Naslavljanje DNA-pomnilnikov

Lidija Stanovnik, Miha Mraz
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana
lidija.stanovnik@fri.uni-lj.si, miha.mraz@fri.uni-lj.si

Izvleček

Potrebe po cenovno učinkovitih pomnilnih medijih za dolgotrajno arhiviranje podatkov so povzročile razvoj nekonvencionalnih platform, med katere sodi tudi trajno pomnjenje podatkov v DNA. Z vsakim novim medijem pa se pojavijo tudi novi problemi. Zaradi drugačnih lastnosti bioloških sistemov se v mediju DNA pojavljajo drugačni tipi napak, kot smo jih vajeni iz digitalnih sistemov, prav tako se razlikuje tudi njihova porazdelitev in pogostost. Da bi težave ustrezno naslovili, je potrebno razviti ustrezne (nove) kode za odkrivanje in odpravljanje napak. V članku spoznamo kode brez prekritij, ki se uporabljajo za dostop do podatkov v DNA-pomnilnikih, in opišemo pristop, s katerim lahko konstruiramo velik naslovni prostor biološkega pomnilnika.

Ključne besede: DNA-pomnilniki, celoštevilska optimizacija, kodi brez prekritij

Addressing of DNA memory

Abstract

Due to the increasing demand for price efficient long-term storage media, over the last decade, unconventional platforms such as the DNA memory have been developed. Every novel medium, however, brings up new problems. Biological systems have specific characteristics that lead to errors in the DNA medium that differ from the ones that occur in the digital systems. Their frequency and distribution also vary. To address these problems accordingly, new error detecting and correcting codes should be developed. Herein, we present non-overlapping codes that are used to access the data stored in DNA memory, and describe a method to construct a large address space of a biological storage device.

Keywords: DNA memory, integer optimization, non-overlapping codes

1 UVOD

Zaradi vse večjih potreb po trajnem shranjevanju podatkov se proizvajalci konvencionalnih pomnilnih platform spopadajo s težavo zagotavljanja zadostnih pomnilnih kapacitet. Eno od možnih rešitev predstavlja razvoj novih pomnilnih platform, med katere sodi tudi trajno pomnjenje podatkov v DNA. Ker je DNA robusten medij in je posledično obstoj podatkov na njem dolgotrajen ter omogoča veliko gostoto zapisa, v teoriji do 455 EB/g [5], je še posebej primeren za pomnjenje arhivskih podatkov [7].

Biološki pomnilni sistemi se po načinu dostopa do podatkov razlikujejo od klasičnih elektronskih sistemov. Do podatkov namreč ne moremo dostopati neposredno preko vodil, ki bi povezovala pošiljatelja in prejemnika, saj so podatki shranjeni v obliki molekul DNA, ki znotraj biološkega pomnilnika niso prostorsko urejene [3]. Dostop do podatkov je omogočen z uporabo tehnik, ki iz nabora vseh molekul DNA, ki so shranjene v pomnilniku, izlušči tiste, v katerih se nahaja nek določen vnaprej izbran podniz. Slednjemu rečemo naslov. Velikost pomnilnika je torej omejena s številom različnih naslovov.

Za kreiranje množice naslovov znotraj posameznega pomnilnika uporabimo kode za odkrivanje in odpravljanje napak [1, 2, 6, 11, 17]. Ena izmed družin kodov, ki se uporabljajo v ta namen, so kodi brez prekritij in njihova posplošitev v šibko nekorelirane kode z dodatnimi zahtevami, da je v naslovu največ pet zaporednih enakih simbolov ter da je v njem delež simbolov iz nabora $\{G, C\}$ blizu 50 % [9, 15, 18, 10]. Slednji zahtevi izvirata iz lastnosti tehnologije, ki

se uporablja za dostop do podatkov, saj je pogostost napačnega dostopa pri zaporedjih, ki zahtevama ne zadoščajo, bistveno večja [12, 13].

Znano je, da velikost največjega koda brez prekritij narašča eksponentno z dolžino kodne besede [8], vendar natančna formula za njen izračun v splošnem ne obstaja. Še manj je znanega o kodih brez prekritij, ki izpolnjujejo dodatne kriterije za uporabo v DNA-pomnilnikih. Zanima nas, ali lahko vseeno (učinkovito) generiramo kode z velikim številom različnih besed.

2 DOLOČANJE NASLOVNEGA PROSTORA POMNILNIKA

Kod brez prekritij je množica besed, za katero velja, da se nobena netrivialna predpona kodne besede ne sme pojaviti kot netrivialna pripona iste ali druge kodne besede [8]. Chee in sodelavci [4] so določili velikosti največjih dvojiških kodov brez prekritij za manjše dolžine s pomočjo iskanja največje klike v grafu. Sestavili so graf, katerega vozlišča so dvojiške besede, za katere velja, da se znotraj iste besede nobena predpona ne pojavi kot pripona. Dve vozlišči so povezali, kadar predpone ene besede niso nastopale kot pripone druge besede in obratno. Čeprav se je v dvojiškem primeru pristop obnesel in bi se ga dalo razširiti tako, da vozlišča sestavljajo le kodne besede, ki imajo kvečjemu pet zaporednih enakih simbolov ter približno polovico simbolov iz nekega nabora, je v splošnem gradnja takega grafa prostorsko in časovno zahtevna. Zaradi oblike grafa lahko tudi pričakujemo, da bodo algoritmi za iskanje največje klike v grafu v praksi počasni [16]. V grafu so številne simetrije, ki bi jih bilo potrebno upoštevati, da bi bil računski problem lažje obvladljiv.

Sami smo ubrali nekoliko drugačen pristop. Problem smo prevedli na celoštevilski optimizacijski problem s polinomsko kriterijsko funkcijo in polinomskimi omejitvami (1). Vsako njegovo optimalno rešitev (x^*, y^*) preslikamo v pripadajočo množico največjih kodov brez prekritij po naslednjem postopku. Množico {*C*, *G*, *T*, *A*} razbijemo na poljubna dva dela tako, da ima del L_1x_1 elementov, del R_1 pa y_1 elementov. Naj (L_jR_{i-j}) označuje množico vseh besed, ki jih dobimo, če staknemo besedo iz množice L_j z besedo iz množice R_{i-j}. Za $i \in \{2, ..., n-1\}$ razbijemo množico $\cup_{j=1}^{i-1} (L_jR_{i-j})$ na poljubna dva dela tako, da ima del L x elementov, del R_i pa y_i elementov. Kodne besede ustrezajo množici $\cup_{i=1}^{n-1} (L_iR_{n-i})$. Spremenljivka n torej označuje dolžino kodnih besed. Če se omejimo na kode, ki se uporabljajo v DNA-pomnilnikih, so smiselne vrednosti za n manjše od 30. Vedoželjen bralec si lahko podrobnosti izpeljave in reševanja optimizacijskega problema prebere v [14].

$$\max \sum_{i=1}^{n-1} x_i y_{n-i}$$

pri pogojih

$$x_1 + y_1 = 4$$

$$x_1 + y_1 = \sum_{j=1}^{i-1} x_j y_{i-j} \; \forall i > 1$$

$$x_1, y_1 > 0$$

$$x_i, y_i \geq 0 \;\; \forall i >$$

$$x_i, y_i \in \mathbb{Z} \;\; \forall i \geq 1$$

Če bi želeli v optimizacijski problem vključiti tudi obe dodatni zahtevi za DNA-pomnilnike, ki smo ju opisali v uvodu, bi morali množice L_i, R_i razbiti na podmnožice tako, da posamezna podmnožica vsebuje zgolj besede z enakim število pojavitev črk iz množice {*G*, *C*}, ki se začnejo z isto črko in enakim številom ponovitev le-te, ter se končajo z isto črko in enakim številom ponovitev le-te. Vsaki podmnožici bi morali pripisati novo spremenljivko ter ustrezno nadomestiti kriterijsko funkcijo in omejitve. Tudi v tem primeru bi lahko nekatere neoptimalne rešitve odstranili iz preiskovalnega prostora s podobnimi matematičnimi izpeljavami, kot so predstavljene v [14] za kode brez prekritij brez dodatnih omejitev.

3 REZULTATI

Eksaktno reševanje opisanega optimizacijskega problema je navkljub zmanjšanju preiskovalnega prostora časovno zelo zahtevno [14]. Dodatne zahteve množico rešitev še dodatno eksponentno povečajo in s tem otežijo reševanje optimizacijskega problema. Če je delež besed, ki kršijo posamezno zahtevo, majhen, je zato bolj smiselno vzeti dobro rešitev osnovnega optimizacijskega problema in iz dobljenega koda brez prekritij izločiti kodne besede, ki ji ne zadoščajo.

Vzeli smo vse optimalne rešitve optimizacijskega problema za štiriške kode dolžine od 3 do 15 simbolov povzete po viru [14]. Za vsak kod smo izračunali delež kodnih besed, ki imajo največ pet zaporednih enakih simbolov, ter delež kodnih besed, ki imajo med 0,4n in 0,6n simbolov iz množice {*C*, *G*}.

Najmanjši in največji dobljeni deleži so prikazani v tabeli 1. Vidimo, da skoraj vse kodne besede zadoščajo kriteriju o omejenem številu zaporednih enakih simbolov, medtem ko le približno polovica kodnih besed zadošča kriteriju o omejenem deležu simbolov iz nabora {C,G}. Slednjega je torej smiselno vključiti v optimizacijski problem kljub povečanju časovne zahtevnosti reševanja problema.

Tabela1: **Najmanjši in največji deleži kodnih besed, ki izpolnjujejo kriterije. Oznaka *n* označuje dolžino kodnih besed, oznaka *N*(*n*) število različnih največjih štiriških kodov brez prekritij dolžine *n* in *S*(*n*) število kodnih besed v največjem štiriškem kodu brez prekritij dolžine *n*.**

			delež kodnih besed z največ 5 enakimi zaporednimi simboli		delež kodnih besed z GC-deležem med 40 in 60 %	
n	***N*(*n*)**	**S(*n*)**	**najmanjši**	**največji**	**najmanjši**	**največji**
3	8	9	100,000 %	100,000 %	44,444 %	44,444 %
4	8	27	100,000 %	100,000 %	69,136 %	69,136 %
5	8	81	100,000 %	100,000 %	32,922 %	32,922 %
6	24	251	99,759 %	99,759 %	54,041 %	63,691 %
7	24	829	99,598 %	99,598 %	26,297 %	31,519 %
8	24	2753	99,643 %	99,643 %	51,086 %	64,620 %
9	24	9805	99,533 %	99,533 %	67,384 %	80,364 %
10	24	34921	99,425 %	99,425 %	46,158 %	57,067 %
11	24	124373	99,317 %	99,317 %	62,152 %	73,050 %
12	120	446496	99,283 %	99,305 %	41,527 %	52,160 %
13	120	1619604	99,187 %	99,201 %	56,957 %	68,453 %
14	240	5941181	99,170 %	99,189 %	69,606 %	81,392 %
15	240	21917583	99,083 %	99,096 %	54,170 %	65,376 %

4 ZAKLJUČEK

Ugotovili smo, da lahko problem največjega naslovnega prostora DNA pomnilnika, ki uporablja kode brez prekritij, rešujemo eksaktno s pomočjo celoštevilske optimizacije. Ker nam pristop vrne vse največje kode, lahko izmed njih izberemo takega, ki ima najbolj ugodne biokemijske lastnosti. Ugotovili smo, da lahko pogoj o omejenem številu enakih zaporednih simbolov preverjamo sprotno, medtem ko bi bilo kriterij o omejenem GC-deležu smiselno vključiti v matematični model in prilagoditi optimizacijski problem.

Reševanje celoštevilskih optimizacijskih problemov s polinomsko kriterijsko funkcijo in polinomskimi omejitvami omogočajo tudi nekatera splošno namenska orodja za reševanje optimizacijskih problemov, ki jih nameravamo preizkusiti v nadaljevanju raziskovalnega dela. Omenjeni pristop ima sicer že znane pomankljivosti. Prva je, da številna orodja ne podpirajo nekonveksnih omejitev, ki so se pojavile v naši formulaciji. Pogosto je tudi taka orodja moč uporabiti, če zna uporabnik kreirati zaporedje relaksacij svojega optimizacijskega problema, ki bo konvergiralo k pravilni rešitvi. Lahko se zgodi, da rešitev konvergira tako počasi, da je pristop neuporaben, ali da se velikost relaksacij poveča čez mejo, ki jo orodja še podpirajo.

LITERATURA

[1] Daniella Bar-Lev, Itai Orr, Omer Sabary, Tuvi Etzion, and Eitan Yaakobi. Deep DNA storage: Scalable and robust DNA storage via coding theory and deep learning. *arXiv preprint arXiv:2109.00031*, 2021.

[2] Meinolf Blawat, Klaus Gaedke, Ingo Huetter, Xiao-Ming Chen, Brian Turczyk, Samuel Inverso, Benjamin W Pruitt, and George M Church. Forward error correction for DNA data storage. *Procedia Computer Science*, 80:1011–1022, 2016.

[3] Avital Boruchovsky, Daniella Bar-Lev, and Eitan Yaakobi. DNA-Correcting Codes: End-to-end Correction in DNA Storage Systems. *arXiv preprint arXiv:2304.10391*, 2023.

[4] Yeow Meng Chee, Han Mao Kiah, Punarbasu Purkayastha, and Chengmin Wang. Cross-bifix-free codes within a constant factor of optimality. *IEEE Transactions on Information Theory*, 59(7):4668–4674, 2013.

[5] George M Church, Yuan Gao, and Sriram Kosuri. Next-generation digital information storage in DNA. *Science*, 337(6102):1628–1628, 2012.

[6] Robert N Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J Stark. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angewandte Chemie International Edition*, 54(8):2552–2555, 2015.

[7] Reinhard Heckel, Ilan Shomorony, Kannan Ramchandran, and NC David. Fundamental limits of DNA storage systems. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 3130–3134. IEEE, 2017.

[8] V.I. Levenshtein. Maximum number of words in codes without overlaps. *Problemy Peredachi Informatsii*, 6(4):88–90, 1970.

[9] Maya Levy and Eitan Yaakobi. Mutually uncorrelated codes for DNA storage. *IEEE Transactions on Information Theory*, 65(6):3671–3691, 2018.

[10] Xiaozhou Lu and Sunghwan Kim. Weakly mutually uncorrelated codes with maximum run length constraint for dna storage. *Computers in Biology and Medicine*, 165:107439, 2023.

[11] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, et al. Scaling up DNA data storage and random access retrieval. *BioRxiv*, page 114553, 2017.

[12] Michael G Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J Lennon, Ryan Hegarty, Chad Nusbaum, and David B Jaffe. Characterizing and measuring bias in sequence data. *Genome biology*, 14:1–20, 2013.

[13] Jerrod J Schwartz, Choli Lee, and Jay Shendure. Accurate gene synthesis with tag-directed retrieval of sequence-verified dna molecules. *Nature methods*, 9(9):913–915, 2012.

[14] Lidija Stanovnik, Miha Moškon, and Miha Mraz. In search of maximum non-overlapping codes. *Designs, Codes and Cryptography*, pages 1–28, 2024.

[15] SM Tabatabaei Yazdi, Yongbo Yuan, Jian Ma, Huimin Zhao, and Olgica Milenkovic. A rewritable, random-access DNA-based storage system. *Scientific reports*, 5(1):1–10, 2015.

[16] Jose L Walteros and Austin Buchanan. Why is maximum clique often easy in practice? *Operations Research*, 68(6):1866–1895, 2020.

[17] SM Yazdi, Ryan Gabrys, and Olgica Milenkovic. Portable and error-free DNA-based data storage. *Scientific reports*, 7(1):1–6, 2017.

[18] SMH Tabatabaei Yazdi, Han Mao Kiah, Ryan Gabrys, and Olgica Milenkovic. Mutually uncorrelated primers for DNA-based data storage. *IEEE Transactions on Information Theory*, 64(9):6283–6296, 2018.

▪

Lidija Stanovnik je diplomirala in magistrirala na Fakulteti za računalništvo in informatiko ter Fakulteti za matematiko in fiziko Univerze v Ljubljani. Zaposlena je kot mlada raziskovalka na Fakulteti za računalništvo in informatiko Univerze v Ljubljani.

▪

Miha Mraz je diplomiral, magistriral in doktoriral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Na isti ustanovi je zaposlen kot redni profesor in poučuje predmete Zanesljivost in zmogljivost računalniških sistemov, Modeliranje računalniških omrežij in Nekonvencionalne platforme in metode procesiranja. Je (so)avtor več kot 200 raziskovalnih del.