Description of powerHDX algorithm

February 2021

Detailed description of the simulation algorithm.
The code is available in the repository.

1 Elements of the code

The simulation code is in `simulate_theoretical_spectra.R` file. It uses internal functions:

- `get_approx_isotopic_distribution (isotopic_distribution.R)`,
- `get_exchange_rates (exchange_rates.R)`,
- `get_exchange_probabilities (simulate_HD_exchange.R)`,
- `get_HD_matrices_using_markov/get_HD_matrices (simulate_HD_exchange.R)`,
- `get_iso_probs_deut (simulate_HD_exchange.R)`.

2 Parameters needed to run the simulation

The simulation is run for a given peptide. The parameters are:

- `sequence` - sequence of the peptide, single string. Necessary to run the simulation. There is no validation, except for checking if the length is greater than 0 in get_exchage_rate function.
- `charge` - charge of the peptide. default value: NULL. If not provided, the random value from range 2:6 is assigned. There is no possibility for simulating more than one charge value as it is done in the experiment.
- `times` - time points of the measurement. default value: c(60, 600) [seconds]
- `protection_factor` - protection factor - vector of values or one value for all amides. default value: 1 (indicates that the exchange rate is equal to the intristic exchange rate)
- `pH` - pH. default value: 7.5
- temperature - temperature. default value: 15 [celcius]
- `n_molecules` - number of peptide molecules. default value: 100
- `time_step_const` - time step constant. default value: 1. *Value that indicates the length of the time step of the simulation. The bigger the time step, the fewer time points are simulated.*
- `if_corr` - ph correction indicator. default value: 0, the value of pH is equal to pD. If there is correction, the pD = pH + 0.4. (Conelly et al 1993)
- `min_probability` - minimal accepted probability. default value: 10E-4, threshold for accepting the values of the isotopic probabilities (intensity).
- `use_markov` - usage of Markov chains indicator. default value: TRUE, as it fastens the calculation

3 How the simulation works

Notation:
n - length of a sequence
Pf - protection factor

3.1 Prepare essential parameters.

Initially parameter `sequence` is split into a character vector. From now, `sequence` will indicate the vector of amino acids. If a single value of `protection_factor` is provided, the vector of that value of the length of the `sequence` is created. If no value of `charge` is provided, it is assigned randomly from the range from 2 to 6. There is no validation of the parameters.

3.2 Calculate the isotopic distribution of an undeuterated peptide that is required to get an empirical distribution.

Approximate isotopic distribution is calculated via `get_approx_isotopic_distribution`. The function takes two arguments: `sequence` and `min_probability` (default to 0.001).

Additional file `sysdata.RDA` contains the maximal possible occurrence of the isotopes C_{13}, N_{15}, O_{18}, S_{34} in the respective amino acids, and their masses. Based on that, the maximal possible number of molecules of the isotopes in the sequence is calculated.

Peptide mass is the sum of the masses of amino acids and H_2O mass - as it includes the N terminal group (H) and C terminal group (OH).

Next, the distributions of mentioned isotopes are calculated under the assumption that the occurrence of i^{th} considered isotope has a binomial distribution with parameters n_i (maximal possible occurrence in the sequence) and p_i (natural richness - possibility of occurrence in the universe). For the oxygen molecules we have to take into account that oxygen occurs in diatomic molecule. Calculation of the sulfur distribution takes into account its rare occurance.

The final isotopic distribution is computed as a convolution of obtained distributions with the probabilities greater than `min_probability`. It is a vector of probabilities of possible monoisotopic masses.

The number of exchangeable amides is computed as the length of the sequence, reduced by the number of prolines located on the third of further position.

The function `get_approx_isotopic_distribution` returns the mass of the peptide (`peptide_mass`), final distribution (`isotopic_distribution`) of the isotopes, number of significant probabilities minus one (`max_ND`) and number of exchangeable amino acids (`n_exchangeable`).

3.3 Calculate exchange rates that are required to obtain exchange probabilities.

The exchange rates are calculated via `get_exchange_rates` function. The adjunctive calculations take place in the internal functions from `exchange_rates` file. Namely:

- `get_F_const` - this function uses two parameters: `temp_kelvin` (initial `temperature` parameter in celcius converted into temperature in kelvin) and `gas_constant` which is 1.9858775. The value $Q_7isafactoradjustedtothemeasuredtem$

$$Q_7 = \frac{1}{\texttt{gas\_constant}} \left(\frac{1}{\texttt{temp\_kelvin}} - \frac{1}{293} \right) \tag{1}$$

The output of `get_F_const` is a list of $Ft_X = \exp\{-Q_7 \cdot Ea_X\}$ where $X \in \{A, B, W\}$ (A - Acid, B - Base, W - Water) and Ea_A, Ea_B, Ea_W are tabular values for energies of activation.

- `get_poly_const` - this function calculates constants depending on provided `mol_type` and type of exchange - `exchange` (HD or DH, default to HD). If the `mol_type` is *poly* then the constant for condition X is calculated as follows

$$K_{Xpoly} = \frac{10^{K_{X\,exp}}}{60} \tag{2}$$

where $K_{X\,exp}$ are tabular constants depending on the type of exchange. If `mol_type` is *oligo*, the constants are scaled accordingly:

$$K_{Xoligo} = K_{Xpoly} \cdot c_X \tag{3}$$

where c_X are constants 2.34, 1.35 and 1.585 for acid, base and water, respectively.

The function returns a list of Ka, Kb and Kw corresponding to the chosen `mol_type`.

- `get_pkc` - this function calculates supplementary constants for aspartic acid (Asp), glutamic acid (Glu) and histidine (His). Values for mentioned amino acids are pH and temperature dependent, in contrary to the rest amino acids with fixed values. This function needs `temp_kelvin` value, `gas_constant` and the type of exchange `exchange` (default: *HD*).

  Depending on provided `exchange` direction tabular values of exponents E_{const} are assigned. For Asp, Glu and His pKc constants are calculated based on the following formula:

$$pKc = -\log_{10}\left(10^{E_{const}} \cdot \exp\left\{\frac{-E_a}{\texttt{gas\_constant}}\left(\frac{1}{\texttt{temp\_kelvin}} - \frac{1}{278}\right)\right\}\right) \tag{4}$$

  where E_a are energies off activation for given amino acid and the chosen `exchange` direction. The function returns a list of *asp*, *glu* and *his* (pKc values corresponding to amino acids).

- `get_exchange_constants` - this function uses the parameters `pH`, `pkc_consts` calculated by `get_pkc` and `k_consts` (as described above). The output of `get_exchange_constants` is a matrix named `constants` of tabular and calculated constants (specifically for Asp, Glu, His, $C - Term$ and $NHMe$) based on the equation:

$$const = \log_{10}\left(\frac{10^{c_1 - \texttt{pH}} + 10^{c_2 - pKc}}{10^{-pKc} + 10^{-pH}}\right) \tag{5}$$

  where c_1 and c_2 are constants for protonated or deprotonated amide and pKc are constants obtained by the function `get_pkc` for respective acids.

The function `get_exchange_rates` requires the parameters `sequence`, `exchange` (default: *HD*), `pH` (default: 9), `temperature` (celcius, default: 15), `mol_type` (default: *poly*) and correction factor `if_corr` (1 or 0, default: 0). The correction of pH is taken into account for calculation of pD:

$$pD = pH + 0.4 \cdot \texttt{if\_corr}. \tag{6}$$

Next, the provided temperature is converted into K and the internal functions `get_F_const`, `get_poly_const` and `get_pkc` are evaluated.

Using obtained matrix of constants and provided `sequence` F_a and F_b are calculated for each amino acid in the sequence, with respect to the previous and next amino acid in the sequence. For the amino acids in the middle of the sequence, the following formula is used:

$$F_x = 10^{previous\_x + current\_x} \tag{7}$$

where x is either a or b, and $previous\_x$ is the acid/base factor for previous amino acid in the sequence, and $current\_x$ for the amino acid it is calculated for. If the amino acid is next to the C- or N-term, the term-effect is taken in the account.

Finally, the exchange rate k_c for the amino acid is the sum of catalysis constants for acid, base and water (Conelly et al, 1993). Namely:

$$k_c = k_{acid} + k_{base} + k_{water} \tag{8}$$

where

$$k_{acid} = F_a \cdot K_a + D \cdot F_{ta}, \tag{9}$$

$$k_{base} = F_b \cdot K_b + OD \cdot F_{tb}, \tag{10}$$

$$k_{water} = F_b \cdot K_w \cdot F_{tw}. \tag{11}$$

D and OD indicates deuterium and deuterium oxide concentration. F_a and F_b are values calculated specifically for given amino acid, as described above. K_a and K_b are values computed by `get_poly_constants` function, based on the mole type. F_{ta}, F_{tb} and F_{tw} are values computed by `get_F_const` function and described above.

The obtained exchange rates are stored in vector `kcHD` or `kcDH` according to the exchange direction. They are used to calculate the exchange probabilities thus both `kcHD` and `kcDH` are necessary as we take the possibility of back-exchange into account.

3.4 Prepare time points of possible exchanges.

To obtain time points (not to be confused with time points of the measurement) of possible exchanges, two parameters are necessary: `time_step_constant` and exchange rates computed by `get_exchange_rates` function (described in **??** section). To calculate the size of a single time step, the maximal possible exchange rate $kmax$ is needed:

$$kmax = \max\{\max\{kcDH\}, \max\{kcHD\}\} \tag{12}$$

where $kcHD$ and $kcDH$ are the vectors of exchange rates for each amino acid from the sequence in the appropriate direction. The size of a time step is a quotient of `time_step_constant` and maximal exchange rate $kmax$

$$\Delta t = \frac{\texttt{time\_step\_constant}}{kmax}. \tag{13}$$

The time points of possible exchanges are arithmetic sequences from 0 to chosen time points of measurement `times` by Δt. The length of this sequence is the number of time points of possible exchanges. The vector of the numbers of steps between provided times of measurement `times` is constructed for the Markov chain approach.

3.5 Calculate probabilities of exchanges that are required to simulate the exchange process.

Exchange probabilities are calculated via `get_exchange_probabilities` function. The essential parameters for the function are `protection_factor`, exchange rates kcHD and kcDH (described in **??** section) and the size of a single time step (Δt) (described in **??** section). The process is defined as a series of steps from the time sequence, and each step depends on the state in the previous one. Therefore, the probabilities of changing the state are conditional probabilities - probabilities of particular state in $(k+1)^{th}$ step given particular state in k^{th} step. For simplification the following notation is used:

$$\begin{aligned}
P(X_{k+1} = H \mid X_k = H) &= P\left(H \rightarrow H\right), \\
P(X_{k+1} = D \mid X_k = H) &= P\left(H \rightarrow D\right), \\
P(X_{k+1} = H \mid X_k = D) &= P\left(D \rightarrow H\right), \\
P(X_{k+1} = D \mid X_k = D) &= P\left(D \rightarrow D\right),
\end{aligned}$$

where X_k is the random variable describing a possible state of an isotope of a hydrogen (H or D) at the time k (any time point of the simulated time steps of experiment). The probabilities are described by equations **??** and **??**.

$$P\left(H \rightarrow D\right) = 1 - \exp\left(\frac{-kcHD \cdot \Delta t}{Pf}\right), \tag{14}$$

$$P\left(D \rightarrow H\right) = 1 - \exp\left(\frac{-kcDH \cdot \Delta t}{Pf}\right). \tag{15}$$

Under the assumptions mentioned before, the following equalities are satisfied:

$$P\left(H \rightarrow H\right) = 1 - P\left(H \rightarrow D\right), \tag{16}$$

$$P\left(D \rightarrow D\right) = 1 - P\left(D \rightarrow H\right). \tag{17}$$

Equations **??** and **??** describe the probabilities of staying in the same state.

The output of `get_exchange_probabilities` function is a list of two vectors: vector HD for probabilities $P\left(H \rightarrow D\right)$ and vector DH for probabilities $P\left(D \rightarrow H\right)$.

3.6 Calculate matrices of simulated exchange required for obtaining empirical distribution.

Matrices of simulated exchange can be obtained in two ways. The parameter `markov` indicates whether Markov chain (`get_HD_matrices_using_markov` function) or standard approach (`get_HD_matrices` function) is used.

Due to the evaluation time, the recommended method is the Markov chain approach. It requires `sequence`, transition probabilities (described in **??** section), the vector of numbers of steps between given time points (described in **??**), and the number of peptide molecules `n_molecules` (default: 100). Considered process is a Markov chain with transition probabilities ($P(H \to D)$, $P(D \to H)$) and states (H, D) (in the code denoted by $0's$ for hydrogens and $1's$ for deuters). Based on the notation provided in **??** section, the transition matrices for each amino acid are created as follows:

$$P = \begin{pmatrix} P\left(H \to H\right) & P\left(H \to D\right) \\ P\left(D \to H\right) & P\left(D \to D\right) \end{pmatrix} = \begin{pmatrix} 1 - P\left(H \to D\right) & P\left(H \to D\right) \\ 1 - P\left(D \to H\right) & P\left(D \to H\right) \end{pmatrix}. \tag{18}$$

The initial state of the process is $\pi(0) = \begin{pmatrix} 1 & 0 \end{pmatrix}$, as it starts with hydrogens. Since P^k is equal to the k-step transition probability matrix, the probability distribution of the Markov chain at a time k can be found as described in equation **??**.

$$\pi(k) = \pi(0)P^k = \begin{pmatrix} P(X_k = H) \\ P(X_k = D) \end{pmatrix}^T. \tag{19}$$

Using obtained probabilities, states H or D are sampled for M peptide molecules (`n_molecules)` for each of N amino acids and stored in a $M \times N$ dimensional matrix for each of the time points of the measurement `times`. The columns of the matrices respective to the first two amides or prolines, are set to zeros (implying hydrogens). The exchange of the first two amino acids is not measurable due to impact of back-exchange (Conelly et all, 1993) and proline does not have a exchangeable hydrogen. Matrices are stored in a list of matrices (`HD_matrices`) - each matrix for the respective time point of the measurement `times`.

The second part of simulation of the matrices is `get_HD_matrices` function. The parameters are `sequence`, transition probabilities (described in **??** section), time steps sequence (from 0 to the largest value of `times` by Δt) and the time points of the measurement `times`.

The provided time sequence is split into intervals between time points of the measurement `times`), to make the simulation faster in case of more than one time point of the measurement. In such a situation, the exchange to the next time point is obtained as the exchange to the previous time point and its continuation.

The simulation starts with the creation of a matrix of dimension $M \times N$, where M denotes the number of peptide molecules (`n_molecules`) and N denotes the number of amino acids. Each entry of this matrix corresponds to a single exchange site. Within the matrix, 0 denotes hydrogen and 1 denotes deuterium. The matrix is initialized with 0s or 1s, depending of the direction of the exchange.

At each time point in the time sequence:

- change 1 to 0 with probability $P(H \to D)$ in each entry of the matrix from the previous iteration,
- change 1 to 0 with probability $P(D \to H)$ in each entry of the matrix from the previous iteration.

As expained before, the matrix columns respective to the first two amides or prolines, are set to 0 (implying hydrogens). Matrices are stored as a list of matrices (`HD_matrices`) - each matrix for the respective time point of the measurement from `times`.

3.7 Calculate isotopic probabilities (intensity) and mass-to-charge ratio (m/z).

Isotopic probabilities are calculated via `get_iso_probs_deut` function. The function uses `HD_matrices` (described in **??** section), `n_exchangeable`, `max_ND`, `isotopic_distribution` and `peptide_mass` (described in **??** section), time points of the measurement `times`, peptide `charge` and `pH`.

The following calculations are performed for each time point of the measurement from `times`.

Firstly, an observed distribution of ions is computed using the internal function `get_observed_iso_dist`. It takes parameters: HD matrix (element of `HD_matrices` from **??** section), `isotopic_distribution` from **??** section and `n_exchangeable`.

The exchangeable-hydrogen distribution describing the increase of the mass is obtained from the HD matrix and the number of exchangeable hydrogens `n_exchangeable`. First, the numbers of hydrogens exchanged in each molecule are calculated as sums of rows of the HD matrix. Next, a vector of the counts is built and stored in a vector of length `n_exchangeable` plus one (for the lack of exchange). In order to obtain fractions counts are averaged.

The isotopic probabilities for the deuterated peptide are computed as the convolution of obtained distribution and the isotopic distribution for the undeuterated peptide (`isotopic_distribution`) as it is a sum of those variables (see *"Computational methods and challenges in hydrogen/deuterium exchange mass spectrometry"* by Claesen & Burzykowski p. 656 - 659, Deconvolution-Based Approach).

The function `get_observed_iso_dist` returns a vector of observed isotopic distribution (`observed_dist`).

The observed peaks for mass spectrum are observed isotopic probabilities calculated via `get_observed_iso_dist`.

The m/z values for the deuterated peptide are calculated using the `peptide_mass`, `charge` and constants - deuteron mass (1.00628) and proton mass (1.007276). Starting from the m/z value for the monoisotopic peak, the difference between the mass of deuteron and proton divided by charge of the peptide ion is added.

The output of the function `get_iso_probs_deut` is a data frame with the variables: Exposure (time point of measurement consistent with given HD matrix), Mz - m/z values, Intensity - isotopic probabilities and PH - pH.

3.8 Prepare the final result.

To the calculated results is added a minimal exchange control - for time point 0. The m/z values are obtained as ratio of the `peptide_mass` magnified by proton mass and the peptide charge. The distribution of undeuterated peptide from **??** section is the intensities vector.

The output of the function `simulate_theoretical_spectra` is a data table of the following variables:

- `Exposure` - time point of a measurement
- `Mz` - mass-to-charge ratio
- `Intensity` - isotopic probabilities larger than `min_probability` (the smaller ones are zeroes)
- `Sequence` - character sequence provided by user
- `PF` - protection factor
- `Charge` - charge
- `PH` - pH