# Accepted Manuscript


Grammar compressed sequences with rank/select support

Alberto Ordóñez, Gonzalo Navarro, Nieves R. Brisaboa

PII: S1570-8667(16)30034-X
DOI: http://dx.doi.org/10.1016/j.jda.2016.10.001
Reference: JDA 664

To appear in: *Journal of Discrete Algorithms*

Please cite this article in press as: A. Ordóñez et al., Grammar compressed sequences with rank/select support, *J. Discret. Algorithms* (2016), http://dx.doi.org/10.1016/j.jda.2016.10.001

# Grammar Compressed Sequences with Rank/Select Support [1]

Alberto Ordóñez* Gonzalo Navarro† Nieves R. Brisaboa*

* Database Laboratory, Universidade da Coruña, Spain
† Department of Computer Science, University of Chile, Chile

**Abstract**

Sequence representations supporting not only direct access to their symbols, but also rank/select operations, are a fundamental building block in many compressed data structures. Several recent applications need to represent highly repetitive sequences, and classical statistical compression proves ineffective. We introduce, instead, grammar-based representations for repetitive sequences, which use up to 6% of the space needed by statistically compressed representations, and support direct access and rank/select operations within tens of microseconds. We demonstrate the impact of our structures in text indexing applications.

*Keywords:* Grammar compression, repetitive sequences, text indexing

## 1. Introduction

Given a sequence $S[1,n]$ over an alphabet $\Sigma = [1,\sigma]$, an intensively studied problem in recent years has been how to represent $S$ space-efficiently while supporting these three operations:

- $\texttt{access}(S,i)$, which returns $S[i]$, with $1 \leq i \leq n$.
- $\texttt{rank}_b(S,i)$, which returns number of occurrences of $b \in \Sigma$ in $S[1,i]$, with $0 \leq i \leq n$.
- $\texttt{select}_b(S,i)$, which returns the position of the $i$-th occurrence of $b \in \Sigma$ in $S$, with $0 \leq i \leq \texttt{rank}_b(S,n)$ and $\texttt{select}_b(S,0) = 0$.

The data structures supporting these three operations will be called `rsa` structures (for `rank`, `select`, `access`). Their popularity owes to the wide number of applications in which they are particularly useful. For instance, we can

[1] An early partial version of this paper appeared in *Proc. SPIRE 2014* [46].