# README FOR GETGEOMETRY_V2.M, RUNGETGEOMETRY_V2.M

PETER A. EMBACHER

Tested on Windows 10, Matlab 2019b.

Used in: Embacher PA, Germanova TE, Roscioli E, McAinsh AD, Burroughs NJ: Bayesian inference of multi-point macromolecular architecture mixtures at nanometre resolution

**Input files.** Allowed input files are text files with the following formatting:

```
MYFIRSTLINE
comment:     MYCOMMENT
nostates:    Z
nomarkers:   J
dim:         3
nodata:      N




f1_1_x f1_1_y f1_1_z
f2_1_x f2_1_y f2_1_z
...
fJ_1_x fJ_1_y fJ_1_z
f1_2_x f1_2_y f1_2_z
...
fJ_N_x fJ_N_y fJ_N_z
```

Where:

- `MYFIRSTLINE` is an arbitrary single line (is skipped when reading).
- `MYCOMMENT` is an arbitrary string, that will be saved in the output files for identification.
- `Z` is the number of polygon states we want to analyse in the dataset (read as an integer).
- `J` is the number of polygon nodes/fluorophore colours (read as an integer).
- `N` is the number of (full polygon) datapoints (read as an integer).
- `fj_n_d` is the d-th space component of the j-th fluorophore of the n-th datapoint (read as a double).

The five blank lines between nodata and the first line containing data are all skipped when reading.

*Date*: 02/15/22.

**Settings.** Open "rungetgeometry_v2.m"; the following settings can be chosen:

- in line 7: `filename` is a string and has to be set to the name of the input file (including filetype).
- in line 8: `subsamplefreq` can be set to any integer $\geq 1$ how many Markov chain iterations are computed per one recorded sample (i.e. for "`subsamplefreq=1`" every sample is recorded, for "`subsamplefreq=2`" every second sample is recorded,...). Note, the total number of recorded samples is always set to 10000, so a higher `subsamplefreq` increases the total number of samples. If a Markov chain should not converge, a higher number for the `subsamplefreq` can help.
- in line 9: `nochains` can be set to any integer $\geq 1$ and denotes how many independent Markov chains are run. The chains are run in parallel, as far as enough CPUs are available to Matlab.
  In the paper this was chosen as 5.
- in line 10: `sorttype` decides how the state labels will be alocated, if multiple states with the same prior are inferred; can be set to '`none`' (no reallocation of labels), '`circumference`' (state-labels assigned based on sum of distances between all pairs of polygon nodes), '`proportion`' (state labels assigned based on state-proportion in first sub-dataset).
  If `nostates`$= 1$, this setting has no influence on the outcome. For the multi-state examples in the paper, we used: '`circumference`' for the two-state examples with CenpC–Ndc80C–Ndc80N, Nnf1–Ndc80C–Ndc80N, '`proportion`' for all others.
- in line 12: `posstatevskin` is a `Z`$\times$`N` sized matrix of logicals; for each state and each datapoint its entry is "`true`", if the state is allowed for this datapoint, and "`false`" otherwise. Each datapoint has to have at least one allowed state.
  This can be used for "informing datasets", i.e. datasets, where not all states are allowed.
  A (commented out) example is given in line 13.
- in line 14: `datasetvskin` is a `S`$\times$`N` sized matrix of logicals, where `S` is the number of sub-datasets in the input file (all sub-datasets share the same states in terms of polygon templates and measurement errors, but have independent state proportions). The number `S` of sub-datasets in implicitly defined by the user by setting datasetvskin. The entries are "`true`", if the datapoint is in the respective dataset, otherwise "`false`". Each datapoint has to be in exactly one dataset.
  A (commented out) example is given in line 15.

**Preliminaries.** Place the files of both functions (`rungetgeometry_v2.m`,`getgeometry_v2.m`) and the input file in the same folder.

**Run and output.** Execute `rungetgeometry_v2` with Matlab.

The program will start running, giving regular updates about its current state in Matlab's controlwindow. A text-file is created named "getgeometry controlwindow output.txt", recording the controlwindow output, while the program runs. A subfolder is created named after the date and time, when the program starts, and the suffix "getgeometry output". This will contain all output files, particularly:

- The Matlab structures underlying the MCMC, one for each independent Markov chain - this would allow to extract the Markov chain for further post-processing within Matlab.
- Text files of the Markov chain evolution, one for each independent Markov chain - this would allow to read the Markov chain evolution for further post-processing with non-Matlab programs.
- Matlab plots (.fig, .png) for the summary statistics of all independent Markov chains combined; particularly the Markov chain evolution and marginal posterior histograms for:
  - the lengths between any pair of markers (and each state)
  - the measurement error of the xy- and z-direction of each marker (and each state)
  - the state proportion of each state and each sub-dataset

Means and standard deviations of these marginal posteriors are also output in the controlwindow together with their respective Gelman-Rubin statistic (called `GRR_simple`).

The output text-file is formatted as follows:

```
version:       2
DATE
filename:      filename
comment:       MYCOMMENT
chaincomment:  CHAINCOMMENT
nostates:      Z
nomarkers:     J
dim:           3
nodatasets:    S
nodata:        N
samplerange:   [BURNIN,MAXITERATION]
poststatevskin:
posstatevskin1_1 posstatevskin1_2 ...posstatevskin1_N
posstatevskin2_1 posstatevskin2_2 ...posstatevskin2_N
...
posstatevskinZ_1 posstatevskinZ_2 ...posstatevskinZ_N
datasetvskin:
datasetvskin1_1 datasetvskin1_2 ...datasetvskin1_N
datasetvskin2_1 datasetvskin2_2 ...datasetvskin2_N
...
datasetvskinS_1 datasetvskinS_2 ...datasetvskinS_N
lengthprior: rectangle
+0.00000e+00 +Inf ...
merrprior: gamma
-5.00000e-01 +0.00000e+00 ...
pprior: none
+0.00000e+00 ...
```

1

X1_1_x X1_1_y X1_1_z
X1_2_x X1_2_y X1_2_z
...
XZ_J_x XZ_J_y XZ_J_z
$\sigma$1_1_xy $\sigma$1_1_xy $\sigma$1_1_z
$\sigma$1_2_xy $\sigma$1_2_xy $\sigma$1_2_z
...
$\sigma$Z_J_xy $\sigma$Z_J_xy $\sigma$Z_J_z
T1_1_x T1_1_y T1_1_z
T1_2_x T1_2_y T1_2_z
...
TZ_N_x TZ_N_y TZ_N_z
R1_1_xx R1_1_xy R1_1_xz
R1_1_yx R1_1_yy R1_1_yz
R1_1_zx R1_1_zy R1_1_zz
R1_2_xx R1_2_xy R1_2_xz
...
RZ_N_zx RZ_N_zy RZ_N_zz
p1_1 p1_2 ...p1_S p2_1 ...pZ_S
$\zeta$1 $\zeta$2 ...$\zeta$Z
$\log(\mathbb{L} \cdot \pi_X)$

2

...

---

Where (additionally to the quantities already introduced above):

- DATE is the date and time, when the process started running.
- CHAINCOMMENT is the number designated to the independent Markov chain by the program ($\leq$nochains).
- BURNIN is the burnin.
- MAXITERATION is the total number of Markov chain iterations.