Wydział Matematyki i Nauk Informacyjnych
POLITECHNIKA WARSZAWSKA

tidysq: efektywne przetwarzanie sekwencji biologicznych w R

Mateusz Bąkała, Dominik Rafacz
promotor: *dr Michał Burdukiewicz*

Potrzeba

Język **R** jest jednym z podstawowych narzędzi wykorzystywanych przez bioinformatyków. Posiada wiele pakietów do analizy **sekwencji biologicznych** (aminokwasy, DNA, RNA), jednak istniejące rozwiązania te mają wiele wad: są bardzo niewydajne i nieczytelne dla użytkownika (*seqinr* [1]), mają zbyt wiele funkcjonalności dla początkujących (*ape* [2]), trudny do nadbudowywania i nieczytelny kod (*Biostrings* [3], część *Bioconductora*) oraz posiadają nieczytelne interfejsy niezgdone z metodyką **"tidy"** i brak sprawdzania błędów (wszystkie wymienione)

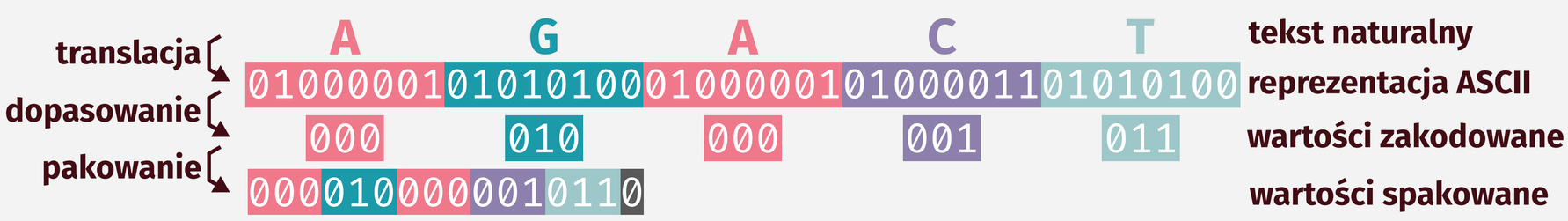
Cele

Z niedoskonałości innych pakietów wynikają cele, które przed sobą postawiliśmy:

- **Efektywność,**
- **Bezpieczeństwo i wygoda**
- **Rozszerzalność**

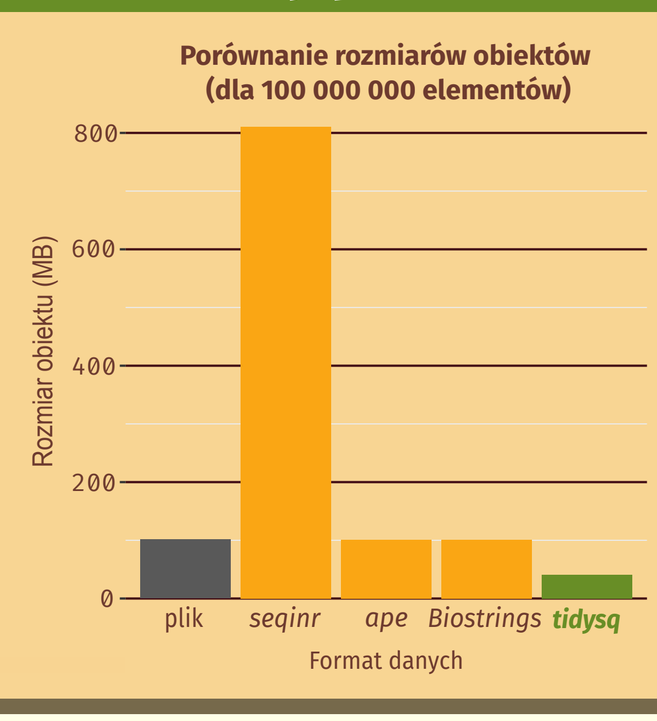
Wydajność: Pakowanie sekwencji biologicznych

Kluczowym rozwiązaniem zastosowanym w pakiecie jest **pakowanie sekwencji biologicznych**. Wykorzystujemy fakt, że zwykle sekwencje przechowywane są w postaci tekstu naturalnego, tzn. jako ciąg liter pochodzących ze ściśle określonego zbioru, który jest dużo mniejszy niż liczba dostępnych znaków. Aby ograniczyć zużycie pamięci, każdemu elementowi sekwencji (bazie nukleotydowej, reszcie aminokwasowej) przypisujemy określoną wartość liczbową, kodowaną na liczbie bitów mniejszej niż 8 (8 to liczba bitów zajmowanych przez pojedynczą literę ASCII). Następnie za pomocą operacji bitowych wykonywanych na poziomie **C++** wyrównujemy te wartości w ten sposób, aby zredukować wykorzystywaną liczbę bajtów.



W efekcie uzyskujemy znaczną redukcję zajmowanej pamięci operacyjnej, zarówno względem formatów innych pakietów, jak i względem miejsca zajmowanego przez pliki na dysku.

Rezultaty pakowania



Bezpieczeństwo i wygoda użytkowania: standardy tidy

W języku **R** coraz powszechniejsze staje się podejście **"tidy"** do danych, reprezentowane przez pakiety z rodziny *tidyverse* [4]. Podejście takie jest wygodne dla użytkownika i umożliwia analizę danych w logiczny, ustrukturyzowany, funkcjonalny sposób. Dlatego też ***tidysq*** implementuje te standardy. W szczególności dane przechowywane są jako **wektory sekwencji biologicznych** (z wykorzystaniem *vctrs* i *tibble*), które można przechowywać — tak jak inne dane — wewnątrz ramek danych. Na danych takich można wykonywać **zwektoryzowane operacje**. Ponadto użytkownik posiada kontrolę nad tym jakie elementy znajdują się w sekwencjach i jest informowany o wszystkich niestandardowych sytuacjach oraz błędach.

Rezulatatem jest kod czytelny, integrowalny z resztą *tidyverse* (w szczególności z pakietem *dplyr* [5]) oraz bezpieczeństwo w zakresie typów danych i poprawności wykonania operacji.

Przykładowy kod

funkcje pakietu ***tidysq***
funkcje pakietu ***dplyr***

```
library(tidysq)
library(dplyr)

read_fasta("example.fasta") %>%
    filter(sq %has% "LXG") %>%
    mutate(beg = bite(sq, 1:3) %>%
    group_by(beg) %>%
    summarise(n = n())
```

Rozszerzalność: integracja z C++, Rcpp, STD

Pakiet składa się z dwóch podstawowych części: **pakietu R** oraz **biblioteki C++**, zintegrowanych za pomocą pakietu *Rcpp* [6]. Kluczowe dla wydajności operacje wykonywane są na poziomie **C++** (w szczególności pakowanie sekwencji), natomiast język **R** zapewnia wygodny interfejs użytkownika i wstępne przetwarzanie. Struktura obiektów umożliwia łatwe nadbudowywanie na fundamencie ***tidysq*** — dobrze udokumentowane i czytelne wysoko- i niskopoziomowe API gwarantuje łatwe dodawanie kolejnych modułów, zarówno na poziomie **R**, jak i **C++**. Wykorzystanie metaprogramowania dodatkowo pozwala programiście pracować w wyabstrahowaniu od wewnętrznej reprezentacji danych (obecnie wektory mogą być przechowywane w formacie *Rcpp* lub jako wektory z *STD*, można też dodać kolejne reprezentacje).

Plany rozwoju

Wśród planowanych rozszerzeń pakietu znajdują się następujące funkcjonalności:

- Odczytywanie i zapisywanie innych formatów plików
- Integracja z frameworkiem do uczenia maszynowego *mlr*
- Wykonywanie kodowania aminokwasów
- Analiza przyrównań
- Wizualizacja statystyk sekwencji

1. ***seqinr***: seqinr.r-forge.r-project.org
2. ***ape***: ape-package.ird.fr
3. ***Biostrings***: bioconductor.org/packages/Biostrings
4. ***tidyverse***: www.tidyverse.org
5. ***dplyr***: dplyr.tidyverse.org
6. ***Rcpp***: rcpp.org

Specjalne podziękowania dla:
- *Leona Eyricha Jessena (Technical University of Denmark)* — za współtworzenie konceptu oraz konsultacje merytoryczne
- *Jadwigi Słowik (Politechnika Warszawska)* — za wskazówki techniczne i polecenie wykorzystanych narzędzi

github.com/BioGenies/tidysq
dominikrafacz@gmail.com
erdaradun.gaztea@gmail.com