ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

Katedra přírodovědných oborů

Predikce a vizualizace proteinových sekundárních struktur na genomových datech

Prediction and visualization of protein secondary structures on genomic data

Diplomová práce

Studijní program: Biomedicínská a klinická technika
Studijní obor: Přístroje a metody pro biomedicínu

Vedoucí práce: Mgr. Jan Pačes, Ph.D.

Mgr. et Bc. Petr Adámek

Kladno 2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	**Adámek**	Jméno:	**Petr**	Osobní číslo:	**368212**
Fakulta:	**Fakulta biomedicínského inženýrství**				
Garantující katedra:	**Katedra přírodovědných oborů**				
Studijní program:	**Biomedicínská a klinická technika**				
Studijní obor:	**Přístroje a metody pro biomedicínu**				

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Predikce a vizualizace proteinových sekundárních struktur na genomových datech

Název diplomové práce anglicky:

Prediction and visualization of protein secondary structures on genomic data

Pokyny pro vypracování:

Jeden z přístupů vyhledávání doposud neznámých členů proteinových rodin je definovat různé typy sekvenčních charakteristik a motivů, které jsou obsaženy ve všech nebo alespoň většině členů takových rodin. Pro efektivní prohledávání velkého množství potenciálních cílů je nutné tyto charakteristiky predikovat automatizovaně a přehledně prezentovat. Téma této diplomové práce je predikovat struktury, které jsou součástí antivirového proteinu Bst2 a vytvořit přehledné grafické rozhraní, pomocí kterého bude možno identifikovat potenciální geny Bst2 v nově sekvenovaných organizmech, kde zatím nebyla jejich přítomnost prokázána. Na zadaném úseku DNA nejprve predikovat ORF (open reading frames, otevřené čtecí rámce), ty přeložit do proteinů a v nich určovat tři druhy sekundárních proteinových struktur - transmembránové oblasti, coiled-coil a GPI modifikaci - pomocí standardních algoritmů (viz reference). Výstupem práce bude nástroj pro predikci sekvence proteinu Bst2 a jiných příbuzných protivirových genů, případně identifikace Bst2 genů v nových organizmech.

Seznam doporučené literatury:

[1] Krogh A, Larsson B, von Heijne G, Sonnhammer EL, Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes, J Mol Biol, číslo 305, 2001, 567-580 s.
[2] Lupas A, Van Dyke M, Stock J., Predicting coiled coils from protein sequences, Science, číslo 252, 1991, 1162-1164 s.
[3] Eisenhaber B, Bork P, Eisenhaber F., Prediction of potential GPI modification sites in proprotein sequences, J Mol Biol, číslo 292, 1999, 741-758 s.
[4] Krchlíková V, Fábryová H, Hron T, Young JM, Koslová A, Hejnar J, Strebel K, Elleder D, Antiviral Activity and Adaptive Evolution of Avian Tetherins, J Virol, ročník 12, číslo 94, 2020, e00416-20 s.

Jméno a příjmení vedoucí(ho) diplomové práce:

Mgr. Jan Pačes, Ph.D.

Jméno a příjmení konzultanta(ky) diplomové práce:

Ing. Martin Otáhal, Ph.D.

Datum zadání diplomové práce: **1.10.2020**
Platnost zadání diplomové práce: **31.10.2022**

doc. Ing. Lenka Lhotská, CSc.
podpis vedoucí(ho) katedry

prof. MUDr. Jozef Rosina, Ph.D., MBA
podpis děkana(ky)

PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci s názvem „Predikce a vizualizace proteinových sekundárních struktur na genomových datech“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně dne 13.05.2021 …..……..……...

Mgr. et Bc. Petr Adámek

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu své diplomové práce panu Mgr. Janu Pačesovi, Ph.D. a také panu MUDr. Danielu Ellederovi, Ph.D. za jejich odbornou pomoc při řešení práce. Další poděkování patří též panu Ing. Martinu Otáhalovi, Ph.D. jako konzultantovi.

ABSTRAKT

Predikce a vizualizace proteinových sekundárních struktur na genomových datech

Diplomová práce popisuje nástroj pro hledání kandidátních genů rodiny genů *tetherin*/*bst2* u obratlovců. Hledání je založeno na třech různých charakteristických motivech nalézaných u všech členů této genové rodiny. Definované proteinové motivy jsou: transmembránové oblasti, coiled-coil struktura a GPI (glykofosfatidylinositol) modifikace/kotva. Nástroj vyvinutý v této práci bere jako vstup oblast DNA a automatizuje vyhledávání a identifikaci motivů, načež vytváří přehledný textový a grafický výstup ve formě HTML stránky. Vzhledem k relativně široké definici charakteristických motivů je nástroj určen jako pomocník pro odborníka a je zaměřen na prezentaci komplexních výstupů ze standardních algoritmů uživatelsky přívětivým způsobem.

Klíčová slova

predikce, proteinové sekundární struktury, tetherin, Python, HTML

ABSTRACT

Prediction and visualization of protein secondary structures on genomic data

The diploma thesis describes a tool for a search for candidate genes of the *tetherin*/*bst2* gene family in vertebrates. The search is based on three different characteristic motifs found in all members of this gene family. The defined protein motifs are: transmembrane regions, coiled-coil structure and GPI (glycophosphatidylinositol) modification/anchor. The tool developed in this work takes as an input region of a DNA and automates search and identification of motifs, than creates clear textual and graphics output in a form of HTML page. Because of a relatively broad definition of characteristic motifs the tool is meant as a helper for an expert and is focused on presenting complex outputs from standard algorithms in a user friendly way.

Keywords

prediction, protein secondary structures, tetherin, Python, HTML

Obsah

1 Úvod

Je obecně známo, že se lidstvu již podařilo sekvenovat genom řady organismů (tj. určit pořadí „písmen“-bazí genetického kódu) včetně člověka samotného. Jedním z nových problémů, které návazně nastaly, je vyhledat a identifikovat geny v takovýchto sekvencích a jedním z přístupů hledání genů je využití znalosti typických motivů ve formě sekundárních proteinových struktur v rámci dané genové rodiny.

Tato práce si nemůže klást za úkol vytvoření nástroje na hledání všech potenciálních genů u všech organismů, ale zaměřuje se na jednu konkrétní rodinu genů u obratlovců. Konkrétně se jedná o identifikaci rodiny genů, kódující jeden ze známých protivirových proteinů, obecně tzv. tetherin, jehož typickým reprezentantem je např. protein Bst2. Práce by měla též vytvořit nástroj využitelný pro hledání těchto genů i v případě řady druhů obratlovců, kde zatím jejich přítomnost nebyla prokázána, ale obvykle se předpokládá.

Doposud je práce při hledání těchto genů značně manuální a zdlouhavá. Cílem práce je vytvořit nástroj, který by jejich hledání významně zautomatizoval a poskytl přehlednou grafickou reprezentaci výsledku jejich hledání.

Způsob hledání genů v této práci napodobuje do značné míry přirozené procesy v přírodě. Nejprve jsou na zadaném úseku DNA a jeho komplementárním vláknu vyhledány otevřené čtecí rámce (ORF, open reading frames), ty jsou přeloženy do sekvence aminokyselin v proteinech. Následně lze pomocí sady standardních algoritmů na těchto aminokyselinových sekvencích vyhledávat sekundární proteinové struktury typické pro gen *bst2* a příbuzné protivirové geny. Těmito proteinovými strukturami jsou transmembránové oblasti, coiled-coil struktury a GPI modifikace.

2 Přehled současného stavu

2.1 Základní molekulárně biologické poznatky

2.1.1. Biologické souvislosti ohledně tetherinů

Tato práce se zabývá vyhledáváním genů pro protein (přesněji glykoprotein) Bst2, což je jeden z tzv. tetherinů. Bst2 (bone marrow stromal cell antigen 2) nebo též CD317 (cluster of differentiation 317) nebo též HM1.24 byl nejprve popsán u člověka a dalších savců [1,2,3]. Dle [1] byl objeven u všech zástupců savců studovaných do té doby na přítomnost genů pro tetherin/Bst2. Později byly ortology genu *bst2* nalezeny u dalších obratlovců, např. ptáků, aligátorů, želv a různých zástupců ryb. [2]

Původně byl objeven jako faktor zodpovědný za defekt v uvolňování HIV-1 vironů (postrádajících gen *vpu*) z CD4+ buněk tj. T-lymfocytů a makrofágů a hromadění těchto virových částic na jejich povrchu. [1]

Gen pro Bst2 je indukovatelný interferonem typu I (IFN I) produkovaným jako odpověď na virovou infekci. IFN I stimuluje v buňkách „protivirový stav“, který vede ke spuštění řady signálních drah, zřejmě jednak tlumících některé geny ale také spouštějících expresi řady genů odpovědných za obranu hostitele proti virové infekci. Genů indukovaných interferonem v savčích buňkách jsou řádově stovky [3]. Obecně se tyto protivirové geny a jejich produkty označují jako restrikční faktory, mezi tyto geny patří též geny pro tetherin/Bst2. [1,2]

I bez indukce IFN I je tetherin exprimován na povrchu některých buněk, např. především B-lymfocytů, plasmatických buněk a dendritických buněk. Jeho produkce může být také snadno zvýšena u myeloidních buněk a dalších leukocytů účinkem např. prozánětlivých cytokinů. [1]

Mechanismus účinku proteinu Bst2 spočívá v zachycení (angl. tether = přivázat, připoutat) obalených virových částic k povrchu buňky, z níž se tyto viriony snaží uvolnit. Jedná se o zacílení na strukturu, v tomto případě na membránu, kterou virus nemůže dostatečně pozměnit mutací. Díky tomuto mechanismu účinku může být zacílen na řadu druhů obalených virů. Existuje obrana těchto virů, ta spíše spočívá v zamezení vnitrobuněčné přípravy tetherinu a jeho umístění do cytoplasmatické membrány, např. nasměrováním produktu protivirového genu k degradaci. [1,2]

Sekvenční analýzy ukázaly poměrně vysoký pozitivní selekční tlak v evoluci savců vpřípadě vývoje tetherinů a to zvláště oblastech, které jsou snadněji napadnutelné antagonisty produkovanými viry. [1,2]

Tetherin je poměrně malý membránový protein typu II tvořený 181 aminokyselinovými zbytky a o molekulární váze mezi 29 až 33 kDa v závislosti na stupni glykosylace. Má nezvyklou topologii s oběma konci upevněnými v membráně. N-konec obsahuje krátký nitrobuněčný úsek a dále transmembránovou strukturu ukotvující jej jedním

průchodem v membráně buňky, C-konec obsahuje tzv. GPI (glycosyl-phosphatidylinositol) kotvu. [1,3]

Tyto dvě transmembránové struktury jsou navzájem spojeny coiled-coil strukturou (angl. coil = cívka, spirála). Jedná se o dva polypeptidové řetězce ve formě α-helixu (pravotočivé šroubovice) stabilizované několika disulfidovými vazbami – tyto dva polypeptidové řetězce se kolem sebe obtáčejí a jsou tak k sobě přiloženy dvě molekuly tetherinu. Tetheriny tak vytvářejí dimer, což je jedna z podmínek pro jejich řádnou funkci. Tato nezvyklá architektura proteinu a nikoliv nutně konkrétní primární struktura je kritická pro funkci tetherinu. [1,3]

GPI kotva má tendenci umístit svůj C-konec do mikro-oblastí bohatých na cholesterol. Tyto oblasti jsou preferovány i obalenými viry pučícími z membrány. [1]

Tetheriny se nacházejí jak na povrchu buňky v cytoplasmatické membráně, tak některých membránách nitrobuněčných kompartmentů – především v Golgiho komplexu a v endosomech. [1]

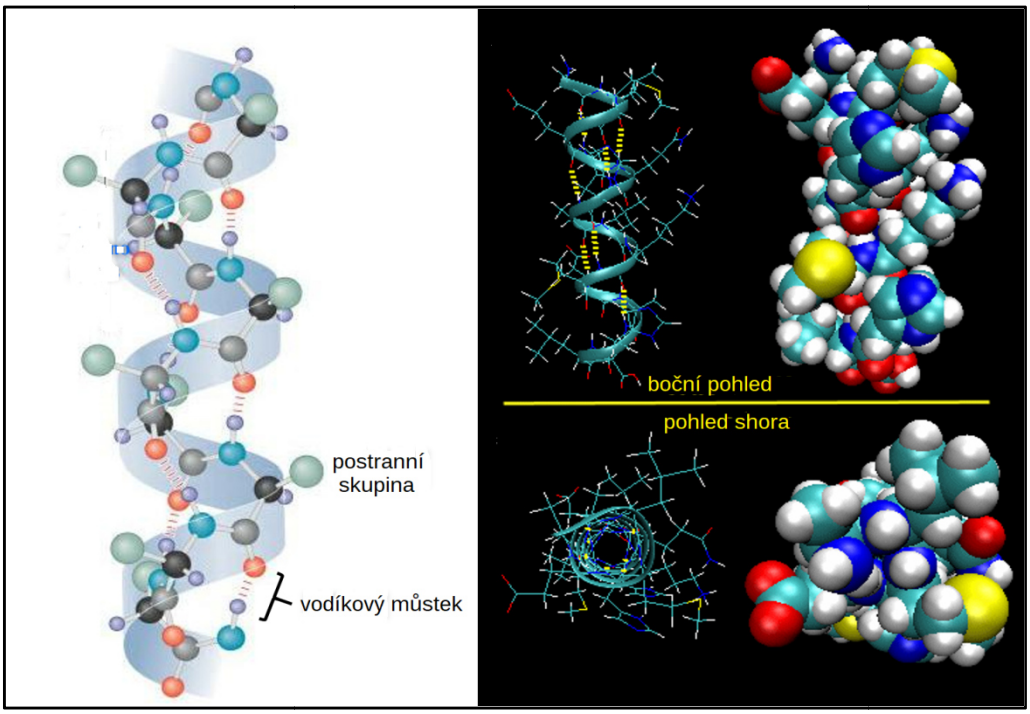
Byl prokázán účinek tehtherinů proti všem třídám retrovirů, filovirů (virus Eboly a Marburg), arenavirů, paramyxovirů, gama-herpes virů (herpesvirus asociovaný s Kaposiho sarkomem) a rhabdovirů. Tetheriny účinkují spojením membrány pučícího viru s buněčnou membránou nebo spojením dvou obalených virionů. Délka tethrinu se odhaduje na 17 nm, ale byly nalezeny vláknité struktury s tetherinem mnohem delší, nežli je předpokládaná délka tetherinu. [1]

2.1.2 Podrobnější popis sekundárních struktur tetherinu

Délka polypetidového řetězce bílkovin je nejčastěji v rozmezí 50 až 2000 aminokyselinových zbytků, nejčastěji přibližně 300 aminokyselinových zbytků. Velikost domén proteinů je obvykle mezi 40 a 350 aminokyselinovými zbytky. [4] Tetherin obsahuje 181 aminokyselinových zbytků. [1]

V případě tetherinu se jedná o tři domény: transmembránovou část, coiled-coil strukturu a GPI (glycosyl-phosphatidylinositol) modifikaci/kotvu.

V případě transmembránových domén se obvykle jedná o α-helix, tj. pravotočivou šroubovici. Tato sekundární struktura je poměrně rigidní a je držena pohromadě vodíkovými můstky mezi polárními skupinami jednotlivých peptidových vazeb. Tyto polární skupiny jsou odstíněny od hydrofobního vnitřku cytoplasmatické membrány nepolárními postranními řetězci těchto aminokyselin (aminokyseliny s hydrofobním postranním řetězcem jsou: glycin, alanin, valin, leucin, isoleucin, cystein, methionin, fenylalanin, tryptofan a prolin). V případě tetherinu se jedná jen o jeden průchod membránou (typicky o délce 23 aminokyselinových zbytků). Jeden řetězec polypeptidu obecně může procházet membránou vícekrát. [4] Na obrázku 2.1 níže je vyobrazena struktura α-šroubovice.



Obr. 2.1: Stavba α-šroubovice proteinu (dle: [5])

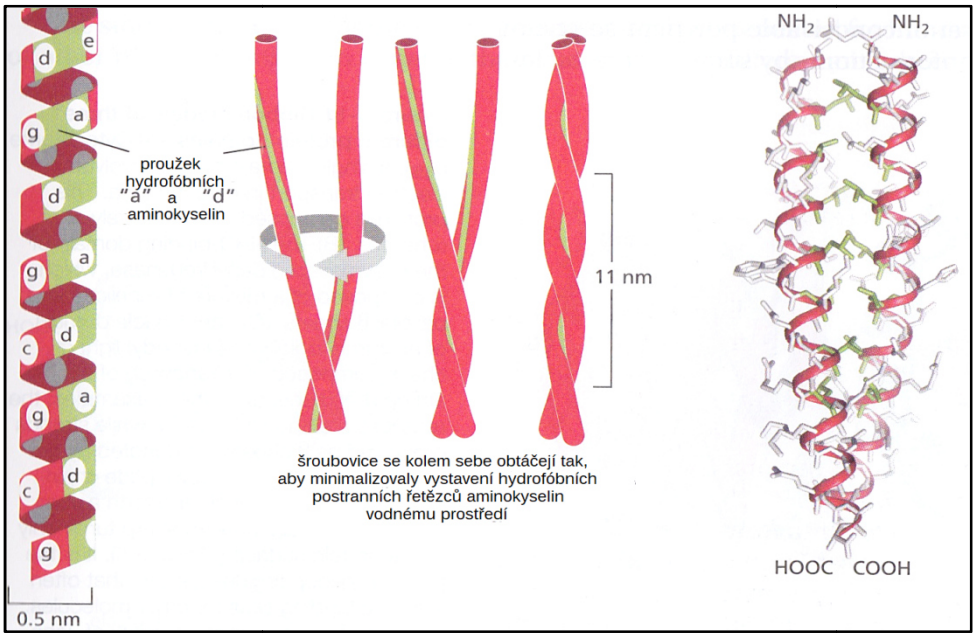
V dalším úseku tehterinu je vytvořena relativně pevná coiled-coil (CC) struktura. Lze rozlišovat kanonické a nekanonické coiled-coil domény. Kanonické CC domény jsou tvořené dvěma nebo více α-šroubovicemi v paralelním nebo antiparalelním uspořádání, které se kolem sebe stáčejí do pravidelné levotočivé nad-šroubovice. Typické je pro ně opakování sekvence sedmi aminokyselin („heptad“). Jejich stabilita vychází z pravidelného zapadání postranních řetězců residuí jedné šroubovice do „jamek“ v sousední šroubovici, čemuž se říká sbalení typu „knoflíky-do-jamek“ („knobs-into-holes“). [53]

Opakující se sekvence o větší délce než sedm aminokyselinových zbytků, residuí, jsou podkladem pro tvorbu nekanonických CC domén. Takovéto opakování lze popsat jako kombinaci např. úseků tří a čtyř residuí. Např. se opakuje uspořádání segmentů: 3+4+4, kdy vzniká opakování o délce 11 residuí („hendecad“). Hendecad je charakteristický pro mírně pravotočivé nad-šroubovice CC struktur. V mnoha přirozených CC strukturách mohou být přechody mezi různými délkami opakování. Odchylka od kanonického heptadu vede ke vzniku další formy sbalení nad-šroubovice nazývané „knoflíky-v-knoflících “ („knobs-into-knobs“). Přitom nezáleží na typu opakování ani na jeho kontextu. Strukturální omezení tvorby této nad-šroubovice jsou jiná než v případě heptadů, ale vedou též ke vzniku periodického opakování pozic s hydrofobními a hydrofilními aminokyselinovými zbytky. [53]

Pro CC domény nemohou být použity nedeformované α-šroubovice s typickými 3,63 residui na otočku. V CC strukturách je počet residuí na otočku nižší nebo vyšší jak 3,63. Při počtu residuí na otočku pod 3,63 vznikají levotočivé nad-šroubovice, při počtu residuí na otočku nad 3,63 vznikají pravotočivé nad-šroubovice. V levotočivých kanonických CC doménách je počet residuí na otočku snížen na 3,5 (7 residuí na 2 otočky). Nekanonické CC

domény mají např. 11 residuí na 3 otočky (3,67 residua na otočku), 15 residuí na 4 otočky (3,75 residua na otočku) nebo 19 residuí na 5 otoček (3,8 residua na otočku). [53]

CC struktura v tetherinu je tvořena dvěma α-šroubovicemi, kdy každá z nich má nepolární postranní řetězce aminokyselin v jednom souvislém pruhu jen na jedné straně svého obvodu. Každá α-šroubovice v CC struktuře tetherinu je tvořena opakujícím se motivem sedmi aminokyselin obvykle značených „abcdefg“, kdy aminokyseliny na pozici „a“ a „d“ jsou hydrofobní a umístěné v souvislém pruhu na stejném místě obvodu α-helixu. Tyto α-šroubovice se kolem sebe obtáčejí tak, aby nepolární postranní řetězce byly uvnitř coiled-coil struktury. Stavba CC struktury je na obrázku 2.2 níže. CC struktura je typická pro řadu fibrilárních proteinů, kromě tetherinů je to např. myosin ve svalovém vláknu nebo α-keratin v pokožce a jejích derivátech. [4]



Obr. 2.2: Uspořádání coiled-coil struktury (dle: [4])

Proteiny s GPI strukturou jsou běžné v celé živočišné říši. Jedná se o proteiny na jejichž C-konci, karboxylovém konci, je tzv. GPI kotva, kterou je protein ukotven do membrány. [5]

Přesněji protein ukotvený pomocí GPI je na svém karboxylovém konci připojen fosfodiesterovou vazbou fosfatidyletanolaminu k „jádru“, tvořeném trimanosyl-non-acetyl-glukosaminem (Man3-GlcN). Oligosacharidové Man3-GlcN jádro může postoupit řadu různých modifikací během sekrece z buňky. Redukující konec glukosaminu je připojen k fosfatidylinositolu (PI). PI je poté připojen další fosfodiesterovou vazbou k hydrofobní molekule membrány např. diacetylglycerolu. [5] – viz obr. 2.3.

Protein určený pro GPI modifikaci nese v blízkosti svého C-konce určité sekvence sloužící jako signál pro rozpoznání „místa střihu“, tzv. ω-místa k odebrání C-koncové části a připojení GPI kotvy. [70]

„Místo střihu“ je tzv. ω-místo, residua směrem k C-konci jsou číslována zvyšujícími se čísly (ω + 1, ω + 2 atd.). Residua směrem k N-konci proteinu jsou číslována snižujícími se čísly (ω – 1, ω – 2 atd.). [70]

Sekvenční motiv pro vytvoření GPI modifikace je obecně následující [70]:

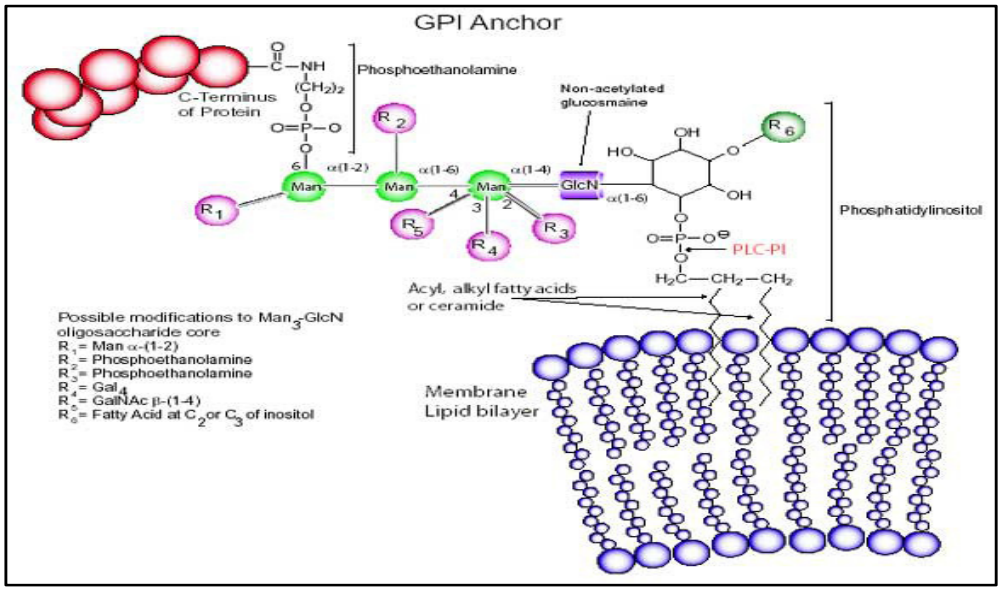
- nestrukturovaná spojovací oblast o asi 11 residuích (ω – 11 až ω – 1),

- oblast s malými residui (ω – 1 až ω + 2), což zahrnuje i ω místo pro sestřih proproteinu a pro následné připojení GPI kotvy,

- mezerníková oblast (ω + 3 až ω + 8) s mírně polárními residui a možným hydrofobním ostrůvkem na pozici ω + 4 a ω + 5,

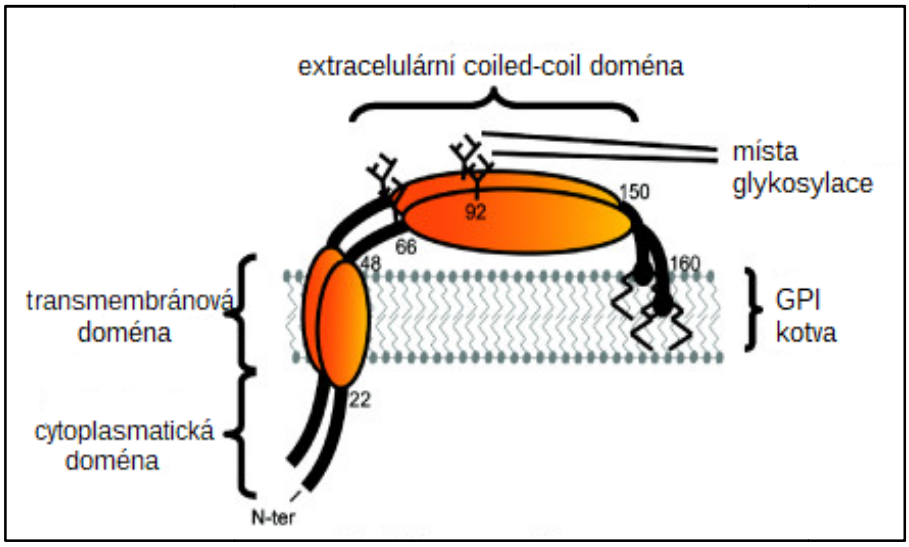
- hydrofobní řetězec od pozice ω + 9 nebo ω + 10 po C-konec. [70]

Funkce proteinů s GPI kotvou je různá: kromě tetherinu např. enzymatická, antigenní nebo adhezní funkce, nebo podíl na transdukci signálu z receptoru přes buněčnou membránu. [5]



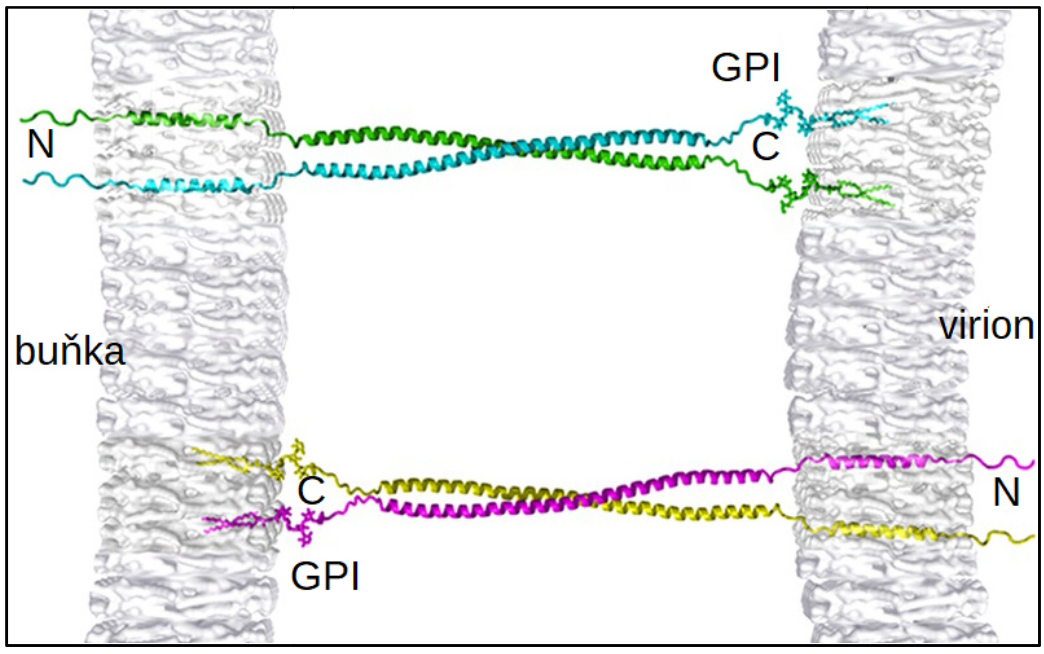
Obr. 2.3: Obecná struktura GPI kotvy proteinu (zdroj: [6])

Na obrázku 2.4 níže je zachyceno uspořádání dimeru tetherinu v membráně buňky, připraveného na svoji funkci, tj, zachycení virionu pučícího z membrány.



Obr. 2.4: Dimer tetherinu připravený v membráně buňky (dle: [7])

Dimer tetherinu může při vazbě virionu k buňce mít jak N-konec v membráně buňky a C-konec v membráně virionu, tak je možné i obrácená orientace tetherinu – viz obrázek 2.5 níže. Případně tetherin může znemožnit vstup virionu do další buňky spoutáním dvou obalených virových částic.



Obr. 2.5: Navázání virionu k buňce pomocí tetherinu (dle: [8])

2.2 Přehled nástrojů pro vyhledávání sekundárních struktur tetherinů v nukleotidových sekvencích

V této kapitole je pojednáno o používaných nástrojích pro vyhledávání otevřených čtecích rámců (ORFů), transmembránových domén, coiled-coil domén a GPI (glycosyl-phosphatidylinositol) modifikací.

2.2.1 Definice otevřeného čtecího rámce

Otevřený čtecí rámec (open reading frame, ORF, dále obvykle jen ORF) je základním pojmem v molekulární genetice a bioinformatice. Detekce ORFu je prvním nutným krokem k nalezení genů kódujících proteiny. Tato detekce je poměrně lehká *in silico,* ale poté je třeba další úsilí k prokázání, že se jedná o protein-kódující úsek DNA. [78]

Nejprve je třeba rozlišit pojem čtecí rámec a otevřený čtecí rámec (ORF). Čtecí rámec je jedna ze šesti možností jak přeložit danou dvou-vláknovou DNA do sekvence aminokyselin tj. proteinu. Pro daný čtecí rámec je ORF oblast nepřerušená STOP kodonem a je ohraničená dle některé z možných definicí. Tudíž ORF je oblast sekvence, která je otevřená translaci a je indikátorem potenciálního protein-kódujícího genu. [78]

Nejběžnější definice (označíme ji jako „definice č. 1“) ORFu zní takto: ORF je sekvence DNA, jejíž délka v párech bazí je dělitelná třemi bezezbytku a začíná translačním START kodonem (ATG) a končí jedním ze STOP kodonů (TAA, TAG, TGA). [78]

Obvyklost této definice je dána zřejmě historickými důvody, neboť první genomy, které byly čteny a přečteny, byly genomy prokaryot (a virů). Genomy prokaryot jsou podstatně méně komplexní nežli genomy eukaryot, především díky nepřítomnosti sestřihu. U eukaryot je tato definice platná jen u zralé mRNA, nebo vzácných genů jen s jedním exonem a dále u genů, kde všechny introny mají délku dělitelnou třemi a neobsahují STOP kodon v daném čtecím rámci, což lze považovat za poměrně vzácnou situaci. Této definici také způsobují problém kodony pro aminokyselinu methionin, neboť touto aminokyselinou jednak začíná syntéza každého proteinu (je tedy pro ni určen START kodon ATG a je to též jediný kodon kódující methionin) a zároveň se methionin může vyskytnout v podstatě kdekoliv uvnitř kódovaného proteinu. V tomto případě se pak např. musejí využít informace z kontextu v daném místě vlákna, jako je blízká přítomnost promotoru v 5‘ oblasti od START kodonu. V případě vícero tripletů ATG blízko sebe některé programové nástroje založené na definici č. 1 považují za START kodon jen první z ATG tripletů. [78]

U eukaryot je situace od dost složitější díky sestřihu tj. odstraňování intronů z hnRNA, prekurzoru mRNA. Introny často obsahují STOP kodony a také mohou při svém zachování v sekvenci způsobit posun čtecího rámce a bývá problém přesně určit sestřihová místa. [78]

Možnost sestřihu může být ošetřena jednoduše aplikováním „definice č. 2“, která se umí vypořádat i se STOP kodony v rámci intronů. ORF dle této definice neobsahuje celou kódující DNA sekvenci, ale třeba jen jeden potenciální exon nebo několik exonů. Definice

zde může být trochu „rozvolněna“ a za ORF považovat maximální rozsah dané sekvence nepřerušené STOP kodonem v daném čtecím rámci. Důležitým je zde i poznatek, že 5‘ nepřekládaná oblast obvykle obsahuje STOP kodon, takže takto vyznačená oblast nebývá o moc delší, nežli při oblast, která by začínala START kodonem. [78]

Můžeme ještě uvést „definici č. 3“, ve které jsou určovány nejprve exony nalezením sestřihových míst a následně jsou určeny krajní exony na 5‘ a 3‘ konci pomocí určení translační START a STOP pozice. Nicméně nalezení sestřihových míst je podstatně náročnější nežli jen nalezení START nebo STOP kodonu. Definice č. 3 bývá v literatuře zmíněna poměrně vzácně. Všechny tyto tři definice jsou užitečné a bývají implementovány v programech. [78]

V tomto kontextu je též vhodné porovnat význam pojmů ORF a exon. Tyto dva pojmy se liší již jen tím, že sestřihové místo a STOP kodon nejsou identické. Také v sousedním intronu se nemusí vyskytovat STOP kodon a tudíž jeden ORF může zahrnovat více jak jeden exon. Definice č. 2 poměrně jasně rozlišuje mezi ORFem, exonem a kódující DNA sekvencí. Tato definice může být také snadno implementována v počítači, je to též ta nejobecnější definice a může být použita i u prokaryot a metagenomových sekvencích. [78]

Definici č. 2 jako jedinou též používáme v této práci pro nalezení ORFů.

2.2.2 Seznam nástrojů pro vyhledávání otevřených čtecích rámců v nukleotidových sekvencích

Nástroje pro vyhledávání otevřených čtecích rámců (ORFů) v nukleotidových sekvencích jsou v současné době následující:

- ORF Finder,
- ORF Investigator,
- OrfPredictor,
- ORFik,
- getorf,
- OrfM,
- orfipy.

Případné ostatní nástroje autor považuje za nástroje se zcela minoritním použití.

2.2.3 Podrobnější popis nástrojů pro vyhledávání otevřených čtecích rámců v nukleotidových sekvencích

2.2.3.1 ORF Finder

ORF Finder (dostupný na stránkách National Center for Biotechnology Information, NCBI: https://www.ncbi.nlm.nih.gov/orffinder/) je nástroj s grafickým výstupem použitelný pro vyhledání všech otevřených čtecích rámců s uživatelem určenou minimální délkou ORFu v sekvenci nukleotidů zadané uživatelem nebo již přítomných v databázi. Tento nástroj identifikuje všechny ORFy v zadaném i v komplementárním vlákně vždy ve všech třech možných čtecích rámcích. Umožňuje vyhledávání jak dle standardního genetického kódu tak dle alternativních genetických kódů. [9, 10, 11]

Ve svém webovém rozhraní zobrazuje 6 horizontálních barevně označených úseček, kdy každá odpovídá jednomu nalezenému ORFu. (Pro zadané vlákno existují 3 čtecí rámce a stejně tak i pro komplementární vlákno.) Krom toho jsou v grafickém výstupu ve formě tabulky jednotlivé ORFy očíslovány, ke každému ORFu je určeno, zda se jedná o ORF v zadaném nebo komplementárním vláknu, je určen čtecí rámec, je určena pozice nukleotidu pro začátek a konec ORFu a je zobrazena též délka ORFu v počtu nukleotidů a v počtu aminokyselin. [9, 10]

Získaná aminokyselinová sekvence může být uložena v několika různých formátech (např. protein FASTA, CDS FASTA, ANS.1). Sekvence aminokyselin pak může být porovnávána se záznamy v databázích pomocí BLAST serveru. [9, 11]

2.2.3.2 ORF Investigator

ORF Investigator je nástroj, který poskytuje informaci o kódujících a nekódujících oblastech DNA, ale dokáže též provést párové globální porovnání různých genů a jiných úseků DNA. Je dostupný na: https://sites.google.com/site/dwivediplanet/ORF-Investigator. Tento nástroj efektivně hledá ORFy a vytváří odpovídající aminokyselinové sekvence ve formě jednopísmenkového kódu. Párové globální zarovnání mezi sekvencemi jej činí vhodným pro detekci různých mutací včetně jednonuklidových polymorfismů. Toto zarovnání je roztaženo po celé délce sekvence tak, aby zahrnulo co nejvíce odpovídajících sekvencí nukleotidů až po úplný konec sekvence. Pro porovnání sekvencí genů byl použit Needlemanův-Wunschův algoritmus a kód programu byl vytvořen v programovacím jazyku PERL, což je vhodné zejména pro uživatele operačního systému Windows, ale je tím zajištěna i obecně maximální interoperabilita mezi všemi běžně používanými operačními systémy. [11, 12]

Použitelným vstupem programu je sekvence DNA ve FASTA formátu. Program využívá jednoduchý algoritmus pro nalezení ORFů. Nejprve v zadané DNA sekvenci určí všech 6 čtecích rámců. Následně detekuje ORFy, jejich délku, pozici a jako konečný výsledek

aminokyselinovou sekvenci v jednopísmenkovém kódu. Pro párové globální porovnání program používá dynamický Needlemanův-Wunschův algoritmus, který porovnává každý pár znaků v obou DNA sekvencích a určí odpovídající si sekvence. Toto porovnání zahrnuje určení odpovídajících a neodpovídajících si znaků včetně ošetření mezer v porovnávaných sekvencích, takže určí maximální možnou shodu mezi oběma sekvencemi. U použitého algoritmu bylo matematicky prokázáno, že poskytuje nejlepší možné porovnání dvou sekvencí při dané míře jejich shody. [12]

2.2.3.3 OrfPredictor

Jedním z webových serverů pro identifikaci protein-kódujících oblastí je též OrfPredictor. Je dostupný na: bioinformatics.ysu.edu/tools/OrfPredictor.html a určen je pro EST-sekvence. Výstupem je předpovězená peptidová sekvence ve FASTA formátu a popisný řádek s ID dotazu, použitým čtecím rámcem a nukleotidovou pozicí kde daná kódující oblast začíná a končí. OrfPredictor usnadňuje anotaci EST-sekvencí, především pro projekty s rozsáhlými daty s EST-sekvencemi. [79]

EST tj. expressed sequence tag je krátká sekvence části cDNA. EST-sekvence byly navrženy jako strategie pro popis cDNA zhruba v polovině 90tých let 20. století. Postupné zlepšení sekvenačních metod snížilo cenu získávání sekvencí a zvýšilo atraktivitu výzkumu založeném na EST-sekvencích. Je to nyní jedna z nejvíce používaných metod pro vyhledávání genů a popis genomů. Anotace EST a cDNA často zahrnuje i identifikaci potenciálních protein-kódujících oblastí. [79]

Všechny eukaryotní mRNA obsahují souvislé sekvence nukleotidů kódujících protein. Zralá eukaryotní mRNA na 5‘ konci obsahuje tzv čepičku, 5‘ nepřekládanou oblast, poté protein-kódující oblast (ORF) a 3‘ nepřekládanou oblast následovanou poly-A koncem. Protein-kódující oblast na mRNA začíná START kodonem AUG, který též určí čtecí rámec, a končí jedním ze tří možných STOP kodonů v daném čtecím rámci (UAA, UAG, UGA). [79]

Většina cDNA knihoven je tvořena užitím oligo(dT) primerů pro přímou syntézu komplementárního vlákna DNA pomocí reverzní transkriptázy. V zásadě všechny takto vytvořené cDNA obsahují informaci o 3‘ konci a poly-A oblasti mRNA. EST-sekvence jsou získány jedním průchodem čtení sekvence cDNA a zahrnují i její 5‘ a 3‘ konec. Kvalitní systémy pro získání sekvence poskytují obvykle délku vlákna o zhruba 700-800 nukleotidech, přičemž délka nepřekládaných oblastí na 5‘ a 3‘ konci bývá obvykle mnohem menší nežli 500 nukleotidů. Tudíž se předpokládá, že EST-sekvence obvykle obsahuje protein-kódující oblast. [79]

Při analýze EST-sekvencí se překrývající úseky cDNA spojují do jedné cDNA pro odstranění redundance v datech. Většina EST sekvencí zahrnuje jen část dané mRNA, proto je větší výzvou předpovědět kódující oblast uvnitř EST sekvence nežli v plně sekvenované cDNA. Je též obtížné odlišit translační START kodon ATG od ostatních ATG kodonů. Najít START kodon je též obtížné z toho důvodu, že obecně neexistuje konsenzus ohledně toho,

jaké sekvence by měly obklopovat START kodon. U savců již existuje tento ustálený konsensus ohledně sekvence kolem strat kodonu: GCCRCCaugG, kde R označuje purinovou bazi (A nebo G). [79]

BLASTX dokáže porovnáním sekvence nukleotidů s databází proteinů hodnověrně identifikovat protein-kódující oblasti uvnitř DNA, pokud je zde dostatečná podobnost mezi zadanou DNA sekvencí, přesněji jejím translačním produktem a položkou v databázi proteinů. Chyby sekvenace mohou narušit správnou translaci ORFu, BLASTX může též detekovat posun čtecího rámce při případných delecích a inzercích. Pokud je dosaženo významné shody při porovnání pomocí BLASTX, pak je algoritmus OrfPredictoru používá jako návod k identifikování translačních čtecích rámců a kódujících oblastí. V případě EST sekvencí bez nalezené shody v databázi jsou jejich translační čtecí rámce a kódující sekvence predikovány v závislosti na přítomnosti a umístění intrinsických značek v sekvenci, které zahrnují START kodony, 5‘ a/nebo 3‘ STOP kodony a úseky pro poly-A konec. [79]

Algoritmus OrfPredictoru používá následující pravidla pro nalezení protein-kódujících oblastí a translačního čtecího rámce [79]:

V případech, kdy BLASTX identifikuje významnou podobnost zadané sekvence s databází, je použit výstup BLASTX a jsou aplikována pravidla 1 až 9 uvedená níže. Pokud zde naopak dojde ke konfliktu, pak mají přednost pravidla 1 a 2. Pro sekvence bez významné podobnosti jsou použita pravidla 3 až 10 [79]:

Pravidlo 1: Predikovaná kódující oblast musí obsahovat alespoň část translatované zadané DNA při použití BLASTX.

Pravidlo 2: Pokud dojde k posunu čtecího rámce, je přiřazeno první určení čtecího rámce po použití BLASTX.

Pravidlo 3: Pokud se v potenciálně protein-kódující oblasti, která je ohraničena translačním START a STOP kodonem, nenachází STOP kodon, pak je kódující oblast predikována od START kodonu po STOP kodon.

Pravidlo 4: U sekvence, která obsahuje poly-A signaturu, ale neobsahuje STOP kodon, se předpokládá, že neobsahuje žádnou kódující oblast.

Pravidlo 5: Pokud se v sekvenci nachází jeden nebo více START kodonů ATG a všechny jsou směrem ke 3‘ konci od jednoho nebo několika STOP kodonů, pak je jako START kodon použit první ATG kodon od 5‘ konce.

Pravidlo 6: Aby oblast mohla být označena za protein-kódující, pak také musí být dlouhá mezi START a STOP kodonem minimálně 90 nukleotidů (tj. kóduje peptid minimálně ze 30ti aminokyselin).

Pravidlo 7: Pokud sekvence zahrnuje poly-A signaturu a předchází jí STOP kodon, pak je za kódující oblast považována oblast od STOP kodonu k 5‘ konci.

Pravidlo 8: Pokud sekvence postrádá signaturu poly-A sekvence a kóduje ORF bez STOP kodonů, pak je celá oblast označena za protein-kódující. Ačkoliv pak je ve vzácných případech 5‘ nepřekládaná oblast označena jako kódující.

Pravidlo 9: V případě STOP kodonu bez přítomnosti signatury poly-A sekvence je obtížné určit, zda se jedná o 3‘ nebo 5‘ STOP kodon. Vzhledem k tomu, že cDNA klony bývají s větší pravděpodobností zkráceny na svém 5‘ konci, program předpokládá, že oblast do STOP kodonu k 5‘ konci je kódující oblast.

Pravidlo 10: Nejdelší ORF v šesti možných čtecích rámcích je vybrán jako kódující oblast. [79]

2.2.3.4 ORFik

ORFik je programový balíček (stále ještě ve vývoji ale s dostupnými stabilními verzemi) napsaný v programovacím jazyce R, určený pro analýzu transkriptů DNA a produktů jejich translace v běžných sekvenčních datech a datech next-generation sequencing technologií (NGS) jako je Ribo-Seq, RNA-Seq, TCP-Seq a CAGE. Je dostupný na: https://github.com/Roleren/ORFik. [80]

Tento programový balíček je napsán tak, aby mohl být použit pro analýzu jakékoliv transkripční oblasti. Programový balíček byl vytvořen během zkoumání Ribo-Seq dat (dat o translatovaných oblastech transkriptů) v otevřených čtecích rámcích a je to jeho primární účel užití. [80]

ORFik je extrémně rychlý díky naprogramování v C++ a užití programových balíčků data.table a GenomicRangers pro R. Programový balíček ORFik v současnosti podporuje např. [80]:

- velmi rychlé nalezení ORFů ve zkoumaném genomu nebo v setu DNA sekvencí a transkriptů,
- stovky funkcí pomáhajících s analýzou běžných sekvenačních dat, RNA-seq dat, CAGE dat, Ribo-seq dat, TCP-seq dat a RCP-seq dat,
- nalezení nových míst startu transkripce použitím CAGE dat,
- různá ověření identity genů, více jak 30 funkcí (např. FLOSS, ORFscore, entropy), které jsou podpořeny odbornou literaturou,
- funkce pro rozšíření balíčku GenomicRangers pro rychlejší shlukování, třídění, filtrování dat atd.,
- několik utilit pro tvorbu standardizovaných grafů,
- automatické stahování anotací jakýchkoliv již stanovených genomů,
- automatické stahování a přejmenování souborů z SRA (Sequence Read Archive),
- zjednodušení práce s masivním množstvím dat pomocí ORFik experiment-class tvořícím tabulku všech knihoven aktuálního experimentu. [80]

2.2.3.5 getorf

Tento program vyhledává a vypisuje sekvence ORFů pro jednu nebo více nukleotidových sekvencí. Za ORF považuje buďto sekvenci ohraničenou STOP kodony nebo

START kodonem na začátku a STOP kodonem na konci sekvence. ORF dokáže poskytnout jako sekvenci nukleotidů nebo přeloženou do sekvence aminokyselin. Případně program dokáže poskytnout oblast kolem START kodonu nebo kolem počátečního nebo koncového STOP kodonu daného ORFu. Kodony, včetně START a STOP kodonu, jsou dány translační tabulkou genetického kódu, kdy program umožňuje vybrání translační tabulky dle vyšetřovaného organismu. Výstupem je soubor se sekvencemi ORFů s minimální defaultní délkou ORFu 30 nukleotidů (tj. 10 aminokyselinových zbytků). [13]

Vstupem je standardní EMBOSS sekvence (také známá jako USA sekvence), kdy getorf dokáže číst i více nukleotidových sekvencí. Vstupní formát může být specifikován užitím příkazu v příkazové řádce: „-sformat xxx“, kde „xxx“ má uživatel nahradit požadovaným vstupním formátem. Podporované formáty jsou: gff, gff2, gff3, embl, em, gb, refseq, ddbj, refseqp, pir, swiss, sw, dasgff a debug. [13]

Výstupem je soubor se sekvencemi a s příponou *.orf obdobný FASTA formátu. Obdobně jako FASTA formát obsahuje hlavičky pro název výstupních sekvencí. Tento název nalezeného ORFu je vytvořen z názvu vstupní sekvence s připojeným znakem podtržítka a s jedinečným pořadovým číslem nalezeného ORFu. Popis nalezeného ORFu obsahuje počáteční a koncovou polohu ORFu. Zmíněné přiřazené číslo ORFu k původnímu názvu zadané sekvence slouží jen pro vytvoření unikátního jména zjištěné sekvence ORFu a neposkytuje další informace např. o pořadí, pozici nebo náležitosti k jednomu z možných vláken DNA. Pokud je nalezen ORF v komplementárním vláknu vůči zadanému vláknu, pak je číslo startovní pozice nižší nežli STOP kodonu. Např.: „>V00321_3 [339 – 27]“. Popis ORFu pak také obsahuje nápis „REVERSE SENSE“. [13]

Pokud sekvence byla specifikována jako kruhová sekvence, tj. typy prokaryotního plazmidu nebo prokaryotního chromozomu, (v příkazové řádce je třeba zadat přepínač: „-circular“) pak může ORF potenciálně pokračovat za „konec“ vstupní sekvence. Toto je ošetřeno tak, že se sekvence zopakuje za sebou třikrát. Jakémukoliv ORFu, který vyjde delší jak trojnásobek délky celé sekvence (tj. takový, který nezaznamená STOP kodon ani jednou během celé sekvence) je přiřazena délka o trojnásobku délky celé sekvence. V případě, že ORF je delší nežli délka kruhové sekvence, prochází tedy znovu tzv. „bodem zlomu“ a je mu ve výstupu též přiřazen popis: „ORF crosses the breakpoint“. [13]

2.2.3.6 OrfM

OrfM je dle autorů [14] rychlý prediktor otevřených čtecích rámců na metagenomových datech. Jedním z častých současných úkolů je nalezení a přeložení vhodných úseků DNA postrádajících STOP kodony. Obecně výpočetní nástroje pro nalezení ORFů jsou relativně pomalé a jsou pověstným „úzkým hrdlem“ vzhledem k rychlosti, jakou narůstá objem sekvenačních dat. Toto „úzké hrdlo“ se projevuje především u metagenomových dat. [14]

OfrM využívá Aho-Corasickové algoritmus [15] pro určení ORFů na DNA nepřerušených STOP kodony. Provedení benchmarku ukázalo, že OrfM je asi pětkrát rychlejší nežli podobné nástroje (konkrétně GetOrf a Translate), přičemž nalezne identické ORFy. Přestože OrfM je nezávislý na platformě použité pro sekvenování, je vhodný především pro velká kvalitní data, jaké např. produkují Illumina sekvenátory. [14]

Predikce ORFů může být v metagenomice provedena na dokončených populačních genomech, nahrubo sestavených populačních genomech, sestaveném kontingu nebo na jednotlivě stanovených sekvencích. [14]

Hledání genů v jednotlivých metagenomových sekvencích („gencentrická analýza“) je užitečné, když nejsou k dispozici referenční genomy a sestavení jednotlivých sekvencí je výpočetně příliš náročné nebo je případné mikrobiální společenstvo příliš komplexní na úspěšné sestavení sekvencí do celků. [14]

Konvenční nástroje pro predikci genů v dlouhých sekvencích využívají takové informace, jako je znalost statistického rozložení kodonů v oblasti genu. Pro predikci genů v krátkých sekvencích se ale tyto informace o kontextu stávají nehodnověrné. [14]

V bakteriálních a archeobakteriálních genomech je situace jednodušší. neboť tyto sekvence nejsou přerušeny introny a mezi-genové oblasti jsou minimální, takže i u krátkých sekvencí lze očekávat přítomnost genu. [14]

Predikce ORFů na datech z prvních next-generation sequencing platform (např. Roche 454) byly obtížné neboť tyto platformy poskytovaly sekvence zatížené chybami typu delecí a inzercí. V kontrastu s tím jsou dnešní sekvenátory od firmy Illumina, kde chyby typu delecí a inzercí jsou vzácné, sekvence jsou vysoké kvality a chyby, pokud nastanou, jsou spíše typu substitucí. [14]

Nalezení ORFů je podstatně lehčí nežli nalezení přímo genů nebo hledání na bázi tvorby překladu všech šesti čtecích rámců a hledání shody v databázích proteinů pomocí nástroje BLAST. Přesto současné prediktory ORFů nejsou schopny zpracovat data v rozsahu moderních metagenomů nad 500 Gbp. V tomto případě je účinným nástrojem OrfM. [14]

Vstupem nástroje OrfM je buď soubor o formátu FASTA nebo o formátu FASTQ (komprimovaný nebo nekomprimovaný gzip formát) a může přijmout i jiný formát, pokud je konvertován do FASTA formátu a je streamován přes přes pipe-line UNIX STDIN. OrfM manipuluje se vstupním souborem pomocí užitím kseq.h (dostupný na: http://lh3lh3.users.sourceforge.net/kseq.shtml). OrfM sám udává jako přednastavenou minimální délku ORFu 96 bp (tj. výsledný polypeptid má mít minimálně 32 aminokyselinových zbytků). Tento práh byl dán současnou převahou sekvencí dlouhých 100 bp od Illumina HiSeq sekvenátorů, kdy tato délka ORFu je minimální, při které lze určit všech 6 čtecích rámců. Všechny ORFy, které překračují prahovou délku, jsou nahlášeny i pokud se překrývají. OrfM dokáže použít standardní genetický kód ale i dalších 18 alternativních genetických kódů. [14]

Výstupem OrfM je FASTA soubor s aminokyselinovými sekvencemi, kdy hlavička sekvence je táž, jakou má vstupní sekvence, ale s přidáním řetězce v obecném tvaru „_X_Y_Z“ k prvnímu slovu původní hlavičky, kdy „X“ označuje startovní pozici, „Y“ je

číslo čtecího rámce a „Z“ je číslo ORFu. Toto schéma pojmenování zaručuje, že ORF je plně lokalizován ve vstupní sekvenci a že název ORFu je unikátní. OrfM může též na požádání poskytnout i odpovídající vstupní sekvenci nukleotidů. [14]

Algoritmus OrfM se liší od ostatních nástrojů. Ostatní nástroje určí všech 6 možných čtecích rámců a poté v těchto řetězcích hledají STOP kodony. OrfM určí STOP kodony v sekvencích přímo použitím vyhledávacího slovníku Aho-Corasickové [15]. [14]

Výkonost, rychlost OrfM byl srovnána s nástrojem GetOrf (verze 6.6.0) a nástrojem Translate (verze 1.9g+cvs20050121). Nástroje byly porovnány užitím tří veřejných souborů dat na jednom jádru z 20ti o 2,3 GHz na Intel Xenon E5-2650. Nástroj Translate byl použit při minimální délce ORFu 32 aminokyselin a nástroj GetOrf s 96 nukleotidů tak, aby byla omezena minimální délka výstupu na minimální přednastavenou, defaultní cut-off hodnotu nástroje OrfM. Ve všech případech byl OrfM rychlejší. Vyžadoval 21 % času nutného pro nástroj Translate a 14 % nutného času pro nástroj GetOrf. Soubor ORFů produkovaný těmito všemi nástroji byl stejný. Sekvence s ambiquitním kódem byly vynechány z tohoto porovnávání. [14]

2.2.3.7 orfipy

Orfipy je nástroj napsaný v jazyce python/cython k vyhledání ORFů v zadané DNA sekvenci rychlým a flexibilním způsobem. Jádro algoritmu pro vyhledávání ORFů je implementováno v jazyce cynthon pro zrychlení práce programu. Orfipy je kompatibilní s pythonem verze 3.6 a vyšší. K dispozici je např. na: https://pypi.org/project/orfipy/. Dalšími populárními nástroji jsou OrfM a getorf. Ve srovnání s OrfM a s getorf poskytuje orfipy více možností pro jemné naladění podmínek pro vyhledávání ORFů. Orfipy využívá vhodně vícejaderné procesory počítačů a je především rychlejší pro datové soubory s mnoha menšími FASTA sekvencemi jako jsou nově sestavované transkriptomy. Jako vstup je možné použít FASTA formát nebo jeho komprimovanou verzi ve FASTA nebo FASTQ formátu. Jemné ladění vstupních podmínek hledání zahrnuje např. specifikování START a STOP kodonů, hlášení částečných ORFů (postrádajících jen START nebo jen STOP kodon) a použití uživatelem zadaných translačních tabulek. Výsledky mohou být uloženy v různých formátech, např. v objemu dat efektivním BED formátu, nebo ve formátu FASTA. [11, 16]

Orfipy efektivně vyhledává ORFy pomocí Aho-Corasickové algoritmu s užitím knihovny pyahocorasick (dostupné na: http://pypi.org/project/pyahocorasick/). Hledání ORFů je též zrychleno její vektorizací. Orfipy se snaží vhodně využít všech jader procesoru počítače pro paralelní zrpacování FASTA sekvencí. Počet paralelních procesů je určen programem v závislosti na úloze a v závislosti na aktuálním záboru operační paměti a vytížení jader procesoru. [16]

Vstupem programu je FASTA soubor s případně i mnoha sekvencemi. S použitím programového balíčku pyfaidx tvoří orfipy index ze vstupního FASTA souboru pro lehčí a

efektivní přístup ke vstupním sekvencím. Dále může uživatel zvolit vstupní parametry jako jsou minimální a maximální délka ORFu, seznam START a STOP kodonů a/nebo uživatelem definované translační tabulky. [16]

Orfipy poskytuje řadu výstupních formátů jako jsou nukleotidové nebo peptidové sekvence ve FASTA souboru nebo souřadnice ORFů v BED formátu. Protože počet vstupních sekvencí může být velký při analýze transkriptů z meta-sestavených sekvencí nebo z velkých genomů, může vypsání ORFů ve formě FASTA souborů zabírat poměrně hodně místa na pevném disku. BED soubory šetří místo na pevném disku vypsáním jen souřadnic ORFů v daném vlákně. [16]

Při porovnání s jinými nástroji pro vyhledávání ORFů (např. getorf a OrfM), poskytuje orfipy výraznější flexibilitu a jemnější ladění vstupních podmínek a forem výstupu. Také ve většině scénářů vstupních podmínek je orfipy mnohem rychlejší nežli getorf a rychlejší nebo podobně rychlý jako OrfM. [16]

2.2.4 Seznam nástrojů pro vyhledávání transmembránových sekundárních proteinových struktur v aminokyselinových sekvencích

Nejužívanější nástroje pro vyhledávání transmembránových sekundárních proteinových struktur v aminokyselinových sekvencích jsou v současné době následující:

- TMHMM Server v. 2.0,
- HMMTOP 2.0,
- TM Finder,
- Phobius,
- MINNOU Server,
- CCTOP,
- tmhmm.py 1.3.1,
- pyTMHMM 1.3.2.

Případné ostatní nástroje lze považovat za nástroje s minoritním použitím.

2.2.5 Podrobnější popis nástrojů pro vyhledávání transmembránových sekundárních proteinových struktur v aminokyselinových sekvencích

2.2.5.1 TMHMM Server v. 2.0

Díky pokroku v klonování DNA a v sekvenačních technikách DNA je získáno mnoho proteinových sekvencí. Extrémní hydrofobicita většiny transmembránových proteinů výrazně ztěžuje detailní strukturní analýzu technikami, jako jsou nukleární magnetická rezonance a krystalografie za použití Roentgenova záření. Mezi téměř 10 000 položkami v PDB databázi (dostupné na: https://www.rcsb.org, rok 2001) je jen málo membránových proteinů se stanovenou strukturou na atomové úrovni. Díky tomuto se nutným nástrojem pro studium různých detailních interakcí v rámci membránových domén stala počítačová tvorba modelů a simulace. Stalo se tak nezbytným předpokladem pro tvorbu modelů vymezení oblastí s TM strukturou v proteinech. [17]

TMHMM Server v. 2.0 využívá pravděpodobnostní rámec skrytého Markovova modelu (HMM, hidden Markov model) pro predikci transmembránových šroubovic. Skryté Markovovi modely byly úspěšně použity např. při modelování statistické struktury genomů, proteinových rodin a struktur genů. [18]

TMHMM server verze 2 je dostupný na: http://www.cbs.dtu.dk/services/TMHMM/.

Dle [18] se jedná o novou metodu pro predikci lokalizace a orientace α-šroubovic v proteinech procházejících membránou. Úzké přiřazení mezi biologickými a početními stavy umožňovalo odvodit, které části modelu jsou důležité pro zachycení informace kódující membránovou topologii. Model byl odhadnut pomocí diskriminativních metod a metod maximálních pravděpodobností a byla přidána metoda pro přesnější přiřazování hranic α-šroubovic. V testech metodou křížové validace TMHMM správně předpověděl celou topologii 77 % sekvencí v datasetu s 83 proteiny se známou topologií a v datasetu se 160 proteiny se z větší části známou topologií. Výsledky TMHMM jsou srovnatelné s jinými metodami. [18]

Základní princip spočívá v definování sady stavů, kdy každý odpovídá specifické oblasti či místu v proteinu, který má být modelován. V nejjednodušším případě se model pro transmembránový úsek proteinu skládal ze tří stavů: jeden pro vnitřní smyčku, druhý pro transmembránový úsek a třetí pro zevní smyčku. Každý stav má přiřazenou distribuci pravděpodobnosti pro každou z 20ti kódovaných aminokyselin. Stavy jsou na sebe napojeny v biologicky hodnověrném způsobu. Např. stav pro cytosolovou smyčku je napojen jednak sám na sebe, neboť smyčka může být delší než 1 aminokyselinový zbytek, a je napojen také na stav transmembránové šroubovice, protože na sebe mohou navazovat. Pravděpodobnosti jednotlivých aminokyselin a pravděpodobnosti průchodu membránou jsou učeny standardními inferenčními technikami, které počítají maximální dané posteriorní pravděpodobnosti a skutečně pozorované frekvence výskytu. Definováním stavů pro aminokyselinové zbytky v transmembránových šroubovicích a jiných stavů pro aminokyselinové zbytky ve smyčkách a jiných stavů pro aminokyseliny ve větších doménách

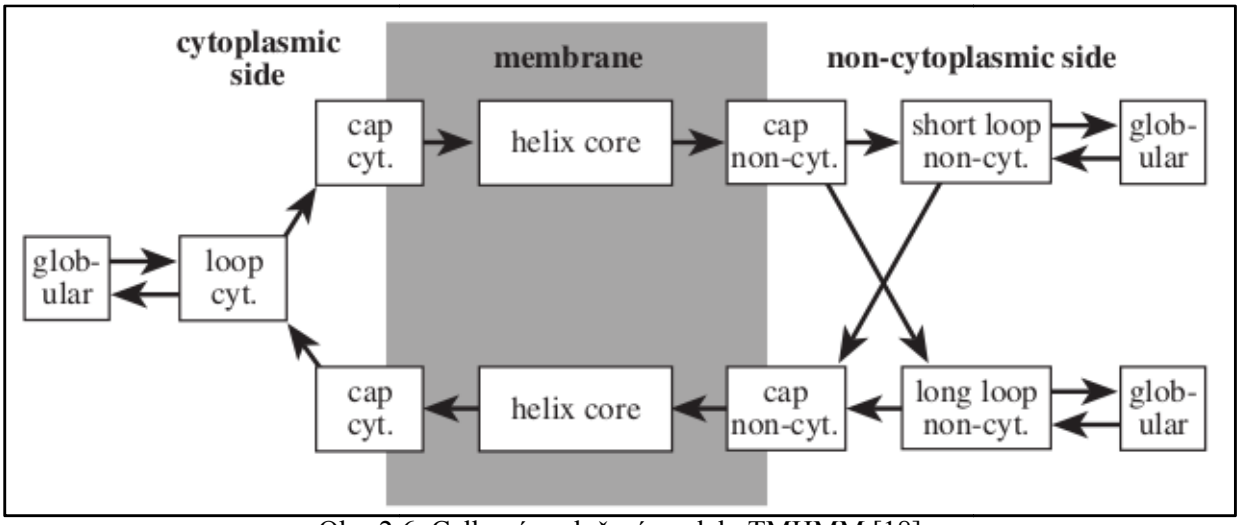
mimo membránu a jejich spojením do cyklu můžeme vytvořit model, který se svou architekturou velmi podobá biologickému systému, který modelujeme. Pokud jsou parametry modelu vyladěny na zachycení biologické reality, pak by průchod sekvence proteinu skrze stavy s nejvyšší pravděpodobností měl být schopen odhalit správně topologii proteinu. [18]

Metoda HMM nepracuje s žádnými fixně danými prahovými hodnotami a pravidly. Optimální cesta přes HMM je nalezena jedním krokem. Proto je použití HMM metody výhodné v situacích, kdy je třeba kombinovat několik „signálů“ pro nalezení správné topologie. [18]

Např. segment, který by normálně nebyl označen za transmembránový z důvodu slabé hydrofobnosti, může být stále pomocí HMM predikován jako TM úsek, pokud to okolní topogenní signály podporují. To je např. v situaci u proteinu vícekrát procházejícím membránou, kde transmembránové úseky spolu interagují pomocí hydrofilních zbytků aminokyselin, jako např. u iontových kanálů. [18]

Základní architektura TMHMM je ukázána na obrázcích 2.6, 2.7 a 2.8 níže. Jsou možné tři hlavní možnosti umístění aminokyseliny: v transmembránové šroubovici, v jejím zakončení na úrovni fosfátových skupin fosfolipidové dvojvrstvy membrány („vršek“ šroubovice, angl. „cap“) a ve smyčkách mimo membránu. Kvůli rozdílné distribuci aminokyselin na různých místech však můžeme použít a rozlišit sedm různých stavů: jeden pro jádro šroubovice, dva pro „vršky“ šroubovice na každé straně membrány, jeden pro smyčky na cytosolové straně membrány, po jednom pro dlouhé a krátké smyčky na extracelulární straně membrány a jedna pro globulární „domény“ uprostřed některé ze smyček. Pravděpodobnosti aminokyselin všech stavů stejného typu jsou navzájem svázány, tj. jsou odhadovány kolektivně. [18]

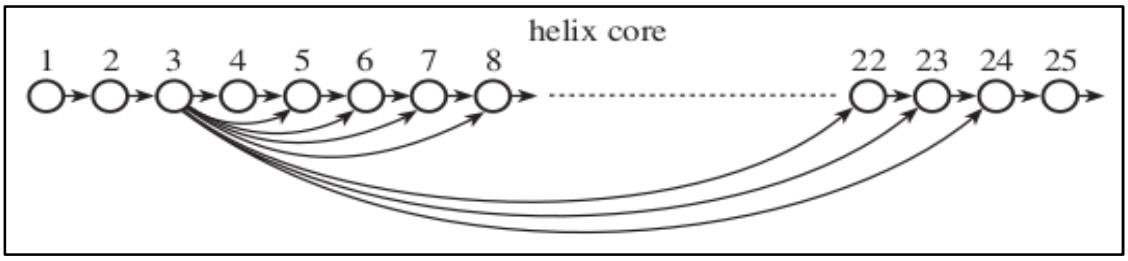
Na obrázku 2.6 je ukázáno celkové rozložení modelu TMHMM. Každý rámeček odpovídá jednomu nebo více možných stavů. Části modelu se stejným textem jsou svázány, tj. jejich parametry jsou stejné. [18]



Obr. 2.6: Celkové rozložení modelu TMHMM [18]

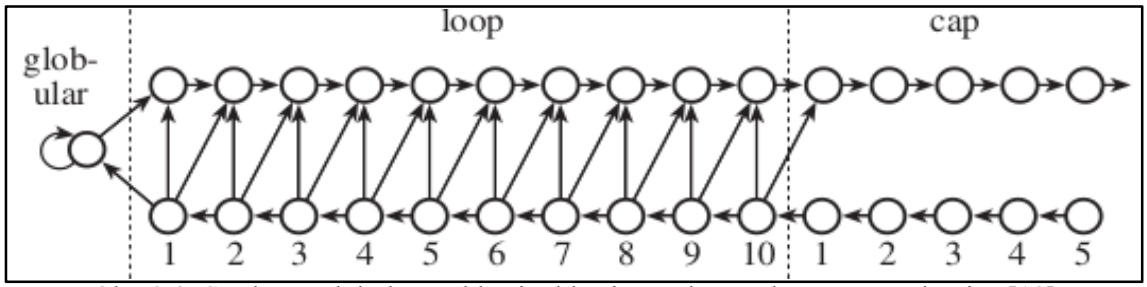
Na obrázku 2.7 je diagram pro část s názvem „helix core“ v modelu zachyceném na obrázku 2.6. Z části „vršku“ šroubovice („cap cyt..“ nebo „cap non-cyt.“, dle obr. 2.6) je na obr. 2.7 možný vstup jen do uzlu číslo 1. První tři a poslední dva uzly je nutno projít, ostatní

lzeobejít. Tento model zahrnuje délku segmentu od 5ti do 25ti aminokyselinových zbytků. [18]



Obr. 2.7: Diagram pro část s názvem „helix core“ v modelu zachyceném na obrázku 2.6 [18]

Na obrázku 2.8 je struktura globulární oblasti, oblasti smyčky a „vršku“, zakončení šroubovice („cap“). V každé této oblasti jsou pravděpodobnosti jednotlivých aminokyselin spolu svázány. Všechny tři rozdílné oblasti smyček jsou tímto způsobem modelovány, ale mají rozdílné parametry v některých oblastech. [18]



Obr. 2.8: Struktura globulární oblasti, oblasti smyčky a zakončení šroubovice [18]

Transmembránová šroubovice je modelována pomocí dvou oblastí „vršků“, zakončení, každý o délce 5 aminokyselinových zbytků a obklopující jádro šroubovice o délce mezi 5ti až 25ti aminokyselinovými zbytky, což umožňuje zohlednit šroubovici o délce od 15ti do 35ti aminokyselinových zbytků. Tyto oblasti zakončení šroubovice mají svoji vlastní distribuci aminokyselin a to i zvlášť na intra- a extracelulární straně membrány. Tato zakončení jsou však označovány společně jako zakončení membrány jak pro trénink modelu, tak pro samotné predikce. Přestože model obsahuje dva sety transmembránových stavů, dle toho v jakém směru prostupuje šroubovice membránou, jsou jejich parametry zrcadleny a spolu svázány. [18]

Smyčky mezi šroubovicemi jsou modelovány pomocí modulů, které obsahují 2×10 stavů v konfiguraci ve formě „žebříku“ a jedné ve stavu, kdy vytváří sama se sebou smyčku. Myšlenka je taková, že prvních 10 stavů by mělo obsahovat většinu topogenního signálu, zatímco větší, globulární domény jsou modelovány jednoduchým způsobem tak, že vytvářejí smyčku samy se sebou a s neutrální distribucí aminokyselin. [18]

Dlouhé smyčky na extracelulární straně, které obsahují globulární domény, se jeví, že mají jiné vlastnosti nežli krátké smyčky. Nevykazují konzistentně přítomnost kladně nabitých aminokyselinových zbytků. Proto jsou extracelulární smyčky modelovány dvěma různými cestami. HMM tudíž obsahuje na vnější straně membrány dva paralelní moduly. Během

tréningu jsou extracelulární smyčky s délkou více jak 100 aminokyselinových zbytků speciálně označeny a přiřazeny modulu pro delší smyčku. [18]

Celkový počet volných parametrů v celém modelu je 216, což může být porovnáno s neuronovými sítěmi, které jich obsahují desítky tisíc. [18]

2.2.5.2 HMMTOP 2.0

HMMTOP 2.0 server (Hidden Markov Model for Topology Prediction) je predikční server pro určování transmembránové topologie proteinu. Předpovídá jak lokalizaci šroubovicových transmembránových (TM) segmentů tak topologii transmembránových proteinů. Je dostupný na: http://www.enzim.hu/hmmtop. [19]

Metoda je založena na principu, že topologie transmembránových proteinů je určena maximální divergencí aminokyselinového složení sekvenčních segmentů. Uživateli je umožněno přidat doplňující informace o lokalizaci TM segmentu, čímž je podpořena síla predikce – může i u více segmentů sám určit, do jaké z pěti možných oblastí (použitých u HMMTOP 2.0) spadají: intracelulární, extracelulární, intracelulární část šroubovice, extracelulární část šroubovice a samotná transmembránová šroubovice. Informace o segmentech je včleněna do Baum–Welchova algoritmu zadáním podmíněných pravděpodobností. [19]

HMMTOP 2.0 server má poměrně vysokou přesnost předpovědí TM úseků. Např. predikce topologie human multidrug transporter-associated proteinu (MRP1) často selhávala u jiných predikčních metod. Protein MRP1 náleží k rodině proteinů ABC (ATP Binding Cassette), která má společný rys – poměrně velké množství TM úseků. HMMTOP 2.0 server zde předpovídá správný počet (17) TM sekvencí. A to jen za použití zadané sekvence, defaultních parametrů a bez zadání doplňujících informací. Nicméně dva z těchto 17 úseků při této predikci náleží do velké cytoplasmatické smyčky a dva TM úseky jsou vynechány v C-terminální oblasti. Dodáním několika podpůrných informací vede ke zcela přesně stanovené topologii a přesnému umístění všech TM úseků. [19]

2.2.5.3 TM Finder

V případě absence strukturních dat s vysokým rozlišením mohou být segmenty prostupující membránou v TM doméně navrhnuty jen z aminokyselinové sekvence. Příslušnost k transmembránovému segmentu se určí stanovením hydrofobicity úseku pomocí posuvného okna. Vývoj škál hydrofobnosti byl tradičně založen na kombinaci statistické analýzy a vlastností samotné aminokyseliny jak ve formě monomeru, tak ve formě aminokyselinového zbytku v globulárním proteinu a v krátkém hydrofilním peptidu. Experimentální determinace hydrofobicity může být provedena užitím sérií modelů peptidů napodobujících TM strukturu s prototypem sekvence ve formě:

KKAAAXAAAAAXAAWAAXAAAKKKK-amid (kde podtržená sekvence reprezentuje hydrofobní jádro o dostatečné délce pro prostoupení lipidové dvojvrstvy membrán a kde X odpovídá jakékoliv právě zkoumané geneticky kódované aminokyselině). Tato experimentální determinace vedla ke stanovení míry hydrofobnosti pro 20 kódovaných aminokyselin díky použití měření retenčních časů těchto modelových peptidů pomocí HPLC (High Pressure Liquid Chromatography). [17]

Když byly odpovídající si sekundární struktury charakterizovány ve vodném a v micelárním prostředí, vytvářely peptidy náhodné nebo částečně šroubovicové struktury ve vodném prostředí, ale při určité aminokyselinové sekvenci vytvořily plně helikální strukturu po integraci do micelární membrány. Tyto experimenty odhalily, že existuje práh hydrofobnosti odpovídající přibližně poly-alaninové sekvenci. Byly zkoumány hydrofobicity sekvencí přirozených proteinů v databázích SWISS-PROT a TMbase pomocí škály odvozené od HPLC metody a byl použit uvedený práh hydrofobicity. Zjistilo se, že skoro 97 % TM segmentů proteinů má hydrofobicitu nad uvedeným prahem, ale skoro 80 % šroubovic (o délce nad 19 aminokyselinových zbytků), které nejsou v membránách, nesplňovalo tento limit. Tudíž tento práh může být použit jako jedno z důležitých vodítek pro odlišení TM segmentů od šroubovic kompletně ve vodě rozpustných proteinů. [17]

Transmembránové segmenty jsou většinou α-helixy. Samotná tvorba šroubovicové struktury proteinu je ovlivněna především molekulárním okolím a také umístěním některých aminokyselinových zbytků. Tyto aminokyseliny mohou pomáhat tvořit nebo naopak rozvolňovat šroubovici také v závislosti na tom, zda se jedná o vodné nebo hydrofobní prostředí. Tato okolnost je zvýrazněna např. faktem, že aminokyseliny s rozvětveným postranním řetězcem (valin, leucin, izoleucin) a aminokyselina glycin dle statistické analýzy u proteinů rozpustných ve vodě znemožňují tvorbu α-helixů, ale naopak z asi 40 % tvoří transmembránové segmenty. Tudíž škály odvozené pro stanovení tvorby šroubovicové struktury u rozpustných proteinů nemohou být použity pro membránové proteiny. Tato okolnost může být ošetřena měřením cirkulárního dichroismu. Byly vyšetřeny série modelových peptidů napodobujících TM segmenty nacházející se v nepolárním rozpouštědle (n-butanolu) a získáno tak měřítko schopnosti jednotlivých aminokyselin přispívat k tvorbě šroubovicové struktury v nepolárním prostředí. Když bylo toto měřítko aplikováno na proteinové databáze, projevilo se plně rozlišení mezi šroubovicemi TM segmentů a šroubovicemi neprocházejícími membránou. Ukázalo se tak, že kromě podmínky splnění určitého prahu hydrofobnosti existuje minimální požadavek na složení sekvence pro tvorbu šroubovice vhodné pro prostředí vnitřku membrány. [17]

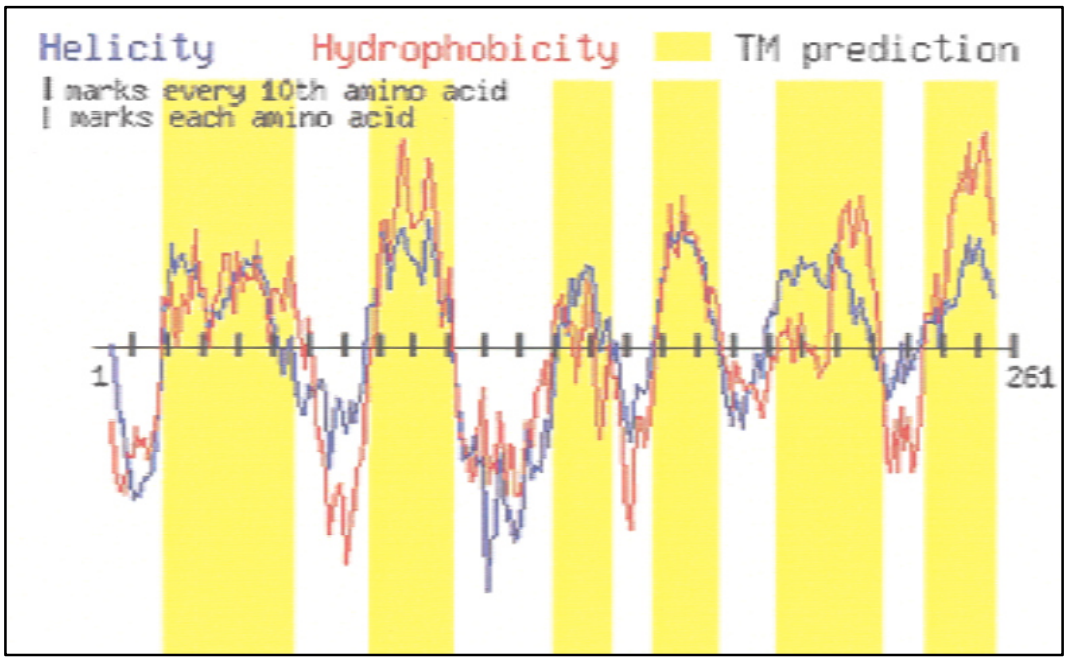
TM Finder má v sobě implementován tento nutný práh hydrofobnosti a také podmínky pro tvorbu šroubovice v nepolárním prostředí tak, že dokáže lokalizovat TM segment proteinu jen z primární sekvence u naprosté většiny proteinů. [17]

Při konstrukci programu TM Finder byla vytvořena škála hydrofobicity pro jednotlivé aminokyseliny. Tato hodnota byla odvozena z retenčních časů v HPLC pro každý peptid typu KKAAAXAAAAAXAAWAAXAAA-KKKK-amid, kde X označuje umístění zkoumané aminokyseliny. Byla dopočtena bezrozměrná relativní škála od 5 po -5. Kdy 5 je pro nejvíce

hydrofobní aminokyselinu (fenylalanin) a -5 je pro nejvíce polární aminokyselinu (lysin). Samostatně byla počítána jiným postupem míra hydrofobnosti aminokyseliny cystein, který byl do zkoumaného peptidu vložen jen jedenkrát a to doprostřed a na krajní X pozice byl umístěn leucin. Bylo třeba se vyhnout situaci, kdy by cysteinové zbytky vzájemně tvořily disulfidové můstky. [17]

Dále byla vytvořena škála schopnosti jednotlivých aminokyselin tvořit šroubovici v nepolárním prostředí. Tato škála byla odvozena od měření cirkulárního dichroismu v nepolárním rozpouštědle (n-butanolu), kdy se jedná o bezrozměrnou veličinu. V případě hodnoty rovné jedné se jedná o schopnost dané části sekvence proteinu tvořené danou aminokyselinou nabýt z přibližně 50 % konformaci α-šroubovice v nepolárním prostředí. Tento rozsah byl tabelován pro všech 20 kódovaných aminokyselin a kolísá od hodnoty 1,29 pro izoleucin po hodnotu 0,57 pro prolin. Vyšší hodnota znamená výraznější schopnost tvořit α-šroubovici v nepolárním prostředí. [17]

Prototyp programu byl nejprve vyvinut v Microsoft Visual Basicu a MS Excelu. Webová verze byla napsána v C a Perlu. Obrázek 2.9 ukazuje grafický výstup programu. TM Finder akceptuje vstup ve formátu SWISS-PROT a výstupní TM segmenty jsou dány kombinací predikce hydrofobicity a míry schopnosti tvorby šroubovice v nepolárním prostředí, které byly přiřazeny každé aminokyselině. Je počítán klouzavý průměr za použití posuvného okna. TM segmenty jsou vyznačeny na základě aminokyselinových zbytků, které přesahují jak práh hydrofobnosti, tak zmíněnou schopnost tvorby šroubovice. Byla také přidána možnost spojování TM segmentů oddělených krátkými mezerami, což snižuje míru nejistoty. Nejistota by mohla být dána lokálně se vyskytujícími hydrofilními zbytky aminokyselin a/nebo výběrem velikosti okna tj. rozsahu zkoumané sekvence. [17]



Obr. 2.9: Grafický výstup programu TM Finder [17]

Uživatelé mohou nastavit pět parametrů v závislosti na velikosti a typu proteinu pro získání co nejlepších predikcí. Nastavitelné parametry jsou [17]:

- velikost posuvného okna na N-konci proteinu,
- velikost posuvného okna na C-konci proteinu,
- minimální délka jádra šroubovice,
- maximální délka překlenované mezery,
- minimální délka segmentu.

Defaultní nastavení těchto parametrů vychází z jejich optimalizace pro trénovací set proteinů se stanovenými transmembránovými strukturami. Tento set je zveřejněn na webové stránce TM Finderu Program je dostupný na: http://tmfinder.research.sickkids.ca/cgi-bin/TMFinderForm.cgi. [17]

Program je určen pro transmembránové α-šroubovice. Existují i membránové proteiny, tvořené tzv. β-soudkem (např. poriny), které musejí být hledány jinými algoritmy. [17]

2.2.5.4 Phobius

Phobius kombinuje transmembránovou topologii a predikci signálního peptidu (určení signálního peptidu plní také např. nástroj SignalP). Pro tento nástroj je vytvořeno webové rozhraní (http://phobius.binf.ku.dk a http://phobius.cgb.ki.se). Jsou také dostupné stand-alone verze pro akademické uživatele Linuxu a SunOS na vyžádání. Server Phobius je implementován jako Perl CGI skript. Grafy jsou tvořeny pomocí nástroje gnuplot. Normální predikce jsou tvořeny dle nepublikovaného ANHMM balíku od stejných autorů. Omezené predikce jsou tvořeny pomocí HomologHMM balíku popsaném v [20]. [21]

Často nástroje pro predikci TM topologie signální peptid zaměňují za TM doménu a také naopak nástroje pro predikci signálních peptidů často stanovují TM domény jako signální peptidy. Tento fakt bývá často přehlédnut při testování takových nástrojů a bývá hlavní příčinou rozdílných výstupů těchto nástrojů. Radou zde obvykle bývá vyřadit predikované signální peptidy před stanovováním TM domén a nebo vyřadit proteiny se známou TM doménou před stanovováním signálních peptidů. Chybovost mezi těmito zkříženými predikcemi je zhruba stejná a tak i posledně uvedením ošetřením se získá stejně, jako se ztratí. [21]

Phobius obsahuje skrytý Markovův model, který má v sobě zahrnuty dva submodely – samostatně pro signální peptid a pro TM domény. Prediktor je pak nucen si vybrat mezi těmito dvěma variantami. Následující benchmark ukázal snížení falešně stanovených TM domén oproti nástroji TMHMM o 4 % a falešná predikce TM šroubovic oproti nástroji SignalP byla snížena o 8 %. Phobius zvedl přesnost stanovení čistě TM domén oproti nástroji

TMHMM z 44,5 % na 53,9 %. Phobius řeší problém způsobem, který standardní nástroje nemohou zajistit. [21]

Příslušné webové rozhraní přijímá vstupní formát FASTA se sekvencí aminokyselinových zbytků v peptidech a proteinech, jež má prozkoumat. Preferovaný je up-load FASTA souboru se sekvencí. Predikce mohou být poskytnuty v krátkém výstupu ve formě jedné řádky a nebo v dlouhém výstupu ve formě UniProt tabulky. Všechny predikce mohou být také doplněny grafem lokalizace posteriorní pravděpodobností pro daný aminokyselinový zbytek v závislosti na celé sekvenci zadaného proteinu. Program umožňuje „normální" predikci nebo „predikci s omezením". [21]

Predikce s omezením umožňuje zpřesnění predikce zahrnutím informací o lokalizaci vyšetřované sekvence v rámci proteinu nebo můžeme mít data typicky z fúzí reportérových genů, experimentů s protilátkami a nebo mít informaci o daném místě z důvodu znalosti nutné funkce dané sekvence. Server Phobius poskytuje uživateli možnost specifikovat takováto omezení predikce. Uživatel může mj. určit, zda daný aminokyselinový zbytek je ve smyčce v cytosolu nebo zevním prostředí buňky, nebo zda ví, že se jedná o TM segment. Také může zadat, že N-koncová část proteinu je signálním peptidem. [21]

2.2.5.5 MINNOU Server

MINNOU Server (Membrane protein IdeNtification withOUt explicit use of hydropathy profiles and aligments) používá strategii, která je založena na kompaktní reprezentaci dané aminokyseliny a jejího okolí. Může být použit pro predikce jak transmembránových (TM) α-šroubovic tak tzv. β-soudků, které bývají také umístěny do membrány. Server používá „strukturální profily" založené na predikci pro každý aminokyselinový zbytek. Pro každý aminokyselinový zbytek je předpovězena „relativní přístupnost rozpouštědla" (relative solvent accessibility, RSA profil) neboli „relativní přístupná plocha" (relative accessible surface area). A také je přiřazena pravděpodobnost přítomnosti aminokyselinového zbytku v různých sekundárních strukturách (SS profil). Na tento iniciální predikční krok se může pohlížet jako na efektivní projekci informace kódované pomocí mnohonásobných zarovnání (multiple aligment, MA) proteinových sekvencí redukovaných do předpovězených RSA/SS profilů. [22]

Každý aminokyselinový zbytek je zde reprezentovaný pěti čísly: předpovězená hodnota RSA, hodnověrnost predikce RSA hodnoty, a tři předpovězené pravděpodobnosti pro každou ze tří uvažovaných sekundárních struktur (šroubovice, β-list resp. „cívka", coil, nebo obecně jiná, tj. obvykle globulární, struktura). [22]

Tato kompaktní reprezentace vychází z pozorování, že po natrénování této metody na proteinech rozpustných ve vodě, kde byla určena pro zjištění struktur ukrytých v hydrofobním vnitřku proteinu, lze tuto metodu použít i pro predikci transmembránových úseků. Byla také zjištěna poměrně vysoká přesnost předpovědí těchto úseků. [22]

Pro vytvoření sady trénovacích dat bylo třeba získat neredundantní a maximálně reprezentativní seznam proteinů se známou strukturou. Byla využita MPtopo databáze membránových proteinů ve verzi z června 2004. Byly získány řetězce proteinů s přesně stanovenou 3D strukturou šroubovic, šroubovic s ne zcela přesně stanovenou 3D strukturou a dalších proteinů s 3D stanovenou strukturou β-soudku nebo jiných membránových proteinů. Byly vyloučeny šroubovice s příliš nízkým prostorovým rozlišením struktury. V této souvislosti se znatelně zredukovalo množství příkladů použitelných pro trénink neuronových sítí (NS). Nicméně použitá metoda s kompaktní reprezentací aminokyselinového zbytku a jeho okolí je méně citlivá na tento nedostatek nežli metody založené na kompletním porovnávání proteinových sekvencí. Pro vyřazení redundantních sekvencí byl použit BLASTP program [23]. Trénovací sada byla též rozšířena přidáním signálních peptidů (z databáze signálních peptidů PrediSi [24]). Byl také použit kontrolní set globulárních proteinů s 13 proteiny, ve kterých byly falešně pozitivně určeny TM sekvence. Tyto úseky spolu s přidáním 15ti aminokyselinových zbytků na každém konci byly přidány do trénovacího setu. Tudíž trénovací set pro určení TM šroubovic byl složen z TM úseků, fragmentů globulárních proteinů a ze signálních peptidů. [22]

Při vývoji serveru byly porovnány výsledky lineárních a nelineárních klasifikátorů, tedy přístup pomocí lineární diskriminační analýzy (LDA) a pomocí neuronových sítí (NS). Klasifikátory na bázi LDA byly použity jako reference pro posouzení nelineárních modelů. Neuronové sítě byly použity pro výsledný predikční systém. [22]

Architektura všech vyzkoušených NS byla podobná. Byly použity tři vrstvy neuronů, vstupní, vnitřní – skrytá a výstupní. Sousední vrstvy byly plně propojeny. Počet příznaků pro reprezentování každého aminokyselinového zbytku se měnil od jednoho po šest pro testy s kompaktním vyjádřením vlastností aminokyselinového zbytku a jeho okolí a 20 pro metody s mnohonásobným zarovnáním. Např. při použití 5 příznaků pro jednu aminokyselinu se vstup skládal až ze 155 čísel, což reprezentovalo posuvné okno pro 31 aminokyselinových zbytků (což byla maximální používaná délka okna). Ve výstupní vrstvě byly 2 uzly. Jeden aktivován v situaci, kdy se jedná o sekvenci uvnitř membrány a druhý opačně, kdy se jedná o sekvenci mimo membránu. Všechny NS byly trénovány použitím Rprop algoritmu [25, 26]

Poté byl přidán dvoustupňový predikční systém. Druhá vrstva NS provedla zprůměrování a vyhlazení iniciální klasifikace získané první vrstvou. Druhá vrstva zvyšuje přesnost předpovědí TM šroubovic ve smyslu lepší senzitivity i specificity. Přesto stále byly predikovány příliš krátké nebo dlouhé šroubovice. Proto byla odhadnuta pravděpodobnost výskytu šroubovic v závislosti na jejich délce a použita pro filtr pro druhou vrstvu NS pro vyvarování se takto nepřirozených predikcí. Ve výsledku je finální filtr použit pro rozdělení příliš dlouhých a nebo pro vyřazení příliš krátkých TM šroubovic. [22]

Byla zhodnocena senzitivita a specificita této nové metody používající NS pomocí TM Benchmark serveru. Metoda MINNOU serveru byla poměřována s metodami: PHDhtm, HMMTOP2, TMHMM1, DAS, TopPred2 a SOSUI. MINNOU dosahuje v rámci jednoho aminokyselinového zbytku nejvyšší přesnosti (89 %). Přesnost předpovědí pro segmenty proteinů dosáhl přesnosti 80 %, což je méně než u PHDhtm a HMMTOP2. Některé výše

zmíněné metody mají též vyšší míru přesnosti pro aminokyselinový zbytek (89-90 %) nežli MINNOU (85 %) pro data s nízkým prostorovým rozlišením 3D struktury α-šroubovic. Tato data nebyla zahrnuta do trénovacího setu pro MINNOU server. [22]

Evaluace pomocí TMH Benchmarku odhalila také vyšší míru nepřesnosti předpovědí pro gloubulární proteiny a mírně i pro signální peptidy. Nicméně se MINNOU server v tomto aspektu ukázal zřetelně lepší než zbylé výše zmíněné metody. Výjimkou je metoda Phobius, se kterou je srovnatelný. [27]

Výraznou přesnost předpovědi měla metoda MINNOU serveru pro iontové kanály. Pro 7 iontových kanálů zahrnutých v setu proteinů s vysokým rozlišením 3D struktury pro tréning dosáhl přesnosti 92 % pro aminokyselinový zbytek a korelační koeficient 0,81, což jsou lepší hodnoty nežli např. pro HMMTOP2 a DAS metody. [22]

Pomocí křížové validace byla odhadnuta přesnost metody MINNOU serveru v případě NS jako klasifikátoru s jednou vnitřní, skrytou vrstvou neuronů pro predikci TM šroubovic na 74 %. Výsledný protokol pro predikci TM šroubovic založený na dvoustupňovém NS klasifikátoru byl otestován pomocí TMH Benchmark serveru. Získaná přesnost je 89 % pro jednotlivé aminokyselinové zbytky a pro segmenty proteinů 80 %. MINNOU server měl též nejnižší míru chyby předpovědi pro globulární proteiny a signální peptidy ve srovnání s výše uvedenými dalšími metodami a podobnu míru chybovosti jako metoda Phobius. [22,27]

2.2.5.6 CCTOP

CCTOP (Consensus Constrained TOPology prediction) server (dostupný na: http://cctop.enzim.ttk.mta.hu) je webová aplikace, poskytující predikci transmembránové (TM) topologie, tj. TM α-šroubovic. Využívá mj. 10 „state-of-the-art“ predikčních metod topologie. CCTOP server zahrnuje topologické informace z existujících experimentálních a výpočetních zdrojů dostupných v databázích PDBTM [28, 29, 30], TOPDB [31, 32] a TOP-DOM [33]. Tyto databáze používají pravděpodobnostní rámec skrytých Markovových modelů (HMM, hidden Markov model). Server poskytuje možnost nejprve predikovat signální peptidy a rozlišení mezi TM úsekem a globulární doménou. [34]

Úvodní výsledky mohou být přepočítány vybráním nebo vypnutím některé z predikčních metod nebo mapovacích experimentů nebo může uživatel zadat některá vlastní omezení. Dle [34] vykazuje nejvyšší výkony v rámci tehdejších možností (rok 2015). [34]

CCTOP server byl vytvořen pro predikci lidského transmembránového proteomu, který je obsažen v HTP (Human Transmembrane Proteome) databázi [35]. Pokud je v databázích experimentálních a výpočetních zdrojů PDBTM, TOPDB a TOP-DOM obsažena již informace se segmenty se známou topologií nebo obsahují některé homologní proteiny, je tato informace přidána do omezení v HMM. [34]

Metoda CCTOP serveru se skládá ze tří hlavních kroků: odstranění sestřihových míst ze zadané sekvence, filtrování proteinů a predikce topologie. Segmenty signálních peptidů bývají často zaměněny za TM segmenty, proto je nejprve zahrnuta detekce těchto úseků

pomocí programu SignalP 4.0 [36]. Tento krok může být přeskočen, pokud je nalezen homologní protein databázi TOPDB, který nemá tuto signální sekvenci. [34]

Dalším krokem po odstranění sestřihových míst je odlišení globulárních a TM proteinů. K tomu jsou použity nástroje Phobius [21], Scampi-single [37] a TMHMM [44]. Jestliže jakékoliv dvě z těchto metod označí segment jako transmembránový, pak je protein klasifikován jako TM protein. [34]

Pro konsensus predikce topologie bylo vybráno deset metod v závislosti na jejich dostupnosti a výkonosti v různých benchmark testech: HMMTOP [19,38], MemBrain [39], MEMSAT-SVM [40], Octopus [41], Philius [42], Phobius [21], Pro- a Prodiv-TMHMM [43], Scampi-MSA [37] a TMHMM [18,44]. [34]

Výsledky predikcí těchto metod jsou použity jako podklad pro ten samý HMM jaký je použit v HMMTOP ale s jinými vahami. Váhy závisejí na přesnosti z každé z použitých metod měřených na benchmarku se setem dat z databáze Human Transmembrane Proteome [35].

Pro další zpřesnění předpovědí jsou pro každou zadanou sekvenci sesbírány informace o homologních proteinech z databáze PDBTM, experimentech na homologních sekvencích z databáze TOPDB a konzervativně lokalizovaných sekvencích a motivů z databáze TOP-DOM. Pokud jsou v některé z těchto databází příslušné informace, jsou automaticky sesbírány a včleněny do pravděpodobnostního rámce poskytnutého ve skrytém Markovovo modelu (HMM), popsaném v [45] a na stránkách serveru CCTOP (http://cctop.enzim.ttk.mta.hu). [34]

Pro stanovení spolehlivosti predikce jsou sečteny posteriorní pravděpodobnosti z HMM pro každý hlavní typ skrytého stavu (intracelulární, membránový, smyčka a mimobuněčný) v každé pozici sekvence TM proteinu. Spolehlivost je pak průměr z těchto součtů cesty HMM s nejpravděpodobnějšími stavy dané Viterbiho algoritmem, viz [46] a manuál na stránkách CCTOP serveru (viz výše). [34]

Aby bylo možné zvládnout vysoké nároky na výpočetní výkon u těchto náročných úkolů s vícero metodami, byla vytvořena vícevrstvá aplikační architektura. Čas do vytvoření predikce kolísá od několika málo minut až asi po 30 minut v závislosti na délce zadané sekvence a zatížení serverů HPC klastru pro CCTOP. Výpočetní část je předána do vyhrazené fronty s vyváženým zatížením HPC klastru, kde jsou izolovány některé uzly pro vyhovění těmto požadavkům. K dispozici jsou dvě různá rozhraní (GUI) pro použití CCTOP: webový server s GUI vhodným pro webové browsery napsaný v C++ za použití Wt knihovny pro programování pro web spolu s XBuiler knihovnou dříve vyvinutou autory serveru CCTOP. Druhým je nenáročné rozhraní pro skripty založené na PHP. Výsledky jsou graficky zobrazeny a mohou být také staženy ve formátu XML. [34]

2.2.5.7 tmhmm.py 1.3.1

Nástroj tmhmm.py 1.3.1 (dostupný na: https://pypi.org/project/tmhmm.py/) je implementací skrytého Markovova modelu (HMM, hidden Markov model) pro vyhledávání

transmembránových úseků proteinů. Toto použití HMM bylo popsáno prvně v [18]. Stejný HMM je popsán v části 2.2.5.1 TMHMM Server v. 2.0. Nástroj tmhmm.py 1.3.1 je určen pro Python verze 3.5 a vyšší a je spustitelný jen v rámci operačního systému Linux. Tato implementace zahrnuje též velmi rychlý skript v jazyce Cython, který využívá Viterbiho algoritmus [46] pro anotaci proteinové sekvence. Výstupní soubory tohoto nástroje jsou podobné jako u originální implementace dostupné na: http://www.cbs.dtu.dk/services/TMHMM/ a https://services.healthtech.dtu.dk/service.php?TMHMM-2.0. [81]

Původní implementace ošetřuje možný výskyt ambiguitního kódu a mezer nedokumentovaným způsobem. Nástroj tmhmm.py 1.3.1 tuto možnost vůbec neošetřuje a v této situaci prostě selže. Jediná oprava spočívá v nahrazení neznámých znaků, znaky nebo sekvencí založeném na odborné znalosti uživatele. [81]

Při tvorbě programového nástroje pro tuto diplomovou práci byl použit mj. právě nástroj tmhmm.py 1.3.1.

2.2.5.8 pyTMHMM 1.3.2

Nástroj pyTMHMM 1.3.2 (dostupný na: https://pypi.org/project/pyTMHMM/) je implementací skrytého Markovova modelu (HMM, hidden Markov model) pro vyhledávání transmembránových úseků proteinů. Toto použití HMM bylo popsáno prvně v [18]. Stejný HMM je popsán v části 2.2.5.1 TMHMM Server v. 2.0. pyTMHMM 1.3.2 je další verzí nástroje tmhmm.py 1.3.1. Nástroj pyTMHMM 1.3.2 je určen pro Python verze 3.5 a vyšší a je spustitelný jen v rámci operačního systému Linux. Tato implementace zahrnuje též velmi rychlý skript v jazyce Cython, který využívá Viterbiho algoritmus [46] pro anotaci proteinové sekvence. Výstupní soubory tohoto nástroje jsou podobné jako u originální implementace dostupné na: http://www.cbs.dtu.dk/services/TMHMM/ a https://services.healthtech.dtu.dk/service.php?TMHMM-2.0. [82]

Původní implementace ošetřuje možný výskyt ambiguitního kódu a mezer nedokumentovaným způsobem. Nástroj pyTMHMM 1.3.2 tuto možnost vůbec neošetřuje a v této situaci prostě selže. Jediná oprava spočívá v nahrazení neznámých znaků, znaky nebo sekvencí založeném na odborné znalosti uživatele. [82]

2.2.6 Seznam nástrojů pro vyhledávání coiled-coil sekundárních proteinových struktur v aminokyselinových sekvencích

Nejužívanější nástroje pro vyhledávání coiled-coil sekundárních proteinových struktur v aminokyselinových sekvencích jsou v současné době následující:

- COILS version 2.2,
- MARCOIL,
- Multicoil2,

a

- DeepCoil 2.0.1.

Existují i další nástroje. Ty lze považovat za nástroje s relativně minoritním použitím.

2.2.7 Podrobnější popis nástrojů pro vyhledávání coiled-coil sekundárních proteinových struktur v aminokyselinových sekvencích

2.2.7.1 COILS version 2.2

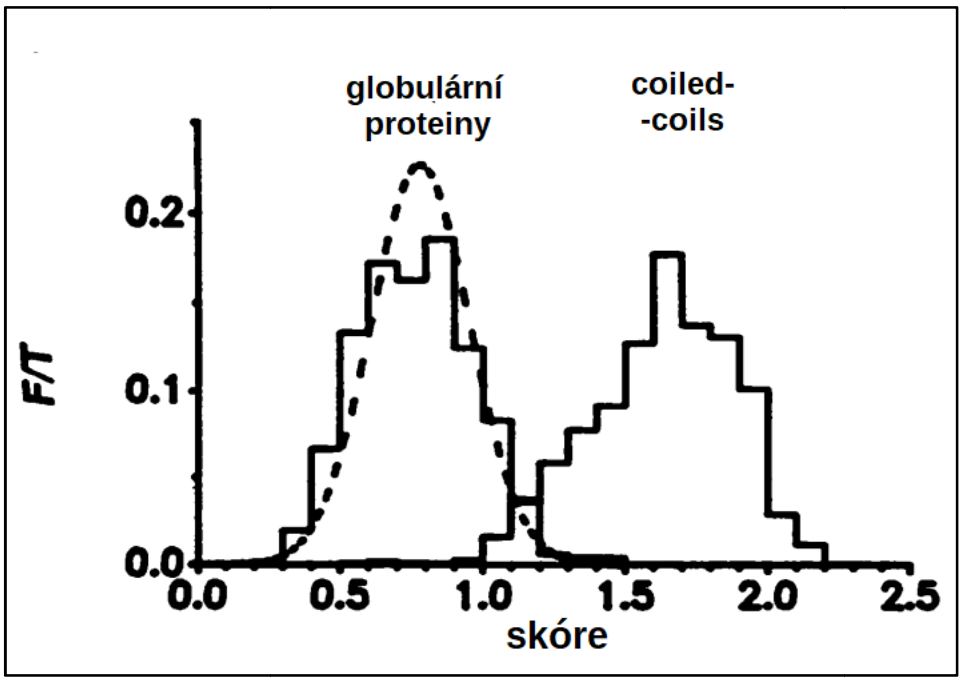
Pravděpodobnost, že aminokyselinový zbytek, residuum v proteinu, je součástí coiled-coil (CC) sekundární struktury, může být zhodnocena porovnáním jeho sousedních sekvencí se sekvencemi proteinů s již stanovenou coiled-coil strukturou. CC struktura (viz také kapitola 2.1.2) je tvořena pomocí opakující se sekvence (obvykle) sedmi aminokyselinových zbytků označovaných „a“ až „g“, kde „a“ a „d“ bývají hydrofobní. [47]

Pro vyhledávání CC struktur byly nejprve stanoveny relativní frekvence aminokyselin v CC strukturách. Tyto relativní frekvence byly použity pro vyhodnocení sekvencí pomocí posuvného okna o délce 28 aminokyselinových zbytků. Tato délka posuvného okna byla zvolena proto, že ty nejkratší peptidy, které ještě vykazují CC strukturu, bývají složeny ze čtyř nebo pěti opakování skupiny sedmi aminokyselin. Předběžné skóre pro každý aminokyselinový zbytek v rámci posuvného okna bylo počítáno přiřazením každému aminokyselinovému zbytku relativní frekvenci a spočtením geometrického průměru všech těchto frekvencí v rámci daného umístění posuvného okna. Jelikož se jedná o opakující se skupiny sedmi aminokyselin, může mít posuvné okno sedm různých pozic a aminokyselinový zbytek může být v jakékoliv pozici v rámci 28 míst posuvného okna, pak je celkem 196 předběžných skóre pro jeden aminokyselinový zbytek. Z nich je vždy vybráno to s nejvyšší

hodnotou pro daný aminokyselinový zbytek. Pokud se aminokyselinový zbytek nacházel blíže jak 28 pozic od konce vyšetřované sekvence, bylo stanoveno méně předběžných skóre. [47]

COILS je program, který porovnává zadanou sekvenci se sekvencemi proteinů se známými dvouvláknovými coiled-coil strukturami a odvozuje skóre podobnosti. Porovnáním tohoto skóre s distribucí skóre pro globulární a pro coiled-coil struktury vypočítává program pravděpodobnost, že daná sekvence zaujme coiled-coil konformaci. [48]

Na obrázku 2.10 níže je zachycen rozdíl v distribuci skóre mezi globulárními proteiny a coiled-coil strukturami.



Obr. 2.10: Distribuce skóre mezi globulárními proteiny a coiled-coil strukturami (dle: [47])

Program COILS jako vstupní formáty akceptuje formát CGC, FASTA formát a uživatel si může také v rámci několika omezení vytvořit vlastní formát. [48]

Program též poskytuje čtyři možnosti formy výstupu. Defaultní varianta poskytuje pořadové číslo aminokyselinového zbytku, jeho typ, rámec a pravděpodobnost tvorby coiled-coil struktury. Výsledek je prezentován ve formě sloupců. Druhá možnost „a“ je podobná defaultní. Jen s tím rozdílem, že výsledek je zobrazen ve formě řádků a pořadová čísla aminokyselinových zbytků jsou ukázána škálou nad zobrazenou sekvencí. Zobrazení určených pravděpodobností je zkráceno na první číslici pravděpodobnosti. Možnost „b“ se ptá uživatele na velikost posuvného okna a poskytuje jen skóre. To dává možnost uživateli prohlédnout si jen skóre jinak „schované“ v předchozích možnostech za pravděpodobnostmi. Možnost „c“ je užitečná pro skenování velmi velkých proteinů nebo souborů s mnoha proteiny, kdy zobrazení je stejné jako v defaultní verzi, ale jsou zobrazeny jen oblasti s pravděpodobnostmi pro coiled-coil struktury nad určitou prahovou hodnotu, kterou nastaví uživatel. [48]

Program je založen na použití „pozičně specifické skórovací matice“ (PSSM, Position Specific Scoring Matrix) s pevnou délkou obvykle 28 residuí. Před zpracováním zadané sekvence program nabídne dvě možné skórovací matice, se kterými může porovnávat sekvenci. MTK matice je odvozena od sekvence tropomyosinů, myosinů, keratinů (tj. intermediárních filament typu I a II). Druhá matice, matice MTIDK, je odvozena od sekvencí myosinů, paramyosinů, tropomyosinů, intermediárních filament typu I až V, desmosomálních proteinů a kinesinů. [48]

Matice MTIDK poskytuje o něco lepší rozlišení mezi globulárními proteiny a proteiny s coiled-coil strukturou. MTK matice je více specifická pro dvouvláknové struktury. MTIDK matice poskytuje realističtější zhodnocení jiných typů CC struktur. [48]

Program COILS je specifický pro levotočivé CC struktury vystavené vodnímu prostředí. Jiné typy jako CC uvnitř proteinů, TM šroubovice a jiné pravotočivé šroubovice program nedetekuje. [48]

Také neposkytuje rozhodnutí typu ano-ne dle prahové hodnoty. Poskytuje set pravděpodobností popisujících potenciál tvořit CC strukturu. To znamená, že i při vysoké pravděpodobnosti, např. 90 %, nemusí daný úsek proteinu tvořit CC strukturu. [48]

Rozdíl v distribuci skóre mezi globulárními proteiny a proteiny s CC strukturou klesá velmi rychle se zmenšující se délkou posuvného okna. Pro vyhledávání nových CC úseků proteinů je vhodné použít okno o délce 28 aminokyselinových zbytků, residuí, ve speciálních případech o délce 21 residuí. Okno o délce 14 residuí by mělo být použito jen pro analýzu lokálních parametrů v oblastech s nalezenou CC strukturou. [48]

Sekvence s vysokou pravděpodobností tvorby CC struktury u globulárních proteinů málokdy přesahují délku 30 residuí a nebývají delší nežli 35 residuí. Pokud se vyskytne sekvence o délce více jak 35 residuí s pravděpodobností nad 80-90 %, je vhodné předpokládat, že pravděpodobnost tvorby CC struktury je zde vyšší nežli předpovězená. [48]

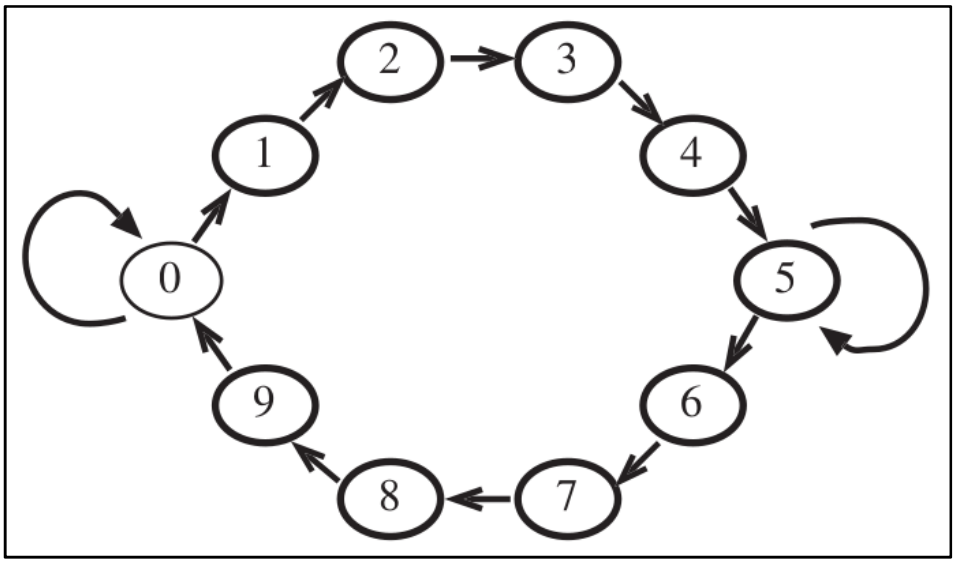
2.2.7.2 MARCOIL

Mnoho prediktivních metod je založeno buďto na „pozičně specifické skórovací matici“ (PSSM, Position Specific Scoring Matrix) s pevnou délkou nebo na skrytém Markovovu modelu (HMM, Hidden Markov Model) bez okna. [49]

MARCOIL je nástroj založený na HMM pro rozpoznání proteinů s coiled-coil (CC) sekundární strukturou v genomovém měřítku. MARCOIL poskytuje o něco lepší predikce nežli tradiční PSSM algoritmus především pro některé proteinové rodiny a pro krátké úseky s CC strukturami. Tento predikční program a databáze jsou dostupné na: http://www.wehi.edu.au/bioweb/Mauro/Marcoil. [49]

HMM se staly standardní technikou při analýze sekvencí, kdy jsou založeny na konzistentním pravděpodobnostním rámci, pro nějž je známa řada dobrých algoritmů. Jejich aplikace není přímočará a výpočetně jsou náročnější, tudíž jsou pomalejší, ale zároveň jsou flexibilnější. [49]

MARCOIL obsahuje 64 stavů. Prvním z nich je referenční, základní stav indikovaný číslem 0. Dalších 63 stavů jsou označeny číslem skupiny (1-9) a písmenem („a“ až „g“) odpovídající jedné z pozic ve skupině opakujících se sedmi aminokyselin („heptadu“). Skupiny 1 až 4 modelují první 4 aminokyselinové zbytky, residua v CC doméně (tj. blíže N-konci proteinu), skupiny 6 až 9 poslední 4 residua (tj. blíže C-konci proteinu). Vnitřní residua heptadu jsou ve skupině 5, která může navázat sama na sebe a tak se heptad může vícekrát opakovat. Náčrt povolených přestupů mezi skupinami stavů je na obr. 2.11. Např. sekvence ze dvou heptadů začínající „b“ pozicí vyžaduje následující přestupy mezi stavy: 0-1b-2c-3d-4e-5f-5g-5a-5b-5c-5d-6e-7f-8g-9a-0. [49]



Obr. 2.11: Přehled povolených přechodů mezi stavy MARCOIL-HMM

(dle: [49])

Při každém začátku se vychází ze stavu 0 do jednoho ze sedmi stavů první skupiny. Stavy skupiny 5 jsou napojeny jednak opět na skupinu 5 tak na skupinu 6. Přechody v rámci skupiny 5 jsou důležité pro pokračování CC domény ve vícero heptadech a přechod do skupiny 6 odpovídá popisu posledních 4 residuí. [49]

Rychlost nástroje MARCOIL je limitovaná. Náročnost výpočtů roste sice lineárně s délkou sekvence a množstvím sekvencí pro zpracování ale vyžaduje mnohem více operací nad jedním residuem. MARCOIL má celkem 4096 potenciálních přechodů, používáno je jen 456 a každý z nich vyžaduje kolem 912 násobení na jeden aminokyselinový zbytek. Přesto lze MARCOIL použít pro rozsáhlé aplikace, např. pro popis všech známých a předpovězených proteinů lidského genomu. [49]

2.2.7.3 Multicoil2

α-šroubovicové coiled-coil (CC) struktury mohou obecně zaujmout více různých topologií včetně paralelních a antiparalelních dimerů a trimerů. Multicoil2 dokáže určit lokalizaci CC struktury a také stupeň oligomerizace (dvě nebo tři šroubovice). Využívá flexibility skrytého Markovova modelu (HMM, Hidden Markov Model) v Markovově náhodném poli (MRF, Markov Random Field). [50]

Program Multicoil2 a trénovací databáze jsou přístupné na: http://multicoil2.csail.mit.edu po vyžádání. [50]

Multicoil2 je postaven na algoritmu nástroje Paircoil [51,52], který využívá pravděpodobnostní rámec k detekci CC struktury. Využívá známé frekvence párů aminokyselinových zbytků, residuí v proteinech se známou CC strukturou. Multicoil (předchůdce Multicoil2) používá dvě databáze sekvencí. Jednak se sekvencemi dimerů a dále se sekvencemi trimerů. Z těchto databází odvozuje frekvence párů residuí, které mohou být použity pro stanovení tendence daného úseku proteinu tvořit CC strukturu (dimery nebo trimery). Tento přístup je limitován použitím posuvného okna s pevnou délkou (obvykle 21 nebo 28 residuí). [50]

Multicoil2 je určen především pro predikci stavu oligomerizace CC domén. Využívá kombinace pravděpodobnostních modelů posuvného okna a HMM v Markovově náhodném poli (MRF, Markov Random Field). Ze setu trénovacích proteinových rodin byly spočteny různé rysy sekvencí residuí, což je využito pro predikci dimerů a trimerů CC struktur. Tyto predikce generují potenciály pro MRF. MRF pak zpracovává aminokyselinové sekvence a vrací pravděpodobnosti stavů po jednotlivých residuích. Parametry MRF jsou optimalizovány pro vysokou výkonost užitím logistické regrese. Tato regrese spoléhá na osm sekvenčních rysů pro předpověď CC struktur nebo jejich stav oligomerizace. Rysy, které jsme shledali užitečné, jsou: pravděpodobnost dimeru, pravděpodobnost trimeru, pravděpodobnost, že sekvence nekóduje CC strukturu, korelace v rámci dimeru na pozicích 1 až 7, korelace v rámci trimeru na pozicích 1 až 7, korelace sekvence bez CC struktury na pozicích 1 až 7, hydrofobicita residuí na pozici „a“ a „d“ (pozice 1 a 4). [50]

Jsou také použity další rysy včetně délky CC šroubovice, proměnné pro náboje residuí na různých místech opakující se skupiny sedmi residuí (heptad), velikost residuí na různých místech heptadu, a započtené frekvence jednotlivých residuí hned před začátkem a za koncem CC šroubovice. Tyto zadávané rysy ale nezvyšují výrazně schopnost modelu pro predikci CC struktury. [50]

Multicoil a Multicoil2 mohou být využity pro odlišení CC struktur a šroubovic, které netvoří CC strukturu. Odlišení je dáno zhodnocením celkové CC pravděpodobnosti pro každou pozici daného residua vůči sumě předpovězených pravděpodobností pro dimery a trimery CC domén. Algoritmus nástroje Multicoil2 překonává dřívější Multicoil a také např. dřívější nástroj Paircoil2. Multicoil2 má falešnou pozitivitu na 0,3 % a 91,8% schopnost detekovat CC struktury v testovacích pozitivních a negativních databázích. Multicoil2 je

výkonný především v oblasti šedé zóny pro anotaci struktur oproti použití čistě HMM, který v takovýchto situacích obvykle selhává. Celkově jsou predikce pomocí Multicoil2 dobré, ale liší se výrazně mezi některými jednotlivými rodinami CC proteinů. Každá rodina CC proteinů se liší množstvím sekvencí a residuí. Proteinové rodiny se špatnými predikcemi mají své specifické rysy sekvencí residuí. Některé rodiny mohou mít rysy sekvencí typické jak pro dimery a trimery CC struktur. Takové sekvence mohou často tvořit jak dimery tak trimery CC domény. [50]

2.2.7.4 DeepCoil 2.0.1

DeepCoil je relativně nově vyvinutý nástroj pro vyhledávání coiled-coil (CC) struktur založený na neuronových sítích [53]. V benchmarku provedeném autory programu vykazoval výrazně lepší výsledky nežli současné nástroje (v době psaní článku [53]) např. PCOILS [54] a Marcoil [49] jak v predikci kanonických tak nekanonických CC domén. Dále při použití DeepCoil na lidském genomu bylo detekováno mnoho CC domén, které zůstávaly nedetekovány jinými metodami. Vyšší senzitivita nástroje DeepCoil umožňuje jeho aplikaci na přesnou detekci CC domén v rámci celých genomů. [53]

DeepCoil je nástroj založený na neuronových sítích. Je určen pro predikci kanonických i nekanonických CC domén. Hledání pomocí DeepCoil může být založeno na zjišťování sekvence (DeepCoil_SEQ) nebo na sekvenčním profilu (DeepCoil_PSSM). DeepCoil_SEQ používá jako vstup samostatnou proteinovou sekvenci. DeepCoil_PSSM k dané proteinové sekvenci používá také evoluční informaci získanou z mnoha porovnání sekvencí. [53]

Neuronová síť programu DeepCoil byla implementována v nástroji Keras [55]. Tato síť se skládá mj. ze dvou konvolučních vrstev, každá s 64mi filtry. První vrstva skenuje sekvenci posuvným oknem o 28 aminokyselinových zbytcích, druhá o 21 residuí. Na konvoluční vrstvy navazuje hustě propojená vrstva 128mi neuronů a výstupní vrstva. Pro aktivaci většiny vrstev byla použita „ReLU“ aktivační funkce. Výjimkou je výstupní vrstva, kde byla použita funkce „softmax“. Pro trénování neuronové sítě nástroje DeepCoil bylo použito 10 438 proteinových struktur s 4140 nepřerušovanými CC doménami. [53]

Při provedení důkladného benchmarku vykazovaly obě verze DeepCoil lepší výkon nežli ostatní nástroje v době psaní článku [53]. DeepCoil byl porovnáván s nástroji COILS [47], PCOILS [54], Marcoil [49], Multicoil2 [50] a CCHMM_PROF [56]. [53]

2.2.8 Seznam nástrojů pro vyhledávání proteinů s GPI modifikací v aminokyselinových sekvencích

Používané nástroje pro vyhledávání proteinů s GPI (glycosyl-phosphatidylinositol) modifikací v aminokyselinových sekvencích jsou v současné době následující:

- Big-PI Predictor,
- PredGPI,
- NetGPI – 1.1,
- GPI-SOM.

K nástroji Big-PI Predictor není dostupná literatura zabývající přímo popisem samotného Big-PI Predictoru. Dále dříve také používaný nástroj FragAnchor již není dostupný on-line. Ostatní nástroje zde nepopisované lze považovat za nástroje s relativně minoritním použitím.

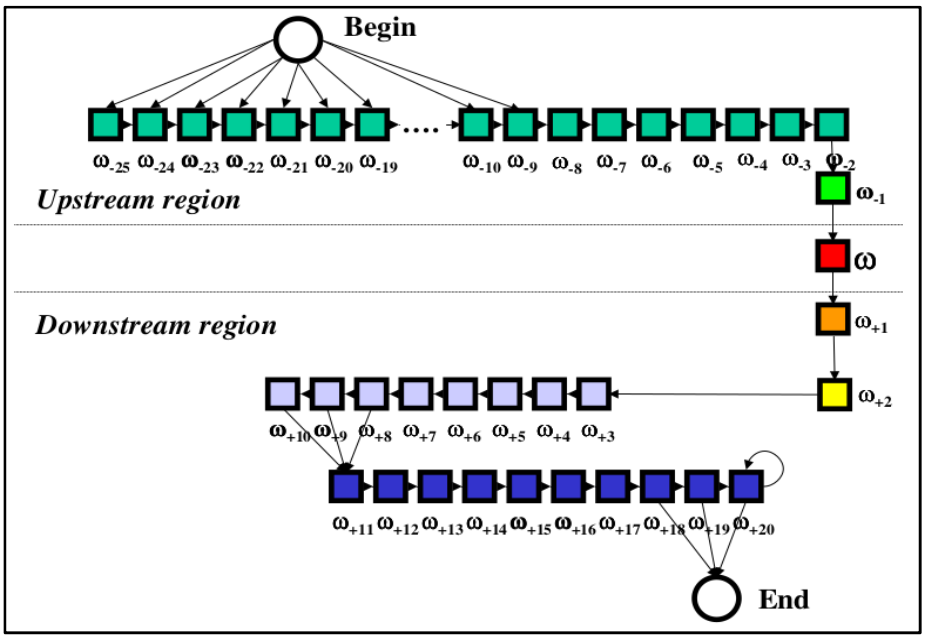
2.2.9 Podrobnější popis nástrojů pro vyhledávání proteinů s GPI modifikací v aminokyselinových sekvencích

2.2.9.1 PredGPI

PredGPI je predikční metoda, která spojuje skrytý Markovův model (HHM, Hidden Markov Model) a Support Vector Machine diskriminátor (SVM diskriminátor). PredGPI dokáže určit přítomnost místa pro GPI modifikaci, tj. přesně určit pozici střihu tzv. ω-místo. Trénování proběhlo na neredundantním setu dat s experimentálně ověřenými proteiny s GPI kotvou. [57]

Hlavní rysy charakterizující C-koncovou část proteinu s GPI modifikací mohou být převedeny do skrytého Markovova modelu (HMM). HMM je model sestavený ze stavů, kdy každý reprezentuje jednu pozici podél sekvence. Typické kompozice residuí v různých oblastech sekvencí jsou popsány průměry pravděpodobností přiřazených každému stavu. Stavy jsou dále propojeny pravděpodobnostmi přechodu mezi stavy. [58] [57]

Na obrázku 2.12 je zobrazen model určený k popisu C-koncového segmentu o délce 40 residuí u proteinu s GPI modifikací. Obsahuje 46 stavů se středem na ω-místě.. Stavy zobrazené na obrázku 2.12 stejnou barvou mají stejné parametry, takže model popisuje rozdílné zóny pomocí rozdílných kompozicí residuí. [57]



Obr. 2.12: HMM model C-koncové části proteinu [57]

ω-místo, jedno residuum „proti proudu“ („upstream“) a dvě residua „po proudu“ („downstream“) jsou popsány nezávislým rozdělením pravděpodobností. Oblast proti proudu je popsána jedním setem pravděpodobností. Oblast po proudu dvěma sety pravděpodobností. Jsou také použity dva speciální stavy pro začátek a pro konec procesu, které neposkytují žádný znak. Topologie přechodů popisuje C-koncový odstřižený propeptid (část sekvence následující po proudu za ω-místem) o délce více jak 16 residuí a modeluje experimentálně zjištěné délky tohoto úseku. [57]

Pro danou sekvenci je spočtena pravděpodobnost dle HMM a ta je poskytnuta jako vstup pro SVM diskriminátor (viz dále). [57]

SVM (Support Vector Machine) diskriminátor byl poprvé uveden v [59]. SVM diskriminátor je schopen optimálně rozlišit mezi dvěma třídami objektů. Vstupy jsou zakódovány číselným vektorem a poté mapovány na *h*-dimenzionální prostor *H*. Algoritmus SVM je schopen vytvořit (*h*-1)-dimenzionální nadrovinu v *H* prostoru pro rozlišení objektů do dvou tříd. Pro rozlišení proteinů s a bez GPI kotvy byla použita SVM-light implementace SVM volně dostupná na: http://svmlight.joachims.org. Vstup kombinuje pravděpodobnost výstupu HMM modelu s informací odvozenou od celé sekvence, z C-koncového a z N-koncového úseku. Je vytvořen vektor pro každou sekvenci residuí z 83 elementů a popisuje celkové složení sekvence, rysy sekvence na N-konci zahrnující signální peptid a C-konec se signálem pro sestřih. [57]

Byla porovnána výkonost PredGPI s jinými veřejně dostupnými prediktory. Jmenovitě BIG-PI [60], DGPI [61], GPI-SOM [62], FragAnchor [63] a Mem.Type-2L [64]. [57]

BIG-PI je první veřejně zpřístupněná metoda pro predikci GPI modifikací. Nabízí 4 prediktory pro různé říše živočichů a prvoků. Tato metoda je schopna rozpoznat jen

polovinu proteinů s prokázanou GPI kotvou. Četnost falešně pozitivních případů má jen 0,3 %. Prediktory DGPI a GPI-SOM dokážou rozpoznat větší podíl proteinů s GPI modifikací, ale podíl falešně pozitivních případů je u nich velmi vysoký (2,3 % u DGPI a 1,7 % u GPI-SOM). FragAnchor dokáže zachytit 70 % proteinů s GPI kotvou a míru falešně pozitivních případů má stejnou jako BIG-PI prediktor. PredGPI dokáže zachytit 77 % proteinů s GPI modifikací a četnost falešně pozitivních případů má poloviční oproti BIG-PI a FragAnchor. Při zvýšení prahu pro četnost falešně pozitivních jen na 0,5 % dosáhl PredGPI prediktor schopnosti identifikovat 89 % proteinů s GPI kotvou. Úspěšnost jiných metod sahá od 84 % správně předpovězených proteinů u prediktoru FragAnchor po 43 % u prediktoru Mem.Type-2L. [57]

Po tréningu na všech dostupných proteinech získaných datasetů se PredGPI prediktor mýlil v stanovení ω-místa jen ve dvou případech z 26ti. V obou případech se PreGPI mýlil jen o jednu pozici. DGPI byl schopen správně předpovědět ω-místo v 17ti z 26ti a GPI-SOM v 15ti z 26ti případů. BIG-PI prediktor má obdobnou účinnost v předpovědi ω-místa jako PredGPI, kdy je správně určil ve 23 případech z 26ti. [57]

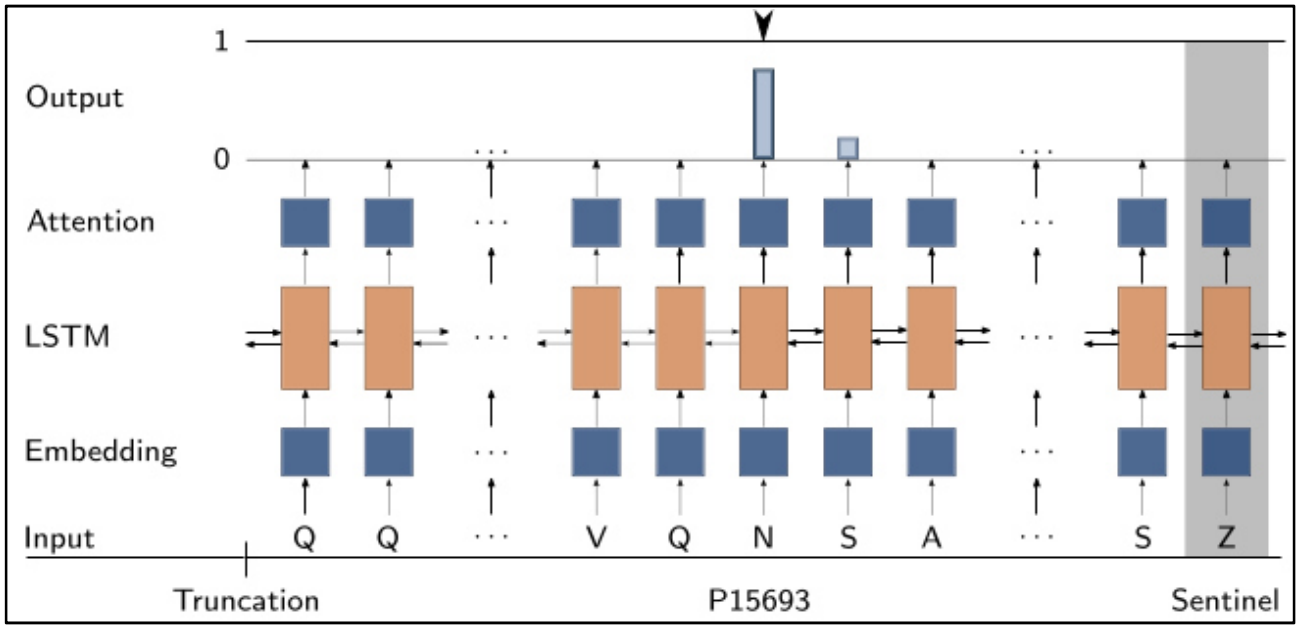
Prediktor PredGPI je volně dostupný na: http://gpcr.biocomp.unibo.it/predgpi. Pro každý zadaný protein predikční systém poskytuje nejpravěpodobnější ω-místo spolu s mírou pravděpodobnosti přítomnosti GPI kotvy vyjádřené jako index specificity (počítaný jako: 1 – FP, FP = podíl falešně pozitivních předpovědí). Pokud je index specificity vyšší jak 99,9 %, je predikce označena jako „vysoce pravděpodobná“. Pokud je index specificity je mezi 99,9 % a 99,5 %, pak je predikce označena jako „pravděpodobná“. Pokud je index specificity mezi 99,5 % a 99,0 % je predikce označena jako „méně pravděpodobná“. Uživatel si může vybrat mezi konzervativním a nekonzervativním HMM pro predikci pozice ω-místa. [57]

2.2.9.2 NetGPI – 1.1

NetGPI je založen na rekurentních neuronových sítích a použití aktuálních setů dat k tréningu neuronové sítě. Simultánně detekuje signál v aminokyselinové sekvenci proteinu pro GPI modifikaci a zjišťuje též pozici ω-místa. K použití přes webové rozhraní je NetGPI dostupný na: https://services.healthtech.dtu.dk/service.php?NetGPI. Úložiště kódu je dostupné na: https://github.com/mhgislason/netgpi-1.1. [65]

NetGPI nezjišťuje přítomnost signálního peptidu na N-konci proteinu. Spoléhá na to, že ke C-konci proteinu se nacházejí sekvence s dostatečnou informací pro určení GPI modifikace. Proto také vyšetřuje jen posledních 100 C-koncových aminokyselinových zbytků. Při predikci GPI modifikace jsou dva úkoly. Stanovit, zda je přítomen signál pro GPI modifikaci a určit místo sestřihu tj. ω-místo. V NetGPI jsou tyto dva úkoly shrnuty do jednoho, tj. do maximalizace pravděpodobnosti polohy v sekvenci. K parametrizaci distribuce podmíněné pravděpodobnosti byla použita neuronová síť známá jako Long-Short Term

Memory (LSTM) [66] a distribuovaná reprezentace aminokyselin [67]. Viz obrázek 2.13 níže. [65]



Obr. 2.13: Diagram modelu použité neuronové sítě NetGPI [65]

Na obrázku 2.13 výše je diagram modelu použité neuronové sítě NetGPI ilustrující jak model vybere jednu pozici v sekvenci jako nejpravěpodobnější ω-místo, tj. místo sestřihu. V tomto případě je použit protein P15693 z UniProt databáze. Z proteinu je ponecháno jen posledních 100 C-koncových residuí a je ke konci sekvence je přidán tzv. sentinel. Předpovězené ω-místo je v tomto případě Asparagin (N). Pokud by byla určena jako nejpravděpodobnější pozice sentinelu, byl by protein označen jako protein bez možné GPI modifikace. [65]

Byl proveden benchmark ve srovnání se všemi třemi nástroji dostupnými v době psaní článku [65]: Big-Π, GPI-SOM a PredGPI. Při tomto benchmarku NetGPI např. dosáhl nejvyšší četnosti správně pozitivních (TPR, true positive rate) případů 0,975. GPI-SOM měl druhou nejvyšší míru TPR (0,950). NetGPI také dosáhl nejvyšší přesnosti 0,834, druhou nejvyšší měl nástroj Big-Π (0,830). Dále Big-Π měl nejnižší míru falešně pozitivních případů 0,010. NetGPI měl druhou nejnižší míru falešně pozitivních případů (0,012). Dále např. v trénovacím setu nástroje Big-Π bylo nově nalezeno 17 proteinů s GPI modifikací. Z nich správně NetGPI určil 8, Big-Π určil 8 a GPI-SOM určil správně 9 ze 17ti těchto proteinů. [65]

2.2.9.3 GPI-SOM

Pro identifikaci *in silico* signálů C-koncové části proteinupro GPI modifikaci byla natrénovaná neuronová síť na proteinech s prokázanou GPI kotvou a při tréningu neuronové sítě byly průběžně optimalizovány vstupní parametry. Výsledkem je Kohonenova samo-organizující se mapa nazvaná GPI-SOM (GPI Self Organizing Map). GPI-SOM předvídá GPI proteiny s vysokou přesností. V kombinaci s nástrojem SignalP byl GPI-SOM použit pro vyhledávání GPI ukotvených proteinů při prohlídkách celých genomů různých eukaryotů. C-koncový signál pro připojení GPI kotvy má smysl jen v kontextu současně se vyskytující N-terminální exportní sekvence. Nástroj pro predikci této sekvence je např. SignalP, což je program využívající skrytý Markovův model a neuronovou síť [68]. S výjimkou specializovaných parazitů je patrný trend pro zvyšující se podíl GPI ukotvených proteinů ve větších genomech. [62]

GPI-SOM je dostupný on-line na: http://gpi.unibe.ch a zdrojový kód (napsaný v jazyce C) je též dostupný na stejné webové stránce. Vstup sekvencí je možný jen ve FASTA formátu. [62]

Neuronové sítě Kohonenova typu jsou též nazývány samo-organizující se mapy (SOM, self-organizing maps). Jedná se o poměrné výkonné nástroje pro klasifikaci informací skrytých ve velkých setech dat. Učení SOM probíhá ve formě přenastavování vah spojení (synapsí) mezi jednotkami (neurony). Jedná se o učení neuronové sítě bez učitele. Výhodou je nulové uplatnění předpojatosti člověka. SOM rozliší vzorce v setech dat bez počáteční znalosti, kolik různých vzorů vstupní soubor dat obsahuje. Výstup SOM může být také snadno zobrazen ve formě 2D mapy. [62]

Neuronová síť je implementována pomocí knihovny umělých neuronových sítí (ANNLIB, artificial neural network library, vyvinuté týmem A.Hoekstra, M.A.Kraaijveld, D.de Ridder, W.F.Schmidt, Pattern Recognition Group, Delft University of Technology). Neuronová síť byla napsána v jazyce C. PNG obrázky 2D map jsou generovány použitím GD grafické knihovny (http://www.Boutell.com). Webové rozhraní bylo napsáno v Perl-cgi. [62]

Kohonenova SOM byla trénována 5000 koly. Prvotní nastavení vah bylo vybráno náhodně. Přenastavování vah bylo omezeno jen na vítěznou jednotku a její sousedy. Toto přenastavování vah odpovídalo Gaussově funkci s maximem na vítězné jednotce. Po každém cyklu byly určeny vyhrávající jednotky pro validační set a počet jednotek odpovídajících současně na pozitivní i negativní sety dat byl vzat jako negativní míra kvality. Mapa byla uložena jen v případě, že počet takto nerozhodnutých jednotek byl menší než v předchozím kroku. Pro vizuální zhodnocení byla každá jednotka reprezentována barevným čtvercem podle příslušnosti ke třídě (protein s GPI (zeleně), protein bez GPI (modře) a proteiny bez přesného určení (červeně)) a intenzitou barvy odpovídající, jak často byla jednotka zasažena. Po tréningu byly prázdné jednotky ve 2D mapě klasifikovány dle jejich okolí. [62]

Byla vyzkoušena řada vstupních formátů. Byly počítány virtuální potenciály (VP) pro jednotlivé aminokyselinové zbytky, residua v sekvenci analogicky podle vzorce určeného pro

DNA sekvence [69]. Jako vstup byly nejprve vzaty VP v oknu o posledních 32 residuích na C-konci proteinu. Ve výsledné verzi se počítá jen s 22 nejdůležitějšími pozicemi. [62]

Dále byla určena tzv. zentriola (Z) pro každou aminokyselinu zadanou ve vstupní sekvenci. Zentriola reprezentuje průměrnou pozici aminokyseliny váženou její blízkostí k C-konci. Pro aminokyselinu, která není ve vstupní sekvenci, se Z rovná nule. [62]

Byla také zhodnocena kvalita předpokládaného ω-místa pomocí skórovací matice pro triplet ω, ω + 1 a ω + 2. Skórovací matice je založena na znalosti již potvrzených ω-míst [70, 71, 72, 73]. Nejvyšší skóre bylo přiřazeno serinu následovanému alaninem a glycinem. Skóre hydrofobicity bylo určeno dle [74]. [62]

Převedení biologické sekvence do formy čitelné vstupní vrstvou neuronové sítě bezpodmínečně vede ke ztrátě informace, neboť není příliš praktické vyjadřovat molekulární strukturu v číslech. Pro GPI-SOM byl vyvinut nový formát číselné reprezentace sekvence residuí pro identifikaci proteinů s GPI modifikací dle jejich posledních, C-koncových 32 resp. 22 residuí. [62]

V začátcích vývoje GPI-SOM byly používány kolineární verze, což znamenalo, že každý vstupní neuron přímo reprezentoval danou pozici aminokyseliny. Bylo použito 2D rozhraní s 20ti vstupními jednotkami pro každou z 32 koncových pozic. Výsledná síť dosahovala přesnosti cca 97 %, ale byla nepraktická pro velký objem dat a dlouhé výpočetní časy. [62]

Výpočty byly zrychleny reprezentováním jedné pozice jednou jednotkou místo 20ti. Pro tento účel byly residua nahrazeny jejich relativní hydrofobicitou [74]. Tento přístup zvýšil četnost špatných predikcí, hlavně ve zvýšení podílu falešně pozitivních případů. [62]

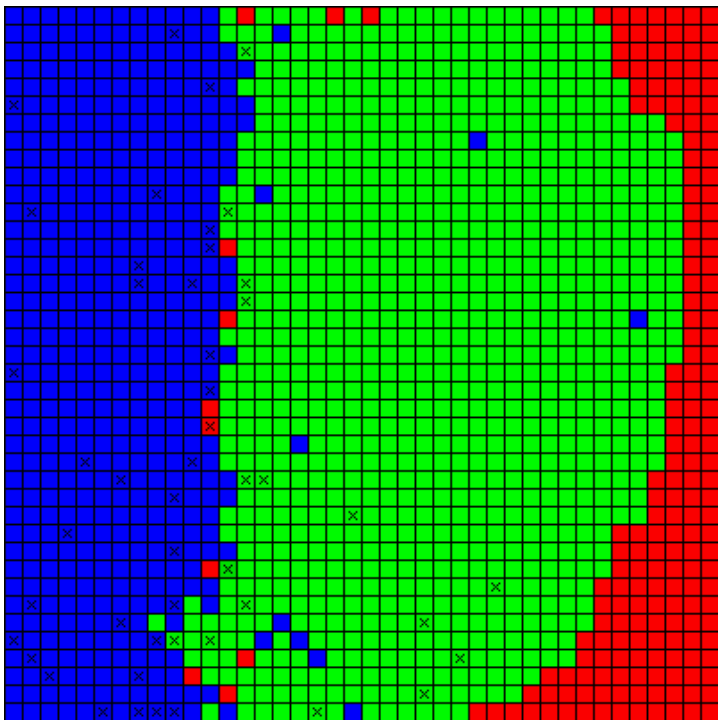
Použití VP zřetelně redukuje velikost vstupu a nutný čas na výpočty oproti kolineární reprezentaci, ale vede jen k cca 85% přesnosti předpovědí. Tato vysoká chybovost byla výrazně snížena přidáním vstupních jednotek pro relativní hydrofobicitu (H) na každé pozici. Kombinace pozičně transformovaného parametru (tj. VP) pro každou z 20ti aminokyselin s kolineární reprezentací hydrofobicity pro každou pozici z 32 a posléze 22 pozic na C-konci proteinu zřejmě dostačuje jako vstupní formát pro rozpoznání proteinů s GPI kotvou. Samotné VP a samotné H nevedly k dostatečnému výstupu. Pojmu VP je blízký pojem zentrioly (Z) (viz výše). Zentriola v kombinaci s H na každé pozici vedlo k nejnižší míře chybovosti předpovědí GPI proteinů. Používaným vstupním vektorem nejprve bylo spojení Z a H. [62]

Nejčastějším zdrojem falešně pozitivních případů byly transmembránové úseky proteinů blízko C-konce na posledních 30ti pozicích. Proto byly přidány dvě extra jednotky do vstupní vrstvy. Jedna pro kvalitu předpokládaného ω-místa a druhá pro jeho pozici. Výsledný vstupní vektor obsahoval 44 komponent daných kombinací Z + H + ω. [62]

Vstup se tak skládá ze 44 čísel: zentriola pro každou 20ti aminokyselin (20 jednotek), hydrofobicita vybraných 22 posledních pozic (22 jednotek) a kvalita a pozice nejlepší nalezené shody pro předpokládané ω-místo (2 jednotky). [62]

GPI-SOM by implementován jako Kohonenova SOM (samo-organizující se mapa) se vstupní vrstvou 44 neuronů. Při rozměrech 40×40 jednotek 2D výstupní mapy bylo dosaženo

nejnižší míry chybovosti, tj. minima neuronů excitovaných jak pozitivním tak negativním setem dat. Příklad grafického výstupu je na obrázku 2.14 níže. [62]



Obr. 2.14: Finální výstupní 2D mapa GPI-SOM [62]

Na obrázku 2.14 výše je mapa 40×40 jednotek. Tato mapa je vždy vyplněna kompletně. Mapa je rozdělena do tří polí: proteiny s GPI modifikací (v zelené oblasti), proteiny bez GPI modifikace (v modré oblasti) a neurčené proteiny (v červené oblasti). Na webovém rozhraní je možné v rámci tohoto obrázku po kliknutí na zatržený čtverec zobrazit protein dle zadaného FASTA formátu, jehož se dané určení týká. [62]

Výstupní vrstva je čtvercová mapa o 1600 jednotkách, kde se od sebe jednoznačně separují proteiny s a bez GPI modifikace. GPI-SOM vykázal senzitivitu cca 96 %. Stanovení selektivity je méně přímočaré, neboť její zhodnocení závisí hodně na vybraném negativním setu proteinů. Hlavním zdrojem falešně pozitivních nálezů byly integrální membránové proteiny s transmembránovými doménami na jejich C-konci. Bylo ukázáno experimentálně, že jednobodová mutace může stačit na přeměnu signálu pro přiřazení GPI kotvy v transmembránovou doménu. Špatné přiřazení transmembránových (TM) proteinů k proteinům s GPI kotvou se může minimalizovat vyřazením sekvencí s vícenásobnými transmembránovými doménami. Při použití GPI-SOM toto prováděno není, neboť i v případě přítomnosti TM domény nelze vyloučit přítomnost správného signálu pro připojení GPI kotvy. [62]

V souhrnu lze říci, že GPI-SOM v době svého vzniku byl novým přístupem pro výpočetní predikci signálů v proteinových sekvencích pro připojení GPI kotvy a byl vítaný jako jeden z dalších nástrojů (vedle Big-PI a DGPI) pro GPI predikci. [62]

3 Cíle práce

Cílem této práce je vytvoření nástroje, jenž usnadní vyhledávání genů genové rodiny *bst2* a příbuzných protivirových genů, tzv. *tetherinů* v genomech obratlovců včetně přehledného grafického výstupu ve formě HTML souboru.

Tento nástroj by měl být využitelný i pro vyhledávání kandidátních genů příslušné genové rodiny v genomech řady dalších druhů obratlovců, kde příslušné geny dosud nalezeny nebyly, ale je zde důvodný předpoklad jejich existence.

Práce využije jeden z přístupů k hledání kandidátních genů v genomech, tj. vyhledávání sekvenčních motivů a charakteristik, jež jsou běžně obsaženy ve všech či většině členů dané genové rodiny.

Cílem této práce je postup významně zautomatizovat současně s poskytnutím vhodného grafického výstupu. Vstupem tohoto nástroje bude pouze sekvence DNA, ve formě TXT souboru, a výstupem bude HTML stránka se zobrazením všech důležitých údajů ohledně přítomnosti a polohy hledaných sekundárních proteinových struktur v rámci zadaného i komplementárního vlákna.

Do značné míry bude v algoritmu napodoben přirozený průběh toku informace z DNA do proteinu odehrávající se běžně v přírodě. Na zadaném i komplementárním vláknu DNA budou vyhledány otevřené čtecí rámce (ORFs, open reading frames - obvykle dále tzv. „ORFy“). ORFy budou hledány dle své nejběžnější definice, tedy jako úsek vlákna DNA mezi dvěma STOP kodony, jehož délka v počtu párů bazí je dělitelná třemi beze zbytku. [78] ORFy, které budou splňovat dále podmínku uživatelem zadané minimální délky (typicky 100 či 150 párů bazí), budou přeloženy do sekvence aminokyselinových zbytků v peptidech/proteinech. Na těchto sekvencích aminokyselin budou dále pomocí standardních algoritmů vyhledány sekundární proteinové struktury typické pro gen *bst2* a příbuzné protivirové geny. Celkem budou hledány tyto tři sekundární struktury: transmembránové domény, coiled-coil domény a tzv. GPI (glycophosphatidylinositol) modifikace/kotvy.

4 Metody

Pro tuto diplomovou práci byl sepsán skript, jenž byl spouštěn na stolním počítači. Použitý stolní počítač byl notebook AMD Ryzen 5 4600H with Radeon Graphics × 12 s 15,1 GiB RAM.

Ve stolním počítači byla použita tato distribuce operačního systému Linux: Release Linux Mint 20 Ulyana 64-bit, Kernel Linux 5.4.0-66-generic x86_64, MATE 1.24.0.

Při zpracování úseku lidské DNA o délce cca 16,6 kbp (je vždy zpracováno zadané i komplementární vlákno) a při použité minimální délce otevřeného čtecího rámce (ORFu) 150 bp trval běh skriptu na uvedeném stolním počítači přibližně 3 minuty a 45 sekund. Byla zvolena varianta ošetření ambiguitního kódu s číslem 3, tj. ORF s byť i jen jedním neznámý znakem byl zahozen.

Jako programovací jazyk byl použit Python 3.8.5, samotný skript byl psán v prostředí IDLE 3.8.5 (Python‘s Integrated Development and Learning Environment). Výsledná délka skriptu je přibližně 11600 řádků kódu (se zakomentovanými částmi kódu, s poznámkami a s prázdnými řádky pro přehlednost kódu).

4.1 Popis skriptu

Na samotném začátku kódu je spuštěno počítání času běhu skriptu. Ukončeno je až na předposledním řádku kódu. Dále následuje promazání úložišť na pevném disku, tj. jsou smazány všechny png a svg obrázky, faa, fasta, txt a další soubory z předchozího běhu programu. Poté je načtena sekvence vyšetřovaného úseku DNA ve formě jen samotné sekvence v txt souboru. Z načtené sekvence jsou odstraněny případné znaky konce řádku („\n“). Je dotvořeno komplementární vlákno k zadanému vláknu DNA. Je ošetřena délka zadané sekvence, zda není pod nebo nad délkovým limitem v bp. Standardně je ve skriptu použitá minimální délka vstupní sekvence 320 bp a maximální délka vstupní sekvence 32000 bp. Dále je zadaná minimální délka následně vyhledávaných otevřených čtecích rámců (ORFů) v bp. Např. 150 bp. Zadaná minimální délka ORFu je převedena do absolutní hodnoty a zaokrouhlená na celé číslo. (To je pro případ zadání záporné hodnoty a/nebo necelého čísla.) Skript na začátku běhu vypíše, jaká je ve skutečnosti použitá minimální délka ORFu (po zaokrouhlení a vytvoření absolutní hodnoty). Zadaná délka ORFu je porovnána s limitem pro minimální a maximální délku ORFu. Obvykle je minimum pro délku ORFu 30 bp a jako možné maximum je použita délka zadané sekvence DNA (může se stát, že jako vstupní sekvence byl použit jeden delší ORF). Je použita tato definice ORFu: ORF je definován jako úsek DNA ohraničený dvěma STOP kodony a jeho délka v bp je dělitelná třemi beze zbytku [78].

Jsou vyhledány STOP kodony (místo jejich začátku) v zadaném (a posléze i v komplementárním) vlákně využitím regulární exprese. Jsou vyhledávány současně všechny tři možné STOP kodony v DNA vlákně: TAA, TAG, TGA.

Dále uživatel volí jednu ze tří možností ošetření ambiguitního kódu. Jsou očíslovány 1, 2 a 3. První varianta ve skriptu provádí nahrazení místa s neznámým znakem náhodně zvolenou aminokyselinou (do výběru aminokyseliny nejsou použity STOP kodony).

Druhá varianta provádí vystřižení a zahození všech neznámých znaků (jeden neznámý znak může zastupovat i více pozic ve vlákně, čili i tak předem dochází již k posunu čtecího rámce).

Poslední, třetí varianta vyřazuje z dalšího zpracování všechny ORFy s byť i jen jedním neznámým znakem.

Další postup je stejný po zadané i pro komplementární vlákno. Vesměs bude popisován postup jen pro zadané vlákno.

Nalezené pozice STOP kodonů (viz výše) jsou roztříděny do tří čtecích rámců. Z části se liší zacházení s nalezenými ORFy dle toho, jaké je vybráno ošetření ambiguitního kódu. Ve všech třech případech je nejprve pro každý čtecí rámec ošetřeno, zda se v něm nacházejí minimálně dva nalezené STOP kodony. V daném čtecím rámci lze následně vyšetřovat, zda se dle definice jedná o ORF, jen v případě přítomnosti alespoň dvou STOP kodonů. Tímto roztříděním do čtecích rámců je současně zajištěno, že úseky mezi hledanými STOP kodony jsou dělitelné třemi beze zbytku.

V případě první a druhé varianty ošetření ambiguitního kódu je zde postup stejný. Zpracovává se soubor STOP kodonů jen v daném čtecím rámci. Čili jsou prováděna celkem tři hledání ORFů – zvlášť pro každý čtecí rámec. V případě, že vzdálenost mezi pozicemi STOP kodonů je menší nežli nastavená minimální délka ORFů je takovýto „ORF“ přeskočen. V případě, že vzdálenost je rovna nebo větší nežli stanovená minimální délka ORFu, jsou použity pozice těchto dvou sousedních STOP kodonů pro vyjmutí sekvence samotného ORFu ze zadaného vlákna a je též zaznamenána pozice začátku a konce ORFu dle pozice obou sousedních STOP kodonů.

Druhá varianta ze tří ošetření ambiguitního kódu spočívá ve zmíněném „vystřižení“ neznámých znaků. Poté, co je načteno zadané vlákno, jsou neznámé znaky v něm nahrazeny „prázdným“ znakem, čili daný neznámý znak nebo sekvence neznámých znaků je vystřižena a sousední úseky sekvence DNA jsou na sebe napojeny. V případě tohoto ošetření ambiguitního kódu je počítána délka zadaného vlákna až po vyřazení neznámých znaků a porovnána s výše uvedeným limitem pro minimální a maximální povolenou délku zadané sekvence. Tato sekvence po vyřazení neznámých znaků je stejně dlouhá i v případě komplementárního vlákna, čili porovnání s minimální a maximální možnou délkou sekvence je provedeno jen na tomto jednom místě se zadaným vláknem. V případě zadaného i komplementárního vlákna je vyhledání STOP kodonů užitím regulární exprese provedeno až po vyřazení neznámých znaků.

V případě třetí možnosti ošetření ambiguitního kódu (vyřazení ORFu byť i jen s jedním neznámým znakem) jsou nejprve všechny neznámé znaky převedeny na jeden neznámý znak („W“). Poté jsou vyhledávány ORFy stejně jako v předchozích dvou případech, ale následuje ošetření, zda ORF obsahuje neznámý znak. Pokud jej neobsahuje, je s ním zacházeno dále stejně jako v předchozích případech. Pokud obsahuje alespoň jeden neznámý znak, je zahozen do „odpadu“ – je uchován v proměnné pro tyto „nevhodné“ ORFy a stejně je i tak pro tento ORF zaznamenám jeho počátek a konec do další proměnné. Tyto proměnné však nejsou dále ve skriptu používány.

Nalezené ORFy pro každé ošetření ambiguitního kódu a pro každý ze tří možností čtecích rámců jsou následně shrnuty do jedné proměnné a stejně tak nalezené jejich počátky a konce (dále často tzv. „souřadnice“) jsou shrnuty také do své vlastní proměnné.

Následuje překlad získaných ORFů do aminokyselinových sekvencí polypeptidů, proteinů. Byly získány proměnné pro standardní genetický kód a též osmnáct alternativních genetických kódů ze stránky [75]. Vytvořený skript je vhodný jen pro standardní genetický kód a dva alternativní genetické kódy (kód archeí a rostlinných plastidů a kvasinkový alternativní jaderný kód). Skript vyžaduje, aby v rámci daného genetického kódu byl stejný počet STOP kodonů a stejná pozice STOP kodonů v rámci kódu jako u standardního kódu. Toto je splněno jen pro dva alternativní genetické kódy. Tyto rozdílné kódy se však v kódování aminokyselin mohou libovolně lišit.

Nejprve je ověřeno, zda seznam ORFů v zadaném (resp. komplementárním) vláknu obsahuje alespoň jeden ORF v alespoň jednom čtecím rámci. Pokud jej neobsahuje, je v případě zadaného vlákna zobrazen nápis „Zadny ORF a peptid v zadanem vlaknu.“.

Analogicky u komplementárního vlákna. V případě přítomnosti alespoň jednoho ORFu je tento přeložen do aminokyselinové sekvence.

Reprezentace genetického kódu zde využívá tři proměnné, seznamy typické pro Python 3. Tyto seznamy jsou ve skriptu označeny „Base1“, „Base2“ a „Base3“.V každém seznamu je jedna položka s řetězcem kombinací čtyř písmen jakéhokoliv genetického kódu v DNA (A, T, G a C). Každý z těchto řetězců obsahuje 20 znaků. Při myšleném vytvoření sloupce v rámci těchto tří po sebe seřazených řetězců získáme část kódu reprezentující vždy jen jednu danou aminokyselinu pro daný kód (např. standardní kód) pokud jej seřadíme pod tyto tři řetězce s písmeny genetického kódu. Skript v této části čte zadanou sekvenci DNA po tripletech, kodonech. Pro první místo tripletu označí číslem, jeho výskyt v prvním seznamu (Base1), analogicky pro druhé písmeno tripletu ve druhém seznamu (Base2) a analogicky pro třetí písmeno v tripletu (Base3). Vzniknou tak 3 seznamy čísel. Tyto seznamy jsou převedeny na množiny a je proveden postupně průnik všech těchto tří množin. Využívá se zde vlastnosti reálného světa přírody, že - pokud nebyl přítomen ani na jednom místě tripletu neznámý znak - pak průnik těchto tří množin dá jen jedno číslo označující pozici, z níž má být vybrána daná aminokyselina z daného kódu.

Pokud se v sekvenci nacházel neznámý znak v daném místě tripletu, vyjde pro daný neznámý znak prázdný seznam a prázdná příslušná množina. Tato možnost může nastat jen pro první variantu ošetření ambiguitního kódu. V případě druhé možnosti ošetření ambiguitního kódu jsou neznámé znaky již před touto částí kódu vystřiženy, v případě třetí možnosti ošetření ambiguitního kódu jsou již před touto částí kódu vyřazeny ORFy s neznámými znaky. První varianta ošetření ambiguitního kódu spočívá v nahrazení kodonu byť jen s jedním neznámým znakem náhodně vybranou aminokyselinou. (STOP kodony jsou z možnosti nahrazení neznámého znaku vyloučeny.)

Jsou takto prohledávány jen ORFy, které byly vybrány jako ORFy v předchozí části skriptu. V této části jsou vytvořeny jako jeden dlouhý řetězec i s udanými místy STOP kodonů (označené „*“). V další části kódu jsou od sebe tyto přeložené ORFy odděleny právě pomocí znaku pro STOP kodon.

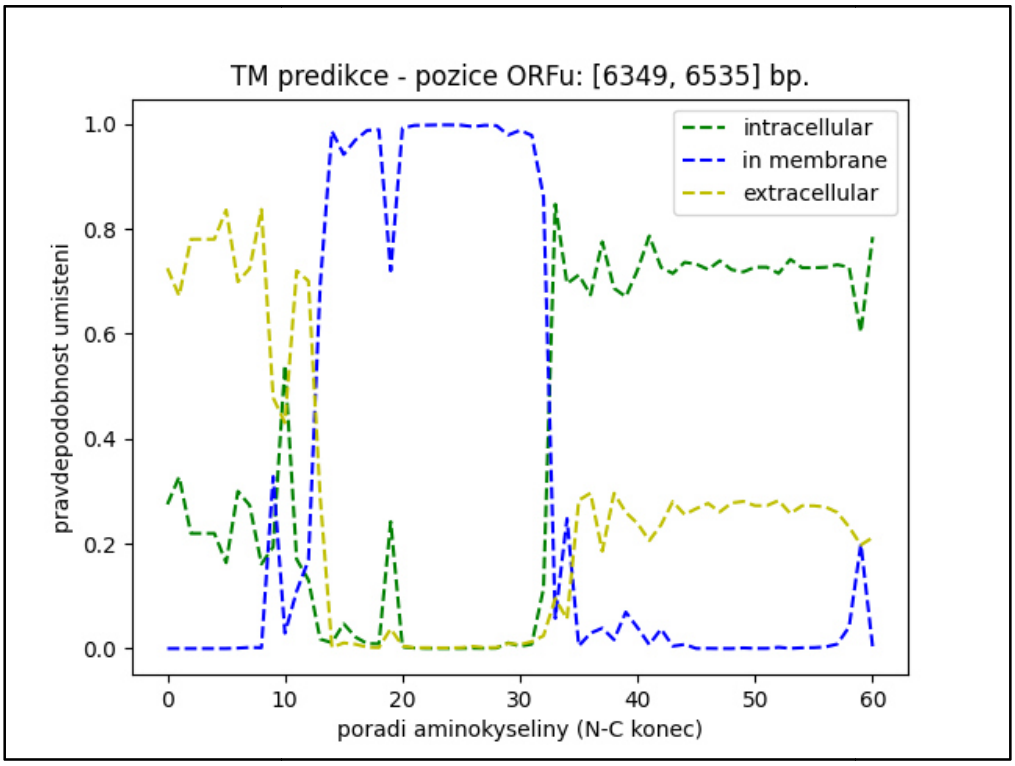
V další části kódu jsou ORFy přeložené do sledu aminokyselinových zbytků, (poly)peptidů, ukládány ve formě faa souboru. Skript zpracovává postupně ORFy jak za sebou v sekvenci DNA následují, dle nich vytváří sekvence polypeptidů. K seznamu ORFů existuje seznam souřadnic začátku a konce každého ORFu. Seznam souřadnic ORFů je seznam seznamů. Jedna položka v tomto seznamu obsahuje seznam o dvou položkách (začátek a konec ORFu). Tyto souřadnice jsou přiřazeny každému polypeptidu ukládanému do faa souboru.

Obdobně jsou získány sekvence polypeptidů pro zmiňované dva alternativní genetické kódy (kód archeí a rostlinných plastidů a kvasinkový alternativní jaderný kód). Tyto sekvence jsou také uloženy ve formě faa souborů. Dále ve skriptu tyto faa soubory použity nejsou. Polypeptidové sekvence pro standardní kód jsou uloženy také ve formátu txt a fasta. Tyto soubory neobsahují hlavičky pro jednotlivé sekvence a také tyto soubory nejsou dále ve skriptu použity.

Pro vyhledávání ORFů s transmembránovou (TM) sekundární strukturou a vykreslení pozice TM struktury v rámci ORFu je použita implementace skrytého Markovova modelu (HMM, hidden Markov model) popsaná v [18] a v kapitolách 2.2.5.7 tmhmm.py 1.3.1 a 2.2.5.1 TMHMM Server v. 2.0. Konkrétně se jedná o implementaci pomocí nástroje tmhmm.py 1.3.1 popisovaného v kapitole 2.2.5.7. Tento nástroj je ke stažení na: https://pypi.org/project/tmhmm.py/. Nástroj byl vybrán díky možnosti nainstalovat jej jako balíček přímo v rámci Pythonu 3.8.5. Tato verze tmhmm.py (1.3.1) byla zveřejněna 13. března 2020. Existuje již novější verze ve formě dalšího programu pyTMHMM 1.3.2 (viz kapitola 2.2.5.8 pyTMHMM 1.3.2). Ke stažení je k dispozici na:

https://pypi.org/project/pyTMHMM/. Tento programový balíček pro Python 3 byl vydán 12. ledna 2021. Vyzkoušen a použit pro skript nebyl z časových důvodů.

Příslušné ORFy s detekovanou TM sekundární strukturou jsou ve skriptu spočteny. Jsou též zachyceny a zachovány souřadnice daného ORFu s TM strukturou. Pro daný ORF je vždy vytvořen obrázek zachycující předpovězenou situaci ve formě obrázku se zachycením pravděpodobnosti pro daný aminokyselinový zbytek, zda se nachází intracelulárně, v transmembránovém úseku nebo extracelulárně. Viz obrázek 4.1 níže. Na obrázku 4.1 je zachycena předpovězená pravděpodobnost HMM modelem pro intracelulární, transmembránovou (TM) a extracelulární lokalizaci daných aminokyselinových zbytků. V rámci vytvořené HTML stránky (viz níže) jsou poskytnuty k obrázku vždy i některé další informace – zda se jedná o zadané nebo o komplementární vlákno, o jaký čtecí rámec se jedná, jaká je pozice daného ORFu v rámci vlákna a též je zobrazena délka daného vlákna. V rámci obrázku je poskytnuta znovu informace o umístění ORFu (v hranatých závorkách – začátek a konec). V tomto příkladu obrázku pro ORF se jedná o ORF na pozici 6349 až 6535 (v bp) v rámci zadaného vlákna. Na vodorovné ose je pořadí aminokyseliny po přeložení do aminokyselinové sekvence od N po C-konec. Na svislé ose pravděpodobnost daného umístění aminokyselinového zbytku od 0 (0 %) po 1 (100 %). Modře je zobrazen průběh pravděpodobnosti pro TM část, zeleně pro intracelulární část a hnědě pro extracelulární část.



Obr. 4.1: Zachycení pravděpodobnosti TM struktury v rámci ORFu

Tyto obrázky jsou na konci skriptu použity pro tvorbu HTML stránky.

Hned za částí skriptu pro vytvoření tohoto obrázku je část skriptu vytvářející přehledový obrázek s vyznačeným umístění daného ORFu v rámci vlákna DNA. Viz obrázek 4.2 níže. Zde v příkladu na pozici 6349 až 6535 bp z celkové délky vlákna 16570 bp. I tento obrázek je poté použit při generování HTML stránky na konci skriptu.

Obr. 4.2: Grafické vyznačení pozice daného ORFu

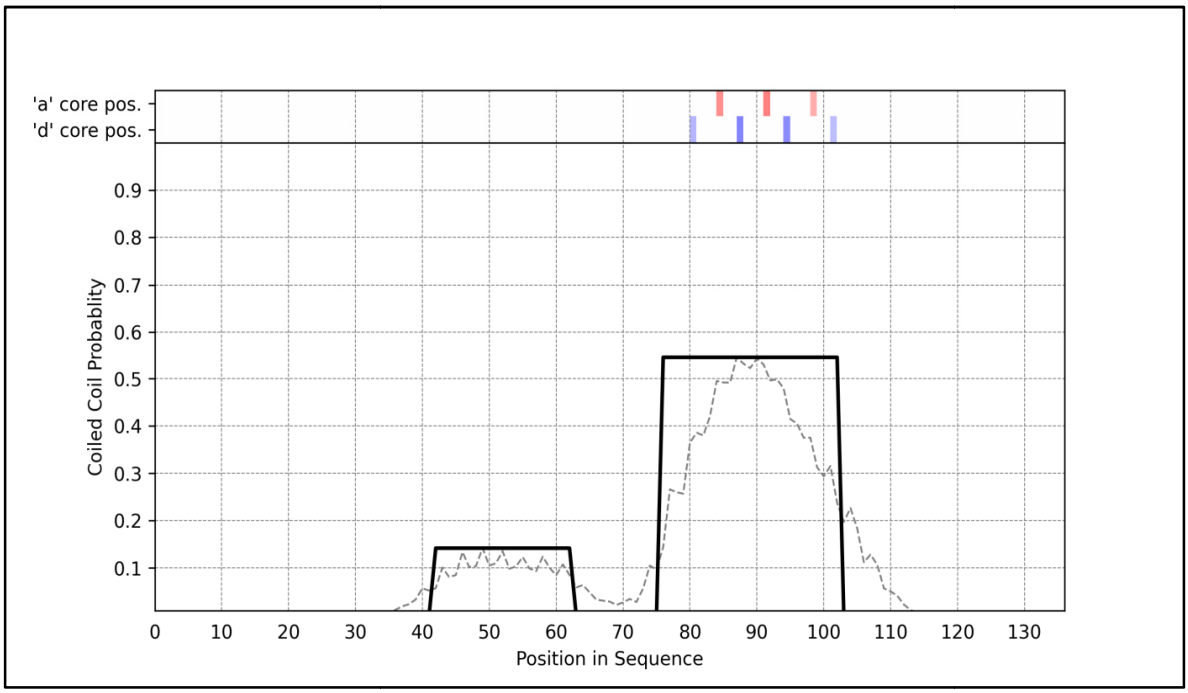
Na obrázku 4.2 výše je modrou čarou vyznačeno celé vlákno zadané DNA. Silnější červenou čarou je zachycena pozice daného ORFu v rámci vlákna.

Tyto obrázky jsou vytvořeny pro TM, CC i GPI sekundární strukturu zvlášť pro zadané a zvlášť pro komplementární vlákno. Může být vytvořen tento obrázek pro stejný ORF pro každpu ze tří struktur. V tom případě je několikrát přepsán na pevném disku tím samým obrázkem a poté je tento jeden obrázek použit v HTML stránce jen jedenkrát pro zobrazenou kombinaci obrázků s TM a/nebo s CC a/nebo s GPI pro daný ORF v daném vlákně.

Stejný postup platí i pro komplementární vlákno.

Pro vyhledání ORFů s coiled-coil (CC) sekundární strukturou a vykreslení průběhu pravděpodobnosti pro CC strukturu v daném ORFu byl použit program DeepCoil 2.0.1 popsaný v [53] a v kapitole 2.2.7.4 DeepCoil 2.0.1. Ke stažení je tento nástroj na: https://pypi.org/project/deepcoil/. DeepCoil 2.0.1 byl vybrán (dále obvykle jen DeepCoil), protože jej lze snadno spustit a využít v rámci samotného skriptu v Pythonu 3. V případě nástroje DeepCoil se jedná v době psaní této práce o nejnovější verzi. Verze byla vydána 30. listopadu 2020.

Pro použití v rámci skriptu je nejprve vybrán práh pro zachycení nalezené CC struktury až od určité její pravděpodobnosti. Lze samostatně zvolit práh pro zadané a pro komplementární vlákno. Práh je posuzován dle nejvyšší míry pravděpodobnosti předpovězené CC struktury v rámci daného ORFu a je brán včetně udané hodnoty prahu. Práh je zadán ve formě desetinného čísla. Např. hodnota 0.10 znamená, že maximální předpovězená pravděpodobnost CC struktury v rámci ORFu s CC strukturou musela být minimálně 10 %. Dále je ošetřena možnost, že v rámci daného vlákna se nenachází žádný ORF. Poté je využit samotný DeepCoil. Je mu poskytnut faa soubor s dříve získanými sekvencemi aminokyselinových zbytků pro standardní kód. (Soubory faa vytvořené pro jiné kódy nejsou ve skriptu po jejich vytvoření nijak použity.) Jsou ukládány do samostatné složky všechny vygenerované obrázky, tj. i ty bez předpovězené CC struktury. Do jiných složek (zvlášť je složka pro zadané a pro komplementární vlákno) jsou ukládány obrázky s detekovanými CC strukturami od určité dosažené pravděpodobnosti rovnající se minimálně zvolenému prahu. Z této složky jsou obrázky poté použity v generované HTML stránce na konci skriptu. Obrázky jsou roztříděny pro zadané a komplementární vlákno do samostatných složek a v rámci nich rozlišeny jen dle „souřadnice“ počátku ORFu v rámci daného vlákna. Příklad grafického zachycení výskytu a umístění CC struktury v rámci ORFu přeloženého do sekvence aminokyselin je na obrázku 4.3 níže.



Obr. 4.3: Získaný obrázek zachycující pravděpodobnost a umístění CC struktur

Na obrázku 4.3 výše jsou zachyceny dvě předpovězené CC struktury. Na vodorovné ose je zachycena sekvence aminokyselin pro daný ORF od N po C-konec. Svislá osa odpovídá pravděpodobnosti předpovězené CC struktury. Pozice ORFu v daném vlákně není zachycena. Program generující tyto obrázky tuto možnost neposkytuje, ale v rámci HTML stránky je pozice daného ORFu s touto strukturou vyobrazena.

V následující části kódu jsou vytvářeny obrázky pro souhrn a zobrazení umístění jednotlivých ORFů v zadaném a též ve stejné formě v komplementárním vlákně. Pro dané vlákno jsou vytvořeny tři obrázky. Po jednom obrázku pro daný čtecí rámec. Na svislé ose každého z obrázků je souřadnice počátku ORFu v bp, na vodorovné ose je souřadnice konce ORFu v bp. Jednotlivé ORFy jsou v těchto grafech zachyceny jako drobné kroužky (Jsou v každém obrázku v jiné barvě: v červené pro 1. čtecí rámec, v modré pro druhý čtecí rámec a zelené pro 3. čtecí rámec.) Přičemž ORFy jsou takto uspořádány v okolí diagonály obrázku. Vygenerované tři obrázky jsou poté v HTML stránce uspořádány horizontálně vedle sebe zleva doprava od prvního po třetí čtecí rámec. Tyto obrázky nejsou příliš důležité a slouží jen pro vykreslení situace rozložení ORFů v daném vlákně.

Pod touto trojicí obrázků je další trojice obrázků. Tato trojice je také uspořádána horizontálně. V levém obrázku jsou zobrazeny ORFy s TM strukturou, v prostředním s CC strukturou a v pravém obrázku s GPI strukturou. Ve skriptu jsou nejprve tvořeny obrázky pro CC, poté pro TM a dále pro GPI sekundární strukturu.

V případě obrázku pro předpovězené TM sekundární struktury souřadnice na vodorovné ose odpovídá přesně umístění počátku TM struktury. Vykreslené úsečky tedy svým umístěním na vodorovné ose vyznačují začátky TM struktur. Ke každé úsečce v tomto grafu je přidán při jejím horním konci v rámci grafu popis. Horizontálně orientovaným číslem je počet aminokyselin v TM struktuře pro daný ORF. Další údaje jsou pro úsporu místa v grafu otočeny o 90°. Jedná se o udání místa počátku ORFu (pro vyhledání ORFu v následujících tabulkách – viz níže) a poté o udání místa počátku TM struktury.

V případě grafu pro CC sekundární struktury umístění dolních konců úseček na vodorovné ose odpovídá pozici pro maximum pravděpodobnosti předpovězené CC struktury.

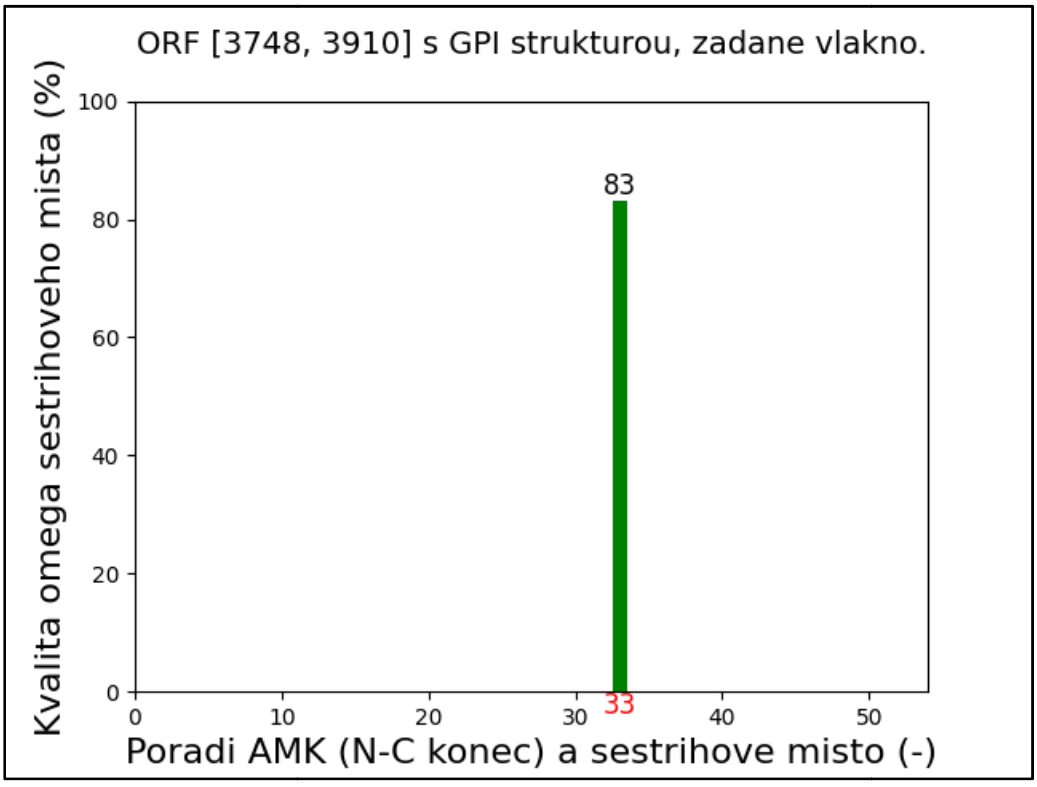
Při horním konci úsečky v grafu je přidán popis: pravděpodobnost CC struktury zaokrouhlená na desetinu procenta, pozice ORFu (pro vyhledání v tabulkách uvedených na HTML dále na stránce – viz níže) a pozice maxima pravděpodobnosti pro předpovězenou CC sekundární strukturu.

V případě grafu pro GPI modifikace umístění pat úseček na vodorovné ose odpovídá pozici sestřihového ω-místa pro GPI modifikaci. Při horním konci úsečky v rámci grafu jsou uvedeny: kvalita sestřihového místa v procentech, pozice (počátek v rámci daného vlákna) příslušného ORFu a pozice sestřihového ω-místa v rámci vlákna.

Svislé osy se tedy liší pro každý z těchto tří obrázků. V případě TM struktur je zajištěno díky přírodním zákonitostem, že se daná TM struktura v dané sekvenci aminokyselin vyskytuje nebo nevyskytuje. Pro svislou osu zde je vybrán počet aminokyselin pro danou TM strukturu v daném ORFu. V obrázku pro CC struktury je na svislé ose pravděpodobnost dané CC struktury. U obrázku pro GPI strukturu je dle svislé osy určená kvalita ω-místa.

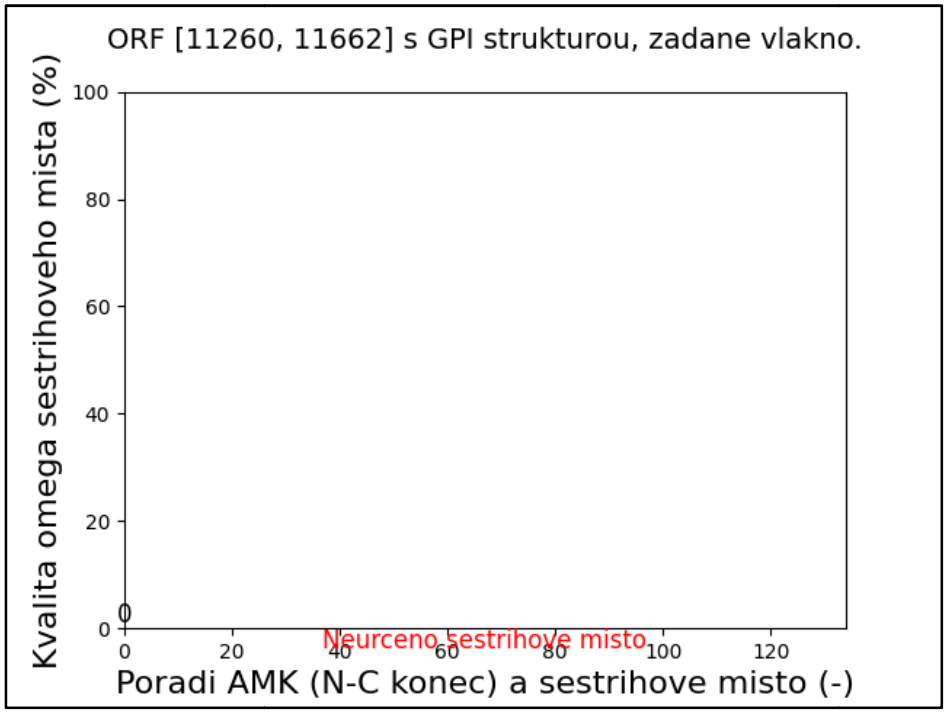
Pro určení ORFů s GPI strukturou a vytvoření souborů s informacemi nutnými pro vytvoření obrázku pro GPI strukturu byl použit program GPI-SOM popsaný v kapitole 2.2.9.3 GPI-SOM a v [62]. Tento program byl vybrán, neboť jako jediný byl k dispozici ke stažení a bylo možné jej nainstalovat a dále používat bez nutnosti internetového připojení. V případě jiných nástrojů dostupných jen on-line by mohl též nastat problém se zadáním úkolu a následně s převzetím výsledků z těchto programů pro další zpracování ve skriptu v Pythonu 3. Program je volně ke stažení na: http://gpi.unibe.ch.

Program kromě určení ORFu (dle jeho pozice) poskytuje informace pro vykreslení obrázku se zachycením situace pro daný ORF. Příklady jsou zobrazeny na obrázcích 4.4 a 4.5 níže.



Obr. 4.4: Vykreslení pozice a kvality ω-místa pro ORF s GPI strukturou

Na obrázku 4.4 výše je zachycen příklad ORFu v zadaném vláknu s GPI strukturou. Souřadnice ORFu jsou vždy v těchto obrázcích vykresleny (nahoře v hranatých závorkách). V tomto případě je pozice ORFu v rozsahu 3748 až 3910 (v bp) v zadaném vlákně. Na vodorovné ose je pořadí aminokyselinového zbytku v rámci ORFu od N po C-konec. Červeným číslem na vodorovné ose je zobrazeno ω-místo. Na svislé ose je kvalita ω-místa v procentech.



Obr. 4.5: Vykreslení ORFu s předpovězenou GPI strukturou bez určení ω-místa

Obrázek 4.5 výše je příkladem vykreslení grafu pro ORF, u nějž byla předpovězena GPI struktura, ale bez udání pozice a kvality ω-místa. Na vodorovné ose je nahrazeno červené číslo s udáním sestřihového ω-místa červeným nápisem „Neurceno sestrihove místo“ a není vykreslena úsečka odpovídající míře kvality ω-místa.

Ve skriptu jsou mj. určeny pozice ORFů (jen počátky ORFů) s TM, CC a GPI strukturami. Zvlášť pro zadané a zvlášť komplementární vlákno. Tyto seznamy jsou použity pro spočtení a určení ORFů dle možných kombinací, které mohou nastat. Seznamy jsou převedeny na množiny a jsou s nimi provedeny množinové operace. Je určováno těchto sedm možných kombinací: ORF jen s TM strukturou, ORF jen s CC strukturou, ORF jen s GPI strukturou, ORF jen s TM a s CC strukturou, ORF jen s TM a s GPI strukturou, ORF jen s CC a s GPI strukturou, ORF se všemi strukturami tj. s TM, CC a GPI strukturou. Pro každou z těchto kombinací je určena celá souřadnice ORFu (začátek a konec) a je spočtena velikost každé z těchto množin tj. kolik je ORFů s daným omezením zvlášť v zadaném a zvlášť v komplementárním vlákně. Informace o velikosti těchto množin je poté uvedena v HTML stránce.

Následuje dynamické vygenerování HTML stránky dle situace, která v zadaném a komplementárním vlákně nastala. Pro dynamické vygenerování HTML stránky byl použit program Airium 0.2.1. Je volně ke stažení je na: https://pypi.org/project/airium/. Tato verze byla vydána 7. prosince 2020. K dispozici je novější verze 0.2.2 vydaná 31. ledna 2021. Ke stažení je na stejném místě jako starší verze. Novější verze nebyla nainstalována a použita. Byl zde předpoklad, že při použití vyšší verze nedojde k výraznému zlepšení tvorby HTML stránky.

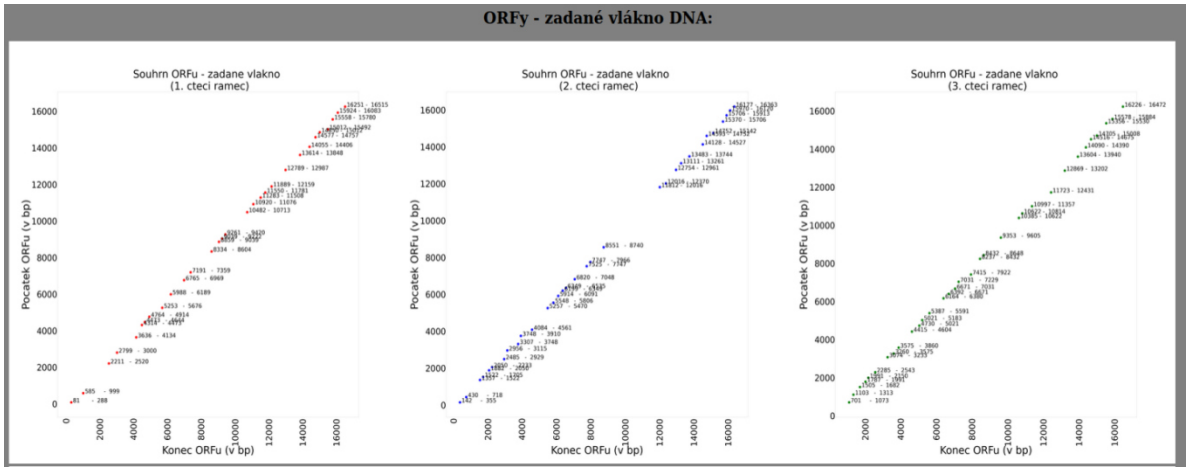
4.2 Popis vytvořené HTML stránky

Pro titulek HTML stránky (dále obvykle jen „stránka“) byl zvolen nápis „TM, CC a GPI predikce“. Na začátku stránky je nejprve jen textový výčet nejdůležitějších údajů zjištěných pro zadané a komplementární vlákno.

Je zde nejprve poznámka, že skript je použitelný jen pro standardní kód a dále kód archeí a rostlinných plastidů a kvasinkový alternativní jaderný kód. Dále jsou zde uvedeny:

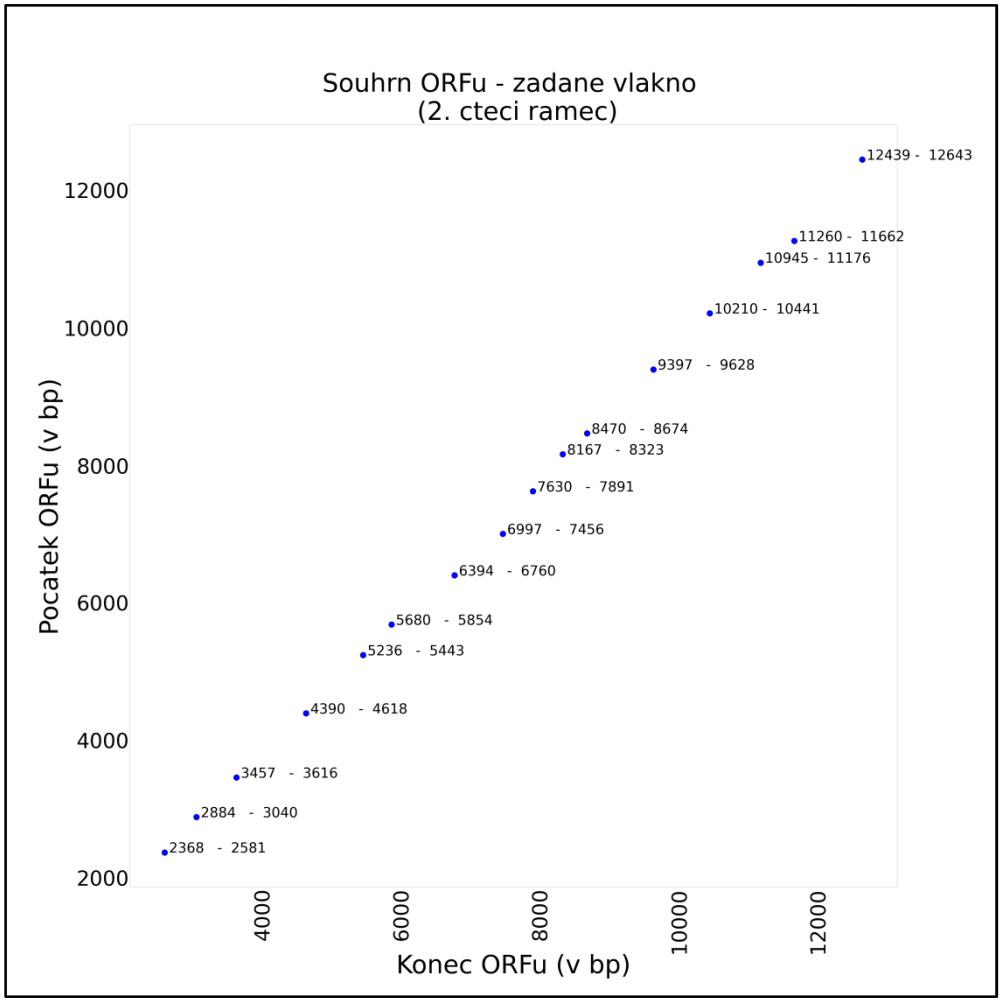
- dolní limit délky vstupní sekvence (v bp),
- horní limit délky vstupní sekvence (v bp),
- délka zadané NK (nukleové kyseliny),
- minimální délka ORFu (v bp),
- počet ORFů v zadaném vláknu,
- počet ORFů v komplementárním vláknu,
- celkový počet ORFů v obou vláknech,
- počet ORFů s TM sekundární strukturou (zadané vlákno),
- počet ORFů s TM sekundární strukturou (komplementární vlákno),
- počet ORFů s CC sekundární strukturou (zadané vlákno),
- počet ORFů s CC sekundární strukturou (komplementární vlákno),
- práh pro detekci CC sekundární struktury - zadané vlákno (v %),
- práh pro detekci CC sekundární struktury - komplementární vlákno (v %),
- počet ORFů s GPI sekundární strukturou (zadané vlákno),
- počet ORFů s GPI sekundární strukturou (komplementární vlákno),
- počet ORFů jen s TM sekundární strukturou (zadané vlákno),
- počet ORFů jen s TM sekundární strukturou (komplementární vlákno),
- počet ORFů jen s CC sekundární strukturou (zadané vlákno),
- počet ORFů jen s CC sekundární strukturou (komplementární vlákno),
- počet ORFů jen s GPI sekundární strukturou (zadané vlákno),
- počet ORFů jen s GPI sekundární strukturou (komplementární vlákno),
- počet ORFů jen s TM a CC sekundární strukturou (zadané vlákno),
- počet ORFů jen s TM a CC sekundární strukturou (komplementární vlákno),
- počet ORFů jen s TM a GPI sekundární strukturou (zadané vlákno),
- počet ORFů jen s TM a GPI sekundární strukturou (komplementární vlákno),
- počet ORFů jen s CC a GPI sekundární strukturou (zadané vlákno),
- počet ORFů jen s CC a GPI sekundární strukturou (komplementární vlákno),
- počet ORFů s TM, CC a GPI sekundární strukturou současně (zadané vlákno),
- počet ORFů s TM, CC a GPI sekundární strukturou současně (komplementární vlákno).

Pod tímto výčtem následuje na stránce trojice obrázků s již zmiňovaným souhrnem ORFů v zadaném vlákně. Zobrazení ORFů je zde vlášť pro 1., 2. a 3. čtecí rámec. Viz ilustrativní obrázek 4.6 níže.



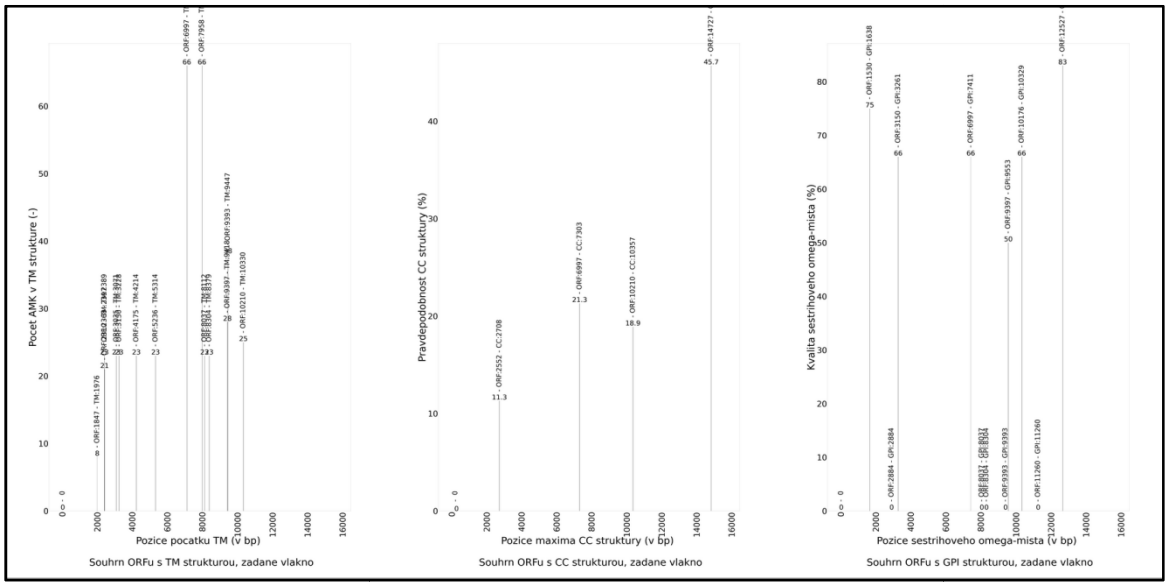
Obr. 4.6: Vzhled trojice obrázků se souhrnem ORFů v daném vlákně

Na obrázku 4.7 níže je uveden příklad obrázku se souhrnem ORFů pro zadané vlákno, druhý čtecí rámec. Jedná se o jeden ze tří grafů zobrazených souhrnně na obrázku 4.6 výše.



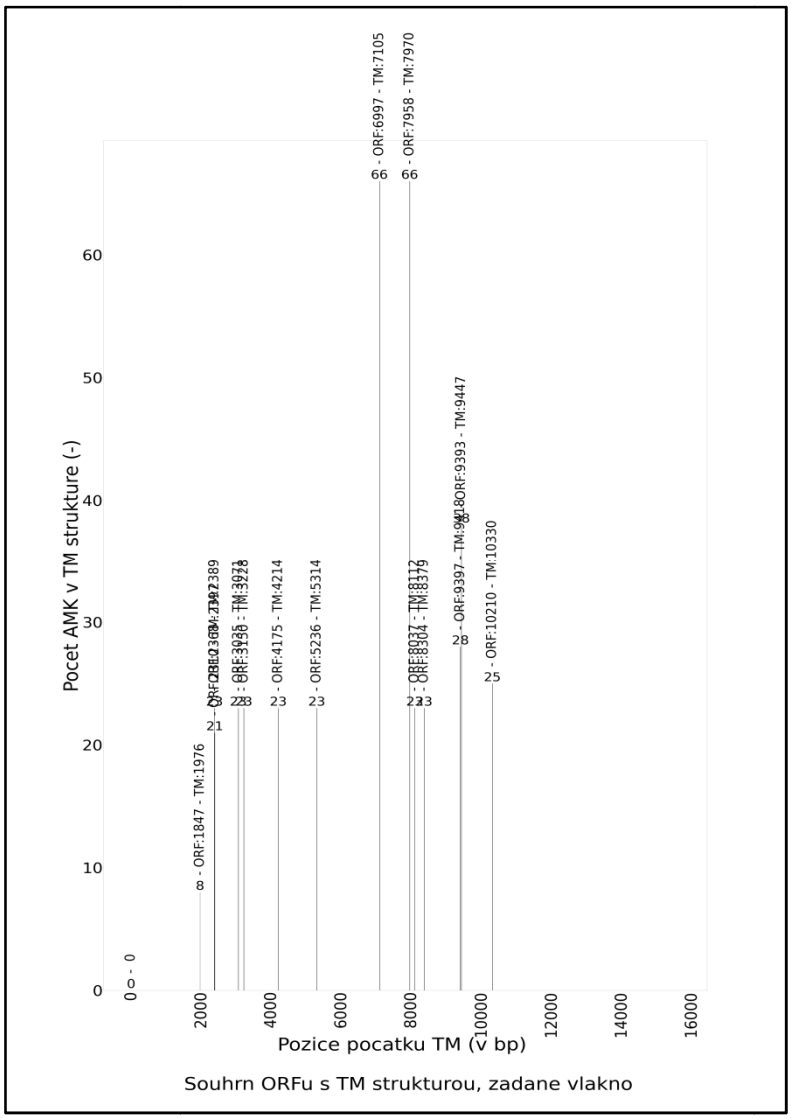
Obr. 4.7: Zobrazení souhrnu ORFů pro dané vlákno a jeden čtecí rámec

Dále se na HTML stránce nachází níže trojice obrázků pro TM, CC a GPI struktury. Viz ilustrativní obrázek 4.8 níže.



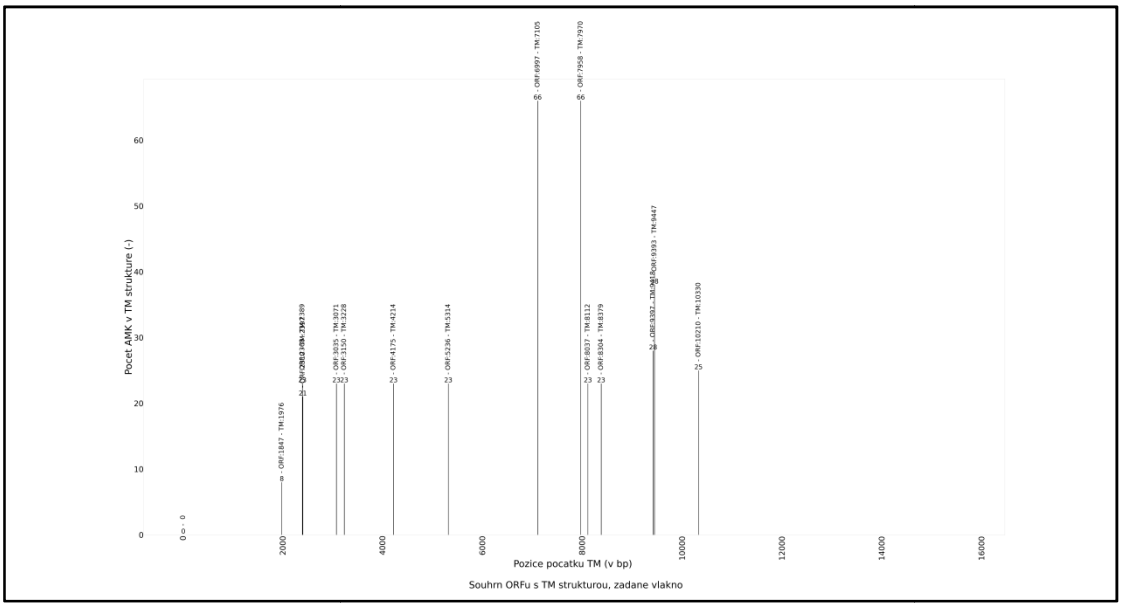
Obr. 4.8: Vzhled trojice obrázků pro souhrn TM, CC a GPI struktur v daném vlákně

Na obrázku 4.9 níže je uveden příklad zobrazení pro TM sekundární struktury pro dané vlákno. Jedná se o obrázek první zleva ze souhrnu na obrázku 4.8 výše. (Na těchto obrázcích je vždy zachycena situace pro všechny tři čtecí rámce do jednoho obrázku).

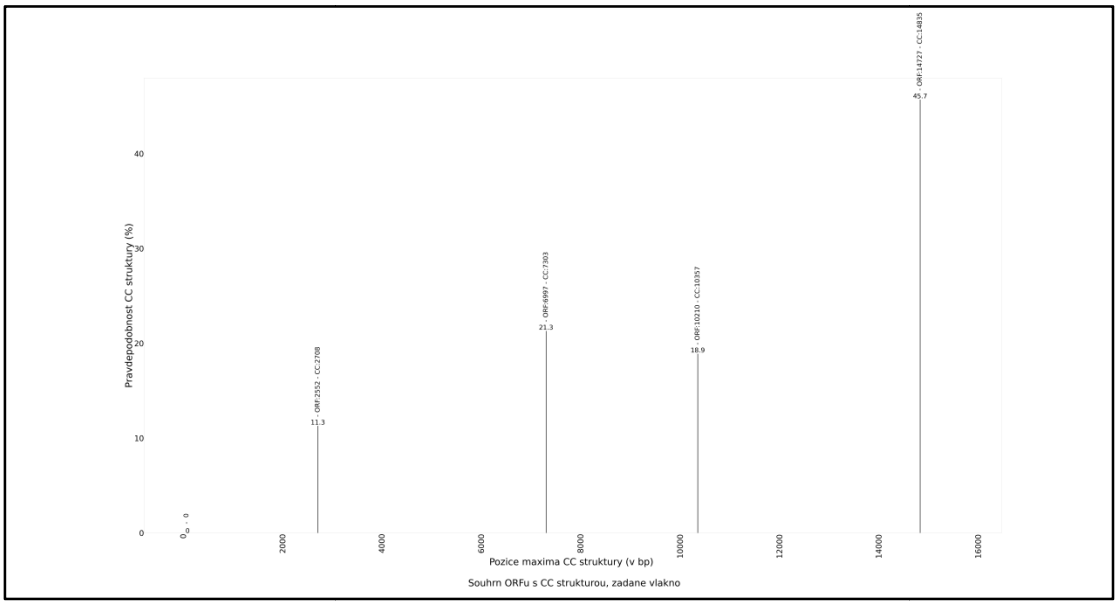


Obr. 4.9: Souhrn ORFů s TM strukturami pro jedno z vláken DNA

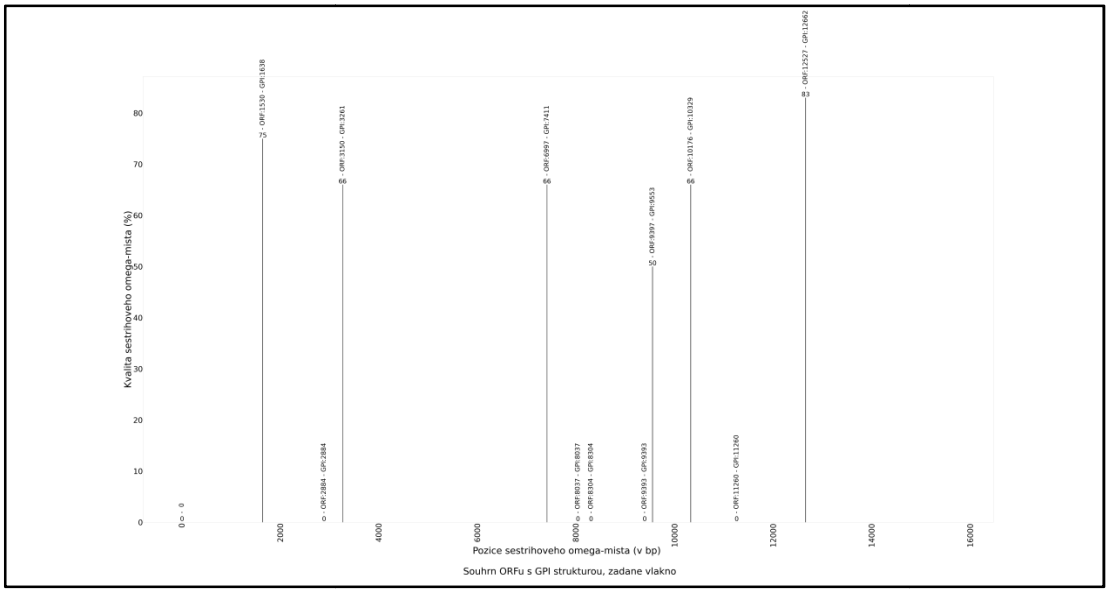
Ve vygenerované HTML stránce jsou dále také zobrazeny zvětšené jednotlivé obrázky uvedené zde v souhrnném obrázku 4.8 výše. Tj. tři obrázky pro každé z vláken DNA. Shora dolů jde o obrázky pro TM sekundární struktury, poté CC struktury a na konec pro GPI modifikace. Cílem bylo roztáhnout vodorovnou osu tak, aby se zobrazované informace pro jednotlivé ORFy a sekundární struktury co nejméně překrývaly - viz obrázky 4.10, 4.11 a 4.12 níže (zde pro zadané vlákno, vždy jsou shrnuty všechny čtecí rámce do jednoho obrázku).



Obr. 4.10: Zvětšené zobrazení pro TM struktury jednoho vlákna DNA



Obr. 4.11: Zvětšené zobrazení pro CC struktury jednoho vlákna DNA



Obr. 4.12: Zvětšené zobrazení pro GPI modifikace jednoho vlákna DNA

Na stránce se poté nacházejí dvě tabulky uspořádané pod sebou. První z nich shora rozděluje ORFy dle nalezené sekundární struktury do tří sloupců. V prvním sloupci je TM, ve druhém CC a ve třetím GPI struktura. V každém sloupci jsou tyto ORFy roztříděné do tří čtecích rámců (shora dolů: 1., 2., 3. čtecí rámec). V této tabulce není zohledněno, zda se v daném ORFu nenachází ještě i jiná sekundární struktura. Tabulka slouží k rychlému nalezení daného ORFu s danou strukturou dle obrázků 4.10, 4.11 a 4.12 uvedených výše se souhrnem sekundárních struktur. Na obrázku 4.13 níže je vyobrazen výřez z výše zmíněné první tabulky.

ORFy s TM strukturou - zadané vlákno	ORFy s CC strukturou - zadané vlákno	OF
1. čtecí rámec: ORF-pozice: [585, 999] ORF-pozice: [2211, 2520] ORF-pozice: [3636, 4134] ORF-pozice: [5253, 5676] ORF-pozice: [13614, 13848]	**1. čtecí rámec:** ORF-pozice: [585, 999] ORF-pozice: [14577, 14757]	
2. čtecí rámec: ORF-pozice: [3307, 3748] ORF-pozice: [6349, 6535]	**2. čtecí rámec:** ORF-pozice: [14128, 14527]	

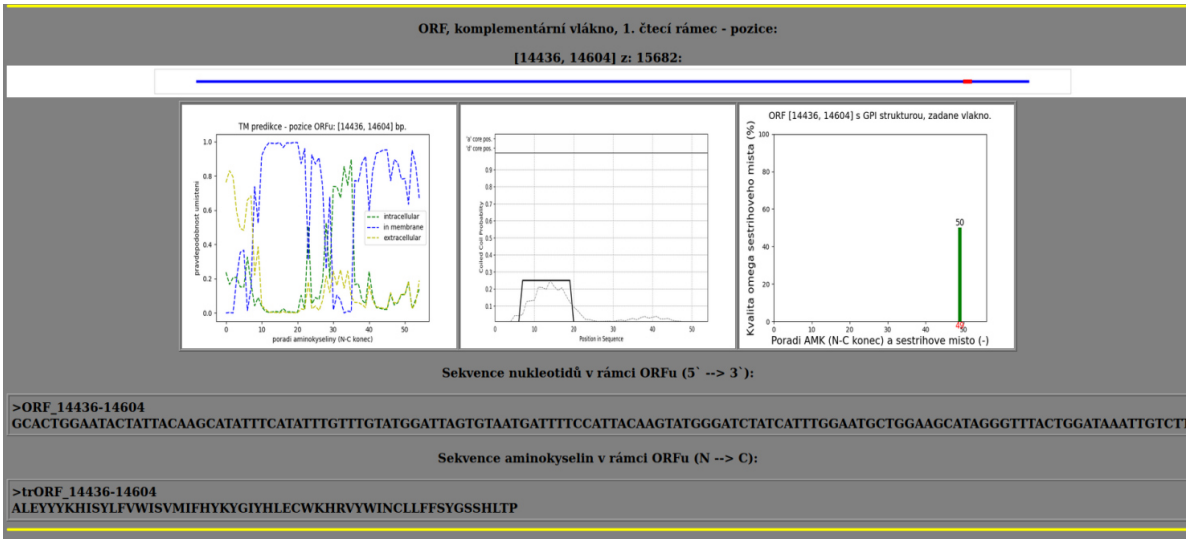
Obr. 4.13: Výřez z první tabulky HTML stránky

Níže následuje další tabulka, kde jsou nalezené ORFy roztříděny do skupin zobrazených v jednotlivých řádcích tabulky. Dle výše zmíněného třídění do sedmi možných kombinací. Např. v prvním řádku jsou zachyceny ORFy obsahující jen TM sekundární strukturu, na pátém řádku jsou ORFy se zachycenou TM a GPI strukturou a na posledním sedmém řádku je souhrn ORFů obsahujících jak TM, tak CC tak i GPI strukturu. Každý řádek je dále roztříděn do tří sloupců dle tří čtecích rámců. Na obrázku 4.14 níže je zobrazen výřez z této tabulky.

Obr. 4.14: Výřez z druhé tabulky HTML stránky

Tato sestava, tj. trojice obrázků se souhrnem ORFů, dvě trojice obrázků se vyznačením zvlášť TM, CC a GPI sekundárních struktur v zadaném vlákně a dále obě tabulky pro zadané vlákno jsou vytvořeny následně na stránce též pro komplementární vlákno.

Pod těmito spíše souhrnnými informacemi je vytvořeno podrobnější vyobrazení situace v rámci daného ORFu, v němž byla detekována, předpovězena alespoň jedna ze stanovovaných sekundárních struktur (TM, CC a GPI). Na tyto jednotlivá zobrazení pro konkrétní ORF vedou odkazy vytvořené v rámci této HTML stránky v obou výše zmíněných tabulkách. Nejprve jsou v jednom souvislém sledu vyobrazeny situace pro zadané vlákno a poté, tj. již na závěr stránky, jsou obdobná zobrazení pro komplementární vlákno. Na ilustrativním obrázku 4.15 níže je zachycen příklad pro ORF s předpovězenými všemi třemi sekundárními strukturami.



Obr. 4.15: Způsob vyobrazení situace v ORFu se všemi strukturami

Na obrázku 4.15 výše je příklad vyobrazení ORFu, u nějž byly předpovězeny všechny tři stanovované sekundární struktury (zleva doprava: TM, CC a GPI). Nejvýše je uvedeno, že se jedná o ORF v rámci komplementárního vlákna, 1. čtecí rámec a je uvedena pozice (v bp) tohoto konkrétního ORFu (začátek 14436, konec 14604 – z celkové délky vlákna 15682 bp). Následuje obrázek s vykreslením pozice ORFu v rámci vlákna a dále tři obrázky s vyobrazením předpovězených struktur v rámci daného ORFu.

Níže je uvedena sekvence nukleotidů v rámci daného ORFu ve FASTA formátu (tj. s hlavičkou identifikující ORF).

Pod ní je obdobně uvedena sekvence aminokyselinových zbytků pro daný ORF ve formátu FASTA (tj. s hlavičkou identifikující ORF).

Na závěr skript vypíše název souboru, pod kterým byla daná vygenerovaná HTML stránka uložena. V případě ošetření ambiguitního kódu pomocí první verze (volené číslem 1 – viz výše) je skript ukončen příkazem „quit()“. Uživateli je zobrazen text doporučující v následujícím dialogovém okně pro setrvání v shellu zmáčknout tlačítko „Cancel“. Zbylá dvě ošetření ambiguitního kódu již takovéto omezení neobsahují.

5 Výsledky

Za výsledek této práce lze považovat vytvořený program, také HTML stránku jím vygenerovanou a dále použití výsledků (zachycené v HTML stránce) běhu programu na zadané sekvenci DNA. Cílem bylo poskytnout odborníkovi soubor dat včetně grafů pro rozhodnutí, zda v daném místě DNA se může nacházet gen pro tetherin/Bst 2 protein. Nebylo ambicí naprogramovat analogii rozhodování experta, ale pouze poskytnout informace pro jeho rozhodnutí.

Lze konstatovat, že v této práci se zřejmě jedná o první systematické vyhledávání antivirového genu *bst-2* a příbuzných protivirových genů pomocí *de novo* predikce z genomových dat. Prozatím byly k dispozici jen programy pro samostatné predikce TM nebo CC domén nebo ω-míst pro GPI modifikaci. Nebyl zde však nástroj, který by byl určen pro predikce těchto struktur dohromady na rozsáhlejším úseku DNA.

Použitý způsob hledání genů pro tetherin/Bst2 protein (viz kapitola 4 Metody) vychází ze skutečnosti, že nukleotidová a návazně aminokyselinová sekvence je u tetherinů velmi variabilní. Různé tetheriny sdílejí jen výsledné uspořádání sekundárních struktur. Tyto geny tedy nelze vyhledávat jinou cestou.

Současně s vyhledáváním genů pro tetheriny je vyhledáván i TM-CC(aT) gen. Což je gen neznámé funkce, má ale podobné domény a v evoluci zřejmě společného předka s tetheriny.

Programem jsou prozkoumávány jednotlivé ORFy nikoliv exony. Samotné exony je obtížné predikovat a zvláště v případě neznámých sekvencí DNA je i výhodnější vyšetřovat nejprve otevřené čtecí rámce.

V HTML stránce důležitou informaci obsahují především grafy s udanými pozicemi kódovaných sekundárních struktur (TM, CC a GPI), jejich ORFů a s udaným základním popisem těchto predikovaných sekundárních struktur. Z těchto grafů vede informace o umístění daného ORFu k odkazům v tabulkách na HTML stránce a ty odkazují na podrobnější popis/vykreslení situace v daném ORFu v souboru podrobnějších obrázků uvedených níže na stránce.

Pro posouzení kvality předpovědí vytvořeného programu budou výsledky běhu programu vyzkoušeny v této kapitole na úsecích DNA kódujících tetheriny s již dříve určenými exony a strukturami jimi kódovanými.

K tomu budou použity především grafy se znázorněnými pozicemi kódovaných sekundárních struktur a přiřazené číselné údaje, které bude možno porovnat s reálnou situací v dané sekvenci DNA.

Použito bude pět lokusů od různých zástupců obratlovců, kde je známo rozmístění kódování TM a CC domén a ω-místa pro GPI modifikaci. Konkrétně od člověka (*Homo sapiens sapiens*), od kura domácího (*Gallus gallus*), od myši domácí (*Mus musculus*), od luskouna ostrovního (*Manis javanica*) a od kaloně vábivého (*Pteropus alecto*).

Výsledek bude ukázán také na čistě náhodných sekvencích (s rovnoměrným rozložením pravděpodobnosti výskytu pro všechny baze DNA bez dalších podmínek).

Nejvhodnější variantou ošetření ambiguitního kódu je varianta třetí, tj. zahození ORFu s byť i jen jedním neznámým znakem. Proto při zpracování dat pro kapitolu 5 Výsledky byla použita jen tato varianta.

5.1 Příklad použití – úsek lidského genomu

Skript byl nastaven na minimální délku ORFu 150 bp. Pro ošetření ambiguitního kódu byla vybrána varianta, kdy ORF byť i jen s jedním neznámým znakem je zahozen. Jako úsek DNA byl vybrán úsek lidského genomu dlouhý 16570 párů bazí (bp). Obsahuje gen pro tetherin/Bst2 a vedlejší gen označovaný jako TM-CC(aT) – (aT znamená „adjacent to tetherin“).

Popis polohy lokusu v lidském genomu:
>hg38_dna range=chr19:17389648-17406217

, čili se jedná o lidský genom verze 38 (hg38), chromozom 19, uvedený rozsah umístění na devatenáctém chromozomu je v nukleotidech resp. párech bazí.

Gen pro tetherin je prozkoumán velmi dobře, trochu méně již gen TM-CC(aT). U obou genů jsou již s jistotou určeny kódované sekundární struktury a v případě genu pro tetherin i ω-místo pro připojení GPI kotvy.

Na obrázku 5.1 níže je zobrazeno rozložení oblastí kódujících sekundární struktury a ω-místo. Zleva doprava v orientaci 5‘ konec → 3‘ konec. Samotné vlákno DNA o rozsahu 16570 bp je zobrazené horizontální černou úsečkou, červeně jsou úseky kódující TM (transmembránové) domény, tmavě modře CC (coiled-coil) domény a žlutou značkou je označena pozice ω-místa pro připojení GPI kotvy.

Obr. 5.1: Přehled úseků kódujících TM a CC domény a GPI ω-místo

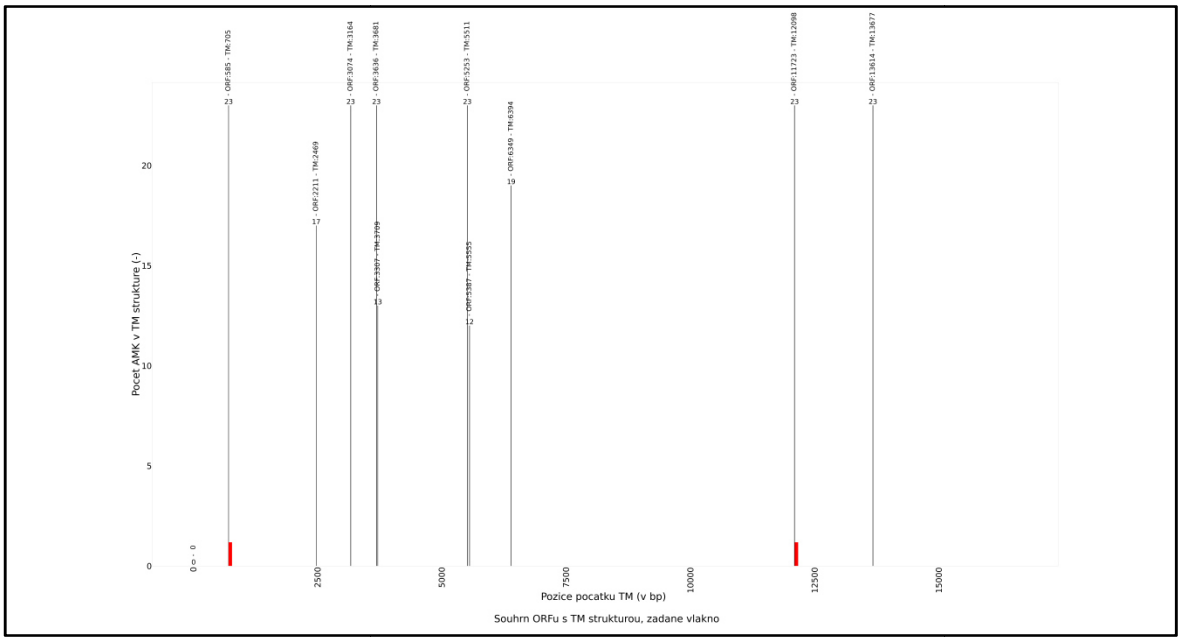
Úseky kódující TM domény:
– 705 až 774 bp,
– 12101 až 12170 bp.

Úseky kódující CC domény:
– 816 až 924 bp,
– 1780 až 1847 bp,

– 2028 až 2089 bp,
– 2393 až 2445 bp,
– 12236 až 12383 bp,
– 14371 až 14468 bp,
– 14874 až 15007 bp,
– 15558 až 15574 bp.

Pozice ω-místa pro připojení GPI kotvy:
– 2463 bp.

Porovnání predikce TM domén a reálného rozložení kódování TM domén ve vyšetřovaném vláknu DNA – viz obrázek 5.2 níže (červeně jsou pozice reálně přítomných úseků DNA kódujících TM domény):



Obr. 5.2: Porovnání předpovědí TM struktur a reálné situace v úseku DNA

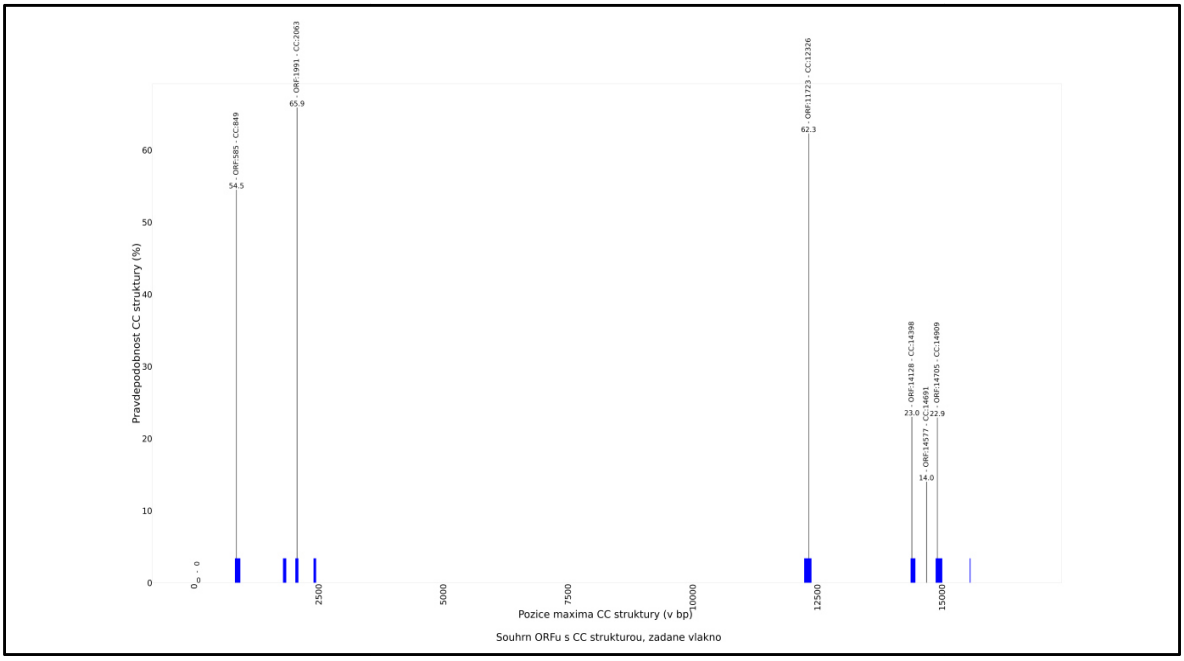
Na vodorovné ose obrázku 5.2 je rozsah vlákna DNA v párech bazí (bp), na svislé ose počet aminokyselinových zbytků v dané predikované TM doméně. Úsečky kolmé na vodorovnou osu v obrázku 5.2 udávají pozici začátku predikované kódované TM struktury.

Na obrázku 5.2 výše je patrné, že ke shodě mezi predikcí a reálnou pozicí úseků kódujících TM sekundární struktury došlo jen dvakrát v případě první a předposlední předpovězené TM struktury.

Udané predikce, kde je shoda nebo téměř shoda s reálnou situací jsou:
– reálná poloha začátku TM struktury: 705 bp, předpovězená: 705 bp,
– reálná poloha začátku TM struktury: 12101 bp, předpovězená: 12098 bp.

Zbylé predikce jsou falešně pozitivní nálezy. Jedná se zde tedy o 8 falešně pozitivních nálezů. Není zde falešně negativní nález.

Porovnání predikce CC domén a reálného rozložení kódování CC domén ve vyšetřovaném vláknu DNA – viz obrázek 5.3 níže (modře jsou pozice reálně přítomných úseků DNA kódujících CC domény):



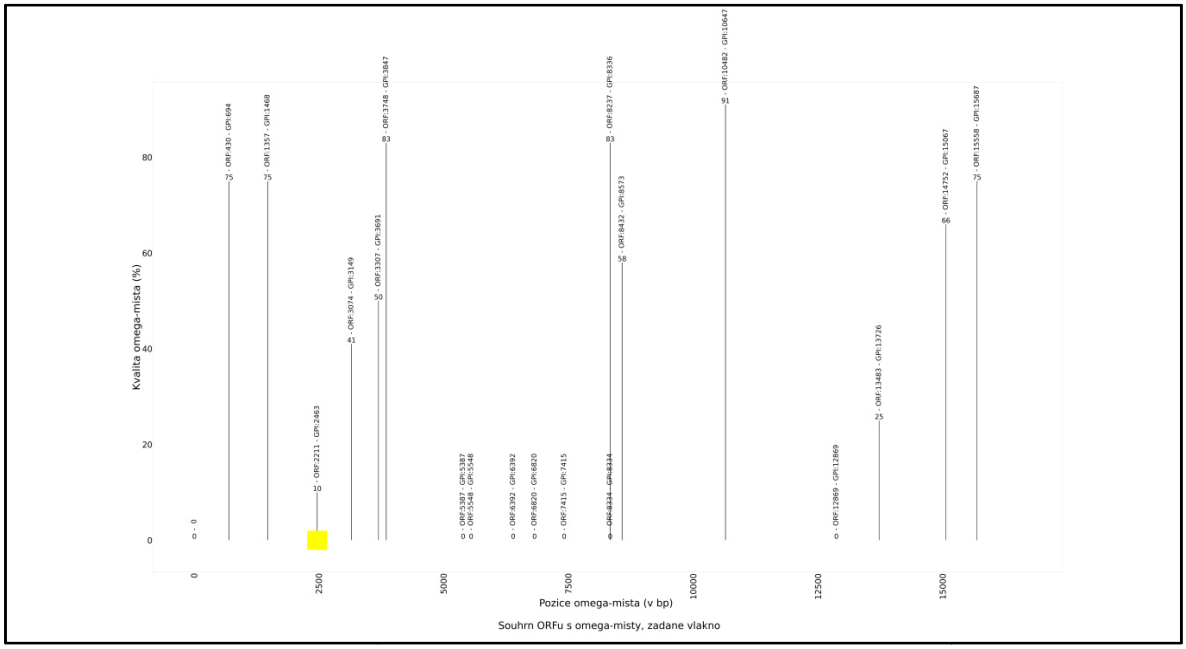
Obr. 5.3: Porovnání předpovědí CC struktur a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.3 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je maximální pravděpodobnost dané predikované CC struktury. Úsečky kolmé na vodorovnou osu v obrázku 5.3 udávají pozici předpovědi CC struktury s nejvyšší pravděpodobností v rámci daného ORFu, kde byla predikována CC struktura.

Úspěšnost, resp. neúspěšnost predikování CC struktury v rámci reálného umístění kódování CC domén:

– 816 až 924 bp – předpověď: 849 bp,
– 1780 až 1847 bp – nepředpovězeno, falešně negativní výsledek,
– 2028 až 2089 bp – předpověď: 2063 bp,
– 2393 až 2445 bp – nepředpovězeno, falešně negativní výsledek,
– 12236 až 12383 bp – předpověď: 12326 bp,
– 14371 až 14468 bp – předpověď: 14398 bp,
– 14874 až 15007 bp – předpověď: 14909 bp,
– 15558 až 15574 bp – nepředpovězeno, falešně negativní výsledek (jedná se o velmi krátký úsek – 16 bp),
– falešně pozitivní výsledek: predikce na pozici 14691 bp.

Porovnání predikce ω-místa pro GPI modifikaci a reálného umístěníω-místa ve vyšetřovaném vláknu DNA – viz obrázek 5.4 níže (žlutou značkou je reálná pozice ω-místa):



Obr. 5.4: Porovnání předpovědi ω-místa a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.4 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je kvalita ω-místa v procentech. Úsečky kolmé na vodorovnou osu v obrázku 5.4 udávají pozici předpovědi ω-místa pro GPI modifikaci.

Reálná pozice ω-místa pro připojení GPI kotvy:
– 2463 bp.

Tato pozice je určena jako jedno z predikovaných ω-míst – přesně na pozici 2463 bp. Na obrázku 5.4 výše se jedná o predikci třetí zleva. Kvalita ω-místa je zde stanovena pouze jako 10%.

Ostatní predikce jsou falešně pozitivní nálezy. Jedná se o 18 falešně pozitivních nálezů. Z nich 7 nemá určenu pozici a kvalitu ω-místa.

5.2 Příklad použití – úsek genomu myši domácí

Skript byl nastaven na minimální délku ORFu 150 bp. Pro ošetření ambiguitního kódu byla vybrána varianta, kdy ORF byť i jen s jedním neznámým znakem je zahozen. Jako úsek DNA byl vybrán úsek genomu myši domácí (*Mus musculus*) dlouhý 19192 párů bazí (bp). Obsahuje gen pro tetherin/Bst2 a vedlejší gen označovaný jako TM-CC(aT) – (aT znamená „adjacent to tetherin“).

Popis polohy lokusu v myším genomu:
>mm10_dna range=chr8:71519771-71538962

, jedná se o genom myši domácí verze 10 (mm10), chromozom 8, uvedený rozsah umístění na osmém chromozomu je v nukleotidech resp. párech bazí.

Gen pro tetherin je prozkoumán velmi dobře, trochu méně již gen TM-CC(aT). U obou genů jsou již s jistotou určeny kódované sekundární struktury a v případě genu pro tetherin i ω-místo pro připojení GPI kotvy.

Na obrázku 5.5 níže je zobrazeno rozložení oblastí kódujících sekundární struktury a ω-místo. Zleva doprava v orientaci 5‘ konec → 3‘ konec. Samotné vlákno DNA o rozsahu 19192 bp je zobrazené horizontální černou úsečkou, červeně jsou úseky kódující TM (transmembránové) domény, tmavě modře CC (coiled-coil) domény a žlutou značkou je označena pozice ω-místa pro připojení GPI kotvy.

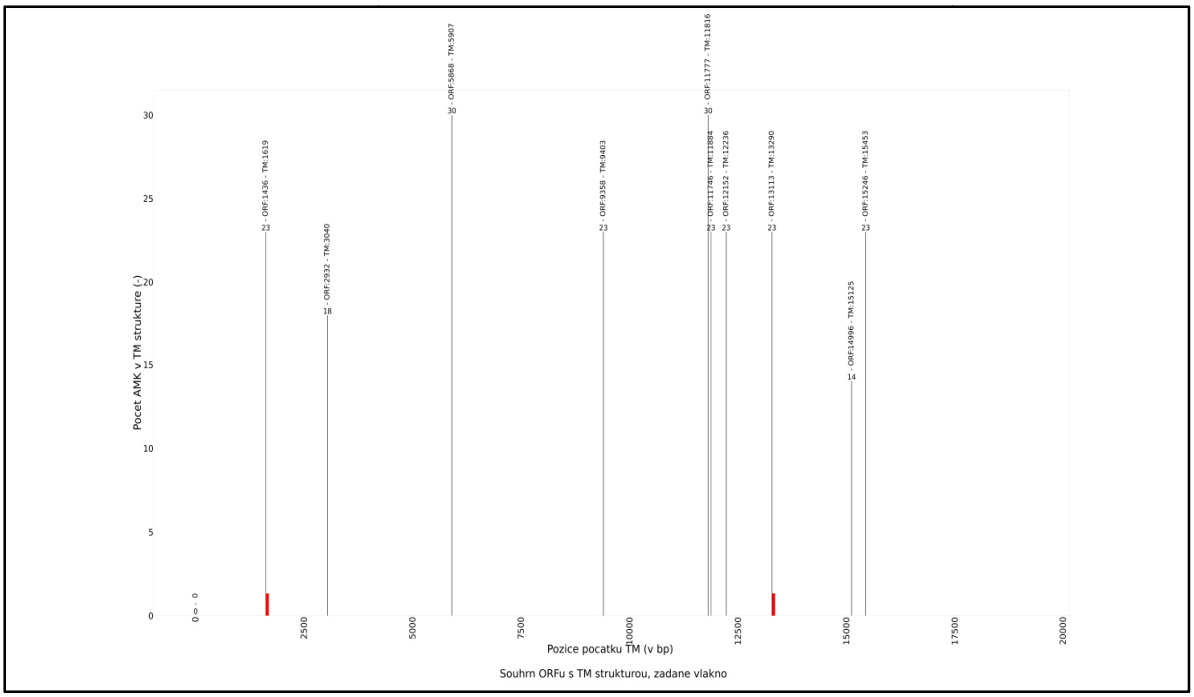
Obr. 5.5: Přehled úseků kódujících TM a CC domény a GPI ω-místo

Úseky kódující TM domény:
- 1619 až 1688 bp,
- 13293 až 13362 bp.

Úseky kódující CC domény:
- 1712 až 1835 bp,
- 2528 až 2604 bp,
- 3137 až 3177 bp,
- 4170 až 4204 bp,
- 13428 až 13578 bp,
- 16406 až 16503 bp,
- 16991 až 17124 bp,
- 17580 až 17626 bp.

Pozice ω-místa pro připojení GPI kotvy:
– 4210 bp.

Porovnání predikce TM domén a reálného rozložení kódování TM domén ve vyšetřovaném vláknu DNA – viz obrázek 5.6 níže (červeně jsou pozice reálně přítomných úseků DNA kódujících TM domény):



Obr. 5.6: Porovnání předpovědí TM struktur a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.6 je rozsah vlákna DNA v párech bazí (bp), na svislé ose počet aminokyselinových zbytků v dané predikované TM doméně. Úsečky kolmé na vodorovnou osu v obrázku 5.6 udávají pozici začátku predikované kódované TM struktury.

Na obrázku 5.6 výše je patrné, že ke shodě mezi predikcí a reálnou pozicí úseků kódujících TM sekundární struktury došlo jen dvakrát v případě první a osmé predikce TM struktury z celkem deseti.

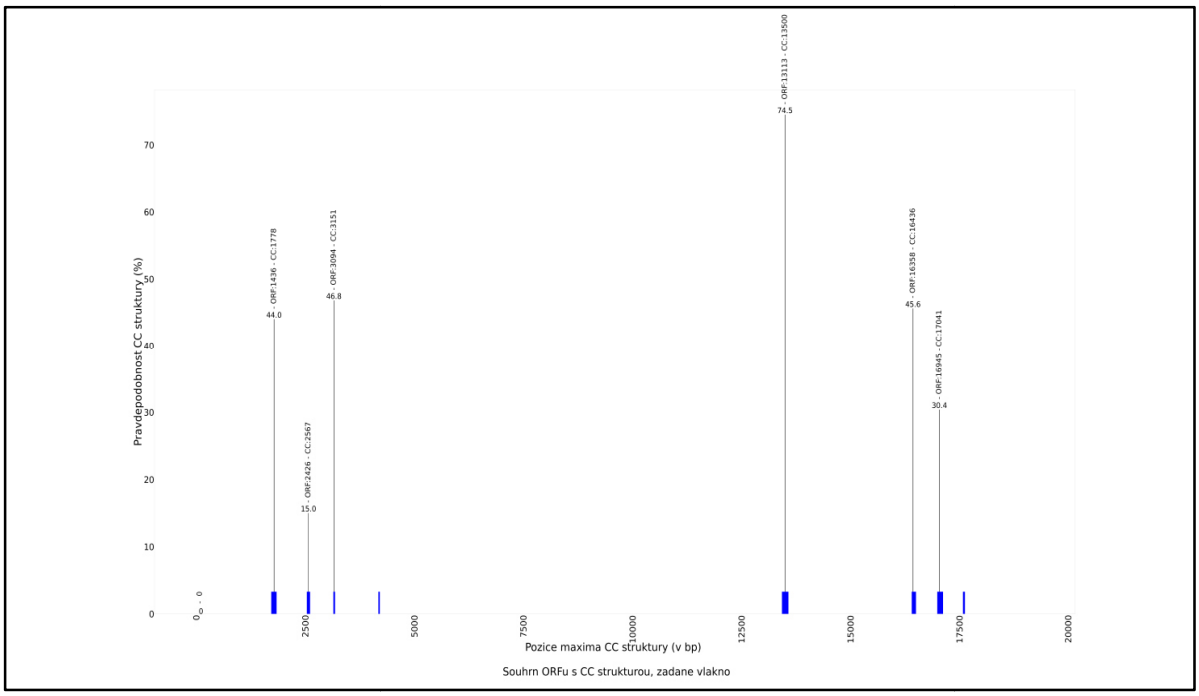
Udané predikce, kde je shoda nebo téměř shoda s reálnou situací jsou:

– reálná poloha začátku TM struktury: 1619 bp, předpovězená: 1619 bp,

– reálná poloha začátku TM struktury: 13293 bp, předpovězená: 13290 bp.

Zbylé predikce jsou falešně pozitivní nálezy. Jedná se zde tedy o 8 falešně pozitivních nálezů z celkem 10 predikcí. Není zde falešně negativní nález.

Porovnání predikce CC domén a reálného rozložení kódování CC domén ve vyšetřovaném vláknu DNA – viz obrázek 5.7 níže (modře jsou pozice reálně přítomných úseků DNA kódujících CC domény):



Obr. 5.7: Porovnání předpovědí CC struktur a reálné situace v úseku DNA

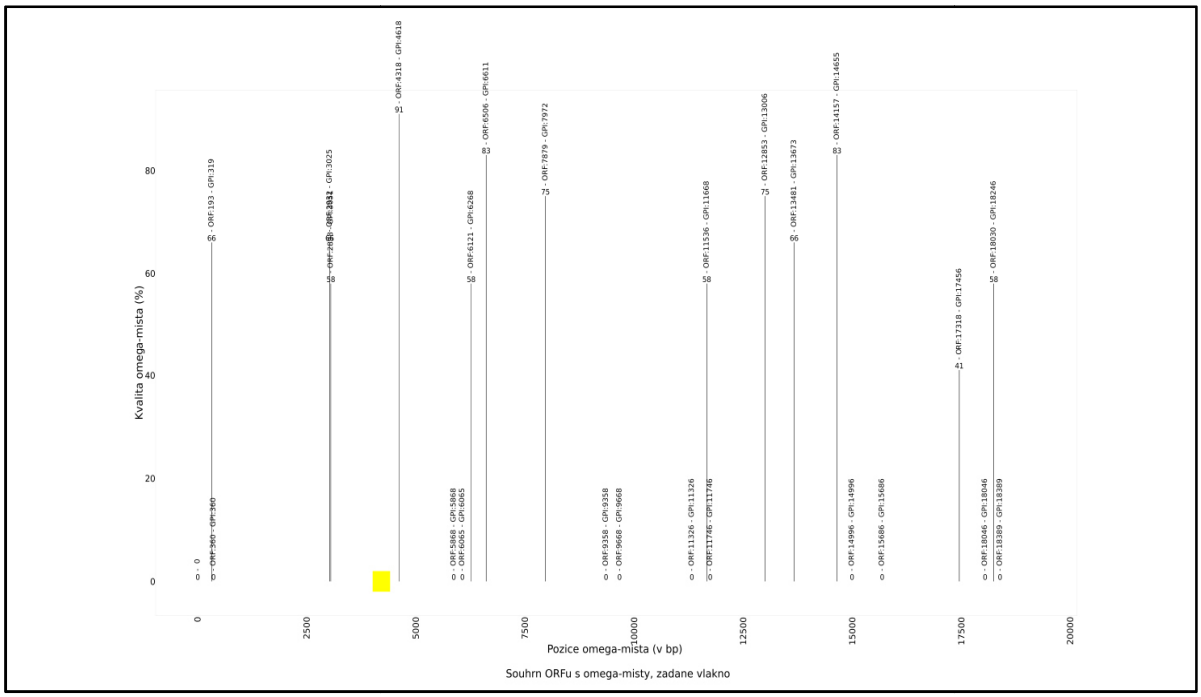
Na vodorovné ose obrázku 5.7 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je maximální pravděpodobnost dané predikované CC struktury. Úsečky kolmé na vodorovnou osu v obrázku 5.7 udávají pozici předpovědi CC struktury s nejvyšší pravděpodobností v rámci daného ORFu, kde byla predikována CC struktura.

Úspěšnost, resp. neúspěšnost predikování CC struktury v rámci reálného umístění kódování CC domén:

- 1712 až 1835 bp - předpověď: 1778 bp,
- 2528 až 2604 bp - předpověď: 2567 bp,
- 3137 až 3177 bp - předpověď: 3151 bp,
- 4170 až 4204 bp - nepředpovězeno, falešně negativní výsledek,
- 13428 až 13578 bp - předpověď: 13500 bp,
- 16406 až 16503 bp - předpověď: 16436 bp,
- 16991 až 17124 bp - předpověď: 17041 bp,
- 17580 až 17626 bp - nepředpovězeno, falešně negativní výsledek.

Není zde falešně pozitivní výsledek.

Porovnání predikce ω-místa pro GPI modifikaci a reálného umístění ω-místa ve vyšetřovaném vláknu DNA – viz obrázek 5.8 níže (žlutou značkou je reálná pozice ω-místa):



Obr. 5.8: Porovnání předpovědi ω-místa a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.8 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je kvalita ω-místa v procentech. Úsečky kolmé na vodorovnou osu v obrázku 5.8 udávají pozici předpovědi ω-místa pro GPI modifikaci.

Reálná pozice ω-místa pro připojení GPI kotvy:
– 4210 bp.

Tato pozice není predikována. Nejbližším predikovaným místem je pozice 4618 bp. Kvalita ω-místa je v této nejbližší predikci stanovena na 91 %.

Další predikce jsou též falešně pozitivní nálezy. Jedná se celkem o 24 falešně pozitivních nálezů. Z nich 11 nemá určenu pozici a kvalitu ω-místa.

5.3 Příklad použití – úsek genomu kura domácího

Skript byl nastaven na minimální délku ORFu 150 bp. Pro ošetření ambiguitního kódu byla vybrána varianta, kdy ORF byť i jen s jedním neznámým znakem je zahozen. Jako úsek DNA byl vybrán úsek genomu kura domácího (*Gallus gallus*) dlouhý 6948 párů bazí (bp). Obsahuje gen pro tetherin/Bst2 a vedlejší gen označovaný jako TM-CC(aT) – (aT znamená „adjacent to tetherin“).

Popis polohy lokusu v kuřecím genomu:
>galGal6_dna range=chr28:3531163-3538110

, jedná se o kuřecí genom verze 6 (galGal6), chromozom 28, uvedený rozsah umístění na dvacátém osmém chromozomu je v nukleotidech resp. párech bazí.

Gen pro tetherin je prozkoumán velmi dobře, trochu méně již gen TM-CC(aT). U obou genů jsou již s jistotou určeny kódované sekundární struktury a v případě genu pro tetherin i ω-místo pro připojení GPI kotvy.

Na obrázku 5.9 níže je zobrazeno rozložení oblastí kódujících sekundární struktury a ω-místo. Zleva doprava v orientaci 5‘ konec → 3‘ konec. Samotné vlákno DNA o rozsahu 6948 bp je zobrazené horizontální černou úsečkou, červeně jsou úseky kódující TM (transmembránové) domény, tmavě modře CC (coiled-coil) domény a žlutou značkou je označena pozice ω-místa pro připojení GPI kotvy.

Obr. 5.9: Přehled úseků kódujících TM a CC domény a GPI ω-místo

Úseky kódující TM domény:

- 1990 až 2059 (bp),
- 3303 až 3372 (bp).

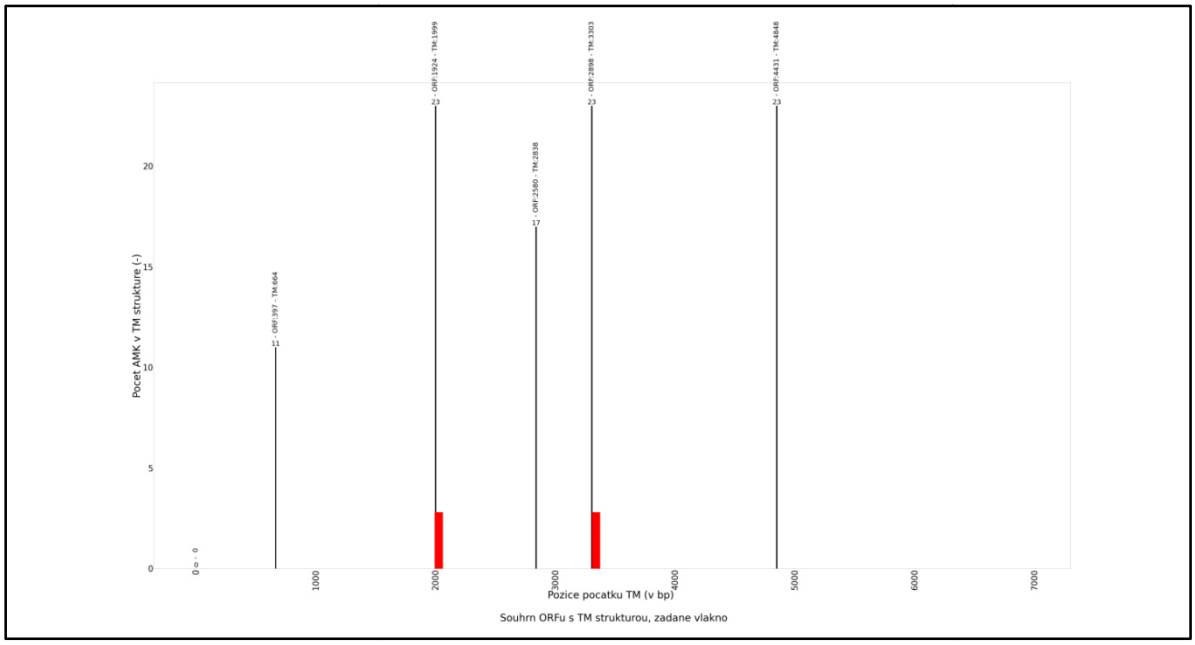
Úseky kódující CC domény:

- 2122 až 2314 (bp),
- 2387 až 2484 (bp),
- 2601 až 2671 (bp),
- 2741 až 2802 (bp),
- 3384 až 3597 (bp),
- 4366 až 4463 (bp),
- 4545 až 4678 (bp),
- 4760 až 4806 (bp).

Pozice ω-místa pro připojení GPI kotvy:

– 2817 bp.

Porovnání predikce TM domén a reálného rozložení kódování TM domén ve vyšetřovaném vláknu DNA – viz obrázek 5.10 níže (červeně jsou pozice reálně přítomných úseků DNA kódujících TM domény):



Obr. 5.10: Porovnání předpovědí TM struktur a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.10 je rozsah vlákna DNA v párech bazí (bp), na svislé ose počet aminokyselinových zbytků v dané predikované TM doméně. Úsečky kolmé na vodorovnou osu v obrázku 5.10 udávají pozici začátku predikované kódované TM struktury.

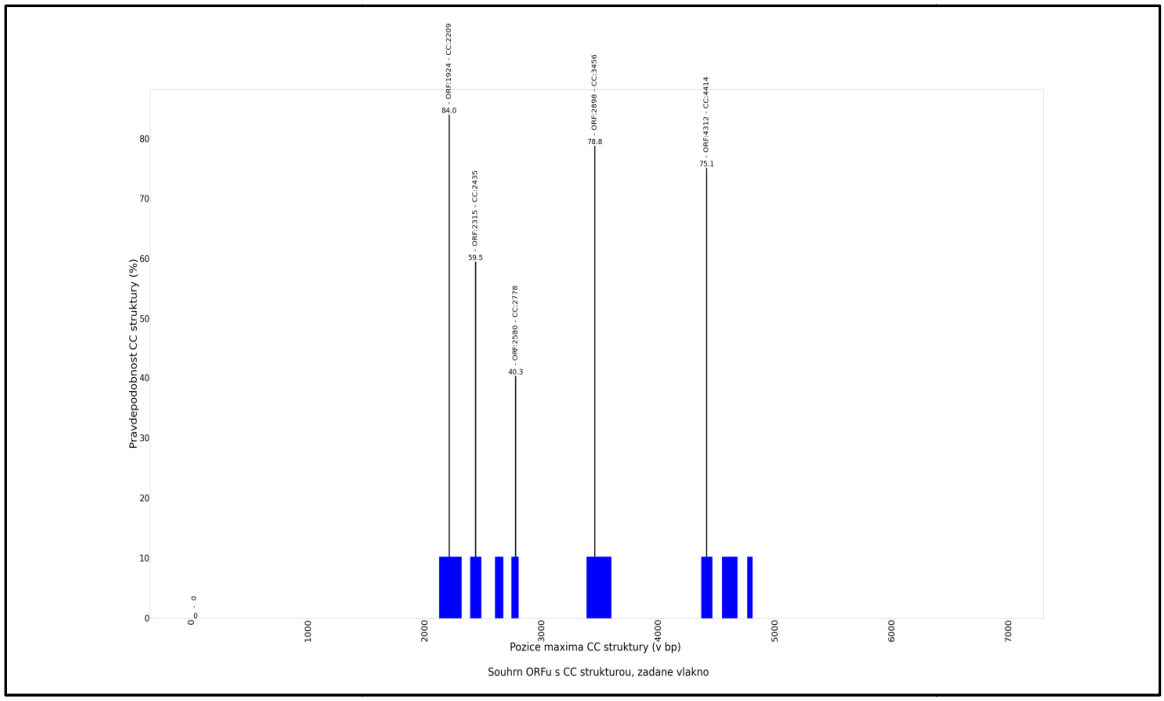
Na obrázku 5.10 výše je patrné, že ke shodě mezi predikcí a reálnou pozicí úseků kódujících TM sekundární struktury došlo jen dvakrát v případě druhé a předposlední, čtvrté předpovězené TM struktury.

Udané predikce, kde je shoda nebo téměř shoda s reálnou situací jsou:

– reálná poloha začátku TM struktury: 1990 bp, předpovězená: 1999 bp,
– reálná poloha začátku TM struktury: 3303 bp, předpovězená: 3303 bp.

Zbylé predikce jsou falešně pozitivní nálezy. Jedná se zde tedy o 3 falešně pozitivní nálezy z celkem pěti predikcí. Není zde falešně negativní nález.

Porovnání predikce CC domén a reálného rozložení kódování CC domén ve vyšetřovaném vláknu DNA – viz obrázek 5.11 níže (modře jsou pozice reálně přítomných úseků DNA kódujících CC domény):



Obr. 5.11: Porovnání předpovědí CC struktur a reálné situace v úseku DNA

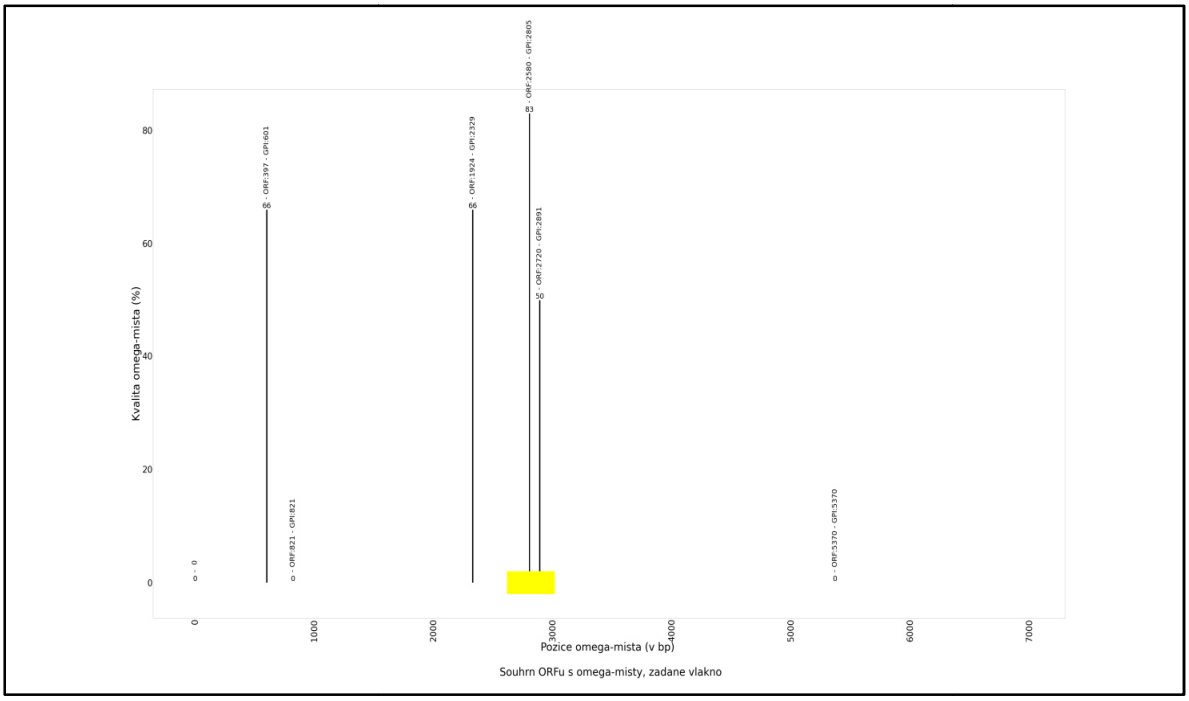
Na vodorovné ose obrázku 5.11 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je maximální pravděpodobnost dané predikované CC struktury. Úsečky kolmé na vodorovnou osu v obrázku 5.11 udávají pozici předpovědi CC struktury s nejvyšší pravděpodobností v rámci daného ORFu, kde byla predikována CC struktura.

Úspěšnost, resp. neúspěšnost predikování CC struktury v rámci reálného umístění kódování CC domén:

– 2122 až 2314 bp – předpověď: 2209 bp,
– 2387 až 2484 bp – předpověď: 2435 bp,
– 2601 až 2671 bp – nepředpovězeno, falešně negativní výsledek,
– 2741 až 2802 bp – předpověď: 2778 bp,
– 3384 až 3597 bp – předpověď: 3456 bp,
– 4366 až 4463 bp – předpověď: 4414 bp,
– 4545 až 4678 bp – nepředpovězeno, falešně negativní výsledek,
– 4760 až 4806 bp – nepředpovězeno, falešně negativní výsledek.

Není zde falešně pozitivní výsledek.

Porovnání predikce ω-místa pro GPI modifikaci a reálného umístění ω-místa ve vyšetřovaném vláknu DNA – viz obrázek 5.12 níže (žlutou značkou je reálná pozice ω-místa):



Obr. 5.12: Porovnání předpovědi ω-místa a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.12 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je kvalita ω-místa v procentech. Úsečky kolmé na vodorovnou osu v obrázku 5.12 udávají pozici předpovědi ω-místa pro GPI modifikaci.

Reálná pozice ω-místa pro připojení GPI kotvy:

– 2817 bp.

Tato pozice není predikována. Nejbližším predikovaným místem je pozice 2805 bp. Kvalita ω-místa je v této nejbližší predikci stanovena na 83 %.

Další predikce jsou falešně pozitivní nálezy. Jedná se o 5 falešně pozitivních nálezů. Z nich 2 nemají určenu pozici a kvalitu ω-místa.

5.4 Příklad použití – úsek genomu luskouna ostrovního

Skript byl nastaven na minimální délku ORFu 150 bp. Pro ošetření ambiguitního kódu byla vybrána varianta, kdy ORF byť i jen s jedním neznámým znakem je zahozen. Jako úsek DNA byl vybrán úsek genomu luskouna ostrovního (*Manis javanica*) dlouhý 17754 párů bazí (bp). Obsahuje gen pro tetherin/Bst2 a vedlejší gen označovaný jako TM-CC(aT) – (aT znamená „adjacent to tetherin“).

Popis polohy lokusu v genomu luskouna ostrovního:
>ref|NW_023435982.1|:679097-704212 Manis javanica isolate MJ74 unplaced genomic scaffold, YNU_ManJav_2.0 scaffold_88, whole genome shotgun semence

U luskouna ostrovního je dostupná genomová sekvence, ale není zatím určena identita jednotlivých chromozómů. Genom je ve stadiu jednotlivých sekvencí (scaffolds). V tomto případě se jedná o scaffold 88 a na něm o pozici 679097-704212.

Gen pro tetherin je prozkoumán velmi dobře, trochu méně již gen TM-CC(aT). U obou genů jsou již s jistotou určeny kódované sekundární struktury a v případě genu pro tetherin i ω-místo pro připojení GPI kotvy.

Na obrázku 5.13 níže je zobrazeno rozložení oblastí kódujících sekundární struktury a ω-místo. Zleva doprava v orientaci 5‘ konec → 3‘ konec. Samotné vlákno DNA o rozsahu 17754 bp je zobrazené horizontální černou úsečkou, červeně jsou úseky kódující TM (transmembránové) domény, tmavě modře CC (coiled-coil) domény a žlutou značkou je označena pozice ω-místa pro připojení GPI kotvy.

Obr. 5.13: Přehled úseků kódujících TM a CC domény a GPI ω-místo

Úseky kódující TM domény:

- 5864 až 5933 bp,
- 15151 až 15220 bp.

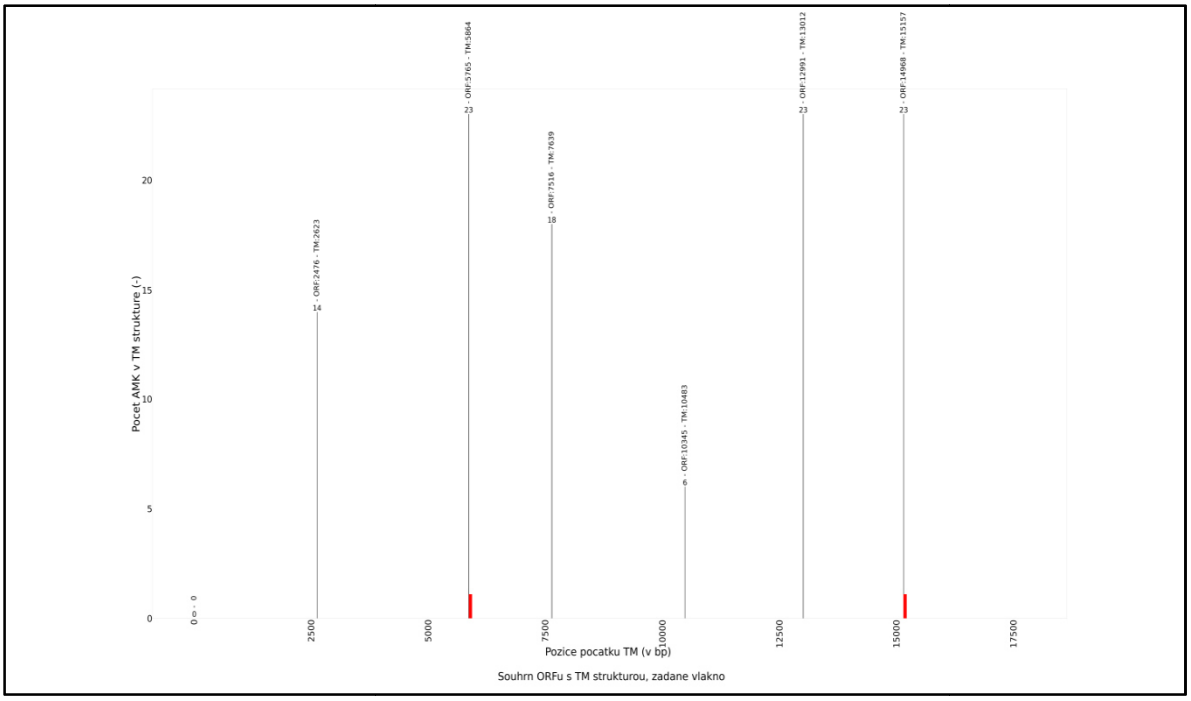
Úseky kódující CC domény:

- 5987 až 6080 bp,
- 6935 až 7011 bp,
- 7258 až 7319 bp,
- 7581 až 7603 bp,
- 15298 až 15439 bp,
- 16489 až 16586 bp,
- 16958 až 17091 bp,
- 17551 až 17591 bp.

Pozice ω-místa pro připojení GPI kotvy:

– 7606 bp.

Porovnání predikce TM domén a reálného rozložení kódování TM domén ve vyšetřovaném vláknu DNA – viz obrázek 5.14 níže (červeně jsou pozice reálně přítomných úseků DNA kódujících TM domény):



Obr. 5.14: Porovnání předpovědí TM struktur a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.14 je rozsah vlákna DNA v párech bazí (bp), na svislé ose počet aminokyselinových zbytků v dané predikované TM doméně. Úsečky kolmé na vodorovnou osu v obrázku 5.14 udávají pozici začátku predikované kódované TM struktury.

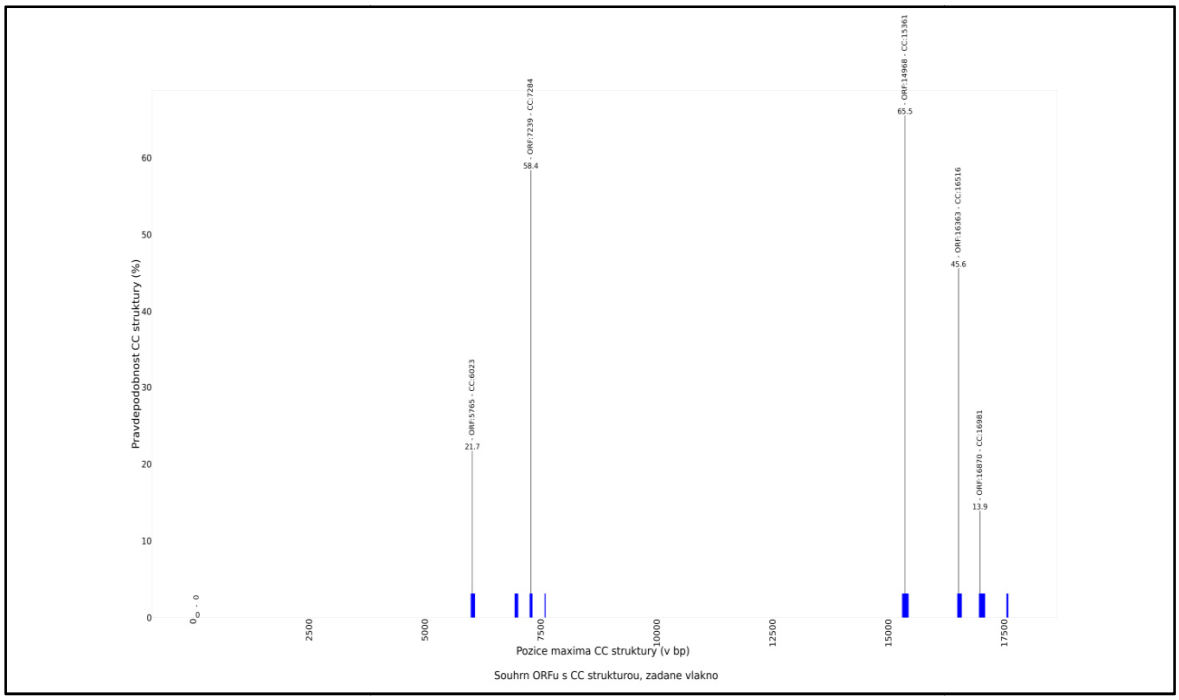
Na obrázku 5.14 výše je patrné, že ke shodě mezi predikcí a reálnou pozicí úseků kódujících TM sekundární struktury došlo jen dvakrát v případě druhé a šesté predikce TM struktury z celkem šesti.

Udané predikce, kde je shoda nebo téměř shoda s reálnou situací jsou:

– reálná poloha začátku TM struktury: 5864 bp, předpovězená: 5864 bp,
– reálná poloha začátku TM struktury: 15151 bp, předpovězená: 15157 bp.

Zbylé predikce jsou falešně pozitivní nálezy. Jedná se zde tedy o 4 falešně pozitivní nálezy z celkem 6ti predikcí. Není zde falešně negativní nález.

Porovnání predikce CC domén a reálného rozložení kódování CC domén ve vyšetřovaném vláknu DNA – viz obrázek 5.15 níže (modře jsou pozice reálně přítomných úseků DNA kódujících CC domény):



Obr. 5.15: Porovnání předpovědí CC struktur a reálné situace v úseku DNA

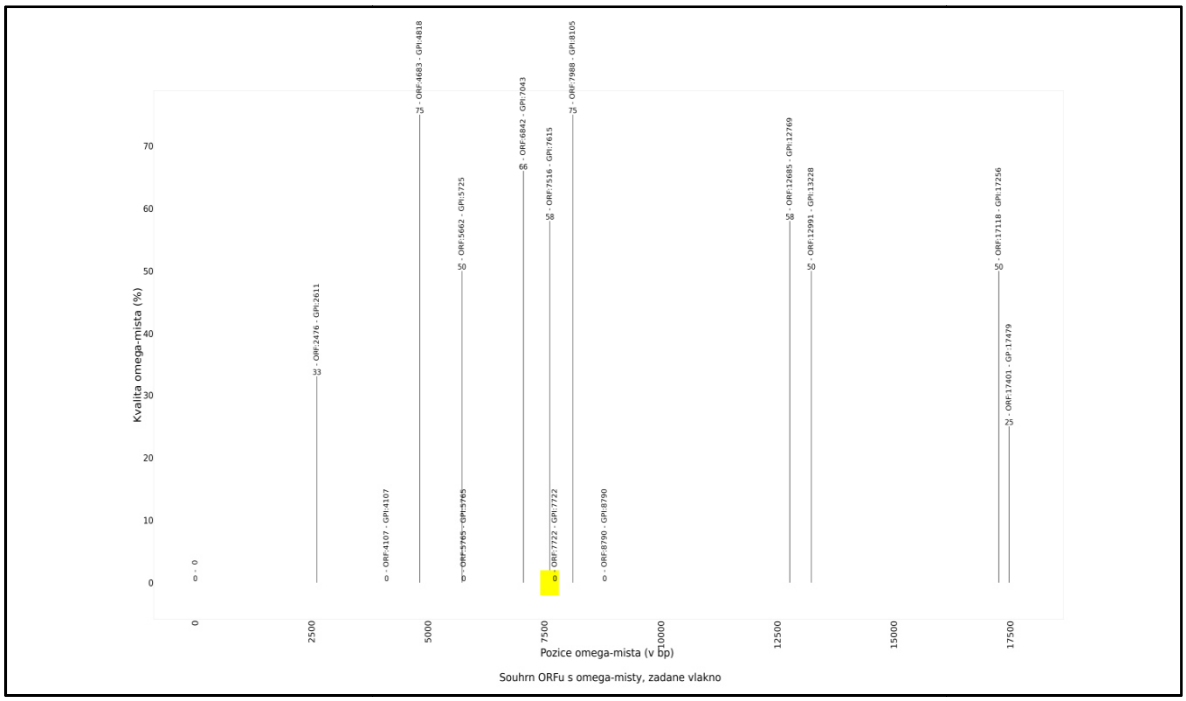
Na vodorovné ose obrázku 5.15 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je maximální pravděpodobnost dané predikované CC struktury. Úsečky kolmé na vodorovnou osu v obrázku 5.15 udávají pozici předpovědi CC struktury s nejvyšší pravděpodobností v rámci daného ORFu, kde byla predikována CC struktura.

Úspěšnost, resp. neúspěšnost predikování CC struktury v rámci reálného umístění kódování CC domén:

- 5987 až 6080 bp - předpověď: 6023 bp,
- 6935 až 7011] bp - nepředpovězeno, falešně negativní výsledek,
- 7258 až 7319 bp - předpověď: 7284 bp,
- 7581 až 7603 bp - nepředpovězeno, falešně negativní výsledek,
- 15298 až 15439 bp - předpověď: 15361 bp,
- 16489 až 16586 bp - předpověď: 16516 bp,
- 16958 až 17091 bp - předpověď: 16981 bp,
- 17551 až 17591 bp - nepředpovězeno, falešně negativní výsledek.

Není zde falešně pozitivní výsledek.

Porovnání predikce ω-místa pro GPI modifikaci a reálného umístění ω-místa ve vyšetřovaném vláknu DNA – viz obrázek 5.16 níže (žlutou značkou je reálná pozice ω-místa):



Obr. 5.16: Porovnání předpovědi ω-místa a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.16 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je kvalita ω-místa v procentech. Úsečky kolmé na vodorovnou osu v obrázku 5.16 udávají pozici předpovědi ω-místa pro GPI modifikaci.

Reálná pozice ω-místa pro připojení GPI kotvy:

– 7606 bp.

Tato pozice není predikována. Nejbližším predikovaným místem je pozice 7615 bp. Kvalita ω-místa je v této nejbližší predikci stanovena na 58 %.

Další predikce jsou též falešně pozitivní nálezy. Jedná se celkem o 14 falešně pozitivních nálezů. Z nich 4 nemají určenu pozici a kvalitu ω-místa.

5.5 Příklad použití – úsek genomu kaloně vábivého

Skript byl nastaven na minimální délku ORFu 150 bp. Pro ošetření ambiguitního kódu byla vybrána varianta, kdy ORF byť i jen s jedním neznámým znakem je zahozen. Jako úsek DNA byl vybrán úsek genomu kaloně vábivého (*Pteropus alecto*) dlouhý 15785 párů bazí (bp). Obsahuje gen pro tetherin/Bst2 a vedlejší gen označovaný jako TM-CC(aT) – (aT znamená „adjacent to tetherin“).

Popis polohy lokusu v genomu kaloně vábivého:
>ref|NW_006429864.1|:923553-939337 Pteropus alecto unplaced genomic scaffold, ASM32557v1 scaffold 160, whole genome shotgun sequence

U kaloně vábivého je dostupná genomová sekvence, ale není zatím určena identita jednotlivých chromozómů. Genom je ve stadiu jednotlivých sekvencí (scaffolds). V tomto případě se jedná o scaffold 160 a na něm o pozici 923553-939337.

Gen pro tetherin je prozkoumán velmi dobře, trochu méně již gen TM-CC(aT). U obou genů jsou již s jistotou určeny kódované sekundární struktury a v případě genu pro tetherin i ω-místo pro připojení GPI kotvy.

Na obrázku 5.17 níže je zobrazeno rozložení oblastí kódujících sekundární struktury a ω-místo. Zleva doprava v orientaci 5‘ konec → 3‘ konec. Samotné vlákno DNA o rozsahu 15785 bp je zobrazené horizontální černou úsečkou, červeně jsou úseky kódující TM (transmembránové) domény, tmavě modře CC (coiled-coil) domény a žlutou značkou je označena pozice ω-místa pro připojení GPI kotvy.

Obr. 5.17: Přehled úseků kódujících TM a CC domény a GPI ω-místo

Úseky kódující TM domény:

- 4047 až 4116 bp,
- 10555 až 10624 bp.

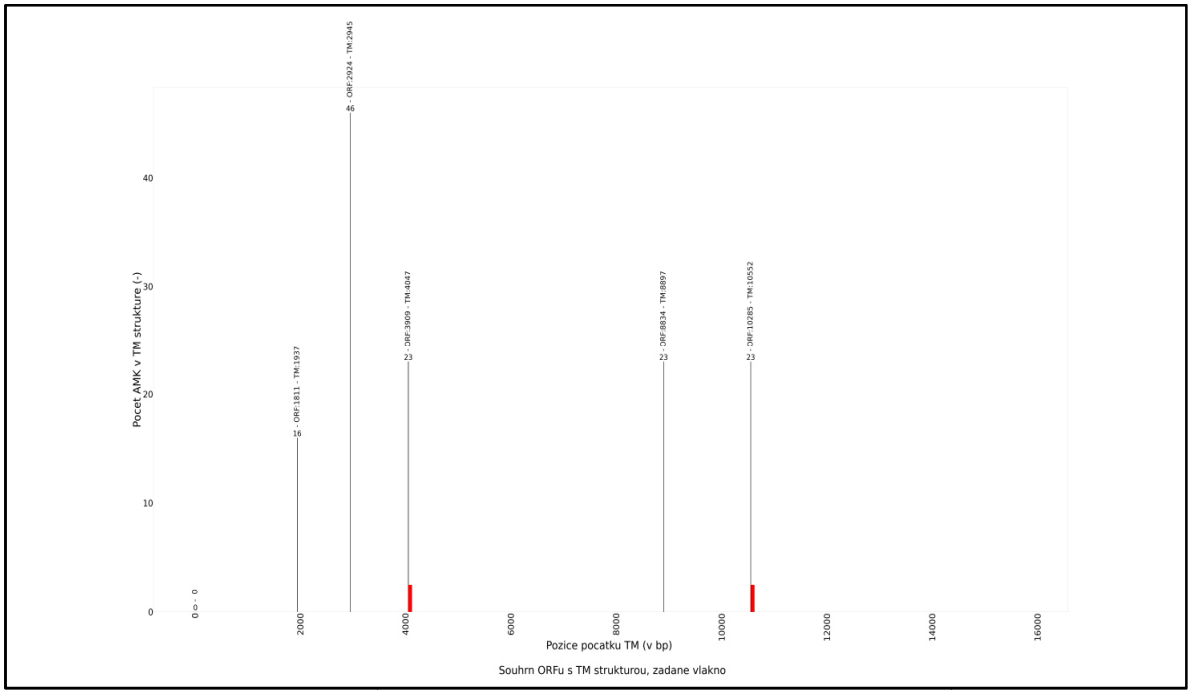
Úseky kódující CC domény:

- 4140 až 4263 bp,
- 5090 až 5166 bp,
- 5375 až 5436 bp,
- 5686 až 5717 bp,
- 10693 až 10843 bp,
- 11931 až 12021 bp.

Pozice ω-místa pro připojení GPI kotvy:

– 5720 bp.

Porovnání predikce TM domén a reálného rozložení kódování TM domén ve vyšetřovaném vláknu DNA – viz obrázek 5.18 níže (červeně jsou pozice reálně přítomných úseků DNA kódujících TM domény):



Obr. 5.18: Porovnání předpovědí TM struktur a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.18 je rozsah vlákna DNA v párech bazí (bp), na svislé ose počet aminokyselinových zbytků v dané predikované TM doméně. Úsečky kolmé na vodorovnou osu v obrázku 5.18 udávají pozici začátku predikované kódované TM struktury.

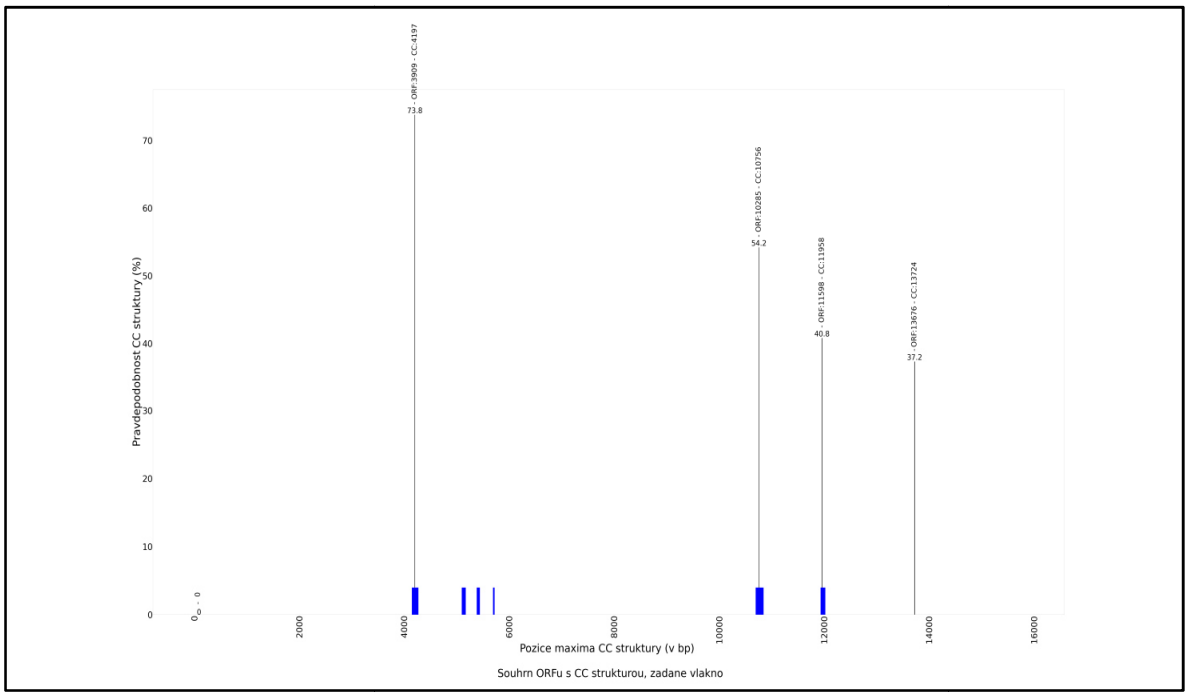
Na obrázku 5.18 výše je patrné, že ke shodě mezi predikcí a reálnou pozicí úseků kódujících TM sekundární struktury došlo dvakrát v případě třetí a páté predikce TM struktury celkem pěti predikcí.

Udané predikce, kde je shoda nebo téměř shoda s reálnou situací jsou:

– reálná poloha začátku TM struktury: 4047 bp, předpovězená: 4047 bp,
– reálná poloha začátku TM struktury: 10555 bp, předpovězená: 10552 bp.

Zbylé predikce jsou falešně pozitivní nálezy. Jedná se zde tedy o 3 falešně pozitivní nálezy z celkem 5ti predikcí. Není zde falešně negativní nález.

Porovnání predikce CC domén a reálného rozložení kódování CC domén ve vyšetřovaném vláknu DNA – viz obrázek 5.19 níže (modře jsou pozice reálně přítomných úseků DNA kódujících CC domény):



Obr. 5.19: Porovnání předpovědí CC struktur a reálné situace v úseku DNA

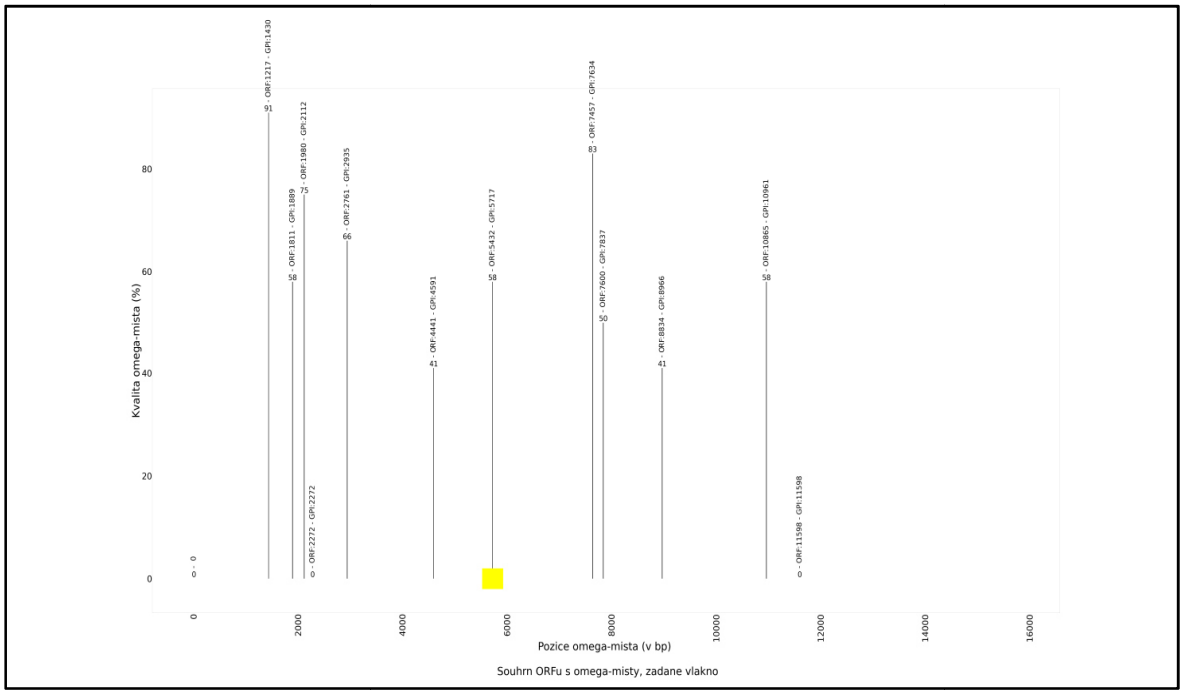
Na vodorovné ose obrázku 5.19 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je maximální pravděpodobnost dané predikované CC struktury. Úsečky kolmé na vodorovnou osu v obrázku 5.19 udávají pozici předpovědi CC struktury s nejvyšší pravděpodobností v rámci daného ORFu, kde byla predikována CC struktura.

Úspěšnost, resp. neúspěšnost predikování CC struktury v rámci reálného umístění kódování CC domén:

- 4140 až 4263 bp - předpověď: 4197 bp,
- 5090 až 5166 bp - nepředpovězeno, falešně negativní výsledek,
- 5375 až 5436 bp - nepředpovězeno, falešně negativní výsledek,
- 5686 až 5717 bp - nepředpovězeno, falešně negativní výsledek,
- 10693 až 10843 bp - předpověď: 10756 bp,
- 11931 až 12021 bp - předpověď: 11958 bp.

Falešně pozitivní výsledek: 13724 bp.

Porovnání predikce ω-místa pro GPI modifikaci a reálného umístěníω-místa ve vyšetřovaném vláknu DNA – viz obrázek 5.20 níže (žlutou značkou je reálná pozice ω-místa):



Obr. 5.20: Porovnání předpovědi ω-místa a reálné situace v úseku DNA

Na vodorovné ose obrázku 5.20 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je kvalita ω-místa v procentech. Úsečky kolmé na vodorovnou osu v obrázku 5.20 udávají pozici předpovědi ω-místa pro GPI modifikaci.

Reálná pozice ω-místa pro připojení GPI kotvy:
– 5720 bp.

Tato pozice není predikována. Nejbližším predikovaným místem je pozice 5717 bp. Kvalita ω-místa je v této nejbližší predikci stanovena na 58 %.

Další predikce jsou falešně pozitivní nálezy. Jedná se celkem o 12 falešně pozitivních nálezů. Z nich 2 nemají určenou pozici a kvalitu ω-místa.

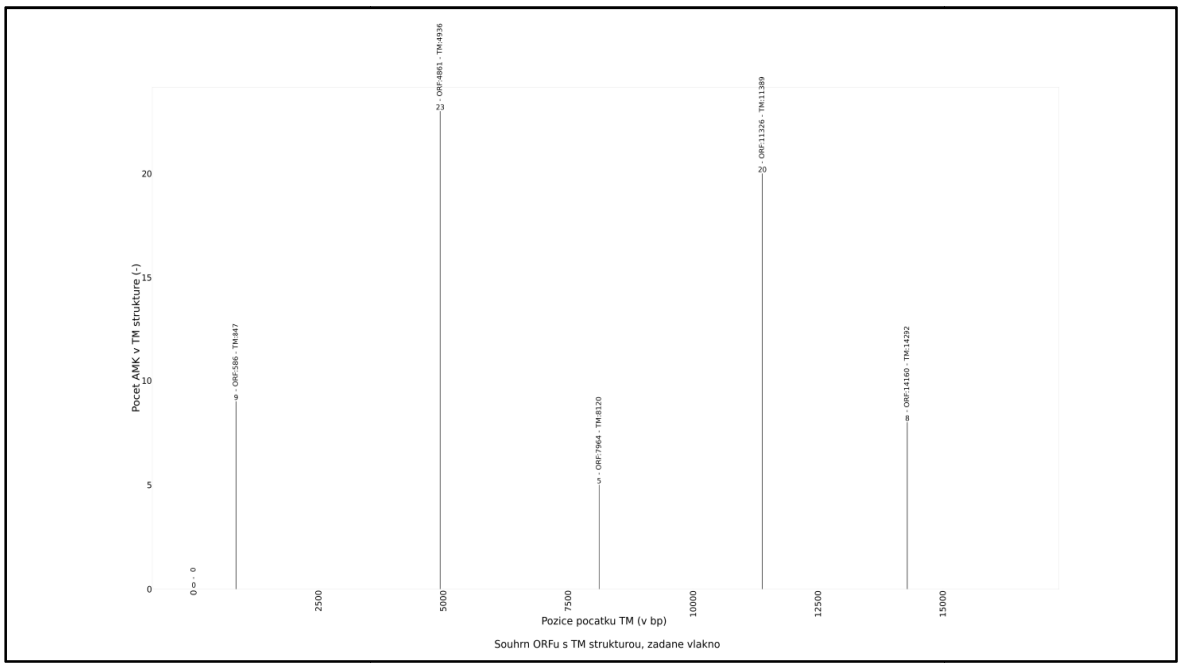
5.6 Příklad použití – náhodně vygenerované sekvence DNA

Skript byl nastaven na minimální délku ORFu 150 bp. Jako úsek DNA byl vybrán náhodně vygenerovaný úsek DNA dlouhý 16500 párů bazí (bp). Náhodná sekvence je s rovnoměrným rozložením pravděpodobnosti výskytu pro všechny baze DNA bez dalších podmínek.

Na obrázcích 5.21, 5.22 a 5.23 níže jsou zobrazeny výsledky predikce TM a CC domén a predikce ω-místa pro připojení GPI kotvy ve zmíněné náhodné sekvenci. V tomto úseku DNA samozřejmě nejsou anotovány žádné geny. Nelze tedy predikce porovnávat s

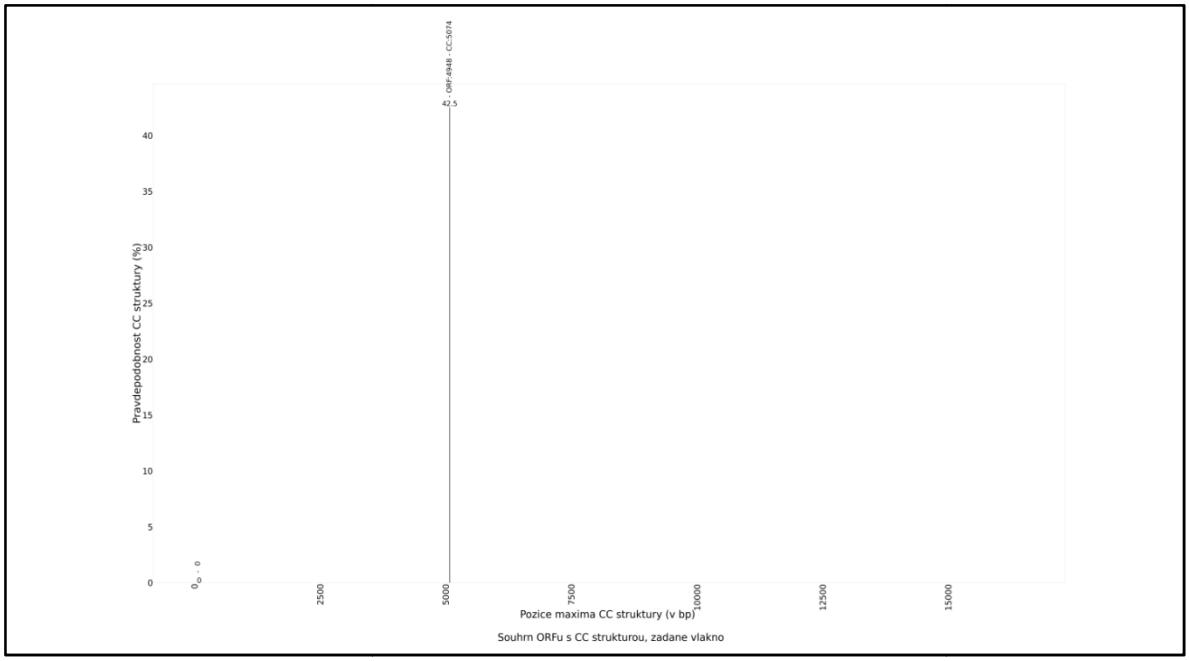
reálným, známým rozložením úseků kódujících TM a CC domény a ω-místa pro GPI modifikaci.

Obrázky 5.21, 5.22 a 5.23 níže jsou pouze ilustrativní. Pro tyto obrázky byly vybrány predikce s přibližně průměrným zastoupením předpovězených struktur. Bylo vygenerováno celkem 30 náhodných sekvencí o délce 16500 bp. Pro souhrn dat z nich slouží tabulka 5.1 níže.



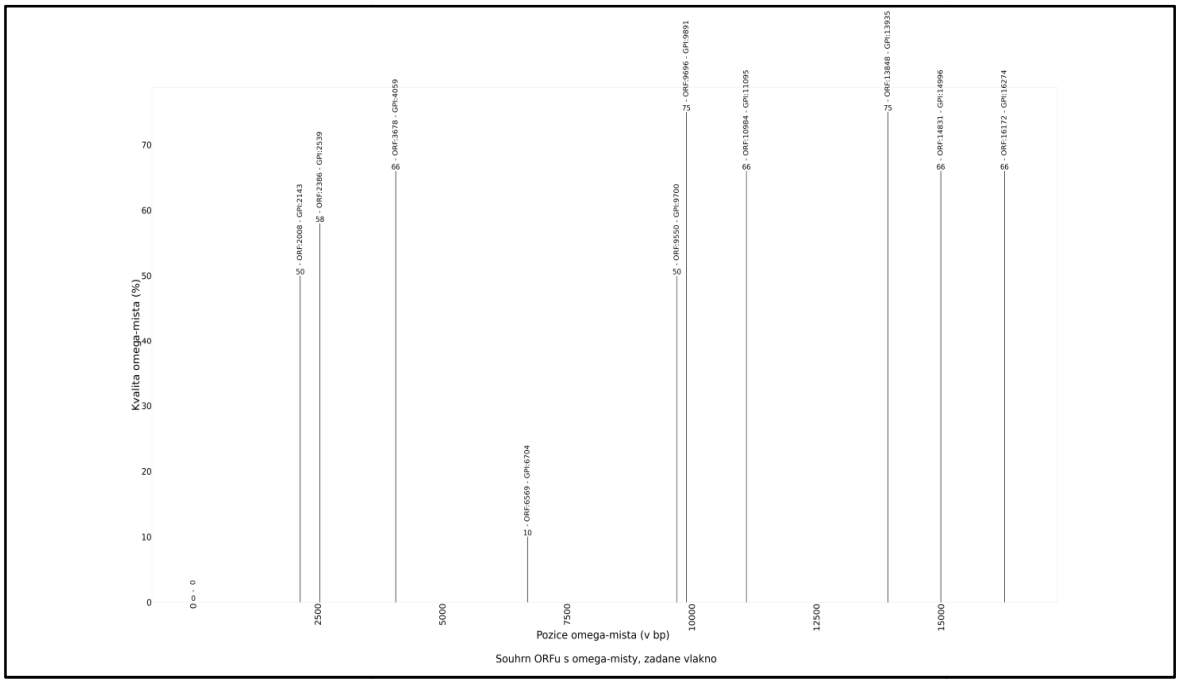
Obr. 5.21: Predikce TM domén v náhodné sekvenci DNA

Na vodorovné ose obrázku 5.21 je rozsah vlákna DNA v párech bazí (bp), na svislé ose počet aminokyselinových zbytků v dané predikované TM doméně. Úsečky kolmé na vodorovnou osu v obrázku 5.21 udávají pozici začátku predikované kódované TM struktury.



Obr. 5.22: Predikce CC domén v náhodné sekvenci DNA

Na vodorovné ose obrázku 5.22 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je maximální pravděpodobnost dané predikované CC struktury. Úsečky kolmé na vodorovnou osu v obrázku 5.22 udávají pozici předpovědi CC struktury s nejvyšší pravděpodobností v rámci daného ORFu, kde byla predikována CC struktura.



Obr. 5.23: Predikce ω-míst pro GPI modifikaci v náhodné sekvenci DNA

Na vodorovné ose obrázku 5.23 je rozsah vlákna DNA v párech bazí (bp), na svislé ose je kvalita ω-místa v procentech. Úsečky kolmé na vodorovnou osu v obrázku 5.23 udávají pozici předpovědi ω-místa pro GPI modifikaci.

Tabulka 5.1 níže slouží k přehledu dat ze 30ti náhodně vygenerovaných sekvencí.

Tabulka 5.1: Souhrn pro 30 náhodných sekvencí

Číslo náhodné sekvence	**TM**	**CC**	**GPI (all)**	**GPI (val)**	**GPI (0)**
1	5	0	8	5	3
2	6	1	10	10	0
3	3	1	9	5	4
4	6	2	9	8	1
5	4	2	9	7	2
6	5	2	15	9	6
7	4	1	19	16	3
8	7	2	9	8	1
9	7	0	9	4	5
10	2	1	6	4	2
11	5	1	7	6	1
12	5	0	10	8	2
13	7	0	6	5	1
14	7	1	13	11	2
15	1	0	7	7	0
16	6	1	10	7	3
17	5	0	12	8	4
18	5	1	9	7	2
19	4	2	10	7	3
20	6	0	10	5	5
21	4	0	8	5	3
22	8	1	17	12	5
23	5	1	4	3	1
24	7	1	11	6	5
25	8	1	14	12	2
26	3	1	5	5	0
27	2	1	14	7	7
28	4	1	10	8	2
29	2	2	16	8	8
30	5	2	5	1	4
průměr:	4,93	0,97	10,03	7,13	2,90
směrodatná odchylka:	1,84	0,72	3,65	3,00	2,06
variační koeficient:	0,372	0,743	0,364	0,421	0,709
průměr na 10 kbp:	2,99	0,59	6,08	4,32	1,76
směr. odch. (na 10 kbp):	1,11	0,44	2,21	1,82	1,25

K tabulce 5.1 výše:

- sloupec označen TM: počet predikovaných TM struktur v dané sekvenci,
- sloupec označen CC: počet predikovaných CC struktur v dané sekvenci,
- sloupec označen GPI (all): počet všech ORFů s ω-místy v dané sekvenci,
- sloupec označen GPI (val): počet ORFů v dané sekvenci s predikovanými ω-místy se stanovenou pozicí a kvalitou,
- sloupec označen GPI (0): počet ORFů v dané sekvenci s predikovanými ω-místy bez stanovené pozice a kvality,
- předposlední řádek udává, kolik predikovaných struktur připadá v průměru na 10 kbp,
- poslední řádek udává směrodatnou odchylku na 10 kbp.

Tabulka 5.2 níže slouží k přehledu dat z reálných sekvencí.

Tab. 5.2: Přehled dat z reálných sekvencí

	TM	**CC**	**GPI (all)**	**GPI (val)**	**GPI (0)**	**bp**
člověk	10	6	19	12	7	16570
myš domácí	10	6	24	13	11	19192
kur domácí	5	5	6	4	2	6948
luskoun ostrovní	6	5	14	10	4	17754
kaloň vábivý	5	4	12	10	2	15785
průměr na 10 kbp:	5,00	3,86	9,62	6,35	3,27	
směr.odch. (na 10 kbp):	1,54	1,71	1,99	0,61	1,56	
variační koeficient (pro 10 kbp):	0,310	0,440	0,210	0,100	0,480	

K tabulce 5.2 výše:

- sloupec označen TM: počet predikovaných TM struktur v dané sekvenci,
- sloupec označen CC: počet predikovaných CC struktur v dané sekvenci,
- sloupec označen GPI (all): počet všech ORFů s ω-místy v dané sekvenci,
- sloupec označen GPI (val): počet ORFů v dané sekvenci s predikovanými ω-místy se stanovenou pozicí a kvalitou,
- sloupec označen GPI (0): počet ORFů v dané sekvenci s predikovanými ω-místy bez stanovené pozice a kvality.

V tabulce 5.3 níže je uveden soupis počtu predikcí TM domén, jejich reálného počtu a počtu TM domén, u nichž byly predikce úspěšné. Pro počet predikcí, reálný počet a pro počet úspěšně stanovených domén je uveden přepočet na úsek vlákna DNA o délce 10 kbp.

Tab. 5.3: Uvedené počty predikcí, reálného počtu a úspěšných predikcí TM domén

	TM predicted	**TM real**	**TM hits**	**bp**
člověk	10	2	2	16570
myš domácí	10	2	2	19192
kur domácí	5	2	2	6948
luskoun ostrovní	6	2	2	17754
kaloň vábivý	5	2	2	15785
průměr na 10 kbp:	5,00	1,50	1,50	

K tabulce 5.3 výše:

- sloupec označen TM predicted: počet predikovaných TM domén programem,
- sloupec označen TM real: počet reálně kódovaným TM domén v zadaném vlákně DNA,
- sloupec označen TM hits: počet úspěšně predikovaných TM domén.

V tabulce 5.4 níže je uveden soupis počtu predikcí CC domén, jejich reálného počtu a počtu CC domén, u nichž byly predikce úspěšné. Pro počet predikcí, reálný počet a pro počet úspěšně stanovených domén je uveden přepočet na úsek vlákna DNA o délce 10 kbp.

Tab. 5.4: Uvedené počty predikcí, reálného počtu a úspěšných predikcí CC domén

	CC predicted	**CC real**	**CC hits**	**bp**
člověk	6	8	5	16570
myš domácí	6	8	6	19192
kur domácí	5	8	5	6948
luskoun ostrovní	5	8	5	17754
kaloň vábivý	4	6	3	15785
průměr na 10 kbp:	3,86	5,76	3,61	

K tabulce 5.4 výše:

- sloupec označen CC predicted: počet predikovaných CC domén programem,
- sloupec označen CC real: počet reálně kódovaným CC domén v zadaném vlákně DNA,
- sloupec označen CC hits: počet úspěšně predikovaných CC domén.

V tabulce 5.5 níže je uveden soupis počtu predikcí ω-míst pro GPI modifikaci, jejich reálného počtu a počtu ω-míst pro GPI modifikaci u nichž byly predikce úspěšné. Pro počet predikcí, reálný počet a pro počet úspěšně stanovených ω-míst je uveden přepočet na úsek vlákna DNA o délce 10 kbp.

Tab. 5.5: Uvedené počty predikcí, reálného počtu a úspěšných predikcí GPI ω-míst

	GPI (all) predicted	**GPI real**	**GPI hits**	**bp**
člověk	19	1	1	16570
myš domácí	24	1	0	19192
kur domácí	6	1	0	6948
luskoun ostrovní	14	1	0	17754
kaloň vábivý	12	1	0	15785
průměr na 10 kbp:	9,62	0,75	0,12	

K tabulce 5.5 výše:

- sloupec označen GPI (all) predicted: počet všech programem přesně predikovaných ω-míst nebo ORFů s ω-místem bez přesnějšího určení,
- sloupec označen GPI real: počet reálně přítomnýchω-míst v zadaném vlákně DNA,
- sloupec označen GPI hits: počet úspěšně predikovaných ω-míst.

6 Diskuse

Jak již bylo uvedeno výše v kapitole 5 Výsledky, lze konstatovat, že se v této práci nejspíše jedná o první systematické vyhledávání antivirového genu *bst-2* a příbuzných protivirových genů pomocí *de novo* predikce z genomových dat. Prozatím byly k dispozici jen programy pro samostatné predikce TM nebo CC domén nebo ω-míst pro GPI modifikaci. Nebyl zde však dosud nástroj, který by byl určen pro predikce těchto struktur dohromady na rozsáhlejším úseku DNA.

Z velké části je výsledná úspěšnost predikcí programu u 5ti vybraných organismů shrnuta v tabulkách 5.3, 5.4 a 5.5. V tabulce 5.1 jsou shrnuty výsledky pro 30 náhodně vygenerovaných sekvencí. Jedná se o sekvence s rovnoměrně rozloženou pravděpodobností výskytu každé ze čtyř bazí DNA bez dalších podmínek nebo omezení. Získané hodnoty v tabulkách 5.3, 5.4 a 5.5 vycházejí z poměrně malého počtu prováděných predikcí, tj. co se týče posuzovaných reálných lokusů se známou lokalizací kódovaných struktur. Přesto je možné předpokládat, že dostatečně vystihují míru schopnosti programu predikovat tyto struktury.

V tabulce 5.3 jsou uvedené počty predikcí, reálného počtu a úspěšných predikcí TM (transmembránových) domén. Program se snaží stanovit začátek kódované TM domény. Kódovaná TM doména v uvedených případech byla vždy kódována jako jeden celek v rámci jednoho exonu. Za nedůležitější parametry považuje autor průměrné hodnoty vztažené na 10 kbp. V pěti vyšetřovaných lokusech se známým rozmístěním kódovaných TM domén byly vždy přítomny dvě kódované TM domény. Tyto domény byly též v případě každého lokusu detekovány programem, pokud bereme za správný výsledek drobnou odchylku od přesného začátku kódované TM domény.

Přesněji tyto odchylky kolísají ve svém rozsahu od 3 bp (1 kodon, 1 aminokyselina ve výsledné sekvenci) až po 9 bp (3 aminokyseliny ve výsledné sekvenci). Celkem se v pěti lokusech nacházelo 10 kódovaných TM domén. Z těchto deseti TM domén bylo přibližně předpovězeno 5. Jedenkrát byla odchylka 9 bp, jedenkrát 6 bp a třikrát 3 bp. S výjimkou jedné se tyto omyly týkaly druhé predikované TM domény (ve směru od 5‘ konce).

Pokud zanedbáme tyto drobné odchylky, pak v každém lokusu přítomné 2 kódované TM domény byly detekovány vždy, čili zde nebyl falešně negativní případ predikce. K těmto úspěšným predikcím bylo vždy vytvořeno několik neúspěšných predikcí, tj. falešně pozitivních predikcí.

Ve zkoumaných lokusech se známým kódováním domén a ω-míst byly vždy přítomny 2 kódované TM domény. Tyto domény jsou 2 i v případě jednoho vlákna o délce necelých 7 kbp. V případě zadaných delších vláken je logické, že program predikuje i další, neexistující TM domény, čili poskytnutí delší sekvence jen se dvěma známými TM doménami je pro program mírně „podvod“ v podstatě uměle snižující míru úspěšnosti programu. Pokud toto vezmeme v potaz, pak zde dochází k dobré shodě mezi tabulkami 5.1 a 5.3 – dle tabulky 5.3 na 10 kbp připadá 5 predikovaných TM domén, z toho 2 jsou správně predikovány a zbylé 3

predikované TM domény odpovídají poměrně přesně třem TM doménám na 10 kbp v náhodné sekvenci.

Z 20ti kódovaných aminokyselin je hydrofobních celkem 10 aminokyselin, což je rovná polovina. Za hydrofobní aminokyseliny jsou počítány tyto: glycin, alanin, valin, leucin, isoleucin, cystein, methionin, fenylalanin, tryptofan a prolin. V použitých celkem 5ti lokusech se známým umístěním kódovaných TM domén bylo 10 kódovaných TM domén. Každá z kódovaných TM domén měla délku 23 aminokyselinových zbytků. Podle dosavadních výsledků (zachycených i na obrázcích 5.2, 5.6, 5.10, 5.14, 5.18) pro jednu TM doménu procházející jedenkrát cytoplasmatickou membránou je obvykle potřeba 23 aminokyselinových zbytků. Dle podkapitoly 2.2.5.1 predikční program zahrnuje do celé TM domény i tzv. vršky, zakončení o délce 5 aminokyselin, které obklopují jádro transmembránové šroubovice o délce mezi 5ti až 25ti aminokyselinových zbytků, což má zohlednit TM doménu o celkové délce 15 až 35 aminokyselinových zbytků [18]. Vzhledem k tomu, že správně predikované TM domény v našem případě měly vždy 23 aminokyselinových zbytků, poté na jejich jádro připadá 13 aminokyselinových zbytků. Dle tabulky 5.1 vychází na 10 kbp náhodné sekvence 2,99 TM domény. Pravděpodobnost zařazení hydrofobní aminokyseliny (při rovnoměrné pravděpodobnosti pro všechny aminokyseliny) je ½. Pokud bychom požadovali, aby jádro TM domény bylo tvořeno souvislou sekvencí 13ti hydrofobních aminokyselin, poté bychom na 10 kbp při jednoduchém počítání dle výrazu:

$$\left(\frac{1}{2}\right)^{13} \times \left(\frac{10^4}{3}\right) \simeq 0{,}407$$

získali uvedených cca 0,407 takovéto sekvence z náhodné sekvence 10 kbp DNA.

Práce predikčního programu je však výrazně složitější. Predikční program používá i topologické znalosti. Např. segment, který by normálně nebyl označen za transmembránový z důvodu slabé hydrofobnosti, může být stále predikován jako TM úsek, pokud to okolní topogenní signály podporují. [18]

Přesto je zřejmě patrné, že predikčnímu programu může stačit i relativně náhodné výraznější seskupení hydrofobních aminokyselin k deklarování daného úseku DNA jako úseku kódujícího TM doménu.

V tabulce 5.4 jsou uvedené počty predikcí, reálného počtu a úspěšných predikcí CC (coiled-coil) domén. Program se zde snaží stanovit pozici (ve výsledném vlákně aminokyselinu) s nejvyšší dosaženou pravděpodobností predikce CC domény. Kódovaná CC doména byla vždy rozdělena do několika exonů, čili program se zde snaží predikovat tyto části CC domény. Za nedůležitější parametry opět považuje autor průměrné hodnoty vztažené na 10 kbp. Z tabulky 5.4 je zřejmé, že predikovaných CC „domén“ je méně, nežli se ve skutečnosti ve zkoumaných vláknech vyskytuje. Predikovaných úseků domén je 3,86, nacházejících se je 5,76 (na 10 kbp). Predikované úseky domén také nebyly vždy úspěšné. Úspěšně bylo predikovaných 3,61 z 3,86 úseků CC tomén (na 10 kbp). Z uvedeného též vyplývá, že se zde vyskytují jak falešně pozitivní tak falešně negativní predikce.

Program nedokázal predikovat především kratší úseky DNA kódující CC doménu. Nejdelší úsek, který nebyl predikován, měl délku 133 bp, nejkratší úsek 16 bp, aritmetický průměr délky těchto nepredikovaných úseků je 55 bp. Naproti tomu nejkratší správně predikovaný úsek měl délku 40 bp, nejdelší úsek 213 bp a aritmetický průměr správně predikovaných úseků je 113 bp.

Predikce CC domén je přesnější nežli predikce TM domén nebo ω-míst pro GPI modifikaci. Dle tabulky 5.4 a hodnot pro 10 kbp je úspěšnost správné predikce vůči všem přítomným CC doménám: (3,61 / 5,76)·100 % ≈ 62,7 %. Predikovat CC doménu je zřejmě snadnější nežli je tomu u zbývajících dvou případů. Jednak musí být splněn požadavek určité minimální délky daného peptidu. Např. v této práci pro predikce použitá knihovna DeepCoil 2.0.1 vyžaduje minimální délku peptidu 20 aminokyselinových zbytků. Predikované ve vlákně DNA mohou být i kratší úseky, ty však musejí být součástí delšího peptidu, který predikční program již prozkoumá. Také v daném peptidu se musí vyskytovat periodická sekvence sedmi aminokyselinových zbytků (tzv. heptad) z nichž dva musejí být zbytky hydrofobních aminokyselin na přesně daných pozicích. Nemusí se jednat o jednu stálou sekvenci aminokyselin. Na stejné pozici heptadu ale musí být aminokyselina s podobným postranním řetězcem, tj. s podobnými vlastnostmi. Z těchto důvodů je také predikce úseku DNA kódující část CC domény méně úspěšná u kratších úseků DNA.

V tabulce 5.1 je uvedeno, že v náhodně vygenerované sekvenci program detekuje jen asi 0,6 domény nebo její části na 10 kbp. Dle tabulky 5.4 vychází těchto falešně pozitivních predikcí jen 0,25 na 10 kbp. Tato hodnota ale vychází z poměrně malého počtu zpracovaných lokusů.

V tabulce 5.5 je uveden soupis počtu predikcí ω-míst pro GPI (glycosyl-phosphatidylinositol) modifikaci, jejich reálného počtu a počtu ω-míst pro GPI modifikaci u nichž byly predikce úspěšné. Program se zde snaží predikovat ω-místo, tj. jednu peptidovou vazbu, kde dojde odstřižení C-konce peptidu a připojení GPI kotvy. Za nedůležitější parametry opět považuje autor průměrné hodnoty vztažené na 10 kbp. Dle tabulky 5.5 vychází úspěšnost predikcí nízká, tj. jen 0,12 správné predikce na 10 kbp oproti reálně přítomným 0,75 ω-míst na 10 kbp. Jednotlivé odchylky od reálné pozice ω-místa v pěti predikcích jsou 0, 408, 12, 9 a 3 bp. Průměrná hodnota z nich je 86,4 bp. Pokud by postačovalo predikovat ω-místo přibližně, např. až po uvedených 12 bp (včetně), tj. „jen“ o 4 aminokyseliny dále, pak by se tabulka 5.5 přetransformovala v tabulku 6.1níže.

Tab. 6.1: Upravená úspěšnost predikcí ω-míst při umožnění mírných odchylek

	GPI (all) predikce	**GPI real**	**GPI hits**	**bp**
člověk	19	1	1	16570
myš domácí	24	1	0	19192
kur domácí	6	1	1	6948
luskoun ostrovní	14	1	1	17754
kaloň vábivý	12	1	1	15785
průměr na 10 kbp:	**9,62**	**0,75**	**0,65**	

V tabulce 6.1 výše vychází přítomnost reálných ω-míst pro GPI modifikaci 0,75 na 10 kbp, úspěšných predikcí 0,65 na 10 kbp. Dle tohoto „úhlu pohledu“ jsou pak tyto predikce relativně úspěšné: (0,65 / 0,75)·100 % ≈ 87 %.

Platí také obdobné jako u predikcí TM domén (viz výše). V zadaných lokusech bylo vždy přítomno jen jedno ω-místo bez ohledu na celou délku vlákna. Je zřejmé, že program takto predikuje ω-místa i v jiných úsecích DNA. Čili se zvětšující se délkou vlákna DNA, kde je přítomno jen jedno ω-místo, „uměle“ klesá relativní úspěšnost programu.

Dle výše uvedeného je zcela přesná predikce ω-místa pro GPI modifikaci poměrně obtížná. Pro tento účel byla použita natrénovaná neuronová síť GPI-SOM (Self Organizing Map) – viz kap. 2.2.9.3. Nízká úspěšnost zcela přesných predikcí (viz tabulka 5.5) je způsobena již jen tím, že je zde snaha najít de facto jen jeden konkrétní bod, jednu peptidovou vazbu, v sekvenci polypeptidu a následně tak místa v DNA. Natrénovaná neuronová síť používá sice znalost obecné stavby polypeptidu v celkové délce několika desítek aminokyselinových zbytků kolem předpokládaného ω-místa, to ale nepostačuje ke vždy zcela přesnému určení jedné konkrétní peptidové vazby. Pro ilustraci též např. uvedeme, že bylo experimentálně prokázáno, že i jednobodová mutace může stačit na přeměnu signálu pro přiřazení GPI kotvy v transmembránovou doménu [62].

Získané výsledky (viz kapitola 5 Výsledky) jsou dány použitou metodou hledání (viz kapitola 4 Metody). Pro posouzení funkčnosti programu bylo použito 5 lokusů od různých zástupců obratlovců a 30 náhodně vygenerovaných sekvencí. V každém z těchto 5ti lokusů se od 5‘ konce nacházel gen pro tetherin a poté TM-CC(aT) gen. Gen pro tetherin obsahoval vždy postupně za sebou (od 5‘ konce) kódovanou TM doménu (která byla vždy kódována vcelku v rámci jednoho exonu), poté CC doménu rozdělenou do několika exonů a na závěr kódoval ω-místo pro GPI modifikaci. TM-CC(aT) gen kódoval vždy ty samé struktury stejným způsobem s výjimkou ω-místa, které nikdy nekódoval.

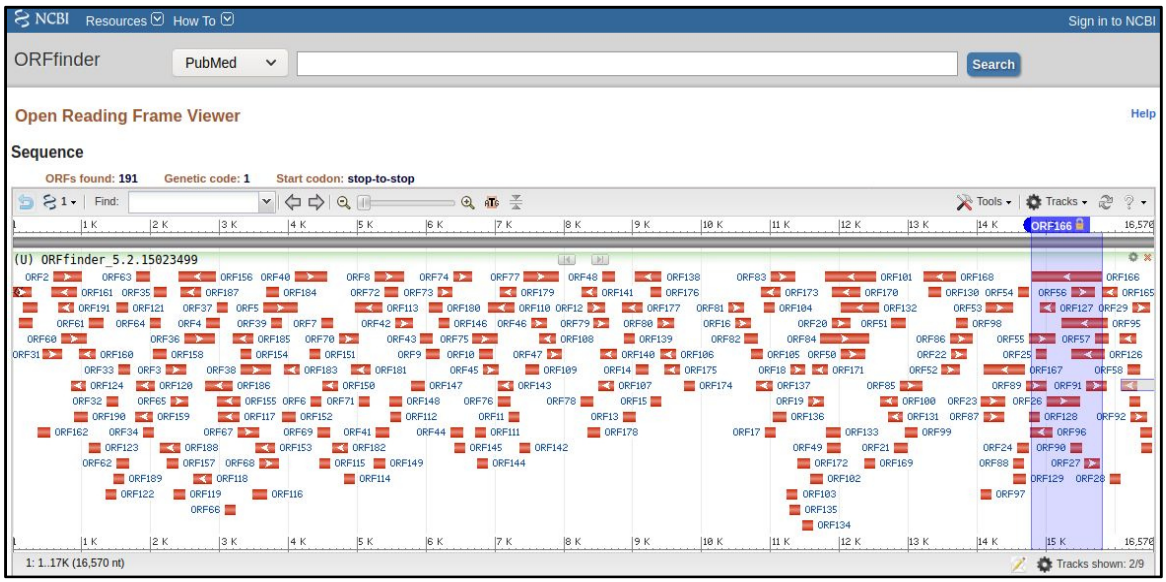
Tyto geny jsou v této práci hledány dle sekundárních struktur, které kódují (TM a CC domény, ω-místo). Tento způsob je u genů pro tehterin nutný, neboť konkrétní sekvence nukleotidů v genu a aminokyselinových zbytků následně v proteinu je u tetherinů extrémně variabilní a pro vznik tetherinu jen stačí zachovat 3 domény. Pro každou z těchto domén není tedy striktně „předepsána“ jedna nebo několik málo konkrétních sekvencí. Tetheriny jsou pod pozitivní, tj. rozrůzňující, selekcí, neboť jsou v evolučním střetu s virovými proteiny, které se snaží efekt tetherinů blokovat [76]. Naprostou většinu ostatních genů lze vyhledávat dle sekvenční podobnosti.

TM-CC(aT) gen je vyhledáván tedy současně s genem pro tetherin. Je to sice gen neznámé funkce, ale má podobné domény a zřejmě společného předka v evoluci.

V této práci jsou též analyzovány ORFy (otevřené čtecí rámce) a nikoliv přímo exony. Je to z důvodu náročnější predikce přítomnosti (začátku a konce) exonu a pro případný neznámý úsek DNA je i vhodnější začít analýzou ORFů.

V použitém úseku lidské DNA (kapitola 5.1) o délce 16570 bp je 191 ORFů (při minimální délce ORFu 150 bp a vyřazování ORFů i s jen jedním neznámým znakem) – viz ilustrativní obrázek 6.1. Dříve bylo nutno všechny tyto ORFy ručně prozkoumat. Výsledný

nástroj této práce tuto činnost automatizuje a v HTML stránce výsledek znázorní v textové a grafické formě.



Obr. 6.1: Ilustrace počtu ORFů vyhledaných v nástroji ORFfinder na stránkách NCBI (dle [77])

V případě takto vyšších počtů ORFů lze předpokládat, že se v predikcích vyskytnou falešně pozitivní i falešně negativní případy.

V celé práci je používána jedna definice ORFu a to definice „č. 2“ dle [78] a dle kapitoly 2.2.1.

V programu je také použit limit 32 kbp jako maximální délka zpracovávaného vlákna DNA. Toto omezení je zavedeno především kvůli možnostem graficky znázornit souhrn ORFů s TM nebo CC doménami nebo ω-místy pro GPI modifikaci.

Uživatel může zvolit jednu ze tří možností ošetření ambiguitního kódu. Jsou očíslovány 1, 2 a 3. První varianta ve skriptu provádí nahrazení místa s neznámým znakem náhodně zvolenou aminokyselinou (do výběru aminokyseliny nejsou použity STOP kodony).

Druhá varianta provádí vystřižení a zahození všech neznámých znaků (jeden neznámý znak může zastupovat i více pozic ve vlákně, čili i tak předem dochází již k posunu čtecího rámce).

Poslední, třetí varianta vyřazuje z dalšího zpracování všechny ORFy s byť i jen jedním neznámým znakem.

Nejvhodnější variantou ošetření ambiguitního kódu je varianta třetí, tj. zahození ORFu s byť i jen jedním neznámým znakem. Proto při zpracování dat pro kapitolu 5 Výsledky byla použita jen tato varianta.

Jak bylo řečeno na začátku kapitoly 5, za výsledek této práce lze považovat vytvořený program, také HTML stránku jím vygenerovanou a dále použití výsledků (zachycené v HTML stránce) běhu programu na zadané sekvenci DNA.

Důležitým výstupem práce je nástroj ve formě programu napsaném v programovacím jazyce Python 3.8.5. Tento program není určen pro celo-genomové prohlížení sekvencí DNA, ale je to pomocník pro jemné dohledávání struktur v předem vybraném kratším úseku DNA v (obvykle nastaveném) rozsahu 320 bp po 32000 bp. Např celá délka lidského genomu je přibližně 3,2 miliard bp.

V další práci by program mohl být rozšířen o možnost delší sekvenci rozdělit na kratší zpracovatelné úseky (zvl. vzhledem ke grafickému zobrazení souhrn ORFů s TM nebo CC doménami nebo ω-místy pro GPI modifikaci). Tyto úseky zpracovat a zobrazit pro ně výsledky zvlášť, možná i s volbou zpracovat jen některé úseky.

Vzhledem k výsledkům zachyceným v kapitole 5 a jejich interpretaci v této kapitole lze konstatovat, že přes některá omezení lze tento program považovat za potenciálně užitečného pomocníka.

7 Závěr

Tato práce si kladla za cíl vytvořit program pro vyhledávání jedné skupiny genů, tzv. tetherinů, u obratlovců. Jedná se o vyhledávání jedné genové rodiny, jejímž reprezentantem je např. gen *bst2*. Přičemž vyhledávat tetheriny lze v podstatě jen na základě jimi kódovaných proteinových struktur. Program neměl být určen k celo-genomovým analýzám, ale je určen jako pomocník pro prohledávání kratších úseků genomu (v uvedené práci typicky do 20 kbp). I v těchto kratších úsecích genetického kódu byla manuální práce poměrně zdlouhavá. Za výstup programu byla zvolena forma HTML stránky.

Vytvořený program lze za takového pomocníka považovat a lze předpokládat jeho potenciální další využití. Dle dříve uvedeného lze považovat zadání práce za splněné.

Seznam použité literatury

[1] LE TORTOREC, Anna, Suzanne WILLEY a Stuart J. D. NEIL. Antiviral Inhibition of Enveloped Virus Release by Tetherin/BST-2: Action and Counteraction. *Viruses* [online]. 2011, **3**(5), 520-540 [cit. 2021-02-12]. ISSN 1999-4915. Dostupné z: doi:10.3390/v3050520

[2] KRCHLÍKOVÁ, Veronika, Helena FÁBRYOVÁ, Tomáš HRON, et al. Antiviral Activity and Adaptive Evolution of Avian Tetherins. *Journal of Virology* [online]. 2020, **94**(12), e00416-20 [cit. 2021-02-12]. ISSN 0022-538X. Dostupné z: doi:10.1128/JVI.00416-20

[3] BLANCO-MELO, Daniel, Siddarth VENKATESH a Paul D. BIENIASZ. Origins and Evolution of tetherin , an Orphan Antiviral Gene. *Cell Host & Microbe* [online]. 2016, **20**(2), 189-201 [cit. 2021-02-12]. ISSN 19313128. Dostupné z: doi:10.1016/j.chom.2016.06.007

[4] Molecular biology of the cell / Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, Peter Walter ; with problems by John Wilson, Tim Hunt. -- Sixth edition. -- New York : GS Garland Science, Taylor & Francis Group, [2015]. -- ©2015. -- xxxiv, 1342, 34, 53, 1 stran : ilustrace (některé barevné) ; 28 cm ; ISBN 978-0-8153-4464-3

[5] Chapter 2: Protein Structure. *WESTERN OREGON UNIVERSITY* [online]. Monmouth, Oregon, U.S.A.: WESTERN OREGON UNIVERSITY, 2021 [cit. 2021-02-14]. Dostupné z: https://wou.edu/chemistry/courses/online-chemistry-textbooks/ch450-and-ch451-biochemistry-defining-life-at-the-molecular-level/chapter-2-protein-structure/

[6] Gpi Anchor Structure. *Czech Republic | Sigma-Aldrich* [online]. Darmstadt, Germany: Merck, 2021 [cit. 2021-02-13]. Dostupné z: https://www.sigmaaldrich.com/life-science/proteomics/post-translational-analysis/glycosylation/structures-symbols/gpi-anchor-structure.html

[7] DUBÉ, Mathieu, Mariana G BEGO, Catherine PAQUAY a Éric A COHEN. Modulation of HIV-1-host interaction: role of the Vpu accessory protein. *Retrovirology* [online]. 2010, **7**(1) [cit. 2021-02-14]. ISSN 1742-4690. Dostupné z: doi:10.1186/1742-4690-7-114

[8] YANG, H., J. WANG, X. JIA, et al. Structural insight into the mechanisms of enveloped virus tethering by tetherin. *Proceedings of the National Academy of Sciences* [online]. 2010, **107**(43), 18428-18432 [cit. 2021-02-14]. ISSN 0027-8424. Dostupné z: doi:10.1073/pnas.1011485107

[9] "ORFfinder" (https://www.ncbi.nlm.nih.gov/projects/gorf/). www.ncbi.nlm.nih.gov.

[10] Finding ORF of a Given Sequence. *VALUE @ Amrita* [online]. Kollam, Kerala, Indie: Amrita university, 2021 [cit. 2021-02-22]. Dostupné z: https://vlab.amrita.edu/?sub=3&brch=273&sim=1432&cnt=1

[11] Open reading frame. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-02-22]. Dostupné z: https://en.wikipedia.org/wiki/Open_reading_frame

[12] ORF Investigator: A New ORF finding tool combining Pairwise Global Gene Alignment. *Research Journal of Recent Sciences* [online]. 2012, **2012**(11), 32-35 [cit. 2021-02-22]. Dostupné z: http://www.isca.in/rjrs/archive/v1/i11/6.ISCA-RJRS-2012-339.pdf

[13] EMBOSS getorf. *EMBOSS* [online]. Cambridge, UK: EMBOSS: The European Molecular Biology Open Software Suite, 2002 [cit. 2021-02-22]. Dostupné z: http://emboss.sourceforge.net/apps/cvs/emboss/apps/getorf.html

[14] WOODCROFT, Ben J., Joel A. BOYD a Gene W. TYSON. OrfM: a fast open reading frame predictor for metagenomic data. *Bioinformatics* [online]. 2016, **32**(17), 2702-2703 [cit. 2021-02-22]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/btw241

[15] Efficient String Matching: An Aid to Bibliographic Search. *Programming Techniques* [online]. 1975 [cit. 2021-02-22]. Dostupné z: https://cr.yp.to/bib/1975/aho.pdf

[16] Orfipy: a fast and flexible tool for extracting ORFs. *BioRxiv* [online]. 2020 [cit. 2021-02-22]. Dostupné z: https://www.biorxiv.org/content/10.1101/2020.10.20.348052v1.full.pdf

[17] DEBER, C. M. TM Finder: A prediction program for transmembrane protein segments using a combination of hydrophobicity and nonpolar phase helicity scales. *Protein Science* [online]. **10**(1), 212-219 [cit. 2021-03-07]. ISSN 09618368. Dostupné z: doi:10.1110/ps.30301

[18] A hidden Markov model for predicting transmembrane helices in protein sequences. *Intelligent Systems for Molecular Biology (ISMB-98)* [online]. 1998, **1998** [cit. 2021-03-07]. Dostupné z: https://www.aaai.org/Papers/ISMB/1998/ISMB98-021.pdf

[19] TUSNADY, G. E. a I. SIMON. The HMMTOP transmembrane topology prediction server. *Bioinformatics* [online]. 2001, 17(9), 849-850 [cit. 2021-03-07]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/17.9.849

[20] KALL, L., A. KROGH a E. L. L. SONNHAMMER. An HMM posterior decoder for sequence feature prediction that includes homology information. *Bioinformatics* [online]. 2005, 21(Suppl 1), i251-i257 [cit. 2021-03-07]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/bti1014

[21] KALL, L., A. KROGH a E. L.L. SONNHAMMER. Advantages of combined transmembrane topology and signal peptide prediction--the Phobius web server. *Nucleic Acids Research* [online]. 2007, 35(Web Server), W429-W432 [cit. 2021-03-07]. ISSN 0305-1048. Dostupné z: doi:10.1093/nar/gkm256

[22] CAO, Baoqiang, Aleksey POROLLO, Rafal ADAMCZAK, Mark JARRELL a Jaroslaw MELLER. Enhanced recognition of protein transmembrane domains with prediction-based

structural profiles. *Bioinformatics* [online]. 2006, 22(3), 303-309 [cit. 2021-03-07]. ISSN 1460-2059. Dostupné z: doi:10.1093/bioinformatics/bti784

[23] ALTSCHUL, S. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* [online]. 25(17), 3389-3402 [cit. 2021-03-07]. ISSN 13624962. Dostupné z: doi:10.1093/nar/25.17.3389

[24] HILLER, K., A. GROTE, M. SCHEER, R. MUNCH a D. JAHN. PrediSi: prediction of signal peptides and their cleavage positions. *Nucleic Acids Research* [online]. 2004, 32(Web Server), W375-W379 [cit. 2021-03-07]. ISSN 0305-1048. Dostupné z: doi:10.1093/nar/gkh378

[25] RIEDMILLER, M. a H. BRAUN. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: *IEEE International Conference on Neural Networks* [online]. IEEE, 1993, s. 586-591 [cit. 2021-03-07]. ISBN 0-7803-0999-5. Dostupné z: doi:10.1109/ICNN.1993.298623

[26] ZELL, Andreas, Niels MACHE, Ralf HÜBNER, Günter MAMIER, Michael VOGT, Michael SCHMALZL a Kai-Uwe HERRMANN. SNNS (Stuttgart Neural Network Simulator). SKRZYPEK, Josef, ed. *Neural Network Simulation Environments* [online]. Boston, MA: Springer US, 1994, 1994, s. 165-186 [cit. 2021-03-07]. The Kluwer International Series in Engineering and Computer Science. ISBN 978-1-4613-6180-0. Dostupné z: doi:10.1007/978-1-4615-2736-7_9

[27] KÄLL, Lukas, Anders KROGH a Erik L.L SONNHAMMER. A Combined Transmembrane Topology and Signal Peptide Prediction Method. *Journal of Molecular Biology* [online]. 2004, 338(5), 1027-1036 [cit. 2021-03-07]. ISSN 00222836. Dostupné z: doi:10.1016/j.jmb.2004.03.016

[28] TUSNADY, G. E., Z. DOSZTANYI a I. SIMON. Transmembrane proteins in the Protein Data Bank: identification and classification. *Bioinformatics* [online]. 2004, 20(17), 2964-2972 [cit. 2021-03-07]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/bth340

[29] TUSNADY, G. E. PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank. *Nucleic Acids Research* [online]. 2004, 33(Database issue), D275-D278 [cit. 2021-03-07]. ISSN 1362-4962. Dostupné z: doi:10.1093/nar/gki002

[30] KOZMA, Dániel, István SIMON a Gábor E. TUSNÁDY. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. *Nucleic Acids Research* [online]. 2012, 41(D1), D524-D529 [cit. 2021-03-07]. ISSN 0305-1048. Dostupné z: doi:10.1093/nar/gks1169

[31] TUSNADY, G. E., L. KALMAR a I. SIMON. TOPDB: topology data bank of transmembrane proteins. *Nucleic Acids Research* [online]. 2007, 36(Database), D234-D239 [cit. 2021-03-07]. ISSN 0305-1048. Dostupné z: doi:10.1093/nar/gkm751

[32] DOBSON, László, Tamás LANGÓ, István REMÉNYI a Gábor E. TUSNÁDY. Expediting topology data gathering for the TOPDB database. *Nucleic Acids Research*

[online]. 2015, 43(D1), D283-D289 [cit. 2021-03-07]. ISSN 1362-4962. Dostupné z: doi:10.1093/nar/gku1119

[33] TUSNADY, G. E., L. KALMAR, H. HEGYI, P. TOMPA a I. SIMON. TOPDOM: database of domains and motifs with conservative location in transmembrane proteins. *Bioinformatics* [online]. 2008, 24(12), 1469-1470 [cit. 2021-03-07]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/btn202

[34] DOBSON, László, István REMÉNYI a Gábor E. TUSNÁDY. CCTOP: a Consensus Constrained TOPology prediction web server. *Nucleic Acids Research* [online]. 2015, 43(W1), W408-W412 [cit. 2021-03-07]. ISSN 0305-1048. Dostupné z: doi:10.1093/nar/gkv451

[35] DOBSON, László, István REMÉNYI a Gábor E. TUSNÁDY. The human transmembrane proteome. *Biology Direct* [online]. 2015, 10(1) [cit. 2021-03-07]. ISSN 1745-6150. Dostupné z: doi:10.1186/s13062-015-0061-x

[36] PETERSEN, Thomas Nordahl, Søren BRUNAK, Gunnar VON HEIJNE a Henrik NIELSEN. SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nature Methods* [online]. 2011, 8(10), 785-786 [cit. 2021-03-07]. ISSN 1548-7091. Dostupné z: doi:10.1038/nmeth.1701

[37] BERNSEL, A., H. VIKLUND, J. FALK, E. LINDAHL, G. VON HEIJNE a A. ELOFSSON. Prediction of membrane-protein topology from first principles. *Proceedings of the National Academy of Sciences* [online]. 2008, 105(20), 7177-7181 [cit. 2021-03-07]. ISSN 0027-8424. Dostupné z: doi:10.1073/pnas.0711151105

[38] TUSNÁDY, Gábor E. a István SIMON. Principles governing amino acid composition of integral membrane proteins: application to topology prediction 1 1Edited by J. Thornton. *Journal of Molecular Biology* [online]. 1998, **283**(2), 489-506 [cit. 2021-03-07]. ISSN 00222836. Dostupné z: doi:10.1006/jmbi.1998.2107

[39] SHEN, Hongbin, James J. CHOU a Bostjan KOBE. MemBrain: Improving the Accuracy of Predicting Transmembrane Helices. *PLoS ONE* [online]. 2008, 3(6) [cit. 2021-03-07]. ISSN 1932-6203. Dostupné z: doi:10.1371/journal.pone.0002399

[40] NUGENT, Timothy a David T JONES. Transmembrane protein topology prediction using support vector machines. *BMC Bioinformatics* [online]. 2009, 10(1) [cit. 2021-03-07]. ISSN 1471-2105. Dostupné z: doi:10.1186/1471-2105-10-159

[41] VIKLUND, Håkan a Arne ELOFSSON. OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. *Bioinformatics* [online]. 2008, 24(15), 1662-1668 [cit. 2021-03-07]. ISSN 1460-2059. Dostupné z: doi:10.1093/bioinformatics/btn221

[42] REYNOLDS, Sheila M., Lukas KÄLL, Michael E. RIFFLE, Jeff A. BILMES, William Stafford NOBLE a Burkhard ROST. Transmembrane Topology and Signal Peptide Prediction

Using Dynamic Bayesian Networks. *PLoS Computational Biology* [online]. 2008, 4(11) [cit. 2021-03-07]. ISSN 1553-7358. Dostupné z: doi:10.1371/journal.pcbi.1000213

[43] VIKLUND, Håkan a Arne ELOFSSON. Best α-helical transmembrane protein topology predictions are achieved using hidden Markov models and evolutionary information. *Protein Science* [online]. 2004, 13(7), 1908-1917 [cit. 2021-03-07]. ISSN 09618368. Dostupné z: doi:10.1110/ps.04625404

[44] KROGH, Anders, Björn LARSSON, Gunnar VON HEIJNE a Erik L.L SONNHAMMER. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes11Edited by F. Cohen. *Journal of Molecular Biology* [online]. 2001, 305(3), 567-580 [cit. 2021-03-07]. ISSN 00222836. Dostupné z: doi:10.1006/jmbi.2000.4315

[45] BAGOS, Pantelis G, Theodore D LIAKOPOULOS a Stavros J HAMODRAKAS. *BMC Bioinformatics* [online]. 7(1) [cit. 2021-03-07]. ISSN 14712105. Dostupné z: doi:10.1186/1471-2105-7-189

[46] RABINER, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* [online]. 77(2), 257-286 [cit. 2021-03-07]. ISSN 00189219. Dostupné z: doi:10.1109/5.18626

[47] LUPAS, A., M. VAN DYKE a J. STOCK. Predicting coiled coils from protein sequences. *Science* [online]. 1991, **252**(5009), 1162-1164 [cit. 2021-03-11]. ISSN 0036-8075. Dostupné z: doi:10.1126/science.252.5009.1162

[48] COILS version 2.2. *COILS* [online]. [cit. 2021-03-11]. Dostupné z: https://embnet.vital-it.ch/software/coils/COILS_doc.html

[49] DELORENZI, M. a T. SPEED. An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. *Bioinformatics* [online]. 2002, **18**(4), 617-625 [cit. 2021-03-11]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/18.4.617

[50] TRIGG, Jason, Karl GUTWIN, Amy E. KEATING, Bonnie BERGER a Ozlem KESKIN. Multicoil2: Predicting Coiled Coils and Their Oligomerization States from Sequence in the Twilight Zone. *PLoS ONE* [online]. 2011, **6**(8) [cit. 2021-03-11]. ISSN 1932-6203. Dostupné z: doi:10.1371/journal.pone.0023519

[51] BERGER, B., D. B. WILSON, E. WOLF, T. TONCHEV, M. MILLA a P. S. KIM. Predicting coiled coils by use of pairwise residue correlations. *Proceedings of the National Academy of Sciences* [online]. 1995, **92**(18), 8259-8263 [cit. 2021-03-11]. ISSN 0027-8424. Dostupné z: doi:10.1073/pnas.92.18.8259

[52] MCDONNELL, A. V., T. JIANG, A. E. KEATING a B. BERGER. Paircoil2: improved prediction of coiled coils from sequence. *Bioinformatics* [online]. 2006, **22**(3), 356-358 [cit. 2021-03-11]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/bti797

[53] LUDWICZAK, Jan, Aleksander WINSKI, Krzysztof SZCZEPANIAK, Vikram ALVA, Stanislaw DUNIN-HORKAWICZ a John HANCOCK. DeepCoil—a fast and accurate prediction of coiled-coil domains in protein sequences. *Bioinformatics* [online]. 2019, **35**(16), 2790-2795 [cit. 2021-03-11]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/bty1062

[54] GRUBER, M., J. SODING a A. N. LUPAS. REPPER--repeats and their periodicities in fibrous proteins. *Nucleic Acids Research* [online]. 2005, **33**(Web Server), W239-W243 [cit. 2021-03-11]. ISSN 0305-1048. Dostupné z: doi:10.1093/nar/gki405

[55] F. Chollet. Keras. https://github.com/fchollet/keras, 2015.

[56] BARTOLI, L., P. FARISELLI, A. KROGH a R. CASADIO. CCHMM_PROF: a HMM-based coiled-coil predictor with evolutionary information. *Bioinformatics* [online]. 2009, **25**(21), 2757-2763 [cit. 2021-03-11]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/btp539

[57] PIERLEONI, Andrea, Pier MARTELLI a Rita CASADIO. PredGPI: a GPI-anchor predictor. *BMC Bioinformatics* [online]. 2008, **9**(1) [cit. 2021-03-16]. ISSN 1471-2105. Dostupné z: doi:10.1186/1471-2105-9-392

[58] *Biological sequence analysis Probabilistic models of proteins and nucleic acids*. Seventh printing 2002. Cambridge: Cambridge University Press, 1998. ISBN 0 521 63971 3.

[59] CORTES, Corinna a Vladimir VAPNIK. Support-vector networks. *Machine Learning* [online]. 1995, **20**(3), 273-297 [cit. 2021-03-16]. ISSN 0885-6125. Dostupné z: doi:10.1007/BF00994018

[60] EISENHABER, Birgit, Peer BORK a Frank EISENHABER. Prediction of Potential GPI-modification Sites in Proprotein Sequences. *Journal of Molecular Biology* [online]. 1999, **292**(3), 741-758 [cit. 2021-03-16]. ISSN 00222836. Dostupné z: doi:10.1006/jmbi.1999.3069

[61] Kronegg J, Buloz D: Detection/prediction of GPI cleavage site (GPI-anchor) in a protein (DGPI).

[62] FANKHAUSER, N. a P. MASER. Identification of GPI anchor attachment signals by a Kohonen self-organizing map. *Bioinformatics* [online]. 2005, **21**(9), 1846-1852 [cit. 2021-03-16]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/bti299

[63] POISSON, Guylaine, Cedric CHAUVE, Xin CHEN a Anne BERGERON. FragAnchor: A Large-Scale Predictor of Glycosylphosphatidylinositol Anchors in Eukaryote Protein Sequences by Qualitative Scoring. *Genomics, Proteomics & Bioinformatics* [online]. 2007, **5**(2), 121-130 [cit. 2021-03-16]. ISSN 16720229. Dostupné z: doi:10.1016/S1672-0229(07)60022-9

[64] CHOU, Kuo-Chen a Hong-Bin SHEN. MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-

PSSM. *Biochemical and Biophysical Research Communications* [online]. 2007, **360**(2), 339-345 [cit. 2021-03-16]. ISSN 0006291X. Dostupné z: doi:10.1016/j.bbrc.2007.06.027

[65] GÍSLASON, Magnús Halldór, Henrik NIELSEN, José Juan ALMAGRO ARMENTEROS a Alexander Rosenberg JOHANSEN. Prediction of GPI-anchored proteins with pointer neural networks. *Current Research in Biotechnology* [online]. 2021, **3**, 6-13 [cit. 2021-03-16]. ISSN 25902628. Dostupné z: doi:10.1016/j.crbiot.2021.01.001

[66] HOCHREITER, Sepp a Jürgen SCHMIDHUBER. Long Short-Term Memory. *Neural Computation* [online]. 1997, **9**(8), 1735-1780 [cit. 2021-03-16]. ISSN 0899-7667. Dostupné z: doi:10.1162/neco.1997.9.8.1735

[67] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems – Volume 2, Curran Associates Inc., USA. pp. 3111–3119. http://dl.acm.org/citation.cfm?id=2999792.2999959.

[68] NIELSEN, Henrik, Jacob ENGELBRECHT, Søren BRUNAK a Gunnar Von HEIJNE. A Neural Network Method for Identification of Prokaryotic and Eukaryotic Signal Peptides and Prediction of their Cleavage Sites. *International Journal of Neural Systems* [online]. 2011, **08**(05n06), 581-599 [cit. 2021-03-16]. ISSN 0129-0657. Dostupné z: doi:10.1142/S0129065797000537

[69] AIRES-DE-SOUSA, J. a L. AIRES-DE-SOUSA. Representation of DNA sequences with virtual potentials and their processing by (SEQREP) Kohonen self-organizing maps. *Bioinformatics* [online]. 2003, **19**(1), 30-36 [cit. 2021-03-16]. ISSN 1367-4803. Dostupné z: doi:10.1093/bioinformatics/19.1.30

[70] EISENHABER, B., P. BORK a F. EISENHABER. Sequence properties of GPI-anchored proteins near the omega-site: constraints for the polypeptide binding site of the putative transamidase. *Protein Engineering Design and Selection* [online]. 1998, **11**(12), 1155-1161 [cit. 2021-03-16]. ISSN 1741-0126. Dostupné z: doi:10.1093/protein/11.12.1155

[71] Udenfriend,S. and Kodukula,K. (1995) Prediction of ω site in nascent precursor of glycosylphosphatidylinositol protein. Methods Enzymol., 250, 571–582.

[72] KODUKULA, K, L D GERBER, R AMTHAUER, L BRINK a S UDENFRIEND. Biosynthesis of glycosylphosphatidylinositol (GPI)-anchored membrane proteins in intact cells: specific amino acid requirements adjacent to the site of cleavage and GPI attachment. *Journal of Cell Biology* [online]. 1993, **120**(3), 657-664 [cit. 2021-03-16]. ISSN 0021-9525. Dostupné z: doi:10.1083/jcb.120.3.657

[73] GERBER, L.D., K KODUKULA a S UDENFRIEND. Phosphatidylinositol glycan (PI-G) anchored membrane proteins. Amino acid requirements adjacent to the site of cleavage and PI-G attachment in the COOH-terminal signal peptide. *Journal of Biological Chemistry*

[online]. 1992, **267**(17), 12168-12173 [cit. 2021-03-16]. ISSN 00219258. Dostupné z: doi:10.1016/S0021-9258(19)49819-0

[74] KYTE, Jack a Russell F. DOOLITTLE. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology* [online]. 1982, **157**(1), 105-132 [cit. 2021-03-16]. ISSN 00222836. Dostupné z: doi:10.1016/0022-2836(82)90515-0

[75] Genomika: analýza a algoritmy - Cvičení 1. *Bioinformatický server* [online]. Praha, 2019 [cit. 2021-03-24]. Dostupné z: http://bio.img.cas.cz/GAA2020/E1/

[76] DUGGAL, Nisha K. a Michael EMERMAN. Evolutionary conflicts between viruses and restriction factors shape immunity. Nature Reviews Immunology [online]. 2012, 12(10), 687-695 [cit. 2021-5-2]. ISSN 1474-1733. Dostupné z: doi:10.1038/nri3295

[77] Open Reading Frame Finder. National Center for Biotechnology Information [online]. Bethesda, USA: National Center for Biotechnology Information, National Library of Medicine, 2021 [cit. 2021-5-2]. Dostupné z: https://www.ncbi.nlm.nih.gov/orffinder/

[78] SIEBER, Patricia, Matthias PLATZER a Stefan SCHUSTER. The Definition of Open Reading Frame Revisited. *Trends in Genetics* [online]. 2018, **34**(3), 167-170 [cit. 2021-5-12]. ISSN 01689525. Dostupné z: doi:10.1016/j.tig.2017.12.009

[79] MIN, X. J., G. BUTLER, R. STORMS a A. TSANG. OrfPredictor: predicting protein-coding regions in EST-derived sequences. *Nucleic Acids Research* [online]. 2005, **33**(Web Server), W677-W680 [cit. 2021-5-12]. ISSN 0305-1048. Dostupné z: doi:10.1093/nar/gki394

[80] GitHub - Roleren/ORFik. *GitHub* [online]. 2021 [cit. 2021-2-22]. Dostupné z: https://github.com/Roleren/ORFik

[81] Tmhmm.py 1.3.1. *The Python Package Index (PyPI)* [online]. The Python Software Foundation, 2021 [cit. 2021-5-12]. Dostupné z: https://pypi.org/project/tmhmm.py/

[82] PyTMHMM 1.3.2. *The Python Package Index (PyPI)* [online]. The Python Software Foundation, 2021 [cit. 2021-5-12]. Dostupné z: https://pypi.org/project/pyTMHMM/

Příloha A: Příklad HTML stránky

TM, CC a GPI predikce:

Skript použitelný pro: standardní kód, kód archeí a rostlinných plastidů a kvasinkový alternativní jaderný kód.

dolní limit délky vstupní sekvence je (bp): **320**

horní limit délky vstupní sekvence je (bp): **32000**

nukleotidů (délka zadané NK) je (bp): **15785**

minimální délka ORFu je (bp): **150**

počet ORFů v zadaném vlákně je: **68**

počet ORFů v komplementárním vlákně je: **70**

celkový počet ORFů v obou vláknech je: **138**

počet ORFů s TM sekundární strukturou (zadané vlákno): **5**

počet ORFů s TM sekundární strukturou (komplementární vlákno): **4**

počet ORFů s CC sekundární strukturou (zadané vlákno): **4**

počet ORFů s CC sekundární strukturou (komplementární vlákno): **2**

práh pro detekci CC sekundární struktury - zadané vlákno (%): **10.0**

práh pro detekci CC sekundární struktury - komplementární vlákno (%): **10.0**

počet ORFů s GPI sekundární strukturou (zadané vlákno): **12**

počet ORFů s GPI sekundární strukturou (komplementární vlákno): **14**

počet ORFů jen s TM sekundární strukturou (zadané vlákno): **1**

počet ORFů jen s TM sekundární strukturou (komplementární vlákno): **1**

počet ORFů jen s CC sekundární strukturou (zadané vlákno): **1**

počet ORFů jen s CC sekundární strukturou (komplementární vlákno): **1**

počet ORFů jen s GPI sekundární strukturou (zadané vlákno): **9**

počet ORFů jen s GPI sekundární strukturou (komplementární vlákno): **11**

počet ORFů jen s TM a CC sekundární strukturou (zadané vlákno): **2**

počet ORFů jen s TM a CC sekundární strukturou (komplementární vlákno): **0**

počet ORFů jen s TM a GPI sekundární strukturou (zadané vlákno): **2**

počet ORFů jen s TM a GPI sekundární strukturou (komplementární vlákno): **2**

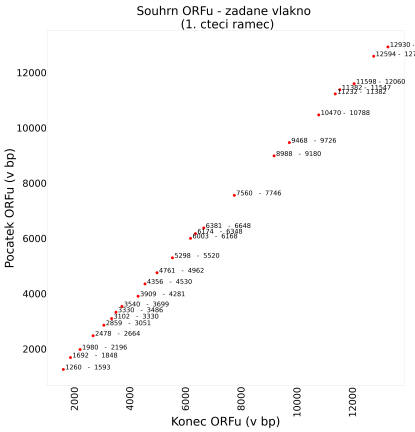
počet ORFů jen s CC a GPI sekundární strukturou (zadané vlákno): **1**

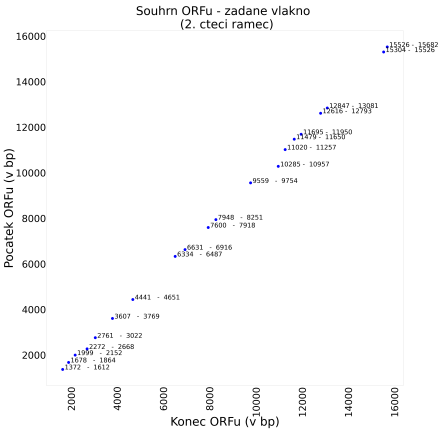
počet ORFů jen s CC a GPI sekundární strukturou (komplementární vlákno): **0**

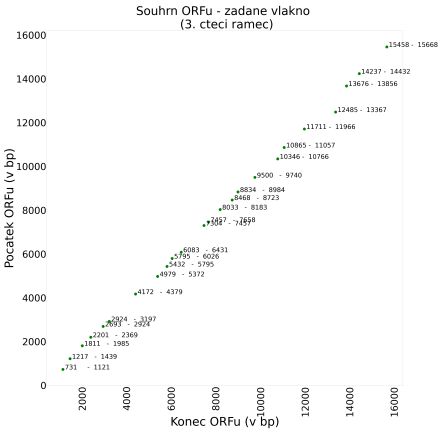
počet ORFů s TM, CC a GPI sekundární strukturou současně (zadané vlákno): **0**

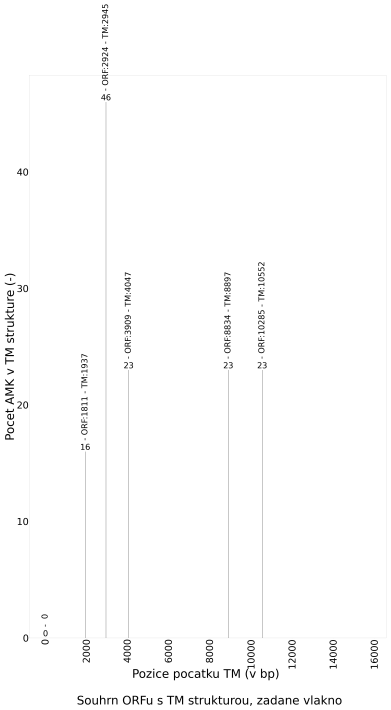
počet ORFů s TM, CC a GPI sekundární strukturou současně (komplementární vlákno): **1**

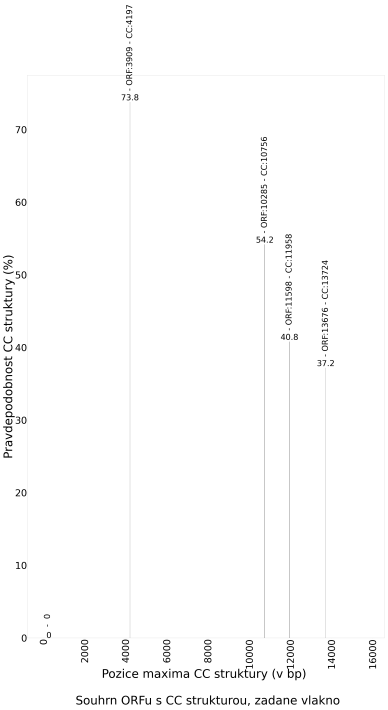
ORFy - zadané vlákno DNA:

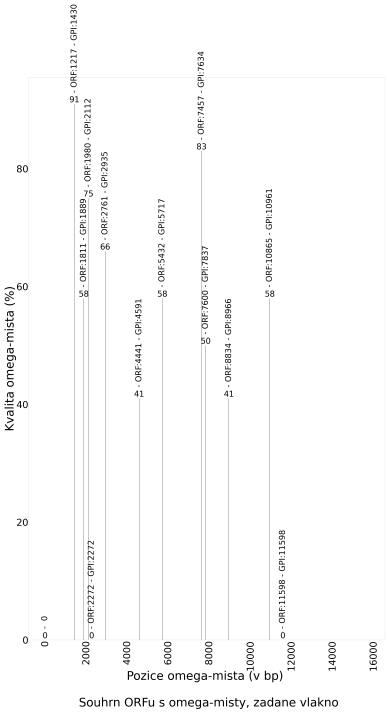


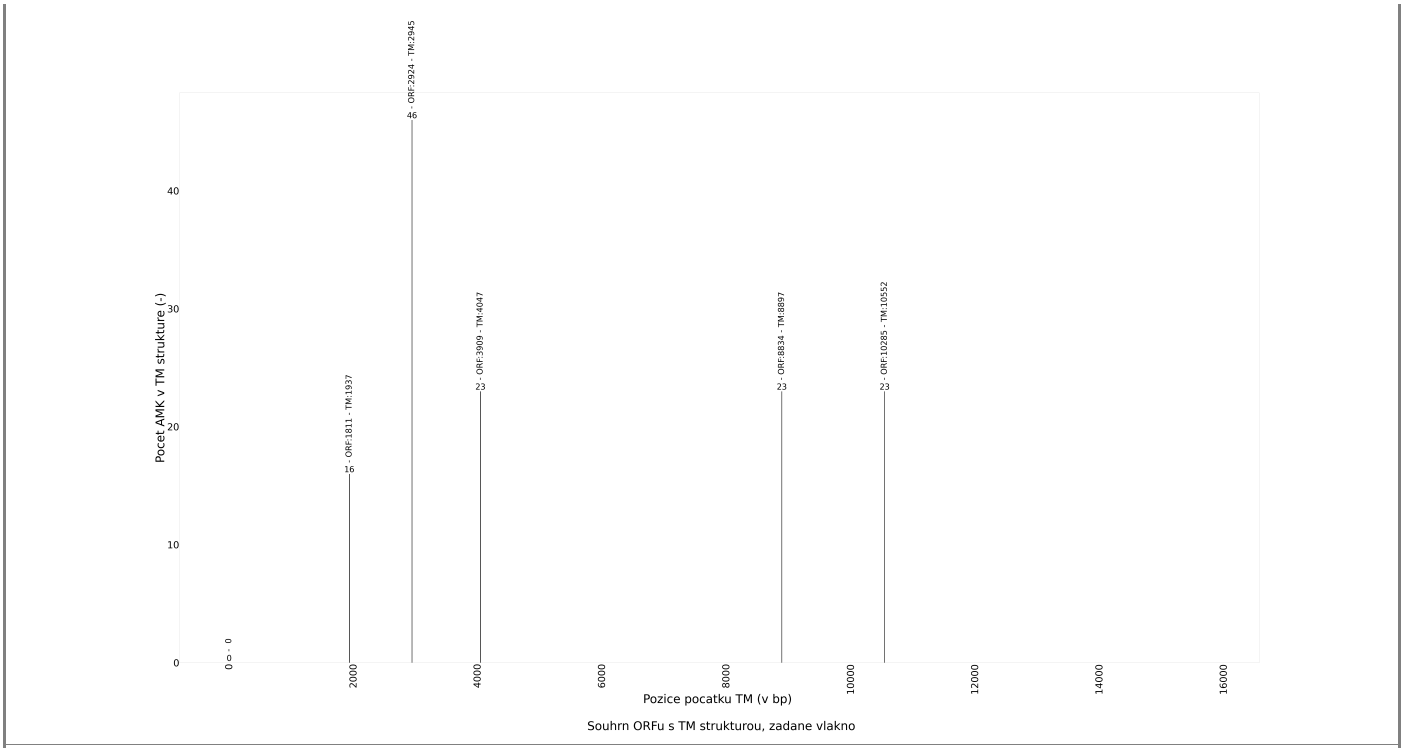




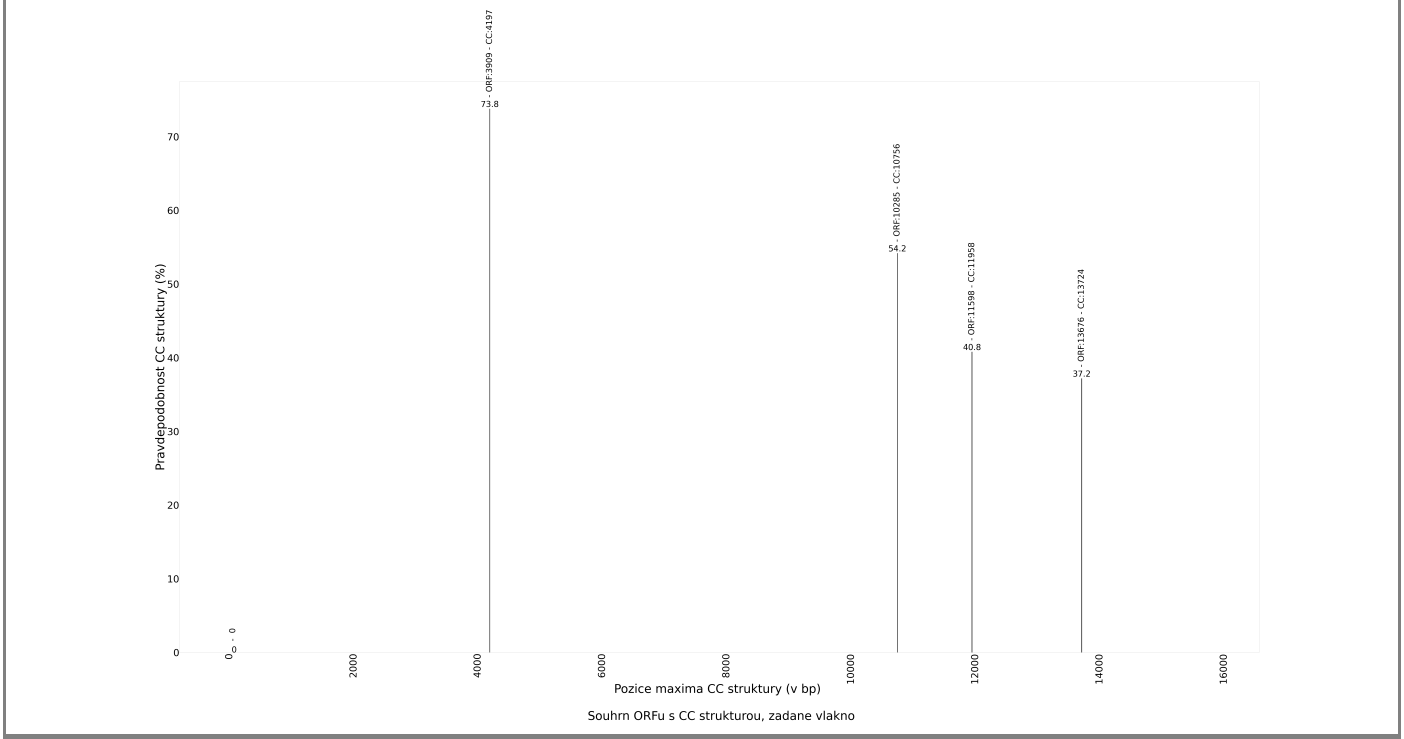




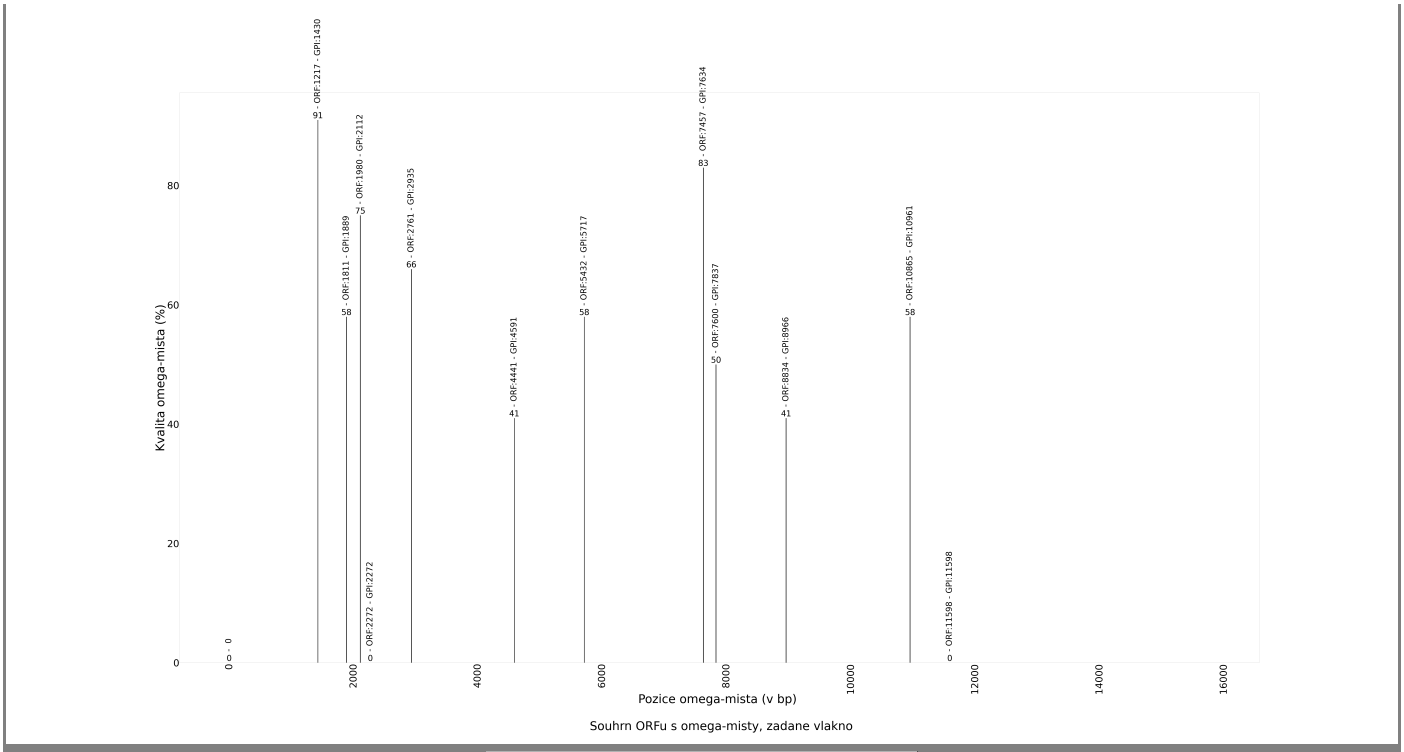




Souhrn ORFu s TM strukturou, zadane vlakno



Souhrn ORFu s CC strukturou, zadane vlakno

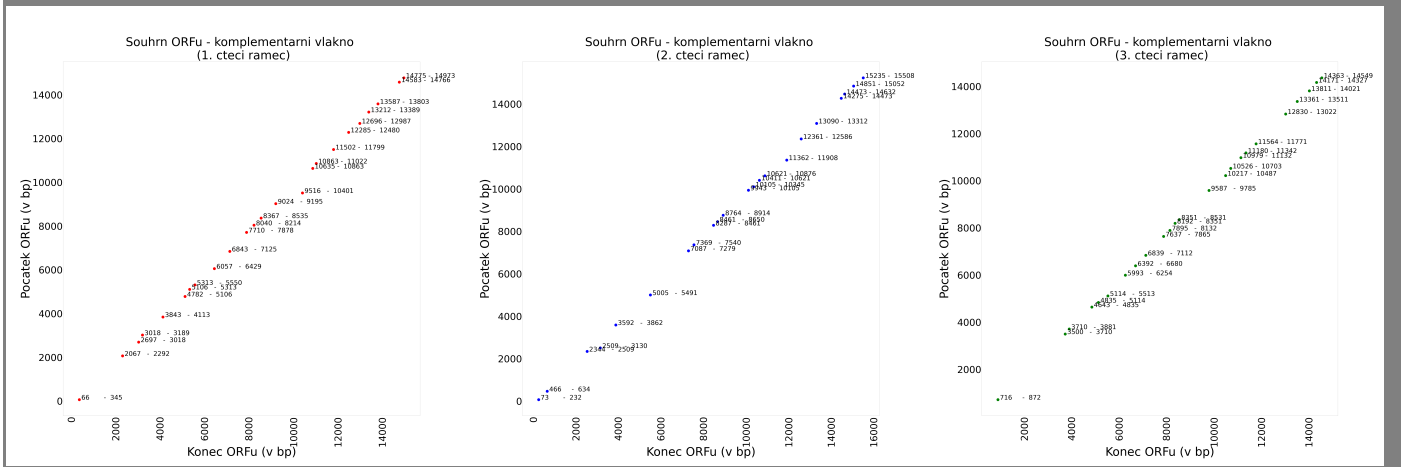


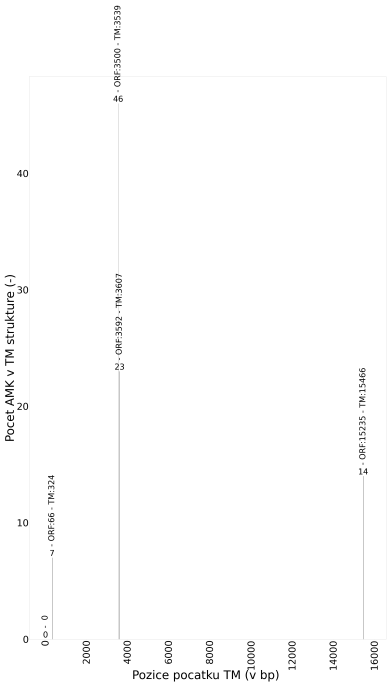
Souhrn ORFu s omega-misty, zadane vlakno

ORFy s TM strukturou - zadané vlákno	ORFy s CC strukturou - zadané vlákno	ORFy s GPI strukturou - zadané vlákno
1. čtecí rámec: ORF-pozice: [3909 4281]	**1. čtecí rámec:** ORF-pozice: [3909 4281] ORF-pozice: [11598 12060]	**1. čtecí rámec:** ORF-pozice: [1980 2196] ORF-pozice: [11598 12060]
2. čtecí rámec: ORF-pozice: [10285 10957]	**2. čtecí rámec:** ORF-pozice: [10285 10957]	**2. čtecí rámec:** ORF-pozice: [2272 2668] ORF-pozice: [2761 3022] ORF-pozice: [4441 4651] ORF-pozice: [7600 7918]
3. čtecí rámec: ORF-pozice: [1811 1985] ORF-pozice: [2924 3197] ORF-pozice: [8834 8984]	**3. čtecí rámec:** ORF-pozice: [13676 13856]	**3. čtecí rámec:** ORF-pozice: [1217 1430] ORF-pozice: [1811 1985] ORF-pozice: [5432 5795] ORF-pozice: [7457 7658] ORF-pozice: [8834 8984] ORF-pozice: [10865 11057]

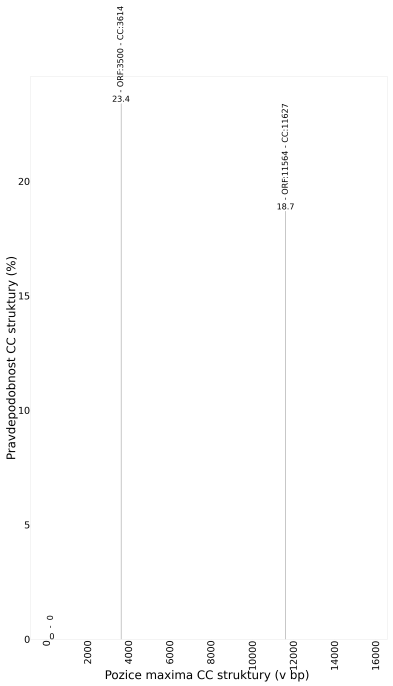
ORFy jen s TM strukturou - zadané vlákno - 1. čtecí rámec:	**ORFy jen s TM strukturou - zadané vlákno - 2. čtecí rámec:**	**ORFy jen s TM strukturou - zadané vlákno - 3. čtecí rámec:** ORF-pozice: [2924 3197]
ORFy jen s CC strukturou - zadané vlákno - 1. čtecí rámec:	**ORFy jen s CC strukturou - zadané vlákno - 2. čtecí rámec:**	**ORFy jen s CC strukturou - zadané vlákno - 3. čtecí rámec:** ORF-pozice: [13676 13856]
ORFy jen s GPI strukturou - zadané vlákno - 1. čtecí rámec: ORF-pozice: [1980 2196]	**ORFy jen s GPI strukturou - zadané vlákno - 2. čtecí rámec:** ORF-pozice: [2272 2668] ORF-pozice: [2761 3022] ORF-pozice: [4441 4651] ORF-pozice: [7600 7918]	**ORFy jen s GPI strukturou - zadané vlákno - 3. čtecí rámec:** ORF-pozice: [1217 1430] ORF-pozice: [5432 5795] ORF-pozice: [7457 7658] ORF-pozice: [10865 11057]
ORFy jen s TM a CC strukturou - zadané vlákno - 1. čtecí rámec: ORF-pozice: [3909 4281]	**ORFy jen s TM a CC strukturou - zadané vlákno - 2. čtecí rámec:** ORF-pozice: [10285 10957]	**ORFy jen s TM a CC strukturou - zadané vlákno - 3. čtecí rámec:**
ORFy jen s TM a GPI strukturou - zadané vlákno - 1. čtecí rámec:	**ORFy jen s TM a GPI strukturou - zadané vlákno - 2. čtecí rámec:**	**ORFy jen s TM a GPI strukturou - zadané vlákno - 3. čtecí rámec:** ORF-pozice: [1811 1985] ORF-pozice: [8834 8984]
ORFy jen s CC a GPI strukturou - zadané vlákno - 1. čtecí rámec: ORF-pozice: [11598 12060]	**ORFy jen s CC a GPI strukturou - zadané vlákno - 2. čtecí rámec:**	**ORFy jen s CC a GPI strukturou - zadané vlákno - 3. čtecí rámec:**
ORFy s TM, CC a GPI strukturou - zadané vlákno - 1. čtecí rámec:	**ORFy s TM, CC a GPI strukturou - zadané vlákno - 2. čtecí rámec:**	**ORFy s TM, CC a GPI strukturou - zadané vlákno - 3. čtecí rámec:**

ORFy - komplementární vlákno DNA:

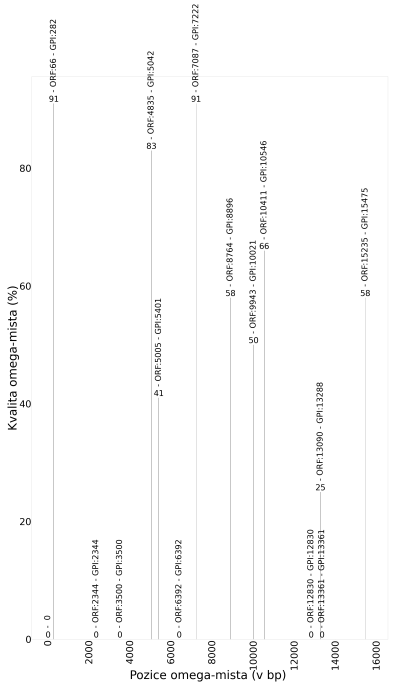




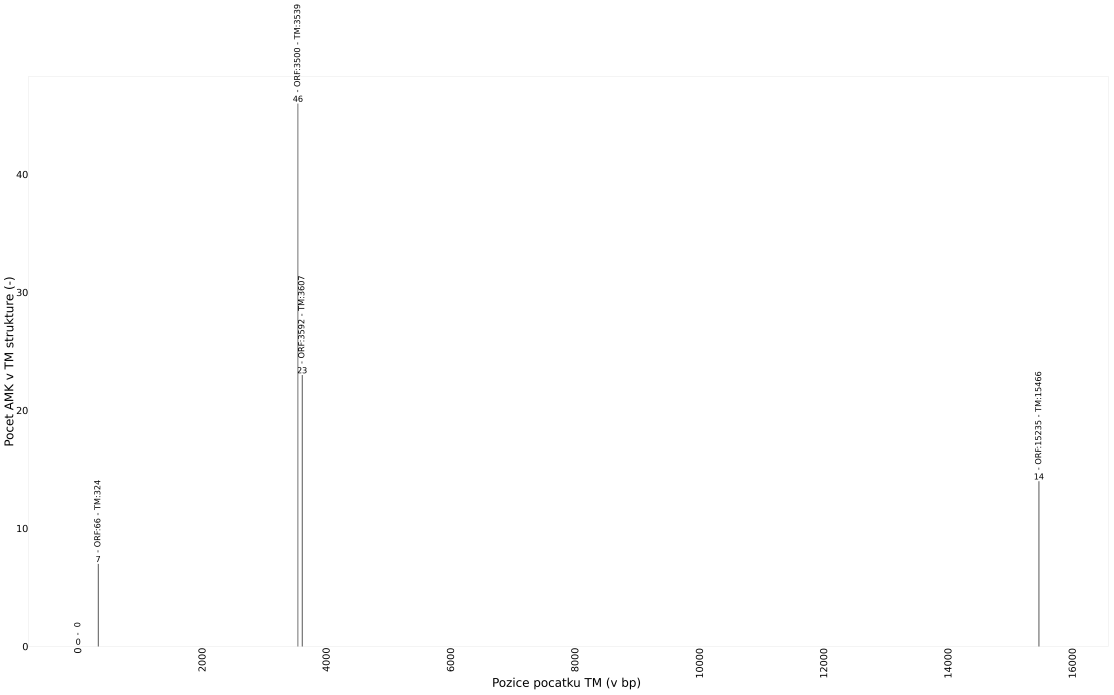
Souhrn ORFu s TM strukturou, komplementarni vlakno



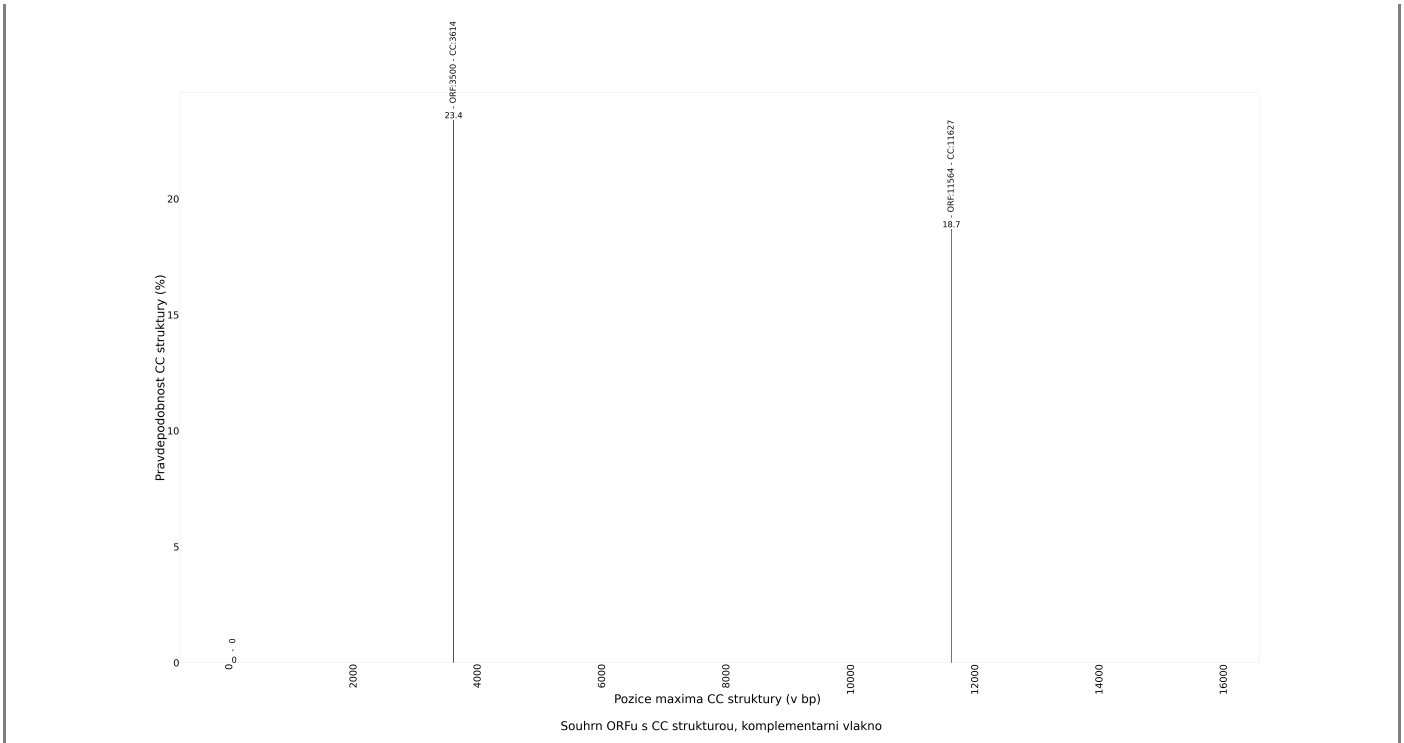
Souhrn ORFu s CC strukturou, komplementarni vlakno



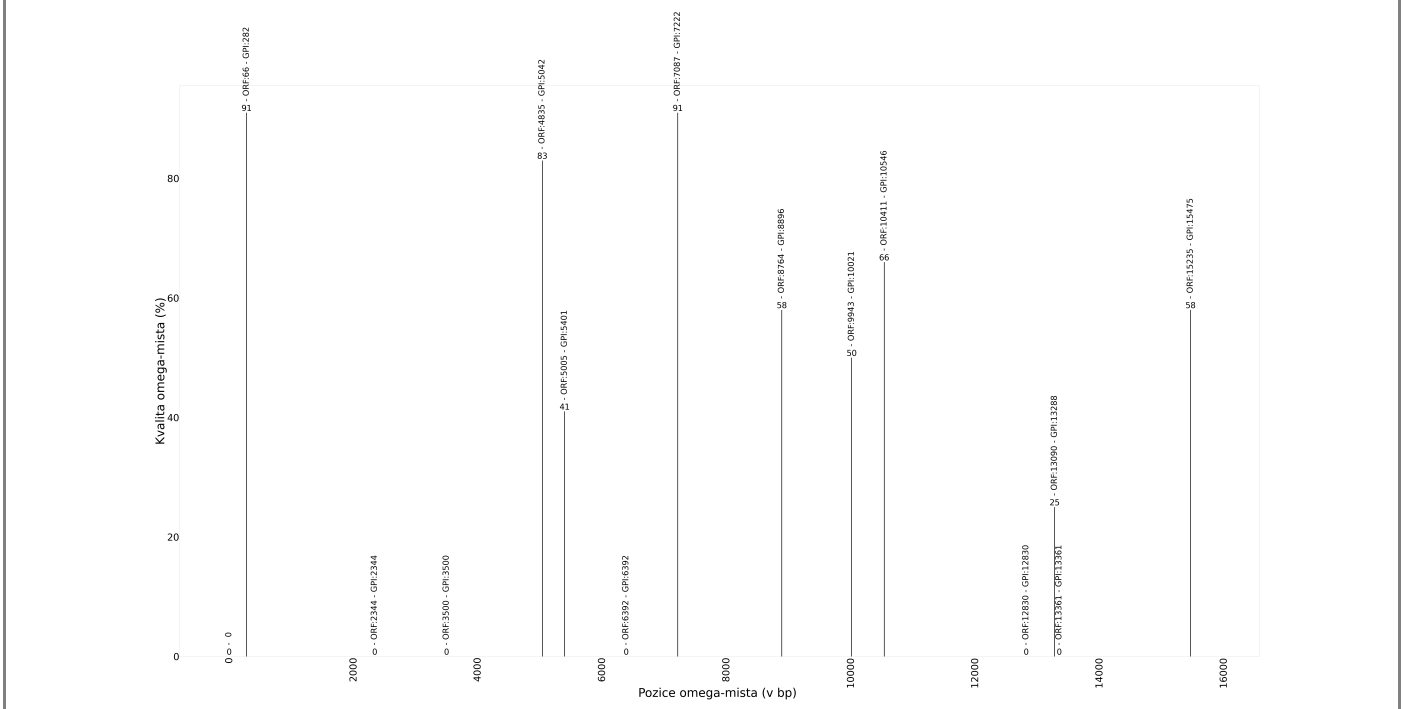
Souhrn ORFu s omega-misty, komplementarni vlakno



Souhrn ORFu s TM strukturou, komplementarni vlakno



Souhrn ORFu s CC strukturou, komplementarni vlakno



Souhrn ORFu s omega-misty, komplementarni vlakno

ORFy s TM strukturou - komplementární vlákno	ORFy s CC strukturou - komplementární vlákno	ORFy s GPI strukturou - komplementární vlákno
1. čtecí rámec: ORF-pozice: [66, 345]	**1. čtecí rámec:**	**1. čtecí rámec:** ORF-pozice: [66, 345]
2. čtecí rámec: ORF-pozice: [3592, 3862] ORF-pozice: [15235, 15508]	**2. čtecí rámec:**	**2. čtecí rámec:** ORF-pozice: [2344, 2509] ORF-pozice: [5005, 5491] ORF-pozice: [7087, 7279] ORF-pozice: [8764, 8914] ORF-pozice: [9943, 10105] ORF-pozice: [10411, 10621] ORF-pozice: [13090, 13312] ORF-pozice: [15235, 15508]
3. čtecí rámec: ORF-pozice: [3500, 3710]	**3. čtecí rámec:** ORF-pozice: [3500, 3710] ORF-pozice: [11564, 11771]	**3. čtecí rámec:** ORF-pozice: [3500, 3710] ORF-pozice: [4835, 5114] ORF-pozice: [6392, 6680] ORF-pozice: [12830, 13022] ORF-pozice: [13361, 13511]

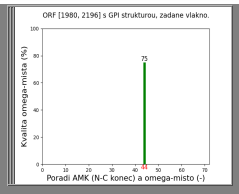
ORFy jen s TM strukturou - komplementární vlákno - 1. čtecí rámec:	**ORFy jen s TM strukturou - komplementární vlákno - 2. čtecí rámec:** ORF-pozice: [3592, 3862]	**ORFy jen s TM strukturou - komplementární vlákno - 3. čtecí rámec:**
ORFy jen s CC strukturou - komplementární vlákno - 1. čtecí rámec:	**ORFy jen s CC strukturou - komplementární vlákno - 2. čtecí rámec:**	**ORFy jen s CC strukturou - komplementární vlákno - 3. čtecí rámec:** ORF-pozice: [11564, 11771]
ORFy jen s GPI strukturou - komplementární vlákno - 1. čtecí rámec:	**ORFy jen s GPI strukturou - komplementární vlákno - 2. čtecí rámec:** ORF-pozice: [2344, 2509] ORF-pozice: [5005, 5491] ORF-pozice: [7087, 7279] ORF-pozice: [8764, 8914] ORF-pozice: [9943, 10105] ORF-pozice: [10411, 10621] ORF-pozice: [13090, 13312]	**ORFy jen s GPI strukturou - komplementární vlákno - 3. čtecí rámec:** ORF-pozice: [4835, 5114] ORF-pozice: [6392, 6680] ORF-pozice: [12830, 13022] ORF-pozice: [13361, 13511]
ORFy jen s TM a CC strukturou - komplementární vlákno - 1. čtecí rámec:	**ORFy jen s TM a CC strukturou - komplementární vlákno - 2. čtecí rámec:**	**ORFy jen s TM a CC strukturou - komplementární vlákno - 3. čtecí rámec:**
ORFy jen s TM a GPI strukturou - komplementární vlákno - 1. čtecí rámec: ORF-pozice: [66, 345]	**ORFy jen s TM a GPI strukturou - komplementární vlákno - 2. čtecí rámec:** ORF-pozice: [15235, 15508]	**ORFy jen s TM a GPI strukturou - komplementární vlákno - 3. čtecí rámec:**

ORFy jen s CC a GPI strukturou - komplementární vlákno - 1. čtecí rámec:	ORFy jen s CC a GPI strukturou - komplementární vlákno - 2. čtecí rámec:	ORFy jen s CC a GPI strukturou - komplementární vlákno - 3. čtecí rámec:
ORFy s TM, CC a GPI strukturou - komplementární vlákno - 1. čtecí rámec:	ORFy s TM, CC a GPI strukturou - komplementární vlákno - 2. čtecí rámec:	ORFy s TM, CC a GPI strukturou - komplementární vlákno - 3. čtecí rámec:
		ORF pozice: [3500, 3710]

jednotlivé ORFy (zadané vlákno):

ORF, zadané vlákno, 1. čtecí rámec - pozice:

[1980, 2196] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

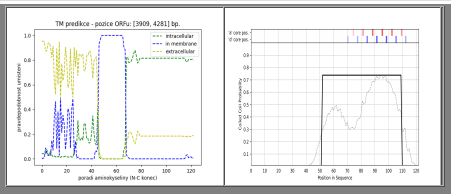
>ORF_1980-2196
TCTAACCAGAAGAACTTAGAAATGTGGGAGGAAAATGTATTTTTCCACCTTTCGCTCAATACCCTTGGTATTGAGGGGTCCAGTGGCACAGATGCTGCATACATTTCATTATCGACGATCACAACACCCTCCTCTTCACTCTCATCACTTTCCCAGGGATTCCACCTCTTAGCCTTCCAATCAGGCTTTGTC

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_1980-2196
SNQKNLEMWEENVFFHLSLNTLGIEGSSGTDAAYISLSTITTPSSSLSSLSQGFHLLAFQSGFVESSPTLL

ORF, zadané vlákno, 1. čtecí rámec - pozice:

[3909, 4281] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

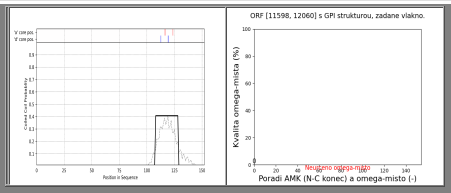
>ORF_3909-4281
CTTCTTCTCCTGACTCCTCCTCAAGCTCCATATATCCCAGTGCAGATCTGGATGGCACCCACTTTATACCACTATTTTCCTGTGCCCATGGATGACCACTCAAAGAAGGTAGTGCTGGGGAACCGAAACCTGCCAAGGTGGCTGTGGATCCTGCTGGTCCTGGTGATACTGGGTCTGACTGTGGCCGTG

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_3909-4281
LLLLTPPQAPYIPVQIWMAPTLYHYFPVPMDDHSKKVVLGNRNLPRWLWILIVLVILGLTVAVIVLAVENSSEACKNGLQAEQKCRNKTHLLELQLTQTQESLGVAKAQAASCNQTVVSCCHS

ORF, zadané vlákno, 1. čtecí rámec - pozice:

[11598, 12060] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

>ORF_11598-12060
TTGTTTGAAACCCCAGAGAGCGAGTCTAGATGTAGTTGGGATTGGAACCTGAATCTGACTGCAGAGCTTCAACTGTTTTCACATCCCAGTGGCTTGAGGTTGGGGGCCAGTGAGCCTGGGCAGAGACCTCCTTCCAGGCAGTCAGCTGTCACCACGCGGTACCCCAGACTTTTGCGGCCACTACTGTG

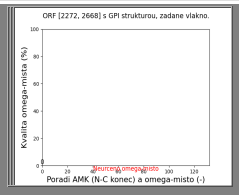
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_11598-12060
LFETPESESRCSWDWNLNLTAELQLFSHPSGLRLGASEPGQRPPSRQSAVTTRYPRLLRPLLCFTTTGPQRTFQQSSGNTSVVSPGEGGADSLLLPSFPLSLPLHCFTLQSQLQTQLMTLKTEMDEAKAQGTQMGAENGALTGVLGRKRAGQV

Zpět na začátek stránky.

ORF, zadané vlákno, 2. čtecí rámec - pozice:

[2272, 2668] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

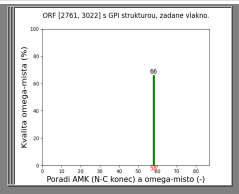
>ORF_2272-2668
AGTGAGGGAAGTTTGGCATTCGCCTTGGTTGCAAAATTTAGGGGAGCACCAAAAGGTTCTGTTATTATCATTCCCATTTTACAGTTATGGAAACTGAGGCCAAGAAGAATTTGCTTGCCTAGAGTCCCACAGGGAGGCAGGGCTGCTCACACACAGCCTTGGCCTCTGACCCACAATTGTGGCCCCAAG

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_2272-2668
SEGSLAFAIVAKFRGAPKGSVIIPILQLWKLRPRRICLPRVPQGGRAAHTQPWPLTHNCGPKVTLKFEGQHSAGCISQLYPGLCLCTWGSGQKGRPQLMGHVSSPRLLQQDGPPCLVIWLLRGFVGGVSR

ORF, zadané vlákno, 2. čtecí rámec - pozice:

[2761, 3022] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

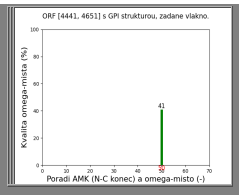
>ORF_2761-3022
AATGGCCCACCTAACAGAATCTGGGTCTCAATCAGCCAGCAGCAGGAAGCCTGTAGCTGCTCCCTTGGCCACACGGAGAGGAGGTCAGTGTGGCCTGAGTCCTCTCCAACCTCCCTCCAGGGTCCAGCCCTCTTTTGGGGCCTCTCCCAGGATGTACCCCTAGACCCTTTGGGTTCAGCATGTCCCAA

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_2761-3022
NGPPNRIWVSISQQQEACSCSLGHTERRSVWPESSPTSLQGPALFWGLSQDVPLDPLGSACPNYLSSKCAPCSVAIFLCFPPLLLP

ORF, zadané vlákno, 2. čtecí rámec - pozice:

[4441, 4651] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

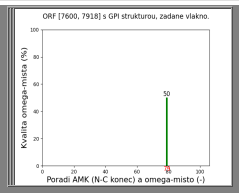
>ORF_4441-4651
ACTCTCTCTGGGGGTGCTGGTGTGGGGTGTGGCCTTAAAGCTCTTCAGAGCACTCAAATCAGGGTCCTCAGTATCCCCACTAGAATGTAGACGAATCTCCAACTCTAGGGTGTTGGTCCAGGGGAACAGAAATCCCCCGTGGGAGTTGGGATGGGGACCTCACCAGACCTTGAAGAAGGCAGGGGCTC

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_4441-4651
TLSGVLVWGVALKLFRALKSGSSVSPLECRRISNSRVLVQGNRNPPWELGWGPHQTLKKAGALILLLHP

ORF, zadané vlákno, 2. čtecí rámec - pozice:

[7600, 7918] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

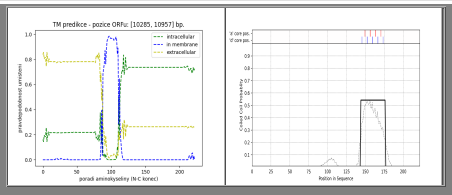
>ORF_7600-7918
TCCCTTCCTCTTCTGTTATTTCCTTCCCTCGAGTTCTGGAATCCCAACTATTGTCTAGGCGACTTCCTGTCCACAGCACTGGGCCTTTTGCCCAGATTTCTGCACTGCCCTCTGGGCCTATGGTCCCCAGCCCGGATGGTCAGTAGAGTCCCCTCCATGGCTCCTACTGTCCACCTAGGGACCCCCAAGC

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_7600-7918
SLPLLLFPSLEFWNPNYCLGDFLSTALGLLPRFLHCPLGLWSPARMVSRVPSMAPTVHLGTPKHHPCHRPPGFLVPSPCSSCFTLHDHFSMYGLCQFPSALTLTF

ORF, zadané vlákno, 2. čtecí rámec - pozice:

[10285, 10957] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

>ORF_10285-10957
GGCTTAGAGGACAGCACGCTGGAGGCAGAGGCAGAACTGGAGCCCAGGATGCCCAGTCTGATCCTGGGCTCGGGTAGCCTGGTCTGGTCCCCGACAGGAACCCAGTCCCTCCCAAGTGATTCGAGAAGCAGCAGCCGGGCGGGGCGGAACCGTGGGTGTCCCTGGCGTCCCAAGGTGCCAATTAA

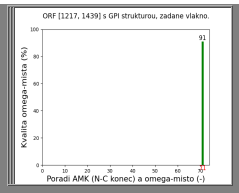
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_10285-10957
GLEDSTLEAEAELEPRMPSLILGSGSLVWSPTGTQSLPSDSRSSSRAGRNRGCPWRPKVPIKCSWLSGDRAVDGSMAEPGPEPGRAWRVLALCGAAVFLAAAAAGAALLAWNLAAAASRGSRCPEPDQGVNATAPPWDRVPEVEELLRRLAEATQREEVLTRKLDQAEGVRWELEEALRVCEGRQVCE

Zpět na začátek stránky.

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[1217, 1439] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

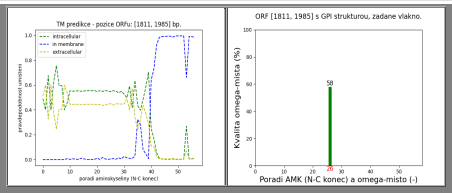
>ORF_1217-1439
AATAATGCCATCTTTAAGAGAGCGAAGACCTTGAACCTCCTGAGCGCCGTGCAGGCACTGCCAGGGGGCAAGGTGTCTGGTGACACTGCTCATTTTCTCTGCCTCCTGCTGGGCAGAGAAAGGCCATCACCCCCACACATCCCACGCCACAGTAACAATTCCTGGATACTTCCCCTGGCCTTTTGCCAC

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_1217-1439
NNAIFKRAKTLNLLSAVQALPGGKVSGDTAHFLCLLLGRERPSPPHIPRHSNNSWILPLAFCQKLAAMSTSSA

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[1811, 1985] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

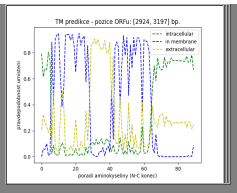
>ORF_1811-1985
AGTCCCAAACCACAACGCACCACCCCCTTCTCCTTAGCCCCAGTGGCATCTGAGCCTCAATTGCCGATCTGTCCTGGGGCCTCATATTCTTATGGCACCCCGTGTATCGTAAGTAATAAAGCTGGTTATTTTCTCCTGTTAGTCTGTCTTGTACTAATTAGATTATTAGTCTAA

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_1811-1985
SPKPQRTTPFSLAPVASEPQLPICPGASYSYGTPCIVSNKAGYFLLIVCLVLIRLLV

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[2924, 3197] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

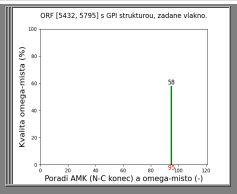
>ORF_2924-3197
ACCCTTTGGGTTCAGCATGTCCCAATTATCTTTCCTCAAAATGTGCTCCTTGCTCTGTGGCCATATTTCTGTGTTTTCCACCTCTCCTCCTCCCCTGAGCCATGGGCCCTGCCTGCATATCCACTGAACTACTGGATGCAATGTTTTACTCTCAACGCCTCATTGACTCTCCTCATGAGGCTGCTCCTGTT

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_2924-3197
TLWVQHVPIIFPQNVLLALWPYFCVFHLSSSPEPWALPAYPLNYWMQCFTLNASLTLLMRLLLLLPHFTDGETEVPRVYLASWDGRGSRF

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[5432, 5795] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

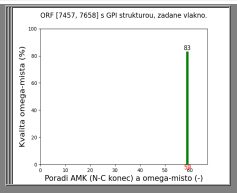
>ORF_5432-5795
GGTCAAGACTTCTCCCCCTTCGTCCCTCCCCAATCCCCTGAACCCCCTCCACTCGGAAAATTCCCATTGCTCTGGGGACAAAGTTTGAGTCTCTGTCTTGTGTCCCGTGTCGCCCAAGCGCGGGTCGCAGCTACCCACTCTCCTCCGCAGCGACCTGGTAGGGGAGGAGAGGAGATTCTGAGGGGCGA

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_5432-5795
GQDFSPFVPPQSPEPPPLGKFPLLWGQSLSLCLVSRVAQARVAATHSPPQRPGRGGEEILRGDGRSAGGARPGSCPNCPHLHPRKEHASGEKNGSTSSRNARSSLVVVVLLSLSFRALLA

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[7457, 7658] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

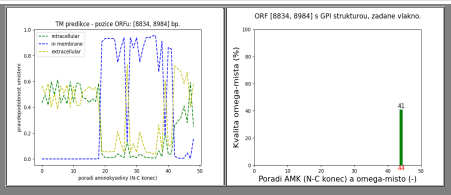
>ORF_7457-7658
CTAGCTTGCAACAATGAATTTCTAGGCGTCTGCCTTCAATTCTTTGTCGACCAGACACTAAGAACAGAGGTCCAGAAGTTCCAGGACTCCCCAGAACTGGTAACACCTTCACCCACTCTCTTGGATCCCTTCATGCCCCATAATCCCTTCCTCTTCTGTTATTTCCTTCCCTCGAGTTCTGGAATCCCAAC

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_7457-7658
LACNNEFLGVCLQFFVDQTLRTEVQKFQDSPELVTPSPTLLDPFMPHNPFLFCYFLPSSSGIPTIV

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[8834, 8984] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

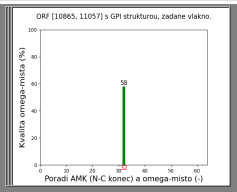
>ORF_8834-8984
GGGGTCCTGCAGGTCTGTGGGGAGTTCCAGCTCCAGCAAGAATCTGGCAGTGACCAGTGGCTGCTGTATGTCTCTGGCCTGAGACTTGGCCTCTCTGGGCTTCTATTTATTCATGTGGAATGGTGGCCAGGGTCGACACTGAGATTCTAG

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_8834-8984
GVLQVCGEFQLQQESGSDQWLLYVSGLRLGLSGLLFIHVEWWPGSTLRF

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[10865, 11057] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

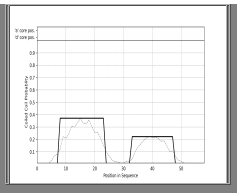
>ORF_10865-11057
GGCATCCAGTGGGGGGAGAGGGCATGCACCCCCCACCCCAGAGATCACCTCCACGCCGGGCCTTGGTTTCCCCAGTTCTTCAATGTGAATAAGAATAGTACCATAATCCCTTCCAACTCCTACATTGCCAGTTTACCCTTGAGTCCATCTTCTGATTTAGGAGGCCCCACCATCCTCTATTTACTGGATTA

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_10865-11057
GIQWGERACTPHPRDHLHAGPWFPQFFNVNKNSTIIPSNSYIASLPLSPSSDLGGPTILYLLD

ORF, zadané vlákno, 3. čtecí rámec - pozice:

[13676, 13856] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

>ORF_13676-13856
TTCAGTATTCAGAAGTTTGTTAATAAATTCACCATACTAATGGATCAAATGAGAAGAGTTCTAAAAAAGGCCATCGAAATCGAGGCCAAGGGGAGGAAAGACTACTATCAATTAGTTAATATTGTTCTGGTGGTACTAGCCAATACAATTAAACAAGAAAAAAGCAATGTACAAATATAG

Sekvence aminokyselin v rámci ORFu (N --> C):

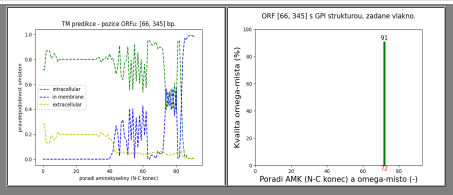
>trORF_13676-13856
FSIQKFVNKFTILMDQMRRVLKKAIEIEAKGRKDYYQLVNIVLVVLANTIKQEKSNVQI

Zpět na začátek stránky.

jednotlivé ORFy (komplementární vlákno):

ORF, komplementární vlákno, 1. čtecí rámec - pozice:

[66, 345] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

>ORF_66-345
TTTTTAAAAAAGAAATATATGTGTATGGTTTTTAAAATGCAAACTTTACAAAATGATATACGGAGAACATGCATCTGCTTTCTTCACCAACCAACCCCCCAGGCCCTTGGTTCCCAACCAGAGGCCACAGCTGTGGCCCGTGTTCTGTGTGGAAATATTTGGATAAAAGTCTCCCTCGGCTTTGCATTTAAA

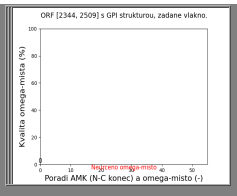
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_66-345
FLKKKYMCMVFKMQTLQNDIRRTCICFLHQPTPQALGSQPEATAVARVLCGNIWIKVSLGFAFKKMYTNYCSSAFYIYVCLRRSALAFLIVA

Zpět na začátek stránky.

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[2344, 2509] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

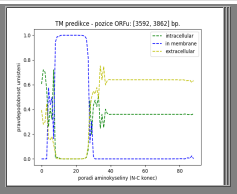
>ORF_2344-2509
CCTCTCCAGGTGACAATGCAGCCAGGGCTGGAACCTAAAGTTCTCTCATCTGTAGGATTAGAACAACACTATCGTTTGTGGAGGCTTAAGTCCCAAATCAGCACTATCTCACTGGGCGACCAAATGGGCCCCTGTTTCCTCCTCTGTAGAATAAGCACTAGATAG

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_2344-2509
PLQVTMQPGLEPKVLSSVGLEQHYRLWRLKSQISTISLGDQMGPCFLLCRISTR

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[3592, 3862] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

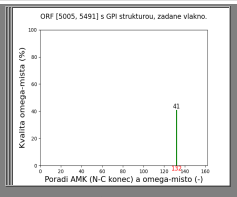
>ORF_3592-3862
GTTTGCATTGATTTTGGAAGTCGTTTATATGGCGTTTGCATATGTTGTTGCATTTGCGTTTTTGTTACCATTGTATATTATGTGCAATTTAATAAGTGTCGTTTGCACCTGTGCGTAATTCCCACCTCTAGACCTGACCAGCACGCTTGCGCCCCAACACACCTGTCAGCGCCCCGTTCTCGGCCCCCATTT

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_3592-3862
VCIDFGSRLYGVCICCCICVFVTIVYYVQFNKCRLHLCVIPTSRPDQHACAPTHLSAPRSRPPFGSLVLWPRPSQSSVSSAGSEAGSAE

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[5005, 5491] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

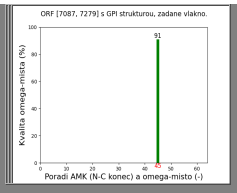
>ORF_5005-5491
CTTCCTGGTCAGAACCTCTTCGCGTTGGGTGGCCTCTGCCAGCCGCCGCAGCAGCTCCTCGACCTCGGGCACCCTGTCCCACGGAGGCGCTGTGGCGTTGACCCCCTGATCTGGCTCTGGGCAGCGAGACCCCCGGGAGGCCGCAGCGGCCAGATTCCAGGCCAGCAGGGCTGCCCCAGCTGCTG

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_5005-5491
LPGQNLFALGGLCQPPQQLLDLGHPVPRRRCGVDPLIWLWAARPPGGRSGQIPGQQGCPSCCCRQEHCGPTQGEHPPGAPWLWPRLGHAAVHSSVPRQPRAFNWHLGTPGTPTVPPRPAAASRITWEGLGSCRGPDQATRAQDQTGHPGLQFCLCLQRAVL

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[7087, 7279] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

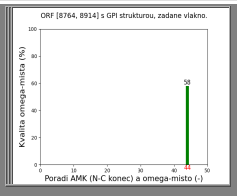
>ORF_7087-7279
GAGAGAAGAGGGTTCTTGGTTTTAGCTATAGGGAATGAATGTAGGGTGCCCCTTAAAGACCGTTTTTCCTGTTTTCCCTTCACAGGTGTGGGAAATGGGTTGTTAGCAATTCCAGCAGAAATCTTCCTTCGTTCTGCAGGTTTCTTTCTGGTCTGGCTGCAGGGCATATTGCAATCTCTGAATACAGCTTC

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_7087-7279
ERRGFLVLAIGNECRVPLKDRFSCFPFTGVGNGLLAIPAEIFLRSAGFFLVWLQGILQSLNTA

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[8764, 8914] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

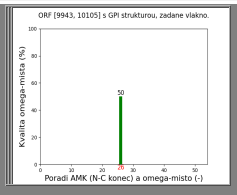
>ORF_8764-8914
CTGTCCTGCGCATGTCTATGTTGGGGTCTTTCTCTATCTCCTTTATTTTCCACTTTAGCAGGAAGTTTTAAACTGCAAAACTGCTTCACGAAGTCAAAACTATTGTTCAAGGCTGTTTTTCATTTTAGTGGCTCTGGGGGTATGTCCTGA

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_8764-8914
LSCACLCWGLSLSPLFSTLAGSFKLQNCFTKSKLLFKAVFHFSGSGGMS

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[9943, 10105] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

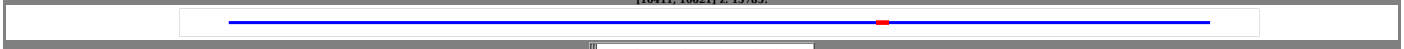
>ORF_9943-10105
CTCCCCGGTGCCCGCTCCTTACCAGCCGGCTTTCTGAGACCTCAGGCCAGCAGAGCTCGGAAGCTCAGACTCAAGAGCACAACAACTACCAGGGAGCTTCTGGCGTTCCTGGAGCTGGTGGAGCCATTTTTCTCGCCAGAGGCATGCTCTTTCCTAGGGTGA

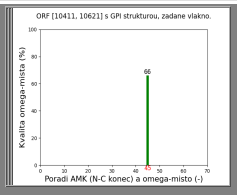
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_9943-10105
LPGARSLPAGFLRPQASRARKLRLKSTTTTRELLAFLEILVEPFFSPEACSFLG

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[10411, 10621] z: 15785:





Sekvence nukleotidů v rámci ORFu (5` --> 3`):

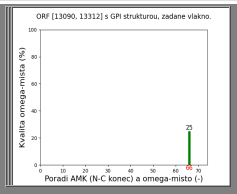
>ORF_10411-10621
GGTGGGACAGAAATCAGGGGGTGTGGCACCGTGGCCACGCCCCTCACCTGGCCCCACTCTTGGACAGGACCTCATAGCTAGTCCTGTCCTCATCCAGGCCCACCCTGGAAACCTAGCGTGCCCTCTCCTCACCTCCACTGGATTCCGTCCCGGTTCCTCCCGACCCCAACCCTGGGGCCCCGCCTCCT

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_10411-10621
GGTESGGVAPWPRPSPGPTLGQDLIASPVLIQAHPGNLACPLLTSTGFRPGSSRPQPWGPASSLVFFSP

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[13090, 13312] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

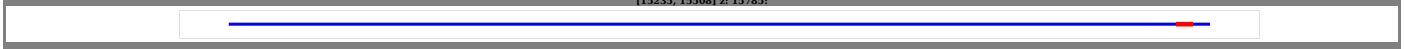
>ORF_13090-13312
GGCCCCCAGTTGGAGATCCCTCTACCTAGACACTCCCCCCACAAAGCCTCTCAGCAGCCAAATGACAAGGCAAGGAGGACCATCTTGCTGAAGAAGTCTGGGGGGAGCTTACGTGTCCCATGAGCTGGGGCCTCCCTTTCTGACCAGAACCCCAGGTGCAAAGACACAACCCTGGGTAAAGCTGACTTA

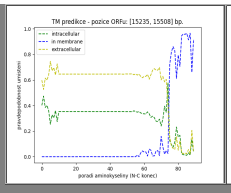
Sekvence aminokyselin v rámci ORFu (N --> C):

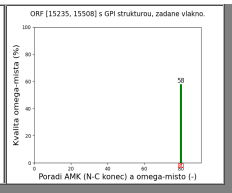
>trORF_13090-13312
GPQLEIPLPRHSPHKASQQPNDKARRTILLKKSGGAYVSHELGPPFLTRTPGAKTQPWVKLTYATCTVLAFKF

ORF, komplementární vlákno, 2. čtecí rámec - pozice:

[15235, 15508] z: 15785:







Sekvence nukleotidů v rámci ORFu (5` --> 3`):

>ORF_15235-15508
ACAAAGATGACGGCTCTGGTGGATGCTGGAGCTCAGAGTAGTCTAATATATGGTGACCATCAGAAGTTTTCTGGCTCCCTCAGCACCATTAATGGTTATGGAGAGCAAGTGGTTATGGCCAAGAAGATCCCTTTGACACTGCAAATTGGGCATTCTTCCCCCCGAGAATATGAAGTTATTTTATTACCCAT

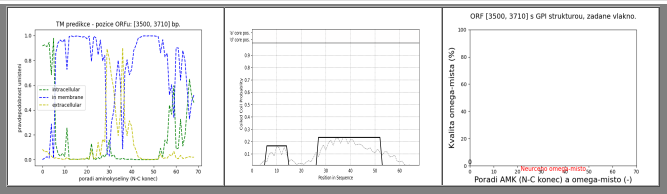
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_15235-15508
TKMTALVDAGAQSSLIYGDHQKFSGSLSTINGYGEQVVMAKKIPLTLQIGHSSPREYEVILLPIFPPLLPFGNHRVVLYSMSLFLFCFNY

Zpět na začátek stránky.

ORF, komplementární vlákno, 3. čtecí rámec - pozice:

[3500, 3710] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

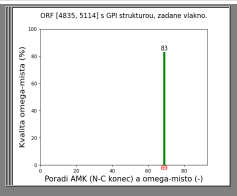
>ORF_3500-3710
TCTGCCTATTGTTTGCGTGCTTTGCTTTATTTGAATTATATTTGTGTGGTCTGTACATGGCCTTTACGGGTCATTTGCATAGCTTTTCATAGGTTTGCATTGATTTTGGAAGTCGTTTATATGGCGTTTGCATATGTTGTTGCATTTGCGTTTTTGTTACCATTGTATATTATGTGCAATTTAATAAGTGTCGTT

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_3500-3710
SAYCLRALLYLNYICVVCTWPLRVICIAFHRFALILEVVYMAFAYVVAFAFLLPLYIMCNLISVVCTCA

ORF, komplementární vlákno, 3. čtecí rámec - pozice:

[4835, 5114] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

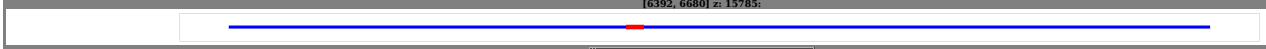
>ORF_4835-5114
AGAACTGGGGAAACCAAGGCCCCGGCGTGGAGGTGATCTCTGGGGTGGGGGGTGCATGCCCTCTCCCCCCACTGGATGCCCTAGCACCCAACCCTGTTCACATACCTGGCGGCCCTCACAGACCCGCAGTGCCTCCTCCAGCTCCCAACGGACACCCTCAGCCTGGTCTAGCTTCCTGGTCAGAACCT

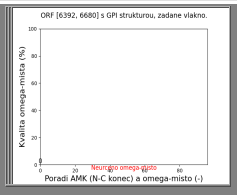
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_4835-5114
RTGETKARRGGDLWGGCMPSPPTGCPSTQPCSHTWRPSQTRSASSSSQRTPSAWSSFLVRTSSRWVASASRRSSSSTSGTLSHGGAVALTP

ORF, komplementární vlákno, 3. čtecí rámec - pozice:

[6392, 6680] z: 15785:





Sekvence nukleotidů v rámci ORFu (5` --> 3`):

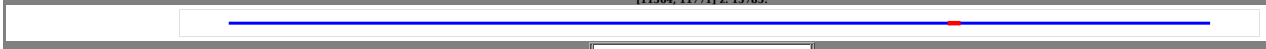
>ORF_6392-6680
TTCAAACAGGTATTTTGGCAATATATCTGTTATCTAGACAAAGTATTCCTAAGGGGAGAGTGGTTCTGCATGTTTACAGAAACAAGACAAGGAGCTGATTACAGCAAGCAAGCATGTTTACAGAAGCAGAACACAGTTGATTACAAGGTTAGCCCTTTGAGTGGAAAATCCCTCTTCATCTCCTTCTCAGAA

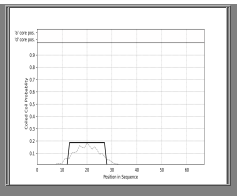
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_6392-6680
FKQVFWQYICYLDKVFLRGEWFCMFTETRQGADYSKQACLQKQNTVDYKVSPLSGKSLFISFSEPSYFPLPQAPFDHFLPPCCNFPLCLILSCLS

ORF, komplementární vlákno, 3. čtecí rámec - pozice:

[11564, 11771] z: 15785:





Sekvence nukleotidů v rámci ORFu (5` --> 3`):

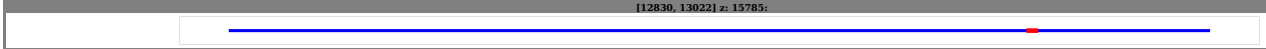
>ORF_11564-11771
GCTTTCCTGGGTCTGGGTCAGTTGGAGCTCCAGGAGGTGGGTTTTATTTCGACACTTCTGCTCTGCTTGGAGGCCATTTTTGCAGGCCTCACTGCTGTTCTCGACGGCGAGGACGATCACGGCCACAGTCAGACCCAGTATCACCAGGACCAGCAGGATCCACAGCCACCTTGGCAGGTTTCGGTTCC

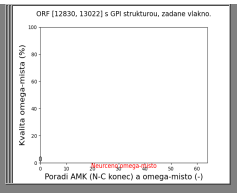
Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_11564-11771
AFLGLGQLELQEVGFISTLLLCLEAIFAGLTAVLDGEDDHGHSQTQYHQDQQDPQPPWQVSVPQHYLL

ORF, komplementární vlákno, 3. čtecí rámec - pozice:

[12830, 13022] z: 15785:





Sekvence nukleotidů v rámci ORFu (5` --> 3`):

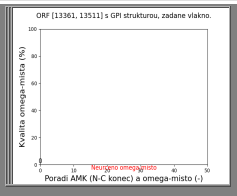
>ORF_12830-13022
TTGGGACATGCTGAACCCAAAGGGTCTAGGGGTACATCCTGGGAGAGGCCCCAAAAGAGGGCTGGACCCTGGAGGGAGGTTGGAGAGGACTCAGGCCACACTGACCTCCTCTCCGTGTGGCCAAGGGAGCAGCTACAGGCTTCCTGCTGCTGGCTGATTGAGACCCAGATTCTGTTAGGTGGGCCA

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_12830-13022
LGHAEPKGSRGTSWERPQKRAGPWREVGEDSGHTDLLSVWPREQLQASCCWLIETQILLGGPF

ORF, komplementární vlákno, 3. čtecí rámec - pozice:

[13361, 13511] z: 15785:



Sekvence nukleotidů v rámci ORFu (5` --> 3`):

>ORF_13361-13511
GCAGCCCTGCCTCCCTGTGGGACTCTAGGCAAGCAAATTCTTCTTGGCCTCAGTTTCCATAACTGTAAAATGGGAATGATAATAACAGAACCTTTTGGTGCTCCCCTAAATTTTGCAACCAAGGCGAATGCCAAACTTCCCTCACTCTAG

Sekvence aminokyselin v rámci ORFu (N --> C):

>trORF_13361-13511
AALPPCGTLGKQILLGLSFHNCKMGMIITEPFGAPLNFATKANAKLPSL

Zpět na začátek stránky.

Příloha B: Obsah přiloženého DVD

- Klíčová slova (čj i aj)
- Abstrakt česky
- Abstrakt anglicky
- Naskenované zadání diplomové práce
- Kompletní diplomová práce
- soubor „17PMP2DP2_368212_Petr_Adamek.zip“ se:
    - souborem „rand_string.py“ pro generování náhodné sekvence DNA,
    - souborem „linky-TM,CC,GPI.py“ pro generování obrázků 5.1, 5.5, 5.9, 5.13 a 5.17 použitých v diplomové práci,
    - složku: „HTMLproTMaCCaGPI“ s:
        - hlavním programem - soubor: „HTMLproTMaCCaGPI.py“,
        - vygenerovanou HTML stránkou: „index23.html“,
        - dalšími složkami a soubory nutnými pro funkci hlavního programu a HTML stránky.