Conformer Filtration (v4.0.0)

Break Change: not compatible with early version ~v3.3.0

Contents

Some Useful Codes Options:

Conformers

We define conformers as the molecules varying at their atomic relative positions, which include variation in bonds and angles. Conformer is used as a symbol name to differentiate the same type molecules changing at atomic relative positions due to bonds length variations and single bond rotations.

For example, bond length changes;

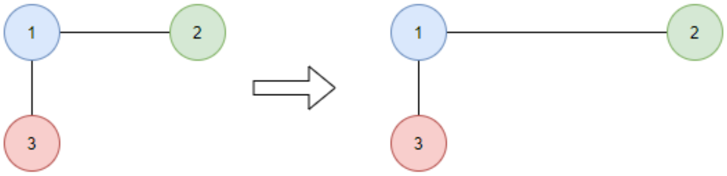


Figure 1: Molecules Vary in Bonds

Angle varying;

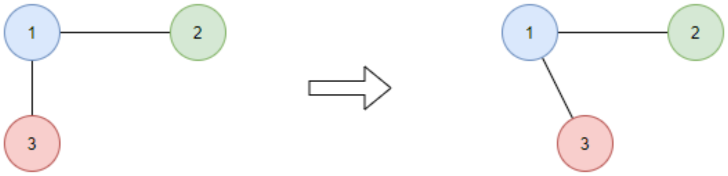


Figure 2: Molecules Vary in Angles

System and Fragment

Filtration is based on system, which is a single molecule used for defining bond/angle connections. Thus, it is a broad name to descript the inner connection information but not refer to any specific molecules.

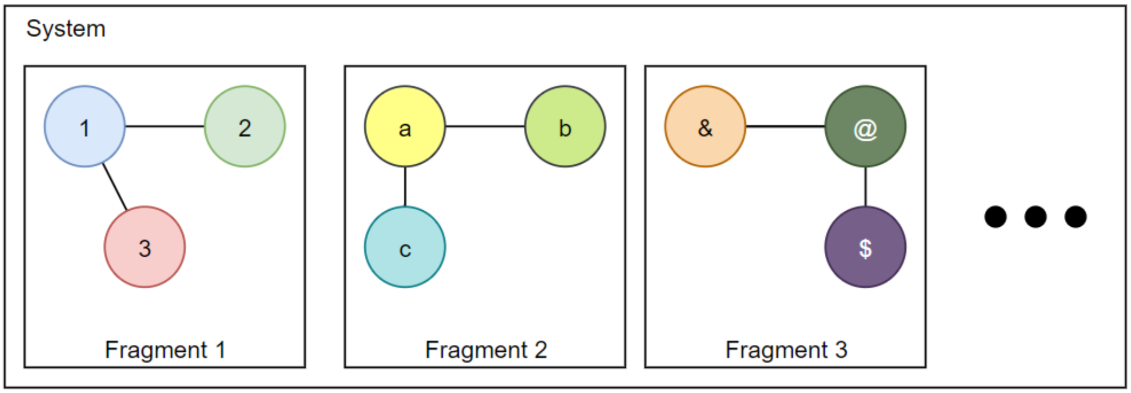


Figure 3: System and Fragments

From above figure, the system may be "split" into many different pieces, and we define them as fragments. Apparently, fragments have no bonded connections with each other.

Connections

In Figure 3, we can get their inner connection is; for bonds:

```
Fragment 1:   1-2,   1-3
Fragment 2:   a-b,   a-c
Fragment 3:  &-@, @-$
```

Connection in angles will be got in similar way.

As a notice, the filtration is based on the input connections, thus different connections will cause different filtration results even for the same set data inputs.

So, it comes to the question, how can we find the optimal connection?

Here are some assumptions.

If BOSS (Biochemical and Organic Simulation System, invented by Dr. Jorgensen) is used for the generations of input data set, since it can only make atomic variations in every Monte Carlo step, so the filtration is performed on the summary of overall changes for the input connections.

For simulation packages, they are built with powerful constraint algorithms, e.g., LINCS for GROMACS; SHAKE for AMBER; SETTLE for water models. They are all served to one common purpose, to hold the "integrity" of the input system, mainly to avoid two extreme artifacts, e.g., broking into pieces or flying ice problem, the former makes system exploded, while the latter causes system to be frozen into rigid body.

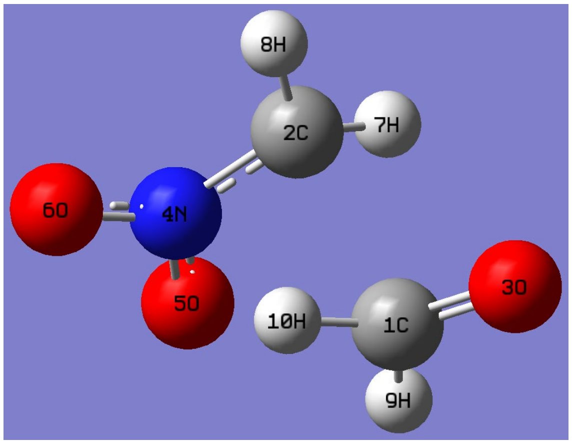


Figure 4: Fragments in Henry reaction

Therefore, if we trust their simulation results, as shown in Figure 4, there is no meaning to do calculation in bond connections `2-7, 2-8, 2-4, 4-5`, .., etc., or angle connections `7-2-8, 7-2-4, 8-2-4`, …, etc.

The focus is paid on the connection among cross-fragments. Thus the suggested bond connections in setting to choose are, `2-1, 2-3, …, 7-1, 7-3, …,` etc.

Because single bond rotation will change the relative positions of its "connected atomic groups", for example, the rotation for group `2,7,8` in single bond `2-4` will change angle for `7-2-5, 7-2-6,` …

However, those changes can be represented by the bond connections, `7-5, 7-6`, …

In conclusion, you should use your best judgement to set up those connections. Please bearing in mind that using the bonds connections is always suggested, due to its faster computation than angle's.

From codes view, to ignore bonds connections calculation: `-bcon []`, similarly, to ignore angles connections calculation, `-acon []`. Any number of square-bracket can be used.

All vs Par

If we have the connection settings:

```
bonds:    1-2,     1-4,      1-5,     1-6
angles:   1-3-2,   1-3-4,    1-3-5
```

We can either perform calculation in all of them or in separately. They are three modes provided:

overall: combine bonds and angles
all: bonds or angles connections
par: individual bonds or angles

The detail implementations will be discussed in Dynamic vs Static. For setting `all`, which means we conclude either all bonds or either all angles into their own considerations, thus only two probability analyses results will be got after filtration is done.

However, if the `par` is chosen, the probability analyses results will be printed out based on your input connection settings. For example, we can analyze the bond par `1-2`, or angle par `1-3-4` probability results.

Because `par` calculations are very time-consuming, they are only executed when you explicitly specify the Boolean option, `-obpar`, `-oapar`.

Userinputs

We human prefer counting things starting from `1`, however, it is not the truth for most computer languages, e.g., Python, SHELL, C, the list will go on, they are starting from `0`.

So from the codes view, there is a Boolean value `userinputs` to indicate whether the connections are starting from `1 (True)`, or starting from `0 (False)`.

Tolerance

Filtration is performed on tolerances: bond (in Angstrom), and angle (in degree).

We define that the conformers whose changes smaller than the setting tolerance will be filtered out. Here is an example by illustrating in bond variation.

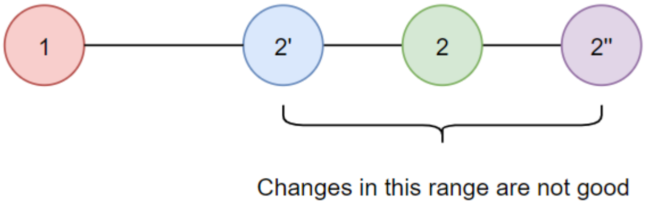


Figure 5: Illustration filtration on bond tolerance

If there are two atoms, by changing position of atom `2`, new molecules will be made. For filtration, if we have,

```
Abs(Distance1-2 – Distance1-2') < btol
Abs(Distance1-2 – Distance1-2") < btol
```

The molecules `1,2'`, and `1-2''` are not good, they are called repeats and will be removed.

Showing in plot, it is like the atomic position variation is not allowed in certain range along axis, see Figure 5. If it is shown in 3D space, the range will be a sphere. Any new molecules generated in those ranges will be thought as repeats.

For angle, if shown in similar way, it will be a cone in 3D space, corresponding to setting `atol`.

File Type Explanation

Currently, only following file formats are supported. The file type is determined by extension.

For txt file:

```
molecules are separated by new line, line starts with char '#' will be ignored,
if any errors happen, the whole set will be skipped.
note: energy info has to be at the end of the first line, equal sign, =, can be used.
       [   # x x x [=] energy
       |   atom-1   x    y    z
mol |   atom-2   x    y    z
       |   atom-3   x    y    z
       |   ...
       [   <new line>
```

For xsf file:

```
molecules are separated by '#', keyword 'ATOMS' is important,
case sensitive, if any errors happen, the whole set will be skipped.
note: energy info has to be at the end of the first line, equal sign, =, can be used.
       [   # x x x [=] energy
       |
       |   ATOMS
mol |   atom-1   x    y    z    else
       |   atom-2   x    y    z    else
       |   atom-3   x    y    z    else
       [   ...
```

For xyz file:

```
molecules are separated by new line, line starts with char '#' will
be ignored, if any errors happen, the whole set will be skipped.
note: energy info has to be at the end of the first line, equal sign, =, can be used.
       [   #  x x x [=] energy
       |   atom-1   x    y    z    else
mol |   atom-2   x    y    z    else
       |   atom-3   x    y    z    else
       [   ...
```

Those information can be printed by option: `-p` or `--file-format-explanations`.

Dynamic vs Static

There are two modes can be selected, dynamic and static mode, where,

```
dynamic: filtration is based on the inputs, changes on the inputs may cause different results
static: filtration is only related with starting index and input tolerance
```

Given a dataset, on the basis of different tolerances and connections, the molecules inside it can be split into many small groups, they can be represented in histograms, and we call them "bins". Molecules inside the same histograms are thought as the "similarities", the actual meaning of "repeats".

To be unified, we will use bins and repeats in the following contents instead.

For the different type of connections, they can be concluded into "complexity", a high level dimensional representation. Thus, we can get a plot like;

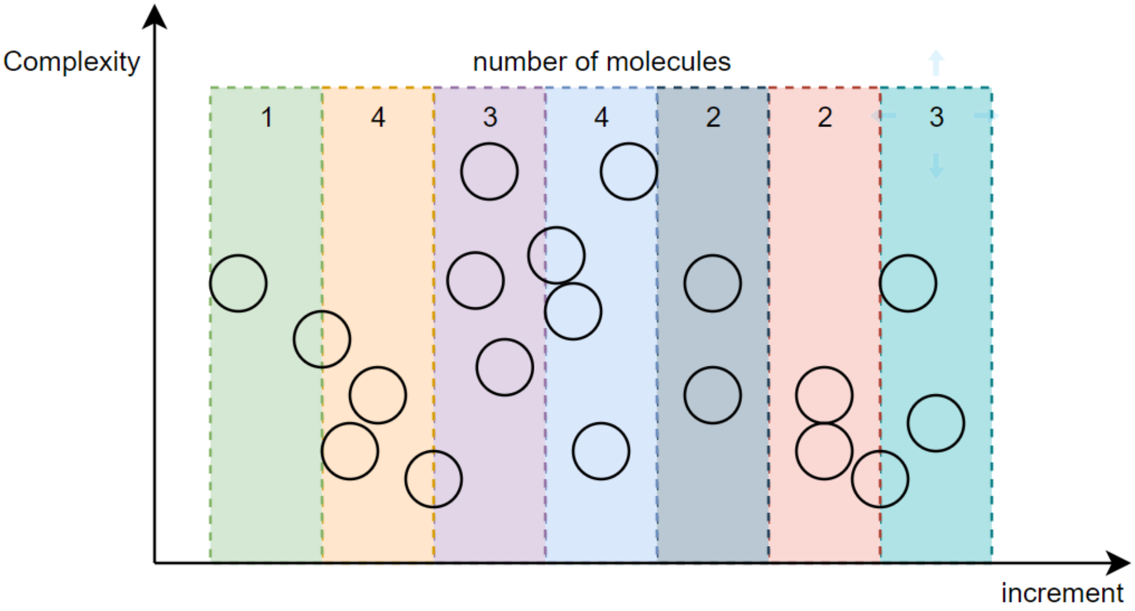


Figure 6: Illustration on Categorizing Molecules

In above figure, there are in total 19 molecules. For the input increment, like complexity, which is a summary term of bonds and angles tolerances, total dataset can be split into 7 bins, the number of molecules in each bin are shown in the top.

To filter them out, only one molecule will be left in every bin.

Dynamic Filtration

Assume there is an edge molecule, the "edge molecule" means the molecule happens to fall on the boundary of the next increment.

During the dynamic filtration, if this type of molecule is happened to be added (by adding more BOSS generated Backup files) or removed (by using index file, the early processed file), the bins will be dramatically changed, please see graph in below.

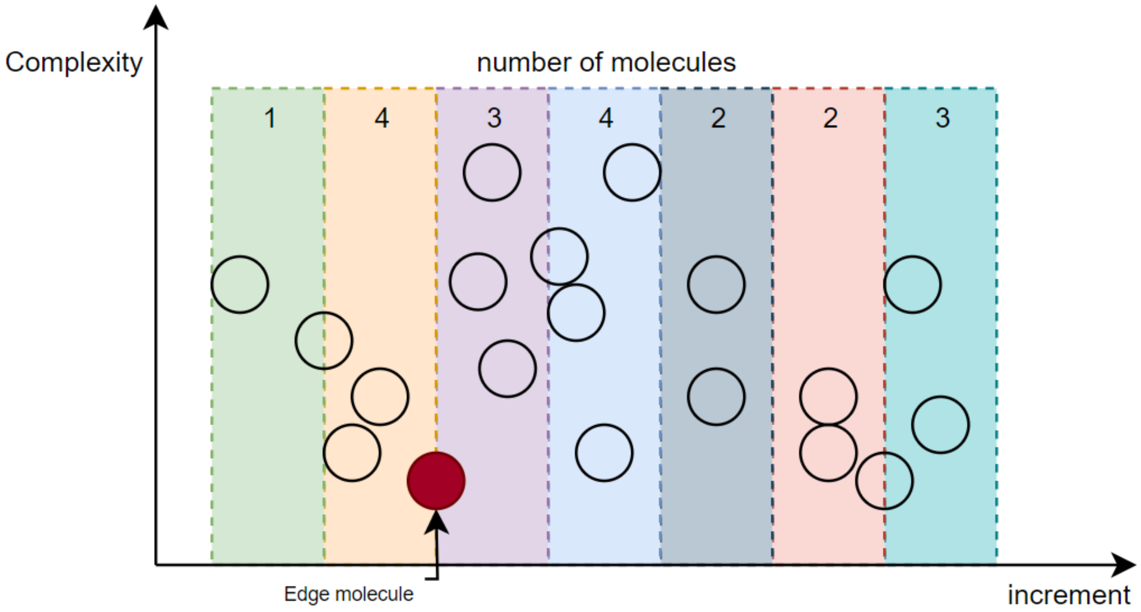


Figure 7: Illustration of Edge Molecule

By removing the edge molecule, it may have "gap" created between two adjacent bins. As a result, all the following bins will be accordingly changed.

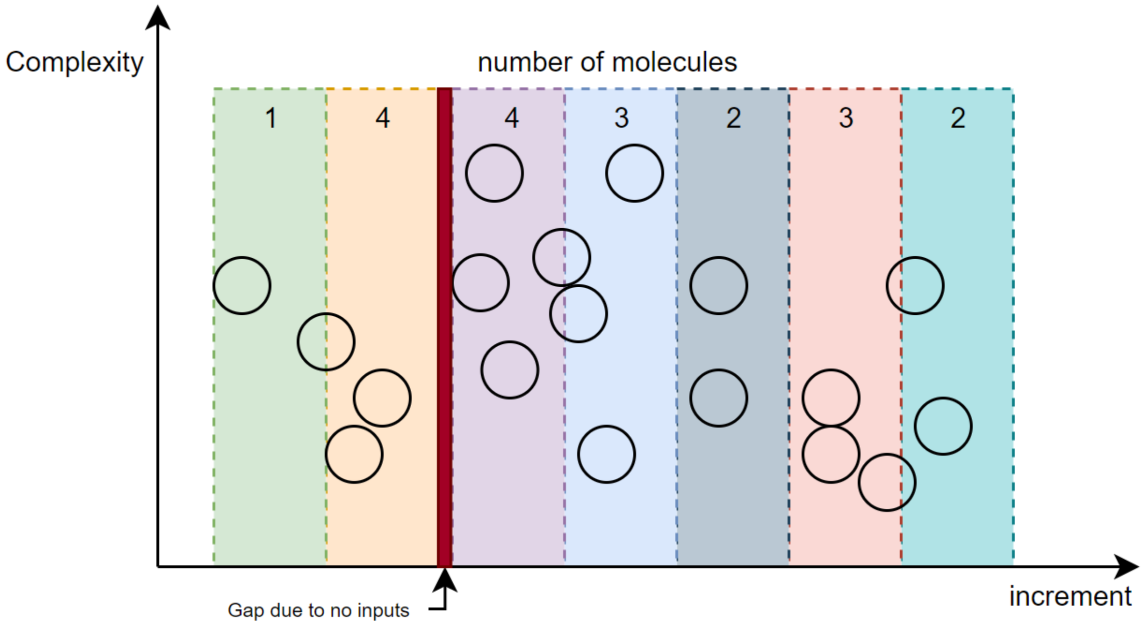


Figure 8: Effects on Removing of Edge Molecule

Since we have understood that by changing the edge molecule will cause the modification of bins, how can it influence the final filtration result? The answer is in the lowest-bit filtration.

Molecule in lowest-bit means the molecule close to the left edge of the bin and in the smallest complexity. To help your understanding, here is a plot;

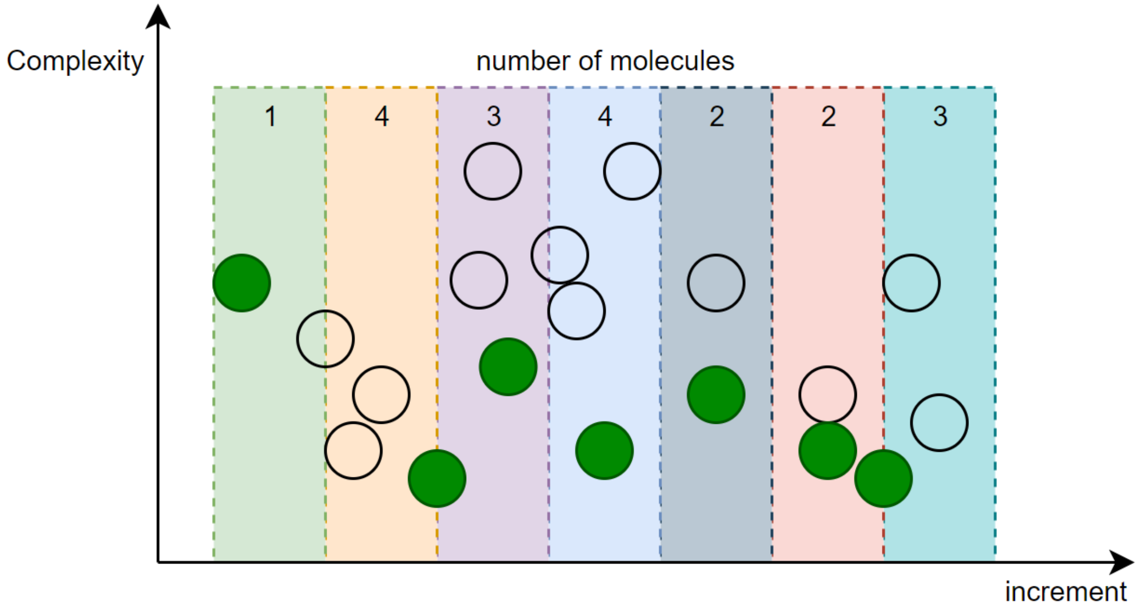


Figure 9: Illustration of Molecules in Lowest-bit (In blue)

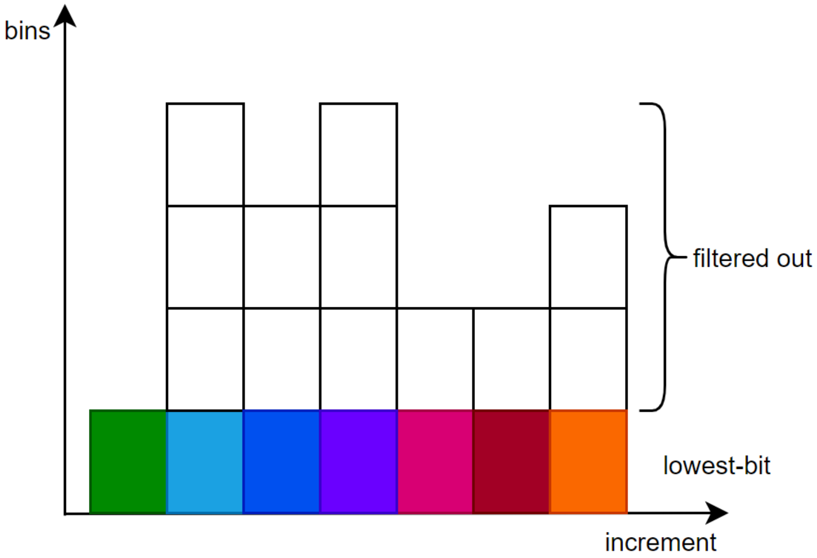


Figure 10: Filtration in Lowest-bit

All molecules in each bin except the molecules in lowest-bit will be removed during dynamic filtration. Therefore, we say its result is based on the input files and call it "dynamic".

In dynamic mode, there are two types calculations are provided, whether in `all` or in `separate`.

```
all: bonds changes + angles changes >= bonds tolerance + angle tolerance
separate: bonds changes >= bond tolerance    or    angles changes > angle tolerance
```

It is easy to be understood. The separate mode can be turned on by option `--separate`.

Static Filtration

On the contrary of dynamic mode filtration, there is the static mode filtration.

In static filtration, there are no edge molecules labelled, the bins are kept in continuous, and no gaps will be allowed. Thus, one of this main feature is that it allows zero-bin exists.

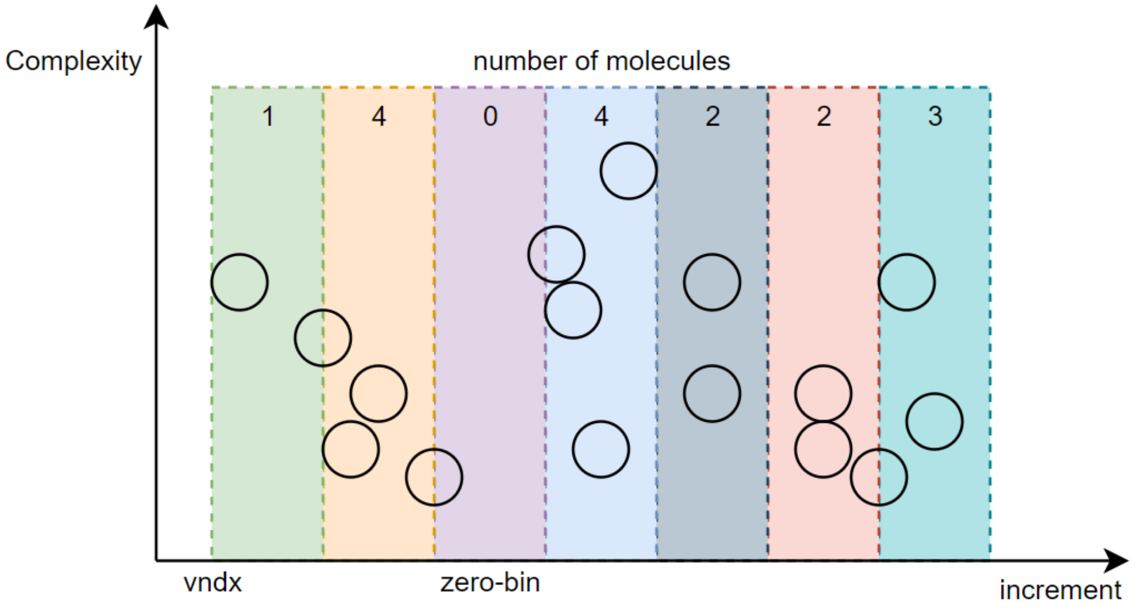


Figure 11: Illustration of Static Filtration

Addition parameter can be provided, `--vndx`, which is a number where to anchor filtration. This value can be any number, if not provided, the smaller lowest-bit molecule will be used instead.

Besides, apart from lowest-bit filtration, the random filtration is also provided.

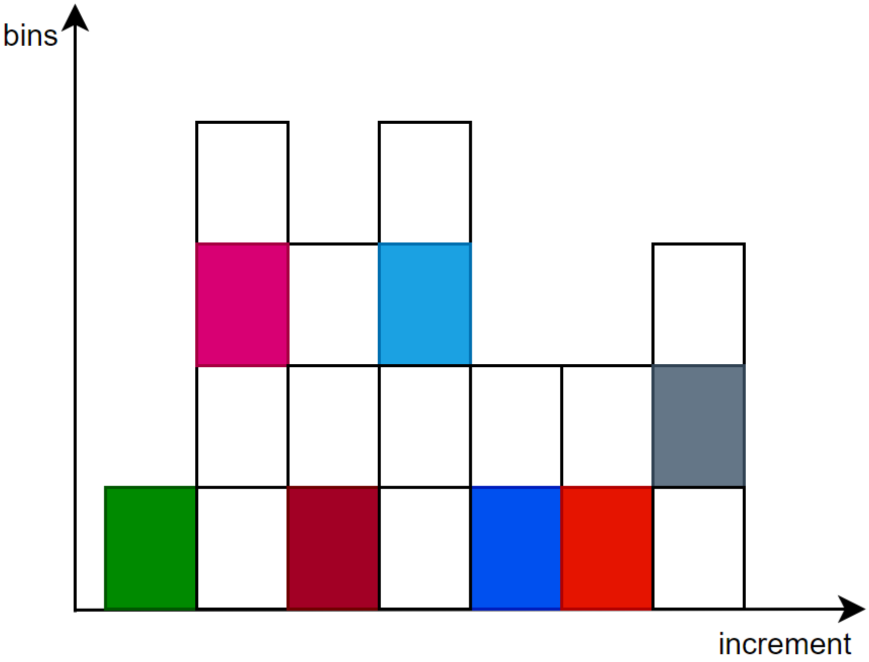


Figure 12: Illustration of Static Random Filtration

If the random filtration is turned on, all molecules in blank in each bin will be filtered out, the correspondent option is `--borandom`, which can be reproduced by `--seed`.

In static mode filtration, the filtration is based on `all` changes (discussed in above), because there is no valid way to deal with the difference of magnitudes

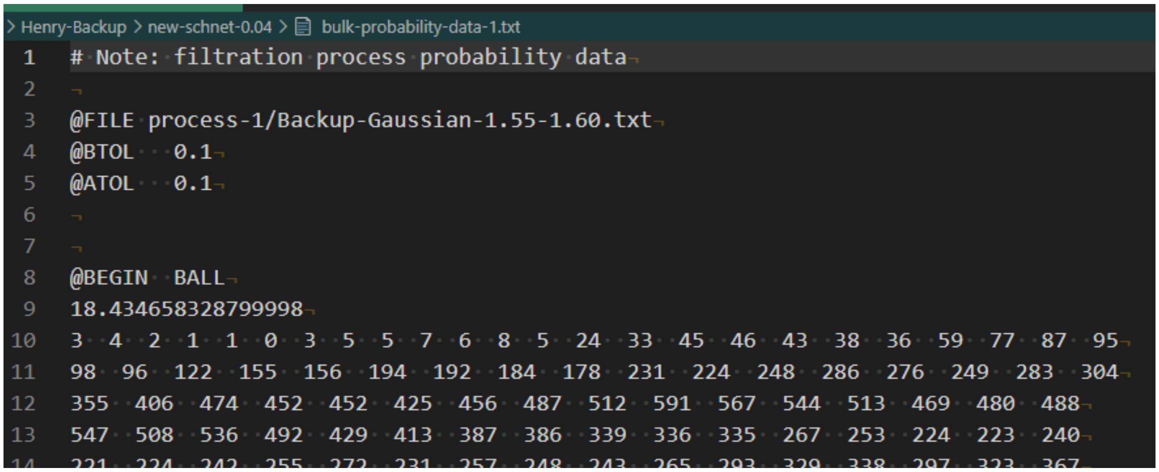
Plot

To help visualize the filtration results, a subcommand plot is integrated.

There are two types of inputs can be specified.

Plot on Probability Data Files

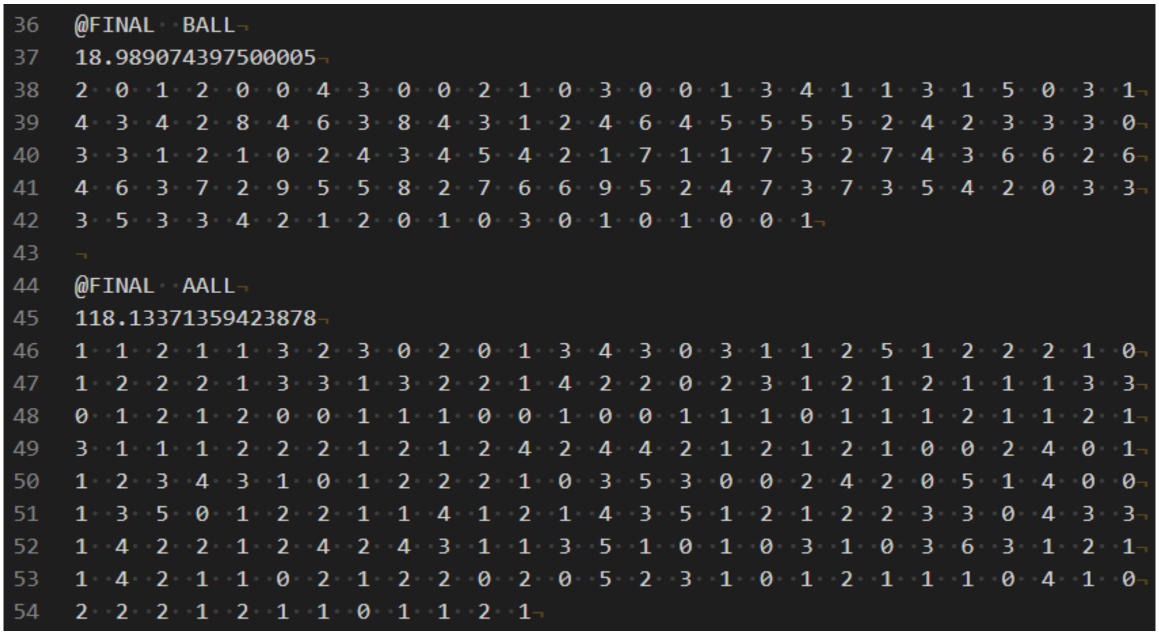
After the filtration is done, a new text file named as `bulk-probability-data-num.txt` will be generated. This file contains inner probability data for plots.


Henry-Backup > new-schnet-0.04 > bulk-probability-data-1.txt

```
# Note: filtration process probability data

@FILE process-1/Backup-Gaussian-1.55-1.60.txt
@BTOL   0.1
@ATOL   0.1


@BEGIN  BALL
18.434658328799998
3  4  2  1  1  0  3  5  5  7  6  8  5  24  33  45  46  43  38  36  59  77  87  95
98  96  122  155  156  194  192  184  178  231  224  248  286  276  249  283  304
355  406  474  452  452  425  456  487  512  591  567  544  513  469  480  488
547  508  536  492  429  413  387  386  339  336  335  267  253  224  223  240
221  224  242  255  272  231  257  248  243  265  293  329  338  297  323  367
```

Figure 13: Illustration of Probability Data File

Each integer value represents number of molecules inside bins. So the goal should be to make them be single digits (i.e., less than 10) after the favorable settings of filtration.



```
@FINAL  BALL
18.989074397500005
2  0  1  2  0  0  4  3  0  0  2  1  0  3  0  0  1  3  4  1  1  3  1  5  0  3  1
4  3  4  2  8  4  6  3  8  4  3  1  2  4  6  4  5  5  5  5  2  4  2  3  3  3  0
3  3  1  2  1  0  2  4  3  4  5  4  2  1  7  1  1  7  5  2  7  4  3  6  6  2  6
4  6  3  7  2  9  5  5  8  2  7  6  6  9  5  2  4  7  3  7  3  5  4  2  0  3  3
3  5  3  3  4  2  1  2  0  1  0  3  0  1  0  1  0  0  1

@FINAL  AALL
118.13371359423878
1  1  2  1  1  3  2  3  0  2  0  1  3  4  3  0  3  1  1  2  5  1  2  2  2  1  0
1  2  2  2  1  3  3  1  3  2  2  1  4  2  2  0  2  3  1  2  1  2  1  1  1  3  3
0  1  2  1  2  0  0  1  1  1  0  0  1  0  0  1  1  1  0  1  1  1  2  1  1  2  1
3  1  1  1  2  2  2  1  2  1  2  4  2  4  4  2  1  2  1  2  1  0  0  2  4  0  1
1  2  3  4  3  1  0  1  2  2  2  1  0  3  5  3  0  0  2  4  2  0  5  1  4  0  0
1  3  5  0  1  2  2  1  1  4  1  2  1  4  3  5  1  2  1  2  2  3  3  0  4  3  3
1  4  2  2  1  2  4  2  4  3  1  1  3  5  1  0  1  0  3  1  0  3  6  3  1  2  1
1  4  2  1  1  0  2  1  2  2  0  2  0  5  2  3  1  0  1  2  1  1  1  0  4  1  0
2  2  2  1  2  1  1  0  1  1  2  1
```

Figure 14: Probability Data after Filtration

The inputs of those data files can be called by `-bf` or `--probdatafilelist`.

Plot on Backup Files

Subcommand plot shares the options with top-level filtration command.

It has two different ways to make those selections.

1) Option `-nl` or `--nmlist` is used to select number of molecules. For example, if we want to compare 5 samples of input conformers, say `10000`, `20000`, `30000`, `40000`, `50000`, declaring them, the plot will be accordingly generated. You can visualize any number of samples in any number of molecules(no bigger than your inputs) in a way before and after filtration.

Due to the selection is in randomly, to reproduce, please specify `--seed`.

This option is used to generate non-equal length of samples, and it is in the highest priority.

2) On the contrary, the "equal" length of samples can be generated in the combined options.

For example, if we simplifying the dataset into 1D axis;

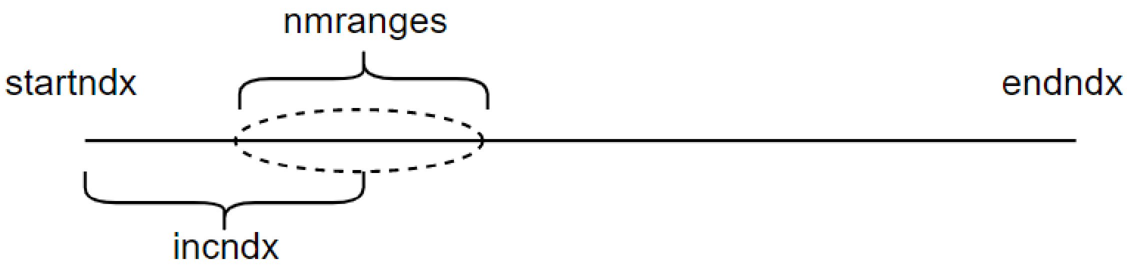


Figure 15: Parameters in Plot Subcommand

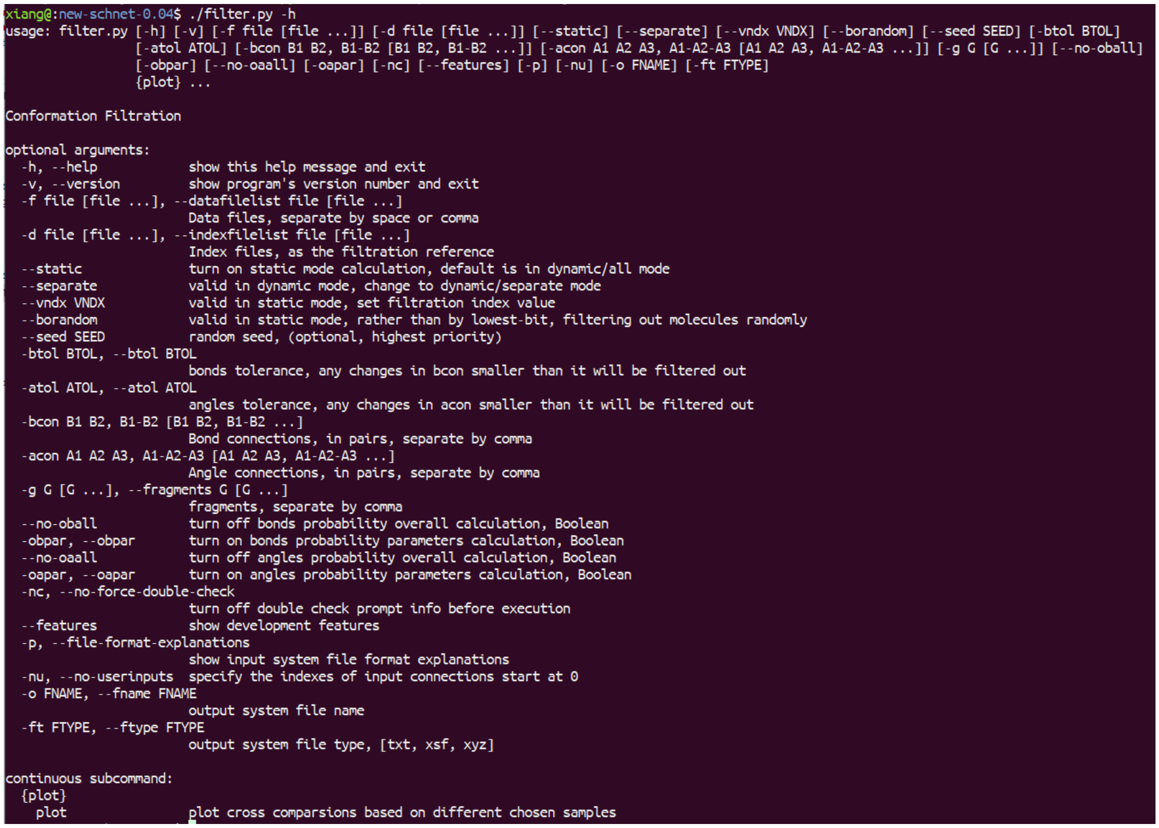
By combining those options, we can choose certain range from `--startndx` and `--endndx` in the given increment `--incndx` to generate samples. The length of generated samples will be equal, unless a random range `--nmranges` is specified, still it can be reproduced by `--seed`.

Examples

Help Usages

1) Overall usage, by executing any one of those three;

```
[python3] ./filter.py -h

[python3] ./filter.py

[python3] ./filter.py --help
```

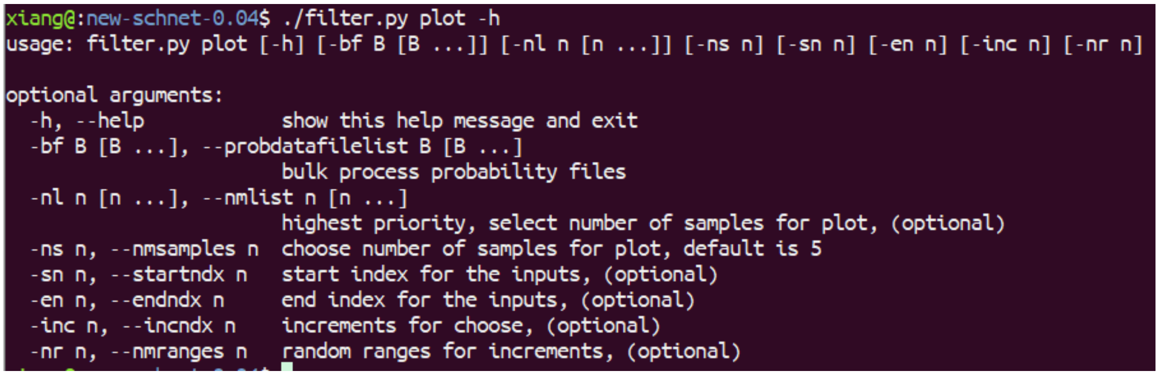


```
xiang@:new-schnet-0.04$ ./filter.py -h
usage: filter.py [-h] [-v] [-f file [file ...]] [-d file [file ...]] [--static] [--separate] [--vndx VNDX] [--borandom] [--seed SEED] [-btol BTOL]
                 [-atol ATOL] [-bcon B1 B2, B1-B2 [B1 B2, B1-B2 ...]] [-acon A1 A2 A3, A1-A2-A3 [A1 A2 A3, A1-A2-A3 ...]] [-g G [G ...]] [--no-oball]
                 [-obpar] [--no-oaall] [-oapar] [-nc] [--features] [-p] [-nu] [-o FNAME] [-ft FTYPE]
                 {plot} ...

Conformation Filtration

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -f file [file ...], --datafilelist file [file ...]
                        Data files, separate by space or comma
  -d file [file ...], --indexfilelist file [file ...]
                        Index files, as the filtration reference
  --static              turn on static mode calculation, default is in dynamic/all mode
  --separate            valid in dynamic mode, change to dynamic/separate mode
  --vndx VNDX           valid in static mode, set filtration index value
  --borandom            valid in static mode, rather than by lowest-bit, filtering out molecules randomly
  --seed SEED           random seed, (optional, highest priority)
  -btol BTOL, --btol BTOL
                        bonds tolerance, any changes in bcon smaller than it will be filtered out
  -atol ATOL, --atol ATOL
                        angles tolerance, any changes in acon smaller than it will be filtered out
  -bcon B1 B2, B1-B2 [B1 B2, B1-B2 ...]
                        Bond connections, in pairs, separate by comma
  -acon A1 A2 A3, A1-A2-A3 [A1 A2 A3, A1-A2-A3 ...]
                        Angle connections, in pairs, separate by comma
  -g G [G ...], --fragments G [G ...]
                        fragments, separate by comma
  --no-oball            turn off bonds probability overall calculation, Boolean
  -obpar, --obpar       turn on bonds probability parameters calculation, Boolean
  --no-oaall            turn off angles probability overall calculation, Boolean
  -oapar, --oapar       turn on angles probability parameters calculation, Boolean
  -nc, --no-force-double-check
                        turn off double check prompt info before execution
  --features            show development features
  -p, --file-format-explanations
                        show input system file format explanations
  -nu, --no-userinputs  specify the indexes of input connections start at 0
  -o FNAME, --fname FNAME
                        output system file name
  -ft FTYPE, --ftype FTYPE
                        output system file type, [txt, xsf, xyz]

continuous subcommand:
  {plot}
    plot                plot cross comparsions based on different chosen samples
```

Figure 16: Overall Help Usage of Filtration Script

2) Subcommand plot usage,

```
[python3] ./filter.py plot -h

[python3] ./filter.py plot --help
```



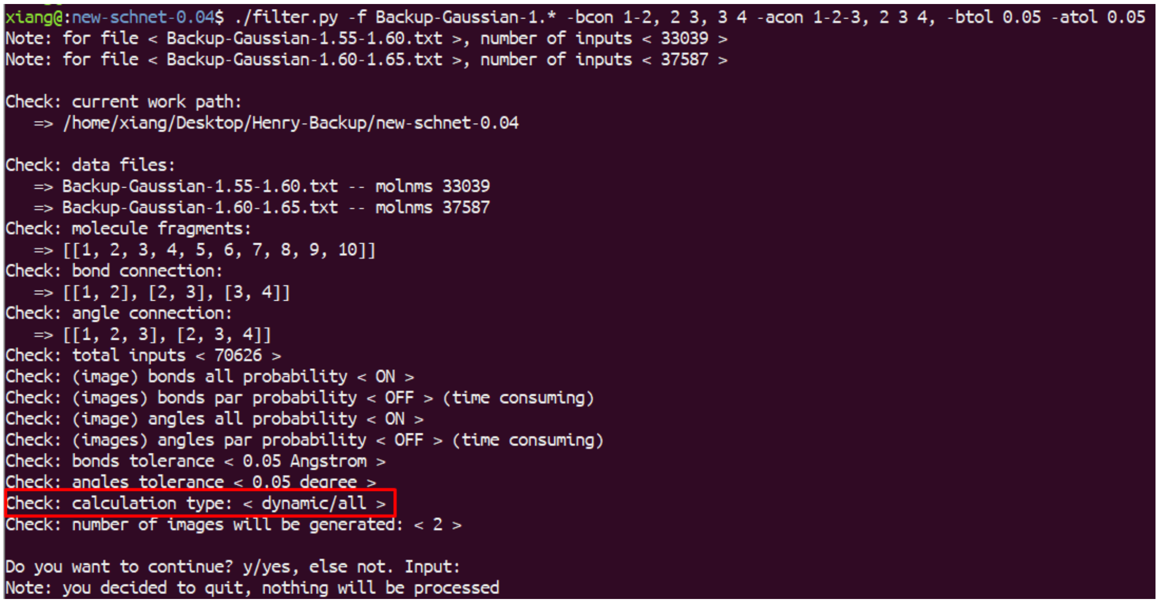
```
xiang@:new-schnet-0.04$ ./filter.py plot -h
usage: filter.py plot [-h] [-bf B [B ...]] [-nl n [n ...]] [-ns n] [-sn n] [-en n] [-inc n] [-nr n]

optional arguments:
  -h, --help            show this help message and exit
  -bf B [B ...], --probdatafilelist B [B ...]
                        bulk process probability files
  -nl n [n ...], --nmlist n [n ...]
                        highest priority, select number of samples for plot, (optional)
  -ns n, --nmsamples n  choose number of samples for plot, default is 5
  -sn n, --startndx n   start index for the inputs, (optional)
  -en n, --endndx n     end index for the inputs, (optional)
  -inc n, --incndx n    increments for choose, (optional)
  -nr n, --nmranges n   random ranges for increments, (optional)
```

Figure 17: Subcommand-plot Help Usage of Filtration Script

Filtration in mode: dynamic/all

Note: connections are separated by comma, thus white space and hyphen can be used.

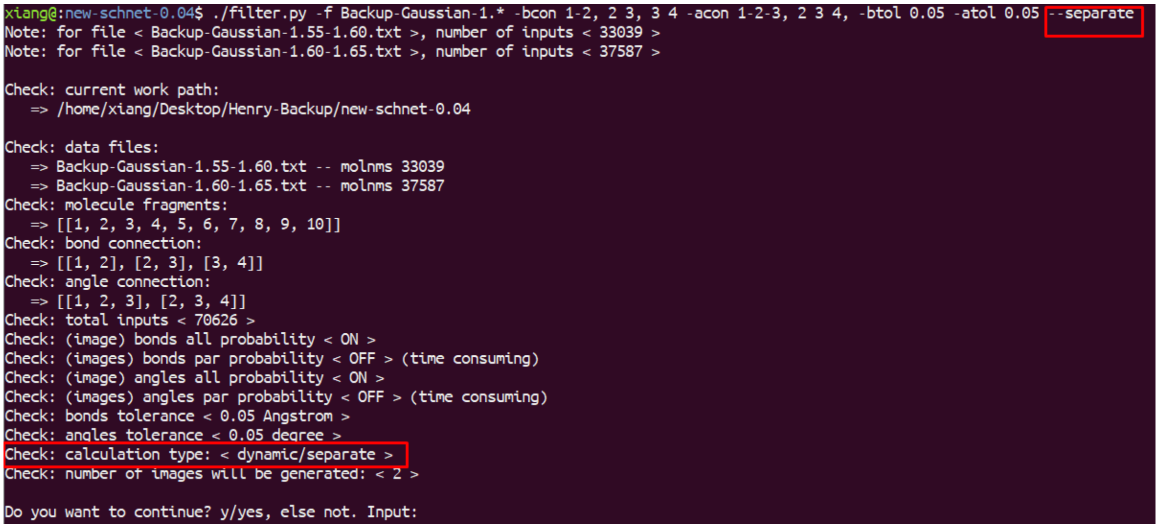
The default calculation type is dynamic/all, so we only need to input the options that we want;



```
xiang@:new-schnet-0.04$ ./filter.py -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >

Check: current work path:
   => /home/xiang/Desktop/Henry-Backup/new-schnet-0.04

Check: data files:
   => Backup-Gaussian-1.55-1.60.txt -- molnms 33039
   => Backup-Gaussian-1.60-1.65.txt -- molnms 37587
Check: molecule fragments:
   => [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]
Check: bond connection:
   => [[1, 2], [2, 3], [3, 4]]
Check: angle connection:
   => [[1, 2, 3], [2, 3, 4]]
Check: total inputs < 70626 >
Check: (image) bonds all probability < ON >
Check: (images) bonds par probability < OFF > (time consuming)
Check: (image) angles all probability < ON >
Check: (images) angles par probability < OFF > (time consuming)
Check: bonds tolerance < 0.05 Angstrom >
Check: angles tolerance < 0.05 degree >
Check: calculation type: < dynamic/all >
Check: number of images will be generated: < 2 >

Do you want to continue? y/yes, else not. Input:
Note: you decided to quit, nothing will be processed
```

Figure 18: Example of Calculation Type dynamic/all

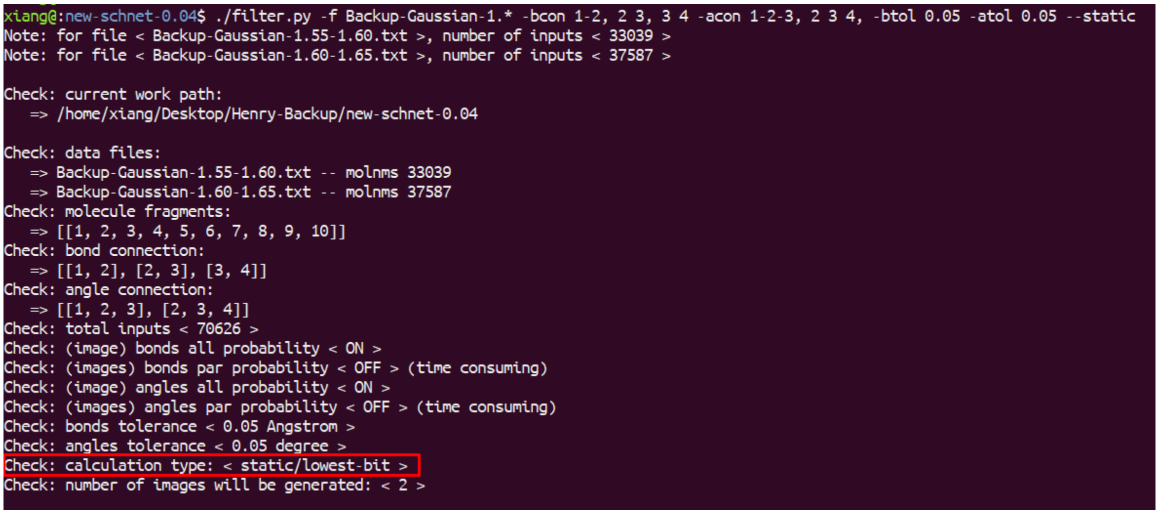
Filtration in mode: dynamic/separate



```
xiang@:new-schnet-0.04$ ./filter.py -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05 --separate
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >

Check: current work path:
   => /home/xiang/Desktop/Henry-Backup/new-schnet-0.04

Check: data files:
   => Backup-Gaussian-1.55-1.60.txt -- molnms 33039
   => Backup-Gaussian-1.60-1.65.txt -- molnms 37587
Check: molecule fragments:
   => [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]
Check: bond connection:
   => [[1, 2], [2, 3], [3, 4]]
Check: angle connection:
   => [[1, 2, 3], [2, 3, 4]]
Check: total inputs < 70626 >
Check: (image) bonds all probability < ON >
Check: (images) bonds par probability < OFF > (time consuming)
Check: (image) angles all probability < ON >
Check: (images) angles par probability < OFF > (time consuming)
Check: bonds tolerance < 0.05 Angstrom >
Check: angles tolerance < 0.05 degree >
Check: calculation type: < dynamic/separate >
Check: number of images will be generated: < 2 >

Do you want to continue? y/yes, else not. Input:
```

Figure 19: Example of Calculation Type dynamic/separate

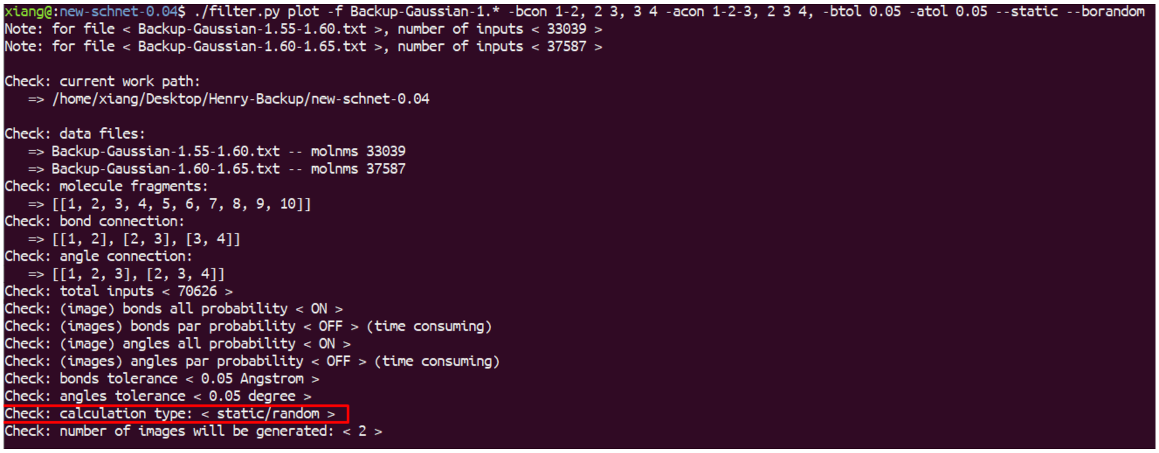
Filtration in mode: static/lowest-bit



```
xiang@:new-schnet-0.04$ ./filter.py -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05 --static
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >

Check: current work path:
   => /home/xiang/Desktop/Henry-Backup/new-schnet-0.04

Check: data files:
   => Backup-Gaussian-1.55-1.60.txt -- molnms 33039
   => Backup-Gaussian-1.60-1.65.txt -- molnms 37587
Check: molecule fragments:
   => [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]
Check: bond connection:
   => [[1, 2], [2, 3], [3, 4]]
Check: angle connection:
   => [[1, 2, 3], [2, 3, 4]]
Check: total inputs < 70626 >
Check: (image) bonds all probability < ON >
Check: (images) bonds par probability < OFF > (time consuming)
Check: (image) angles all probability < ON >
Check: (images) angles par probability < OFF > (time consuming)
Check: bonds tolerance < 0.05 Angstrom >
Check: angles tolerance < 0.05 degree >
Check: calculation type: < static/lowest-bit >
Check: number of images will be generated: < 2 >
```

Figure 20: Example of Calculation Type static/lowest-bit

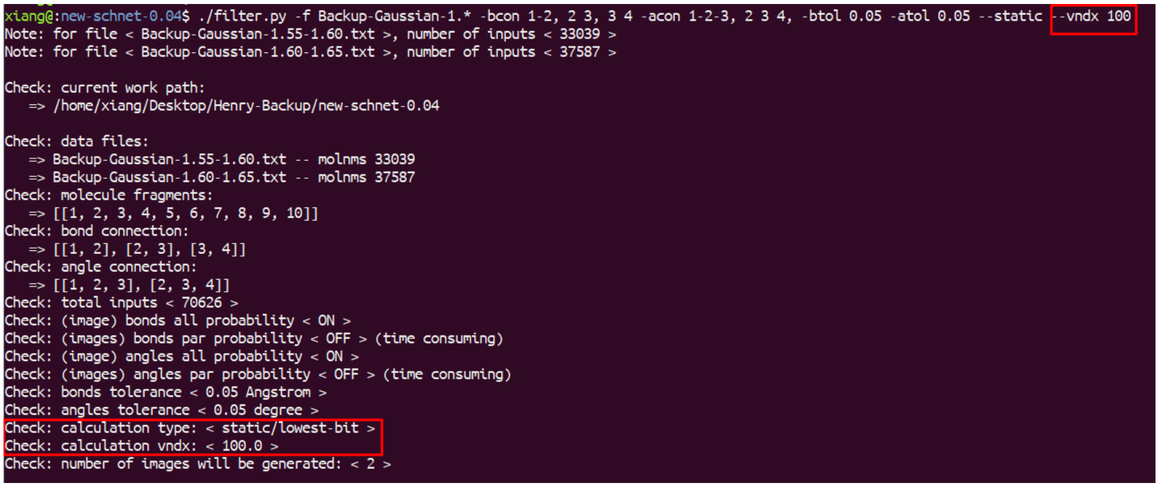
Filtration in mode: static/random



```
xiang@:new-schnet-0.04$ ./filter.py plot -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05 --static --borandom
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >

Check: current work path:
   => /home/xiang/Desktop/Henry-Backup/new-schnet-0.04

Check: data files:
   => Backup-Gaussian-1.55-1.60.txt -- molnms 33039
   => Backup-Gaussian-1.60-1.65.txt -- molnms 37587
Check: molecule fragments:
   => [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]
Check: bond connection:
   => [[1, 2], [2, 3], [3, 4]]
Check: angle connection:
   => [[1, 2, 3], [2, 3, 4]]
Check: total inputs < 70626 >
Check: (image) bonds all probability < ON >
Check: (images) bonds par probability < OFF > (time consuming)
Check: (image) angles all probability < ON >
Check: (images) angles par probability < OFF > (time consuming)
Check: bonds tolerance < 0.05 Angstrom >
Check: angles tolerance < 0.05 degree >
Check: calculation type: < static/random >
Check: number of images will be generated: < 2 >
```

Figure 21: Example of Calculation Type static/random

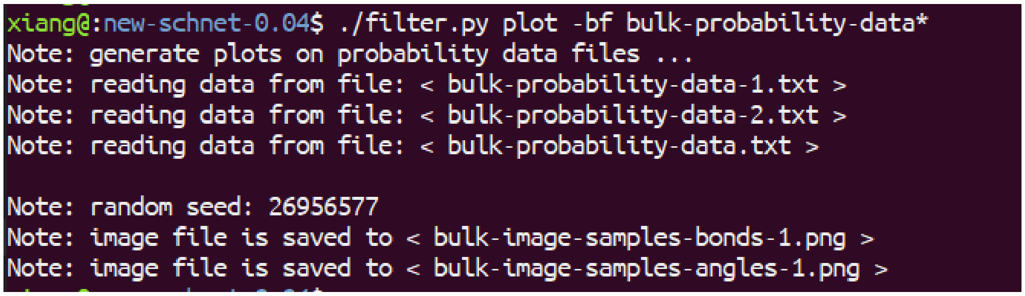
Filtration in mode: static with anchoring value



```
xiang@:new-schnet-0.04$ ./filter.py -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05 --static --vndx 100
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >

Check: current work path:
   => /home/xiang/Desktop/Henry-Backup/new-schnet-0.04

Check: data files:
   => Backup-Gaussian-1.55-1.60.txt -- molnms 33039
   => Backup-Gaussian-1.60-1.65.txt -- molnms 37587
Check: molecule fragments:
   => [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]
Check: bond connection:
   => [[1, 2], [2, 3], [3, 4]]
Check: angle connection:
   => [[1, 2, 3], [2, 3, 4]]
Check: total inputs < 70626 >
Check: (image) bonds all probability < ON >
Check: (images) bonds par probability < OFF > (time consuming)
Check: (image) angles all probability < ON >
Check: (images) angles par probability < OFF > (time consuming)
Check: bonds tolerance < 0.05 Angstrom >
Check: angles tolerance < 0.05 degree >
Check: calculation type: < static/lowest-bit >
Check: calculation vndx: < 100.0 >
Check: number of images will be generated: < 2 >
```

Figure 22: Example of Calculation Type static/lowest-bit with Anchoring Value

Plot on Probability Data Files

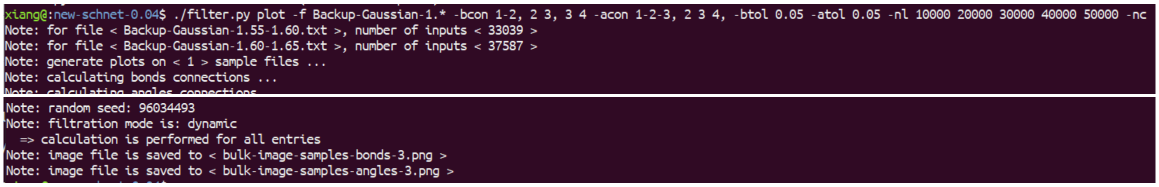


```
xiang@:new-schnet-0.04$ ./filter.py plot -bf bulk-probability-data*
Note: generate plots on probability data files ...
Note: reading data from file: < bulk-probability-data-1.txt >
Note: reading data from file: < bulk-probability-data-2.txt >
Note: reading data from file: < bulk-probability-data.txt >

Note: random seed: 26956577
Note: image file is saved to < bulk-image-samples-bonds-1.png >
Note: image file is saved to < bulk-image-samples-angles-1.png >
```

Figure 23: Example of Subcommand-plot on Probability Data Files

Plot on Backup Files

1) Specify number of molecules in each sample



```
xiang@:new-schnet-0.04$ ./filter.py plot -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05 -nl 10000 20000 30000 40000 50000 -nc
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >
Note: generate plots on < 1 > sample files ...
Note: calculating bonds connections ...
Note: calculating angles connections
Note: random seed: 96034493
Note: filtration mode is: dynamic
  => calculation is performed for all entries
Note: image file is saved to < bulk-image-samples-bonds-3.png >
Note: image file is saved to < bulk-image-samples-angles-3.png >
```

Figure 24: Example of Subcommand-plot on Backup Files with Selected Number of Samples

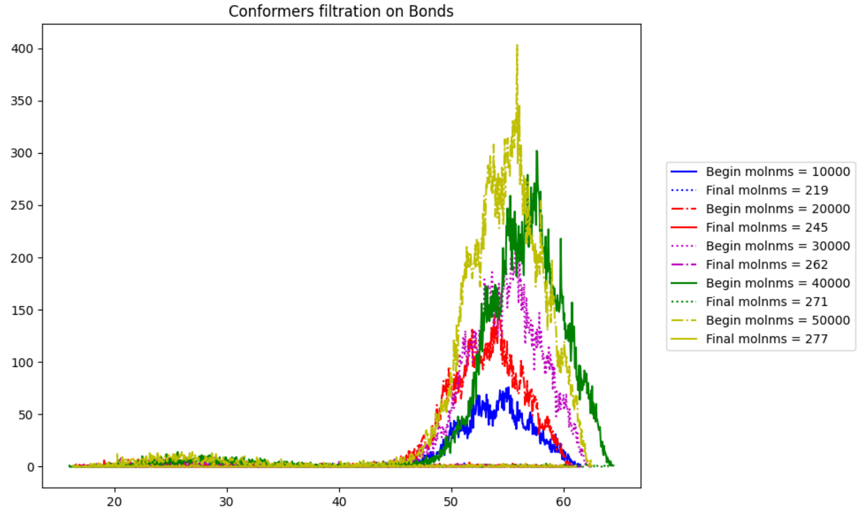


Figure 25: Example of Bonds Plot in Different Samples

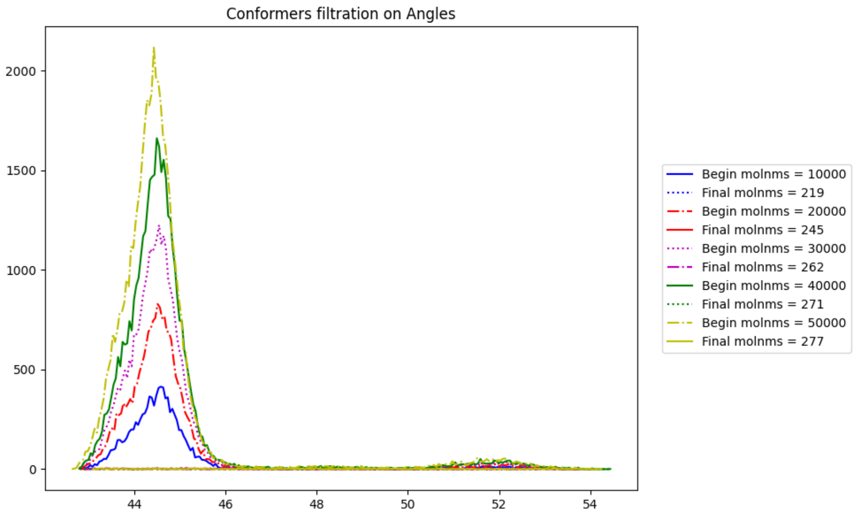
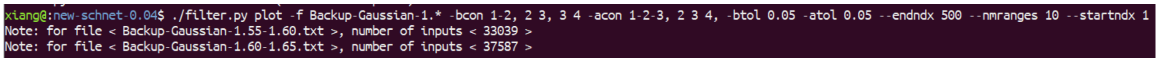


Figure 26: Example of Angles Plot in Different Samples

Note: due to fewer connections were defined, the number of molecules after filtration are very smaller than original inputs. It is just an example though.

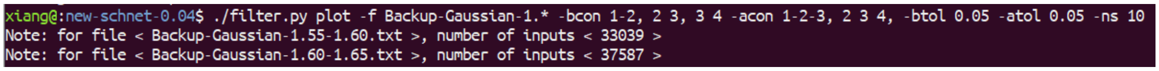
2) Plot on equal consecutive number of samples



```
xiang@:new-schnet-0.04$ ./filter.py plot -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05 --endndx 500 --nmranges 10 --startndx 1
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >
```

Figure 27: Example of Subcommand-plot on Equal Consecutive Number of Samples

3) Plot on equal split number of samples



```
xiang@:new-schnet-0.04$ ./filter.py plot -f Backup-Gaussian-1.* -bcon 1-2, 2 3, 3 4 -acon 1-2-3, 2 3 4, -btol 0.05 -atol 0.05 -ns 10
Note: for file < Backup-Gaussian-1.55-1.60.txt >, number of inputs < 33039 >
Note: for file < Backup-Gaussian-1.60-1.65.txt >, number of inputs < 37587 >
```

Figure 28: Example of Subcommand-plot on Equal Split Number of Samples

Please use it with help of Figure 15.

Code Execution Efficiency

Consider ~1.8 million 10-atomic molecules, number of bonds connections is 8, number of angles connections is 7, in calculation type `dynamic/all`, time cost is less than 40 minutes.