

tidysq: efektywne przetwarzanie sekwencji biologicznych w R

Szczegółowy opis testów

22 stycznia 2021

Mateusz Bąkała, Dominik Rafacz

promotor: Michał Burdukiewicz

wersja 1.0.0

Spis treści

1	Wstęp	3
2	Testy jednostkowe	3
2.1	Testy funkcjonalności <i>Alfabetu</i>	3
2.2	Testy funkcji <code>bite()</code>	4
2.3	Testy funkcji <code>complement()</code>	5
2.4	Testy funkcji <code>export_sq()</code>	6
2.5	Testy funkcji <code>find_invalid_letters()</code>	6
2.6	Testy funkcji <code>find_motifs()</code>	7
2.7	Testy operatora <code>%has%</code>	9
2.8	Testy funkcji <code>import_sq()</code>	9
2.9	Testy funkcji <code>random_sq()</code>	10
2.10	Testy funkcji <code>remove_ambiguous()</code>	10
2.11	Testy funkcji <code>remove_na()</code>	11
2.12	Testy funkcji <code>reverse()</code>	11
2.13	Testy funkcji <code>sq()</code>	12
2.14	Testy funkcji <code>sq_type()</code>	13
2.15	Testy funkcji <code>sqapply()</code>	13
2.16	Testy funkcji <code>substitute_letters()</code>	14
2.17	Testy funkcji <code>translate()</code>	15
2.18	Testy funkcji <code>typify()</code>	15
2.19	Testy konkatencji	16

1 Wstęp

Niniejszy dokument zawiera opis testów funkcjonalnych pakietu. W ramach walidacji *Pakietu* przeprowadzone zostały liczne testy jednostkowe, zapewniające zgodność oczekiwanych typów wejściowych oraz wyjściowych. Z uwagi na wysoki poziom delegacji zadań z poziomu R do C++ większość testowanych funkcjonalności pokrywa wywołania odpowiadających sobie funkcji w obydwu interfejsach.

Pakiet przechodzi pomyślnie wszystkie poniższe przypadki testowe.

2 Testy jednostkowe

Testy funkcjonalności *Alfabetu*

Ekstrahowalność wektora *Liter* użytego przy konstrukcji

Warunki wstępne:

- Istnieje wektor *Liter* `char`.

Procedura testowa:

Konstruowany jest *Alfabet* przy użyciu wewnętrznej funkcji `sq_alphabet()` z dowolnym typem, a następnie na uzyskanym obiekcie aplikowana jest metoda `as.character()`.

Oczekiwana odpowiedź:

Zwrócony wektor napisów jest identyczny z wektorem `char`.

Równoważność indeksowania *Alfabetu* i wektora napisów

Warunki wstępne:

- Istnieje wektor *Liter* `char`.
- Istnieje *Alfabet* `alph` skonstruowany z wektora `char`.

Procedura testowa:

Ekstrahowane są elementy z obiektów `char` oraz `alph` przy użyciu tych samych indeksów różnych od wartości odpowiadającej `NA` dla danego *Alfabetu*.

Oczekiwana odpowiedź:

Zwracane wektory zawierają te same elementy (*Litery*).

Ekstrakcja *Litery* oznaczającej `NA`

Warunki wstępne:

- Istnieje *Alfabet* `alph`.

Procedura testowa:

Ekstrahowany jest element z obiektu `alph` pod indeksem równym wartości odpowiadającej `NA` dla danego *Alfabetu*.

Oczekiwana odpowiedź:

Zwracana jest *Litera* oznaczająca brak wartości.

Zachowanie atrybutów

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Atrybuty zwracanego wektora *Sekwencji* są identyczne z atrybutami `x`.

Wyjątek dla obiektów nie-`sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Ekstrahowanie dodatnimi indeksami

Warunki wstępne:

- Istnieje wektor napisów `seq`.
- Istnieje wektor *Sekwencji* `x` skonstruowany z wektora `seq`.
- Istnieje wektor dodatnich indeksów `ind`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `ind`. Równocześnie wybierany jest analogiczny podzbiór *Liter* w obrębie elementów wektora `seq`.

Oczekiwana odpowiedź:

Zachodzi równoważność przechowywanych *Sekwencji* pomiędzy uzyskanymi obiektem klasy `sq` oraz wektorem napisów.

Ekstrahowanie ujemnymi indeksami

Warunki wstępne:

- Istnieje wektor napisów `seq`.
- Istnieje wektor *Sekwencji* `x` skonstruowany z wektora `seq`.
- Istnieje wektor ujemnych indeksów `ind`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `ind`. Równocześnie wybierane są *Litery* w obrębie elementów wektora `seq` z pominięciem wskazywanych przez negatywne indeksy.

Oczekiwana odpowiedź:

Zachodzi równoważność przechowywanych *Sekwencji* pomiędzy uzyskanymi obiektem klasy `sq` oraz wektorem napisów.

Zastępowanie brakami danych

Warunki wstępne:

- Istnieje wektor *Sekwencji* *x*.
- Istnieje wektor dodatnich indeksów *ind* taki że istnieje liczba większa niż długość przynajmniej jednej *Sekwencji*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x* z parametrem *ind*.

Oczekiwana odpowiedź:

Zwracany obiekt zawiera wartości NA w miejscach odpowiadających indeksom o wartościach większych niż długość odpowiadającej *Sekwencji*. W trakcie działania operacji wyświetlane jest ostrzeżenie o pojawieniu się braków danych.

Wyjątek dla indeksów o mieszanej dodatniości

Warunki wstępne:

- Istnieje wektor *Sekwencji* *x*.
- Istnieje wektor dodatnich i ujemnych indeksów *ind*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x* z parametrem *ind*.

Oczekiwana odpowiedź:

Wyrzucany jest wyjątek o nieprawidłowej wartości indeksów.

Testy funkcji `complement()`

Zachowanie atrybutów

Warunki wstępne:

- Istnieje wektor *Sekwencji* *x*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x*.

Oczekiwana odpowiedź:

Atrybuty zwracanego wektora *Sekwencji* są identyczne z atrybutami *x*.

Wyjątek dla obiektów nie-*sq.dna/sq.rna*

Warunki wstępne:

- Istnieje obiekt *obj* o klasie innej niż *sq.dna/sq.rna*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *obj*.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Podwójne użycie funkcji

Warunki wstępne:

- Istnieje wektor *Sekwencji* *x*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x*, a następnie wywoływana jest rzeczona funkcja na wyniku poprzedniej operacji.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* jest identyczny z *x*.

Testy funkcji `export_sq()`

Wyjątek dla obiektów nie-`sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Zachowanie informacji *Sekwencji*

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje tekstowa reprezentacja klasy z innego pakietu `target`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `target`.

Oczekiwana odpowiedź:

Zwracany jest obiekt o docelowej klasie, zawierający *Sekwencje* równoważne z *Sekwencjami* wektora `x`.

Eksport pojedynczej *Sekwencji* do `Biostrings::XString`

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje napis `target` równy `"Biostrings::DNAString"`, `"Biostrings::RNAString"` albo `"Biostrings::AAString"`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `target`.

Oczekiwana odpowiedź:

Jeśli długość wektora `x` jest równa 1, wówczas zwracany jest obiekt klasy odpowiadającej tej opisanej w obiekcie `target`. W przeciwnym wypadku zwracany jest wyjątek o konieczności zawierania dokładnie jednej sekwencji.

Testy funkcji `find_invalid_letters()`

Zwrócenie listy wektorów napisów

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje *Typ Standardowy* `type`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `type`.

Oczekiwana odpowiedź:

Zwracana lista ma elementy będące wektorami napisów, gdzie każdy napis jest *Literą* pochodzącą z *Alfabetu* wektora `x`.

Wyjątek dla obiektów nie-`sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Wartość NA częścią *Alfabetu*

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`, wśród których występują takie, które zawierają braki danych.
- Istnieje *Typ Standardowy* `type`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `type`.

Oczekiwana odpowiedź:

Elementy listy nie zawierają *Litery* odpowiadającej brakowi danych.

Testy funkcji `find_motifs()`

Wyjątek dla obiektów nie-`sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Zwrócenie ramki danych

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje wektor nazw `x_names`.
- Istnieje wektor *Motywów* `motifs`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrami `x_names` i `motifs`.

Oczekiwana odpowiedź:

Zwracany obiekt jest ramką danych klasy `tibble` o następujących kolumnach: tekstowej `names`, `found` klasy `sq`, tekstowej `sought` oraz liczbowych `start` i `end`. Prototyp kolumny `found` jest identyczny z prototypem wektora `x`.

Wyjątek dla znaków specjalnych w niewłaściwym miejscu

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje wektor nazw `x_names`.
- Istnieje wektor *Motywów* `motifs`, wśród elementów którego znajdują się zdeformowane *Motywy* zawierające znaki `^` nie na początku albo `$` nie na końcu napisu.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x* z parametrami *x_names* i *motifs*.

Oczekiwana odpowiedź:

Zwracany jest wyjątek informujący o zdeformowanym *Motywie*.

Podzbiór *Motywów* w kolumnie *sought*

Warunki wstępne:

- Istnieje wektor *Sekwencji x*.
- Istnieje wektor nazw *x_names*.
- Istnieje wektor *Motywów motifs*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x* z parametrami *x_names* i *motifs*. Z wynikowej ramki danych wyciągana jest kolumna *sought*.

Oczekiwana odpowiedź:

Elementy w wyciągniętej kolumnie stanowią podzbiór wektora *motifs*.

Uzyskiwalność podsekwencji

Warunki wstępne:

- Istnieje wektor *Sekwencji x*.
- Istnieje wektor nazw *x_names*.
- Istnieje wektor *Motywów motifs*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x* z parametrami *x_names* i *motifs*. Następnie dla każdego rzędu w wynikowej ramce danych wybierana jest *Sekwencja*, której indeks w wektorze *x* odpowiada indeksowi wartości z kolumny *names* w wektorze *x_names*. Na *Sekwencji* tej aplikowana jest funkcja *bite()* z ciągiem indeksów *start:end*.

Oczekiwana odpowiedź:

Uzyskana podsekwencja jest identyczna z odpowiadającą jej *Sekwencją* w kolumnie *found*.

Interpretacja *Liter Niejednoznacznych*

- Istnieje wektor *Sekwencji x* o *Typie Standardowym*.
- Istnieje wektor nazw *x_names*.
- Istnieje wektor *Motywów motifs*, wśród elementów którego znajdują się *Motywy* zawierające *Litery Niejednoznaczne*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x* z parametrami *x_names* i *motifs*. Z wynikowej ramki danych wyciągana jest kolumna *found*.

Oczekiwana odpowiedź:

Sekwencje w wyciągniętej kolumnie mają jedną z interpretacji *Liter Niejednoznacznych* w miejscu tej *Litery* w odpowiadającym *Motywie*.

Wyjątek dla obiektów nie-sq

Warunki wstępne:

- Istnieje obiekt *obj* o klasie innej niż *sq*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *obj*.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Interpretacja *Liter Niejednoznacznych*

Warunki wstępne:

- Istnieje wektor *Sekwencji x* o *Typie Standardowym*.
- Istnieje wektor *Motywów motifs*, wśród elementów którego znajdują się *Motywy* zawierające *Litery Niejednoznaczne*.

Procedura testowa:

Wywoływana jest testowana funkcja na operandach *x* oraz *motifs*.

Oczekiwana odpowiedź:

Zwracane wartości logiczne są 'prawdą' wtedy, kiedy *Sekwencje* w operandzie *x* mają jedną z interpretacji *Liter Niejednoznacznych* w miejscu tej *Litery* w odpowiadającym *Motywie*.

Sumowalność logiczna operatora

Warunki wstępne:

- Istnieje wektor *Sekwencji x*.
- Istnieje wektor *Motywów motifs* o długości co najmniej 2.

Procedura testowa:

Wywoływana jest testowana funkcja na operandach *x* oraz *motifs*. Wywoływana jest także testowana funkcja na operandach *x* oraz każdym pojedynczym *Motywie* z wektora *motifs*, a następnie uzyskane wyniki są łączone przy użyciu zwektoryzowanej koniunkcji logicznej.

Oczekiwana odpowiedź:

Uzyskane wektory logiczne są sobie równe.

Testy funkcji *import_sq()*

Zachowanie informacji *Sekwencji*

Warunki wstępne:

- Istnieje obiekt jednej ze wspieranych klas *obj*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie *x*.

Oczekiwana odpowiedź:

Zwracany jest wektor *Sekwencji* klasy *sq*, zawierający *Sekwencje* równoważne z *Sekwencjami obj*.

Testy funkcji `random_sq()`

Nieujemność długości *Sekwencji*

Warunki wstępne:

- Istnieje liczba naturalna `n`.
- Istnieje liczba naturalna `len`.
- Istnieje *Typ Standardowy* bądź *Alfabet* `alph`.
- Istnieje liczba dodatnia `sd`.

Procedura testowa:

Generator losowych długości jest lokalnie mockowany w celu zapewnienia zwrócenia co najmniej jednej ujemnej wartości, a następnie wywoływana jest testowana funkcja na obiekcie `x` z pozostałymi obiektami jako parametry.

Oczekiwana odpowiedź:

Wygenerowane *Sekwencje* mają nieujemne długości.

Testy funkcji `remove_ambiguous()`

Wyjątek dla obiektów *nie-sq_dna/sq_rna/sq_ami*

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq_dna/sq_rna/sq_ami`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Zachowanie obiektów o *Alfabecie Podstawowym*

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x` o jednym *Typów Standardowych* z *Alfabetem Podstawowym*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Zwracany jest oryginalny, niezmienny obiekt `x`.

Redukcja *Alfabetu Rozszerzonego*

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x` o jednym *Typów Standardowych* z *Alfabetem Rozszerzonym*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* o analogicznym *Typie* z *Alfabetem Podstawowym*.

Testy funkcji `remove_na()`

Zachowanie atrybutów

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Atrybuty zwracanego wektora *Sekwencji* są identyczne z atrybutami `x`.

Wyjątek dla obiektów nie-`sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Usunięcie wartości NA

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x` z *Sekwencjami* zawierającymi braki danych.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* nie zawiera *Sekwencji* z brakami danych.

Testy funkcji `reverse()`

Zachowanie atrybutów

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Atrybuty zwracanego wektora *Sekwencji* są identyczne z atrybutami `x`.

Wyjątek dla obiektów nie-`sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Podwójne użycie funkcji

Warunki wstępne:

- Istnieje wektor *Sekwencji* **x**.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie **x**, a następnie wywoływana jest rzeczona funkcja na wyniku poprzedniej operacji.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* jest identyczny z **x**.

Testy funkcji `sq()`

Zgodność długości

Warunki wstępne:

- Istnieje wektor napisów zawierających *Sekwencje* **proto**.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie **proto**.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* ma długość równą wektorowi **proto**. Jeśli nie został przekazany *Alfabet Wieloznakowy* bądź wieloznakowa *Litera* braku danych, wówczas długości *Sekwencji* odpowiadają liczbie znaków elementów wektora **proto**.

Wczytywanie *Liter* spoza *Alfabetu*

Warunki wstępne:

- Istnieje *Typ Standardowy* bądź *Alfabet* **alph**.
- Istnieje wektor napisów zawierających *Sekwencje* **proto**, takie że istnieją znaki nie występujące w *Alfabecie* opisywanym przez obiekt **alph**.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie **proto** z parametrem **alph**.

Oczekiwana odpowiedź:

Zwraca wektor *Sekwencji* z brakami wartości w miejscach, gdzie w wektorze **proto** występowały znaki spoza *Alfabetu*.

Zachłanne wczytywanie *Liter*

Warunki wstępne:

- Istnieje wektor napisów zawierających *Sekwencje* **proto**.
- Istnieje *Alfabet Wieloznakowy* **alph**, taki że istnieją *Litery* stanowiące podciąg znaków innych *Liter*.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie **proto** z parametrem **alph**.

Oczekiwana odpowiedź:

Zwraca wektor *Sekwencji* z zachłannie wczytanymi *Literami*, to znaczy dopasowując lokalnie najdłuższą możliwą *Literę* w danym miejscu *Sekwencji*.

Ignorowanie wielkości znaków

Warunki wstępne:

- Istnieje wektor napisów zawierających *Sekwencje* **proto**, takie że istnieją równocześnie małe oraz wielkie znaki, również tej samej *Litery*.

- Istnieje wektor napisów zawierających *Sekwencje uppercase*, takie że wszystkie małe znaki wektora `proto` zastąpione zostały ich wielkimi odpowiednikami.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `proto` z parametrem `ignore_case` równym `TRUE`. Analogiczne wywołanie jest wykonywane na obiekcie `uppercase` z rzeczonym parametrem oraz bez niego.

Oczekiwana odpowiedź:

Uzyskane wektory *Sekwencji* są identyczne.

Testy funkcji `sq_type()`

Wyjątek dla obiektów nie-sq

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Zwrócenie *Typu*

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Zwracany napis reprezentuje jeden z *Typów*.

Przypisanie wartości

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje *Typ* `type`.

Procedura testowa:

Wartości `sq_type` wektora `x` przypisywana jest wartość `type`.

Oczekiwana odpowiedź:

Działanie przypisania jest równoważne działaniu funkcji `typify()` z przypisaniem rezultatu do zmiennej `x`.

Testy funkcji `sqapply()`

Wyjątek dla obiektów nie-sq

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Zwrócenie listy wyników

Warunki wstępne:

- Istnieje wektor *Sekwencji* x .
- Istnieje funkcja f .

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie x z parametrem f .

Oczekiwana odpowiedź:

Zwracany obiekt jest listą, której każdy element jest jednym z możliwych typów zwracanych przez funkcję f .

Testy funkcji `substitute_letters()`

Wyjątek dla obiektów nie-`sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Konwersja *Typu* do *Niestandardowego*

Warunki wstępne:

- Istnieje wektor *Sekwencji* x .
- Istnieje nazwany wektor napisów `enc`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie x z parametrem `enc`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* ma *Typ Niestandardowy*. Liczba oraz długość *Sekwencji* pozostają niezmienione względem wektora x .

Surjekcja *Alfabetu*

Warunki wstępne:

- Istnieje wektor *Sekwencji* x .
- Istnieje nazwany wektor napisów `enc`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie x z parametrem `enc`, a następnie z rezultatu oraz obiektu x wyciągane są ich *Alfabety*.

Oczekiwana odpowiedź:

Alfabet rezultatu jest nie większy niż ten oryginalnego wektora *Sekwencji*.

Testy funkcji `translate()`

Wyjątek dla obiektów `nie-sq_dna_bsc/sq_rna_bsc`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq_dna_bsc/sq_rna_bsc`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Zwrot *Sekwencji* aminokwasów

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x` typu DNA lub RNA.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* ma *Alfabet Podstawowy* dla aminokwasów oraz odpowiadający mu *Typ*.

Testy funkcji `typify()`

Wyjątek dla obiektów `nie-sq`

Warunki wstępne:

- Istnieje obiekt `obj` o klasie innej niż `sq`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `obj`.

Oczekiwana odpowiedź:

Zwracany jest wyjątek o braku implementacji metody dla klasy przekazanego obiektu.

Docelowy prototyp *Sekwencji*

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje *Typ Standardowy* `type`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `type`.

Oczekiwana odpowiedź:

Atrybuty oraz klasa zwracanego wektora *Sekwencji* odpowiadają prototypowi wektora *Sekwencji* o docelowym *Typie*.

Wczesny zwrot dla sekwencji o docelowym *Typie*

Warunki wstępne:

- Istnieje wektor *Sekwencji* `x`.
- Istnieje *Typ Standardowy* `type` identyczny z *Typem* obiektu `x`.

Procedura testowa:

Wywoływana jest testowana funkcja na obiekcie `x` z parametrem `type`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* jest identyczny z `x`.

Łączenie takich samych *Typów Standardowych*

Warunki wstępne:

- Istnieją minimum dwa wektory *Sekwencji* o jednym z *Typów Standardowych*.

Procedura testowa:

Wszystkie wektory *Sekwencji* przekazywane są do funkcji `c()`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* jest również tego typu oraz długości równej sumie długości przekazanych wektorów.

Łączenie *Podstawowego* i *Rozszerzonego Alfabetu*

Warunki wstępne:

- Istnieją minimum dwa wektory *Sekwencji*, takie że występuje co najmniej jeden wektor o *Typie Standardowym* z *Podstawowym Alfabetem*, zaś reszta (również co najmniej jeden) — o odpowiadającym *Typie Standardowym* z *Rozszerzonym Alfabetem*.

Procedura testowa:

Wszystkie wektory *Sekwencji* przekazywane są do funkcji `c()`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* ma ten sam *Typ Standardowy* z *Alfabetem Rozszerzonym* oraz długość równą sumie długości przekazanych wektorów.

Łączenie dwóch *Typów Nieokreślonych*

Warunki wstępne:

- Istnieją minimum dwa wektory *Sekwencji* o *Typach Nieokreślonych* o nieidentycznych *Alfabetach*.

Procedura testowa:

Wszystkie wektory *Sekwencji* przekazywane są do funkcji `c()`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* ma *Typ Nieokreślony*, *Alfabet* stanowiący sumę teoriomnogościową wszystkich wejściowych *Alfabetów* oraz długość równą sumie długości przekazanych wektorów.

Łączenie *Sekwencji Nietypowych* i *Typów Standardowych*

Warunki wstępne:

- Istnieje minimum jeden wektor *Sekwencji Nietypowych*.
- Istnieje minimum jeden wektor *Sekwencji* o jednym z *Typów Standardowych*.

Procedura testowa:

Wszystkie wektory *Sekwencji* przekazywane są do funkcji `c()`.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji Nietypowych* ma *Alfabet* stanowiący sumę teoriomnogościową wszystkich wejściowych *Alfabetów* oraz długość równą sumie długości przekazanych wektorów.

Łączenie z wektorem napisów

Warunki wstępne:

- Istnieje minimum jeden wektor *Sekwencji* o dowolnym *Typie*.

- Istnieje wektor napisów interpretowalny jako wektor *Sekwencji*.

Procedura testowa:

Wszystkie wektory przekazywane są do funkcji `c()`, tak aby pierwszym wektorem był któryś z wektorów *Sekwencji*.

Oczekiwana odpowiedź:

Zwracany wektor *Sekwencji* ma *Typ* i *Alfabet* zgodny z oczekiwaniami przedstawionymi w pozostałych testach konkatencji oraz długość równą sumie długości przekazanych wektorów. Dodanie oraz odjęcie wektorów napisów nie wpływa na wynikowe *Typ* i *Alfabet*.