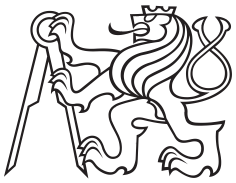
Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

Ukládání archivních dat a efektivní kódování obrazu s využitím DNA

DNA-based Archival Data Storage and Efficient Image Coding

Bc. Matouš Vobr

Vedoucí práce: Ing. Karel Fliegel, Ph.D.
Květen 2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vobr** Jméno: **Matouš** Osobní číslo: **491927**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Elektronika a komunikace**

Specializace: **Audiovizuální technika a zpracování signálů**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Ukládání archivních dat a efektivní kódování obrazu s využitím DNA

Název diplomové práce anglicky:

DNA-based Archival Data Storage and Efficient Image Coding

Pokyny pro vypracování:

Podejte přehled současného stavu v oblasti využití DNA pro ukládání archivních dat. Zaměřte se zejména na přístupy vedoucí k efektivnímu kódování obrazových dat za účelem archivace pomocí DNA a to zejména s ohledem na probíhající standardizační aktivitu JPEG DNA. Seznamte se s dostupnými nástroji pro simulaci souvisejících metod ukládání a kódování dat do DNA. Vybrané přístupy implementujte a ověřte jejich účinnost.

Seznam doporučené literatury:

[1] Pan, C., Tabatabaei, S. K., Tabatabaei Yazdi, S. M. H., Hernandez, A. G., Schroeder, C. M., Milenkovic, O. Rewritable two-dimensional DNA-based data storage with machine learning reconstruction, Nature Communications, 2022.
[2] Dimopoulou M., Antonini M., Barbry P., Appuswamy R. A biologically constrained encoding solution for long-term storage of images onto synthetic DNA, European Signal Processing Conference, 2019.
[3] ISO/IEC JTC 1/SC29/WG1 N100154, DNA-based Media Storage: State-of-the-Art, Challenges, Use Cases and Requirements, 2022 (https://jpeg.org/jpegdna/documentation.html).

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Karel Fliegel, Ph.D. katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **03.02.2024** Termín odevzdání diplomové práce: **24.05.2024**

Platnost zadání diplomové práce: **21.09.2025**

Ing. Karel Fliegel, Ph.D.	doc. Ing. Stanislav Vítek, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji Ing. Karlu Fliegelovi, Ph.D. za vedení mé diplomové práce, za podnětné návrhy, které práci obohatily a za čas, který mi věnoval při konzultacích.

Prohlášení

Prohlašuji, že jsem předloženou diplomovou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 22. května 2024

Abstrakt

Lidstvo dnes čelí výzvě spravovat a uchovávat exponenciálně rostoucí objemy dat na internetových úložištích. Tento nárůst je klasickými úložnými metodami neudržitelný, což vyžaduje hledání udržitelnějších způsobů archivace dat.

Tato práce zkoumá využití umělé syntézy DNA pro archivaci obrazových dat, což představuje inovativní přístup k dlouhodobému uchovávání dat s vysokou hustotou a nízkou energetickou náročností. Hlavním cílem bylo ověřit a porovnat účinnost různých přístupů kódování dat do DNA na vybraném testovacím setu obrazových souborů. Byla hodnocena kvalita rekonstruovaného obrazu, dodržování restrikcí DNA kódu, chybovost, robustnost vůči těmto chybám a výpočetní náročnost komprese a dekomprese.

Byla použita množina nejpoužívanějších či nejnovějších binárních obrazových kodérů skupiny JPEG, kvaternární transkódovací DNA schémata a objektivní metriky pro stanovení kvality obrazu doporučované odvětvím JPEG DNA. Výsledky ukázaly, že schémata Goldman, Grass a Church dosahují nejlepšího dodržování biochemických restrikcí DNA molekul a nejmenší chybovosti. Nejrobustnějším a kompresně nejefektivnějším modelem se ukázala kombinace JPEG XL bitového kodéru s Goldman DNA schématem.

Možná rozšíření této práce zahrnují otestování fontánového DNA kódování pro vyšší hodnoty redundance a prodloužení bitové segmentace, což by mohlo zlepšit vyváženost GC obsahu a tedy snížit chybovost.

Klíčová slova: JPEG DNA, archivace dat, transkódující DNA architektura, objektivní metriky kvality, KODAK.

Vedoucí práce: Ing. Karel Fliegel, Ph.D.
Praha, Technická 1902/2, místnost: B3-556

Abstract

Humanity is currently facing the challenge of managing and preserving exponentially growing volumes of data on internet storage platforms. This growth is unsustainable with traditional storage methods, necessitating the search for more sustainable data archiving solutions.

This thesis explores the use of artificial DNA synthesis for image data archiving, presenting an innovative approach to long-term data storage with high density and low energy consumption. The main objective was to verify and compare the efficiency of various data encoding approaches into DNA on a selected test set of image files. The quality of the reconstructed image, compliance with DNA code restrictions, error rates, robustness to these errors, and computational demands of compression and decompression were evaluated.

A set of the most widely used or latest binary image coders from the JPEG group, quaternary transcoding DNA schemes, and objective metrics for image quality assessment recommended by the JPEG DNA sector were used. The results showed that the Goldman, Grass, and Church schemes achieved the best compliance with biochemical restrictions of DNA molecules and the lowest error rates. The most robust and compression-efficient model was found to be the combination of the JPEG XL bit coder with the Goldman DNA scheme.

Possible extensions of this work include testing fountain DNA coding for higher redundancy values and extending bit segmentation, which could improve GC content balance and reduce error rates.

Keywords: JPEG DNA, data archiving, transcoding DNA architecture, objective quality metrics, KODAK.

Title translation: DNA-based Archival Data Storage and Efficient Image Coding

Obsah

Obrázky

Tabulky

Kapitola 1
Úvod

V dnešní éře digitalizace je neustále rostoucí množství dat k uchování jednou z hlavních výzev moderní společnosti. S nárůstem digitálních informací vzniká stále větší potřeba efektivního a dlouhodobě udržitelného řešení pro jejich uchování. Koneckonců množství uložených dat ve světě mezi lety 2010 a 2023 se zvětšilo na šedesátinásobek (120 ZB) [2]. Energie, kterou datová centra celosvětově spotřebují, vzrostla mezi stejnými roky přibližně čtyřnásobně (10^{15} kWh za rok). [3] Současné metody ukládání dat, založené na konvenčních elektronických médiích, jako jsou pevné disky a serverová úložiště, se potýkají s omezením kapacity a energetickými nároky.

V posledních letech se vědecká komunita obrací k biologickým molekulám jako DNA jakožto potenciálnímu médiu pro ukládání dat. DNA nabízí několik výhod, které ji činí atraktivním kandidátem pro dlouhodobé uchovávání informací. Jednou z těchto výhod je její extrémně vysoká hustota dat, která umožňuje uložit obrovské množství informací do velmi malého objemu. Ku příkladu již zmíněných 120 ZB dat všech informací na internetu by se daly uložit buď na nepředstavitelné množství pevných disků (12×10^{12} terabytových HardDisků), nebo dle propočtů maximální teoretické hustoty DNA jen na 1 cm^3 takové DNA [4]. Navíc DNA vykazuje i velice dlouhou životnost a stabilitu, což z ní činí ideální prostředek pro uchování dat po dlouhou dobu bez značného úbytku kvality.

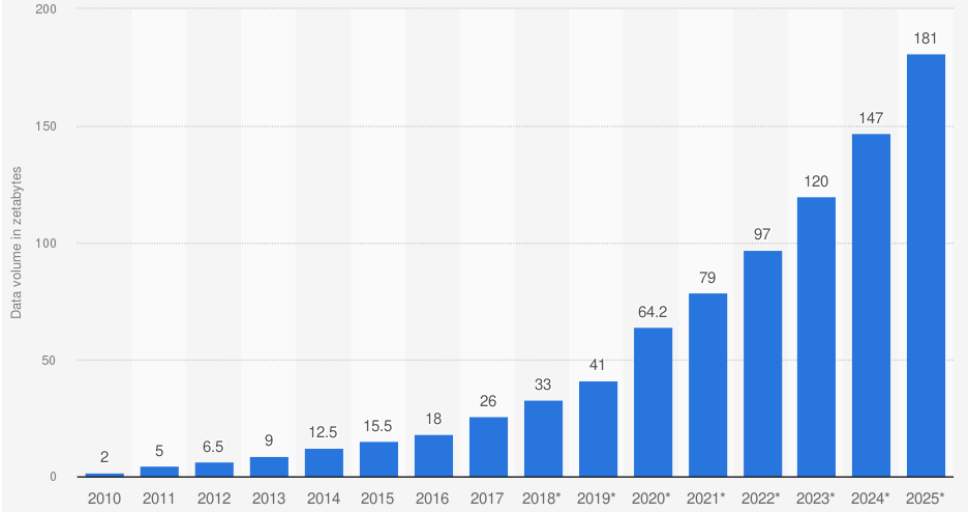
Mezi cíle této práce patří prozkoumání současných poznatků a omezení v oblasti ukládání dat do DNA. Dále je cílem nalézt a popsat konkrétní přístupy, které byly vyvinuty v posledních letech v této oblasti. U zvolených metod se zaměřit na testování jejich účinnosti, objektivní hodnocení kvality generovaných obrazů, simulaci chybovosti těchto metod a stanovení jejich robustnosti na základě zjištěných výsledků.

Kapitola 2 poskytuje nezbytné teoretické základy pro oblast ukládání dat do DNA. Tato kapitola zahrnuje přehled stavby DNA, diskutuje výhody a potenciál ukládání dat do DNA, a popisuje biochemická omezení umělého DNA. Dále jsou zde popsány požadavky vyplývající z těchto omezení na navrhované DNA kódy a hlavní přístupy kódování informace do DNA.

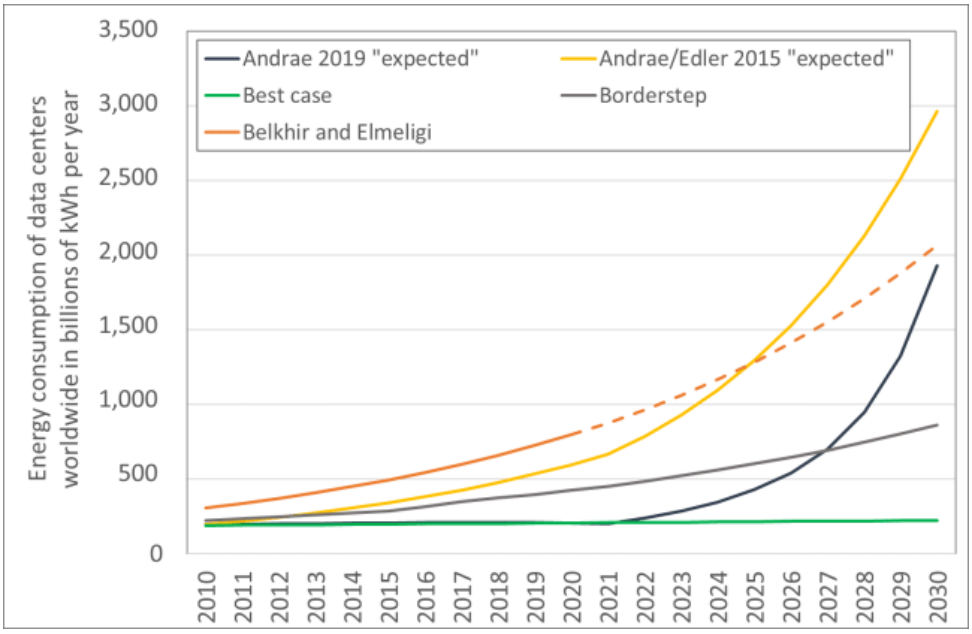
Kapitola 3 se zaměřuje na popis konkrétních implementací kódování obrazových dat do DNA, které byly vyvinuty v posledních letech.

Kapitola 4 je stěžejní částí této práce, kde jsou vybrané přístupy testovány

na specifickém obrazovém datasetu. Jejím cílem je stanovit obrazovou kvalitu ovlivněnou kompresním zkreslením, určit kompresní účinnost, zhodnotit dodržování restrikcí DNA řetězců a simulovat chybovost, která plyne z případného nedodržení těchto restrikcí.



Obrázek 1.1: Celosvětový vývoj množství dat v *ZB*. Převzato z [2].



Obrázek 1.2: Celosvětový vývoj spotřeby energie datových center. Převzato z [3].

Kapitola 2
Teoretické základy ukládání dat do DNA

Tato kapitola poskytuje teoretické základy nezbytné pro pochopení ukládání dat do DNA. V první sekci 2.1 je představena činnost skupiny JPEG DNA, která hraje klíčovou roli v této oblasti. Na základě zdrojů její iniciativy je zde uveden základní koncept DNA, včetně její struktury a vlastností. Dále je popsáno základní schéma kódování dat do DNA řetězců a diskutovány hlavní přístupy kódování informací do DNA. Kapitola také analyzuje biochemická omezení, která jsou stanovena pro procesy ukládání dat do DNA, a jak tato omezení ovlivňují návrh kódovacích metod.

2.1 Iniciativa JPEG DNA

Skupina JPEG (Joint Photographic Experts Group) je známá svými standardy v oblasti obrazové komprese a od roku 2020 se zaměřuje také na standardizaci ukládání dat do DNA prostřednictvím iniciativy JPEG DNA [5].

Cílem JPEG DNA je vytvoření standardu pro efektivní kódování obrazu, který zohledňuje biochemická omezení a nabízí odolnost vůči šumu zaváděnému různými fázemi procesu ukládání. Současný vývoj světového objemu dat potřebných k archivaci nabádá k prozkoumání nových způsobů ukládání dat a právě DNA, díky své vysoké koncentraci informace v malém objemu, dlouhé životnosti a nízké energetické náročnosti, se jeví jako ideální kandidát [5].

Na svém 99. online zasedání vydala JPEG komise konečnou výzvu k podání návrhů (*Call for Proposals*) pro JPEG DNA [6]. Tato výzva byla doprovázena dvěma dalšími dokumenty: Požadavky na JPEG DNA [7] a Společné testovací podmínky JPEG DNA [8].

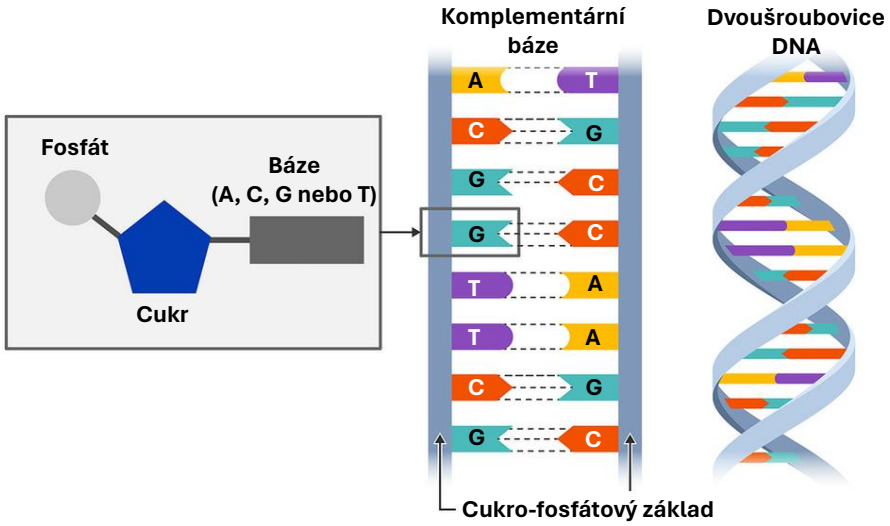
JPEG DNA se aktuálně nachází ve fázi zdokonalení pracovního návrhu, který byl vybrán na základě výzvy k podání návrhů a je neveřejný. Návrh mezinárodního standardu se očekává v lednu 2025, přičemž konečná publikace mezinárodního standardu je plánována na tentýž rok [5].

2.2 Co je to DNA?

DNA (z angl. deoxyribonucleic acid) je zkratka pro deoxyribonukleovou kyselinu. Je to molekula obsažená v buňkách prakticky všech živých organismů. Tvoří základní stavební kámen genetického materiálu a nese informace potřebné pro vývoj, funkci a dědičnost organismů. DNA se skládá z dvojitého řetězce nukleotidů (tzv. komplementárních řetězců), které obsahují dusíkaté báze:

- **Puriny**, což jsou větší molekuly: **adenin (A)** a **guanin (G)**
- **Pyrimidiny**, což jsou menší molekuly: **cytosin (C)** a **thymin (T)**

Tyto báze jsou spojeny deoxyribózou a fosfátem. Tato molekula je esenciální pro uchování a přenos genetické informace mezi generacemi a je zodpovědná za různorodost života na Zemi [9].



Obrázek 2.1: Struktura DNA znázorňující cukr-fostátový základ a komplementární báze dvoušroubovici DNA. Upraveno z [1].

2.3 Výhody ukládání dat do umělých DNA řetězců

V dnešní době, kdy objemy dat potřebné k dlouhodobému uchování stále rostou, se zejména v poslední dekádě objevuje stále více výzkumů, které mapují potenciál archivace dat v umělých DNA řetězcích. Tyto výzkumy přinášejí značné výhody zejména pro účely ukládání v tzv. *cold data centers*, což jsou datová střediska uchovávající data, která jsou nepravidelně a velmi zřídka používaná a aktualizovaná.

Následující body popisují hlavní výhody využití DNA jako prostředku pro uchování dat:

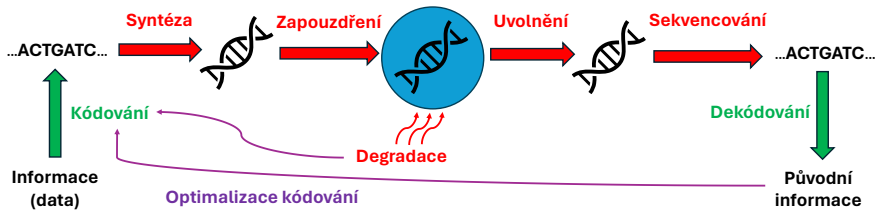
- **Vysoká kapacita a hustota:** DNA má extrémně vysokou informační hustotu, což umožňuje ukládat obrovské množství dat v malém objemu. Dle některých autorů se dnešní kapacita pohybuje až ve stovkách petabytů (PB) na 0,1 mm^3 DNA [10].
- **Dlouhodobá stabilita:** Při vhodném uchování mohou DNA řetězce přežít stovky až tisíce let, což je v porovnání s tradičními datovými médii, která se musejí obměňovat v průměru každé 3 až 5 let, výrazný nárůst životnosti [11, 12].
- **Energetická efektivita:** DNA úložiště mají potenciál být mnohem energeticky úspornější. Mohou snížit energetickou spotřebu až tisíckrát ve srovnání se současnými datovými centry [13, 14].
- **Snadná replikace (kopírování dat):** Díky moderním a neustále se zlepšujícím metodám replikace DNA *in vitro*, jako je například PCR (z angl. *Polymerase Chain Reaction*), je možné rychle a efektivně kopírovat archivní data uložená v DNA. Zatímco kopírování stovek až tisíců terabytů (TB) dat uložených v tradičních úložištích může trvat týdny až měsíce, stejné množství dat uložených v DNA lze pomocí PCR replikovat během několika minut až hodin [15].

2.4 Základní stavba end-to-end procesu ukládání dat do DNA

Základní kroky *end-to-end* procesu ukládání dat do DNA zahrnuje několik klíčových fází, které jsou nezbytné pro úspěšné zakódování, syntézu, uchování, a následné dekódování dat uložených v DNA.

Následující body poskytují přehled každého z těchto kroků [5]:

1. **DNA syntéza:** Syntéza definovaného řetězce DNA (molekuly).
2. **Zapouzdření:** Uchování fyzické uměle vytvořené DNA molekuly v médiu.
3. **Simulace degradace molekuly:** Predikce a příprava na možnou degradaci molekuly způsobenou faktory jako teplota nebo UV záření, včetně **optimalizace** procesu kódování, pro tvorbu robustního systému odolného vůči takovým chybám.
4. **Uvolnění DNA molekuly:** Vyjmutí DNA molekuly z média.
5. **Sekvencování DNA:** Čtení DNA řetězce.
6. **Dekódování:** Přeložení řetězce (ACGT) zpět do původní informace.



Obrázek 2.2: Stavba end-to-end procesu využití DNA jako úložného média a optimalizace takového systému.

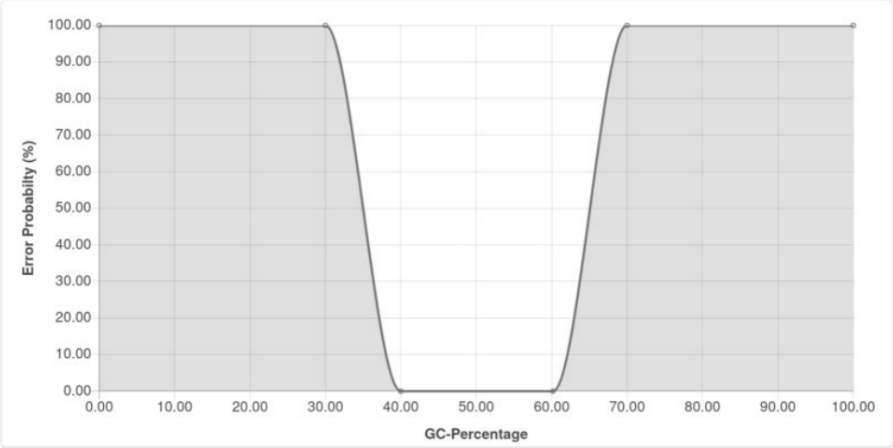
2.5 Biochemická omezení DNA řetězců

Na první pohled by se mohlo zdát, že celý proces přechodu od tradiční binární reprezentace informací (logická 0 a 1) k reprezentaci v podobě řetězců DNA (tedy kvatenární reprezentace čtyř symbolů A,C,G a T) je poměrně jednoduchá a přímočará. Problémem však je, že tato kvaternární reprezentace se neoddělitelně váže ke skutečným fyzickým řetězcům, které je třeba syntetizovat, uchovat a následně zase sekvencovat (přečíst). Na chybovost je pak nejvíce náchylný právě proces sekvencování a to klade na generování takových řetězců *in silico* značná omezení [16].

Metody syntézy, uchování a sekvencování DNA se však vyvíjejí každým rokem a je docela možné, že v blízké budoucnosti budou tato dobře zdokumentovaná omezení podstatně redukována, nebo zcela minulostí [4]:

- **Zákaz homopolymerů:** Generované řetězce by neměly obsahovat za sebou jdoucí nukleotidy stejné báze [5]. Zdroje se pak rozcházejí v tom, jaký maximální počet opakujících se bází je ještě přípustný, ale nejčastěji se uvádí maximálně tři opakování. U tohoto množství se totiž reálnými pokusy syntézy a následného sekvencování ukázalo, že lze ještě s dostatečnou pravděpodobností zaručit jejich úspěšnost [17, 18].
- **Vyrovnaný obsah GC (angl. *GC content*):** Procentuální zastoupení bází Guaninu a Cytosinu v celém oligonukleotidu (krátký umělý jednovláknový DNA řetezec) by mělo být mezi 40% až 50%. To pak zaručuje stabilitu takové molekuly obecně a zároveň snižuje chybovost při jejím sekvencování. Jedná se tedy o nejdůležitější biochemický faktor DNA molekul [5, 16].

$$\mathrm{GC}_{\mathrm{content}} = \frac{\sum(G+C)}{\sum(A+C+G+T)} \times 100 \in [40\%, 50\%] \qquad (2.1)$$



Obrázek 2.3: Simulovaná chybovost při sekvencování řetězců s různým procentuálním zastoupením GC obsahu. Převzato z [16].

- **Opakování nežádoucího vzoru:** V řetězcích by se neměly vyskytovat po sobě jdoucí dvojice až pětice stejných nukleotidů (např. -*ATATAT*-, -*ACGACGACG*-, ...). Zdroje se zde trochu rozcházejí ohledně maximálního přípustného počtu těchto opakování [5, 19], především protože tento vzor opakování se nachází částečně i v lidském genomu jako tzv. *short tandem repeats*. Takové repetice se mohou při buněčné replikaci DNA prodlužovat a tím zastínit některé důležité geny kódující syntézu proteinů, což může vést k mutaci organismu [20]. K podobnému zastínění informace během syntézy umělého DNA může dojít právě i při výskytu takovýchto opakujících se vzorů. Skupina JPEG DNA proto doporučuje v umělých řetězcích se zcela vyhnout těmto opakováním, ale povoluje maximálně čtyři po sobě jdoucí opakování [5].
- **Omezení délky řetězců:** Generovaný kvatenární datový tok by měl být rozdělen na oligonukleotidy (kratší řetězce), které mají délku v rozsahu 50 až 300 nukleotidů. Toto kritérium je stanoveno, protože u delších sekvencí nelze zaručit jejich nulovou chybovost jak při syntéze, tak sekvencování a kratší sekvence by nesly příliš malou část celkové informace. Konkrétní doporučení je tedy délka 200 nukleotidů [5].

2.6 Požadavky na ukládání dat do DNA

V této sekci jsou uvedeny nejdůležitější požadavky na ukládání dat do syntetických DNA řetězců, jak je stanovila skupina JPEG DNA [5]:

1. **Kompresní účinnost:** Při přechodu na reprezentaci dat v kvaternárním systému (systém se čtyřmi symboly) by měl kompresní algoritmus minimálně nezvětšovat počet symbolů potřebných k reprezentaci komprimované informace. Ideálně by měl přinášet výraznou kompresní výhodu.

2. **Náhodný přístup:** Části dat by měly být dekódovatelné bez nutnosti znalosti částí předchozích. Vzhledem k tomu, že je celkový kvaternární datový tok rozdělen na oligonukleotidy, které jsou uloženy jako fyzické molekuly v médiu, nelze je tedy číst v libovolném pořadí.

3. **Robustnost:** I přes dodržení všech biochemických omezení DNA řetězců (viz sekce 2.5) může působením okolního prostředí (při uchování sekvencí) a nedokonalostech v procesu syntézy a sekvencování docházet k chybám. Vytvořený kód by měl být odolný vůči těmto nepříznivým vlivům.

4. **Škálovatelnost:** Při sekvencování oligonukleotidů by mělo docházet k postupnému zpřesňování výsledku v reálném čase, jak jsou čteny jednotlivé sekvence.

5. **Jednoznačnost:** Rekonstruovaná informace musí být jednoznačná, tedy nedochází k žádným nesrovnalostem.

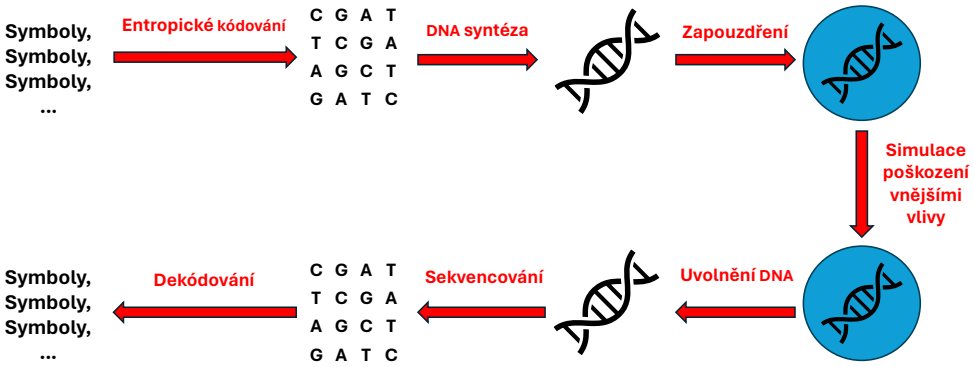
6. **Rozeznatelnost:** Po přečtení neznámé molekuly DNA by mělo být jasně rozpoznatelné, zda je umělého nebo přírodního původu.

2.7 Hlavní přístupy kódování informace do DNA

V této sekci jsou shrnuty dva hlavní přístupy kódování informace (obrázek, pdf, archiv, atd.), jak je stanovuje i skupina JPEG DNA ve svém *Call for Proposals* (viz sekce 2.1). Tato výzva žádá autory, aby předložili své návrhy na standardizovaný kodér, který splňuje maximum pravidel shrnutých v sekcích 2.5 a 2.6. Kompletní požadavky spolu s požadovanými metrikami, referenčními obrázky a dalšími informacemi pro testování účinnosti kodéru lze nalézt pod odkazem [5].

2.7.1 Přímé entropické kódování symbolů do ACGT řetězců

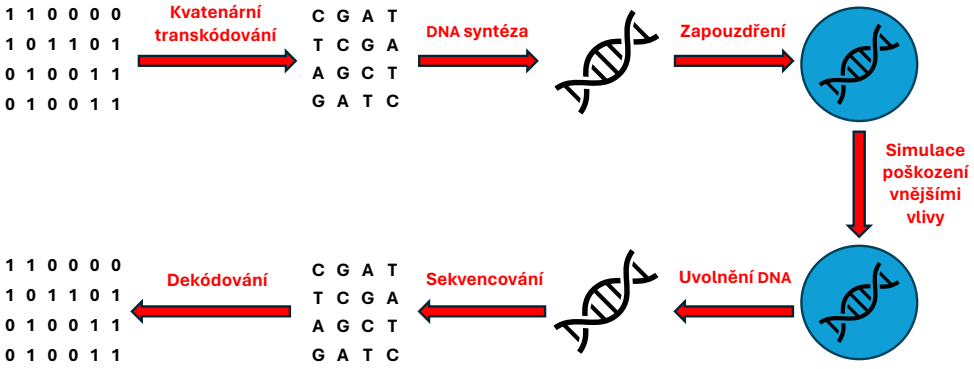
Tento přístup spočívá v přímém kódování symbolů (např. kvantizované DCT koeficienty obrázku) na optimální reprezentaci v kvatenárním systému DNA řetězců. Tento přístup má velký kompresní potenciál, protože hledá optimální reprezentaci surových dat už v požadovaném kvatenárním systému.



Obrázek 2.4: Architektura přímého entropického kódování symbolů informace do kvatenárního DNA kódu a zpětného dekódování.

2.7.2 Transkódování binárních dat do ACGT řetězců

Tento přístup je naopak poměrně jednoduchý a *de facto* rozděluje kodér na dvě nezávislé struktury. Spočívá v tom, že na vstupu nemá k dispozici surová obrazová data, nýbrž již komprimovaný binární tok těchto dat (tedy řetězec logických 0 a 1). Pro tento binární tok pak hledá odpovídající optimální reprezentaci v kvaternárním systému DNA. Lze jej tedy rozdělit na entropický binární kodér dat (který lze zvolit podle charakteru dat) a transkodér binárního toku do ACGT řetězců (nezávislý na typu dat, pracuje pouze s binárními řetězci a není důležité, co bitový tok představuje).



Obrázek 2.5: Architektura transkódování binárního toku do kvatenárního DNA kódu a zpětného dekódování.

Kapitola 3

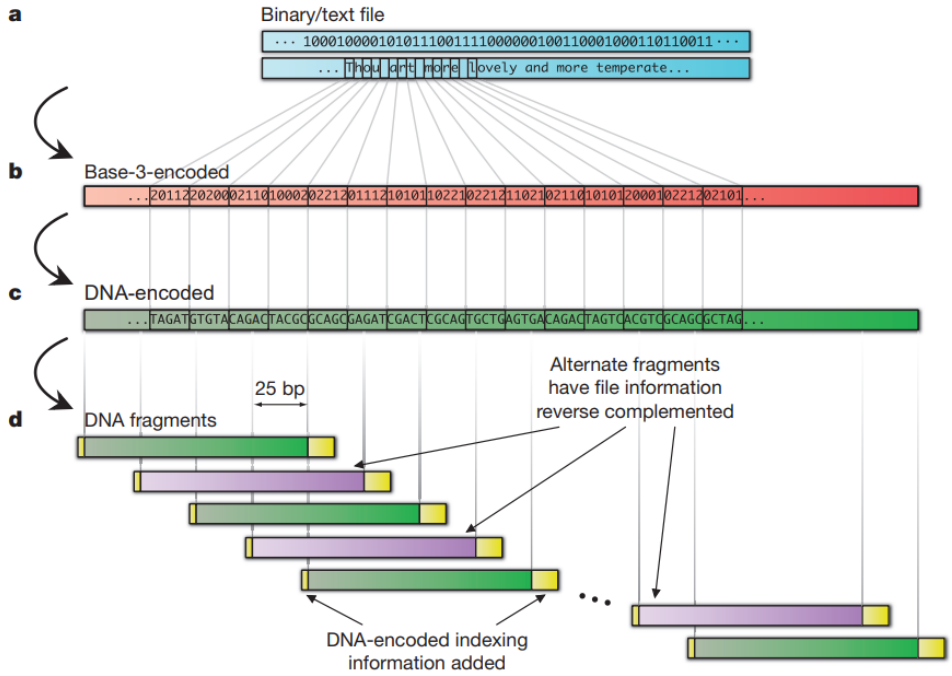
Současný stav

Tato kapitola se zaměří na konkrétní kodovací přístupy dat do DNA řetězců posledních let. Konkrétně z pohledu transkódovacího přístupu binárních dat do DNA na Goldman kódování v sekci 3.1, Grass v sekci 3.2, Churh v sekci 3.3 a Blawat kódování v sekci 3.4. Dále pak transkódující Fontánové DNA kódování v sekci 3.5. Z pohledu přímého entropického kódování jsou zde pak popsány dvě konrétní implementace v sekcích 3.6 a 3.7.

3.1 Goldman schéma

V roce 2013 publikoval Goldman a jeho tým v [10] algoritmus pro transkódování binárních dat do kvatenárního systému DNA, který respektoval tehdy nejdůležitější a poměrně čerstvě popsaná biochemická omezení DNA kódů GC obsah a zákaz homopolymerů (viz sekce 2.5).

Jejich přístup nejdříve data převede do trojkového systému pomocí Huffmanova algoritmu (kratší kódová slova přiřazena frekventovanějším částem dat atd.). Jakmile má každá část dat svou reprezentaci v Huffmanově stromu, tak jednotlivým znakům přiřazuje znak z DNA kvatenárního kódu (A,C,G či T) tak, že respektuje předchozí přiřazení, aby se znaky neopakovaly a tedy nevznikaly homopolymery. Při následném sekvencování řetězce využili délky segmentů 100 znaků s posuvným 75% překryvem. Tím vytvořili čtyřnásobnou redundanci, kterou chtěli zaručit rekonstruovatelnost každé části původního binárního řetězce.



Obrázek 3.1: Schéma principu Goldmanovo kódování. Převzato z [10].

3.2 Grass schéma

Jde o další schéma transkódující architektury, které převádí bitové segmenty do desítkové soustavy a následně využívá předdefinovaná mapovací pravidla pro převod na trojice nukleotidů pomocí mapovací tabulky, která je vytvořena na základě kombinatorických pravidel. Tvořený kvatenární kód je navíc kontrolován na dodržování biochemických restrikcí a zanášen případnou redundancí podle předdefinovaných pravidel, která ji odstraní [21].

3.3 Church schéma

Schéma Church je také transkódující a využívá binární bitové segmenty k výběru z dvojic nukleotidových možností (A, C) a (G, T). Každý bit pak určuje, která dvojice se použije (první, nebo druhá). Pokud jsou poslední tři nukleotidy stejné, vybere se jiný nukleotid z dané dvojice, aby se zabránilo homopolymerům, a tak se pokračuje až do konce bitového toku [22].

3.4 Blawat schéma

Dalším schématem transkódující architektury je Blawat, které kombinuje první tři bity segmentu bitového toku do dvou nukleotidů a poslední dva bity do dalších dvou nukleotidů. Tímto způsobem se postupuje až do zakódování celého segmentu. Kombinace těchto nukleotidů je zvolena tak, aby se

maximalizovalo dodržování biochemických restrikcí a tedy minimalizovala se pravděpodobnost chyb [23].

3.5 DNA Fountain - NOREC4DNA

DNA Fountain je technika kódování dat do DNA, která využívá tzv. *fontánového kódování* (také známé jako *NOREC* z angl. *near-optimal rateless erasure code*, tedy česky *téměř optimální bezchybové kódování s proměnlivou rychlostí*) [24]. Jedná se o způsob segmentace dat na menší bloky, které jsou následně kódovány do DNA. Umožňuje robustní a efektivní ukládání velkého množství dat do molekul DNA. Využívá redundanci informací a chybově odolného kódování, tedy je vhodným přístupem právě pro kódování dat do DNA.

Knihovna sjednocující nejmodernější přístupy fontánového kódování pro účely transkódování dat do DNA řetězců je **NOREC4DNA** [19]. Mezi nejmodernější přístupy patří **Luby-transform (LT) kódování** (viz sekce 3.5.1), **Online kódování** (viz sekce 3.5.2) a **Raptor kódování** (viz sekce 3.5.3).

3.5.1 LT kódování

LT transformace (popsaná v [25]) dělí binární datový tok vstupu na ekvivaletně dlouhé segmenty. Z těch pseudonáhodně vybírá podle pravděpodobnostní distribuční funkce (*RobustSoliton distribuce*) dvojice, které kombinuje pomocí XOR operace, a tím tvoří packety. Informace o indexech dvou segmentů, ze kterých je XOR operací packet vytovřen, a *seed* (náhodnosti) výběrů se k packetu přidávají, aby bylo možné informaci rekonstruovat. Packet se následně převede z binární do kvatenární DNA reprezentace (pomocí Goldmanova přístupu, viz sekce 3.1) a ověří se, zda-li segment splňuje biochemické restrikce (viz sekce 2.5). Pokud ne, tak se generuje nový packet až do chvíle zisku požadovaného počtu packetů stanoveného uživatelem. Ten tedy volí určitou úroveň redundance, tedy kolik více packetů bude generováno, než bylo původních segmentů.

$$Počet\ packetů = (1 + \epsilon) \times Počet\ segmentů \tag{3.1}$$

LT však nezaručuje pro libovolná obrazová data hranici redundance zajišťující celkovou rekonstrukci informace.

3.5.2 Online kódování

Online kódování (podrobněji popsáno v [26]) je rozšířením LT transformace, které řeší problém se zaručením dostatečné reprezentace všech segmentů binárního toku v zakódovaných packetech. Tedy zaručuje rekonstruovatelnost původní informace při daném počtu packetů (čili optimální redundance).

3.5.3 Raptor kódování

Raptor kódování (podrobněji popsáno v [27]) je implementací fontánových kódů s teoreticky lineárním časem kódování. Jedná se tedy o výpočetní optimalizaci výše popsaných postupů (viz sekce 3.5.1 a 3.5.2).

3.6 PAIRCODE

V práci [17] z roku 2019 autoři navrhli asi nejvíce citovaný postup přímého entropického kódování obrazových dat do DNA segmentů. Svůj kodér nazvali PAIRCODE. K vytvoření celé kódové knihy, se kterou kodér pracuje, použili dvě základní knihovny:

$$D_1 = \{AT, AC, AG, TA, TC, TG, CA, CT, GA, GT\} \tag{3.2}$$

$$D_2 = \{A, T, C, G\}, \tag{3.3}$$

které tedy slouží jako stavební kameny pro tvorbu kódových slov. Tyto dvě knihovny a zbytek algoritmu jsou navrženy tak, aby docházelo k požadovanému obsahu GC a zákazu homopolymerů, viz 2.5.

Jejich algoritmus využívá transformace obrazu pomocí DWT a tím vytvoří tzv. *hladiny* (jednotlivé úrovně). Ty uniformně skalárně kvantizují a určí kódovou knihu potřebnou pro zakódování všech kvantizovaných hodnot všech hladin obrazu podle 2 pravidel:

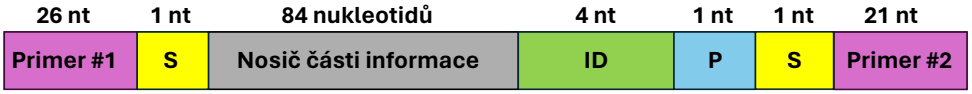
1. Kódová slova **sudé** délky l vytvoří všemi $\frac{l}{2}$ permutacemi z knihovny 3.2
2. Kódová slova **liché** délky l vytvoří všemi $\left\lfloor \frac{l}{2} \right\rfloor$ permutacemi z knihovny 3.2 a jedním znakem na konci z knihovny 3.3.

Nakonec pomocí pseudonáhodného mapování přiřadí jednotlivým kvantizovaným hodnotám jejich kódové slovo z vytvořené knihovny v předchozím kroku. Vytvořený kvatenární tok následně dělí na řetězce délky 84 nukleotidů. K těm navíc autoři přidávají některé další důležité struktury:

- **Primery:** krátké, laboratořemi definované úseky nutné pro zahájení a ukončení syntézy umělých řetězců.
- **Orientační nukleotidy:** určují v jakém směru se má řetězec číst (označují jako S)
- **Identifikační nukleotidy:** určující typ dat či pořadí řetězce (označují jako ID)
- **Paritní kontrolní nukleotid:** ke zjištění chyb v datech (označují jako P)

Nespecifikují však jakou strukturu by např. *ID* nukleotidy měly mít, jelikož toto je předmětem očekávané standardizace skupiny JPEG DNA. Je však vhodné zmínit, že i takové struktury (*Primery*, *ID*, ...) ve výsledných řetězcích budou zastoupeny, i když práce jiných autorů tento fakt příliš nezmiňují.

V roce 2020 autoři tento model vylepšili tím, že nahradili skalární kvantizér vektorovým, čímž zvýšili účinnost komprese [28].



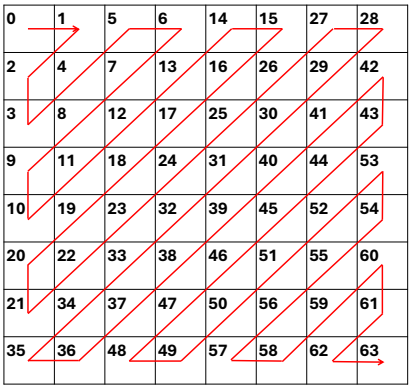
Obrázek 3.2: Struktura jednotlivých částí DNA řetězce v PAIRCODE. Vytvořeno podle [17].

3.7 DNA kodér s proměnnou délkou slov

V roce 2021 v [18] přišel tým s návrhem kodéru obrazových dat do DNA, který nevytvářel fixní délku slov pro jednotlivé kvantizované koeficienty DCT. Při jeho tvorbě se velmi inspirovali klasickou strukturou JPEG 1 kodéru (*Legacy JPEG*) z roku 1994 [29]. Svůj výsledek nazvali jako *JPEG-DNA*.

Princip *JPEG-DNA*:

1. Posun rozsahu surových obrazových dat z [0, 255] na [-128, 127].
2. Výpočet DCT po blocích o rozměrech 8×8 pixelů.
3. Aplikace uniformní skalární kvantizaci podle předdefinovaných kvantizačních tabulek JPEG [29].
4. Čtení bloků v tzv. *zig-zag* směru pro získání 1D vektorů.
5. DC koeficienty (nulté) jsou kódovány rozdílově vzhledem k předchozímu (tedy první je kódován přímo)
6. Velikost rozdílu je poté kategorizována podle velikosti a dané kategorii je přiřazeno kódové slovo podle Goldmanova algoritmu, viz sekce 3.1.
7. DC koeficienty jsou následně kódovány jako složení kódových slov kategorie a dané hodnoty (získáno pomocí *PAIRCODE*, viz sekce 3.6).
8. AC koeficienty jsou kódovány velice podobně, ale místo hodnoty přímo se využívá *RLE* kódování, které se opírá o fakt, že mezi AC koeficienty je mnoho se opakujících nulových hodnot.

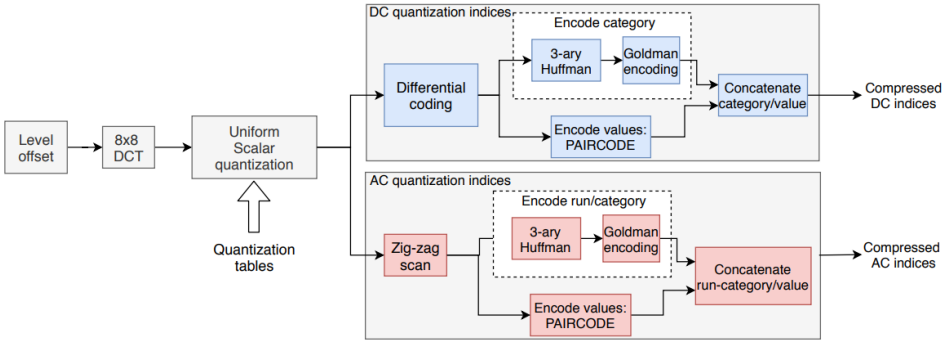


(a) : JPEG-DNA: *Zig-zag* směr čtení matice DCT koeficientů pro stanovení jejich pořadí.

Absolutní hodnota rozdílu	Kategorie (= počet nukleotidů k zakódování)
0	0
[1,5]	2
[6,25]	3
[26, 75]	4
[76, 275]	5
[276, 775]	6
[776, 2775]	7
[2776, 7775]	8

(b) : JPEG-DNA: kategorie rozdílů mezi kódovaným DCT koeficientem a předchozím, které stanovují potřebný počet nukleotidů k zakódování daného rozílu.

Obrázek 3.3: JPEG-DNA: (a) Zig-zag, (b) Kategorie.



Obrázek 3.4: JPEG-DNA: celkové schéma kodéru. Převzato z [18].

3.8 Chamaeleo framework

Chamaeleo, popsané v roce 2020 v [30], je knihovna obsahující největší množství schémat pro transkódování binárních dat do DNA (viz sekce 2.7.2), která je vhodná pro potřeby této práce.

Její součástí jsou optimalizované implementace schémat Goldman, Grass, Church a Blawat. Navíc obsahuje i fontánový DNA transkodér a vlastní transkodér autorů Yin Yang, u kterého však varují, že může vzhledem ke své struktuře náhodně selhat, jelikož nedokáže náhodným párováním binárních sekvencí dosáhnout požadovaného zastoupení všech částí binárního toku. Součástí knihovny je také naivní přístup kódování *BaseCodingAlgorithm* (zkráceně Base ve zbytku práce), který využívá nejjednoduššího mapování dvojic bitů na jeden nukleotid dle pevného mapování. Tento Base algoritmus tedy nepoužívá žádné metody, které by zajišťovaly dodržování biochemických restrikcí, a je tedy vhodný pro porovnání účinnosti s ostatními.

Knihovna Chamaeleo umožňuje různé přístupy porovnat pro dané typy dat a také implementuje metody korekce chybovosti Hamming a ReedSolomon. O těchto metodách je více informací v následujících podsekcích 3.8.1 a 3.8.2.

3.8.1 Hammingův kód

Hammingovy kódy jsou jedním z nejjednodušších a nejefektivnějších způsobů detekce a opravy chyb v binárních datech. Byly vyvinuty Richardem Hammingem v roce 1950. Hammingovy kódy mohou detekovat a opravit jednu chybu v libovolném kódovém slově a detekovat dvě chyby [31].

- **Redundantní bity**: Do původního datového bloku se přidají redundantní bity (paritní bity), které pomáhají detekovat a opravovat chyby. Počet redundantních bitů r potřebných pro kódování n bitů je dán vztahem: $2^r \geq n + r + 1$.
- **Pozice paritních bitů**: Paritní bity jsou v bitovém segmentu umístěny na pozicích, které jsou mocninami dvou (1, 2, 4, 8, ...).
- **Výpočet paritních bitů**: Každý paritní bit kontroluje určité bity v datovém bloku a je nastaven tak, aby celkový počet jedniček v kontrolované skupině byl sudý (parita, tedy logická 1) a pokud byl lichý (neparita, tedy logická 0).
- **Kontrola a oprava**: Při příjmu bitového segmentu upraveného Hammingovým kódem se na stejných pozicích paritní bity vyhodnotí podle stených pravidel, jako se vytvářely. Vyhodnocení rozdílů mezi přijatými paritními bity a znovu vypočtenými umožňuje určité množství chyb opravit.

3.8.2 ReedSolomonovy kódy

ReedSolomonovy kódy jsou víceúčelové chyby opravující kódy, které se široce používají například v digitálních komunikačních systémech nebo úložištích dat. Byly vyvinuty Irvingem Reedem a Gustavem Solomonem v roce 1960. Tyto kódy jsou schopné opravovat chyby vzniklé v celých blocích. Jedná se o zcela jiný revoluční přístup chyb opravujícího kódování (oproti Hamming kódu), který spojuje oblasti modulární aritmetiky a Lagrangeových polynomů. [32, 33].

- **Blokové kódy**: ReedSolomon kódy dělí bitový tok na bloky (segmenty), kde jsou bity shlukovány v symboly. Každý blok se skládá z datových symbolů a redundantních (paritních) symbolů. Obvykle se používají symboly o velikosti 8 bitů (1 bajt) za sebou.
- **Generování paritních symbolů**: Paritní symboly jsou generovány pomocí polynomiální aritmetiky nad konečným polem $GF(2^m)$ (Galoisův prostor, také konečný prostor s 2^m prvky [34]). Generační polynom se aplikuje na datové symboly a výsledkem jsou paritní symboly. Jinými

slovy symboly nesoucí infromaci vedou k vytvoření Lagrangeova generačního polynomu, ten se vyhodnotí navíc i v pozicích určených pro paritní symboly a tím se získají.

- **Detekce a oprava chyb**: Při dekódování se vypočítají syndromy, které indikují přítomnost a pozici chyb v bloku. Syndromy jsou analyzovány pomocí algoritmů, jako je Berlekamp-Massey nebo Euclidův algoritmus, pro nalezení chybových polynomů. Po nalezení chybových polynomů se určí pozice a hodnoty chyb, které se následně opraví.

Kapitola 4

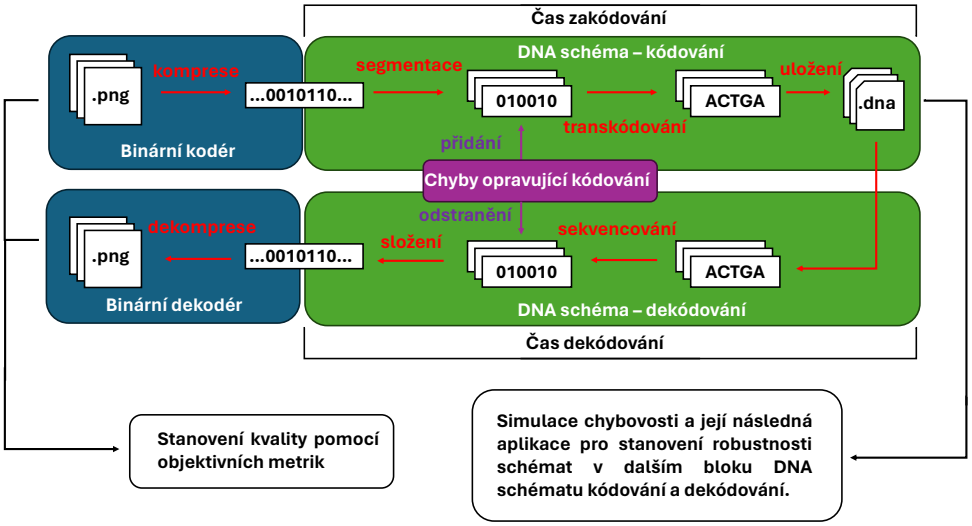
Praktická část

Tato kapitola se zaměřuje na aplikaci transkódovací architektury pro převod binárních dat do DNA řetězců. Tato architektura byla vybrána pro svou univerzálnost, která umožňuje kódovat jakýkoliv typ dat do DNA sekvencí. Efektivní komprese je dosažena výběrem vhodného binárního kodéru pro daná data, v tomto případě statická obrazová data.

Cílem této kapitoly je:

1. Komprimovat vybraná obrazová data (viz sekce 4.1) pomocí specifických obrazových kodérů (viz sekce 4.2).
2. Aplikovat vybraná transkódovací DNA schémata na vytvořené bitové toky (komprimované soubory) (viz sekce 4.3).
3. Ověřit, zda DNA schémata přináší fixní kompresní poměr nezávislý na použitém obrazovém kodéru a určit jaké jsou průměrné délky generovaných DNA segmentů (viz sekce 4.3).
4. Vypočítat objektivní metriky pro stanovení kvality obrazu a určit nejúčinnější bitový kodér na základě těchto výsledků (viz sekce 4.4).
5. Stanovit dodržování biochemických restrikcí pro všechna DNA schémata a simulovat chybovost způsobenou jejich nedodržením (viz sekce 4.5 a 4.6).
6. Použít výsledky simulací ke stanovení robustnosti schémat, porovnat je a stanovit jaké je v tomto ohledu nejúčinnější schéma (viz sekce 4.7).
7. Spočítat a graficky zobrazit výpočetní časy kódování a dekódování komprimovaných souborů jednotlivými transkódovacími DNA schématy (viz sekce 4.8).

Tímto způsobem je stylizována tato kapitola, aby bylo dosaženo kompletního systematického zhodnocení efektivity a robustnosti různých transkódovacích schémat pro ukládání obrazových dat do DNA.



Obrázek 4.1: Kompletní schéma kompresního procesu této práce: komprese binárním kodérem, transkódování binárního toku DNA schématem (s případným chyby opravujícím kódem), stanovení chybovosti a simulace.

4.1 Testovací obrazy

Skupina JPEG již uzavřela tzv. *Call for Proposals* (viz sekce 2.1) a tedy jejich datový set obrazů pro JPEG DNA už není k dispozici v době psaní této práce. Z tohoto důvodu bylo rozhodnuto v této práci použít jiný a velmi často používaný soubor referenčních bezeztrátových obrázků s plným rozsahem v oblasti stanovení účinnosti komprese a zkreslení obrazu přirozené scény. Konkrétně **Kodak Lossless True Color Image Suite** (známé též jako *kodim* obrázky) dostupné z [35], které byly k podobným účelům vyhodnocení účinnosti komprese a způsobeného obrazového zkreslení použity již nesčetněkrát.

(a) : kodim03.png **(b) :** kodim21.png **(c) :** kodim23.png

Obrázek 4.2: Příklady obrázků v databázi KODAK. Dostupné z [35].

4.2 Obrazové kodéry a nastavení

Při výběru vhodných a účinných obrazových kodérů pro účely této práce byly zvoleny tři ze skupiny JPEG, jelikož jsou kodéry této skupiny jedny z nejznámějších, nejpoužívanějších a nejcitovanějších v oblasti efektivní komprese obrazu.

Konkrétně byly vybrány následující kodéry:

- **JPEG** (Legacy) jelikož se jedná o nejrozšířenější a nejpoužívanější formát pro kompresi obrazových dat. Je podporován téměř všemi prohlížeči, operačními systémy a editory. Díky tomu je tento formát vhodný pro zajištění široké kompatibility a snadného sdílení obrazových souborů [29].
- **JPEG 2000**, který je vylepšenou verzí formátu JPEG používající pokročilé kompresní techniky založené na vlnkové transformaci. Díky tomu poskytuje lepší kvalitu obrazu při stejných nebo nižších bitových rychlostech než standardní JPEG [36]. Nerozšíril se sice u veřejnosti tak, jak se očekávalo, ale velice se uplatnil například v medicíně, kde je obrazová data (např. rentgenové snímky) potřeba uchovávat po delší dobu [37].
- **JPEG XL**, který představuje jeden z nejnovějších kodérů z rodiny JPEG dosahující dalšího zlepšení kompresních poměrů [38]. Je navíc navržen tak, aby byl zpětně kompatibilní s klasickým JPEG formátem, což usnadňuje jeho adopci.

Konkrétní použité implementace těchto kodérů jsou pro JPEG a JPEG 2000 z integrované knihovny **SIPS** (*Scriptable image processing system*) systémů MacOS [39]. Pro JPEG XL je pak použita knihovna **libjxl** dostupná z [40].

Obrázky KODAK jsou pomocí skriptů *b1_script_jpeg.sh*, *b2_script_2000.sh* a *b3_script_xl.sh* (dostupných v přiložených souborech) konvertovány do příslušných formátů s proměnným koeficientem kvality (*-q*) v rozsahu od 10 do 100 s krokem 5, což představuje celkem 19 úrovní. Tento rozsah byl zvolen, aby byla zajištěna dostatečná jemnost pro zachycení tvaru křivek objektivních metrik (viz sekce 4.4). Navíc je tento počet kompresních úrovní porovnatelný s jinými prácemi autorů v této oblasti komprese dat do DNA [17, 41]. Pomocí [42] je případně možno převést vytvořené skripty *b1_script_jpeg.sh* a *b2_script_2000.sh* na knihovnu **ImageMagick** [43] dostupnou pro libovolný unixový systém (například Linux) a nelimitovat se pouze na systémy MacOS.

4.3 Transkódovací DNA schémata

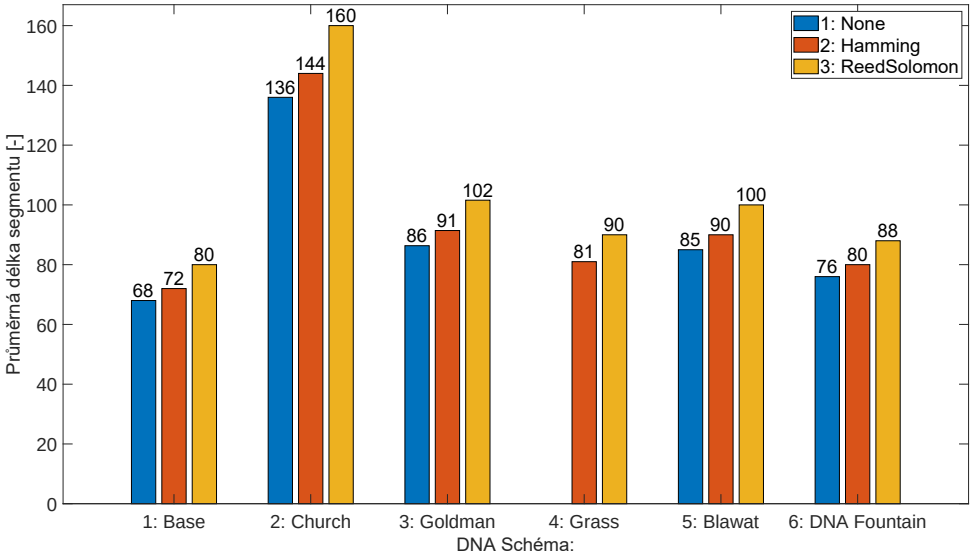
Vzhledem k tomu, že volně dostupná knihovna Chamaeleo (viz sekce 3.8) obsahuje největší množství schémat určených pro transkódování libovolných binárních dat do DNA sekvencí, byla zvolena i pro potřeby této práce.

Vybraná transkódovací schémata zahrnovala **Base** (výchozí naivní schéma, nedodržující žádná biochemická omezení DNA, které slouží spíše jen pro

porovnání účinnosti s jinými), **Church**, **Goldman**, **Grass**, **Blawat** a **DNA-Fountain** (se zvolenou redundancí 50%, aby bylo zajištěno dostatečné zastoupení všech sekvencí binárního toku, viz sekce 3.5). Vzhledem k tomu, že proces dopředného a zpětného transkódování dat do DNA je poměrně náchylný na chybovost (viz sekce 2.5 a 2.7.2), byly navíc použity dvě metody korekce chybovosti **Hamming** a **ReedSolomon**, které jsou v této knihovně také zahrnuty.

Schéma autorů knihovny Chamaeleo (**Yin-Yan** [44]) použito nebylo, jelikož se potvrdilo, že náhodně selhává a jeho výpočetní čas exponenciálně roste se zvyšující se velikostí komprimovaných obrazů. U větších souborů navíc docházelo k častějšímu selhání, což snižovalo statistickou významnost výsledků při vyšších kvalitách.

Všechna vybraná schémata pracovala s bitovým tokem obrazových souborů po sekvencích, jejichž délka byla pro všechny zvolena stejně, na 120 bitů. Tato hodnota byla zvolena tak, aby se u použitých DNA schémat zajistilo, že jakýkoliv výsledný DNA segment bude omezen na maximální dovolenou délku 300 nukleotidů (viz sekce 2.5). V této práci bylo cíleno na kratší segmenty okolo 80 nukleotidů podle vzoru autorů [17] (viz sekce 3.6), protože při reálné syntéze umělých DNA struktur je potřeba ponechat část nukleotidů pro tzv. primery, kterými se syntéza započíná a ukončuje. Navíc je při těchto kratších délkách dostatek prostoru pro splnění požadavků na rozeznatelnost umělého řetězce (viz bod 6 sekce 2.6), který je však předmětem standardizace skupiny JPEG DNA (viz sekce 2.1), a proto se tímto bodem zatím skoro žádný z autorů v této oblasti nezabýval.

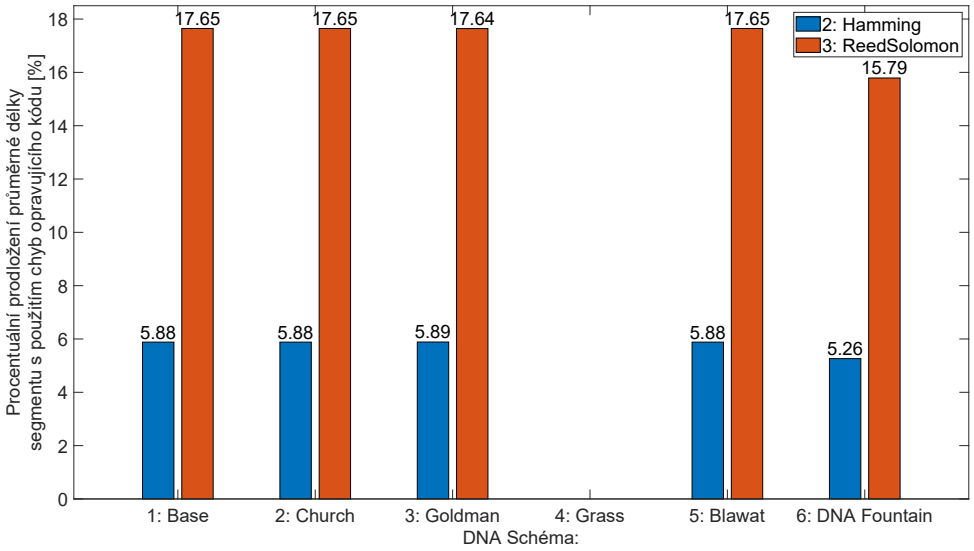


Obrázek 4.3: Průměrná délka jednoho segmentu všech schémat (stejné pro JPEG, JPEG 2000 i JPEG XL).

Data zobrazená na obrázku 4.3 byla získána přeložením všech bitových souborů (`.jpeg`, `.jp2` a `.jxl`) všech úrovní kvality q příslušným DNA schématem do textových souborů `.dna`. Při výpočtu průměrných délek sekvencí,

které DNA schémata vytvářela, se ukázalo, že jsou vždy stejné v rámci celého datasetu komprimovaných obrázků, tedy nezávislé na dané kvalitě q. Délky sekvencí byly tedy zprůměrovány přes počet KODAK komprimovaných obrázků jedné hodnoty kvality q, a poté i přes těchto 19 hodnot q. Tím se získaly průměrné délky sekvencí DNA schémat pro tři kodeky: JPEG, JPEG 2000 a JPEG XL. I tyto výsledky se mezi obrazovými kodeky ukázaly být shodné, takže na obrázku 4.3 je vykreslen pouze JPEG XL obrazový kodér.

Z obrázku 4.3 je patrné, že všechna schémata s daným nastavením délky binárního segmentu skutečně vytvářejí `.dna` soubory o sekvencích s délkou kratší než 300 nukleotidů, a tento výsledek platí pro všechny kodeky (JPEG, JPEG 2000 i JPEG XL). Z grafu je také zřejmé, že data ke čtvrtému schématu (Grass) nejsou dostupná pro případ bez použití chybově opravného kódu, což je způsobeno implementační chybou knihovny Chamaeleo. Tato chyba způsobovala, že schéma nebylo možné úspěšně použít bez dodatečného kódování opravného kódu Hamming nebo ReedSolomon. Vzhledem k tomu, že ukládání dat do DNA je zejména určeno a testováno pro dlouhodobou archivaci a samotný proces syntézy, uchování a sekvencování DNA je velmi náchylný k chybám, je vhodné maximalizovat robustnost takového *end-to-end* systému a použít daná schémata s jedním z chybově opravujících kódů.



Obrázek 4.4: Prodloužení délky segmentu s použiím chyb opravujícího kódu (stejné pro JPEG, JPEG 2000 i JPEG XL).

Data z obrázku 4.4 byla získána určením procentuálního nárůstu průměrné délky DNA sekvence z obrázku 4.3 pro případy s použitím Hamming kódu a ReedSolomon kódu vůči výsledku bez použití dodatkového chyb opravujícího kódování (v obrázku 4.3 označeno jako *None*). Tyto výsledky opět platily vždy stejně bez rozdílu pro všechny tři obrazové kodéry. Z obrázku 4.4 je vidět, že Hammingův kód prodlužoval délku sekvencí prakticky stejně i pro všechna schémata tedy okolo 5,9% o něco méně pak pro schéma DNA Fountain. ReedSolomon pak o cca 17,7%. Tyto hodnoty vyplývají z interního nastavení těchto chyb opravujících kódů v knihovně Chamaeleo [30].

Hammingův kód pracoval se stejně dlouhými bitovými segmenty jako DNA kodér samotný (tedy se 120 bity). Počet paritních bitů r lze tedy nalézt podle nerovnosti:

$$2^r \geq 120 + r + 1 \tag{4.1}$$

To je splněno pro nejmenší $r = 7$ a tedy procentuální prodloužení sekvence P je pak:

$$P = \left(\frac{N-n}{n}\right) \times 100 = \left(\frac{127-120}{120}\right) \times 100 \approx 5.83\% \tag{4.2}$$

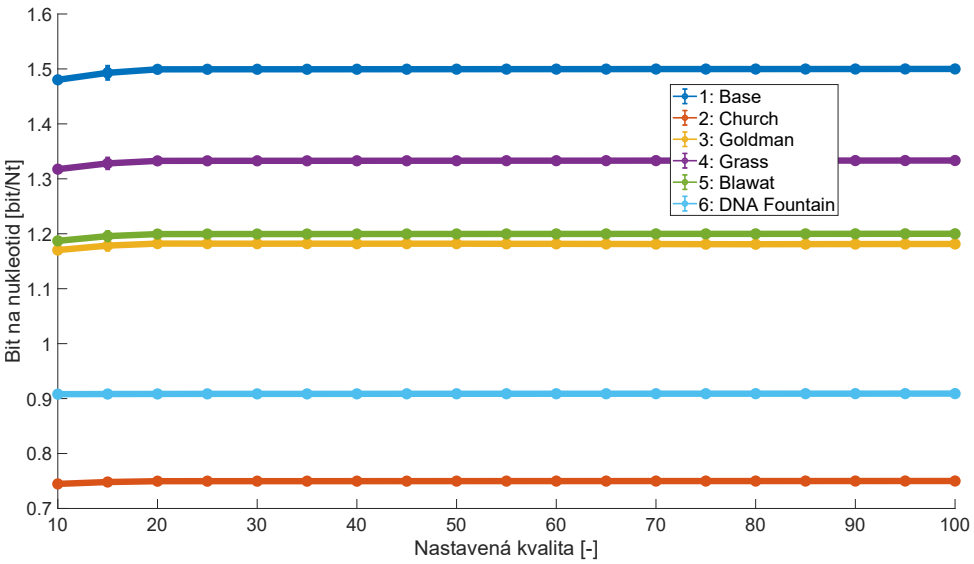
kde N je celkový počet bitů (původních n + paritních p).

ReedSolomon kód pracoval s o něco delšími segmenty (konkrétně 136 bity). Ty převáděl v symboly po 8 bitech, tedy na celkových 17 bajtů a fixně tvořil vždy 3 paritní symboly (bajty). Prodloužení sekvence P lze tedy stanovit pomocí:

$$P = \left(\frac{M-m}{m}\right) \times 100 = \left(\frac{20-17}{17}\right) \times 100 \approx 17.65\% \tag{4.3}$$

kde M je celkový počet bajtů/symbolů (původních m + paritních p).

Menší odchylky od skutečných výsledků jsou dány velikostmi posledních segmentů `.dna` souborů.



Obrázek 4.5: Velikost vstupního obrazového souboru (JPEG, JPEG 2000 nebo JPEG XL) v bitech v poměru ku počtu nukleotidů výstupního DNA souboru daného schématu v závisloti na parametru kvality q bitového kodéru (pro Reed-Solomon kód).

Výsledky zobrazené na obrázku 4.5 byly získány podělením velikosti vstupního souboru v bitech (`.jpeg`, `.jp2` a `.jxl`) počtem nukleotidů výstupního `.dna` souboru vytvořeného vybraným DNA schématem. Výsledky byly zprůměrovány pro jednotlivé hodnoty kvality q přes všech 24 obrázků z KODAK a ke každému průměru byla vypočítána i standardní odchylka.

Na obrázku 4.5 je tento průměrný poměr vykreslen i s odchylkami v závislosti na nastavené kvalitě q. Při výpočtech bylo ověřeno, že výsledky nezávisí na použitém bitovém kodéru (JPEG, JPEG 2000 nebo JPEG XL).

Na obrázku 4.5 je tedy zobrazen případ s použitím JPEG XL kodéru a dodatečným chybově opravným kódem ReedSolomon. Minimální odchylka nastala pouze u JPEG mezi kvalitami 40 a 45, kde došlo ke zvýšení poměru zhruba o 8%. Pravděpodobně v důsledku přechodu použité implementace kodéru JPEG do jiného režimu mezi těmito kvalitami, a tedy použitím jiných kvantizačních tabulek. Rozdíly v rámci jednoho schématu jsou naprosto minimální a závislé pouze na zbytku podílu velikosti vstupního souboru v bitech vůči výchozímu bitovému segmentu (120 bitů), se kterým schémata pracují.

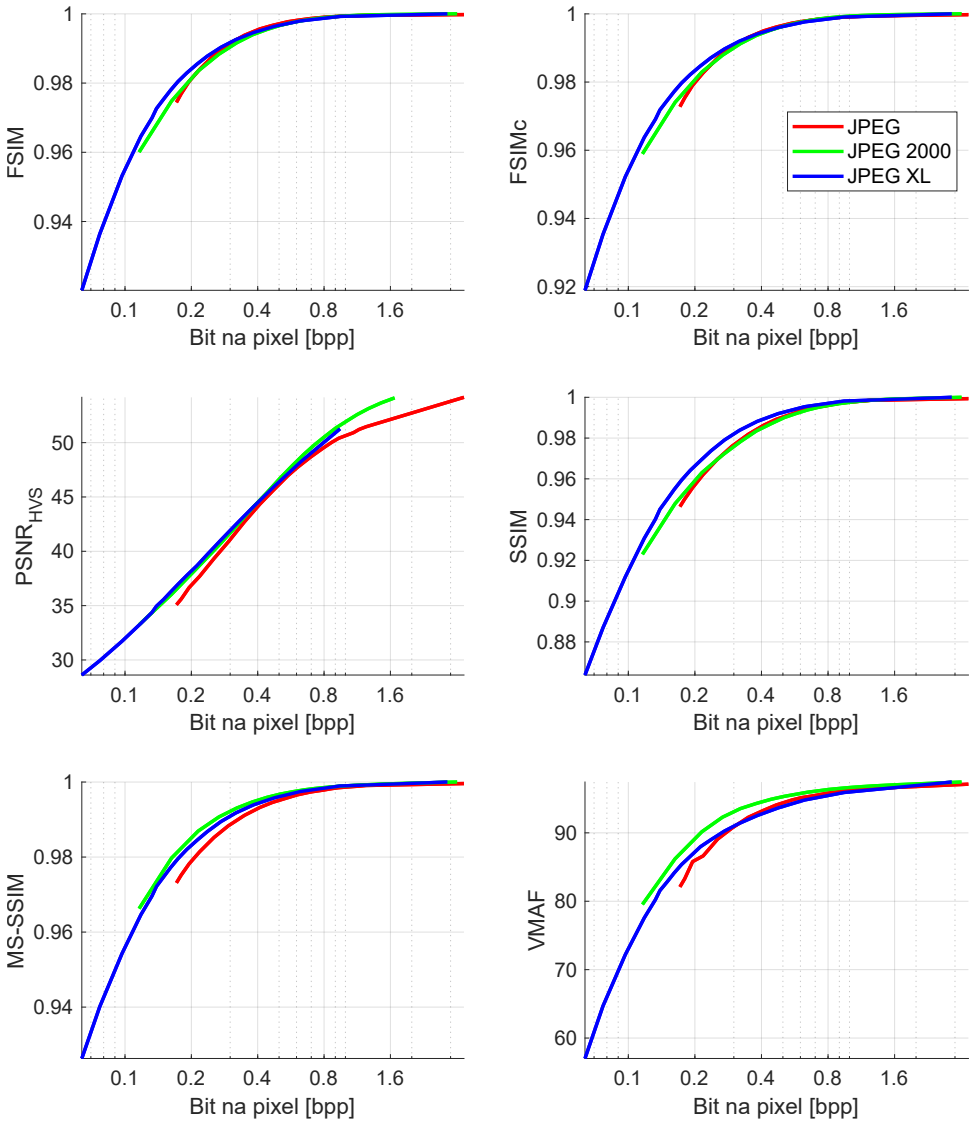
Z obrázku je patrné, že kompresní poměr (počet bitů na nukleotid) je prakticky neměnný pro všechna nastavení kvality v rámci celého schématu. DNA schéma tedy samostatně přináší prakticky fixní kompresní poměr nezávislý na obrazovém kodeku a úrovni kvality q. Tento fakt usnadňuje úkol nalezení vhodného *end-to-end* systému pro ukládání obrazových dat do DNA, protože lze nezávisle hledat vhodný bitový obrazový kodér a vhodné transkódující DNA schéma. Tento poznatek je důležitý pro následující části práce.

4.4 Metriky stanovení kvality obrazu

Pro stanovení vhodného obrazového kodeku z RD křivek (z angl. *rate-distortion*, česky *míra zkreslení*) byly pro účely této práce vybrány metriky:

- **FSIM** a $\mathbf{FSIM_C}$ - použitá implementace dostupná z [45],
- $\mathbf{PSNR_{HVS}}$, **SSIM**, **MS-SSIM** a **VMAF** - použitá implementace dostupná z [46].

Tyto metriky byly vybrány z důvodu doporučení skupiny JPEG pro odvětví JPEG DNA [8]. Mnohé práce navíc popisují dobrou korelaci mezi těmito vybranými metrikami a subjektivním hodnocením pozorovatelů [47, 48, 49].



Obrázek 4.6: RD křivky všech metrik (FSIM, FSIM_C, PSNR_HVS, SSIM, MS-SSIM A VMAF) pro bitové kodéry JPEG (červeně), JPEG 2000 (zeleně) a JPEG XL (modře). Závislost výsledku metriky dané úrovně komprese bitového kodéru na velikosti výstupního komprimovaného souboru (v bitech) vyděleným počtem pixelů obrázku.

Na obrázku 4.6 jsou postupně vykresleny výsledné RD křivky metrik **FSIM**, $\mathbf{FSIM_C}$, $\mathbf{PSNR_{HVS}}$, **SSIM**, **MS-SSIM** a **VMAF** s logaritmickým měřítkem osy x (*bpp - bit na pixel*) pro zvýraznění rozdílů. Tyto křivky byly získány vypočtením vybraných metrik mezi komprimovanými obrázky bitových kodérů (JPEG, JPEG 2000 nebo JPEG XL) s daným nastavením kvality komprese q a původními KODAK obrázky. Výsledky byly zprůměrovány pro jednotlivé hodnoty kvality přes 24 obrázků z KODAK. Standardní odchylky byly velmi malé a spíše ubíraly na přehlednosti vykreslení 4.6, a proto byly vynechány.

Z výsledků obrázku 4.6 je zřejmé, že JPEG XL dosahuje nejvyšších hodnot

všech metrik krom případů MS-SSIM a VMAF, kde je lepší JPEG 2000.

RD křivky PSNR_{HVS} lze navíc porovnat pomocí *Bjøntegaardovy metriky*, která dokáže stanovit průměrný zisk PSNR_{HVS} a průměrnou úsporu bitratu pro křivky JPEG XL a JPEG 2000 vůči křivce JPEG. Použitá implementace této metriky je dostupná z odkazu [50].

Bjontengaardova metrika	Zisk $PSNR_{HVS}$	Úspora bitrate
JPEG 2000	1,10	7,44 %
JPEG XL	1,32	11,64 %

Tabulka 4.1: Bjøntegaardova metrika porovnání průměrného zisku PSNR_{HVS} a úspory bitratu pro JPEG 2000 a JPEG XL vůči JPEG.

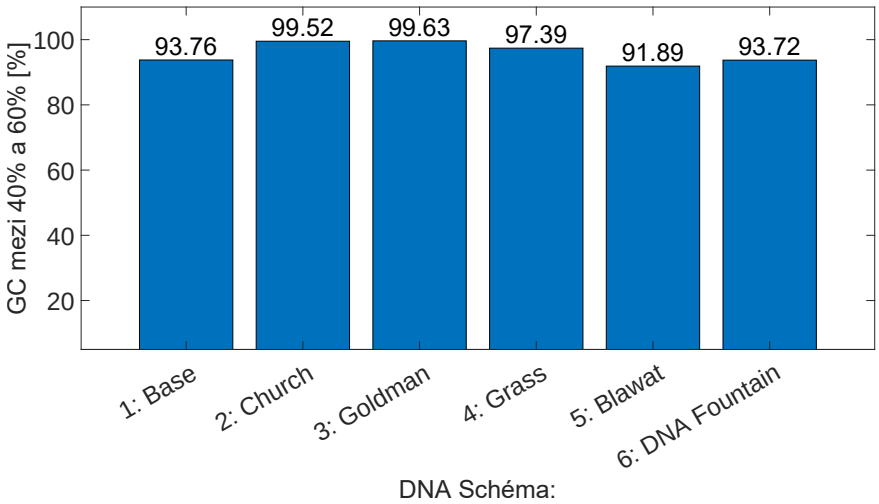
Dle tabulky 4.1 je JPEG XL i tedy z hlediska Bjøntegaardovy metriky nejúčinnějším kodérem z trojice JPEG, JPEG 2000 a JPEG XL.

4.5 Biochemická omezení DNA řetězců - porovnání DNA schémat

V této podkapitole je vyhodnoceno, jak moc schémata dodržují nebo porušují biochemická omezení (viz 2.5) a to z pohledu správného rozmezí obsahu GC, přítomnosti homopolymerů a opakujících se vzorů v sekvencích.

4.5.1 GC obsah

Nejdůležitějším faktorem pro zajištění stabilní molekuly, a tedy vůbec možnosti ukládání dat do DNA sekvencí za účelem archivace, je vyvážený GC obsah mezi 40% a 60% (viz sekce 2.5).



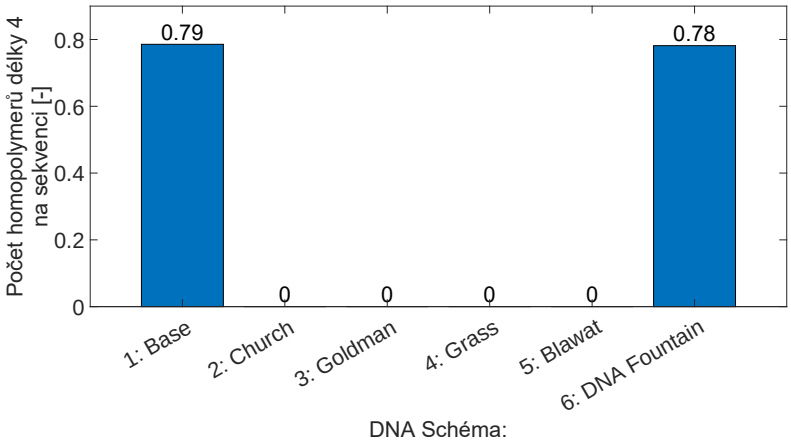
Obrázek 4.7: Průměrné splnění podmínky GC obsahu DNA schématem.

Výsledky z obrázku 4.7 byly získány podle vztahu 2.1 pro každou sekvenci vytvořenou DNA schématem. Hodnoty byly zprůměrovány nejprve přes počet sekvencí v každém souboru a poté přes všechny soubory a všechny úrovně kvality. Vlastní implementace je k dispozici v přiložených souborech v Matlab skriptu s názvem *m5_biochem_constrain.m*, konkrétně funkce *calculateGCContentAndHomopolymers*(). Bylo ověřeno, že se výsledky v závislosti na velikosti souboru resp. na úrovni kvality neměnily.

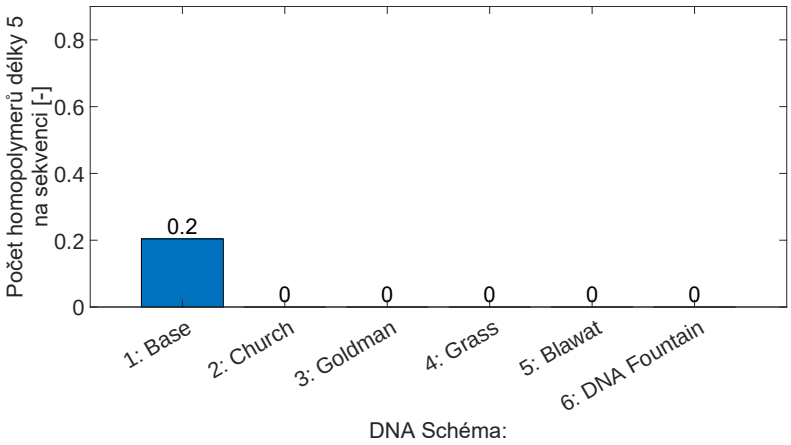
Z výsledků na obrázku 4.7 je patrné, že téměř 100% splnění GC podmínky dosahují pouze schémata **Church** a **Goldman**. Naopak schémata **Base**, **Blawat** a **DNA Fountain** splňují tuto podmínku nejméně, a tedy pravděpodobně nevytvářejí stabilní molekuly. U všech schémat kromě Base (včetně Blawat a DNA Fountain) by pravděpodobně bylo možné dosáhnout lepších výsledků při použití delších segmentů binárního toku než je nastavených 120 bitů (viz sekce 3.1, 3.2, 3.3, 3.4 a 3.5 v kapitole 3). Schéma Base neobsahuje žádné mechanismy pro zajištění GC podmínky, takže jeho výsledky jsou obecně platné.

4.5.2 Homopolymery

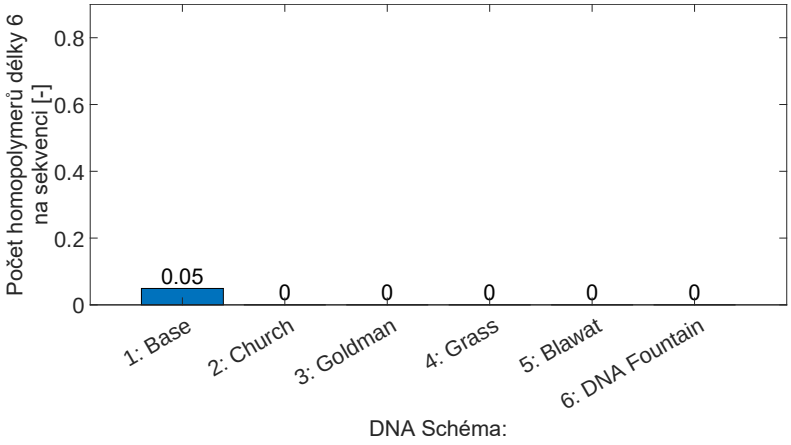
Dalším důležitým faktorem je výskyt homopolymerů v řetězcích, které způsobují nejistotu při jejich sekvencování, protože s dosavadními metodami čtení umělých molekul DNA nelze s jistotou zaručit počet opakujících se homopolymerů pro délky čtyři a více [17].



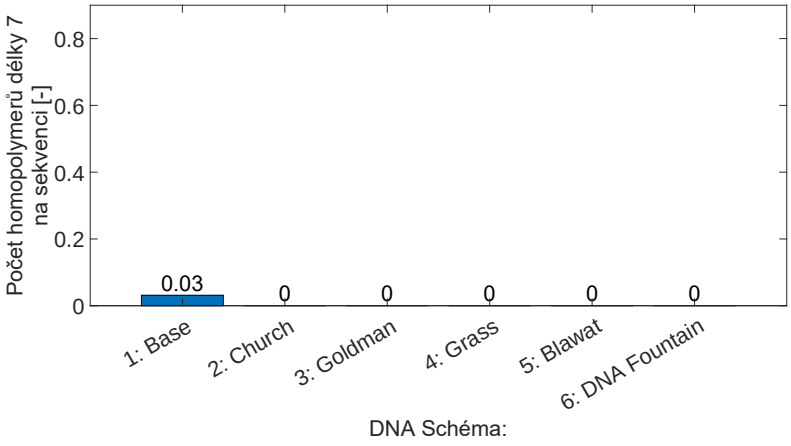
Obrázek 4.8: Průměrný počet homopolymerů délky čtyři na sekvenci všech DNA schémat.



Obrázek 4.9: Průměrný počet homopolymerů délky pět na sekvenci všech DNA schémat.



Obrázek 4.10: Průměrný počet homopolymerů délky šest na sekvenci všech DNA schémat.



Obrázek 4.11: Průměrný počet homopolymerů délky sedm a více na sekvenci všech DNA schémat.

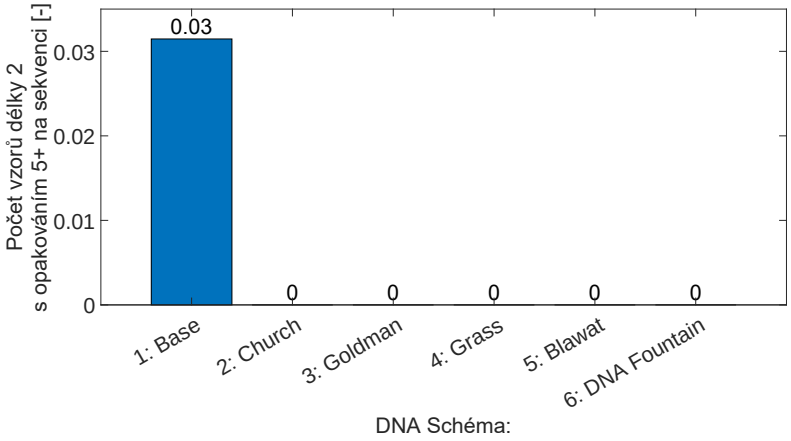
Vlastní implementace, která počítá četnosti homopolymerů různých délek v sekvencích příslušného `.dna` souboru daného DNA schématu, je k dispozici v přiložených souborech v Matlab skriptu s názvem `m5_biochem_constrain.m`, konkrétně ve funkci `countHomopolymers()`.

Výsledky zobrazené na obrázcích 4.8, 4.9, 4.10 a 4.11 byly získány vydělením četnosti sledované délky homopolymeru počtem sekvencí v daném `.dna` souboru a následným průměrováním přes všechny soubory dané kvality q. Ověřením se potvrdilo, že velikost souboru (nastavená kvalita q) ani použitý binární kodér nemají vliv na výsledek.

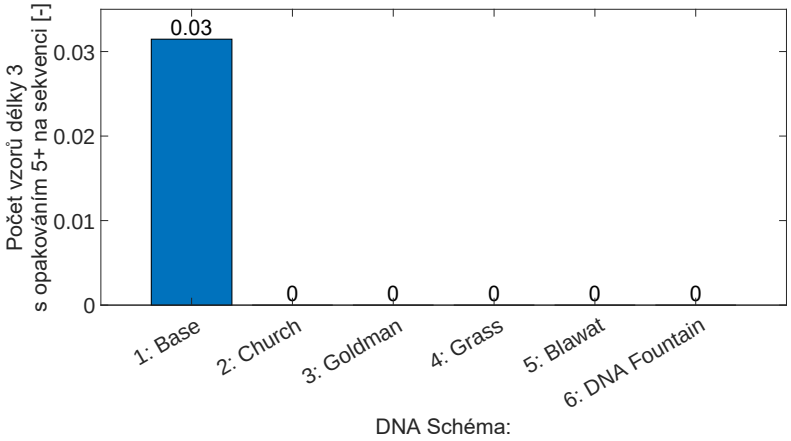
Podle obrázků 4.8, 4.9, 4.10 a 4.11 schémata **Church**, **Goldman**, **Grass** a **Blawat** nevytvářejí žádné nepovolené homopolymery ve svých sekvencích. Schéma **DNA Fountain** vytváří v průměru 0,78 homopolymeru délky čtyři na jednu sekvenci, zatímco schéma **Base** vytváří homopolymery všech pozorovaných délek.

4.5.3 Opakování vzoru

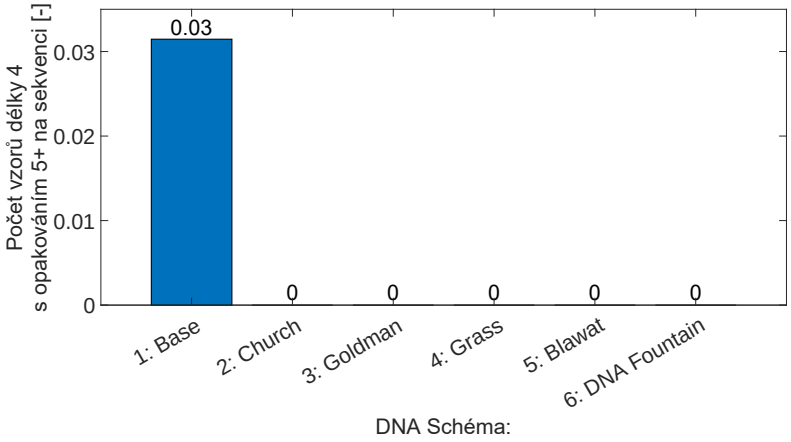
Poslední nežádoucí vlastností DNA řetězců, kterou skupina JPEG DNA popisuje, je přítomnost opakovaného vzoru nukleotidů (viz sekce 2.5). To je problematické, jelikož při syntéze řetězce s opakovaným vzorem délky dva až pět nukleotidů čtyři a vícekrát za sebou může vézt k nežádoucímu opakování takové části sekvence, než se v řetězci skutečně má nacházet. Jinými slovy nelze zaručit, že syntetizovaný řetězec obsahuje skutečně jen ty nukleotidy, které má, protože se repetetivní části mohou duplikovat vícekrát [17, 28, 18].



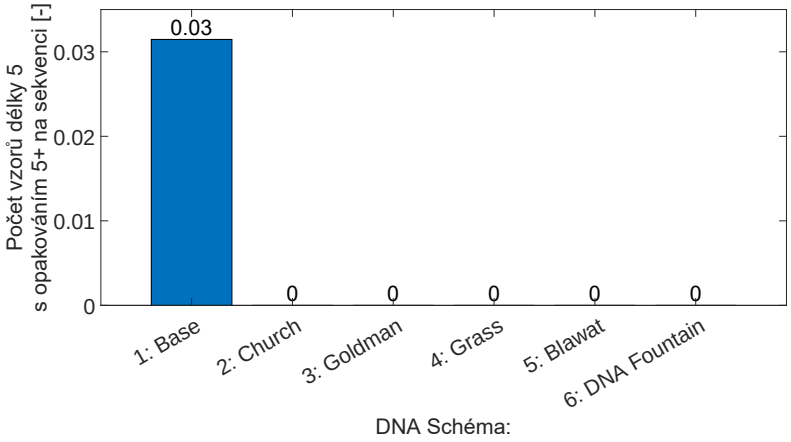
Obrázek 4.12: Průměrný počet opakování vzorů délky dva s opakováním pět a vícekrát za sebou na sekvenci pro všechna schémata.



Obrázek 4.13: Průměrný počet opakování vzorů délky tři s opakováním pět a vícekrát za sebou na sekvenci pro všechna schémata.



Obrázek 4.14: Průměrný počet opakování vzorů délky čtyři s opakováním pět a vícekrát za sebou na sekvenci pro všechna schémata.



Obrázek 4.15: Průměrný počet opakování vzorů délky pět s opakováním pět a vícekrát za sebou na sekvenci pro všechna schémata.

Vlastní implementace, která počítá výsledky obrázků 4.12, 4.13, 4.14 a 4.15 je k dispozici v přiložených souborech v Matlab skriptu s názvem *m5_biochem_constrain.m*, konkrétně funkce *countRepeatingPatterns*().

Podle obrázků 4.12, 4.13, 4.14 a 4.15 žádná schémata krom **Base** nevytvářejí opakujícíc se vzory žádných pozorovaných délek.

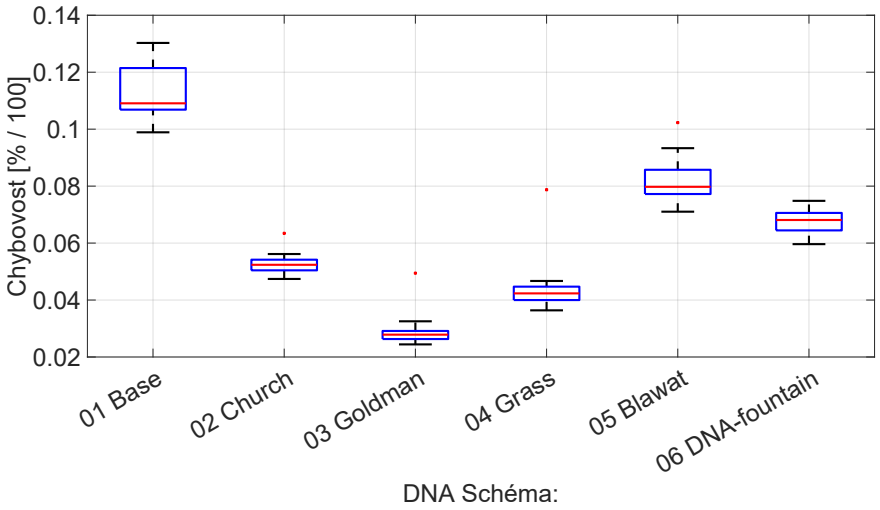
4.6 Simulace chybovosti syntézy, kapsulace a sekvencování DNA řetězců - MESA MOSLA

Jediným volně dostupným nástrojem, který dokáže simulovat chybovost syntézy, zapouzdření, uchování a sekvencování vlastních DNA řetězců, je MESA MOSLA od týmů z Marburské univerzity, univerzity v Giessenu a výzkumného institutu LOEWE v Německu [16]. Jeho implementace je dostupná na GitHubu [51] nebo jako online webová aplikace [52]. Tento nástroj doporučuje také skupina JPEG DNA na svých workshopech [5].

Tento nástroj sdružuje výsledky reálných pokusů syntézy, kapsulace a sekvencování DNA od mnoha světových týmů, které průběžně rozšiřují jeho databázi chybových sekvencí pro různá nastavení. Biochemické restrikce popsané v sekci 2.5 jsou pouze jedním z dobře zdokumentovaných zdrojů chyb DNA sekvencí. K chybám může docházet v částech procesu ukládání dat do DNA (viz sekce 2.4), i když jsou tyto restrikce splněny [16].

Aby byly výsledky simulace replikovatelné a porovnatelné, byly použity následující parametry:

- Syntetizační metoda: **ErrASE** - Tato metoda byla vybrána, protože dle výsledků nejvíce minimalizuje chybovost při syntéze umělých DNA sekvencí [53].
- Počet PCR cyklů: **30** - Tento počet cyklů je běžně používán v laboratorní praxi a zajišťuje dostatečnou amplifikaci DNA, což snižuje chybovost sekvencování [53].
- Počet měsíců kapsulace: **24** - Dvouletá kapsulace simuluje dlouhodobé skladování DNA, což umožňuje studium dlouhodobých účinků na stabilitu a chybovost [16].
- Typ PCR: **Taq** - Enzym Taq DNA polymeráza je standardně používán v PCR a jeho chybovost je dobře zdokumentovaná [53].
- Médium uchování: **E. coli** - Použití E. coli jako média pro uchování DNA je běžné a umožňuje simulovat realistické podmínky uchování [17].
- Zbytek parametrů byl ponechán ve výchozím nastavení, aby byly výsledky konzistentní a srovnatelné s jinými studiemi [52].



Obrázek 4.16: Výsledná chybovost nukleotidů všech DNA schémat s vykresleným prvním a třetím kvartilem s červenou čarou značící medián výsledků chybovostí daného schématu - výsledek vyhodnocení pomocí simulačního programu MESA MOSLA [52].

Výsledky zobrazené na obrázku 4.16 byly získány nahráním vytvořených `.dna` souborů do online instance simulátoru MESA MOSLA [52]. Tento simulátor, při daném nastavení, vrátil textový soubor ve stejném formátu jako původní `.dna` soubor, kde příslušné pozice nukleotidů byly nahrazeny symbolem pravděpodobnosti chyby (chybovosti) ve formátu ASCII o základu 33 [54].

Na obrázku 4.16 jsou vykresleny první a třetí kvartily chybovostí na nukleotid pro všechna schémata standardního vykreslení pomocí funkce *boxplot()* z Matlabu, přičemž červená čára značí medián chybovosti daného schématu. Tyto výsledky dobře odpovídají vlastnímu stanovení dodržování biochemických omezení (viz sekce 4.5).

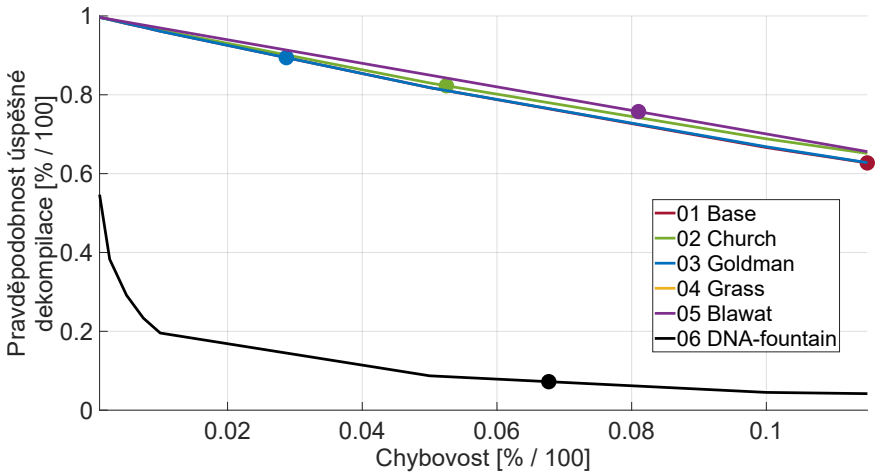
Schéma:	1. Base	2. Church	3. Goldman	4. Grass	5. Blawat	6. DNA Fountain
Průměrná chybovost [%]	11,50	5,25	2,87	4,36	8,10	6,77

Tabulka 4.2: Průměrná chybovost všech DNA schémat na nukleotid (průměrný výsledku z obrázku 4.16).

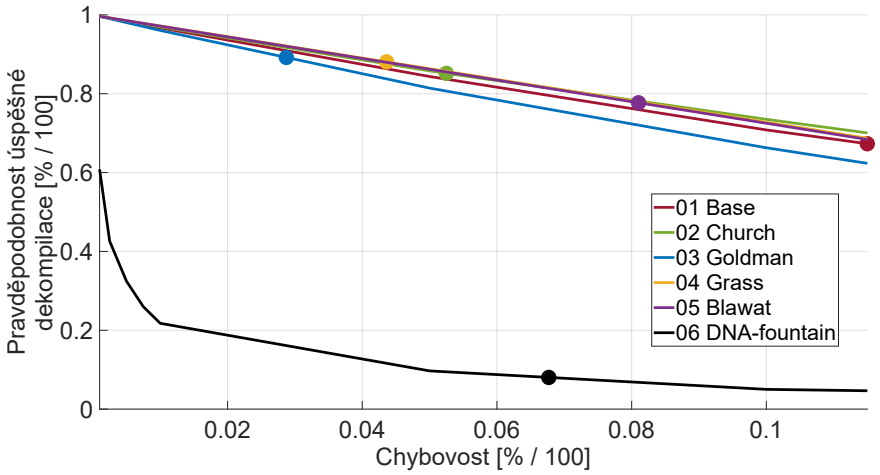
V tabulce 4.2 jsou celkové průměrné chybovosti nukleotidů schémat z obrázku 4.16. Tyto hodnoty jsou v následující sekci použity pro stanovení robustnosti schémat v závislosti na rozmezí chybovosti.

4.7 Porovnání robustnosti schémat v závislosti na chybovosti

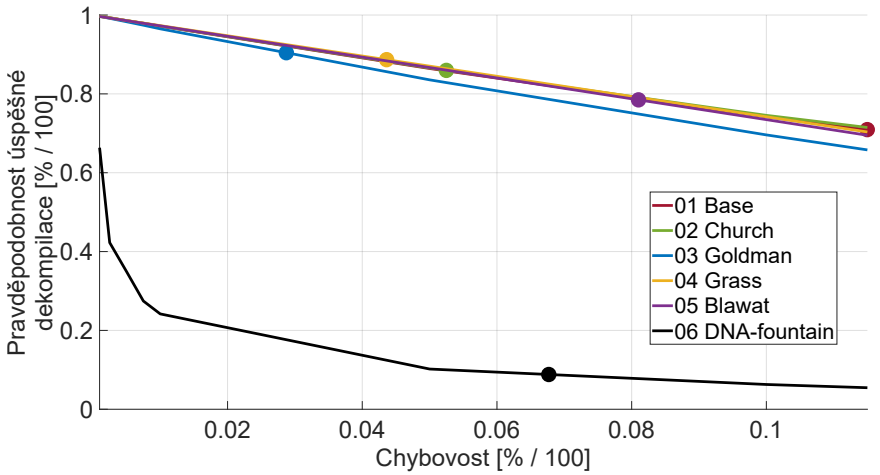
Za použití knihovny Chamaeleo [30] lze nyní stanovit robustnost kódů vytvořených jednotlivými schématy. Možné typy chyb jsou mutace (přeměna), vložení a vyjmutí nukleotidu a podle výsledků simulací MESA MOSLA jsou všechny tyto chyby stejně pravděpodobné. Tento závěr je také podpořen výsledky jiných autorů [17, 23].



Obrázek 4.17: Robustnost schémat bez použití chyb opravujícího kódu se zýrazněnými body konkrétně vykazovaných chybovostí schémat z tabulky 4.2.



Obrázek 4.18: Robustnost schémat s použitít Hammingova chyb opravujícího kódu se zýrazněnými body konkrétně vykazovaných chybovostí schémat z tabulky 4.2.



Obrázek 4.19: Robustnost schémat s použitít ReedSolomon chyb opravujícího kódu se zýrazněnými body konkrétně vykazovaných chybovostí schémat z tabulky 4.2.

Implementace výpočtu průběhů zobrazených na obrázcích 4.17, 4.18 a 4.19 je k dispozici v přiložených souborech v Python skriptu s názvem *p2_robustness_eval.py*. Pro jednotlivá schémata byl stanoven rozsah chybovostí podle tabulky 4.2, konkrétně od 0,1% do 11,5% v deseti hodnotách, aby byl dostatečně zachycen charakter křivek. Na komprimované binární soubory byl případně aplikován chyby opravující kód Hamming či ReedSolomon. Následně byla provedena komprese DNA schématem. Na vytvořený `.dna` soubor byla aplikována daná hodnota chybovosti jakožto pravděpodobnost mutace, inserce a delece nukleotidu. Po aplikaci chybovosti byl `.dna` soubor dekomprimován a případně opraven pomocí Hamming či ReedSolomon kódu. Výsledný bitový tok byl po bitových segmentech porovnán s původním. Délka shodných částí bitového toku vůči celku jsou pak výsledky zobrazené na obrázcích 4.17, 4.18 a 4.19. Zvýrazněné body představují interpolaci hodnot z tabulky 4.2 do průběhů získaných úspěšností dekomprese.

Z obrázků 4.17, 4.18 a 4.19 lze postupně vidět výsledky robustnosti pro případ bez chybově opravného kódu, pro Hammingův kód a pro kód ReedSolomon. U všech schémat, krom DNA Fountain, je závislost lineární a použití chybově opravných kódů zlepšuje robustnost o 2,3% pro Hammingův kód a 7,8% pro kód ReedSolomon (průměrně pro všechna schémata). U DNA Fountain chybově opravné kódy zlepšují robustnost procentuálně stejně, ale vzhledem k charakteru křivky, která vykazuje velmi rychlý spád, není tento přínos příliš významný.

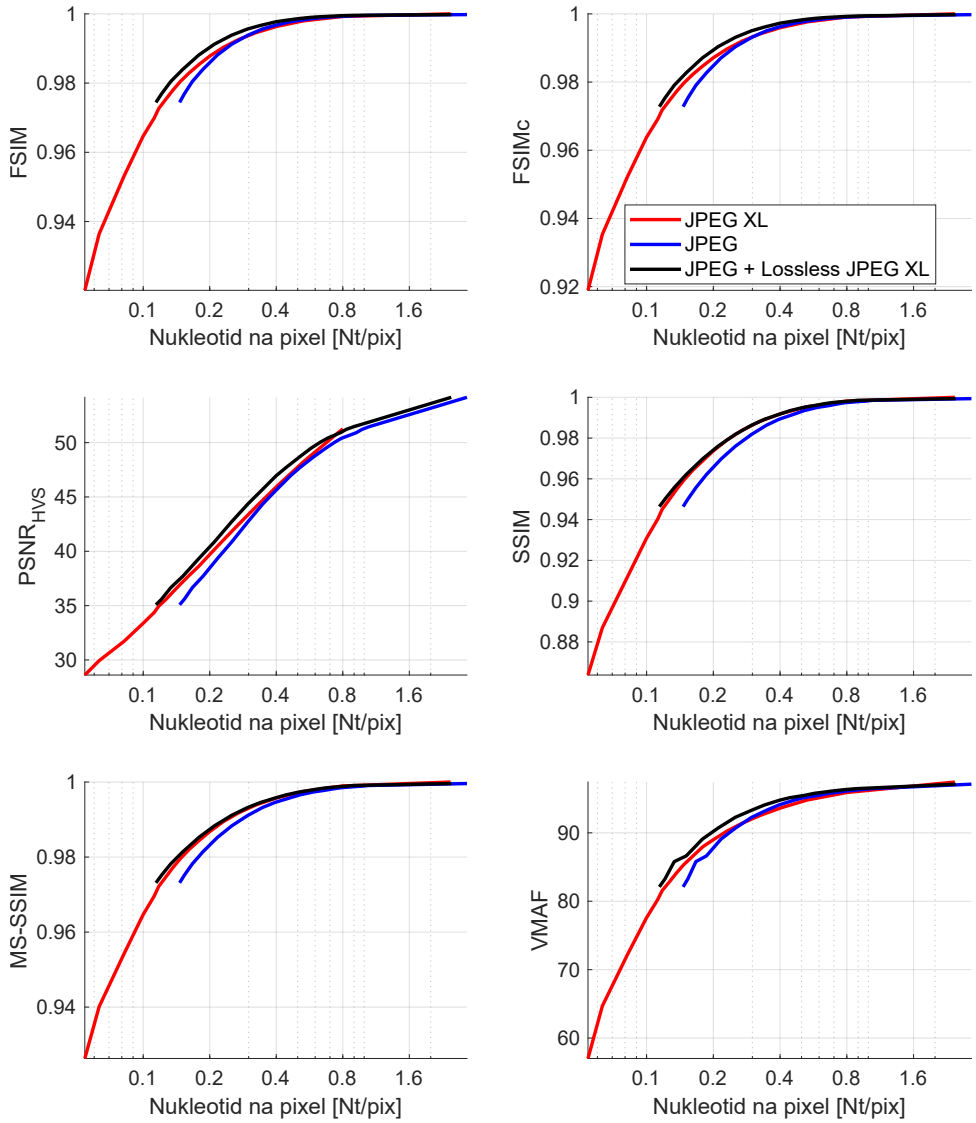
Schéma:		1. Base	2. Church	3. Goldman
Pravděpodobnost úspěšné dekompilace [%]	*Žádné*	62,72	82,29	89,42
	Hamming	67,31	85,17	89,21
	ReedSolomon	70,93	85,96	90,44
		4. Grass	5. Blawat	6. DNA Fountain
	Žádné	/	75,73	7,23
	Hamming	88,07	77,71	8,04
	ReedSolomon	88,68	78,48	8,82

Tabulka 4.3: Pravděpodobnost dekompilace schémat v hodnotách průměrných chybovostí z obrázku 4.16.

V tabulce 4.3 jsou vypsané hodnoty zvýrazněných bodů v pozicích skutečných chybovostí z výsledků 4.2 resp. 4.17, 4.18 a 4.19.

Největší robustnost vykazují postupně schémata Goldman, Grass, Church, Blawat, Base a DNA Fountain.

S ohledem na nejrobustnější schéma a nejúčinnější bitový kodér v této práci jsou na následujícím obrázku vyobrazeny RD křivky všech metrik ze sekce 4.4 pro případ použití JPEG XL kodéru s Goldman schématem a použitím ReedSolomon chybově opravného kódu (černě). Tato kombinace představuje nejvhodnější řešení v jednotkách nukleotid na pixel (*Nt/pix*) v logaritmickém měřítku na ose x pro možné účely porovnání s jinými pracemi v této oblasti. Pro kombinace s jinými DNA schématy (například Grass nebo Church) není vykreslení potřeba, protože zásadní kompresi přináší především bitový kodér (viz sekce 4.3). Křivky pro jiná schémata by byly velmi blízko a ztrácela by se přehlednost. Navíc Grass schéma, které má podle tabulky 4.3 druhou největší úspěšnost dekompilace, má naopak podle obrázku 4.5 horší vlastní kompresní poměr než Goldman schéma. Jak již bylo zmíněno v sekci 4.2, je JPEG XL kompatibilní s klasickým JPEG formátem a lze jej využít pro bezeztrátovou kompresi `.jpeg` souborů jako dodatečnou kompresní nástavbu, což usnadňuje jeho adopci. Tato kombinace komprese JPEG a dodatečná bezeztrátová komprese JPEG XL by v praxi jistě našla větší využití, protože světově se stále více uplatňuje JPEG kodér a velké množství uložených obrazových dat je právě v tomto formátu. Z tohoto důvodu je do následujícího obrázku RD křivek metrik vykreslen ještě výsledek JPEG kodéru (modře) a JPEG kodéru s bezeztrátovou nástavbou JPEG XL (černě), oba samozřejmě s použitím Goldman schématu a chyby opravujícího ReedSolomon kódu.



Obrázek 4.20: RD křivky všech metrik (FSIM, FSIM_C, PSNR_HVS, SSIM, MS-SSIM A VMAF) pro nastavený bitový kodér JPEG XL (červeně), JPEG (modře) a JPEG s bezestrátovou nástavbou JPEG XL (černě). Vše společně s transkódujícím DNA schématem Goldman s použitým chyby opravujícím kódem ReedSolomon.

Z obrázku 4.20 je zřejmé, že kombinace JPEG kodéru s bezeztrátovou nástavbou JPEG XL (černé křivky) dosahuje srovnatelných, a dokonce i mírně lepších výsledků objektivních kvalit obrazu v závislosti na potřebném počtu nukleotidů na pixel, ve srovnání s použitím samotného kodéru JPEG XL (červené křivky).

4.8 Výpočetní rychlost schémat

Pro úplné porovnání je nutné zobrazit výpočetní doby komprese a dekomprese binárních dat těmito schématy (viz obrázek 4.1). V této sekci jsou tyto časy spočteny pro všechna schémata s použitím kódu ReedSolomon. Použití ReedSolomon, Hamming nebo žádného opravného kódu mělo mírný vliv na čas dekódování, který se v tomto pořadí mírně snižoval. Hamming prodlužoval čas dekomprese v průměru o 3% a ReedSolomon v průměru o 15% oproti situaci bez použití opravného kódu.

Výpočet byl proveden na platformě s MacBook Air vybaveným čipem M1 a operačním systémem macOS 14 Sonoma.

Velikost souborů [kB]			Doba zakódování [s]								
Min.	Max.	Prům.	1. Base			2. Church			3. Goldman		
			Min.	Max.	Prům.	Min.	Max.	Prům.	Min.	Max.	Prům.
6,0	16,6	9,4	0,01	0,01	0,01	0,02	0,07	0,04	0,01	0,03	0,02
7,8	19,8	11,3	0,01	0,02	0,01	0,03	0,08	0,04	0,02	0,04	0,02
9,6	24,4	14,3	0,01	0,02	0,01	0,04	0,09	0,05	0,02	0,05	0,03
11,4	29,5	17,4	0,01	0,02	0,01	0,04	0,12	0,07	0,02	0,06	0,03
12,6	32,9	19,6	0,01	0,03	0,02	0,05	0,13	0,08	0,02	0,07	0,04
13,1	34,5	20,5	0,01	0,04	0,02	0,05	0,13	0,08	0,03	0,07	0,04
13,9	37,1	22,1	0,01	0,03	0,02	0,05	0,15	0,09	0,03	0,07	0,04
15,0	39,9	23,8	0,01	0,03	0,02	0,06	0,15	0,09	0,03	0,08	0,05
16,0	42,9	25,7	0,01	0,04	0,02	0,06	0,17	0,10	0,03	0,09	0,05
17,3	46,7	28,1	0,01	0,05	0,02	0,07	0,18	0,11	0,03	0,10	0,06
18,8	52,6	31,3	0,02	0,05	0,03	0,07	0,20	0,12	0,04	0,10	0,06
20,6	59,0	34,9	0,02	0,05	0,03	0,08	0,23	0,14	0,04	0,12	0,07
23,5	67,8	40,2	0,02	0,06	0,03	0,09	0,27	0,16	0,05	0,14	0,08
27,5	78,6	47,0	0,02	0,07	0,04	0,11	0,31	0,18	0,05	0,21	0,10
33,3	94,3	56,5	0,03	0,08	0,05	0,13	0,39	0,22	0,07	0,19	0,11
42,3	115,7	70,2	0,04	0,10	0,06	0,17	0,47	0,27	0,09	0,23	0,14
58,7	145,3	92,4	0,05	0,12	0,08	0,23	0,57	0,36	0,12	0,29	0,18
93,8	198,7	139,4	0,08	0,17	0,12	0,36	0,78	0,55	0,19	0,39	0,28
335,7	557,0	427,9	0,28	0,51	0,37	1,38	2,18	1,69	0,67	1,23	0,88

Tabulka 4.4: Doba zakódování pro schémata: Base, Church a Goldman s chybově opravným kódem ReedSolomon.

Velikost souborů [kB]			Doba zakódování [s]								
Min.	Max.	Prům.	4. Grass			5. Blawat			6. DNA Fountain		
			Min.	Max.	Prům.	Min.	Max.	Prům.	Min.	Max.	Prům.
6,0	16,6	9,4	0,01	0,06	0,03	0,01	0,05	0,03	0,07	0,21	0,11
7,8	19,8	11,3	0,02	0,05	0,03	0,02	0,07	0,03	0,09	0,25	0,14
9,6	24,4	14,3	0,02	0,07	0,04	0,02	0,07	0,04	0,11	0,30	0,18
11,4	29,5	17,4	0,03	0,09	0,05	0,03	0,10	0,05	0,15	0,39	0,22
12,6	32,9	19,6	0,03	0,10	0,06	0,03	0,12	0,06	0,16	0,44	0,25
13,1	34,5	20,5	0,03	0,11	0,06	0,04	0,10	0,06	0,17	0,45	0,26
13,9	37,1	22,1	0,04	0,10	0,06	0,04	0,10	0,06	0,18	0,48	0,28
15,0	39,9	23,8	0,04	0,14	0,07	0,04	0,14	0,07	0,19	0,53	0,30
16,0	42,9	25,7	0,04	0,15	0,08	0,04	0,16	0,08	0,20	0,57	0,33
17,3	46,7	28,1	0,05	0,16	0,09	0,04	0,16	0,09	0,22	0,62	0,36
18,8	52,6	31,3	0,05	0,16	0,09	0,05	0,17	0,10	0,24	0,73	0,41
20,6	59,0	34,9	0,06	0,18	0,10	0,05	0,17	0,10	0,26	0,83	0,46
23,5	67,8	40,2	0,06	0,20	0,12	0,07	0,25	0,12	0,30	0,97	0,54
27,5	78,6	47,0	0,07	0,27	0,14	0,07	0,25	0,13	0,35	1,14	0,64
33,3	94,3	56,5	0,09	0,30	0,17	0,10	0,29	0,16	0,44	1,35	0,78
42,3	115,7	70,2	0,12	0,43	0,21	0,12	0,43	0,21	0,56	1,72	0,99
58,7	145,3	92,4	0,16	0,52	0,27	0,17	0,47	0,27	0,82	2,24	1,35
93,8	198,7	139,4	0,24	0,57	0,40	0,22	0,60	0,40	1,35	4,19	2,21
335,7	557,0	427,9	0,81	1,80	1,18	0,90	1,77	1,23	5,66	10,26	7,46

Tabulka 4.5: Doba zakódování pro schémata: Grass, Blawat a DNA Fountain s chybově opravným kódem ReedSolomon.

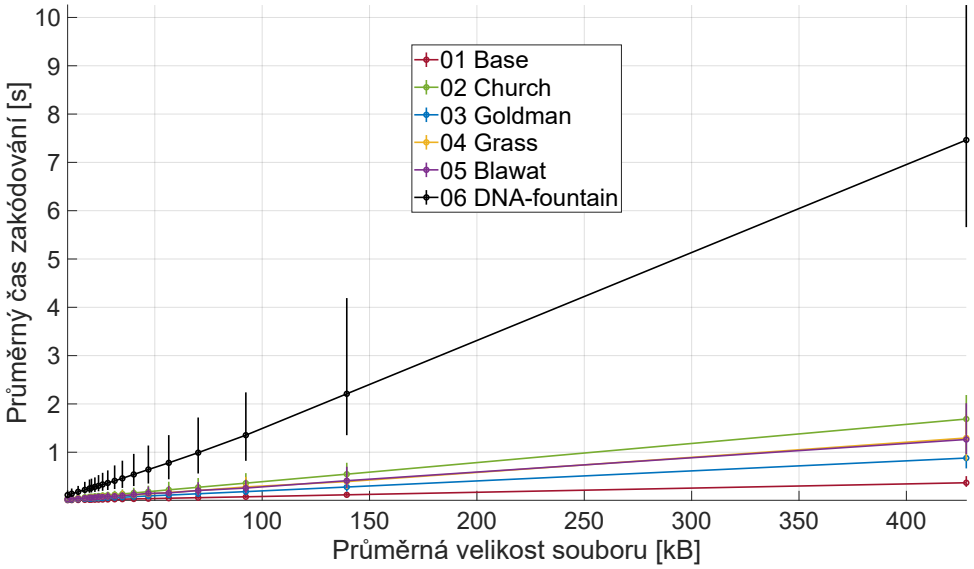
Velikost souborů [kB]			Doba dekódování [s]								
Min.	Max.	Prům.	1. Base			2. Church			3. Goldman		
			Min.	Max.	Prům.	Min.	Max.	Prům.	Min.	Max.	Prům.
6,0	16,6	9,4	0,00	0,01	0,00	0,01	0,03	0,01	0,07	0,20	0,11
7,8	19,8	11,3	0,00	0,01	0,01	0,01	0,03	0,02	0,09	0,26	0,14
9,6	24,4	14,3	0,00	0,01	0,01	0,02	0,04	0,02	0,12	0,29	0,17
11,4	29,5	17,4	0,01	0,02	0,01	0,02	0,05	0,03	0,14	0,36	0,21
12,6	32,9	19,6	0,01	0,02	0,01	0,02	0,05	0,03	0,15	0,40	0,24
13,1	34,5	20,5	0,01	0,02	0,01	0,02	0,06	0,03	0,16	0,42	0,25
13,9	37,1	22,1	0,01	0,02	0,01	0,02	0,06	0,04	0,17	0,45	0,27
15,0	39,9	23,8	0,01	0,02	0,01	0,02	0,06	0,04	0,18	0,48	0,29
16,0	42,9	25,7	0,01	0,02	0,01	0,03	0,07	0,04	0,19	0,53	0,31
17,3	46,7	28,1	0,01	0,03	0,01	0,03	0,10	0,05	0,21	0,57	0,34
18,8	52,6	31,3	0,01	0,03	0,02	0,03	0,08	0,05	0,23	0,63	0,38
20,6	59,0	34,9	0,01	0,03	0,02	0,04	0,10	0,05	0,25	0,71	0,42
23,5	67,8	40,2	0,01	0,04	0,02	0,04	0,11	0,06	0,28	0,82	0,49
27,5	78,6	47,0	0,01	0,04	0,02	0,04	0,12	0,08	0,33	0,95	0,57
33,3	94,3	56,5	0,02	0,05	0,03	0,06	0,15	0,09	0,40	1,14	0,68
42,3	115,7	70,2	0,02	0,06	0,04	0,07	0,19	0,11	0,52	1,42	0,85
58,7	145,3	92,4	0,03	0,09	0,05	0,09	0,23	0,15	0,71	1,75	1,12
93,8	198,7	139,4	0,05	0,11	0,07	0,15	0,33	0,23	1,13	2,40	1,69
335,7	557,0	427,9	0,18	0,34	0,23	0,54	0,94	0,71	4,07	6,77	5,20

Tabulka 4.6: Doba dekódování pro schémata: Base, Church a Goldman s chybově opravným kódem ReedSolomon.

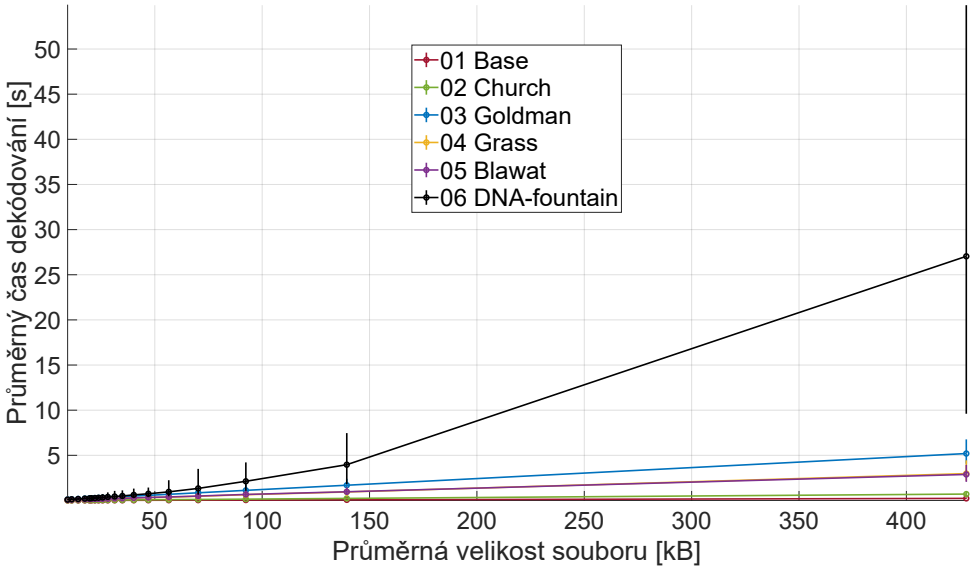
Velikost souborů [kB]			Doba dekódování [s]								
Min.	Max.	Prům.	4. Grass			5. Blawat			6. DNA Fountain		
			Min.	Max.	Prům.	Min.	Max.	Prům.	Min.	Max.	Prům.
6,0	16,6	9,4	0,04	0,14	0,07	0,03	0,13	0,06	0,05	0,21	0,10
7,8	19,8	11,3	0,04	0,13	0,07	0,04	0,13	0,07	0,07	0,31	0,13
9,6	24,4	14,3	0,06	0,16	0,10	0,05	0,18	0,10	0,09	0,32	0,16
11,4	29,5	17,4	0,07	0,24	0,11	0,06	0,24	0,12	0,11	0,45	0,21
12,6	32,9	19,6	0,07	0,24	0,13	0,07	0,25	0,13	0,14	0,45	0,23
13,1	34,5	20,5	0,08	0,29	0,14	0,07	0,27	0,15	0,15	0,48	0,25
13,9	37,1	22,1	0,08	0,30	0,15	0,08	0,24	0,14	0,15	0,46	0,26
15,0	39,9	23,8	0,08	0,33	0,16	0,08	0,27	0,16	0,17	0,55	0,30
16,0	42,9	25,7	0,10	0,36	0,18	0,09	0,33	0,17	0,18	0,70	0,33
17,3	46,7	28,1	0,11	0,40	0,20	0,12	0,37	0,19	0,23	0,90	0,39
18,8	52,6	31,3	0,11	0,41	0,22	0,11	0,37	0,22	0,21	1,06	0,44
20,6	59,0	34,9	0,13	0,48	0,25	0,15	0,38	0,23	0,25	1,12	0,50
23,5	67,8	40,2	0,17	0,55	0,28	0,15	0,45	0,27	0,27	1,32	0,62
27,5	78,6	47,0	0,20	0,64	0,33	0,21	0,60	0,33	0,38	1,43	0,74
33,3	94,3	56,5	0,22	0,69	0,39	0,22	0,77	0,39	0,47	2,24	0,96
42,3	115,7	70,2	0,26	0,66	0,46	0,28	0,92	0,49	0,64	3,50	1,34
58,7	145,3	92,4	0,40	1,10	0,63	0,31	0,98	0,62	1,08	4,22	2,12
93,8	198,7	139,4	0,61	1,23	0,91	0,60	1,51	0,94	1,82	7,46	3,96
335,7	557,0	427,9	1,95	4,24	2,97	1,84	4,02	3,06	9,61	54,86	27,04

Tabulka 4.7: Doba dekódování pro schémata: Grass, Blawat a DNA Fountain s chybově opravným kódem ReedSolomon.

Pro vhodnější grafické porovnání jsou na následujících obrázcích zobrazeny výsledky z tabulek 4.4, 4.5, 4.6 a 4.7 jako průběhy závislosti průměrného času zakódování/dekódování vůči průměrné velikosti souboru s chybovými úsečkami minima a maxima času zakódování/dekódování.



Obrázek 4.21: Průběhy času zakódování binárního souboru DNA schématem v závislosti na průměrné velikosti binárního souboru s chybovými úsečkami rozptylu maximálního a minimálního času zakódování (viz tabulky 4.4 a 4.5).



Obrázek 4.22: Průběhy času dekódování DNA souboru daným DNA schématem v závislosti na průměrné velikosti původního binárního souboru s chybovými úsečkami rozptylu maximálního a minimálního času dekódování (viz tabulky 4.6 a 4.7).

Z výsledků obrázků 4.21 a 4.22 je vidět, že všechna schémata, krom DNA Fountain, mají velmi blízké časy zakódování omezené shora přibližně 2 sekundami a také blízké časy dekódování omezené shora přibližně 5 sekundami pro všechny průměrné velikosti souborů. DNA Fountain je zde viditelně výpočetně náročnější. U větších komprimvaných souborů dosahuje času zakódování až v průměru 7 sekund a času dekódování až 27 sekund.

Kapitola 5

Závěr

V dnešní době, kdy množství digitálních dat exponenciálně roste, je výzkum a vývoj efektivních metod pro ukládání dat do DNA klíčový pro budoucí potřeby archivace. DNA ukládání dat nabízí obrovský potenciál díky své vysoké hustotě informace skutečného objemu, dlouhé životnosti a nízké energetické náročnosti, což ho činí ideálním kandidátem pro dlouhodobou archivaci velkých objemů dat.

Tato práce v kapitolách 2 a 3 seznamuje čtenáře s oblastí využití umělé syntézy DNA pro účely archivace obrazových dat. To zahrnuje základní stavbu a typy kódovacích procesů, restrikce spojené s fyzickými DNA řetězci pro ukládání dat a popsání principů kodérů od autorů, kteří se této problematice věnovali v posledních letech.

Hlavním cílem této práce, popsaným v kapitole 4, bylo na vybraném testovacím setu obrazových souborů a zvolených přístupech kódování dat do DNA ověřit a porovnat jejich účinnost. Konkrétně byla hodnocena kvalita rekonstruovaného obrazu, dodržování restrikcí DNA kódu, chybovost plynoucí z jejich nedodržení, robustnost vůči těmto chybám a výpočetní náročnost komprese a dekomprese. Byla použita množina nejpoužívanějších či nejnovějších binárních obrazových kodérů skupiny JPEG, kvatenární transkódovací DNA schémata, a objektivní metriky pro stanovení kvality obrazu doporučované odvětvím JPEG DNA v rámci probíhající standardizace.

Výsledky ukázaly, že schémata Goldman, Grass a Church dosahují nejlepšího dodržování biochemických restrikcí DNA molekul. V simulaci celého procesu syntézy, kapsulace a sekvencování řetězců tato schémata vykazovala i nejmenší chybovost. Simulace robustnosti všech schémat při jejich vykazované chybovosti ukázala, že největší úspěšnosti dekompilace dosahovala opět schémata Goldman, Grass a Church. Tyto výsledky byly navíc propočítány pro případy bez a s dodatečným chybově opravným kódováním Hamming či ReedSolomon. Ukázalo se, že vliv těchto opravných kódů na zvýšení robustnosti je pozitivní, ale nepříliš efektivní vzhledem k výraznějšímu prodloužení sekvencí, které způsobovaly. Nejrobustnějším a kompresně efektivním modelem se ukázala kombinace JPEG XL bitového kodéru (nebo JPEG s bezeztrátovou JPEG XL nástavbou) s Goldman DNA schématem. Výsledky metrik pro tuto kombinaci jsou kompresní účinností porovnatelné s nejnovějšími pracemi v této oblasti (např. [17, 41] či referenčními kodéry od JPEG DNA [8]).

Vhodným rozšířením této práce by mohlo být otestování použité nebo jiné implementace fontánového DNA kódování pro vyšší hodnoty redundance. Stejně tak by prodloužení bitové segmentace mohlo u většiny použitých schémat přispět k lepšímu dodržení například vyváženého GC obsahu, což by pravděpodobně vedlo ke snížení chybovosti schémat.

Pro oblast dalšího výzkumu doporučuji zaměřit se například na kompresi obrazových dat skenů s textovým obsahem. Využití například modelů NN pro extrakci informací za účelem efektivní komprimace a následného transkódování do DNA by mohlo najít využití v soudním, medicínském či státním archivnictví.

Literatura

[1] "Structure of DNA - Labster — theory.labster.com." https://theory.labster.com/structure-dna/. [Accessed 07-02-2024].

[2] "Data growth worldwide 2010-2025 | Statista — statista.com." https://www.statista.com/statistics/871513/worldwide-data-created/. [Accessed 13-02-2024].

[3] R. Hintemann and S. Hinterholzer, "Energy consumption of data centers worldwide-how will the internet become green?," in *ICT4S*, Lappeenranta, 2019.

[4] R. Carlson, "The world is running out of data storage. here's how dna can save us," *IEEE Spectrum*, vol. 61, no. 3, pp. 28–44, 2024.

[5] "JPEG - JPEG DNA — jpeg.org." https://jpeg.org/jpegdna/index.html. [Accessed 19-05-2024].

[6] P. T. Ebrahimi, "Final Call for Proposals on Digital Media Storage on DNA Support." https://ds.jpeg.org/documents/jpegdna/wg1n100476-099-REQ-Final_Call_for_Proposals_on_Digital_Media_Storage_on_DNA_Support.pdf. [Accessed 19-05-2024].

[7] T. E. Marc Antonini, "Use Cases and Requirements for DNA-based Media Storage version 1.0." https://ds.jpeg.org/documents/jpegdna/wg1n100252-096-REQ-Use_Cases_and_Requirements_for_DNA-based_Media_Storage_v1_0.pdf. [Accessed 19-05-2024].

[8] M. Antonini and T. Ebrahim, "JPEG DNA Common Test Conditions version 1.1." https://ds.jpeg.org/documents/jpegdna/wg1n100517-099-ICQ-JPEG_DNA_Common_Test_Conditions_v2.pdf. [Accessed 13-05-2024].

[9] N. C. Seeman, "Dna nanotechnology: novel dna constructions," *Annual review of biophysics and biomolecular structure*, vol. 27, no. 1, pp. 225–248, 1998.

[10] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized dna," *nature*, vol. 494, no. 7435, pp. 77–80, 2013.

[11] J. C. Peng and G. H. Karpen, "Epigenetic regulation of heterochromatic dna stability," *Current opinion in genetics & development*, vol. 18, no. 2, pp. 204–211, 2008.

[12] R. Blake and S. G. Delcourt, "Thermal stability of dna," *Nucleic acids research*, vol. 26, no. 14, pp. 3323–3332, 1998.

[13] W. Chen, M. Han, J. Zhou, Q. Ge, P. Wang, X. Zhang, S. Zhu, L. Song, and Y. Yuan, "An artificial chromosome for data storage," *National Science Review*, vol. 8, no. 5, p. nwab028, 2021.

[14] J. Zhou, C. Zhang, R. Wei, M. Han, S. Wang, K. Yang, L. Zhang, W. Chen, M. Wen, C. Li, *et al.*, "Exogenous artificial dna forms chromatin structure with active transcription in yeast," *Science China Life Sciences*, pp. 1–10, 2021.

[15] H. Nakano, K. Matsuda, M. Yohda, T. Nagamune, I. Endo, and T. Yamane, "High speed polymerase chain reaction in constant flow," *Bioscience, biotechnology, and biochemistry*, vol. 58, no. 2, pp. 349–352, 1994.

[16] M. Schwarz, M. Welzel, T. Kabdullayeva, A. Becker, B. Freisleben, and D. Heider, "MESA: automated assessment of synthetic DNA fragments and simulation of DNA synthesis, storage, sequencing and PCR errors," *Bioinformatics*, vol. 36, pp. 3322–3326, 03 2020.

[17] M. Dimopoulou, M. Antonini, P. Barbry, and R. Appuswamy, "A biologically constrained encoding solution for long-term storage of images onto synthetic dna," in *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, IEEE, 2019.

[18] M. Dimopoulou, E. G. San Antonio, and M. Antonini, "A jpeg-based image coding solution for data storage on dna," in *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 786–790, IEEE, 2021.

[19] P. M. Schwarz and B. Freisleben, "Norec4dna: using near-optimal rateless erasure codes for dna storage," *BMC bioinformatics*, vol. 22, no. 1, pp. 1–28, 2021.

[20] J. L. Weber and C. Wong, "Mutation of human short tandem repeats," *Human molecular genetics*, vol. 2, no. 8, pp. 1123–1128, 1993.

[21] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on dna in silica with error-correcting codes," *Angewandte Chemie International Edition*, vol. 54, no. 8, pp. 2552–2555, 2015.

[22] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in dna," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012.

[23] M. Blawat, K. Gaedke, I. Huetter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, and G. M. Church, "Forward error correction for dna data storage," *Procedia Computer Science*, vol. 80, pp. 1011–1022, 2016.

[24] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56–67, 1998.

[25] M. Luby, "Lt codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp. 271–271, IEEE Computer Society, 2002.

[26] P. Maymounkov, "Online codes," tech. rep., Technical report, New York University, 2002.

[27] A. Shokrollahi, "Raptor codes," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[28] M. Dimopoulou and M. Antonini, "Image storage in dna using vector quantization," in *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 516–520, IEEE, 2021.

[29] "JPEG - JPEG 1 — jpeg.org." `https://jpeg.org/jpeg/index.html`. [Accessed 12-02-2024].

[30] Z. Ping, H. Zhang, S. Chen, Q. Zhuang, S. Zhu, and Y. Shen, "Chamaeleo: an integrated evaluation platform for dna storage," *Synthetic Biology Journal*, pp. 1–15, 2020.

[31] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[32] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[33] "Reed-solomon codes — cs.cmu.edu." `https://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html`. [Accessed 20-05-2024].

[34] L. Carlitz, "The arithmetic of polynomials in a galois field," *American Journal of Mathematics*, vol. 54, no. 1, pp. 39–50, 1932.

[35] "True Color Kodak Images — r0k.us." `https://r0k.us/graphics/kodak/index.html`. [Accessed 07-05-2024].

[36] "JPEG - JPEG 2000 — jpeg.org." `https://jpeg.org/jpeg2000/index.html`. [Accessed 07-05-2024].

[37] A. PRZELASKOWSKI, "The jpeg2000 standard for medical image applications," *Task Quarterly*, vol. 8, no. 2, pp. 147–158, 2004.

[38] "JPEG - JPEG XL — jpeg.org." `https://jpeg.org/jpegxl/index.html`. [Accessed 07-05-2024].

[39] "Sips MAN page - macOS - — ss64.com." `https://ss64.com/mac/sips.html`. [Accessed 07-05-2024].

[40] "GitHub - libjxl/libjxl: JPEG XL image format reference implementation — github.com." `https://github.com/libjxl/libjxl`. [Accessed 07-05-2024].

[41] "Technical description of the EPFL submission to the JPEG DNA CfP — infoscience.epfl.ch." `https://infoscience.epfl.ch/record/306753?v=pdf`. [Accessed 18-05-2024].

[42] Aerilius, "A shell script to translate Apple's SIPS commands to ImageMagick.." `https://gist.github.com/Aerilius/4557816`. [Accessed 07-05-2024].

[43] "ImageMagick — imagemagick.org." `https://imagemagick.org/index.php`. [Accessed 07-05-2024].

[44] Z. Ping, S. Chen, G. Zhou, X. Huang, S. J. Zhu, H. Zhang, H. H. Lee, Z. Lan, J. Cui, T. Chen, *et al.*, "Towards practical and robust dna-based data archiving using the yin–yang codec system," *Nature Computational Science*, vol. 2, no. 4, pp. 234–242, 2022.

[45] "Feature SIMilarity Index for IQA — www4.comp.polyu.edu.hk." `https://www4.comp.polyu.edu.hk/~cslzhang/IQA/FSIM/FSIM.htm`. [Accessed 12-05-2024].

[46] "GitHub - Netflix/vmaf: Perceptual video quality assessment based on multi-method fusion. — github.com." `https://github.com/Netflix/vmaf/tree/master`. [Accessed 16-05-2024].

[47] J. Ozer, "Mapping SSIM and VMAF Scores to Subjective Ratings — streaminglearningcenter.com." `https://streaminglearningcenter.com/learning/mapping-ssim-vmaf-scores-subjective-ratings.html`. [Accessed 21-05-2024].

[48] F. Ullah, J. Lee, S. Jamil, and O.-J. Kwon, "Subjective assessment of objective image quality metrics range guaranteeing visually lossless compression," *Sensors*, vol. 23, no. 3, p. 1297, 2023.

[49] Visionular-admin, "Making Sense of PSNR, SSIM, VMAF - Visionular — visionular.ai." `https://visionular.ai/vmaf-ssim-psnr-quality-metrics/`. [Accessed 21-05-2024].

[50] "Bjontegaard metric — mathworks.com." https://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaard-metric. [Accessed 12-05-2024].

[51] "GitHub - umr-ds/mesa_dna_sim — github.com." https://github.com/umr-ds/mesa_dna_sim/tree/master. [Accessed 16-05-2024].

[52] "MESA DNA simulator — mesa.mosla.de." https://mesa.mosla.de/. [Accessed 16-05-2024].

[53] N. B. Lubock, D. Zhang, A. M. Sidore, G. M. Church, and S. Kosuri, "A systematic comparison of error correction enzymes by next-generation sequencing," *Nucleic Acids Research*, vol. 45, no. 15, pp. 9206–9217, 2017.

[54] "Quality (Phred) scores — drive5.com." https://www.drive5.com/usearch/manual/quality_score.html. [Accessed 20-05-2024].

Příloha A

Přiložené soubory

Zde je uveden obsah přiloženého `.zip` souboru s popisem jednotlivých skriptů a jejich funkcí:

- `b1_script_jpeg.sh`:
  komprese `.png` na `.jpeg`
- `b2_script_2000.sh`:
  komprese `.png` na `.jp2`
- `b3_script_xl.sh`:
  komprese `.png` na `.jxl`
- `b4_png2y4m.sh`:
  převod `.png` na `.y4m` pro výpočet objektivních metrik pomocí knihovny libvmaf
- `b5_libvmaf.sh`:
  výpočet metrik pro komprimované obrazy pomocí knihovny libvmaf
- `b6_jpeg2jxl.sh`:
  bezeztrátová komprese `.jpeg` na `.jxl`
- `m1_matlab_fsim.m`:
  výpočet metrik FSIM a FSIM_C
- `m2_combine_metrics.m`:
  kombinace všech výsledných metrik do jedné matice
- `m3_sequences_lengths.m`:
  výpočet délek sekvencí `.dna` souborů
- `m4_files_sizes.m`:
  určení velikostí souborů v bitech
- `m5_biochem_constrain.m`:
  vlastní implementace pro stanovení dodržování biochemických restrikcí DNA schémata

- `m6_mesa_simulace.m`:
  vyhodnocení výsledků MESA MOSLA simulace
- `p1_combined_transcode.py`:
  aplikace transkódujících DNA schémat na komprimované obrazové soubory
- `p2_robustness_eval.py`:
  výpočet robustnosti jednotlivých schémat
- `p3_enc_dec_times.py`:
  výpočet časů pro kódování a dekódování DNA schémat