c1 LABEL=sc1
  STEP0=20000  AT0=0.0   KAPPA0=0.0
  STEP1=40000  AT1=1200.0 KAPPA1=0.1
... MOVINGRESTRAINT
PRINT ARG=* STRIDE=50 FILE=COLVAR
PRINT ARG=c1,sc1.* STRIDE=50 FILE=COLVAR_COOR
```

4_coordination_number

```
d1: DISTANCE ATOMS=1,40
# a coordination number of the other particles
c1: COORDINATION GROUPA=1-40 GROUPB=41-80 R_0=3.0 NN=14 MM=16
# a moving restraint using d1: note that the kappa is kept if not specified
MOVINGRESTRAINT ...
  ARG=d1
  STEP0=0      AT0=2.0   KAPPA0=5.0
  STEP1=20000  AT1=40.0
... MOVINGRESTRAINT
# another moving restraint after the steer
 MOVINGRESTRAINT ...
  ARG=c1 LABEL=sc1
  STEP0=20000  AT0=0.0   KAPPA0=0.0
  STEP1=40000  AT1=1200.0 KAPPA1=0.1
... MOVINGRESTRAINT
PRINT ARG=* STRIDE=50 FILE=COLVAR
PRINT ARG=c1.sc1.* STRIDE=50 FILE=COLVAR_COOR
```

define the coordination of the chain (atoms 1-40) with the particles (atoms 41-80)

4_coordination_number

```
d1: DISTANCE ATOMS=1,40
# a coordination number of the other particles
c1: COORDINATION GROUPA=1-40 GROUPB=41-80 R_0=3.0 NN=14 MM=16
# a moving restraint using d1: note that the kappa is kept if not specified
MOVINGRESTRAINT ...
  ARG=d1
  STEP0=0      AT0=2.0   KAPPA0=5.0
  STEP1=20000  AT1=40.0
... MOVINGRESTRAINT
# another moving restraint after the steer
 MOVINGRESTRAINT ...
  ARG=c1 LABEL=sc1
  STEP0=20000  AT0=0.0   KAPPA0=0.0
  STEP1=40000  AT1=1200.0 KAPPA1=0.1
... MOVINGRESTRAINT
PRINT ARG=* STRIDE=50 FILE=COLVAR
PRINT ARG=c1,sc1.* STRIDE=50 FILE=COLVAR_COOR
```

define the coordination of the chain (atoms 1-40) with the particles (atoms 41-80)

You can bias two CV at the same time

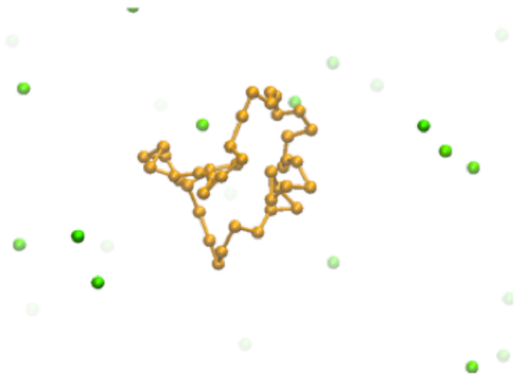
4_coordination_number

```
d1: DISTANCE ATOMS=1,40
# a coordination number of the other particles
c1: COORDINATION GROUPA=1-40 GROUPB=41-80 R_0=3.0 NN=14 MM=16
# a moving restraint using d1: note that the kappa is kept if not specified
MOVINGRESTRAINT ...
  ARG=d1
  STEP0=0      AT0=2.0   KAPPA0=5.0
  STEP1=20000  AT1=40.0
... MOVINGRESTRAINT
# another moving restraint after the steer
 MOVINGRESTRAINT ...
  ARG=c1 LABEL=sc1
  STEP0=20000  AT0=0.0   KAPPA0=0.0
  STEP1=40000  AT1=1200.0 KAPPA1=0.1
... MOVINGRESTRAINT
PRINT ARG=* STRIDE=50 FILE=COLVAR
PRINT ARG=c1.sc1,* STRIDE=50 FILE=COLVAR_COOR
```

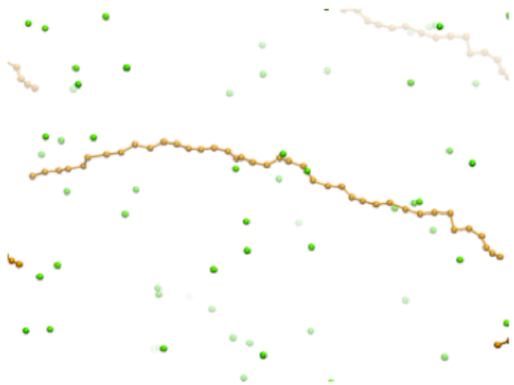
define the coordination of the chain (atoms 1-40) with the particles (atoms 41-80)

You can bias two CV at the same time

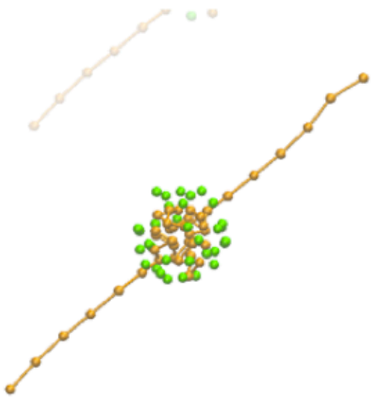
compact
t=0



extended
t=20000



clustered
t=40000



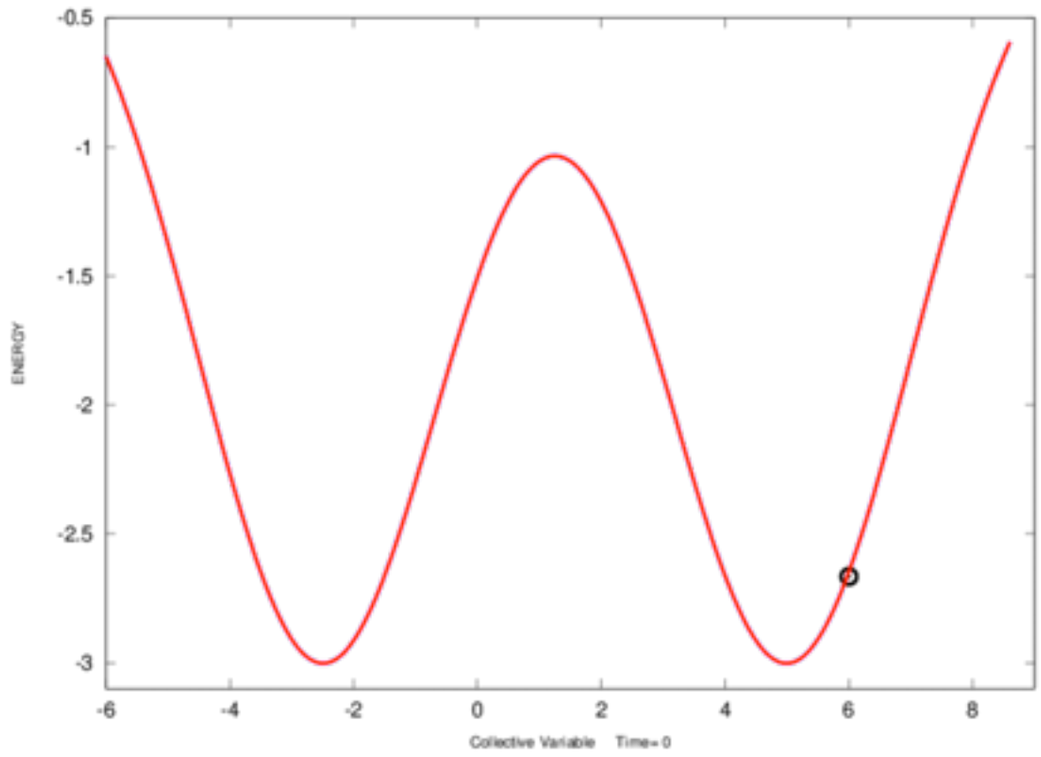
5_metadynamics

* A movie on metadynamics: place computational sand to remove metastabilities

$$V(\boldsymbol{s},t) = \int_0^t dt' \omega \exp\left(-\sum_{i=1}^{d} \frac{(s_i(\mathbf{x}) - S_i(\mathbf{x}(t'))^2}{2\sigma_i^2}\right)$$

5_metadynamics

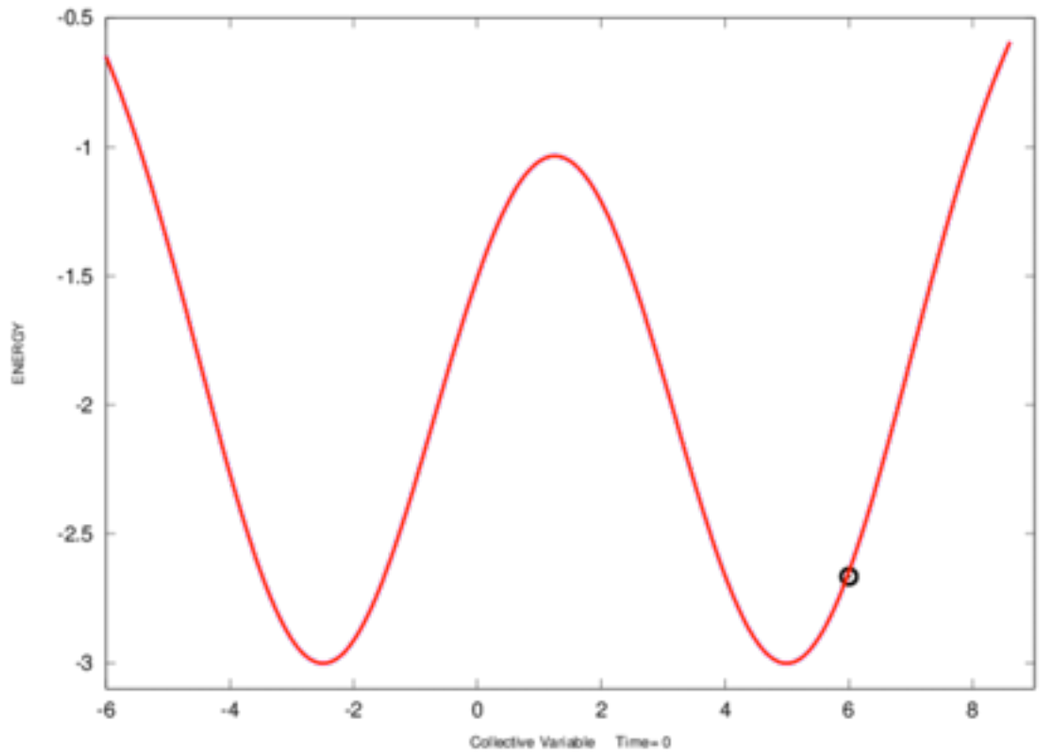
* A movie on metadynamics: place computational sand to remove metastabilities



$$V(\boldsymbol{s},t) = \int_0^t dt' \omega \exp\left(-\sum_{i=1}^{d} \frac{(s_i(\mathbf{x}) - S_i(\mathbf{x}(t'))^2}{2\sigma_i^2}\right)$$

5_metadynamics

* A movie on metadynamics: place computational sand to remove metastabilities

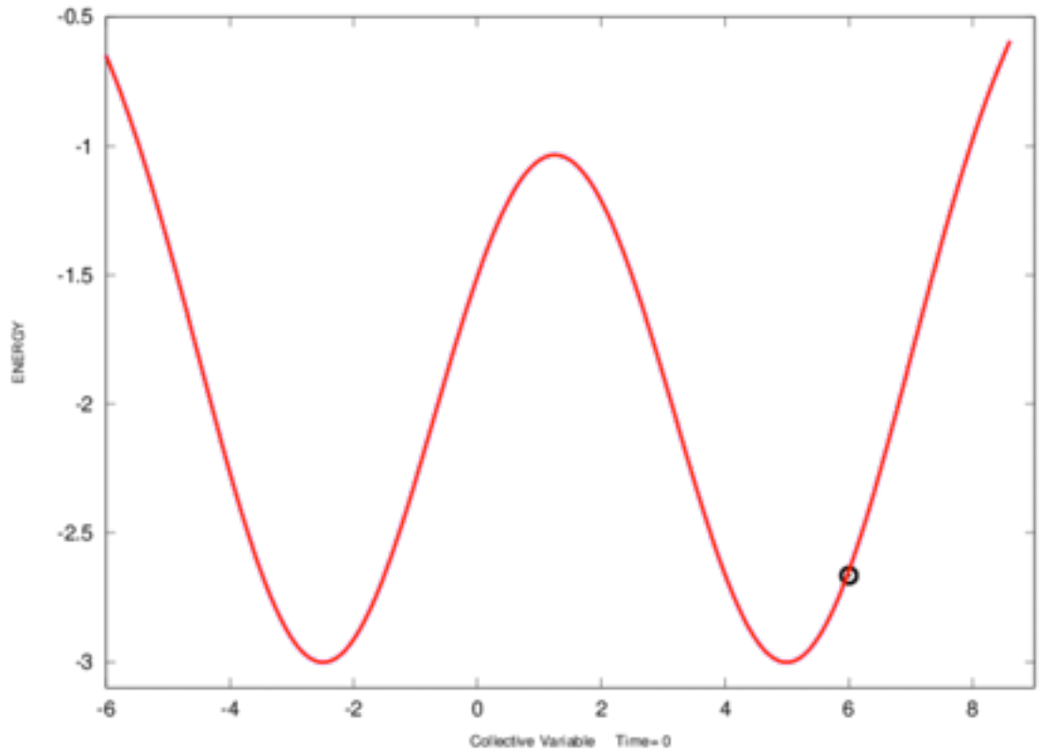


$$V(\boldsymbol{s},t) = \int_0^t dt' \omega \exp\left(-\sum_{i=1}^{d} \frac{(s_i(\mathbf{x}) - S_i(\mathbf{x}(t'))^2}{2\sigma_i^2}\right)$$

how often place the new Gaussian

5_metadynamics

* A movie on metadynamics: place computational sand to remove metastabilities

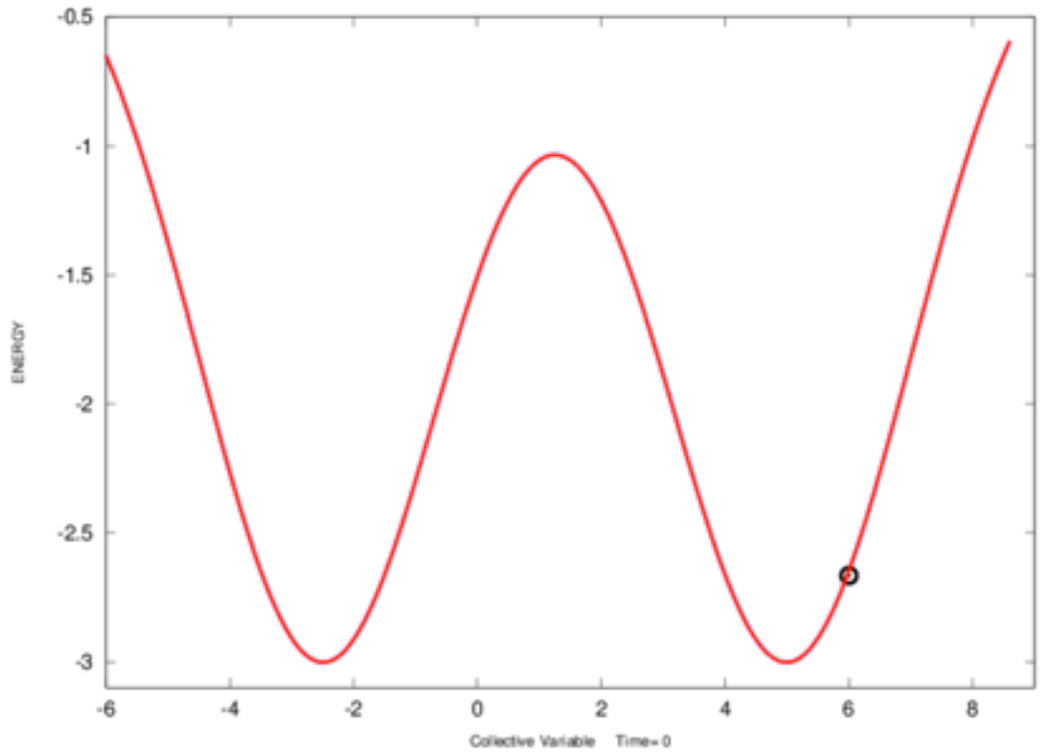


$$V(\boldsymbol{s},t) = \int_0^t dt' \omega \exp\left(-\sum_{i=1}^{d} \frac{(s_i(\mathbf{x}) - S_i(\mathbf{x}(t'))^2}{2\sigma_i^2}\right)$$

how high in energy is the Gaussian

5_metadynamics

* A movie on metadynamics: place computational sand to remove metastabilities

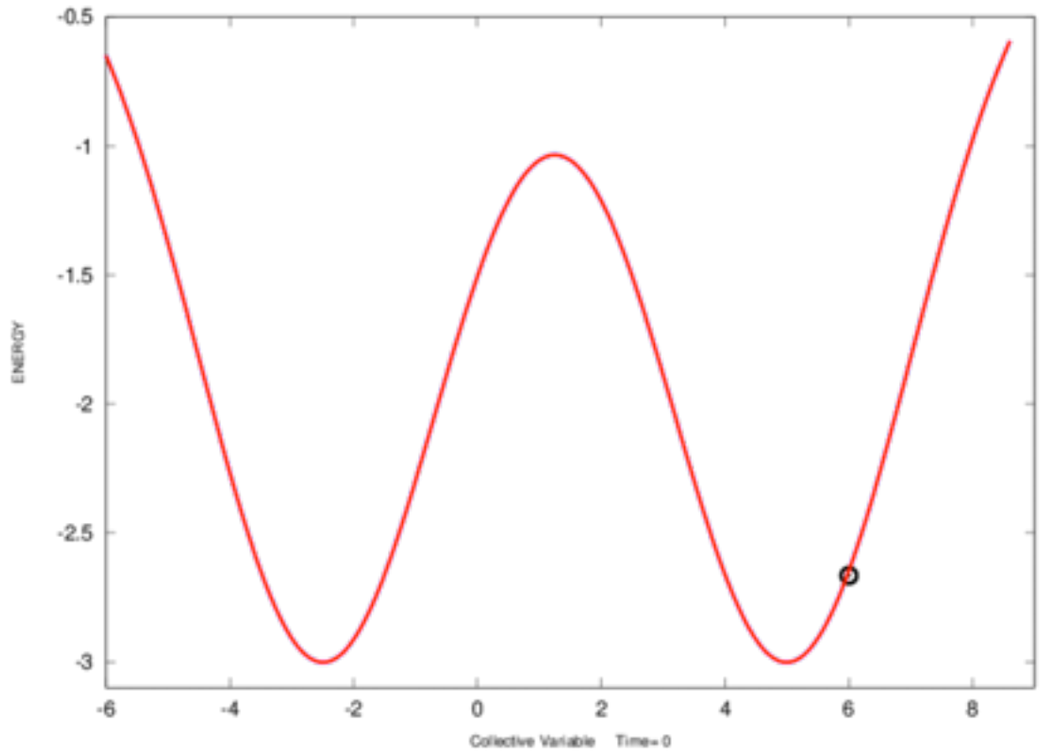


$$V(\boldsymbol{s},t) = \int_0^t dt' \omega \exp\left(-\sum_{i=1}^{d} \frac{(s_i(\mathbf{x}) - S_i(\mathbf{x}(t'))^2}{2\sigma_i^2}\right)$$

how wide is the Gaussian in CV units

5_metadynamics

* A movie on metadynamics: place computational sand to remove metastabilities



$$V(\boldsymbol{s},t) = \int_0^t dt' \omega \exp\left(-\sum_{i=1}^{d} \frac{(s_i(\mathbf{x}) - S_i(\mathbf{x}(t'))^2}{2\sigma_i^2}\right)$$

how wide is the Gaussian in CV units

plumed.dat

```
d1: DISTANCE ATOMS=1,40
meta: METAD  ARG=d1 SIGMA=4.0 HEIGHT=0.3 PACE=100
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

5_metadynamics

* You get a HILLS file (similar to COLVAR in some ways)

```
#! FIELDS time d1 sigma_d1 height biasf
#! SET multivariate false
    4    4.355    1   0.3    1
    8    7.579    1   0.3    1
   12    8.020    1   0.3    1
   16    5.873    1   0.3    1
   20    9.100    1   0.3    1
   24    9.672    1   0.3    1
   28    7.669    1   0.3    1
   32     5.39    1   0.3    1
   36    6.638    1   0.3    1
   40    7.314    1   0.3    1
   44     10.4    1   0.3    1
   48    13.33    1   0.3    1
   52    14.03    1   0.3    1
   56    14.70    1   0.3    1
```

Header: you have position, sigma, height, biasfactor* and multivariate** format specifier

*Well-Tempered metad

** Adaptive Gaussians

5_metadynamics

* To go from the HILLS file to the free-energy estimate you have to run plumed as standalone

```
plumed sum_hills --stride 300 --hills HILLS
```

5_metadynamics

* To go from the HILLS file to the free-energy estimate you have to run plumed as standalone

```
plumed sum_hills --stride 300 --hills HILLS
```

* It produces fes_0.dat fes_1.dat ... files. Let's visualize them

```
gnuplot> plot "fes_0.dat" w l, "fes_1.dat" w l, "fes_2.dat" w l
gnuplot> repl "fes_3.dat" w l, "fes_4.dat" w l
```

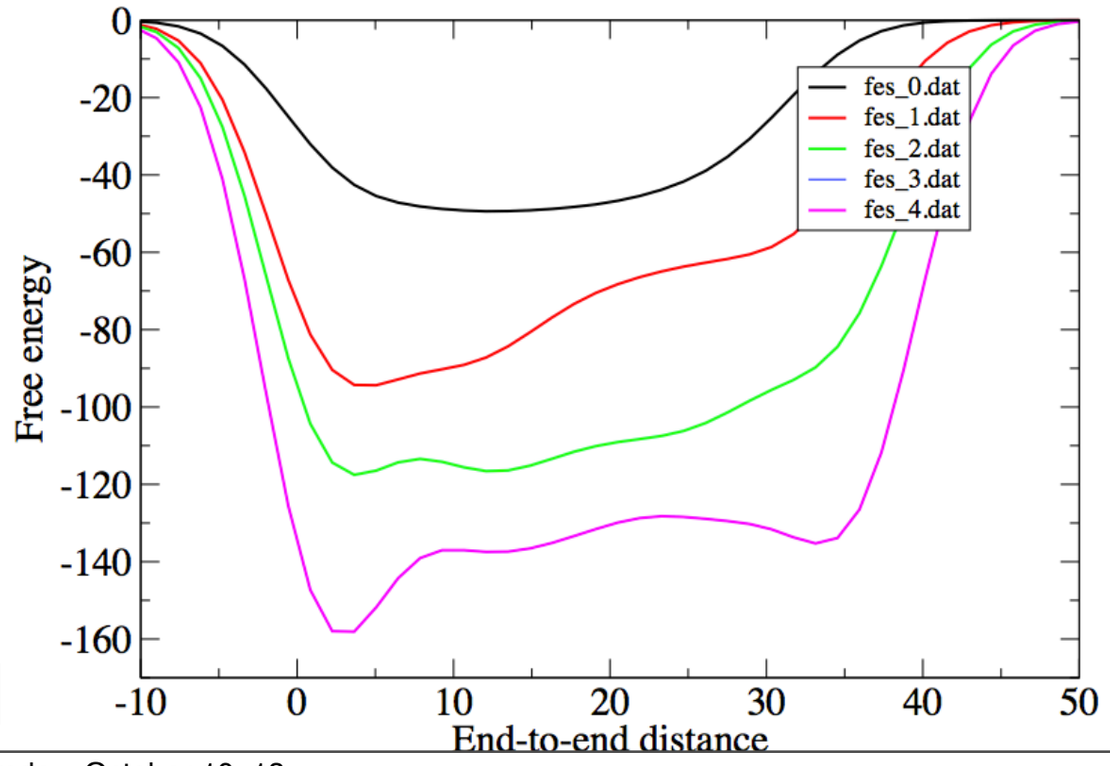
5_metadynamics

* To go from the HILLS file to the free-energy estimate you have to run plumed as standalone

```
plumed sum_hills --stride 300 --hills HILLS
```

* It produces fes_0.dat fes_1.dat ... files. Let's visualize them

```
gnuplot> plot "fes_0.dat" w l, "fes_1.dat" w l, "fes_2.dat" w l
gnuplot> repl "fes_3.dat" w l, "fes_4.dat" w l
```



Wait! The free energy does not converge!

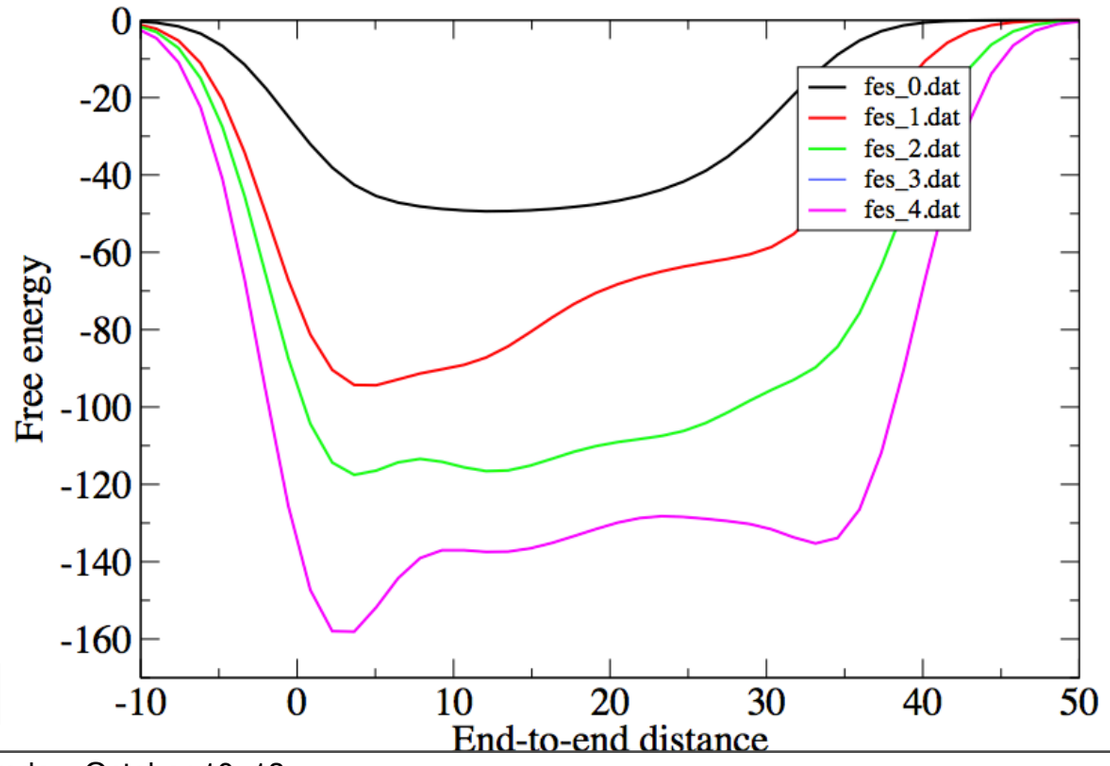
5_metadynamics

* To go from the HILLS file to the free-energy estimate you have to run plumed as standalone

```
plumed sum_hills --stride 300 --hills HILLS
```

* It produces fes_0.dat fes_1.dat ... files. Let's visualize them

```
gnuplot> plot "fes_0.dat" w l, "fes_1.dat" w l, "fes_2.dat" w l
gnuplot> repl "fes_3.dat" w l, "fes_4.dat" w l
```

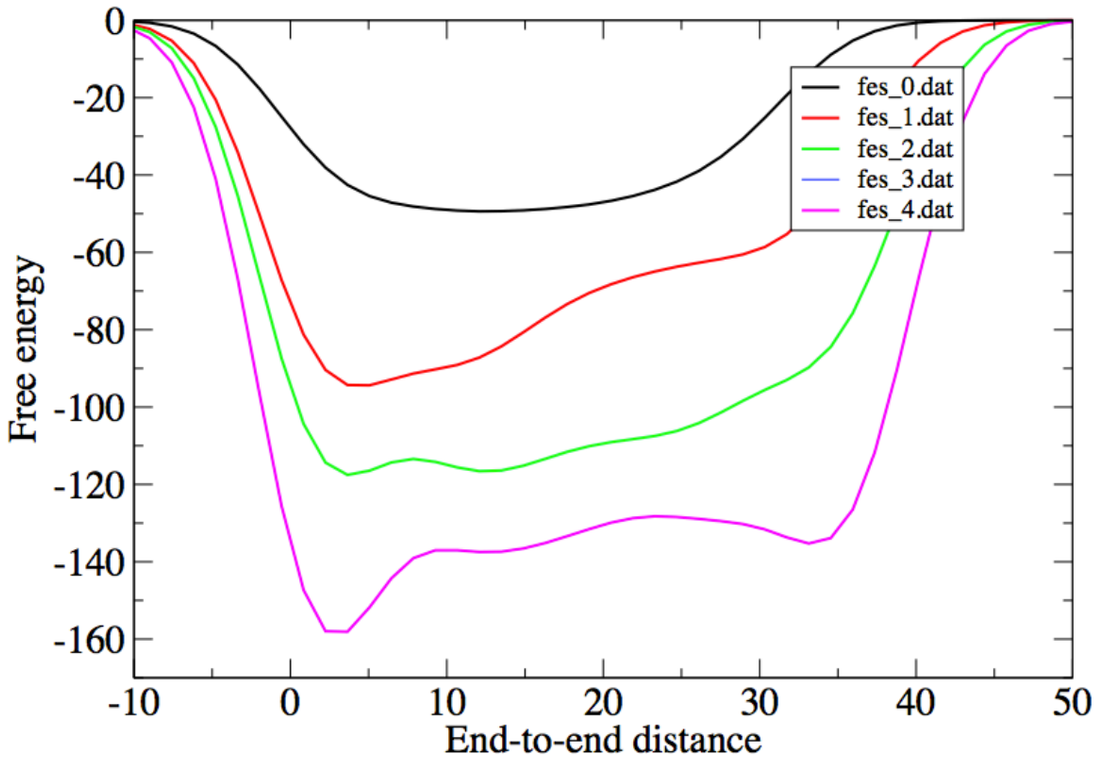


Wait! The free energy does not converge!

Remember: only relative differences matter!

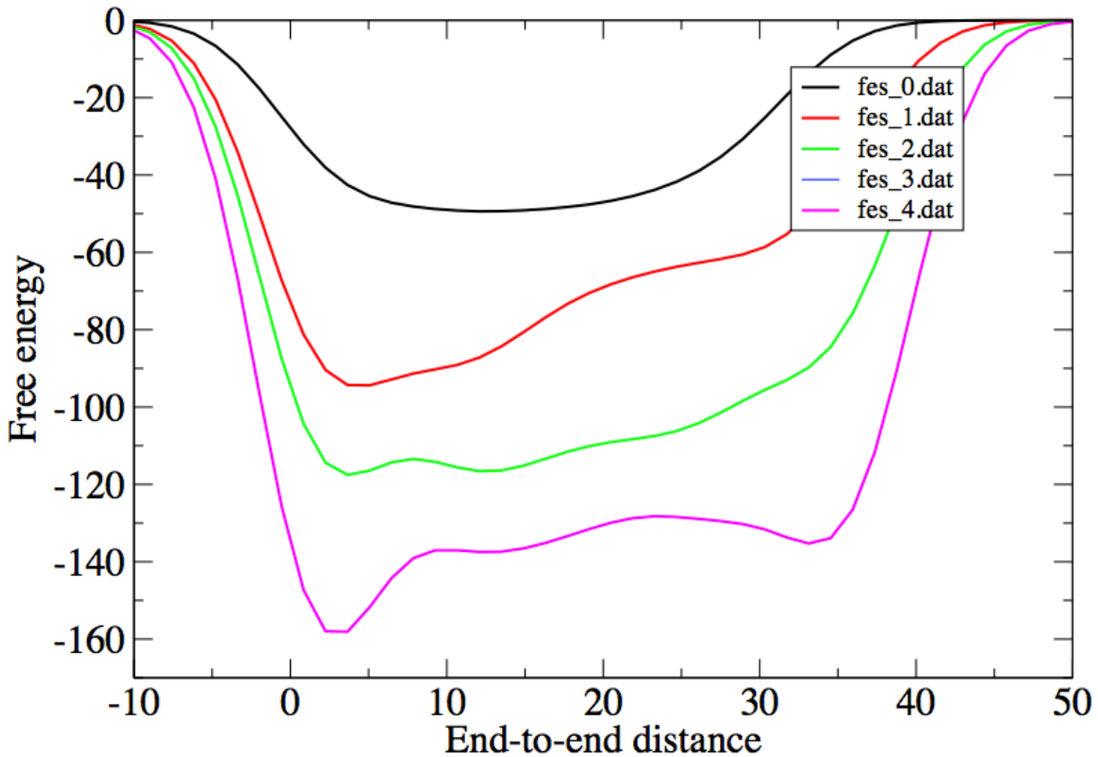
5_metadynamics

* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?



5_metadynamics

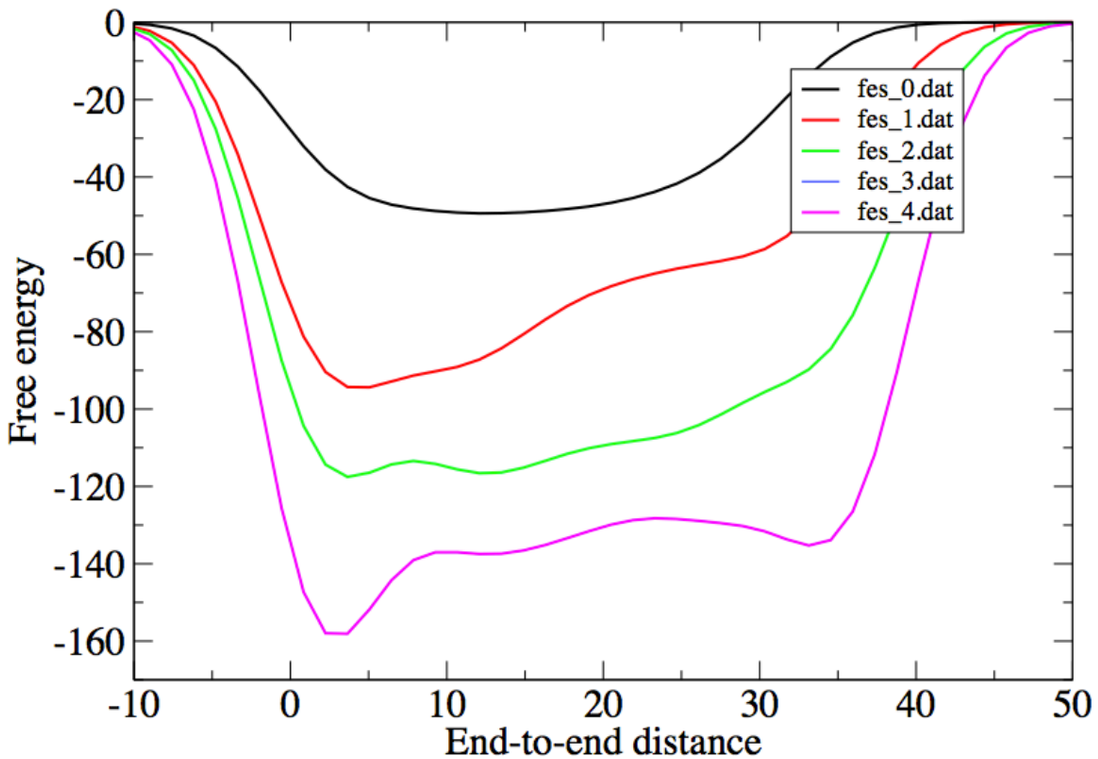
* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?



1) Parameters choice
2) Collective variable choice

5_metadynamics

* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?

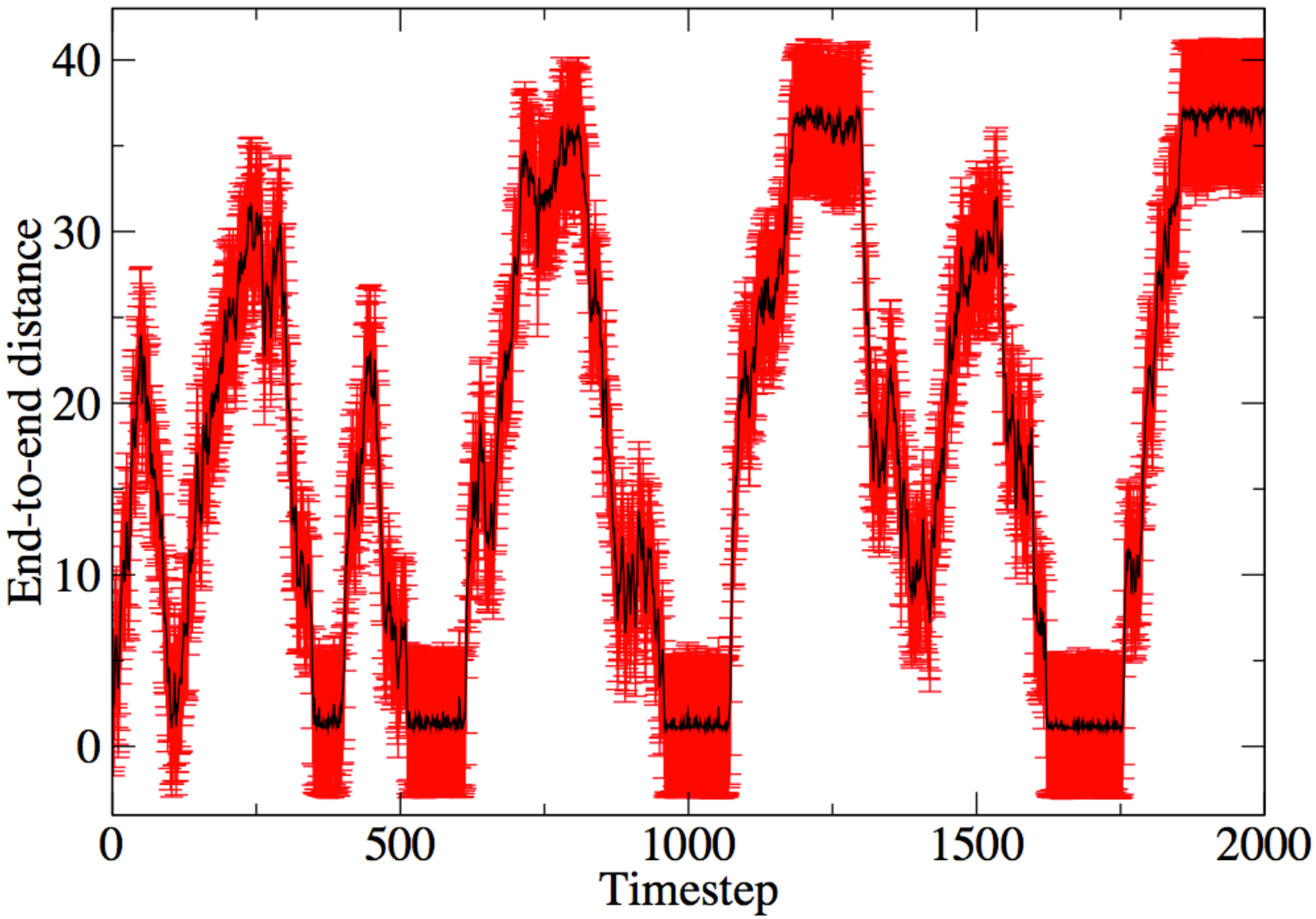


1) Parameters choice
2) Collective variable choice

5_metadynamics

* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?

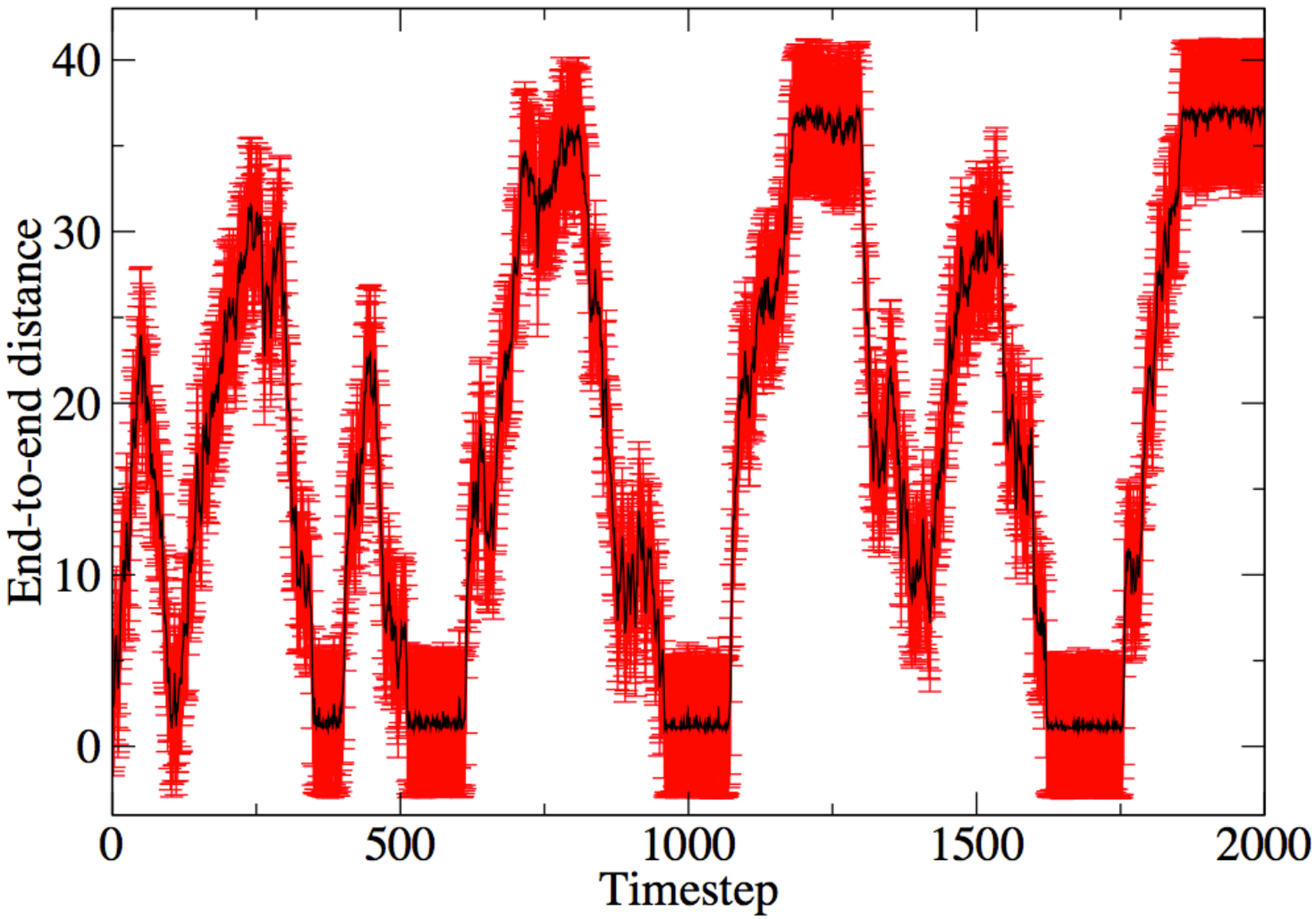
```
gnuplot> p "HILLS" u 1:2:3 w yer,"" u 1:2 w lp
```



5_metadynamics

* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?

```
gnuplot> p "HILLS" u 1:2:3 w yer,"" u 1:2 w lp
```

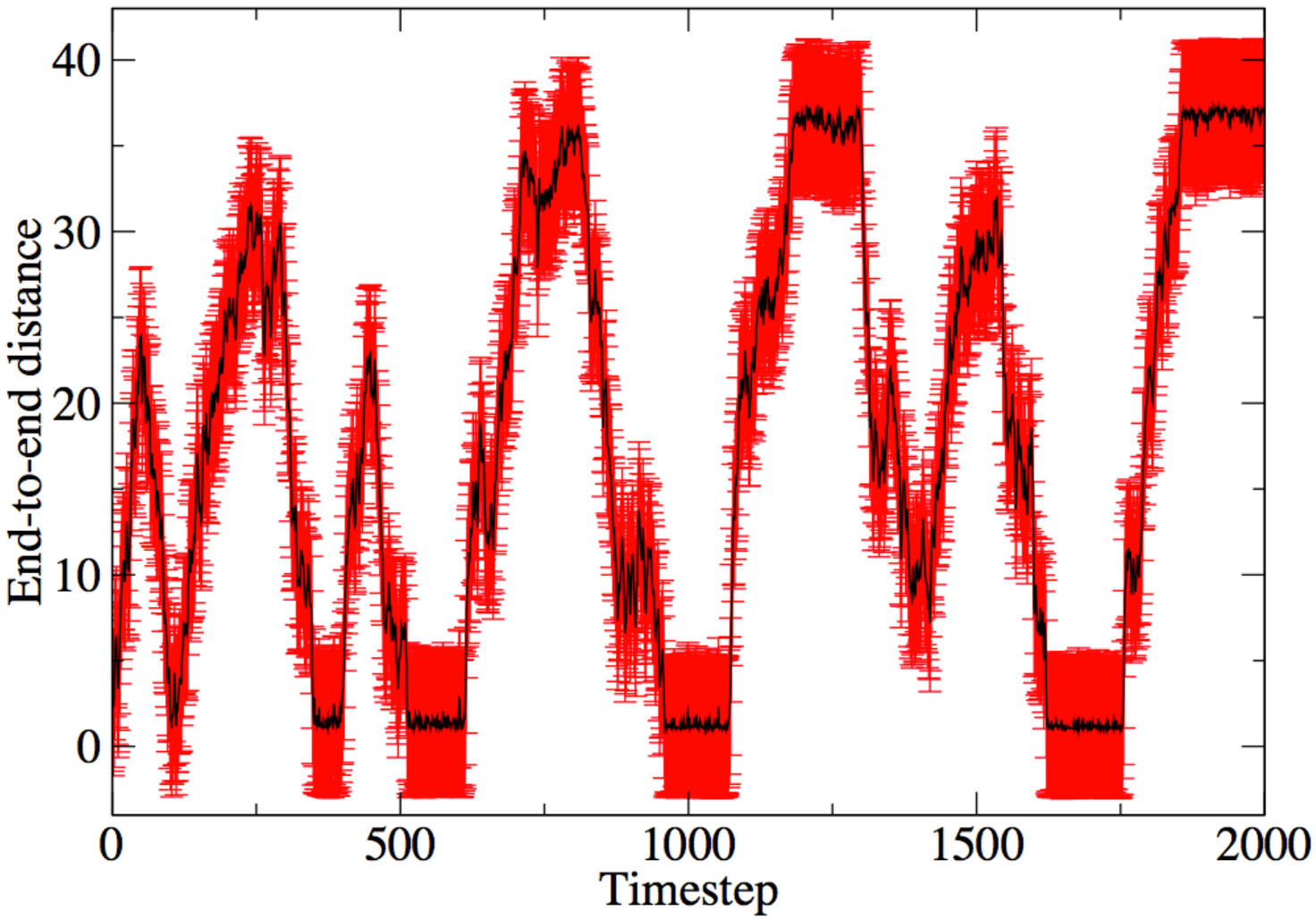


Surfing effect!

5_metadynamics

* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?

```
gnuplot> p "HILLS" u 1:2:3 w yer,"" u 1:2 w lp
```



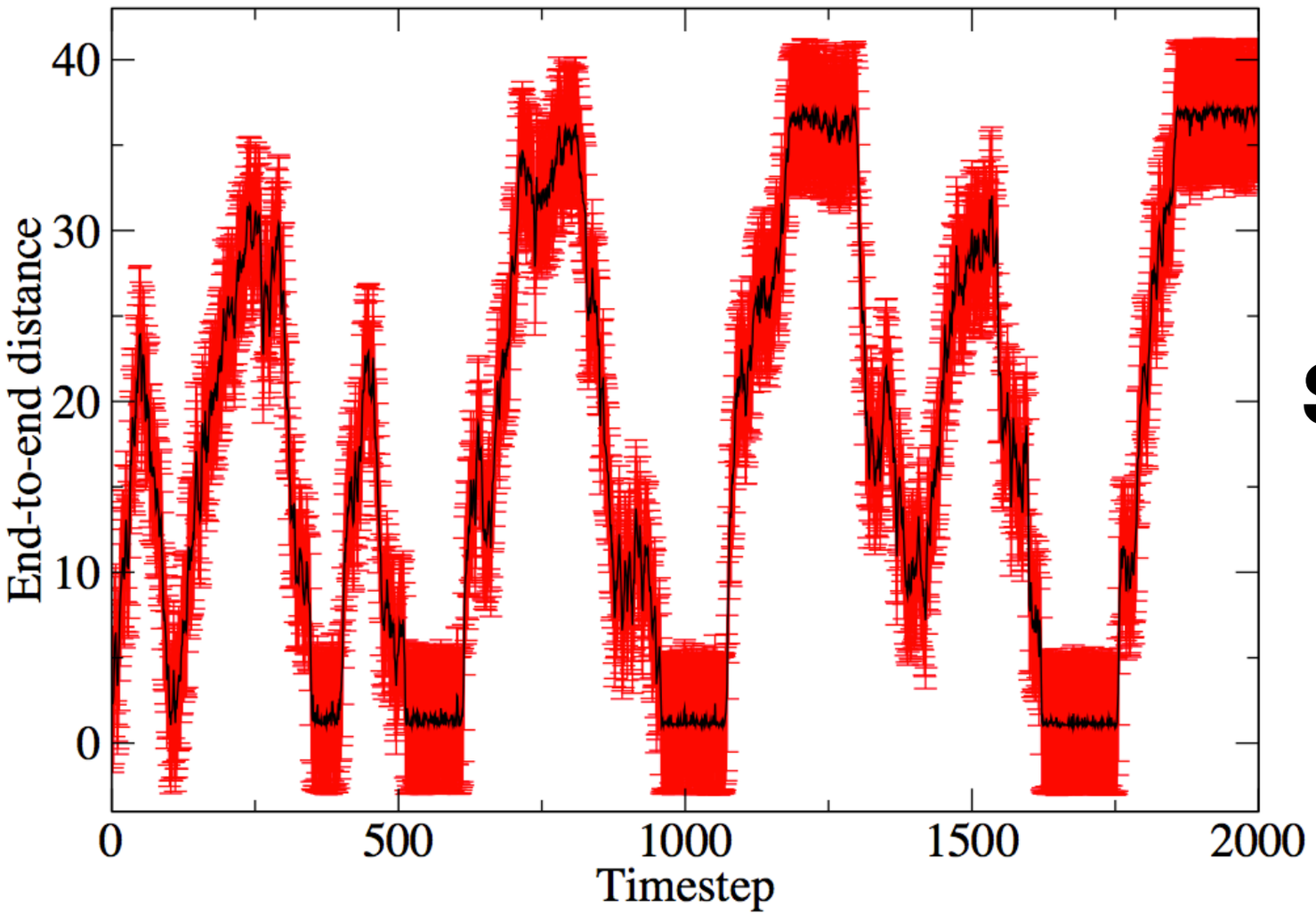
Surfing effect!

Smaller sigma

5_metadynamics

* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?

```
gnuplot> p "HILLS" u 1:2:3 w yer,"" u 1:2 w lp
```



Surfing effect!

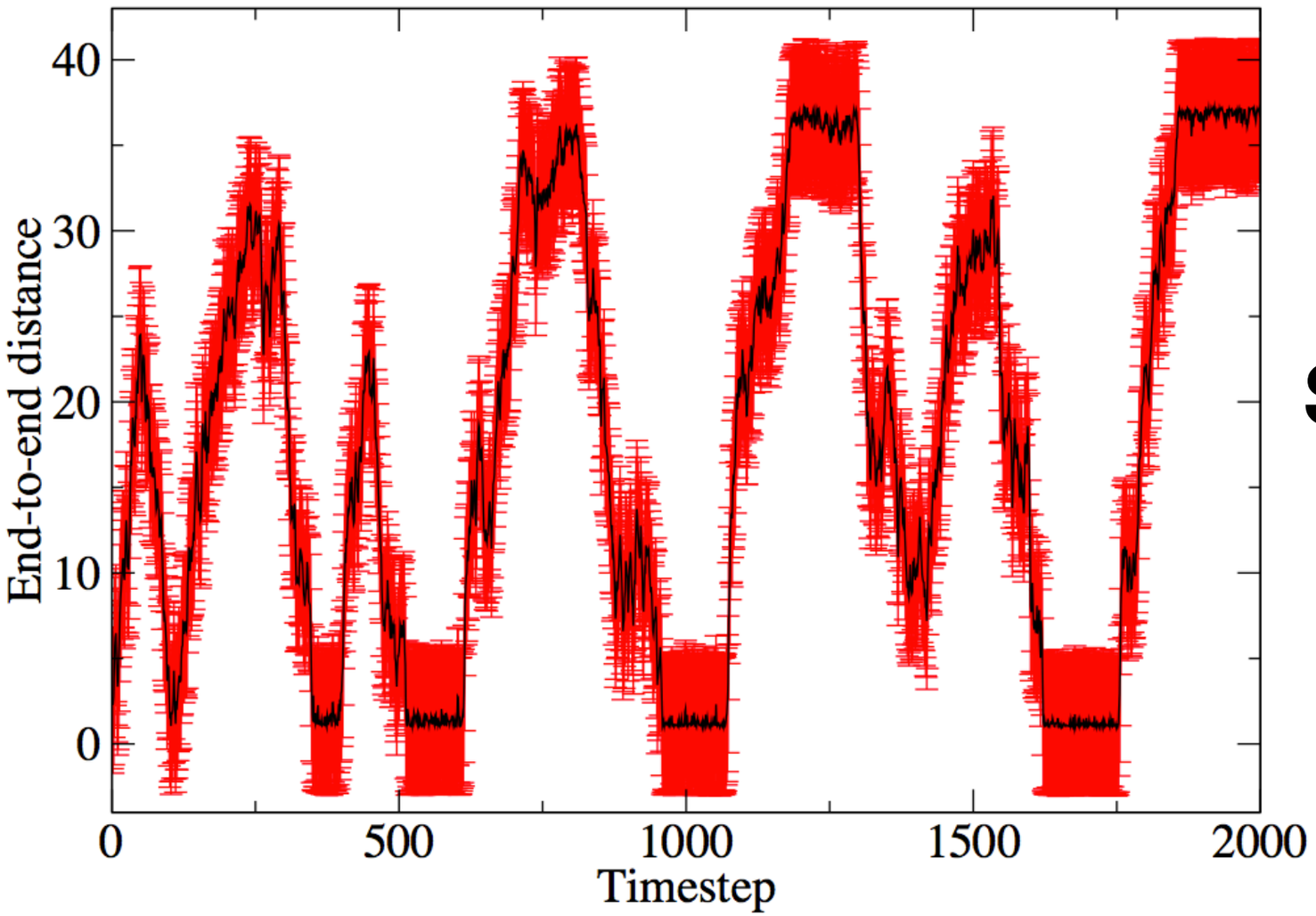
Smaller sigma

Smaller deposition rate (increase PACE)

5_metadynamics

* Anyway this free energy is bad....no clear stable state, no convergence. How to improve?

```
gnuplot> p "HILLS" u 1:2:3 w yer,"" u 1:2 w lp
```



Surfing effect!

Smaller sigma

Smaller deposition rate (increase PACE)

You cannot overcome the typical timescales!

5_metadynamics

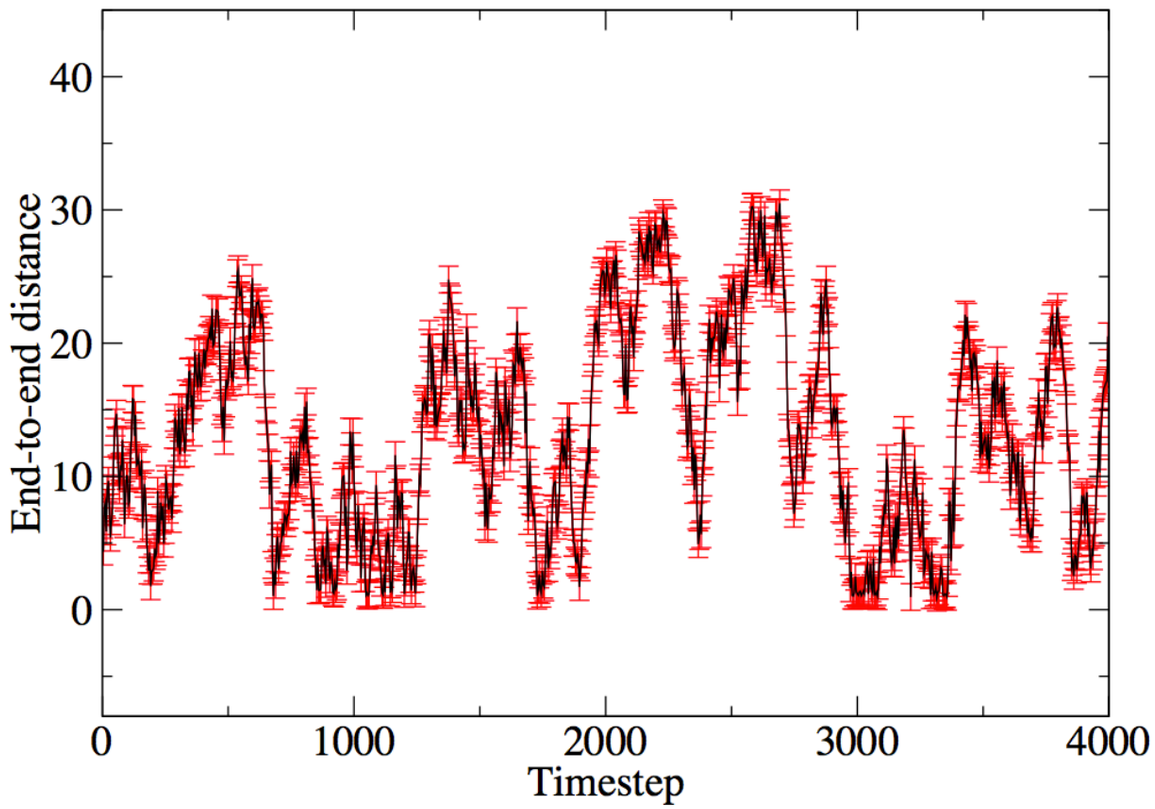
```
d1: DISTANCE ATOMS=1,40
meta: METAD  ARG=d1 SIGMA=1.0 HEIGHT=0.3 PACE=400
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

and increase the simulated timestep in the ESPResSo script!!

5_metadynamics

```
d1: DISTANCE ATOMS=1,40
meta: METAD  ARG=d1 SIGMA=1.0 HEIGHT=0.3 PACE=400
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

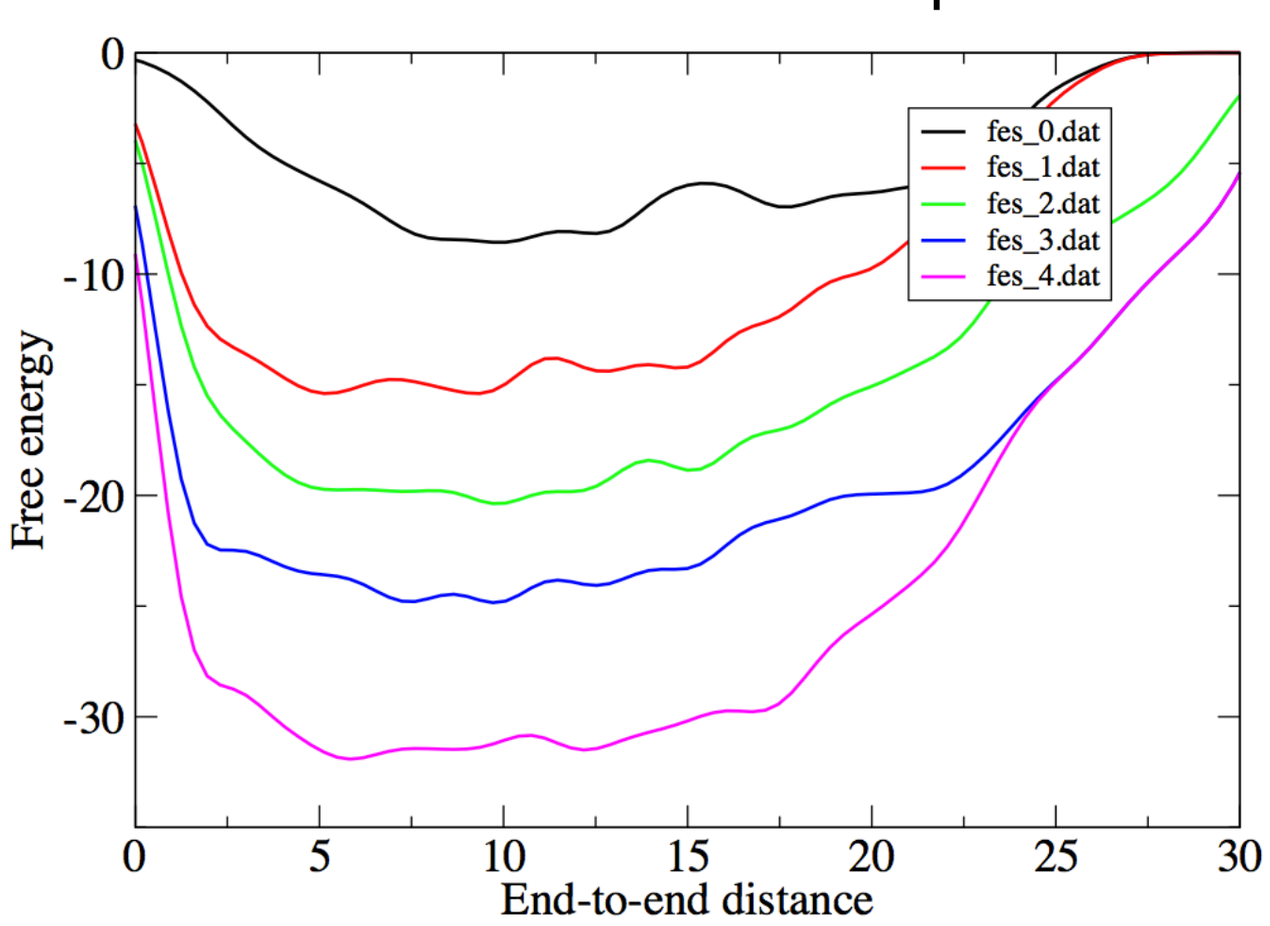
and increase the simulated timestep in the ESPResSo script!!



5_metadynamics

```
d1: DISTANCE ATOMS=1,40
meta: METAD  ARG=d1 SIGMA=1.0 HEIGHT=0.3 PACE=400
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

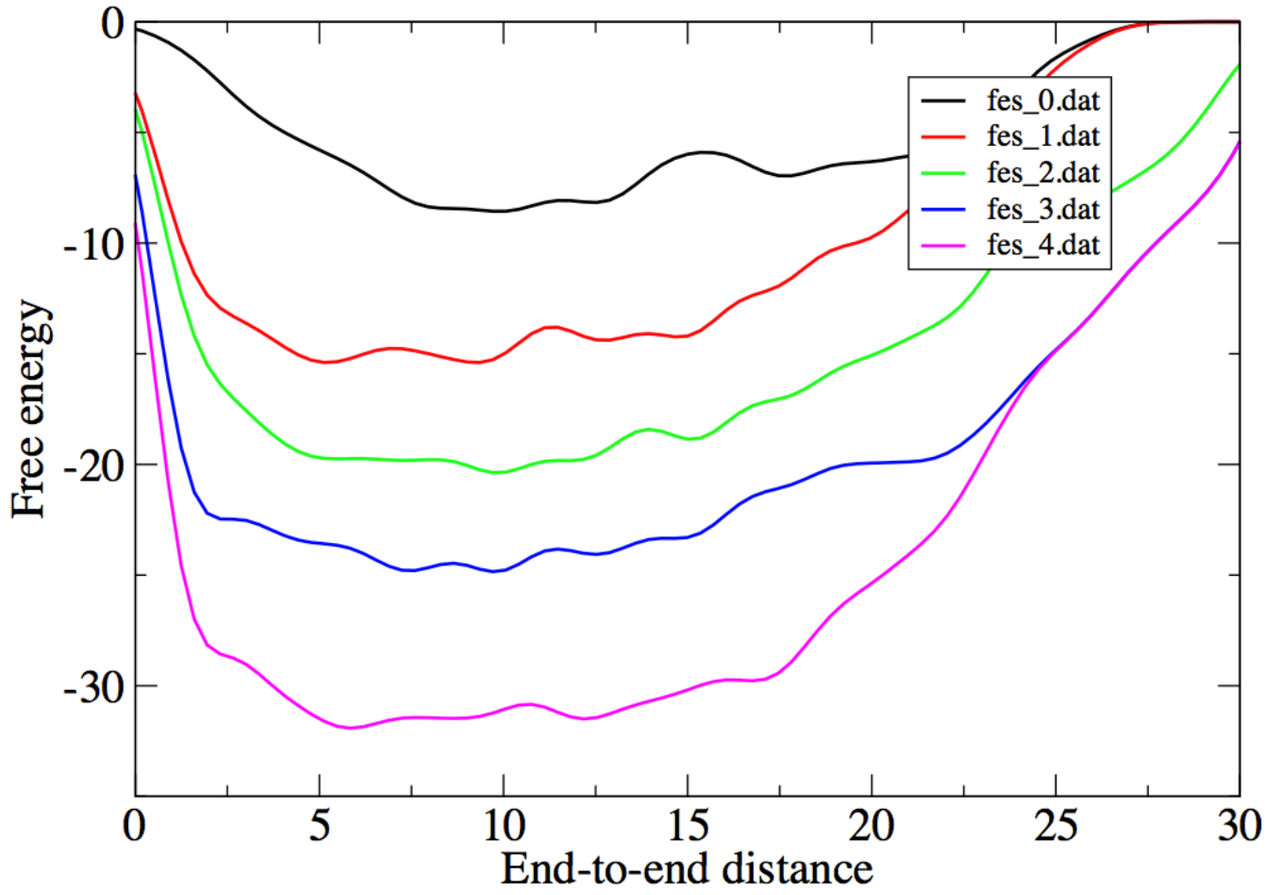
and increase the simulated timestep in the ESPResSo script!!



5_metadynamics

```
d1: DISTANCE ATOMS=1,40
meta: METAD  ARG=d1 SIGMA=1.0 HEIGHT=0.3 PACE=400
PRINT ARG=* STRIDE=50 FILE=COLVAR
```

and increase the simulated timestep in the ESPResSo script!!



Much better: grows parallel in a domain.

Other errors are mainly due to the CV!

Where do I go from here?

* Explore the CVs!!!! They are many! And can be combined to do your own CV in a sort of script!
* Try different methods. There is no “best method”! It depends on your system!
* Plumed is also an analysis tool
* Plumed can be a platform to invent your method and CVs
* Explore the documentation of PLUMED!!! (also developer doc)

PLUMED

USER DEVELOPER

Main Page | Glossary | How-tos

Introduction

Plumed is a plugin that works with a large number of molecular dynamics codes. It can be used to analyse features of the dynamics on-the-fly or to perform a wide variety of free energy methods. The original plumed 1.0 was highly successful and had over 1000 users. Plumed 2.0 constitues an extensive rewrite of the original in a way that makes it more modular and thus easier to implement new methods, more straightforward to add it to MD codes and hopefully simpler to use. This is the user manual - if you want to modify PLUMED or to understand how it works internally, have a look at the developer manual . Alternatively, to look at a list of the new features available in plumed 2.0 and to work out how to install plumed 2.0 in your MD code check out:

- Installation