**Dimerized formin code:**

**Input:**

**Single run:**

gcc -O3 driveMetropolis.c -o metropolis.out -lm

./metropolis.out parameters.txt outputfile verboseTF NFil N iSite baseSepDist Force dimerForce

% parameters.txt

% Where most of the parameters are.

% Don’t change this.

% outputfile

% Change to whatever you want the output file to be called

% verboseTF

% Set to 0 to give summary file, set to 1 to give verbose file (see below)

% NFil

% Sets number of filaments to simulate

% N

% Sets number of segments for all filaments (all will be same length)

% iSite

% Sets single iSite location - NOT USED - SET TO -1 (overwritten later anyway)

% baseSepDist

% How far apart the base of each filament should be placed

% Force

% Strength of force pulling on the ends of the filaments in the z direction

% dimerForce

% Strength of force pulling the ends of the filaments together

Filaments.txt and iSites.txt are no longer used under current settings. To use them instead of results from command line NFil/N/all iSites: change parameters.txt values for iSiteInputMethod from 4 to 2 and filamentInputMethod from 0 to 1.

If using text file input for filaments, iSites:

filaments.txt:

Text file specifying number of filaments (by number of lines) and length of each filament (number on line). Default is single filament (one line) of 60 segments.

iSites.txt:

Text file specifying number and locations of iSites on filaments. Each line corresponds to the iSites on a single filament. Multiple iSites can be specified by separating with spaces or commas on a single line. Note: iSites are indexed starting at 0.

**Output:** Matlab indexing used below. Green highlight indicates indexing required to access more than one filament.

**Summary:**

nt, // 1

NFil, // 2

irLigand, // 3

brLigand, // 4

Force, // 5

kdimer, // 6

dimerDist0, // 7

baseSepDistance, // 8

\**Note: N\_AlliSites = total iSites across all filaments*

For each (N\_AlliSites+1)

POcclude\_NumSites[i] // 8 + i

For each (N\_AlliSites+1)

PAvailable\_NumSites[i] // 8 + (N\_AlliSites+1) + i

For each filament:

N[nf], // 9 + 2\*(N\_AlliSites +1) + (6 + 7\*iSiteTotal[nf] + 2 + NFil + NFil + 2\*iSiteTotal)\*(nf-1)

ksStatistic[nf], // 10 + 2\*(N\_AlliSites +1)

reeBar[nf], // 11 + 2\*(N\_AlliSites +1)

ree2Bar[nf], // 12 + 2\*(N\_AlliSites +1)

rMBar[nf], // 13 + 2\*(N\_AlliSites +1)

rM2Bar[nf]); // 14 + 2\*(N\_AlliSites +1)

For each iSite:

iSite[nf][iy], // 15 + 2\*(N\_AlliSites +1) + 7\*(iy-1)

POcclude[nf][iy], // 16 + 2\*(N\_AlliSites +1) + 7\*(iy-1)

1-POcclude[nf][iy], // 17 + 2\*(N\_AlliSites +1) + 7\*(iy-1)

PMembraneOcclude[nf][iy], // 18 + 2\*(N\_AlliSites +1) + 7\*(iy-1)

Prvec0[nf][iy], // 19 + 2\*(N\_AlliSites +1) + 7\*(iy-1)

rMiSiteBar[nf][iy], // 20 + 2\*(N\_AlliSites +1) + 7\*(iy-1)

rM2iSiteBar[nf][iy]); // 21 + 2\*(N\_AlliSites +1) + 7\*(iy-1)

POccludeBase[nf], // 22 + 2\*(N\_AlliSites +1) + 7\*(iSiteTotal-1)

1-POccludeBase[nf]); // 23 + 2\*(N\_AlliSites +1) + 7\*(iSiteTotal-1)

For each filament a second time:

reeFilBar[nf][nf2]); // 24 + 2\*(N\_AlliSites +1) + 7\*(iSiteTotal-1) + (nf2-1)

For each filament a third time:

ree2FilBar[nf][nf2] // 25 + 2\*(N\_AlliSites +1) + 7\*(iSiteTotal-1) + (NFil-1) + (nf2-1)

For each iSite:

reeiSiteBar[nf][iy] // 26 + 2\*(N\_AlliSites +1) + 7\*(iSiteTotal-1) + (NFil-1) + (NFil-1) + 2\*(iy-1)

ree2iSiteBar[nf][iy] // 27 + 2\*(N\_AlliSites +1) + 7\*(iSiteTotal-1) + (NFil-1) + (NFil-1) + 2\*(iy-1)

**Verbose:**

nt, // 1

E, // 2

dChi[0], // 3

dChi[1], // 4

rate[0], // 5

rate[1], // 6

constraintProposalsTotal, // 7

For each filament:

ree[nf], // 8 + (4 + NFil + 3\*iSiteTotal + 1 + 3 + 3\*N + 3\*iSiteTotal)\*(nf-1)

rM[nf], // 9

rH[nf], // 10

ksStatistic[nf], // 11

For each filament a second time:

reeFil[nf][nf2] // 12 + (nf-1)

For each iSite:

stericOcclusion[nf][iy], // 13 + (NFil-1) + 3\*iy

membraneOcclusion[nf][iy], // 14 + (NFil-1) + 3\*iy

membraneAndSegmentOcclusion[nf][iy], // 15 + (NFil-1) + 3\*iy

stericOcclusionBase[nf]); // 16 + (NFil-1) + 3\*(iSiteTotal-1)

rBase[nf][0], // 17 + (NFil-1) + 3\*(iSiteTotal-1)

rBase[nf][1], // 18 + (NFil-1) + 3\*(iSiteTotal-1)

rBase[nf][2], // 19 + (NFil-1) + 3\*(iSiteTotal-1)

For each N:

r[nf][i][0], // 20 + (NFil-1) + 3\*(iSiteTotal-1) + 3\*i

r[nf][i][1], // 21 + (NFil-1) + 3\*(iSiteTotal-1) + 3\*i

r[nf][i][2], // 22 + (NFil-1) + 3\*(iSiteTotal-1) + 3\*i

For each iSite:

iLigandCenter[nf][iy][0], // 23 + (NFil-1) + 3\*(iSiteTotal-1) + 3\*(N-1) + 3\*iy

iLigandCenter[nf][iy][1], // 24 + (NFil-1) + 3\*(iSiteTotal-1) + 3\*(N-1) + 3\*iy

iLigandCenter[nf][iy][2] // 25 + (NFil-1) + 3\*(iSiteTotal-1) + 3\*(N-1) + 3\*iy