# Poglavje 1

# Biološko procesiranje

Biološko procesiranje temelji na osnovnem kodirnem mediju, ki ga predstavlja molekula *deoksiribonukleinske kisline* (angl. *deoxyribonucleic acid* - DNA) in na koncentracijah opazovanih kemijskih zvrsti v biološkem sistemu. Procesiranje se tako v smislu nosilca podatkov prenese iz klasičnih elektronskih polprevodniških medijev na nov *molekularni* ali *biološki medij*. Nadzorovana dinamika v tovrstnih sistemih temelji na *biotehnoloških metodah* in vidikih *sistemske biologije* [1]. Struktura DNA in posamezne kemijske zvrsti so zanimive za realizacijo pomnjenja in procesiranja, ker istočasno predstavljajo medij "hranjenja" genske zasnove živih organizmov (angl. *blue print*) in so zaradi tega predmet obsežnih raziskav in tehnoloških izboljšav pri njenih manipulacijskih metodah. Ni potrebno posebej omenjati, da je z gensko zasnovo živega bitja determinirana dinamika v njegovih posameznih celicah in s tem posredno dinamičen proces življenja biološkega bitja kot celote.

V pričujočem poglavju si bomo ogledali vse osnovne koncepte, ki nas bodo z različnih zornih kotov skušali prepričati, da je DNA medij eden od možnih medijev pomnjenja in procesiranja bodočnosti. Na tem mestu eksplicitno poudarimo, da so biološke osnove razložene samo do mere, kot je to potrebno za osnovno razumevanje konceptov in so večinoma posplošene v „jezik računalništva".

## 1.1 Definicije sistemske, sintezne in računske biologije

Na samem začetku si oglejmo definicije *sistemske*, *sintezne* in *računske biologije*. Enoličnih virov zanje zaradi nepoenotenega mnenja o njihovi vsebini žal ne moremo navesti. Tako so v nadaljevanju navedene definicije zgolj videnje avtorja pričujočega dela.

**Definicija 1** *Sistemska biologija (angl. system biology) je interdisciplinarna veda, ki skuša čim natančneje razumeti dinamiko procesov v že obstoječih bioloških sistemih, ki jih najdemo v naravnem okolju.*

Razumevanje dinamike je pogojeno z nivojem, s katerega biološki sistem opazujemo. Od spodaj navzgor si ti nivoji sledijo v zaporedju genskega izražanja in proteinov, metabolnih poti, subceličnih mehanizmov, celic, tkiv, organov, organizmov, interakcij med enakimi in različnimi organizmi itd [2].

**Definicija 2** *Sintezna biologija (angl. synthetic biology) je interdisciplinarna veda, ki na področja biologije, medicine, kemije in farmacije uvaja inženirski pristop in postopke, ki celico (ali množico celic) spremenijo tako, da novonastala celica (ali več celic) opravlja neko novo koristno funkcijo, ki jo predhodno ni ali niso opravljale.*

Za predhodnike sintezne biologije lahko štejemo križanje sort in genetski inženiring. Iz definicije je razvidna njena sintezna naravnanost (angl. *bottom up*).

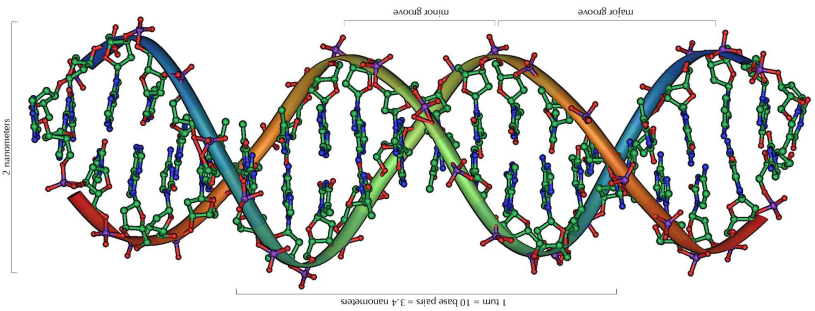
**Definicija 3** *Računska biologija (angl. computational biology) raziskuje in vzpostavlja računske in avtomatizirane metode, ki so v pomoč pri snovanju celic z novimi koristnimi funkcijami.*

Na tem mestu ne smemo pozabiti tudi na področje *bioinformatike* (angl. *bioinformatics*), ki raziskuje in vzpostavlja metode za avtomatizirano hrambo in analizo velikih količin bioloških podatkov. Tako bi lahko naredili grobo posplošitev, da računska biologija služi predvsem kot orodje sintezne biologije, bioinformatika pa predvsem kot orodje sistemske biologije. Marsikdo se z zadnjim stavkom verjetno ne bo strinjal. Sorodno področje bioinformatiki je področje *funkcijske genomike*, ki skuša identificirati funkcionalne značilnosti genskega zapisa posameznega organizma.

V obdobju zadnjih let se uveljavi tudi pojem *bioračunalništva* (angl. *biological computing*), ki raziskuje, kako iz enostavnih bioloških gradnikov (npr. celic) zgraditi sistem ali platformo, na kateri bi se lahko vršilo procesiranje, kot se vrši danes na osnovi polprevodniških gradnikov v elektronskih računalnikih.

## 1.2 Osnove DNA medija

Če kodiranje podatkov v klasičnih digitalnih elektronskih strukturah temelji na logični "0" in "1", njuni vrednosti pa si interpretiramo z napetostnimi nivoji na ključnih točkah digitalnega vezja, **kodiranje podatkov** v DNA temelji na poljubno dolgem nizu (molekuli), ki ga sestavljajo v poljubnem zaporedju vezane baze ali nukleotidi $A$ (adenin), $C$ (citozin), $G$ (gvanin) in $T$ (timin). Poljubna DNA je sestavljena iz dveh nizov (dveh molekul), ki se prepletata v vijačnico, kot je to predstavljeno na sliki 1.1.




Slika 1.1: Slika dvojne vijačnice, kjer se dve molekuli DNA zvijeta v DNA vijačnico - DNA zapis [3].

Gostota kodnega zapisa je pogojena z razdaljo med posameznima vezanima bazama. Omenjena razdalja meri približno 0,35 nm, kar vodi do približne gostote zapisa podatkov 18Mbitov/inch. Če bi vijačnico zvili na površino, bi s tem prišli do gostote zapisa $10^6$ Gbitov/inch$^2$, današnje tehnologije magnetnih trdih diskov pa dosegajo velikostni razred 10Gbitov/inch$^2$. Tako je DNA medij zanimiv že zaradi izredno **velike gostote zapisa**, ki nam jo ponuja.

Povrnimo se k samemu načinu zapisa podatkov. Povedali smo že, da DNA predstavlja vijačnica, sestavljena iz dveh nizov ali dveh molekul. Posamezni niz v vijačnici predstavlja poljubno zaporedje baz iz množice $B = \{A, T, C, G\}$, zaradi komplementarnosti baz ($A$ je komplementarna $T$ in obratno, $C$ pa je komplementarna $G$ in obratno), pa so dovoljeni na istoležnih mestih obeh nizov zgolj komplementarni bazni pari. Če bi torej imeli niz $S = \{ATTACGTCG\}$, bi bil v vijačnico z njim lahko povezan le niz $S' = \{TAATGCAGC\}$.

Predpostavimo, da nam vezava posamezne baze v nizu predstavlja hranjeni podatek. Glede na naravo DNA zapisa velja, da se na istoležnem mestu v komplementarnem nizu hrani komplement hranjenega podatka, kar nam omogoča **redundantno hrambo** podatkov in preko identifikacije lokacije nezadoščanja komplementarnosti posameznega para tudi možnost vpeljave mehanizma detekcije napak (angl. *detecting code*). Do okvar v nizih lahko pride pri njihovi sestavi, ki jo vršijo encimi (verjetnost takšne napake je manjša od $10^{-9}$), ali pa po sestavljanju zaradi zunanjih vplivov (npr. prisotnosti UV sevanja, termične

energije itd.).

Kot smo že omenili so osnovni manipulatorji nad DNA zapisom *encimi*. Pod manipulacijami nad DNA smatramo postopke *sestavljanja*, *rezanja*, *kopiranja* in *popravljanja* DNA. Navedene vrste manipulacij smatramo kot ene od možnih osnovnih operacij DNA procesiranja, ki bi lahko predstavljale vsebinski ekvivalent dvojiškim logičnim operacijam. Pomembna značilnost manipulacij nad DNA vijačnico je tudi **paralelizem** njihovega izvajanja. Slednje pomeni, da prisotnost različnih encimov na različnih delih vijačnice omogoča paralelne posege nad posameznimi deli niza in s tem možnost paralelne obdelave podatkov.

Z vidika implementacij bioloških računalniških sistemov smo do sedaj na področju DNA spoznali pojme **kodiranja podatkov**, **velike gostote zapisa**, **redundantne hrambe podatkov** in **inherentnega paralelizma**. Več zanimivih pojmov si bomo ogledali v naslednjih razdelkih.

## 1.3 Genom in gen

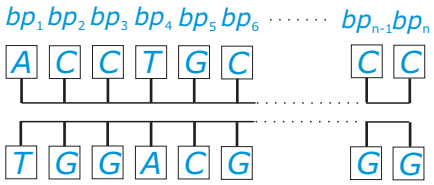
Doslej smo DNA obravnavali kot strukturo dveh prepletenih komplementarnih nizov, sestavljenih na osnovi koda $B = \{A, T, C, G\}$, pri čemer še nismo osvetlili pojmov *genoma* in *gena*. Slednja sta predstavljena v naslednjih razdelkih.

### 1.3.1 Genom

*Genom* opravlja funkcijo nosilca celotne množice podedovanih lastnosti (angl. *blueprint*) vsakega živega organizma. Poenostavljeno bi ga lahko poimenovali kot recept za kreacijo osebka posamezne vrste. Od odkritja obstoja genoma je minilo že več kot 60 let. Zgodovina raziskav, ki so pripeljale do tega odkritja in kasneje na prelomu tisočletja do prvega osnutka kartiranega človeškega genoma, je opisana v delih [4] in [5].

Genom strukturno izenačimo s predhodno opisano DNA. Sestavljen je z dvojno vijačnico *deoksi ribonukleinske kisline* (DNA), ki jo sestavljajo baze ali nukleotidi *gvanina* ($G$), *citozina* ($C$), *adenina* ($A$) in *timina* ($T$). Dolžino dvojne vijačnice izražamo s številom *baznih parov* (angl. *base pairs - bp*), pri čemer je vsak bazni par sestavljen iz dveh komplementarnih baz iz množice $\{G, C, A, T\}$. Nukleotid $G$ je komplementaren neukleotidu $C$ in nukleotid $A$ nukleotidu $T$ ter obratno. Komplementarna nukleotida, ki tvorita bazni par, ležita na istoležnih mestih vsak na svojem delu vijačnice. Grafična ponazoritev genoma (dvojne vijačnice) z $n$ baznimi pari je predstavljena na sliki 1.2. Iz slike je razvidno, da se nam na nivoju genoma ponuja možnost naslavljanja ali **adresiranja v linearnem prostoru**, kar je osnova za delovanje računalniških pomnilnikov.

Z vidika namena pričujočega dela na tem mestu zaobidemo natančnejši biološki opis genoma, kot je npr. njegov način zapisa v kromosomih.




Slika 1.2: Računalniški pogled na genom kot zaporedje komplementarnih baznih parov dolžine $n$.

### 1.3.2 Gen

*Gen* je zapisan na delu DNA vijačnice. Sestavljen je iz dveh podnizov istoležnih baznih parov. V pomenskem smislu gen predstavlja enoto dedovanja. Povedano drugače, je posamezni gen nosilec enotske podedovane lastnosti živega organizma. Pod enotami dedovanja imamo v mislih podvrženost posameznim boleznim (npr. celiakiji), barvo oči, barvo las itd. Odtod lahko na novo definiramo tudi genom, ki je tako s funkcionalnega vidika sestavljen iz zaporedja vseh genov opazovanega biološkega sistema. Ob predpostavki, da je genom sestavljen iz $k$ genov, nam tako zaporedje $n$ baznih parov celotnega genoma razpade na $k$ podzaporedij baznih parov (genov), pri čemer so podzaporedja (geni) lahko različnih dolžin.

Glede na to, da posamezen gen predstavlja navodila za celično dinamiko, ki pripelje do izražanja neke lastnosti organizma, bi lahko gen imenovali tudi za **skupek navodil celice** ali **celični program**. Ker genom vsebuje več genov posledično lahko genom smatramo za množico celičnih programov.

Poglejmo si nekaj konkretnih podatkov o genomu in genu. Človeški genom po trenutnih ocenah (podatek datira na leto 2017) vsebuje nekje od 19.000 do 20.000 genov [6], sestavljen pa je iz približno $3,1 * 10^9$ baznih parov (*bp*) [7]. Podobno dolžino genoma imajo tudi miši. Genom znane bakterije *Escerichie coli*, ki se uporablja za mnoge biološke eksperimente, meri približno $5 * 10^6$ *bp*, genom vinske mušice (lat. *drosophilla melanogaster*) pa približno $139 * 10^6$ *bp*.

## 1.4 Motivacija za procesiranje v DNA okolju

V pričujočem razdelku skušamo bralca motivirati k razumevanju biološkega sistema kot sistema kemijskih reakcij, ki bi jih človek lahko izkoristil v svoje namene na različnih področjih, nenazadnje tudi na področju računalništva. Doslej smo pri obravnavi DNA sistemov naleteli na naslednje zanimive *inheretne* (vgrajene, nerazdružno povezane) biološke značilnosti:

- *kodiranje podatkov* na osnovi kodnega nabora $B = \{A, T, C, G\}$: z vidika procesiranja nam DNA kot medij ponuja možnost štiristanjskega kodiranja podatkov;

- *velika gostota zapisa*: DNA medij nam ponuja za nekaj razredov večjo gostoto hranjenih podatkov od današnjih tehnologij trajnega pomnjenja podatkov;
- *redundanten zapis podatkov*: vsi podatki v DNA zapisu so pomnjeni redundantno (dvokratno) preko komplementa v baznem paru;
- *paralelizem izvajanja*: kemijski procesi omogočajo hkratno obdelavo vezanih baz na večih segmentih;
- *linearni naslovni prostor*: v DNA mediju so naslovi baznih parov določljivi, kar pomeni, da se na bazne pare lahko sklicujemo z naslovi - adresami; z njimi je določena lega baznega para v kodnem zapisu;
- *celični program*: posamezen gen v genomu vsebuja navodila za celično dinamiko in ekspresijo posameznih kemijskih zvrsti (npr. proteinov); delovanje celice je torej programirano z nukleotidnimi pari genoma;

V nadaljevanju pričujočega razdelka predstavimo še nekaj dodatnih motivacijskih momentov, ki nas vabijo k prehodu na biološko računalništvo.

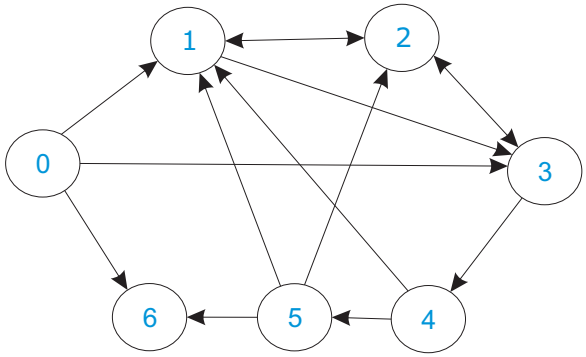
### 1.4.1 Adlemanov zgled procesiranja z DNA

Leonard M. Adleman svoj znani primer procesiranja z DNA predstavi l.1994 in ta še danes predstavlja primer načina reševanja NP-polnega problema, ki ga lahko z visokim paralelizmom rešujemo dosti bolj efektivno v domeni kemijskih reakcij, kot smo tega vajeni v svetu računalništva [8], [9], [10]. Omenjeni zgled predstavlja potrditev smiselnosti koncepta pristopa (angl. *proof of the concept*) k procesiranju v DNA medijih.

#### Predstavitev problema Hamiltonovih poti

Adleman si za zgled reševanja NP-polnega problema zastavi obhod Hamiltonovih poti. Predpostavimo, da smo soočeni s problemom obiska $n$ mest (vozlišč v grafu), med katerimi vodi poljubno število enosmernih ali dvosmernih povezav. Startno in ciljno mesto sta določeni, vsa ostala mesta pa bi si želeli obiskati natanko enkrat. Ni nujno, da za podani graf takšna rešitev sploh obstaja, je pa njeno iskanje pogojeno z veliko algoritmično časovno kompleksnostjo. Slednja strmo raste s povečevanjem števila mest ali vozlišč v grafu. Algoritem za reševanje problema naj bi zgolj potrdil ali zavrnil obstoj enkratnega obhoda grafa, kar predstavlja rešitev.

Na sliki 1.3 je predstavljen graf konkretnega problema, ki si ga je za reševanje zadal Adleman. Obstoj enkratnega obhoda sedmih vozlišč je pogojen s štirinajstimi povezavami (dvosmerne povezave štejejo dvojno), vprašanje, ki si ga Adleman zastavi pa je, če vodi pot iz vozlišča 0 v vozlišče 6 preko vseh vozlišč, pri čemer vsako vozlišče obiščemo natanko enkrat. Zaradi majhnega števila vozlišč lahko rešitev razberemo vizuelno iz samega grafa, vodi pa zaporedoma preko vozlišč 0, 1, 2, 3, 4, 5 in 6.




Slika 1.3: Konkretni primer problema obhoda grafa, ki ga v DNA procesnem mediju rešuje Adleman.

### Prostorska potratnost kot alternativa časovni potratnosti

Osnovna ideja Adlemana ob reševanju problema je bila, da potencialno *časovno* ali *procesno potratnost* algoritma za pregled vseh možnih poti po grafu pri reševanju problema nadomesti s *prostorsko potratnostjo* algoritma.

Vhod v Adlemanov algoritem je usmerjeni graf $G$ z $m$ povezavami in $n$ vozlišči, od katerih je vozlišče $v_{in}$ vhodno in $v_{out}$ izhodno vozlišče (izvor in ponor poti). Algoritem iskanja rešitve (obhoda grafa) Adleman sestavi iz naslednjih korakov:

I. generiraj potencialne poti (zaporedja vozlišč) v grafu $G$ naključno in v velikih količinah;

II. zavrži vse poti, ki nimajo ustreznega ponora in izvora;

III. zavrži vse poti, ki ne obiščejo natanko $n$ vozlišč;

IV. ponavljaj za vsako od $n$ vozlišč ($i = 1, ..., n$):

- preglej vse poti in posamezno pot zavrži, če ne vsebuje trenutnega opazovanega vozlišča $i$;

V primeru, da nam po izvedbi algoritma ostane še kaka pot, le ta predstavlja rešitev problema obhoda grafa vozlišč.

Ob hipotetični predpostavki, da obstaja procesna platforma, ki omogoča za izvedbo vsake operacije predhodno navedenega algoritma le konstantno časovno odvisnost ($O(k)$), lahko ugotovimo, da je časovna kompleksnost algoritma kot celote linearno odvisna od števila $n$ ($O(n)$). Koraki (I), (II) in (III) so namreč po predpostavki izvedljivi v konstantni časovni odvisnosti ($O(k)$), $n$ ponovitev koraka (IV) pa nam časovno odvisnost poveča na ($O(n)$).

Z vidika implementacije je ključna ustrezna izvedba koraka (I). Zadolžen je za generiranje dovolj velike množice možnih poti (sekvence vozlišč), da le

ta vsebuje vsaj eno potencialno rešitev. Za izvedbo koraka (I) sta potrebna masivni paralelizem in nedeterminizem. Adleman vse korake algoritma realizira na osnovi DNA kemijskih reakcij.

### Kodirna osnova izvedbe algoritma

Adlemanova rešitev temelji na kodiranju imena vsakega vozlišča v grafu s sekvenco 20 baznih parov DNA iz množice nukleotidov $B = \{A, T, C, G\}$. *Kodna imena vozlišč* se zgenerirajo naključno, relativno velika dolžina posameznega kodnega imena v primerjavi s številom vozlišč pa naj bi omogočila doseganje različnosti kodnih imen. Predpostavimo, da je nedeterministični postopek dodeljevanja kodnih imen dodelil mestoma 2 in 3 naslednji kodni imeni:

$$s_2 = TATCGGATCG\ GTATATCCGA, \tag{1.1}$$

$$s_3 = GCTATTCGAG\ CTTAAAGCTA. \tag{1.2}$$

Zaradi preglednosti kodnih imen ostalih mest ne bomo navajali. S tem smo pridobili kodna imena vozlišč, ne pa še poimenovanja povezav med vozlišči (poti), ki kot zaporedje tvorijo cilj iskalnega algoritma. Adleman posamezne dele poti poimenuje tako, da za vsako povezavo, ki vodi iz vozlišča $i$ v vozlišče $j$, stakne drugo polovico kode izvornega in prvo polovico kode ponornega vozlišča. S tem pridobi *kodno ime povezave* ali dela poti med vozliščema $i$ in $j$. Povrh vsega staknjen kodni niz negira z Watson-Crickovim komplementom, ki definira negacije po izrazu

$$\overline{T} = A, \overline{A} = T, \overline{C} = G, \overline{G} = C. \tag{1.3}$$

Kodno ime povezave med vozliščema 2 in 3 tako zapišemo z izrazom

$$e_{2\to 3} = CATATAGGCT\ CGATAAGCTC, \tag{1.4}$$

kodo poti iz vozlišča 3 v vozlišče 2 pa z izrazom

$$e_{3\to 2} = GAATTTCGAT\ ATAGCCTAGC. \tag{1.5}$$

### Procesna izvedbe algoritma

Procesno izvedbo algoritma je Adleman izvedel na osnovi sledečih kemijskih procesov:

- encimi (ligaze) sestavljajo kodna imena vozlišč v poljubne DNA sekvence, s čimer posredno dobimo prostor hipotetičnih poti - rešitev; s tem realizira korak algoritma (I); princip kodiranja povezav izhaja iz dejstva, da je vsaka vijačnica sestavljena iz dveh verig; če uspemo na prvo verigo zakodirati kodno ime posameznega vozlišča, se nam na drugi verigi DNA vijačnice avtomatsko generira kodno ime posameznega prehoda;
- filtirna metoda, ki poskrbi za korak (II) je polimerazna verižna reakcija (angl. *polymerase chain reaction - PCR*); zmožna je zavreči vse zlepke kodnih imen poti, ki nimajo ustreznega izvora in ponora;

- filtirna metoda, ki poskrbi za korak (III) je elektroforeza (angl. *gel electrophoresis* - GE); zmožna je zavreči vse zlepke kodnih imen poti, ki nimajo ustrezne dolžine 140 baznih parov;

Po procesni izvedba algoritma imamo torej ustrezno dolge vijačnice DNA (dolžine sovpadajo z vsoto nizov kodnih imen poti), istočasno pa sta začetek in konec verige "ustrezna".

#### Slabosti realizacije Adlemanovega eksperimenta

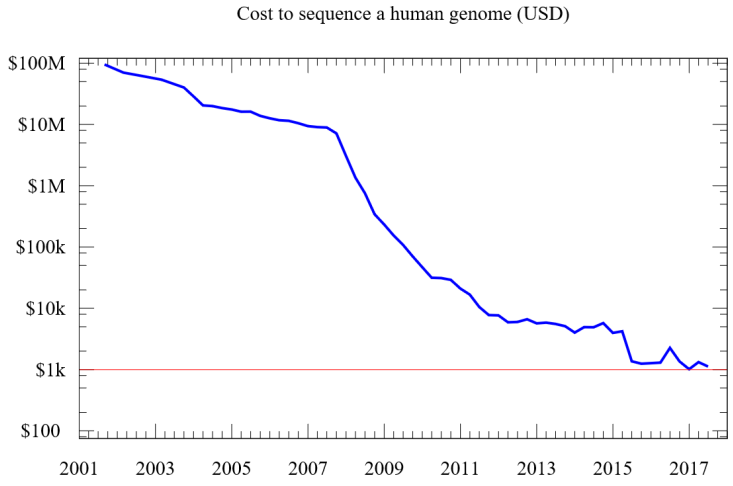
Prva slabost omenjenega pristopa je v enormni porabi prostora za reševanje velikih problemov. Za reševanje problema z 200 vozlišči, bi po [10] potrebovali več kot $3 * 10^{25}$ kg DNA materije, kar presega maso našega planeta. Druga slaba lastnost koncepta je ta, da sinteza DNA ni deterministična, temveč stohastičnega značaja in z velikostjo števila iteracij verjetnost pojavljanja napak v posameznih kemijskih procesih narašča. Več o sami izvedbi Adlemanovega ekserimenta si bralec lahko prebere v delih [8] in [9].

### 1.4.2 Sekveniranje, anotiranje in sintetiziranje genoma

Pod pojmom *sekveniranja* ali *sekvenciranja* genoma ali DNA (angl. *DNA sequencing*) smatramo postopek določanja celotnega nukleotidnega zaporedja v opazovani DNA vijačnici. Leta 2001 J. C. Venter objavi prvi osnutek zaporedja ali „karto“ človeškega genoma [11], od te objave pa so metode sekveniranja (oslikovanja genomov) izredno hitro napredovale. Pri tem so pripomogli tudi zmogljivi računalniški sistemi, s pomočjo katerih se ob sekveniranju pridobljeni podatki iz posameznih kromosomov analizirajo in sestavljajo v dokončno kodno zaporedje genoma. Dober zgodovinski opis razvoja sekveniranja, problematike velikih količin podatkov (angl. *big data*), tekmovanja med različnimi raziskovalnimi skupinami in vloge računalnikov pri sekveniranju najdemo opisan v slovenskem prevodu Venterjeve knjige z naslovom Genom mojega življenja [12]. Današnja cena oslikave človeškega genoma je odvisna od kvalitete naprave, na kateri je oslikava izvedena ter s tem posledično dosežene natančnosti ali zanesljivosti (angl. *accuracy*) in je velikostnega razreda 1.000 USD [13]. Cene oslikav genskega materiala se v letu 2018 gibljejo v velikostnem razredu 1 USD na $10^6$ *bp* [14]. Hitro padanje cene oslikave človeškega genoma je ponazorjeno na sliki 1.4. Funkcijo na sliki 1.4 poimenujemo tudi za Carlsonovo krivuljo ali Carlsonov zakon, ki je ekvivalent Mooreovega zakona, ki velja na področju računalništva. Carlsonov zakon napoveduje, da bo podvojevanje tehnologij DNA sekvenciranja v smislu cene in zmogljivosti potekalo vsaj tako hitro, kot je potekalo podvojevanje zmogljivosti v računalništvu po načelih Mooreovega zakona.

Carlson predicted that the doubling time of DNA sequencing technologies (measured by cost and performance) would be at least as fast as Moore's law

Pod pojmom *anotiranja* genoma smatramo postopek določanja genomskih podzaporedij, ki določajo gene in njihovo funkcionalno identifikacijo (določanje funkcij posameznih genov genoma). Povedano drugače anotiranje skuša povezati




Slika 1.4: Potek padanja cene oslikave človeškega genoma (slika je povzeta neposredno po viru [15]).

del genomskega zapisa - gena z njegovo biološko funkcijo. Če v današnjem času sekveniranje sodi že v komercialno dejavnost, anotiranje še ni prispelo do komercializacije storitev in na tem področju raziskave še intenzivno potekajo. S problematiko anotiranja se ukvarja področje *bioinformatike*.

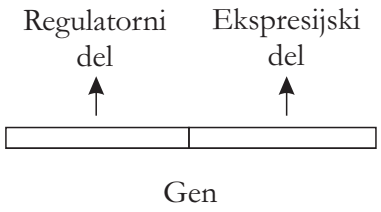
V zadnjem desetletju se je razvilo tudi področje umetnega sintetiziranja genoma, ki je dosegljivo preko komercialnih ponudnikov. Na ta način imamo z računalniškega vidika dostop do možnosti *pisanja* (sintetiziranja) in *branja* (sekveniranja) genoma, kar z računalniškega vidika lahko enačimo s pisanjem in branjem podatkov. Oba postopka sta sicer počasna (praktično neuporabna za potrebe današnjega načina delovanja računalnikov), a vendarle komercialno dosegljiva.

## 1.5 Gensko izražanje

V pričujočem razdelku predstavimo osnove *genske ekspresije* ali *genskega izražanja*. Predhodno smo že omenili, da posamezni gen smatramo za *celični program*, ki definira del celične dinamike. Pri opisovanju genskega izražanja se bomo zopet omejili samo na osnove zanimive z vidika biološkega procesiranja.

Do izražanja gena pride v odvisnosti od prisotnosti ali odsotnosti ustreznih kemijskih zvrsti v celici imenovanih *transkripcijski faktorji*, kjer se opazovani gen kot del genoma nahaja. Rezultat izražanja je tvorba novih kemijskih zvrsti. Posamezen gen je razdeljen na dva dela in sicer na *regulatorni del* in na *ekspresijski del*, kot je prikazano na sliki 1.5. Ekspresijski del imenujemo tudi

za *kodirajoče zaporedje proteina*.




Slika 1.5: Delitev gena na regulatorni in ekspresijski del.

### 1.5.1 Transkripcijski faktorji

Transkripcijski faktorji se delijo na dve skupini in sicer na *aktivatorje* in na *represorje* (imenovane tudi za *inhibitorje*). Njihovi pomeni so sledeči:

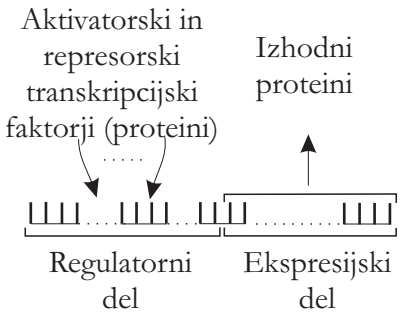
- aktivatorji sprožijo ekspresijo gena, če se preko kemijskih reakcij vežejo na ustrezna *vezalna mesta* regulatornega dela gena; če do te vezave ne pride, se tudi sama ekspresija ne sproži; glede na povedano smatramo aktivatorje za pospeševalce ekspresije gena;
- represorji zavrejo ekspresijo gena, če se preko kemijskih reakcij vežejo na ustrezna *vezalna mesta* regulatornega dela gena; če do te vezave ne pride, ekspresija poteka nemoteno; glede na povedano smatramo represorje za zaviralce ekspresije gena;

Z vezavo transkripcijskih faktorjev na vezalna mesta regulatornega dela tako pride do aktivacije (pospeševanja) ali represiranja (zaviranja) izražanja gena. Vsebino ali produkte izražave definira ekspresijski del gena. Tako v vlogi transkripcijskih faktorjev, kot tudi v vlogi izhodnih kemijskih zvrsti ali produktov (rezultatov izražanja ali ekspresije gena), običajno nastopajo proteini. Grafična ponazoritev izražanja gena je prikazana na sliki 1.6.

Ena od pomembnih zahtevanih lastnosti transkripcijskih faktorjev je njihova *ortogonalnost*, ki predvideva, da med samimi transkripcijskimi faktorji in med njimi in okoljem ne more priti do kemijskih reakcij (do njihovega medsebojnega interagiranja), ki bi spremenile njihovo vlogo v sistemu proženja ekspresije. Transkripcijske faktorje delimo na naravne in umetne. Med slednjimi so trenutno aktualni TAL efektorji. Umetno dodani transkripcijski faktorji ne smejo vplivati na osnovne celične procese gostiteljske celice.

### 1.5.2 Izražanje gena

Ko je izražanje gena omogočeno (aktivirano ali nerepresirano) s strani transkripcijskih faktorjev, je tvorba izhodnih produktov pogojena z DNA zapisom v ekspresijskem delu gena. Ta del DNA zapisa obravnavamo kot navodila za



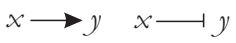

Slika 1.6: Grafična predstavitev izražanja gena.

tvorbo produkta ali ciljnega proteina. Tvorba podukta poteka v zaporedju faz *transkripcije* in *translacije*. Pomena faz sta sledeča:

- v fazi transkripcije se ekspresijski del gena prepiše v mRNA; intenzivnost prepisovanja je neposredno pogojena z vezavo transkripicijskega faktorja na vezalna mesta, posredno pogojena pa s prisotnostjo ali koncentracijo transkripcijskega faktorja v mediju;
- v fazi translacije se mRNA prepiše v ciljni protein ali produkt; intenzivnost prepisovanja je odvisna zgolj od količine mRNA;

### 1.5.3 Vhodno izhodne relacije izražanja gena

Predpostavimo, da kot transkripcijski faktor nastopa samo en protein vrste $x$ in kot končni produkt samo en protein vrste $y$. V tem primeru običajno rečemo, da protein $x$ bodisi aktivira, ali bodisi represira izražanje proteina $y$, grafično pa to ponazorimo, kot je predstavljeno na sliki 1.7.

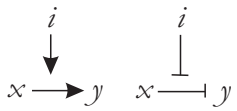



Slika 1.7: Protein $x$ aktivira izražanje proteina $y$ (levi del slike), protein $x$ represira izražanje proteina $y$ (desni del slike).

Proteini tako po eni plati predstavljajo predpogoj za gensko ekspresijo, hkrati pa so tudi njen rezultat. Tako proteini kot rezultati ekspresije lahko predstavljajo transkripcijske faktorje za druge gene, zato genska izražanja in njihove medsebojne povezanosti poimenujemo za *gensko regulatorna omrežja*. V domeni biotehnoloških metod se kot izhodna proteina mnogokrat uporabljata GFP ali RFP (zeleni ali rdeči fluorescentni protein), ki jih najlažje zaznamo kot rezultat ekspresije. Slednje je možno le takrat, če izhodni protein ne nastopa v vlogi transkripcijskega faktorja za drug gen.

Domena vhodnih transkripcijskih faktorjev je lahko tudi bolj kompleksna. Primer kompleksnejše domene so *induktorski proteini*. Le ti se vežejo na preostale vhodne transkripcijske faktorje. V primeru enega vhodnega transkripcijskega faktorja imamo ob vezavi induktorja sledeče možnosti:

- aktivator $x$ je aktiven (aktivira ekspresijo proteina $y$) samo v primeru, če je nanj vezan induktor $i$ (slednje grafično ponazorimo, kot je predstavljeno na levem delu slike 1.8);
- represor $x$ je aktiven (represira ekspresijo proteina $y$) samo v primeru, če nanj ni vezan induktor $i$ (slednje grafično ponazorimo, kot je predstavljeno na desnem delu slike 1.8);




Slika 1.8: Protein $x$ aktivira izražanje proteina $y$ le ob vezanem induktorju $i$ (levi del slike), protein $x$ represira izražanje proteina $y$ le ob nevezanem induktorju $i$ (desni del slike).

### 1.5.4 Koncentracije kemijskih zvrsti kot nosilci podatkov

V predhodnem razdelku smo omenili zgled vpliva prisotnosti proteina $x$ na izražanje proteina $y$. Omenjena relacija temelji na *koncentracijah molekul* obeh opazovanih proteinov. Koncentracijo najpogosteje merimo s številom molekul na prostorninsko enoto ($\frac{mol}{L}$).

Z računalniškega vidika se nam samo po sebi ponuja razmišljanje o visokih in nizkih koncentracijah proteinov, podobno kot smo bili tega vajeni pri napetostnih nivojih v integriranih vezjih. Aktivatorsko relacijo med proteinoma $x$ in $y$ bi tako lahko opisali natančneje s stavkom, da *visoka koncentracija* aktivatorskega proteina $x$ z nekim časovnim zamikom $\Delta T$ povzroči *visoko koncentracijo* izhodnega proteina $y$, represorsko pa s stavkom, da *visoka koncentracija* represorskega proteina $x$ z nekim časovnim zamikom $\Delta T$ povzroči *nizko koncentracijo* izhodnega proteina $y$. Tako ekspresije izhodnih proteinov običajno opazujemo skozi njihove koncentracije. Več o tem v nadaljevanju poglavja.

### 1.5.5 Cilji sintezne biologije v domeni genskega izražanja

Spoznali smo dovolj osnov o genskem izražanju, da lažje definiramo primarna *cilja* in *izziva* sintezne biologije na tem področju. Le ta sta sledeča:

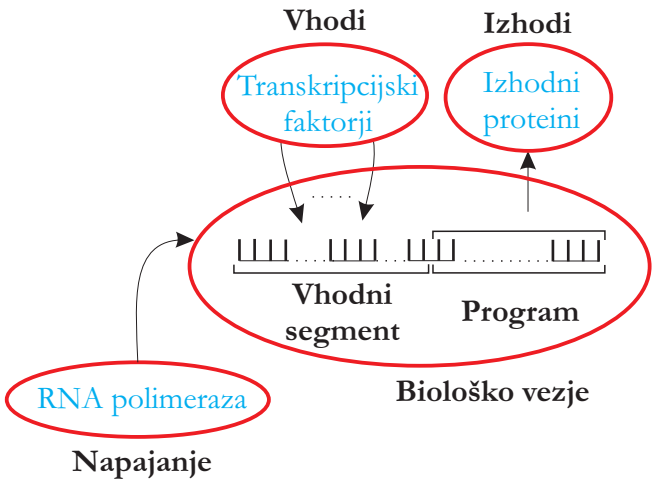
- analiziranje in spreminjanje regulatornega dela gena: z manipulacijo z regulatornim delom določamo, kateri proteini bodo vršili nalogo vplivnih

transkripcijskih faktorjev, preko katerih in koliko vezalnih mest se bodo vezali itd.

- analiziranje in spreminjanje ekspresijskega dela gena: z manipulacijo z ekspresijskim delom določamo ciljne produktne proteine kot rezultate ekspresije;

### 1.5.6 Računalniški pogled na ekspresijo gena

Na koncu razdelka o genskem izražanju vpeljemo primerjavo med mehanizmi genske ekspresije z mehanizmi računalništva. Transkripcijski faktorji tako predstavljajo vhodni segment, izhodni proteini izhodni segment in gen biološko vezje. Ekspresijski del gena lahko enačimo s pojmom programa. Primerjava obeh mehanizmov je predstavljena na sliki 1.9. Posebno vlogo na omenjeni sliki igra doslej neomenjena RNA polimeraza, brez katere ne steče vezava transkripcijskih faktorjev na vezalna mesta, zato jo pomensko lahko izenačimo z napajanjem kot osnovnim regulacijskim momentom današnjih elektronskih računalnikov. Prisotnost RNA polimeraze se izkazuje z vezavo na *promotor* (angl. *promoter*), ki je del regulatornega dela gena.




Slika 1.9: Primerjava mehanizmov genske ekspresije s poznanimi mehanizmi računalništva.

Glede na povedano biološki sistemi na nivoju delovanja DNA vsebujejo podobne sestavne dele kot računalniki. Ti deli so vhodni segment, program in izhodni segment.

## 1.6 Trajno pomnjenje podatkov neposredno v DNA zapisu

V pričujočem razdelku predstavimo možnosti trajnega pomnjenja podatkov v DNA zapisu. Večina pričujočega razdelka je povzeta neposredno po viru [16].

Trajno pomnjenje podatkov *neposredno* v DNA zapisu temelji na *kodiranju* ciljnih dvovrednostnih podatkov (bitov) v DNA zapis, ki bazira na kodnem naboru simbolov $\{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$. Simboli predstavljajo oznake nukleotidov Adenina, Citozina, Gvanina in Timina, ki so osnovni sestavni deli DNA zapisa. Ob hipotetični predpostavki, da imamo *algoritem kodiranja* posameznih kodov kodnega nabora $\{0, 1\}$ v kode kodnega nabora $\{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$, je končni korak do želenega DNA zapisa in s tem trajnega pomnjenja le še postopek umetne sinteze DNA zapisa, ki je danes že tržno dosegljiv za sprejemljivo ceno.

V strokovni literaturi najdemo kar nekaj primerov izvedb pomnjenja podatkov neposredno v zapisu DNA. Večina avtorjev izvedb poudarja naslednje prednosti tovrstnega pomnjenja:

- izredno visoka gostota pomnjenja (največja možna gostota pomnjenja v DNA zapisu je 455EB (exabytes)/gram (1EB = $10^{18}$B = 1.000PB = $10^{6}$TB) [17]),
- izredno dolga življenska doba oziroma stabilnost zapisa v ambientalnem okolju (angl. *long lifespan in low-maintenance environment*) [18],
- izredno dolga predvidena življenska doba formata zapisa, saj DNA zapis postaja *de-facto* standard biološkega zapisa vseh živih organizmov [19],
- izredno velika energetska učinkovitost zapisa (angl. *energy efficiency*).

Avtorji realizacij po drugi plati priznavajo problematičnost zrelosti tehnologije pisanja in branja DNA zapisa. Oba postopka sta navkljub rastoči zanesljivosti namreč še vedno dolgotrajna, cenovno nekonkurenčna in s tem primerna samo za hrambo podatkov za daljša časovna obdobja, kot smo jih vajeni danes (npr. za več desetletij, ali celo stoletij (angl. *century-scale archives*)). Glede na povedano je tovrsten način hrambe primeren predvsem za dolgotrajno pomnjenje velikih količin (angl. *large-scale*) arhivskih podatkov (angl. *archival storage*), pri čemer je ekonomsko opravičeno le zelo redko (angl. *infrequently accessed*) in istočasno parcialno dostopanje do vsebinsko zaključenih enot podatkov, ki so na pomnilnem mediju DNA shranjene v fizičnem sosedstvu, kar pri postopkih branja in pisanja favorizira aplikacije z zaporednim dostopom do podatkov (angl. *sequential access applications*) [17].

### 1.6.1 Algoritem kodiranja pri neposrednem pomnjenju v zapisu DNA

Ob predpostavkah, da je dolžina DNA zapisa lahko poljubno dolga in da je zapis lahko sestavljen iz poljubnega zaporedja nukleotidov (kodov iz kodnega nabora

| $b_{2*i}$ | $b_{2*i+1}$ | $k_i$ |
|---|---|---|
| 0 | 0 | $A$ |
| 0 | 1 | $T$ |
| 1 | 0 | $G$ |
| 1 | 1 | $C$ |

Tabela 1.1: Preslikovalna tabela parov bitov v nukleotidni zapis ($i = 0, ..., n-1$).

$K = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$), smo soočeni s problemom določitve načina kodiranja logičnih vrednosti 0 in 1.

Predpostavimo, da je želena hranjena podatkovna vsebina v nam znanem digitalnem svetu zakodirana v zaporedje sode dolžine $2 * n$ bitov

$$(b_0, b_1, ..., b_{2n-1}), \tag{1.6}$$

želimo pa jo prestaviti v zaporedje kodnega sveta nukleotidov dolžine $l$

$$(k_0, k_1, ..., k_{l-1}). \tag{1.7}$$

Glede na to, da je ciljni kodni nabor $K$ po zalogi vrednosti dvakrat večji od kodnega nabora $\{0, 1\}$, se nam sama od sebe ponuja logika preslikave predstavljena v preslikovalni tabeli 1.1, kjer po dva sosednja bita $(b_{2*i}, b_{2*i+1})$ iz bitnega zaporedja preslikujemo v en nukleotid $k_i$ $(i = 0, ..., n-1)$ po pravilu iz tabele. Tako dobimo kompaktnejši zapis z vidika manjšega števila uporabljenih pomnilniških entitet pomnilnega medija.

### 1.6.2 Churcheva realizacija pomnjenja v zapisu DNA

V delu [17] Church s soavtorji opiše primer hrambe podatkov neposredno v zapisu DNA, v katerega zakodirajo HTML verzijo knjige z naslovom *Regenesis: How Synthetic Biology Will Reinvent Nature and Ourselves* [20] v obsegu 53.426 besed in 11 JPEG slik. Skupna količina podatkov obsega zaporedje $5,27$ Mbitov. Avtorji se pri realizaciji odločijo za manj ekonomičen pristop kodiranja, kot je bil predstavljen v predhodnem razdelku, in sicer za kodiranje logične vrednosti 0 izbirajo med nukleotidoma **A** in **C**, za kodiranje logične vrednosti 1 pa med nukleotidoma **T** in **G**. S tem se skušajo izogniti štirim ali več sosednjim ponovitvam posameznega nukleotida in uravnovešajo frekvenco podzaporedij **CG** na ciljnem kodnem zaporedju. Omejitvi izhajata iz značilnosti postopkov sinteze DNA (pisanja ciljnega zapisa), sekveniranja DNA (branja ciljnega zapisa) in same stabilnosti DNA molekule. Na ta način realizacija izgubi na gostoti zapisa, ker je dolžina podatkovnega zapisa v bitni obliki enaka dolžini nukleotidnega zapisa.

Z arhitekturnega vidika realizacije pomnjenja avtorji podatkovni zapis razdelijo na 96 bitov dolga podzaporedja, vsakemu podzaporedju pa dodajo 19 bitni naslov, ki označuje lego podatkovnega podzaporedja v celotnem zaporedju zapisa. Istočasno zaradi narave pisanja in branja podzaporedju dodajo še 44

kontrolnih bitov. Tako pridejo do segmentiranega načina hranjenja podatkov v segmentih po 159 bitov (ciljnih nukleotidov) in končne dolžine zapisa

$$(96 + 19 + 44) * 54.898 = 8.728.782, \tag{1.8}$$

kjer je zapis segmentiran v 54.898 nukleotidnih podzaporedij (angl. *chunks*), vsota dolžin vseh segmentov zapisa pa 8.728.782 nukleotidov. Podatkovni (kodirani) del je sestavljen iz $(96 * 54.898)$ 5.270.208 nukleotidov. V primerjavi z načinom kodiranja opisanem v razdelku 1.6.1, so avtorji na ta način drastično zmanjšali gostoto zapisa.

### 1.6.3 Goldmanova realizacija pomnjenja v zapisu DNA

V delu [18] Goldman s soavtorji opiše primer hrambe podatkov neposredno v zapisu DNA, v katerega zakodirajo 739KB podatkovne vsebine vseh 154 Shakespearovih sonetov v ASCII obliki, temu pa dodajo še en PDF članek, eno JPEG datoteko, izsek iz govora Martina Luthra Kinga iz leta 1963 v MP3 formatu in izsek o kodiranju v formatu Huffmanovega koda. Pri svojem modelu načina zapisa jih vodijo naslednje smernice:

- podobno kot Church se avtorji zaradi lažjega obvladovanja ciljnega DNA zapisa (njegovega sintetiziranja - pisanja in sekveniranja - branja) odločijo za segmentiran zapis ciljne vsebine v večje število krajših DNA podzaporedij;
- v postopku kodiranja v nukleotidni zapis se odpovedo možnosti sosednosti enakih nukleotidov v ciljnem zapisu zaradi zanesljivosti branja in zaradi tega, ker naravna DNA takšne ponovitve običajno vsebuje; s tem je iz ciljnega nukleotidnega zapisa po kodiranju razvidno, da gre za umetno sintetizirano DNA;
- podatkovne podnize kodirajo večkratno v prekrivnem načinu, s čimer dosežejo štirikratno redundanco zapisa (vsak podatek je shranjen v štirih verzijah) in s tem posredno večjo zanesljivost branja in večjo življensko dobo zapisa;

Z arhitekturnega vidika realizacije pomnjenja avtorji ciljni nukleotidni zapis oblikujejo po naslednjih korakih in pravilih:

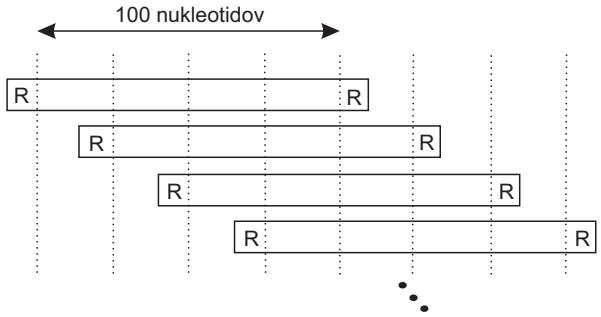
- posamezni bajt izvirnega podatkovnega zapisa prekodirajo v zaporedje petih trovrednostnih entitet - tritov (angl. *trits*) z zalogo vrednosti $K_1 = \{0, 1, 2\}$; slednje si navkljub manjši zalogi vrednosti novega kodnega nabora lahko privoščijo zaradi narave izvirnega podatkovnega zapisa, ki temelji na podmnožici razširjenega ASCII kodnega nabora (8 bitov pokrije 256 možnih različnih vrednosti, 5 tritov pa le 243 različnih vrednosti);
- posamezni trit prekodirajo v nukleotid po kodirni shemi predstavljeni v tabeli 1.2 (levi stolpec tabele označuje predhodno zapisani nukleotid $n_i$, desni del tabele pa izbiro nukleotida $n_{i+1}$ glede na vrednost trita), ki zagotavlja, da v ciljnem zapisu DNA ne bo enakih sosednjih nukloetidov,

| | 0 | 1 | 2 |
|---|---|---|---|
| A | C | G | T |
| C | G | T | A |
| G | T | A | C |
| T | A | C | G |

Tabela 1.2: Kodirna shema tritov v nabor nukleotidov [18].

- izhodiščni podatkovni nabor dolžine 739KB (757.051B) tako prevedejo na zaporedje 3.785.255 ($5 * 757.051$) nukleotidov; slednje razdelijo na podzaporedja po 100 nukleotidov, vsakega od njih pa opremijo z dodatnimi 17 kontrolnimi (CRC) in adresnimi nukleotidi; tovrstnih podzaporedij je 37.853,
- v zadnji fazi vsak podniz dolžine 117 nukleotidov sintetizirajo v prekrivnem načinu (sosednja podniza se prekrivata v 75 nukleotidih) v štirikratni redundanci (glej sliko 1.10); s tem pridejo do 151.348 nukleotidnih podzaporedij dolžine 117 nukleotidov, tako da je vsota vseh podazporedij 17.707.716 nukleotidov; sosednji prekrivni podnizi so kodirani na osnovi Cricks Watsonovega komplementa.

Realizacija ni najbolj kompaktna z vidika gostote zapisa, po drugi plati pa ima vgrajeno redundanco in se ciljni zapis zaradi omejitev evidentno razlikuje od DNA nastale potom naravne evolucije v živečem gostitelju.




Slika 1.10: Prekrivna redundanca sintetizirane DNA [18]. Sosednji prekrivni podnizi so kodirani na osnovi Cricks Watsonovega komplementa ($A = \overline{T}, T = \overline{A}, C = \overline{G}, G = \overline{C}$). R segment predstavlja redundančne podatke neprekrivne narave.

### 1.6.4 Narava pomnjenja v zapisu DNA

V predhodnih razdelkih smo opisali dva primera realizacije pomnjenja podatkov neposredno v DNA zapisu. Pri tem moramo poudariti, da oba zgleda predsta-

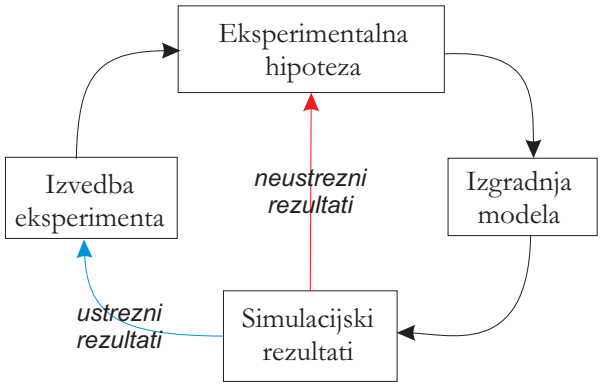
vljata primera *in-vitro* pomnjenja v kontroliranem laboratorijskem okolju, kjer DNA medija ne gosti živeči gostitelj (celica). Povedano drugače, medij hrambe ni osnova za življenje in ekspresijo kemijskih zvrsti nekega živečega organizma. Avtorji se *in-vitro* koncepta poslužujejo zaradi *nedefinirane ekspresije* sintetizirane DNA v gostitelju in možnosti njenih *mutacij* skozi evolucijo gostitelja.

## 1.7 Modeliranje genskega izražanja v bioloških sistemih

Eksperimentiranje na nivoju genskega izražanja bioloških organizmov lahko opišemo z naslednjim zaporedjem korakov:

- izdelava hipoteze o ekspresiji posameznega gena;
- priprava eksperimenta (sintetiziranje umetne DNA itd.);
- izvedba eksperimenta v gostitelju;
- analiza rezultatov;

Tovrstni eksperimenti so dolgotrajni in dragi, nenazadnje že z vidika potrebne drage biotehnološke opreme. V zadnjem desetletju se *računalniško modeliranje genske ekspresije* pri preverjanju hipotez izkaže kot nepogrešljiva pomoč, ki omogoča izločanje izvedb eksperimentov, pri katerih s simulacijo ne moremo potrditi pravilnosti hipoteze o izidu eksperimenta. Na sliki 1.11 je prikazan cikel predhodno naštetih korakov izvedbe eksperimentov ob vključitvi faze modeliranja. Z modro barvo je ponazorjen pristop k izvedbi eksperimenta ob simulacijski potrditvi pravilnosti hipoteze, z rdečo pa pristop k spremembi hipoteze, zaradi ovržbe pravilnosti prvotne hipoteze na osnovi simulacijskih rezultatov.




Slika 1.11: Cikel korakov za uspešno izvedbo eksperimentov genskega izražanja ob vključitvi modeliranja.

### 1.7.1 Vrste modelov genskega izražanja

Modele za potrebe simulacij genskega izražanja delimo na dve osnovni skupini in sicer na *kvalitativne* in *kvantitativne modele*. Prvi zgolj kvalitativno predstavijo relacije med vhodnimi in izhodnimi kemijskimi zvrstmi (npr. *protein x aktivira izražanje proteina y*), drugi pa omenjene relacije ovrednotijo tudi v kvantitativnem (matematičnem) smislu.

Kvantitativne modele delimo na skupini *deterministič*nih in *stohastičnih modelov*. Prvi temeljijo na navadnih diferencialnih enačbah (angl. *ordinary differential equations*), drugi pa na kemijski glavni enačbi (angl. *master equation approach*). Matematične osnove obeh vrst modelov so predstavljene v delu [21].

Za potrebe kvantitativnih modelov, ki temeljijo na matematičnih opisih kemijskih reakcij, potrebujemo vrsto kemijskih enačb s katerimi opišemo kemijske procese, posredno s tem pa vrsto vrednosti spremenljivk, s katerimi opišemo kemijske reakcije vezave transkripcijskih faktorjev, razgradnje kemijskih zvrsti, transkripcije in translacije itd. v opazovanem biološkem sistemu. Med omenjene spremenljivke sodijo hitrost transkripcije, hitrost translacije, osnovno izražanje gena, Hillov koeficient, koeficient aktivacije, hitrost razgradnje posameznega proteina, hitrost razgradnje mRNA itd. V domeni našega dela nas omenjene spremenljivke ne bodo zanimale, njihov obširnejši opis pa bralec najde v delu [21].

### 1.7.2 Vzorčni primeri simulacij ekspresije

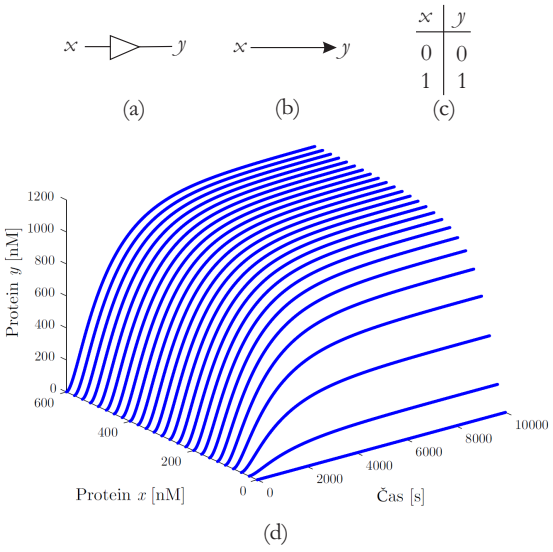
V pričujočem razdelku predstavimo nekaj vzorčnih simulacijskih rezultatov na osnovi modela ekspresije v gostitelju *bakteriofag* $\lambda$. Simulacijski rezultati so pridobljeni na osnovi determinističnih modelov. Več informacij o načinu postavitve modelov bralec lahko najde v viru [21], odkoder so tudi povzete slike simulacijskih rezultatov.

#### Ekspresija gena v domeni delovanja logičnih primitivov

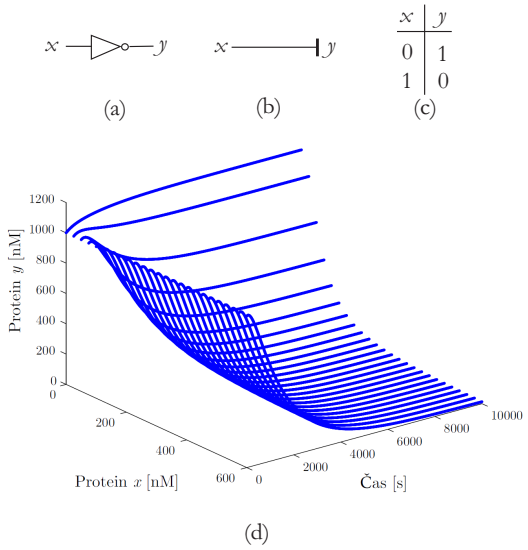
Na slikah 1.12, 1.13, 1.14 ter 1.15 so po vrsti prikazani hipotetični simulacijski odzivi gostitelja, ki ob ustrezni interpretaciji „visokih“ in nizkih „koncentracij“ predstavljajo logične funkcije identitete (gonilnika), negacije, AND in NOR vrat.

Iz simulacijskih rezultatov s slik lahko razberemo naslednje značilnosti simulirane dinamike ekspresije:

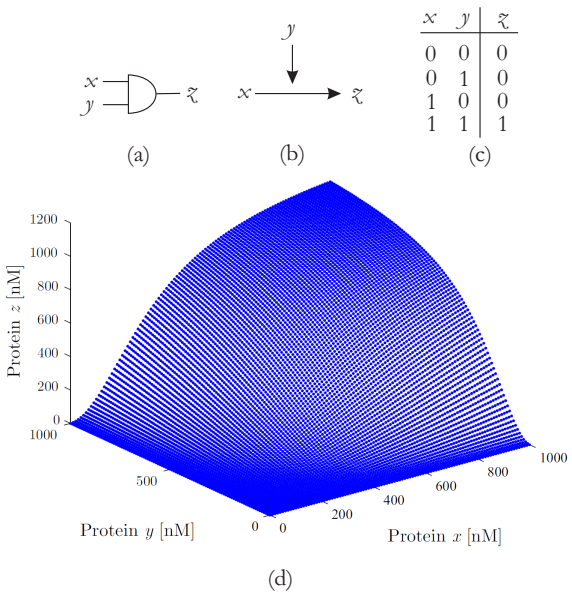
- odziva modelov gonilnika in negacije izkazujeta nelinearnost odziva skozi čas;
- odziv prvih dveh modelov je v primerjavi z odzivom elektronskih vezij dosti bolj počasen (velikostnega reda 1.000 sekund);
- relacije med vhodi in izhodi spremljamo na osnovi koncentracij posameznih kemijskih zvrsti; ni nujno, da so maksimumi koncentracij vseh ke-




Slika 1.12: Logična shema (a), biološka shema (b), pravilnostna tabela (c) in simulacijski odziv (d) gonilnika v gostitelju *bakteriofag* $\lambda$.




Slika 1.13: Logična shema (a), biološka shema (b), pravilnostna tabela (c) in simulacijski odziv (d) negatorja v gostitelju *bakteriofag* $\lambda$.




| $x$ | $y$ | $z$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Slika 1.14: Logična shema (a), biološka shema (b), pravilnostna tabela (c) in simulacijski odziv (d) AND vrat v gostitelju *bakteriofag* $\lambda$.

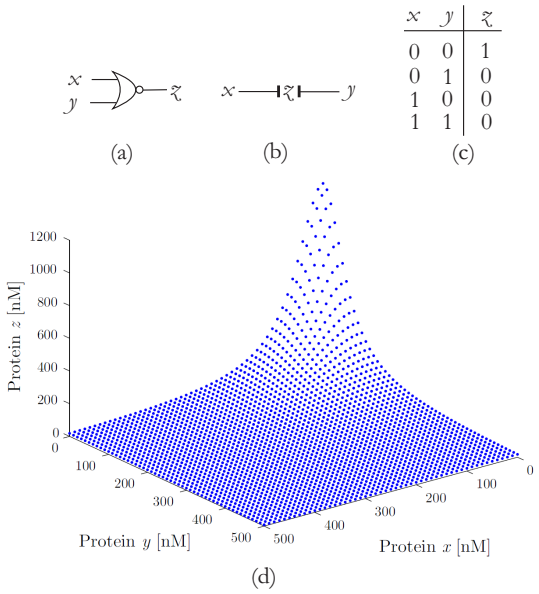
mijskih zvrsti enaki; na slikah simulacijskih odzivov AND in NOR vrat je maksimum izhodne koncentracije npr. večji od vhodnih dveh;

### Ekspresija gena v domeni bistabilnosti sistema

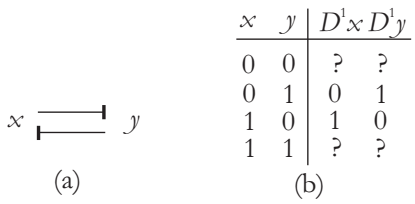
Pod bistabilnim sistemom smatramo vezje, ki je ujeto v eno od dveh možnih stanj in ni zmožno preklopa v drugo stanje, kar predstavlja osnovo za dinamično pomnjenje v bioloških sistemih. Biološka shema in pravilnostna tabela tovrstnega sistema sta prikazani na sliki 1.16. Posamezno stanje sistema je določeno s parom vrednosti spremenljivk $(x, y)$, pri čemer zapis $D^1x$ v pravilnostni tabeli predstavlja vrednost spremenljivke $x$ na naslednjem diskretnem časovnem koraku. Iz biološke sheme je razvidno, da z obojestransko represijo ob predpostavki, da smo na začetku simulacije v stanju $(0, 1)$ ali $(1, 0)$ (imamo „visoko“ natanko eno koncentracijo od obeh opazovanih vstopajočih kemijskih zvrsti), dosegamo ohranjanje stanja. Še več, kemijska zvrst z visoko koncentracijo preko obojestranske represije posredno aktivira sama sebe. Omenjeno značilnost poimenujemo za *posredno samoaktivacijo*.

### Ekspresija gena v domeni RS pomnilne celice

V predhodnem razdelku smo govorili o bistabilnem sistemu, ki je v praksi neuporaben za pomnjenje, ker ni zmožen preklapljati med obema stabilnima stanjema.

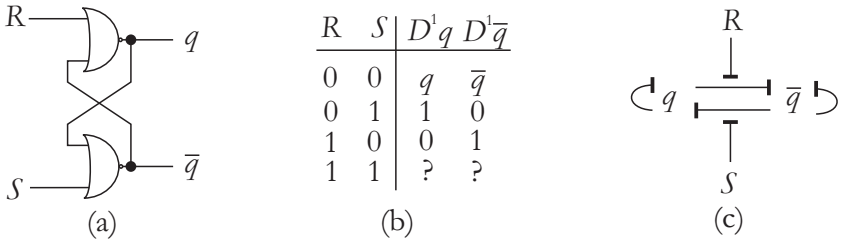



Slika 1.15: Logična shema (a), biološka shema (b), pravilnostna tabela (c) in simulacijski odziv (d) NOR vrat v gostitelju *bakteriofag* $\lambda$.




Slika 1.16: Biološka shema (a) in pravilnostna tabela (b) biološke realizacije bistabilnega sistema.

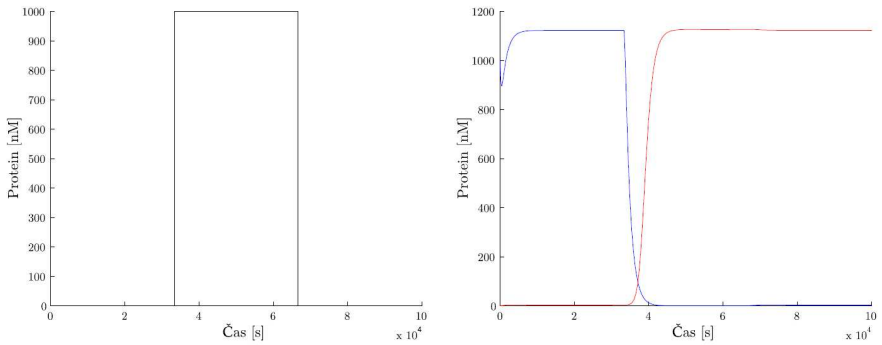
Za namene možnosti preklopa vpeljemo induktorska vhoda $R$ in $S$, ki se ob prisotnosti po vrsti vežeta na represorska vhoda $q$ in $\overline{q}$. Z vpeljavo notacije vhodov $(q, \overline{q})$ ponazorimo, da je le ena od obeh zvrsti lahko v stanju visoke koncentracije. Logična shema, pravilnostna tabela in biološka shema so prikazane na sliki 1.17. Pri tem ima vhod $R$ pomen `RESET` signala, vhod $S$ pa `SET` signala, kot smo jih vajeni v digitalni elektroniki. Na sliki 1.17(c) se nahajajo tudi samorepresivne povezave vhodov $(q, \overline{q})$. Pomen *samorepresije* vhodov je v tem, da v stanju pomnjenja enega stanja (ene visoke koncentracije) lahko pride do prekomerne formacije te kemijske zvrsti, kar bi lahko v nadaljevanju onemogočilo „hiter“ preklop v njeno nizko koncentracijo; slednje imenujemo za *samoregulacijo* proteinov. Samorepresija mora biti „šibkejša“ od represije drugega proteina.




| $R$ | $S$ | $D^1q$ | $D^1\overline{q}$ |
|---|---|---|---|
| 0 | 0 | $q$ | $\overline{q}$ |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | ? | ? |

Slika 1.17: Logična shema (a), pravilnostna tabela (b) in biološka shema (c) RS pomnilne celice v gostitelju *bakteriofag* $\lambda$.

Na sliki 1.18 je prikazan odziv sistema na preklopni signal. Na levem delu slike je prikazan preklopni impulz (npr. signal $R$), na desnem delu slike pa izvedba preklopa, pri čemer gre koncentracija proteina $q$ v nizko stanje, koncentracija proteina $\overline{q}$ pa v visoko. Prehod koncentracije proteina $q$ v nizko stanje je posledica razgradnje ali degradacije proteina, na katerega sicer lahko vplivamo tudi dodatno s posebnimi kemijskimi procesi. Iz slike je razvidno, da je odziv sistema relativno počasen. Pomembna je tudi ugotovitev, da navkljub umiku signala $R$ koncentracija proteina $\overline{q}$ ostaja visoka.



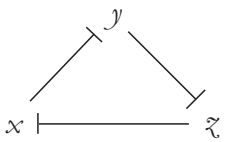

Slika 1.18: Simulacijski odziv preklopa v RS pomnilni celici v gostitelju *bakteriofag* $\lambda$. Levi del slike predstavlja preklopni impulz (npr. signal $R$), desni del slike pa odziv sistema ali prehod proteina $q$ v nizko stanje.

### Ekspresija gena v domeni oscilatorja

Ciklične oscilacije so tako v domeni računalništva, kot tudi v domenah sistemske biologije in medicine eden od ključnih dejavnikov delovanja sistemov. Veliko raziskav v današnjem času je usmerjenih v tako imenovane *cirkadiane ritme*, ki

v bioloških organizmih povzročajo oscilacije in s tem posledično različna stanja in odzivanja bioloških sistemov v odvisnosti od biološke ure.

Na sliki 1.19 je predstavljen umetni biološki oscilator, zasnovan na osnovi represije, zato ga imenujemo tudi za *represilator*. Sestavljen je iz treh vhodnih kemijskih zvrsti - proteinov $x$, $y$ in $z$, ki se med seboj ciklično represirajo, pri čemer mora biti sistem v začetku v stanju, v katerem je le eden od proteinov v stanju visoke koncentracije. Ob predpostavki, da je to protein $x$, bo le ta represiral izražanje proteina $y$, posledično bo nizka koncentracija proteina $y$ omogočala izražanje proteina $z$, slednji pa bo posledično represiral izražanje proteina $x$. Tako pridemo v nekem času, ki bi ga lahko imenovali za periodo procesiranja $\Delta t$, v novo stanje sistema, v katerem bo prisotna le visoka koncentracija proteina $z$. Glede na povedano lahko ugotovimo, da oscilatorni represilator deluje po principu *posredne samorepresije* ali *samorepresije z zakasnitvijo*.




(a)

| $x$ | $y$ | $z$ | $D^1x$ | $D^1y$ | $D^1z$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |

(b)

Slika 1.19: Biološka shema (a) in pravilnostna tabela (b) oscilatornega represilatorja v gostitelju *bakteriofag* $\lambda$.

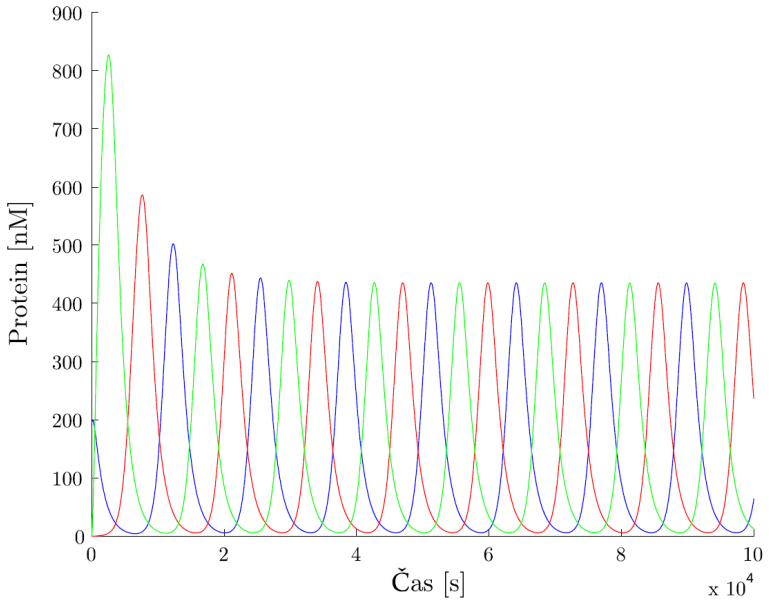
Na sliki 1.20 je prikazan simulacijski odziv oscilatornega represilatorja, pri čemer so funkcijski poteki koncentracij treh proteinov ponazorjeni z različnimi barvami.

Iz biološke sheme na sliki 1.19 lahko sklepamo, da morajo biti oscilatorni represilatorji sestavljeni iz *lihega števila* ciklično represirajočih se proteinov. Pri tem je pomembno tudi to, da so slednji med seboj ortogonalni.

Primarni cilj umetnih bioloških represilatorjev je v samodejnosti vzdrževanja oscilacij kemijskih zvrsti, na pojavnost katerih lahko vežemo proženje željenih funkcij bioloških sistemov. Pri tem se seveda poraja vprašanje, ali za tovrstne umetno prožene oscilacije lahko poljubno določamo njihove amplitude in frekvence.

### Zaključek

V predhodnjih razdelkih smo si ogledali možnosti uporabe genske ekspresije za potrebe procesiranja (izvedbe logičnih funkcij), potrebe pomnjenja in potrebe zagotavljanja oscilacij. Razvoj realizacij tovrstnih *umetnih bioloških vezij* poteka izredno hitro. Tudi Slovenija je na tem področju izredno aktivna. Tako so na Kemijskem inštitutu v Ljubljani pod vodstvom prof. dr. Romana Jerale že v letu 2014 v sesalskih celicah realizirali vseh 16 možnih dvovhodnih dvovrednostnih logičnih funkcij [22].




Slika 1.20: Simulacijski odziv oscilatornega represilatorja v gostitelju *bakteriofag* $\lambda$ ob ustreznem začetnem stanju z le eno visoko koncentracijo enega treh proteinov.

## 1.8 Metrike za snovanje bioloških procesnih sistemov

V pričujočem razdelku se usmerimo na *biološke procesne sisteme* ali *biološke računalnike*, katerih osnovne zmožnosti izdelave decizije, pomnenja in oscilacij smo spoznali v prejšnjem razdelku o modeliranju. Za potrebe njihovega snovanja vpeljemo metrike, ki so podobne metrikam za snovanje elektronskih digitalnih vezij. Le te so po viru [21] sledeče:

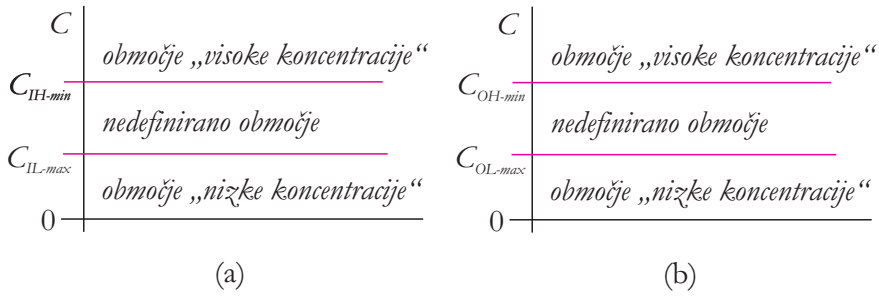
- nivoji koncentracij nosilcev signala sorodni napetostnim nivojem signalov v elektroniki,
- šumne meje,
- širina prepovedanega področja
- preklopni časi,
- vejanje izhodov (angl. *fanout*),
- maksimalna frekvenca delovanja,
- čas osveževanja pri pomnjenju.

Več o pomembnejših od naštetih metrik povemo v naslednjih razdelkih.

### 1.8.1 Določitev logičnih vrednosti nivojev koncentracij

V predhodnjih razdelkih smo že govorili o „nizkem“ in „visokem“ stanju koncentracije, tega pa nismo konkretizirali v obliki formalnega zapisa. Koncentracije snovi na nivoju genskega izražanja opazujemo tako na vhodni strani (strani vstopajočih transkripcijskih faktorjev), kot tudi na izhodni strani (strani izstopajočih produktov ali rezultatov genske eskpresije).

Kot osnovo za klasifikacijo koncentracij v logične vrednosti povzemimo ponazoritev s slike 1.21, povzete po viru [21]. Na levem delu slike je prikazan




Slika 1.21: Intepretacija logične vrednosti posamezne vhodne (a) in izhodne (b) koncentracije $C$ opazovane kemijskih zvrsti.

primer klasifikacije posamezne koncentracije na vhodnem nivoju in na desnem delu slike klasifikacije posamezne koncentracije na izhodnem nivoju. Pomeni posameznih oznak na sliki so sledeči:

- $C_{IH-\min}$ predstavlja najnižjo koncentracijo vhodne kemijske zvrsti, ki bo na nivoju transkripcije še igrala vlogo visokega stanja;
- $C_{IL-\max}$ predstavlja najvišjo koncentracijo vhodne kemijske zvrsti, ki bo na nivoju transkripcije še igrala vlogo nizkega stanja;
- $C_{OH-\min}$ predstavlja najnižjo koncentracijo izhodne kemijske zvrsti, ki bo v domeni vhoda v naslednjo transkripcijo še igrala vlogo visokega stanja;
- $C_{OL-\max}$ predstavlja najvišjo koncentracijo izhodne kemijske zvrsti, ki bo v domeni vhoda v naslednjo transkripcijo še igrala vlogo nizkega stanja;

Predpostavimo, da imamo na vhodnem nivoju $n$ vstopajočih kemijskih zvrsti in na izhodnem nivoju $m$ izstopajočih kemijskih zvrsti. V tem primeru moramo tako vsem $n$ vhodnim kemijskim zvrstem določiti njim lastne vrednosti $C_{IH-\min i}$ in $C_{IL-\max i}$ $(i = 1, ..., n)$, kot tudi vsem $m$ izhodnim kemijskim zvrstem določiti njim lastne vrednosti $C_{OH-\min j}$ in $C_{OL-\max j}$ $(j = 1, ..., m)$. Povedano drugače dopuščamo možnost, da koncentracije različnih kemijskih zvrsti interpretiramo v logične vrednosti na osnovi različnih kriterijev.

Za opazovano kemijsko zvrst glede na njeno koncentracijo tako lahko ocenimo glede na predhodnjo klasifikacijo njeno „visoko“ stanje ali logično enico, njeno „nizko“ stanje ali logično ničlo, ali pa koncentraciji pripišemo nahajanje v nedefiniranem področju, na osnovi katerega ne moremo določiti logične vrednosti nosilne koncentracije. Iz slednjega razloga si želimo, da potek koncentracije skozi nedefinirano področje zgolj čimhitreje prehaja, se pa v njem ne zadržuje.

### 1.8.2 Obravnava šuma

Šumu se v realnih sistemih v praksi ne moremo izogniti, zato moramo imeti vzpostavljene mehanizme za njegovo nevtralizacijo. Na nivoju interpretacije logične vrednosti opazovane kemijske zvrsti na osnovi njene koncentracije to dosegamo z zadoščanjem veljavnosti relacij [21]

$$C_{OH-\min} < C_{OH} - N_H, \tag{1.9}$$

$$C_{OL-\max} > C_{OL} + N_L, \tag{1.10}$$

pri čemer $N_H$ predstavlja maksimalni šum v visokem stanju, $N_L$ maksimalni šum v nizkem stanju, $C_{OH}$ poljubno koncentracijo v visokem stanju ter $C_{OL}$ poljubno koncentracijo v nizkem stanju. Enaka izraza bi lahko zapisali tudi za tretma koncentracij vhodnih kemijskih zvrsti. Zadoščanje navedenim relacijam je eden od kriterijev za postavitev mej za definicijo prepovedanih področij $]C_{IL-\max}, C_{IH-\min}[$ in $]C_{OL-\max}, C_{OH-\min}[$. Glede na povedano je razvidno, da šum koncentracije ne sme „premakniti“ iz definiranega področja koncentracije v nedefinirano.

### 1.8.3 Določitev časa preklopa

Čas preklopa smatramo za čas, ki je potreben za prehod nosilnega signala iz enega stanja v drugega. Povedano drugače gre za čas, ki je potreben ob izvedbi operacije, da sistem preide iz enega stabilnega stanja v drugo stabilno stanje. Na nivoju opazovanja koncentracije kemijskih zvrsti smatramo za prehajanje prehod koncentracije preko nedefiniranega področja. Na tem mestu ločujemo med časom vzpona (angl. *rise time*) $t_r$ in med časom padca (angl. *fall time*) $t_f$. Časa se lahko med seboj drastično razlikujeta. Nenazadnje do tega lahko pride zaradi počasne razgradnje kemijskih zvrsti, na katere nimamo vpliva. Slednje vpliva predvsem na prehod iz visokega stanja v nizko stanje koncentracije.

## 1.9 Osnove snovanja bioloških procesnih sistemov

Z vidika snovanja bioloških sistemov s funkcijo procesiranja se postavlja vprašanje, kako se lotiti sestavljanja kompleksnejših modularnih bioloških sistemov ob željeni logični prevajalni funkciji sistema. Če se omejimo zgolj na domeno genske ekspresije, se nam na nivoju snovanja ponujata dve osnovni možnosti in sicer *snovanje željenega regulatornega dela*, s čimer definiramo vhodni segment

in *snovanje ekspresijskega dela*, s čimer definiramo izhodni segment. Bolj specifično snovanje obsega sledeče dejavnosti:

- določanje transkripcijskih faktorjev;
- določanje izhodnih produktov;
- določanje kinetičnih parametrov, ki definirajo poteke posameznih reakcij;
- določanje vezalnih mest za transkripcijske faktorje itd.;

Trenutno aktualni pristopi k snovanju s sledeči:

- poseganje po knjižnjicah modularnih bioloških procesnih entitet; primer knjižnice je repozitorij bioloških gradnikov BioBricks organizacije iGEM [23];
- avtomatizacija postopkov gradnje na osnovi knjižnic modularnih bioloških gradnikov;
- ostali alternativni pristopi k snovanju bioloških struktur (npr. z uporabo genetskih algoritmov [24]);

## 1.10 Formalni zapisi bioloških procesnih sistemov

Eden od ključnih momentov za hiter razvoj sintezne in sistemske biologije je vpeljava enotnega formalnega zapisovanja bioloških sistemov. Eden od najbolj razširjenih jezikov za tovrstno zapisovanje je SBML označevalni jezik (angl. *system biology markup language*). V njem lahko poljuben obvladljiv biološki sistem opišemo formalno v obliki modela s posebno označevalno notacijo kemijskih reakcij. Tovrstne zapise v SBML formatu razume večina modelirnih in simulacijskih programskih orodij, s pomočjo katerih postanejo zasnove umetnih bioloških sistemov hitro prenosljive in se da preko simulacij pravilnost njihovega delovanja tudi verificirati.

## 1.11 Programska orodja za simulacijo dinamike bioloških sistemov

V zadnjem desetletju so se razširila programska orodja za simulacijo dinamike v bioloških sistemih. Na spletni strani navedeni v viru [25] najdemo pregled orodij z njihovimi osnovnimi funkcionalnostmi, bolj natančen opis orodij pa v viru [26]. V novembru l.2018 lahko tovrstnih programskih orodij naštejemo približno 300.

## 1.12 Povzetek poglavja

V pričujočem poglavju smo se usmerili predvsem na ekspresijo posameznega gena v gostitelju. Danes potekajo intenzivne raziskave tudi na področjih analize *metabolnih* in *signalnih poti* v celicah gostitelja.

Z vidika genske ekspresije lahko dobre plati biološkega procesiranja strnemo v naslednje alinee:

- visoka stopnja paralelizma in s tem velika hitrost procesiranja;
- izredno visoka miniaturizacija pomnjenja;
- možnost procesiranja v fluidnih okoljih (krvi, vodi itd.);
- energetska nepotratnost;

Kot osnovni slabi lastnosti genske ekspresije lahko navedemo njeno počasnost in veliko občutljivost na šum iz okolja.

# Literatura


[1] "Sistemska in sintezna biologija." http://web.bf.uni-lj.si/bi/biokemija/SBD/Docs/sinbiol.pdf, Maj 2015.

[2] "Sistemska biologija." http://www.zrss.si/bzid/geni/pdf/baebler-clanek.pdf, November 2017.

[3] "Deoksiribonukleinska kiselina." https://hr.wikipedia.org/wiki/Deoksiribonukleinska_kiselina, November 2017.

[4] M. Ridley, *Genom: biografija človeške vrste*. Učila International, 2002.

[5] J. D. Watson and A. Berry, *DNK - Skrivnost življenja*. Modrijan založba d.o.o., 2007.

[6] "Human Genome." https://en.wikipedia.org/wiki/Human_genome, November 2017.

[7] N. Cristianini and M.W.Hahn, *Introduction to Computational Genomics - A Case Study Approach*. Cambridge University Press, UK, 2007.

[8] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, 1994.

[9] L. M. Adleman, "Computing with DNA," *Scientific American*, vol. 279, pp. 54–62, 1998.

[10] C. S. Calude and G. Paun, *Computing with cells and atoms*. Taylor and Francis Inc., 2001.

[11] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, and et al., "The Sequence of the Human Genome," *Science*, vol. 291, pp. 1304–1351, Feb. 2001.

[12] J. C. Venter, *Genom mojega življenja*. Modrijan založba d.o.o., 2009.

[13] "Cost per genome." https://www.genome.gov/images/content/costpergenome_2017.jpg, November 2017.

[14] "DNA sequencing." https://en.wikipedia.org/wiki/DNA_sequencing, November 2017.

[15] "Cost to sequence human genome." https://en.wikipedia.org/wiki/DNA_sequencing, November 2018.

[16] M. Moškon, N. Zimic, and M. Mraz, "Realizacija dvojiškega pomnjenja v preprostih bioloških sistemih," *Elektotehniški vestnik*, vol. 83, no. 4, pp. 194–200, 2016.

[17] G. M. Church, Y. Gao, and S. Kosuri, "Next-Generation Digital Information Storage in DNA," *Science*, vol. 337, p. 1628, 2012.

[18] 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, pp. 77–80, 2013.

[19] C. Bancroft, T. Bowler, B. Bloom, and C. Clelland, "Long-term storage of information in DNA," *Science*, vol. 293, no. 5536, pp. 1763–5, 2001.

[20] G. M. Church and E. Regis, *Regenesis: How Synthetic Biology Will Reinvent Nature and Ourselves*. Perseus Books Group, USA, 2012.

[21] M. Moškon, *Modeli in metrike dinamike preklopa v enostavnih bioloških sistemih za potrebe računalniških struktur prihodnosti*. PhD thesis, Univerza v Ljubljani, 2012.

[22] R. Gaber, T. Lebar, A. Majerle, B. Šter, A. Dobnikar, M. Benčina, and R. Jerala, "Designable dna-binding domains enable construction of logic circuits in mammalian cells," *Nature Chemical Biology*, vol. 10, no. 3, pp. 203–208, 2014.

[23] "Registry of standard biological parts." http://parts.igem.org/Main_Page, November 2017.

[24] M. Stražar, M. Mraz, N. Zimic, and M. Moškon, "An adaptive genetic algorithm for parameter estimation of biological oscillator models to achieve target quantitative system response," *Natural Computing*, vol. 13, pp. 119–127, 2014.

[25] "SBML Software Matrix." http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix, November 2017.

[26] "SBML Software Summary." http://sbml.org/SBML_Software_Guide/SBML_Software_Summary, November 2017.