Komprese DNA pomocí víceproudé komprese a predikce báz

Jan Jelínek, Radek Miček

Víceproudá komprese

- angl. Multistream compression (MSC)
- statistická metoda
- autoři: Kochánek, Lánský, Uzel, Žemlička
- lze použít místo Huffmanova kódování nebo aritmetického kódování

Úvod do predikce báz v DNA

- inspirováno PPM
  - hádá následující znak vstupu na základě kontextu
  - kontext = několik znaků, jenž těsně předchází hádanému znaku
  - podívá se, jaké znaky následovaly kontext v již zpracované části vstupu, a podle toho zkusí uhádnout následující znak
  - při kompresi textu se typicky používá kontext délky 6-8 znaků, který se v případě potřeby zkracuje

Úvod do predikce báz v DNA (2)

- protože báze jsou malé, použijeme delší kontext
- v kontextu povolíme díry (znaky na vybraných pozicích budeme ignorovat)

Predikce báz v DNA – konkrétně

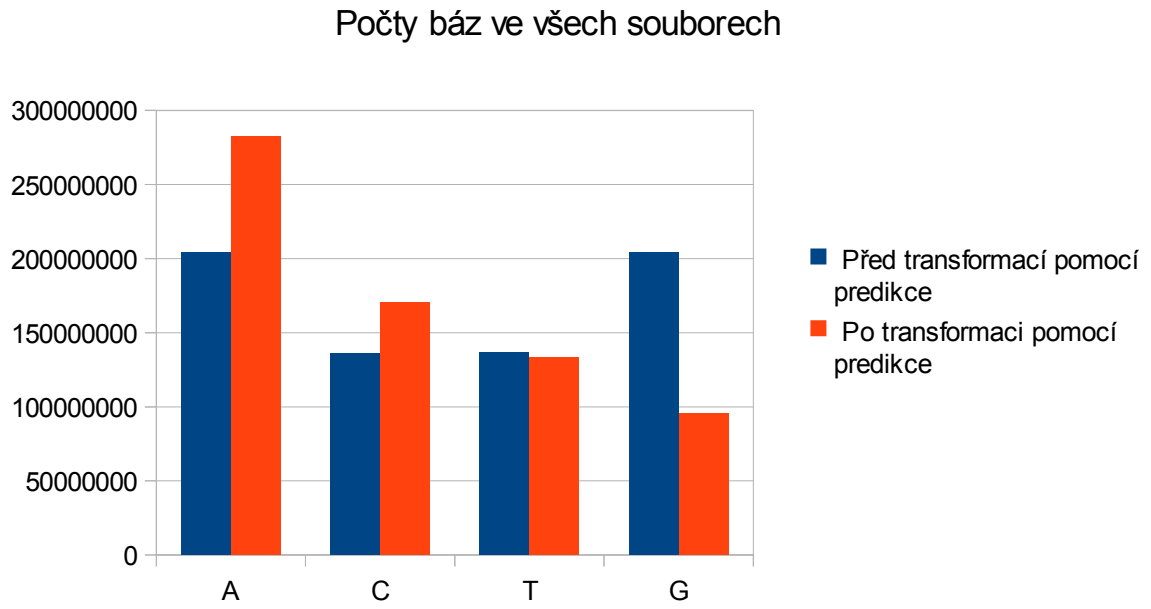
- používáme kontexty délky 32, 16, 8, 4, 2, 1 báz
  - kontext délky 16 báz udržujeme ve třech variantách:
    - neposunutý
    - posunutý o jednu bázi (báze těsně před hádaným znakem se nemusí shodovat)
    - posunutý o dvě báze
- pro každý kontext si udržujeme statistiky báz, které se v daném kontextu vyskytly – z nich odvodíme pravděpodobnosti výskytu báz v onom kontextu
  - příliš staré výskyty zapomínáme – uvažujeme pouze posledních 12 milionů báz (má výrazný vliv na spotřebu paměti)

Transformace DNA pomocí predikce

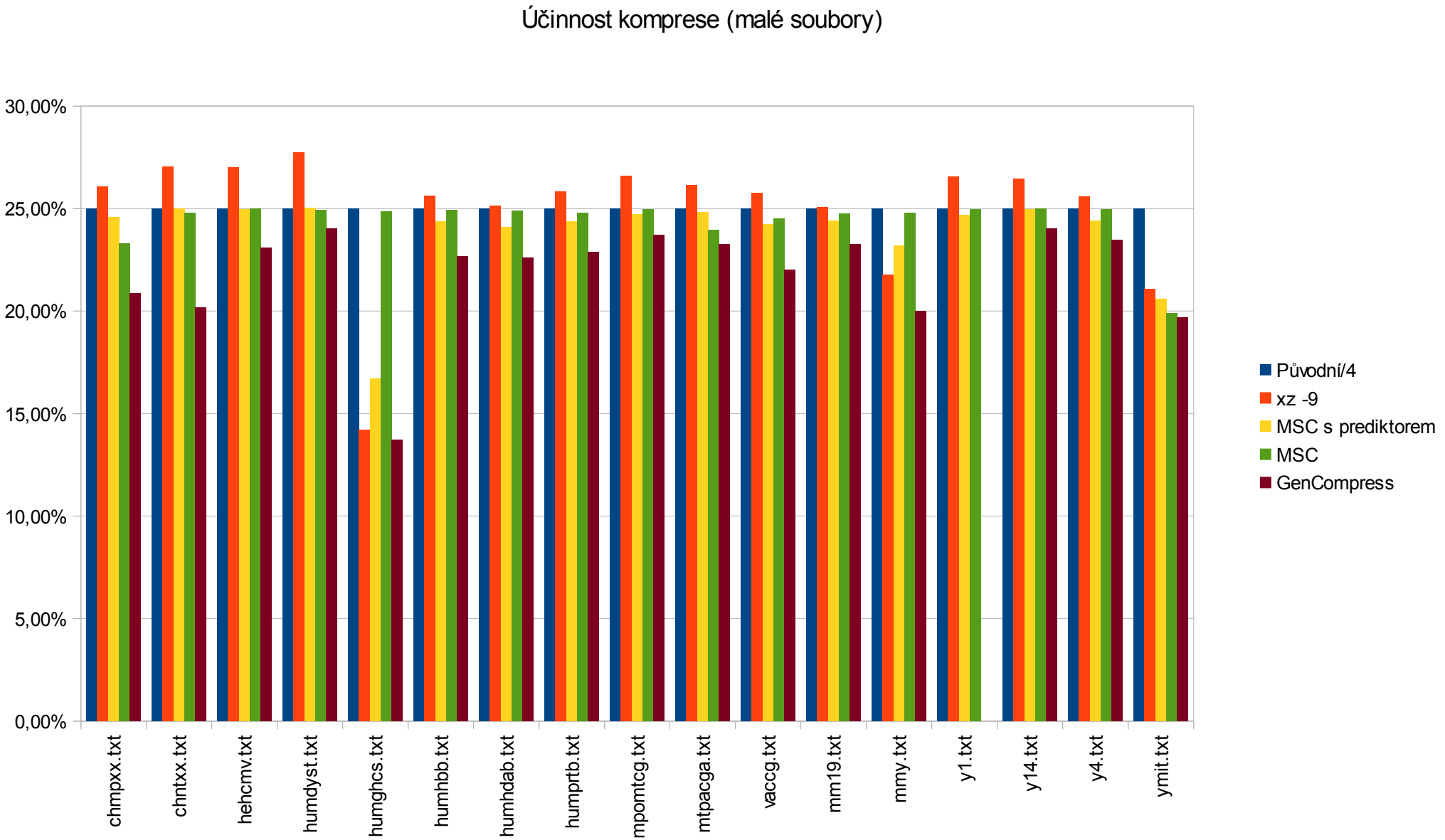
- predikce nám seřadí báze podle pravděpodobnosti výskytu v daném kontextu
- báze na vstupu transformujeme podle pravděpodobnosti výskytu:
  - 1. nejpravděpodobnější báze → a
  - 2. nejpravděpodobnější báze → c
  - 3. nejpravděpodobnější báze → g
  - 4. nejpravděpodobnější báze → t

Důsledky transformace DNA

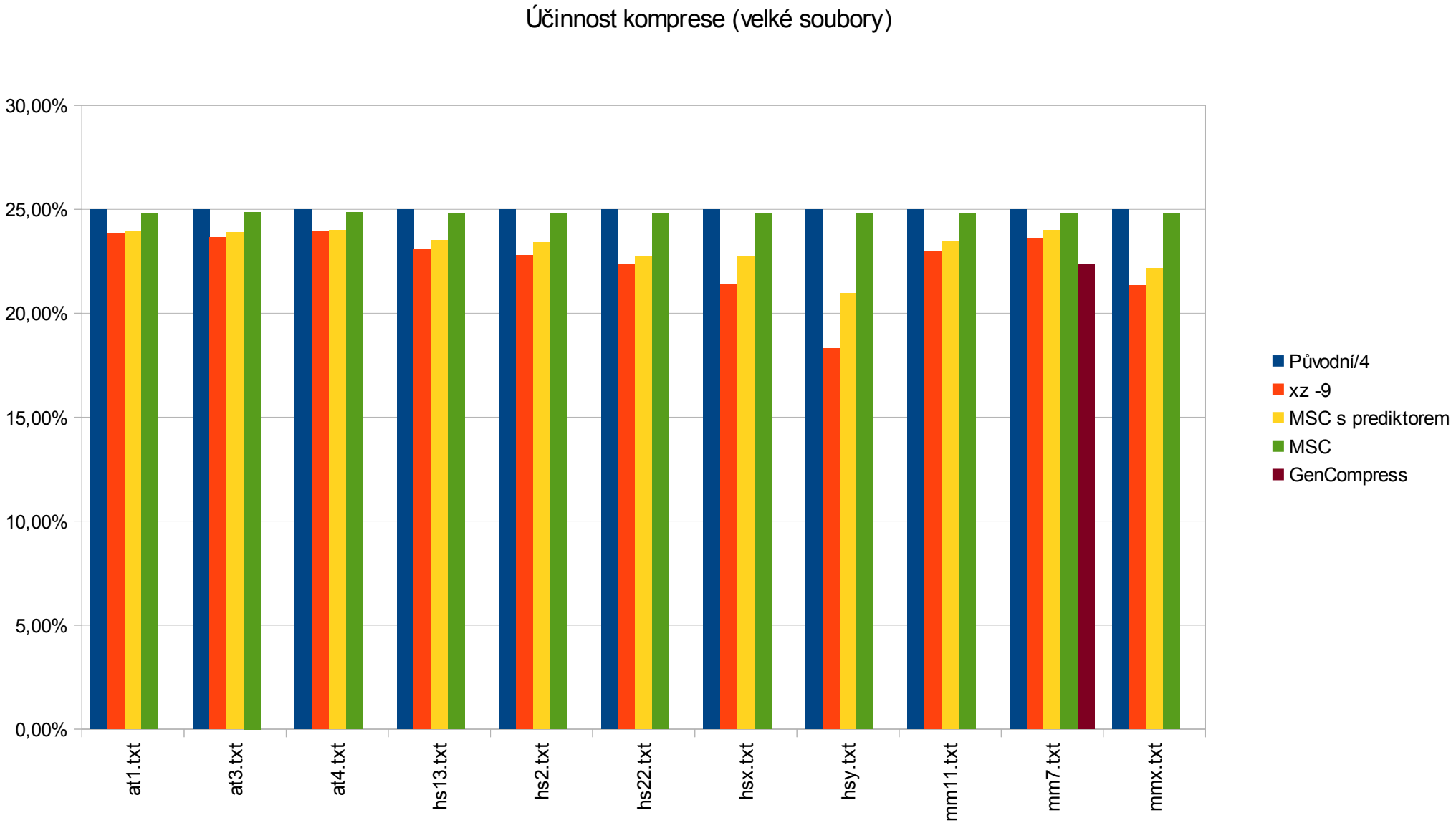
- výstup bude (v ideálním případě) obsahovat hodně bází a, méně bází c, …
- výhodné pro statistické metody komprese
- transformace Manziniho DNA korpusu:



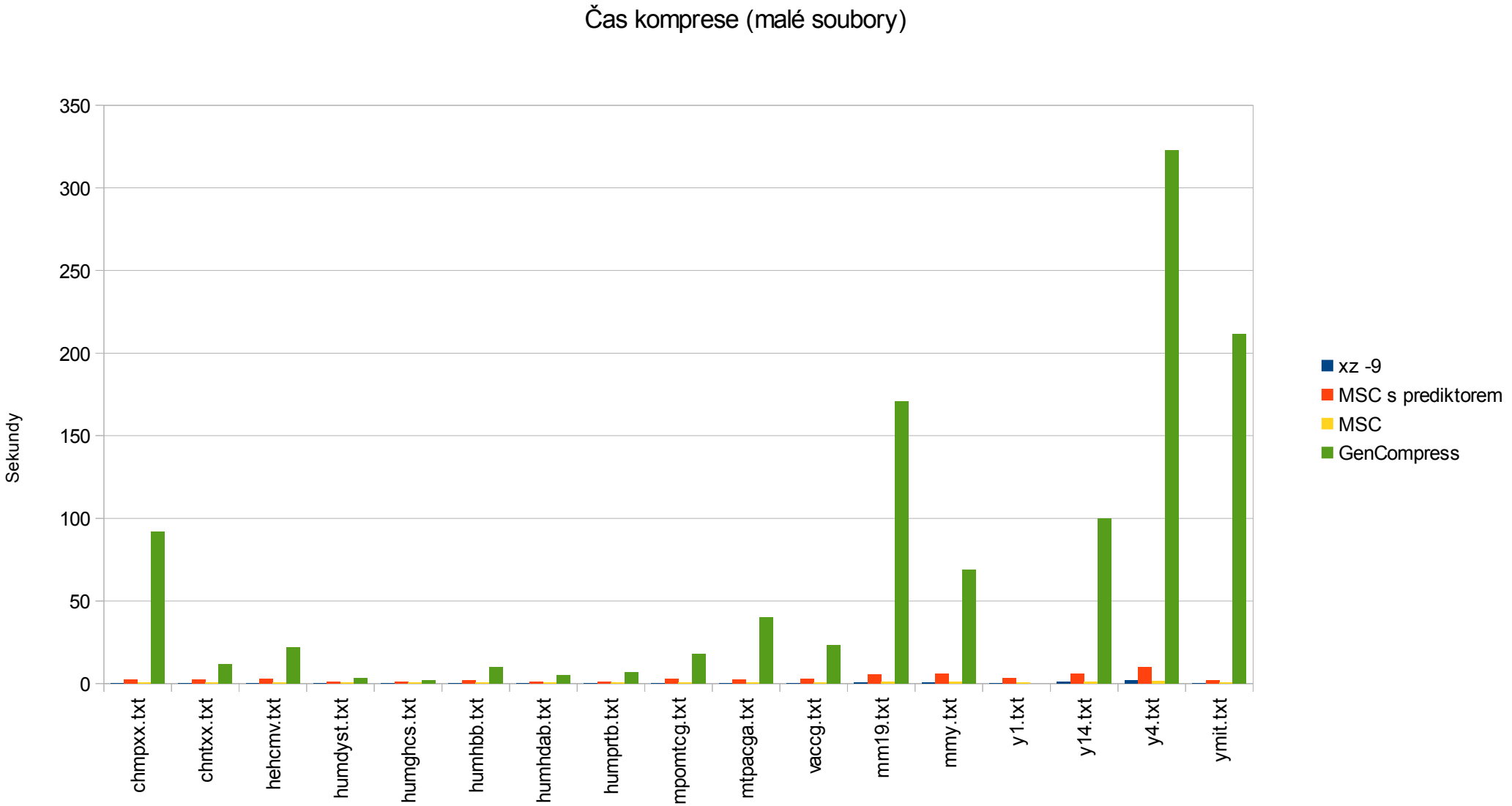
Účinnost komprese



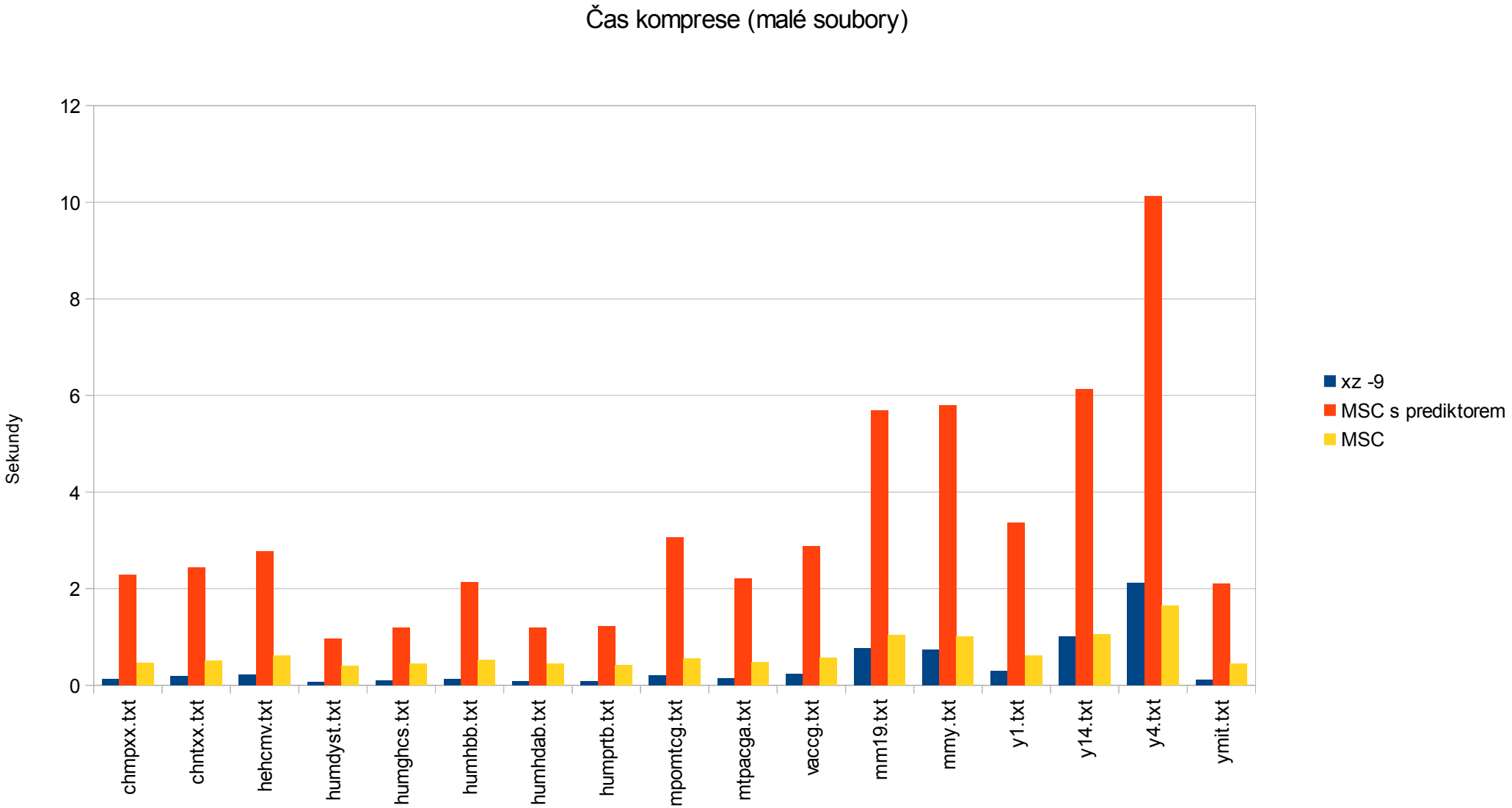
Účinnost komprese (2)



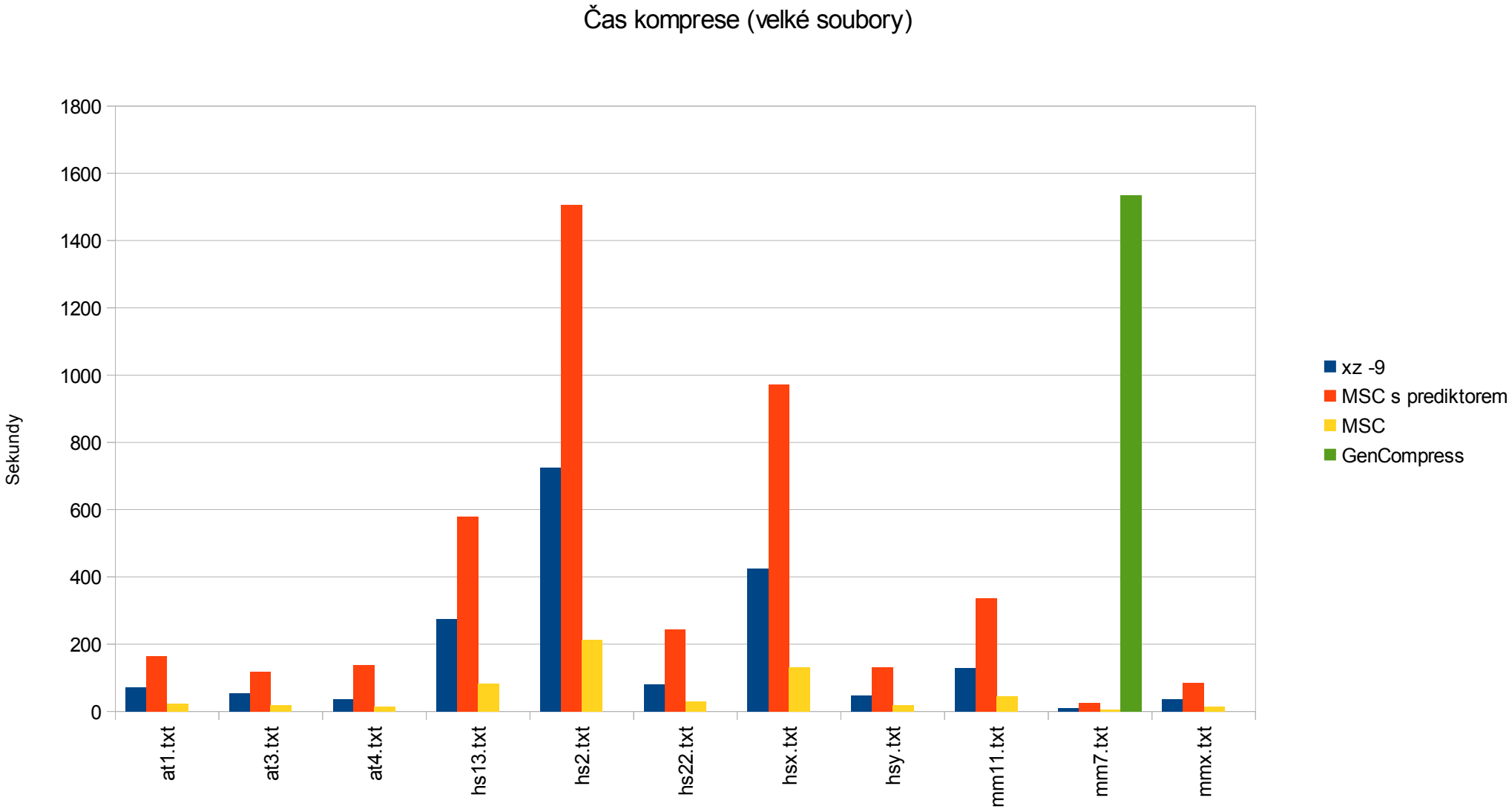
Rychlost komprese



Rychlost komprese (bez GC)



Rychlost komprese (2)



Poznámky k experimentům

- testy byly prováděny na notebooku s procesorem Intel Core i3 M380, 4 GB RAM, Windows 7 Home Premium SP1 (64 bit)
- testované programy byly 32-bitové aplikace
- GenCompress nebyl schopen některé soubory zkomprimovat
  - u největších souborů ihned po spuštění ohlásil, že nelze alokovat potřebnou pamět
  - u středních souborů docela dlouho (i více než hodinu) komprimoval, a poté provedl neplatnou operaci a byl ukončen OS

Implementace predikce

- predikce je implementována v jazyce F#
- aby se zbytečně nezatěžoval garbage collector, jsou alokované objekty poolovány
- vyhledávání v kontextech je implementováno pomocí třídy Dictionary<K, V>
  - prediktor využil 1.7 GB paměti nezávisle na komprimovaném souboru, použitím jiné datové struktury (např. judy array) by šlo spotřebu paměti snížit
- používané kontexty lze snadno změnit
  - např. díry lze nastavit na libovolné pozice

Závěr

- predikce báz v DNA téměř vždy pomohla k lepší kompresi pomocí MSC
- nevýhodou navržené transformace je, že vůbec nezohledňuje velikost pravděpodobností výskytu báz vůči sobě (např. pravděpodobnosti 0.9, 0.09, 0.01, 0 vyústí ve stejné chování jako pravděpodobnosti 0.34, 0.33, 0.32, 0.01)
- jako rozšíření projektu by bylo možné uvažovat prediktor, který by automaticky nastavil délky kontextů a pozice děr

Zdroje

- Manziniho DNA korpus:
  http://people.unipmn.it/~manzini/dnacorpus/
- GenCompress pro Windows (32-bit):
  http://www.cs.cityu.edu.hk/~cssamk/gencomp/downGen.htm
- xz 5.0.3 pro Windows (32-bit):
  http://tukaani.org/xz/