Elastic-Degenerate Strings in Bioinformatics: Motivation and Open Problems

Dominika Bohuslavová [1]

[1]Czech Technical University in Prague

June 9, 2025

Motivation

- **Explosion of biological data**: Modern sequencing technologies produce vast amounts of data.
- **From single sequence to collections**: To ask more complex questions.
- **Research directions**: Take advantages from analyzing a larger number of highly similar sequences (pangenomics, phylogenomics).

- **Population Reference Genome**: A concept representing multiple genomes in a unified data structure — enabling more realistic analysis of biological variability.

Approaches

Approach	Idea / Representation	Tools
k-mer-based	Compact colored de Bruijn Graphs k-mer hash table indices	**Bifrost**, **deBGA**, **Themisto**, **Cortex**, **Metagraph**, **Pufferfish**, **minimap2**
Linearization / Concatenation	Concatenate all sequences: seq1#seq2#... and build global FM-index, r-index	**MONI**, **SPUMONI**, **MOVI**
Variant-aware, MSA-based	Shared regions and differences (SNPs, INDELs) in variation graphs, variable texts	**vg**, **GCSA2**, **gramtools**, **GraphAligner**, **BWBBLE**

String

- **Alphabet** Σ is a finite set of symbols.
  DNA: $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$.

- A **string** X over Σ is a finite sequence of symbols from Σ:

$$X = s_1 s_2 \dots s_n \in \Sigma^n$$

  where $|X| = n$ is the **length** of the string.
- s_i is the i-th character of X; $X[i..j]$ is the substring from i to j.
- ε denotes the **empty string**, i.e., a string of length 0.
- Special symbols as separators or endmarks #, $

+ Degeneracy

- A **degenerate string** X over Σ is a finite sequence of symbols

$$X = S_1 S_2 \dots S_n \in \Sigma^n$$

  where S_i is a i-th set of symbols from Σ, i.e., $S_i \subseteq \Sigma$
- $|X| = n$ is the **length** of the degenerate string.
- $||X|| = N$ is the **size** of the degenerate string, $N = \sum_{i=0}^{n} |S_i|$
- If $|S_i| > 1$, then symbol is **degenerate**.
- **Language** $L(X)$: set of all strings represented by X

Example:

$$X = \mathsf{GT} \left\{ \begin{array}{c} \mathsf{C} \\ \mathsf{G} \end{array} \right\} \mathsf{AT} \left\{ \begin{array}{c} \mathsf{A} \\ \mathsf{C} \\ \mathsf{T} \end{array} \right\} \mathsf{T}$$

+ Elasticity

- An **Elastic-degenerate string** (EDS)

$$X = S_1 S_2 \dots S_n \in \Sigma^n$$

  where S_i is a i-th set of strings over Σ, i.e., $S_i^{i\prime}$ is a $i\prime$-th string in S_i
- **Elasticity:** Every string in $S_i, 0 \leq i \leq n$ may vary in length.
- $|X| = n$ is the **length** of the degenerate string.
- $||X|| = N$ is the **size** of the degenerate string, $N = \sum_{i=0}^{n} \sum_{i\prime}^{|S_i|} |S_i^{i\prime}|$
- **Cardinality** $c = \sum_{i=0}^{n} |S_i|$

Example:

$$X = \{\mathsf{AAC}\} \left\{ \begin{array}{c} \varepsilon \\ \mathsf{TA} \end{array} \right\} \left\{ \begin{array}{c} \mathsf{CGA} \\ \mathsf{G} \end{array} \right\} \left\{ \begin{array}{c} \mathsf{A} \\ \mathsf{GG} \\ \mathsf{T} \end{array} \right\} \left\{ \begin{array}{c} \varepsilon \\ \mathsf{AA} \end{array} \right\},$$

n = 5, N = 15, c = 10

L(X) = {AACCGAA,AACCGAAAA,AACCGAGG,. . . ,AACTAGT,AACTAGTAA}

Pros and Cons

Pros	**Cons**
Straightforward, easy-to-understand representation	No haplotype paths or variant relationships encoded by default
Can model SNPs and small indels compactly	Cannot represent complex structural variants (e.g. inversions, long inserts)
Polynomial-time algorithms for some tasks	Many problems are NP-hard
Linear construction	Compromise in degeneracy level while construction?

Problem Hardness on EDS

Köppl and Olbrich 2024	Longest Repeating Factor	$\mathcal{O}(N^2)$
	Minimal Unique Substring	NP-hard[1]
	Minimal Absent Word	NP-hard[1]
	Anti-Power Detection	NP-hard[1]
	Longest-Previous-Factor	NP-hard[1]

[1] Already hard for binary degenerate strings

Online pattern matching

Elastic-Degenerate String Matching (EDSM) Problem:

Input: EDS X of length n , string pattern P of length m.
Output: All positions in X where at least one occurrence of P ends.

Online Exact EDSM	**Time**	**Space**	**Impl.**
Iliopoulos, Kundu, and S. Pissis 2016	$\mathcal{O}(N + \alpha\gamma nm)$	$\mathcal{O}(m)$	✓
Grossi et al. 2017	$\mathcal{O}(nm^2 + N)$;$\mathcal{O}(N\lceil m/w \rceil)$	$\mathcal{O}(m)$	✓
Aoyama et al. 2018	$\mathcal{O}(nm^{1.5} + N)$	$\mathcal{O}(m)$	✗
Bernardini, Gawrychowski, et al. 2021	$\mathcal{O}(nm^{\omega-1} + N)$[1];$\mathcal{O}(nm^{1.373} + N)$	Not discussed	✗
S. P. Pissis and Retha 2018	$\mathcal{O}(N\lceil M/w \rceil)$	$\mathcal{O}(M\lceil M/w \rceil)$[2]	✓
Cisłak and Grabowski 2020	$\mathcal{O}(occ * m(N/n + \lceil r/w \rceil))$[3]	Not discussed	✓

[1] $\omega < 2.373$
[2] $M = \sum |P_i|$
[3] w: word machine size, d: avg number of variants, r: #samples

Online pattern matching

Elastic-Degenerate String Matching (EDSM) with k Mismatches/Errors:

Input: EDS X of length n and cardinality c, string pattern P of length m.
Output: All positions in X where at least one approximate occurrence (with at most k mismatches/errors) of P ends.

Online Aproximate EDSM	**Time**	**Space**	**Impl.**
Bernardini, Pisanti, et al. 2020	$\mathcal{O}(k^2mc + kN)$[1];$\mathcal{O}(kmc + kN)$[2]	$\mathcal{O}(m)$	✗
S. Pissis, Radoszewski, and Zuba 2025	$\mathcal{O}(kmc + kN)$ [1];$\mathcal{O}(k^{2/3}mc + \sqrt{k}N)$[2]	$\mathcal{O}(m)$	?
Gawrychowski et al. 2025	$\mathcal{O}(nm^{1.5} + N)$	Not discussed	✗

[1] Under Edit distance (EDSM with k Errors)
[2] Under Hamming distance (EDSM with k Mismatches)

EDS Indexes

	Description	**Impl.**
Maciuca et al. 2016	BWT FM-index over PRG with anchors	✓
Gibney 2020	$\mathcal{O}(n^{\alpha}m^{\beta}), \alpha < 0, \beta < 0$; GST $\mathcal{O}(nm^2)$	✗
Cioni, Guerrini, and Rosone 2024	EDS-BWT based on BWT for sequence collections EBWT	✓

Mapping and Alignment

	Description	**Impl.**
Mwaniki and Pisanti 2022	Online $\mathcal{O}(mN)$-time	✗
Büchler, Olbrich, and Ohlebusch 2023	Minimizer based index over PRG	✓

EDS Comparison

Elastic-Degenerate String Intersection (EDSI) problem:

- **Input:** Two ED-strings X_1 (length n_1, cardinality c_1, size N_1) and X_2 (length n_2, cardinality c_2, size N_2)
- **Output:** YES if $L(X_1) \cap L(X_2) \neq \emptyset$, NO otherwise

Reference / Method	Build Time	Lower bound	Impl.
Gabory et al. 2024	$\mathcal{O}(N_1 m_2 + N_2 m_1)$	$\mathcal{O}((N_1 m_2 + N_2 m_1)^{1-\epsilon})$[3]	✓

[3] $\epsilon > 0$ constant from matrix multiplication lower bounds

Take-Away Message & Summary

- **Elastic-Degenerate Strings (EDS)**:
  - They can compactly represent both SNPs and indels.
  - Enable adaptations of string algorithms to variation-aware data.
  - Solutions exist for some tasks (e.g., pattern matching).
  - Only a few implementations are haplotype-aware (using phased EDS).
- How can EDS be built and limited to reflect the best biological image?
- Is phasing EDS that necessary?
- What level of degeneracy is common in real data - in bacteria, eukaryota?
- Do you see other application?

Very nice review on EDS and other variable strings in Ascone et al. 2024.

Acknowledge and Thanks

Thank you for your attention

supervisor: prof. Ing. Jan Holub, Ph.D.

ROBOPROX

CZ.02.01.01/00/22_008/0004590

SGS23/205/OHK3/3T/18

References I

- Aoyama, Kotaro et al. (2018). "Faster Online Elastic Degenerate String Matching". In: *29th Annual Symposium on Combinatorial Pattern Matching (CPM 2018)*. Ed. by Gonzalo Navarro, David Sankoff, and Binhai Zhu. Vol. 105. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 9:1–9:10. ISBN: 978-3-95977-074-3. DOI: 10.4230/LIPIcs.CPM.2018.9. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CPM.2018.9.
- Bernardini, Giulia, Paweł Gawrychowski, et al. (2021). *Elastic-Degenerate String Matching via Fast Matrix Multiplication*. arXiv: 1905.02298 [cs.DS]. URL: https://arxiv.org/abs/1905.02298.
- Bernardini, Giulia, Nadia Pisanti, et al. (2020). "Approximate pattern matching on elastic-degenerate text". In: *Theoretical Computer Science* 812. In memoriam Danny Breslauer (1968-2017), pp. 109–122. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2019.08.012. URL: https://www.sciencedirect.com/science/article/pii/S0304397519305018.
- Gawrychowski, Paweł et al. (2025). *Faster ED-String Matching with k Mismatches*. arXiv: 2503.01388 [cs.DS]. URL: https://arxiv.org/abs/2503.01388.

References II

 Grossi, Roberto et al. (July 2017). *On-line pattern matching on similar texts*. DOI: 10.4230/LIPIcs.CPM.2017.07.

References I

- Cisłak, Aleksander and Szymon Grabowski (2020). *SOPanG 2: online searching over a pan-genome without false positives*. https://arxiv.org/abs/2004.03033. arXiv:2004.03033 [cs.DS], doi:10.48550/arXiv.2004.03033.
- Gabory, Esteban et al. (2024). "Pangenome comparison via ED strings". In: *Frontiers in Bioinformatics* Volume 4 - 2024. ISSN: 2673-7647. DOI: 10.3389/fbinf.2024.1397036. URL: https://www.frontiersin.org/journals/bioinformatics/articles/10.3389/fbinf.2024.1397036.
- Gibney, Daniel (2020). "An Efficient Elastic-Degenerate Text Index? Not Likely". In: *String Processing and Information Retrieval*. Ed. by Christina Boucher and Sharma V. Thankachan. Cham: Springer International Publishing, pp. 76–88. ISBN: 978-3-030-59212-7.
- Köppl, Dominik and Jannik Olbrich (2024). *Hardness Results on Characteristics for Elastic-Degenerated Strings*. arXiv: 2411.10653 [cs.DS]. URL: https://arxiv.org/abs/2411.10653.
- Pissis, Solon, Jakub Radoszewski, and Wiktor Zuba (2025). *Faster Approximate Elastic-Degenerate String Matching*.

References I

Büchler, Thomas, Jannik Olbrich, and Enno Ohlebusch (May 2023). "Efficient short read mapping to a pangenome that is represented by a graph of ED strings". In: *Bioinformatics* 39.5, btad320. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad320. eprint: https://academic.oup.com/bioinformatics/article-pdf/39/5/btad320/50497655/btad320_supplementary_data.pdf. URL: https://doi.org/10.1093/bioinformatics/btad320.

Cioni, Lapo, Veronica Guerrini, and Giovanna Rosone (2024). "The Burrows-Wheeler Transform of an Elastic-Degenerate String". In: *Proceedings of the 25th Italian Conference on Theoretical Computer Science (ICTCS 2024)*. Vol. 3811. Available under CC BY 4.0. CEUR Workshop Proceedings, pp. 1–15. URL: https://ceur-ws.org/Vol-3811/paper240.pdf.

Maciuca, Sorina et al. (2016). "A Natural Encoding of Genetic Variation in a Burrows-Wheeler Transform to Enable Mapping and Genome Inference". In: *Algorithms in Bioinformatics*. Ed. by Martin Frith and Christian Nørgaard Storm Pedersen. Cham: Springer International Publishing, pp. 222–233. ISBN: 978-3-319-43681-4.

References II


Mwaniki, Njagi Moses and Nadia Pisanti (2022). "Optimal Sequence Alignment to ED-Strings". In: *Bioinformatics Research and Applications*. Ed. by Mukul S. Bansal, Zhipeng Cai, and Serghei Mangul. Cham: Springer Nature Switzerland, pp. 204–216. ISBN: 978-3-031-23198-8.


Pissis, Solon P. and Ahmad Retha (2018). "Dictionary Matching in Elastic-Degenerate Texts with Applications in Searching VCF Files On-line". In: *17th International Symposium on Experimental Algorithms (SEA 2018)*. Ed. by Gianlorenzo D'Angelo. Vol. 103. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 16:1–16:14. ISBN: 978-3-95977-070-5. DOI: 10.4230/LIPIcs.SEA.2018.16. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.SEA.2018.16.

References I

Ascone, Rocco et al. (2024). "A Unifying Taxonomy of Pattern Matching in Degenerate Strings and Founder Graphs". In: *24th International Workshop on Algorithms in Bioinformatics (WABI 2024)*. Ed. by Solon P. Pissis and Wing-Kin Sung. Vol. 312. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 14:1–14:21. ISBN: 978-3-95977-340-9. DOI: 10.4230/LIPIcs.WABI.2024.14. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.WABI.2024.14.