

NUPACK 2.1.1 User Guide

(Nucleic Acid Package)

California Institute of Technology

October 7, 2009

1 Overview

NUPACK is a growing software suite for the analysis and design of nucleic acids. The package currently enables thermodynamic analysis at the level of nucleic acid secondary structure for multiple interacting strands (without pseudoknots) or a single strand (with or without a class of pseudoknots). Much of this software may be conveniently run through the NUPACK web application at <http://www.nupack.org>.

Please direct any questions, concerns, comments, or bug reports to support@nupack.org.

2 Directories

After compiling, the root directory `nupack` contains a Makefile and the following sub-directories:

`src`

Source code.

`bin`

Executables.

`lib`

Static libraries defining functions for linking at compile time.

`parameters`

Parameter files for RNA and DNA free energy models.

`doc`

Documentation, including this user guide, reference papers, and example input and output files.

It is convenient to set the environment variable `NUPACKHOME`, specifying an absolute path to the root directory `nupack` (e.g., `NUPACKHOME=/usr/local/nupack` or `NUPACKHOME=/home/username/nupack`).

3 Compilation

Core routines are written in C. To compile the code, type `make` in the root `nupack` directory. This will create the executables described in this user guide.

4 References

The following references provide conceptual background information and details on the algorithm implementations:

1. R.M. Dirks, J.S. Bois, J.M. Schaeffer, E. Winfree, and N.A. Pierce. Thermodynamic analysis of interacting nucleic acid strands, *SIAM Rev*, **49**:65–88, 2007.
\$NUPACKHOME/doc/references/sirev07.pdf
2. R.M. Dirks and N.A. Pierce. An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots, *J Comput Chem*, **25**:1295-1304, 2004.
\$NUPACKHOME/doc/references/jcc04.pdf
3. R.M. Dirks and N.A. Pierce. A partition function algorithm for nucleic acid secondary structure including pseudoknots, *J Comput Chem*, **24**:1664-1677, 2003.
\$NUPACKHOME/doc/references/jcc03.pdf
Supplementary information: \$NUPACKHOME/docs/references/jcc03_supp.pdf

To facilitate precise description of certain quantities without including lengthy explanations in this user guide, mathematical notation and technical terms (shown in italics the first time they appear) are drawn directly from reference [1].

5 Examples

While using this user guide, it may be useful to refer to the examples in the `doc/examples` directory. It contains many sample input and output files. The `README` file in that directory gives detailed information about its contents.

6 NUPACK conventions

Following are formatting standards for all input and output in NUPACK:

- All executables except `concentrations` and `distributions` take input from an input file `prefix.in`, where `prefix` is a command line argument. If `prefix` is not specified or `prefix.in` is absent or improperly formatted, the user is prompted for input on the screen.

- All sequences are listed 5' to 3'. The bases in an *ordered complex* are indexed starting with 1 at the 5'-most base of the first strand and ending at the 3'-most base of the last strand. For example, if an ordered complex has three strands of length 15, 20, and 13, respectively, the fifth base of the third strand has index 40.
- Valid bases are A, C, G, T, and U. For RNA calculations, T is automatically converted to U, and vice versa for DNA calculations.
- *Secondary structures* may be specified in dot/parentheses notation (e.g, ((. . .)) specifies that bases 1 and 2 are paired to bases 7 and 6, respectively, while bases 3, 4, and 5 are unpaired). Different strands are separated by a + sign (no whitespace). Four types of “parentheses” are accepted: (), [], { }, and <>. Within a specified structure, each type of parentheses must satisfy a nesting property but different types need not be nested, allowing specification of *pseudoknotted* structures (though highly nested pseudoknots may not be specifiable with only four types of parentheses).
- Secondary structures may also be specified in *pair list* format, where each line consists of two whitespace-separated integers [i j], $i < j$, specifying that base i is paired to base j . Any secondary structure, including highly-nested pseudoknots, may be specified in this way.
- Comment lines begin with a % symbol.
- In input files, comment lines may be interspersed with input data. However, there may be no empty lines in the input files.

The following physical considerations are universal throughout NUPACK:

- Except where noted, all energy units are kcal/mol and all concentration units are molar.
- The zero free energy reference state for all calculations is a system where all relevant strands are present with no base pairs.
- The base pairs considered in the calculations include Watson-Crick ($A \cdot U/T$ and $G \cdot C$) and wobble ($G \cdot U/T$) pairs.
- Except where noted, all calculations and reported values appropriately take into account *distinguishability corrections* and *strand association penalties*.
- All executables involving *partition function* calculations consider an ensemble $\Omega(\pi)$ containing all *connected* unpseudoknotted secondary structures for a given ordered complex. If `-pseudo` (see below) is selected, $\Omega(\pi)$ also includes the class of pseudoknots specified in Reference [3].

The following option flags are recognized by multiple NUPACK executables:

`-material parameters`

The parameter files defining the nucleic acid material (all assuming 1 M NaCl) are specified via the argument `parameters` which represents either a filename prefix or a shorthand identifier for an included parameter set. If the filename does not contain a relative or absolute path, then the program will look for the files first in the current directory, and then in the directory `$NUPACKHOME/parameters`. Available filename prefixes and the corresponding shorthand identifiers currently include:

- RNA_fold2.3 (default; shorthand: rna)
Mfold v2.3 parameter files *.dG and *.dH for RNA allowing calculations at different temperatures; includes pseudoknot parameters from Reference [3].
- DNA_fold2.3 (shorthand: dna)
Mfold v2.3 parameter files *.dG and *.dH for DNA allowing calculations at different temperatures; there are no pseudoknot parameters.
- RNA_fold3.0 (shorthand: rna37)
Mfold v3.0 parameter file *.dG for RNA for calculations at 37 °C; includes pseudoknot parameters from Reference [3].

DNA/RNA hybrids are not allowed.

-dangles treatment

The way in which dangle energies are incorporated is specified by treatment, which may have the following values:

none: No dangle energies are incorporated.

some: (default) A dangle energy is incorporated for each unpaired base flanking a duplex (a base flanking two duplexes contributes only the minimum of the two possible dangle energies).

all: A dangle energy is incorporated for each base flanking a duplex regardless of whether it is paired.

-T temperature

Temperature specified in °C (default is 37).

-multi

Specify a calculation involving complexes of multiple interacting strands.

-pseudo

Include the class of physically relevant pseudoknots defined in Reference [3] in Ω . This option is currently only available for single-stranded RNA calculations. An error message is returned if `-pseudo` is specified in combination with either `-multi` or `-material dna`.

7 Executables

7.1 Calculate the partition function

Command: pfunc [-T temperature] [-multi] [-pseudo] [-material parameters] [-dangles treatment] [prefix]

Description: Computes the partition function, Q , for an ordered complex over the set $\Omega(\pi)$.

Input:

For single-stranded calculations, the input file contains the strand sequence specified on a single line. If `-multi` is specified, the input file must contain the following entries on separate lines:

- The number of *distinct* strand species, $|\Psi^0|$.

- The sequences for each distinct strand species, each on a separate line.
- L integers from the range 1 to $|\Psi^0|$ representing the distinct circular permutation $\pi \in \Pi$ of the L strands in the ordered complex.

Note that strand species defined on different lines are treated as distinct even if they have the same sequence.

Example 1

Partition function for a single RNA strand at 37°C including pseudoknots.

Input file contents:

```
GGGCUGUUUUUCUCGCUGACUUUCAGCCCCAAACAAAAAUGUCAGCA
```

Command: `pfunc -pseudo`

```
$NUPACKHOME/doc/examples/jcc04_telomerase/jcc04_telomerase
```

Example 2

Partition function for an ordered complex of four DNA strands at 23°C, two of which are *indistinguishable*.

Input file contents:

```
3
AGTCTAGGATTCGGCGTGGGTAA
TTAACCCACGCCGAATCCTAGACTCAAAGTAGTCTAGGATTCGGCGTG
AGTCTAGGATTCGGCGTGGGTAAACACGCCGAATCCTAGACTACTTTG
1 2 2 3
```

Command: `pfunc -T 23 -multi -material dna`

```
$NUPACKHOME/doc/examples/pnas04_hcr/pnas04_hcr_basic
```

Output:

Following header comments, the free energy of the ordered complex (given by $\Delta G = -kT \log Q$) is written to the screen. The value of the partition function is written immediately below.

7.2 Calculate base-pairing observables

Command: `pairs [-T temperature] [-multi] [-pseudo] [-material parameters] [-dangles treatment] [-cutoff cutoffvalue] [prefix]`

Description: Computes *pair probabilities* $p(i_n \cdot j_m; \pi)$ for the ordered complex corresponding to the specified circular permutation $\pi \in \Pi$. When `-multi` is selected, also computes the *expected number of base pairs* $E(i_{\{A\}} \cdot j_{\{B\}}; \pi)$.

Additional option:

`-cutoff cutoffvalue`

Only probabilities and expected values at or above cutoffvalue (default is 0.001) are saved in the output file(s).

Input: Same as for the executable `pfunc`.

Output: The output is written to the files:

- `prefix.ppairs`

Contains the probability of each type of base pair in the ordered complex. The relevant quantities are $p(i_n \cdot j_m; \pi)$, the probability that base i of strand n is paired to base j of strand m in the ordered complex corresponding to distinct circular permutation π . All strands in the ordered complex are considered to be distinct; there are no distinguishability corrections. For example, the two strands labeled 2 in Example 2 are considered distinct. One might think of them as strand $2a$ and $2b$, and a given base of strand $2a$ may have different pair probabilities than the corresponding one in strand $2b$. The total number of bases in the complex is $N = \sum_{l=1}^L N_l$, so indexing bases from 1 to N , the pair probabilities can be stored in a symmetric $N \times N$ matrix. Augmentation by an $N + 1$ st column containing the probability that each base is unpaired causes the rows to sum to unity.

By default, the file is formatted as follows. Following header comments, the first entry is the integer N . The remaining entries come in triplets of the form $[i \ j \ p]$, where $1 \leq i \leq N$ and $1 \leq j \leq N + 1$ are base numbers and p is the probability of the corresponding pair. Values corresponding to $j = N + 1$ represent the probability that base i is unpaired. If `-pseudo` is selected, each row is augmented by two additional columns. The first is the probability that bases i and j form a nested pair and the second is the probability that bases i and j form a non-nested pair. In the case of $j = N + 1$, these additional columns store the probability that bases i and j do not form a nested pair and the probability that they do not form a non-nested pair, respectively.

- `prefix.epairs`

Generated when `-multi` is selected. Similar to `prefix.ppairs` except strands of the same species are considered to be indistinguishable. The relevant quantities are $E(i_{\{A\}} \cdot j_{\{B\}}; \pi)$, the expected number of base i of strand species A that are paired to base j of strand species B in the ordered complex corresponding to distinct circular permutation π . The number of distinct bases in the complex is $N_{\text{distinct}} \equiv \sum_{k \in \Psi^0} N_k$, representing the total number of bases in all $|\Psi^0|$ strand species. Numbering the distinct bases from 1 to N_{distinct} , the distinct base pairs may be represented as a symmetric $N_{\text{distinct}} \times N_{\text{distinct}}$ matrix; by augmenting the matrix with an extra column that contains the expected number of base i of strand species A that are unpaired, each row sums to the number of base i of strand species A in the complex. Note that this numbering system is used even if some sequences listed in the input file are absent from the specified ordered complex.

The file is formatted as follows. Following header comments, the first entry is the integer N_{distinct} , and the remaining entries come in triplets of the form $[i \ j \ E]$, analogously to the `.ppairs` file, except E is the expected number of the corresponding pair. Information is stored only for bases included in the specified ordered complex.

7.3 Find minimum free energy (MFE) secondary structure(s)

Command: `mfe [-T temperature] [-multi] [-pseudo] [-material parameters] [-dangles treatment] [-degenerate] [prefix]`

Description: Compute and store the minimum free energy and MFE secondary structure(s) in $\Omega(\pi)$. If the `-degenerate` flag is selected, all secondary structures that share the same minimum free energy are

stored; otherwise only one MFE structure is stored.

Input: Same format as for the executable `pfunc`.

Output: Output is written to the file `prefix.mfe`. After header comments, each entry describes one of the possibly many degenerate MFE structures. The entries are separated by comment lines (repeated % signs). The first line in each entry is the number of bases in the ordered complex. The second line is the minimum free energy. The third line is the dot/parentheses depiction of the MFE structure. Subsequent lines contain the MFE structure in pair list notation.

7.4 Find all secondary structures within a specified free energy gap of the MFE

Command: `subopt [-T temperature] [-multi] [-pseudo] [-material parameters] [-dangles treatment] [prefix]`

Description: Similar to `mfe` except that all secondary structures in $\Omega(\pi)$ with free energies within the specified (non-negative) free energy gap of the MFE are calculated and stored. This can be very slow and the output very large if the specified gap is too large. The output is sorted by increasing free energy.

Input: Same format as for the executable `pfunc`, plus one additional row containing the energy gap.

Output: Output is written to the file `prefix.subopt` with the same format as for the executable `mfe`.

7.5 Count the number of secondary structures in the ensemble

Command: `count [-multi] [-pseudo] [prefix]`

Description: Similar to `pfunc` but sets all energy parameters to zero, thereby giving a count of the number of secondary structures in $\Omega(\pi)$.

Input: Same format as for the executable `pfunc`.

Output: The number of secondary structures in Ω is written to the screen, preceded by header comments.

7.6 Calculate the free energy of a secondary structure

Command: `energy [-T temperature] [-pseudo] [-multi] [-material parameters] [-dangles treatment] [prefix]`

Description: Calculate the free energy of a given sequence and secondary structure.

Input: Same format as for executable `pfunc`, plus one additional row specifying the secondary structure in dot/parentheses notation. Alternatively, the structure may be represented in pair list notation.

Output: The free energy of the structure is written to the screen, preceded by header comments.

7.7 Calculate the equilibrium probability of a secondary structure

Command: `prob [-T temperature] [-pseudo] [-multi] [-material parameters]
[-dangles treatment] [prefix]`

Description: Calculates the equilibrium probability of a given secondary structure.

Input: Same format as for the executable `energy`.

Output: The equilibrium probability of the given structure is written to the screen, preceded by header comments.

7.8 Calculate the identities and partition functions of all strand complexes up to a specified size

Command: `complexes [-T temperature] [-material parameters] [-ordered]
[-pairs] [-mfe] [-degenerate] [-dangles treatment] [-timeonly] [-quiet]
prefix`

Description: First calculates the identities of all distinct circular permutations $\pi \in \Pi$ of strands for all possible (unpseudoknotted) complexes up to a user-defined size L_{\max} and then calculates their respective partition functions $Q(\pi)$. The partition function for a complex, Q , is obtained by summing over the partition functions of its constituent ordered complexes, $Q(\pi)$ for all $\pi \in \Pi$. Significant additional functionality can be specified via command line flags. The output of complexes can be used as the input to the executables `concentrations` and `distributions`.

Additional options:

`-ordered`

Also store properties for ordered complexes, each corresponding to one distinct circular permutation $\pi \in \Pi$, in addition to the summed properties for complexes.

`-pairs`

Calculate base-pairing observables as for the `pairs` executable.

`-cutoff cutoffvalue`

Only probabilities and expected values at or above cutoffvalue (default is 0.001) are saved in the output file(s) generated when the `-pairs` flag is selected.

`-mfe`

Calculate all minimum free energy structures for each ordered complex as for the `mfe` executable.

Must be used in conjunction with the `-ordered` flag. The `-degenerate` flag is only applicable in conjunction with the `-mfe` flag.

`-timeonly`

After generating all distinct circular permutations, estimate the time it would take to compute all of the partition functions. The partition function calculations are not performed, the time estimate is written to the screen, and no output files are generated.

`-quiet`

Suppress output to the screen.

Input: The input file must contain the following entries on separate lines:

- The number of distinct strand species ($|\Psi^0|$).
- Sequence for each distinct strand species (each on a separate line).
- Maximum complex size (L_{\max}).

In addition to considering all complexes up to a specified maximum number of strands L_{\max} , the optional file `prefix.list` can be used to manually specify ordered complexes with more than L_{\max} strands. The file contains lines of the form:

- $C \ L_1 \ L_2 \ \dots \ L_{|\Psi^0|}$
The $|\Psi^0|$ integers represent the number of copies (possibly zero) of each strand species in the complex of $L = \sum_{k=1}^{|\Psi^0|} L_k > L_{\max}$ strands.
- Subsequent lines containing distinct circular permutations of L integers from the range 1 to $|\Psi^0|$ define the ordered complexes of interest for the most recently defined complex.

Example 3

For a system containing three DNA strand species at 23°C, calculate partition functions, pair probabilities, and MFE structures for all ordered complexes of up to four strands, and for additional larger complexes specified in a `.list` file.

Input file contents:

```
3
AGTCTAGGATTCGGCGTGGGTAA
TTAACCCACGCCGAATCCTAGACTCAAAGTAGTCTAGGATTCGGCGTG
AGTCTAGGATTCGGCGTGGGTAAACACGCCGAATCCTAGACTACTTTG
4
```

List file contents:

```
C 1 2 2
1 2 2 3 3
1 2 3 2 3
C 0 3 2
2 3 2 3 2
C 1 3 2
1 2 2 2 3 3
```

Command: `complexes -T 23 -material dna -ordered -pairs -mfe
$NUPACKHOME/doc/examples/pnas04_hcr/pnas04_hcr`

Output: Unless the `-quiet` flag is selected, `complexes` reports progress to the screen. By default there is one output file:

- `prefix.cx`
Contains the composition and free energy of each complex. The first column is an integer complex identifier, and the next $|\Psi^0| + 1$ columns are $L_1 L_2 \dots L_{|\Psi^0|} \Delta G$.

Depending on the command line options, the following output files may also be written:

- `prefix.ocx`
Generated if `-ordered` is selected. Contains the composition and free energy of each ordered complex. The first and second columns are integer complex and ordered complex identifiers, respectively, and the remaining $|\Psi^0| + 1$ columns are $L_1 L_2 \dots L_{|\Psi^0|} \Delta G$ for the ordered complex.
- `prefix.ocx-key`
Generated if `-ordered` is selected. Contains the distinct circular permutation of strands for each ordered complex. The first and second columns are integer complex and ordered complex identifiers, respectively, and the remaining L columns are integers from the range 1 to $|\Psi^0|$. Note that the value of L may be different for each complex.
- `prefix.cx-pairs`
Generated if `-pairs` is selected. Contains the base-pairing expectation values for each type of distinct base pair in each complex. The relevant quantities are $E(i_{\{A\}} \cdot j_{\{B\}})$, the expected number of base i of strand species A that are paired to base j of strand species B in the complex. The file contains a list of entries each with the same format as the `.epairs` file generated by the executable

pairs. Each entry is separated by comment lines (repeated % symbols). Additionally, each entry begins with a comment line containing the complex identifier id, expressed as “% complexid”.

- prefix.ocx-epairs
Generated if `-pairs` and `-ordered` are selected. Similar to `.cx-epairs` but for ordered complexes. The relevant quantities are $E(i_{\{A\}} \cdot j_{\{B\}}; \pi)$, the expected number of base i of strand species A that are paired to base j of strand species B in the ordered complex corresponding to distinct circular permutation π . The entries are separated by comment lines (repeated % symbols), and each entry begins with a comment line containing the complex identifier id and order identifier iorder, expressed as “% complexid-orderiorder”.
- prefix.ocx-ppairs
Generated if `-pairs` and `-ordered` are selected. Similar to `.ocx-epairs` except that all strands in the ordered complex are assumed to be distinct. The data in each entry are the same as those in the `.ppairs` file produced by the executable `pairs`.
- prefix.ocx-mfe
Generated if `-mfe` and `-ordered` are selected. Contains the minimum free energy and MFE structure(s) for each ordered complex. Each entry is formatted the same as the output for the `mfe` executable. The entries are separated by comment lines (repeated % symbols), and each entry begins with a comment line containing the complex identifier id and order identifier iorder, expressed as “% complexid-orderiorder”. If the `-degenerate` flag is selected, the degenerate MFE structures for a given entry are separated by a comment line of repeated % symbols.

7.9 Calculate the equilibrium concentration of each complex in a dilute solution

Command: `concentrations [-ordered] [-pairs] [-sort method] [-quiet] prefix`

Description: Given user-defined concentrations for each strand species, calculates the equilibrium concentration of each complex species or base pair in a large dilute solution, typical of experimental conditions in a test tube. Partition function information is read in from output files generated with the executable `complexes`.

Additional options:

`-ordered`

Performs the calculation on ordered complexes rather than complexes. The input is read from the prefix.ocx (output from the executable `complexes`).

`-pairs`

Compute base-pairing information for the entire solution using results from prefix.cx-epairs or, if `-ordered` is selected, prefix.ocx-epairs, as output by the executable `complexes`.

`-cutoff cutoffvalue`

Only ensemble pair fractions at or above cutoffvalue (default is 0.001) are saved in the output file prefix.fpairs generated when the `-pairs` flag is selected. Note that cutoffvalue should not be less than that used with `complexes` to generate the input files.

`-sort method`

The argument `method` is one of the following integers:

- 0: Output is listed in the same order as in the input file.
- 1: (default) Output is sorted by the concentration of each complex (default) or ordered complex (if `-ordered` is selected).
- 2: Output is sorted first by the concentration of each complex and then, if `-ordered` is selected, by the concentration of each constituent ordered complex.
- 3: Output is sorted first by the integer complex identifier and then, if `-ordered` is selected, by the ordered complex identifier.
- 4: Output is sorted first by the number of strands in each complex, then by the integers $L_1 L_2 \dots L_{|\Psi^0|}$ defining the number of each strand type in a given complex (with L_1 having the highest precedence, followed by L_2 , and so on), and finally, if `-ordered` is selected, by the integer ordered complex identifier.

`-quiet`

Suppress output to the screen.

Input: By default, input is read from the file `prefix.cx`. Alternatively, specifying `-ordered` causes input to be read from `prefix.ocx`. These files are formatted as for the output of the executable `complexes`. The temperature at which the calculation is done is read from a line in the comments of the `.cx` or `.ocx` input file that reads “% T = `temperature`”, where `temperature` is the temperature in °C. This line is automatically included in all output files of the executable `complexes`.

The input file `prefix.con` specifies the total molar concentration of each of $|\Psi^0|$ strand species on a separate line. The concentration may be in scientific notation (e.g., $1e-6$ for a strand species at μM concentration).

Output: Unless `-quiet` is selected, the following information is written to the screen:

- The error in conservation of mass for each strand species in molar.
- The free energy of the entire solution in kcal/L.
- The wall clock time for the calculation.

The output is written to the files:

- `prefix.eq`
The content is the same as the input file (except resorted, depending on the `-sort` option) with an extra column containing the concentration of the species in molar inserted after the free energy column.
- `prefix.fpairs`
Generated if `-pairs` is selected. Reports the fraction of each distinct base that is paired to each of the other distinct bases in solution. The relevant quantity is $f_A(i_A \cdot j_B)$, the expected fraction of strands of species A for which base i is paired to base j of strand species B . The number of distinct bases in the dilute solution is $N_{\text{distinct}} \equiv \sum_{k=1}^{|\Psi^0|} N_k$, representing the total number of bases in all $|\Psi^0|$

strand species. Numbering the distinct bases from 1 to N_{distinct} , the quantity $f_A(i_A \cdot j_B)$ may be stored as an (asymmetric) $N_{\text{distinct}} \times N_{\text{distinct}}$ matrix; by augmenting the matrix with an extra column that contains the expected fraction of base i of strand species A that are unpaired, each row sums to unity.

The file is formatted as follows. Following header comments, the first entry is the integer N_{distinct} . The remaining entries come in triplets of the form $[i \ j \ f]$, where $1 \leq i \leq N_{\text{distinct}}$ and $1 \leq j \leq N_{\text{distinct}} + 1$ are base numbers and f is the corresponding fraction from the augmented matrix.

7.10 Calculate the equilibrium expected value and distribution for each complex population for a few strands in a dilute solution

Command: `distributions [-ordered] [-maxstates big] [-writestates] [-sort method] [-quiet] prefix`

Description: The executable `distributions` calculates the *partition function* Q_{box} for a *box* containing a small number of strands, given user-defined *populations* for each strand species. This is used to calculate the *expected value* and *probability distribution* of the population of each species of complex (or ordered complex). Partition function information is read from output files generated with the executable `complexes`.

Additional options:

`-ordered`

Performs the calculation on ordered complexes rather than complexes. The input is read from the `prefix.ocx` (output from the executable `complexes`).

`-maxstates big`

The maximum number of states of the box to be enumerated ($|\Lambda|$, default is $1e7$). A segmentation fault will occur if the stack size on your machine is exceeded.

`-writestates`

Write a (typically large) output file describing properties for all population states of the system.

`-sort method`

The argument `method` is one of the following integers:

- 1: (default) Output is sorted by the expected value of the population of each complex or ordered complex (if `-ordered` is selected).
- 2: Output is sorted first by the expected value of the population of each complex and then, if `-ordered` is selected, by the expected value of the population of each constituent ordered complex.
- 3: Output is sorted first by the integer complex identifier and then, if `-ordered` is selected, by the ordered complex identifier.
- 4: Output is sorted first by the number of strands in each complex, then by the integers $L_1 \ L_2 \ \dots \ L_{|\Psi^0|}$ defining the number of each strand type in a given complex (with L_1 having the highest precedence, followed by L_2 , and so on), and finally, if `-ordered` is selected, by the integer ordered complex identifier.

`-quiet`
Suppress output to the screen.

Input: Same as for the executable `concentrations`, except the file `prefix.con` file is replaced by `prefix.count`, which contains the integer population of each of $|\Psi^0|$ strand species on a separate line. The last line of the file contains the volume of the box in liters. This may be entered in scientific notation (e.g., $1.4e-18$).

Output: Unless the `-quiet` flag is selected, the following information is written to the screen:

- The number of states of the box.
- The free energy of the entire box in units of kT and in units of kcal.
- The wall clock time for the calculation.

The output is written to the files:

- `prefix.dist`
The content is the same as the input file (with rows sorted according to `-sort`) with extra columns after the free energy column. The first extra column (for complex or ordered complex j) is the expected value of the population $\langle m_j \rangle$. Subsequent columns are $[p_j(0) \ p_j(1) \ \dots \ p_j(\max(m^0))]$. These represent the probability that complex (or ordered complex) j has population $0, 1, \dots, \max(m^0)$, at equilibrium.
- `prefix.states`
Generated when `-writestates` is selected. Each row corresponds to a population vector, m , for the box. The first column is the probability that the population vector occurs at equilibrium. By default, the remaining entries come in pairs: an integer complex identifier and a nonzero population. If `-ordered` is selected, the remaining entries come in triples: integer complex and ordered complex identifiers and then a nonzero population. This pattern continues for all complexes (or ordered complexes) with non-zero populations.