Úvod do praktické bioinformatiky
Fatima Cvrčková

Velkým úkolem dnešního biologického bádání je dospět od dat ke znalostem. Znám lidi, kteří si myslí, že data už jsou znalosti; ti se však pro změnu pídí po tom, jak dospět od znalostí k pochopení.

Sydney Brenner (The Scientist 16:12, 2002)

Nikdy bych se nebyla pustila do psaní této knihy, a už vůbec bych ji nebyla dopsala, bez mnohých, kterým bych zde ráda poděkovala.

Marcovi van Ranstovi, jednomu z mých prvních průvodců ve světě bioinformatiky, děkuji za inspirující zkušenost spojenou s absolvováním kursu Computational Genomics r. 1995, jakož i za „nultou verzi" adresáře bioinformatických odkazů, který (byť i ve značně přeměněné formě) dodnes používám. Jaroslav Flegr si zasluhuje díky za mnohé konzultace týkající se konstrukce dendrogramů a statistických metod – věřím, že za ta léta by to vydalo za více než semestrální kurs – i za kritické čtení první verze této práce.

Edovi Himelblauovi děkuji za souhlas s přetištěním dvou z jeho kreseb, poprvé uveřejněných v dnes již zaniklém internetového magazínu HMS Beagle.

S obzvláštním potěšením děkuji všem mladším kolegům, kteří se velice rychle v lecčems stali ze studentů učiteli. Za mnohé bioinformatické dovednosti vděčím zejména Markovi Eliášovi, Marianovi Novotnému a Martinovi Potockému, který mi také poskytl podklady pro část kapitoly věnované konstrukci trojrozměrných modelů molekul proteinů.

Struktura tohoto textu se do značné míry zrodila během přípravy mezinárodního kursu bioinformatiky financovaného projektem EU HPRN-CT-2002-00265 „TIPNET".

K tomu, že pobývat v laboratoři buněčné biologie katedry fyziologie rostlin PřF UK bylo a je potěšením, přispěla v různých dobách řada dalších kolegů – především pak Viktor Žárský, který mne uvedl do světa rostlinné buněčné biologie, a s nímž je mi ctí a potěšením spolupracovat již neuvěřitelných deset let. Snad se na mne ostatní nebudou zlobit, že je zde v zájmu únosné délky poděkování neuvádím jmenovitě – až na Radka Bezvodu, který byl snad až příliš shovívavým čtenářem předběžných verzí tohoto textu.

Můj muž, Anton Markoš, mi byl a je nejen morální oporou a člověkem nejbližším, ale i prvním čtenářem a laskavým, leč kritickým oponentem.

Tuto knihu věnuji mému otci, MUDr. Janu Zdeňkovi Cvrčkovi (* 7.6. 1915), se vzpomínkou na to, jak jsem se kdysi ve věku asi tak sedmi let domáhala vysvětlení, co jsou to aminokyseliny. Teď už to doufám vím.

V Praze 12.6.2005
Fatima Cvrčková

Z technických důvodů (obnova poškozeného souboru) se poznámky pod čarou od č. 70 nacházejí na konci souboru.

Obsah

1. K čemu a o čem je tento text

Již celé desetiletí – totiž od zveřejnění první kompletní genomové sekvence bakterie *Haemophilus influenzae* (Fleischmann et al., 1995) – žijeme v postgenomové době, a několik let nás už dělí i od publikace (téměř) úplného genomu huseníčku Thalova (The Arabidopsis Genome Initiative, 2000). Na epochální důležitosti analýzy genomových sekvencí se shodnou nejen úvodníky významných vědeckých časopisů a praktikující molekulární biologové, ale v rostoucí míře i zástupci disciplín systematicko-ekologických, jejichž tradiční doménou bývaly donedávna spíše obory morfologické.

O překotnosti vývoje posledních let svědčí i to, že samotnou **definici bioinformatiky** je poměrně těžké najít v tištěné literatuře mimo specializované publikace, a do značné míry jsme odkázáni na webové zdroje proměnlivé důvěryhodnosti.[1] Zde budeme bioinformatiku chápat, podle definice zveřejněné na poměrně autoritativním serveru `http://bioinformatics.org`, jako „klasickou bioinformatiku“ – tedy oblast na pomezí biologie a informatiky, která se zabývá především zpracováváním, prohledáváním a analýzou dat o sekvenci (pořadí monomerů) a struktuře biologických makromolekul (Counsell, 2004). Termín „bioinformatika“ může však být chápán i ve významu širším jako „využití počítačů k hledání odpovědí na biologické otázky“ (Baxevanis, 2001a), což by ale mohlo zahrnovat třeba i statistické zpracování fyziologických či klinických dat, která se strukturou makromolekul nemají přímou souvislost. V diskusích v odborných kruzích se můžeme naopak setkat i s pojetím užším, podle něhož opravdovým bioinformatikem je jen ten, kdo vyvíjí software pro pokud možno automatizované (či přinejmenším algoritmizované) zpracování a analýzu sekvenčních dat, nebo aspoň spravuje vlastní databázi. „Vysoká“ bioinformatika v tomto smyslu se vyvinula v plnohodnotný obor, z něhož na mnohých univerzitách lze obhájit doktorát a pak se mu věnovat jako celoživotní specializaci. Průvodním jevem specializace je vybudování vlastního souboru problémů, pokládaných za dostatečně důstojné, aby se na nich dala dělat dizertace (např. jakým způsobem najít v sekvenci kompletního eukaryotního genomu hranice všech intronů a exonů s větší než 95 % pravděpodobností).[2] Typickými tématy „vysoké“, „nové“ či „postgenomové“ bioinformatiky jsou katalogizace genů a genových produktů v celogenomovém či ještě větším měřítku (genomika, transkriptomika, proteomika) a srovnávací analýzy většího počtu genomů směřující např. k predikci funkcí dosud neznámých genů a interakcí jejich produktů na základě korelace výskytu homologů těchto genů v různých skupinách organismů.

[1] Na internetu se ovšem v naší postmoderní době – a jinou toto médium nepamatuje – najde leccos, někdy i věci natolik bizarní, že pro pobavení čtenáře stojí za citaci *in extenso*: *Podle definičního třídění ruských vědců rozlišujeme dva obory paranormálních jevů: bioinformatika a bioenergetika. Bioinformatika (tzn. mimosmyslové vnímání, ESP) zahrnuje získávání a výměnu informací mimosmyslovou cestou (nikoli normálními smyslovými orgány). V podstatě rozlišujeme následující formy bioinformace: hypnózu (kontrolu vědomí), telepatii, dálkové vnímání, prekognici, retrokognici, mimotělní zkušenost, „vidění“ rukama nebo jinými částmi těla, inspiraci a zjevení. Naproti tomu bioenergetika popisuje jevy souvisící se vznikem objektivně vnímaných efektů, aniž by brala v potaz normální fyzikální síly nebo energie. Patří sem: psychické pohybování předměty (psychokineze), podobně vyvolaný ohýbací efekt, antigravitační jevy, změny energie, elektromagnetické působení bez dostatečného fyzikálního vysvětlení, nevysvětlitelné chemické a biologické procesy probíhající zřejmě za účasti vědomí, psychometrie a proutkařství* (Novák, 2001). Bioinformatik – i bioenergetik, který absolvoval třeba slavnou přednášku profesora Kubišty na Přírodovědecké fakultě UK – se jen diví, a připomíná si, že ne všemu, co je psáno a co se tváří jako definice, se dá věřit.

[2] Blíže o tomto problému v části 9.1.

Biolog, který nachází jádro svého působení v laboratoři, zůstává vůči „vysoké bioinformatice“ v pozici zvídavého laika, žasnoucího nad cizími výsledky získanými pomocí metod, které se navenek jeví jako černá skříňka. V lepším případě se stává poučeným uživatelem programů, které sestavili jiní. Toto konstatování by nemělo být chápáno jako hanlivé; důraz je totiž na slově *poučeným*. Na tom, že pracuji s černou skříňkou, není nic špatného – kolik průměrný uživatel ví o vnitřní architektuře motoru ultracentrifugy? Experimentující biolog totiž bioinformatické nástroje užívá podobně jako ultracentrifugu. Řeší ovšem otázky jiného druhu než ty, které by, pokud bychom měli pokračovat v našem příměru, zajímaly fyzika studujícího proces ultracentrifugace. Nad 95 % úspěšností predikce obecného intronu v obecném genomu se sice asi zaraduje, ale pro konkrétní intron v konkrétním genu nutně potřebuje predikci pokud možno skoro stoprocentní. Kupodivu takové přesnosti často dosáhnout lze, a to právě pro jednotlivé, dobře vybrané, nadstandardně experimentálními daty i literárními souvislostmi podložené případy. Musíme však rezignovat na obecné algoritmy a vysoce výkonné (*high throughput*) softwarové nástroje, které jsou chloubou „vysoké“ bioinformatiky! Podobně jako se všeobecným zprůmyslněním výroby obuvi dokonce ani v nejbohatších státech nezaniklo ruční šití bot (ale setkáme se s ním se už jen u těch nejdražších, kde každý pár je jedinečný), i zde se vedle vysoké technologie postgenomového věku i nadále uplatní pečlivá ruční práce. Tato „drobná domácí výroba“ však by měla být podložena zkušeností biologa, který sice neovládá rafinované matematické teorie a zpravidla ani neumí programovat, avšak vyzná se v džungli molekul a regulačních drah organismu, kterým se zabývá.

Jak se ale náš experimentátor má stát poučeným uživatelem? Podobně jako u jiných laboratorních dovedností, i tuto je nejlépe pochytit od někoho, kdo to umí.[3] Výhodou ovšem je absolvovat nějaký praktický kurs, popřípadě mít k dispozici literaturu, která by mohla posloužit na způsob itineráře či turistického průvodce. Z cizích publikací z nedávné doby považuji za zdařilou zejména útlou knížečku od Andrey Hansen (2001), která však pokrývá, i když velmi názorně a důkladně, jen část potřebné tématiky (zejména prohledávání sekvenčních databází a konstrukci dendrogramů), a krom toho existuje pouze v němčině. Minoru Kanehisa (2000) podává hezky biologicky zaměřený, spíše teoretický než praktický a poměrně náročný vhled do myšlenkového pozadí řady postupů „vysoké“ bioinformatiky. Podrobnější, ale pro začátečníka asi až příliš hluboký a rozsáhlý přehled, který se dotýká i některých oblastí překračujících běžné uživatelské aplikace, pak poskytuje např. monografie kolektivu autorů ze série „Methods of biochemical analyses“ (Baxevanis a Ouelette, 2001). Česky dosud na příbuzná témata nevyšlo nic; věřím tedy, že je na čase to napravit.

1.1. Modelové problémy

S výjimkou krátkého úvodu věnovaného praktickým předpokladům jednoduché bioinformatické práce (2. kapitola) se zde zaměříme na několik vybraných tématických a metodických oblastí. Výběr vychází ze zkušeností a typických situací, s nimiž se setkává biolog, zabývající se molekulárními mechanismy fungování rostlinné buňky a využívající,

[3] V tomto ohledu si věda zachovala strukturu středověkého cechu, i s institucí mistrů (tedy šéfů laboratoří), putujících tovaryšů (postdoků) a učedníků (diplomantů a doktorandů). Stojí za připomenutí rada, kterou svým budoucím kolegům kdysi udělil pozdně gotický malíř Cennino Cennini: *A co nejdříve můžeš, svěř se vedení mistra, abys se od něj učil: a co nejpozději budeš moci, od něj odejdi*! (Cennini, 1437/1946) – i když o tom, zda je i její druhá polovina ve vědeckém světě hodna následování, bychom mohli diskutovat.

i když nikoli výhradně, jako experimentální model *Arabidopsis thaliana*.[4] Neznamená to však, že by šlo o metody či témata pouze rostlinná. Studovaný organismus se promítá především do výběru přednostně využívaných datových zdrojů. Obdobnou strategii lze však uplatnit i při výzkumu jiných systémů, zejména tam, kde existují srovnatelně důkladně experimentálně prozkoumané modely (např. kvasinky, obratlovci). Mnohé uváděné postupy jsou použitelné univerzálně, jak pro eukaryota, tak i pro prokaryota, jiné jsou vázány na specificky eukaryotické jevy (např. predikce sestřihu jaderných mRNA – 9.1).

Pravděpodobně každý molekulární biolog se někdy ocitne v situaci, kdy potřebuje

- sestrojit mapu úseku DNA (genu, klonu nebo třeba právě připravovaného plasmidu), zjistit v něm přítomnost restrikčních míst a kódujících oblastí, odvodit sekvenci kódujícího řetězce DNA z nekódujícího, navrhnout primery pro PCR (3. kapitola);
- získat co možná nejvíce dostupných literárních, sekvenčních, případně i strukturních a dalších údajů o konkrétním genu či proteinu (4. kapitola) ;
- zjistit, co kóduje (popřípadě čemu se nejvíc podobá) úsek cDNA, který právě naklonoval, ať už jakýmkoli experimentálním postupem (5. a 6. kapitola);
- nalézt ve studovaném biologickém materiálu homology známých genů či proteinů (6. kapitola);
- zjistit, zda zkoumaný protein obsahuje již známé sekvenční či strukturní motivy (7. kapitola);
- porovnat sekvence dvou či více vzájemně příbuzných genů a z výsledku predikovat případné funkční rozdíly mezi nimi (5. a 8. kapitola);
- najít funkční geny v dosud neanotovaném či špatně anotovaném úseku genomové sekvence (9. kapitola);
- posoudit míru příbuznosti genů v rámci velké genové rodiny a z výsledku usoudit na evoluční historii či alespoň funkční rozrůzněnost (10. kapitola).

Těmito úkoly – a některými dalšími problémy, které cestou vyplynou – se budeme zabývat v následujících kapitolách.

Nástroje, které budeme používat, podléhají pochopitelně rychlému vývoji, a i při sebevětší snaze z mé strany budou přinejmenším některé z nich zastaralé už v době, kdy tento text přijde čtenáři do ruky. Snažila jsem se vybírat konkrétní příklady především tak, aby názorně vypovídaly o používaných postupech, někdy i na újmu aktuálnosti. Pochopí-li čtenář princip metody, nebude mít potíže s používáním novějších alternativ, které se mohou z uživatelského hlediska jevit tak trochu jako černá skříňka.[5] Kdo někdy sekvenoval DNA na dlouhém gelu, ví, o čem je řeč, a zajisté si dovede představit, odkud se vzal digitální elektroforetogram z nejnovějšího plně automatizovaného přístroje; naopak už to tak snadné není.

[4] Konkrétní příklady čerpám především z vlastní zkušenosti, což umožňuje zastavit se i nad některými detaily, které se do publikací obvykle nepíší.
[5] Nebo snad bledě šedá skříňka ... třeba s logem Applied Biosystems?

2. Praktické předpoklady

Nebudeme se zde zabývat "vysokou" bioinformatikou, ale využitím vybraných bioinformatických nástrojů v běžné praxi experimentální laboratoře. Proto se budeme snažit vystačit s rozumným minimem – ať již po stránce vybavení (hardwarového i softwarového), tak i co se týče teoretické přípravy. Počítejme s tím, že čtenář je sice specialista, ale v něčem úplně jiném (např. v genetice *Arabidopsis*), a že se nehodlá přeškolovat na programátora, ani nemá v úmyslu věnovat většinu rozpočtu laboratoře na nákup specializovaného programového vybavení, jehož cena se zpravidla pohybuje minimálně v desítkách tisíc korun.

Jedinou opravdu nezbytnou nákladnější součástí vybavení tedy zůstává **počítač**. Pro veškeré zde zmiňované metody bude vyhovovat jakýkoli relativně moderní přístroj "kancelářského" typu (cokoli to v době, kdy čtete tento text, znamená), s rychlým připojením na Internet. Telefonní modem běžných parametrů vám nejspíš stačit nebude, nebo se přinejmenším nedoplatíte (nemáte-li paušál).

Co se týče **softwarového vybavení**, v dalším výkladu budeme předpokládat, že váš počítač je vybaven operačním systémem Windows, v našich krajích u této třídy počítačů nejběžnějším. Většina zde zmiňovaných aplikací však je dostupná i uživatelům jiných systémů, protože buď fungují na serverech přístupných prostřednictvím jakéhokoli webového prohlížeče, nebo sice vyžadují lokální instalaci, ale existují i ve verzích pro Linux, případně MacOS. Pracujete-li ve Windows, vlastníte již spolu se systémem i veškeré nezbytné komerční programové vybavení – totiž webový prohlížeč (Internet Explorer) a jednoduchý textový editor (Poznámkový blok – Notepad či WordPad). Lze samozřejmě používat i jiné webové prohlížeče, a někdy se vám vyplatí i rafinovanější textový editor, jaký bývá součástí komerčních kancelářských balíků. Pravděpodobně tak jako tak na vašem počítači něco takového (např. Microsoft Office) máte. Situací, kdy opravdu potřebujete víc než Notepad, však bude asi méně, než byste čekali, a z používání „chytrého" textového editoru může plynout víc nesnází než užitku (viz 3.1).

Dobrou zprávou je, že mnohé **specializované programy** využívané k analýze biologických sekvencí byly vyvinuty v rámci projektů financovaných z veřejných zdrojů a zůstávají proto veřejně přístupné. Často jsou instalovány na veřejných serverech a dostupné prostřednictvím Sítě, někdy však jde také o freewarové programy k lokální instalaci na váš počítač, které si budete muset sami stáhnout z internetu a nainstalovat.[6] Je proto žádoucí, abyste na vašem počítači měli práva ke změně nastavení a k instalaci programů (tj. lokální administrátorské heslo), což ovšem předpokládá určitou míru zběhlosti v zacházení se systémem, nebo byste aspoň měli být se správcem systému zadobře.

V neposlední řadě nesmíme zapomenout na **data**, která budeme analyzovat. I ta budou nepochybně z převážné většiny pocházet z veřejných databází sekvencí genomových DNA, cDNA, proteinů, konzervativních domén, struktur a motivů a podobně. Těmto zdrojům je věnována 4. kapitola.

Proměnlivá povaha internetu s sebou nutně nese riziko zastarání jakéhokoli konkrétního údaje o veřejně dostupných zdrojích, službách a serverech. Čím větší server či instituce je, tím déle obyčejně vydrží. V hmotném světě je tomu ostatně podobně – Univerzita Karlova, podobně jako dub, celkem prosperuje i po 650 letech, což o menších, byť i špičkových laboratořích, ve

[6] „Freeware" zde znamená, že za legální použití těchto programů pro nekomerční účely nemusíte platit, i když někdy může být vyžadována registrace a závazný souhlas s podmínkami používání.

světě často vedených výzkumníky s termínovanou smlouvou,[7] asi platit nebude. Proto se zde snažím omezit síťový adresář na nutné minimum. Rozsáhlejší a pravidelně aktualizovanou sbírku odkazů najdete stránkách kursu Úvod do bioinformatiky (B130C52), pravidelně vyučovaného na PřF UK, na adrese `http://kfrserver.natur.cuni.cz/bioinfo.html`.

[7] Bližší vhled do tradic organizace výzkumných institucí především ve Velké Británii a USA viz např. Slack (1999/2001).

3. Práce se sekvenčními daty

Přistupme nyní k základům vlastní práce se sekvenčními daty. Tato kapitola bude věnována jednak nejběžnějším formám zápisu nukleotidových i aminokyselinových sekvencí a zacházení s datovými soubory, jednak běžným úkolům, s jakými se dříve či později setká každý, kdo se prakticky věnuje molekulárnímu klonování. Patří sem především

- úpravy zápisu sekvencí do podoby srozumitelné programům pro další analýzy nebo do úhledné podoby pro prezentaci;
- odvození sekvence komplementárního vlákna DNA;
- identifikace kódujících oblastí a překlad DNA sekvence;
- tvorba grafických map ze sekvence nebo z výsledků hrubého restrikčního mapování včetně „klonování *in silico*“;
- návrh primerů pro PCR.

Sekvenčními daty budeme v dalším výkladu rozumět jakoukoli formu zápisu lineární posloupnosti monomerů v molekule biologické makromolekuly – nejčastěji DNA nebo proteinu (se sekvencemi RNA se v praxi setkáváme spíše zřídka, pokud se zrovna nezabýváme virologií). Pro naše potřeby se na sekvenci zpravidla díváme jako na **digitální zápis** posloupnosti jednoznačných znaků odpovídajících jednotlivým typům monomerů po řadě tak, jak se s nimi v molekule setkáváme při postupu ve směru, odpovídajícím směru biosyntézy daného typu molekuly. Pro DNA či RNA je to od 5´ k 3´ konci, pro proteiny od N-konce k C-konci. Standardně se k zápisu sekvencí používají jednopísmenové kódy monomerů tak, jak byly stanoveny Mezinárodní unií pro čistou a aplikovanou chemii (IUPAC), pro zápis aminokyselin nyní již vzácně i třípísmenové zkratky (Tab.3.1.).

Digitalizace výsledků experimentálního stanovení sekvence je záležitostí netriviální, jak snadno nahlédne každý, kdo někdy četl pořadí nukleotidů z autoradiogramu nebo nahlédl do elektroforetogramu pořízeného automatickým sekvenátorem (viz níže a Obr.3.1.). Kromě toho je někdy výhodné vyjádřit jedinou sekvencí konsensus variabilní populace molekul. Proto kódy IUPAC obsahují také možnost zápisu nejednoznačných pozic, a tak i zápis sekvence DNA může obsahovat skoro celou abecedu.[8] A naopak: přihlédneme-li k málo pravděpodobné, leč v principu nikoli vyloučené možnosti existence proteinu složeného výhradně ze zbytků glycinu, alaninu, threoninu a cysteinu, snadno pochopíme, že i sekvence aminokyselin může být zapsána pomocí znaků, za kterými jsme zvyklí vidět nukleotidy. **Informace, zda konkrétní posloupnost znaků vyjadřuje sekvenci nukleotidů nebo aminokyselin, nemůže být obsažena v posloupnosti samé**. Nemůžeme tedy také od žádného programu schopného zacházet se sekvencemi DNA i proteinů očekávat, že pozná nukleotidovou sekvenci od aminokyselinové, pokud mu tuto informaci neposkytneme my – ať již prostřednictvím uživatelského rozhraní, nebo pomocí speciálního formátování vstupních souborů.

Kromě vlastního zápisu sekvencí se budeme postupně setkávat i s jinými typy dat. Budou to zejména

- mapy a anotace, vyjadřující polohu funkčních úseků v rámci sekvence (včetně map genomových a chromozomových);
- vzájemná přiřazení sekvencí (*alignments*) a vzorce (*patterns*), vyjadřující vztahy podobnosti mezi větším počtem makromolekul, případně i jiná vyjádření podobnosti;

[8] Tím však není řečeno, že jakýkoli konkrétní program pro analýzu nukleotidových sekvencí umí zacházet s nejednoznačnými pozicemi.

- údaje o trojrozměrné struktuře proteinů;
- údaje o výsledcích transkriptomických experimentů (*microarrays*).

Soustřeďme se však prozatím na data sekvenční.

Tab. 3.1. IUPAC kódy pro zápis sekvencí nukleotidů a aminokyselin

DNA, RNA	
kód	**báze**
A	adenin
C	cytosin
G	guanin
T	thymin
U	uracil
R	A,G (*purine*)
Y	C,T (*pyrimidine*)
S	G,C (*strong*)
W	A,T (*weak*)
K	G,T (*keto*)
M	A,C (*amino*)
B	C,G,T (*not A*)
D	A,G,T (*not C*)
H	A,C,T (*not G*)
V	A,C,G (*not T,U*)
N	cokoli (*any*)

.	mezera
-	

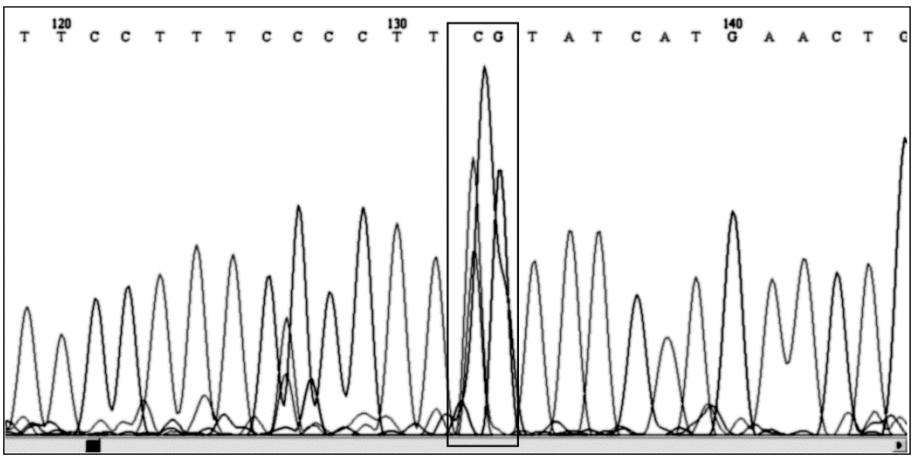
Protein		
kód	**zkratka**	**aminokyselina**
A	Ala	alanin
C	Cys	cystein
D	Asp	aspartát
E	Glu	glutamát
F	Phe	fenylalanin
G	Gly	glycin
H	His	histidin
I	Ile	isoleucin
K	Lys	lysin
L	Leu	leucin
M	Met	methionin
N	Asn	asparagin
P	Pro	prolin
Q	Gln	glutamin
R	Arg	arginin
S	Ser	serin
T	Thr	threonin
V	Val	valin
W	Trp	tryptofan
Y	Tyr	tyrosin
X	Xxx	cokoli
B	Asp, Asn	aspartát nebo asparagin
Z	Glu, Gln	glutamát nebo glutamin

3.1. Zápis sekvencí a běžné formáty datových souborů

Ve své nejpřirozenější podobě – tj. uvnitř živých buněk – jsou sekvence „zapsány“ v posloupnosti monomerů spojených do dlouhých lineárních molekul. Abychom s nimi mohli pracovat, musíme sekvenční informaci přenést do posloupnosti znaků, kterou lze zapsat na elektronické médium. V převážné většině případů se to dnes děje prostřednictvím **sekvenování DNA**, a proto stojí za to se nad touto metodou zamyslet. V technických detailech odkazuji na učebnice molekulární biologie; nás zde bude zajímat pouze osud sekvenční informace samotné.

Sekvenční informace je svou podstatou digitální (a tedy „nehmotná“), a na molekulu DNA můžeme pohlížet jako na médium či nosič, na němž je sekvence zapsána (Zrzavý et al., 2004, s. 69). DNA sama je však současně hmotnou, tělesnou molekulou, která *in vivo* interaguje s řadou buněčných proteinů (což nás v této chvíli až tak moc nemusí zajímat) a *in vitro* v průběhu sekvenování se ne vždy chová ideálně. Běžné metody sekvenování jsou založeny na vytváření souborů definovaných krátkých fragmentů DNA (v rozmezí od několika málo desítek po několik stovek bází) a jejich elektroforetickém dělení s přesností, která umožňuje odlišit fragmenty, které se liší o jediný nukleotid. Elektroforéza však pracuje s velkými

populacemi molekul, které se chovají statisticky, a proto z principu generuje data analogová, nikoli digitální – pokud je kvantifikujeme, dostaneme vždy aspoň mírně „rozmyté“ křivky. Chování fragmentů DNA v průběhu dělení odráží navíc kromě jejich velikosti (molekulové hmotnosti, náboje) i tvar, zejména náchylnost k vytváření stabilních sekundárních struktur. Klasické „ruční“ sekvenování, jehož výstupem byly půlmetrové autoradiogramy, z nichž bylo nutno sekvenci vyčíst a ručně naťukat do počítače, nedovolovalo na tuto skutečnost zapomenout. Dnes však většinou sekvenujeme za pomoci automatických přístrojů, a navíc často v komerčních servisech, které přijmou zkumavku s DNA a vrátí datový soubor či soubory se sekvencí (samozřejmě spolu s fakturou). Uživatel snadno podlehne pokušení přijímat textový výstup ze sekvenátoru jako autentickou sekvenci, a nikoli jako nejlepší možnou, ale ne nutně správnou, variantu čtení analogového záznamu. Vždy se však vyplatí prohlédnout si a uchovat i primární elektroforetogram, který se u nejběžnějších sekvenátorů firmy Applied Biosystems skrývá v souboru s příponou `*.abi` nebo `*.ab1`, a dát si pozor na možná problematická místa (Obr.3.1.).[9]



Obr.3.1. Ukázka analogového zápisu elektroforetogramu a jeho převodu na digitální sekvenci IUPAC znaků ve volně dostupném programu BioEdit (o programu více v části 8.2.). Oblast v rámečku odpovídá zřejmě místu náchylnému k tvorbě sekundárních struktur v molekulách DNA a kompresi proužků. Její čtení zdaleka není jednoznačné, nejen co do druhu bází, ale ani co do jejich počtu (v rámečku by se dala najít i sekvence ACG). V původním výstupu jsou stopy odlišeny barevně.

Sekvenování jiných typů molekul, zejména proteinů, je dnes spíše raritou, používanou pouze ve výjimečných situacích (většina proteinových sekvencí v databázích je odvozena teoreticky ze sekvencí příslušných cDNA). I když metodická stránka věci se liší, ani tam se nemůžeme ubránit obdobným nesnázím pramenícím z nezbytného „analogového“ kroku a následné zpětné digitalizace.

V nejjednodušší digitalizované podobě je sekvence (ať už nukleotidů nebo aminokyselin) zapsána jako prostý řetězec IUPAC znaků (které naštěstí jsou současně znaky ze standardní sady ASCII) v textovém souboru. Tento formát označujeme jako **surová data** (*raw data*, *raw format*). Programy, které dovedou surová data přijmout, zpravidla tolerují roztažení sekvence na více řádků a někdy i mezery; prázdným řádkům je lépe se vyhnout. Udržíme-li délku řádků pod mírně starobylým limitem 255 znaků, jistě nic nezkazíme, a možná si ušetříme nesnáze.

[9] Pokud nemáme opravdu velké štěstí a špičkovou sekvenovací reakci, musí se obvykle brát s rezervou prvních 50 bp a vše nad 450-600 bp (t.j. je radno sekvenci utnout před místem, kde se začne objevovat výrazná frakce nerozlišených bází). Toto je třeba mít na paměti, pokud neznáme elektroforetogram, a máme k dispozici jen digitální zápis, což bývá pravidlem u sekvencí získaných v jiné laboratoři (např. EST sekvence z databází – viz 4.1).

Formát surových dat se však pro dlouhodobé uchovávání sekvencí příliš nehodí, protože neumožňuje uložit dodatečné informace (např. o původu vzorku či typu sekvence) jinam než do názvu souboru, jehož délka je pro mnohé programy omezena (někdy i na tradiční 8+3 znaky zděděné po operačním systému DOS).

Pro uchovávání sekvenčních dat spolu s dalšími průvodními informacemi byla vyvinuta celá řada formátů, některé z nich velmi rafinované. Zejména velké databáze, jako je EMBL, GenBank a DDBJ (viz 4.1.), používají několik vzájemně převoditelných formátů, které poskytují prostor i pro anotaci funkčních úseků sekvencí a překryvů s jinými sekvenovanými fragmenty, literárních citací a podobně. Příkladem takového formátu je interní formát databáze EMBL či tzv. *GenBank flatfile* (viz 4.3, Obr.4.2. a Obr.4.3.). Tyto složité formáty jsou optimalizovány pro vyhledávání podle řady kritérií včetně obsahu anotace a dalších přidružených informací. Pro další zpracování sekvencí však nejsou příliš praktické, protože program, který by si měl vybírat relevantní data (čistou sekvenci) by musel mít speciální vstupní filtry pro všechny varianty formátu všech databází. Krom toho uživatel nevybavený specializovaným softwarem sice dokáže ze souboru vyčíst či zkopírovat data, ale prakticky nepřipadá v úvahu, že by soubor editoval nebo dokonce napsal – požadavky formátu jsou vskutku velmi striktní i pokud jde o počty znaků a mezer, v některých případech i kontrolní součty (*checksums*). Proto se při praktickém zacházení s daty používají formáty jednodušší, jaké lze napsat i přímo z klávesnice; výsadní postavení mezi nimi má formát **FASTA** (Obr.3.2.; název je odvozen od programu FASTA – viz 6.1.), vhodný jak pro sekvence nukleotidové, tak i pro sekvence aminokyselin.

```
>OsFH15
MLARWLLVLLLLLPVSWCHQDRHGRRHYPRRWRSSGSRRELHEPLFPLENAPALPPPPPP
PPAPFFPFLPDSAPPQLPPPVTTPAPAGGAGDGGTDAGAAATGDASSSSSSSASPHPTAP
ANISYMAMPIYHSAPLRSFLSSHRLLTVLLPVAAVLAAVLAAALVYLLTRRRRCSKGEPH
AAHTKAVLLSPGNSTALYDGDHDQHGRGSTATAASSASSPELRPMPPLPRQFQQTRTSMP
STSQTIHEAGAEDKRAPPPQSVRPPPPPPPPPPPPPMPPRTDNASTQAAPAPPPPLPRAG
NGSGWLPRRYTERAAPTVIRASAGAVHPEESPARASPEEKAADAAARPKLKPLHWDKVRP
ASSGRPTVWDQLKASSFRVNEEMIETLFVSNSTRRASKNGVKEANAACCNQENKVLDPKK
SQNIAIMLRALDATKEEVCKALLDGQAESLGTELLETLLKMAPSREEEIKLKEFREDAVS
KLGPAESFLKAVLAIPFAFKRVEAMLYIANFDSEVDYLKTSFKTLEAACEELRGSRLFHK
ILDAVLKTGNRMNTGTNRGNASAFKLDALLKLVDVKGADGKTTLLHFVIEEIVKSEGASI
LATGQTSNQGSAIADDFQCKKVGLRIVASLGGELGNVKKAAGMDSDTLASCVAKLSAGVS
KISEALQLNQQLGSDDHCKRFRASIGEFLQKAEAEITAVQAQESLALSLVRETTEFFHGD
SVKEEGHPLRIFMVVRDFLTVLDHVCKDVGRMNERTAIGSSLRLENAPVLARFNAVQPSS
SEEESSSS
```

Obr.3.2. Sekvence proteinu ve formátu FASTA.

Datový soubor ve formátu FASTA je textový soubor, jehož první řádka začíná znakem > (větší než), za nímž následuje „hlavička“ obsahující název sekvence, anotaci a případné další údaje, které nejsou součástí sekvence samotné. První skupina alfanumerických znaků v „hlavičce“ (až po první mezeru) představuje jedinečný identifikátor sekvence (je vhodné umisťovat sem např. název genu či klonu), na zbytek můžeme pohlížet jako na volitelný komentář. Na dalších řádkách pak následuje surová sekvence. Délka prvního řádku formálně není omezena, ale méně bývá více (pro většinu programů je bezpečná délka do 255 znaků, alespoň pokud si do prvního řádku nenecháme přehorlivým textovým editorem vložit zalomení). Vzhledem k tomu, že programy, které zpracovávají sekvence ve formátu FASTA, obsah hlavičky ignorují, případně používají jen prvních několik znaků jakožto název sekvence, neměl by být jeho obsah nijak omezen, ale některé speciální znaky přece jen mohou

v některých programech působit potíže.[10] Sekvence sama by měla být prosta prázdných řádků a pokud možno i mezer; některé programy mezery tolerují, a jiným nevadí ani číslice v sekvenci. Měli bychom si však od začátku zvykat pracovat s čistým FASTA formátem bez mezer a cizích znaků.

V rámci formátu FASTA je přípustné, a někdy výhodné, spojovat několik sekvencí do jednoho souboru. Hodí se to například k tomu, abychom tyto sekvence mohli naráz importovat do programů pro následné zpracování. V tom případě se mezi sekvencemi toleruje prázdný řádek, i když není nezbytností. Sekvence však musejí být téhož typu, všechny buď nukleotidové, nebo proteinové, protože FASTA formát obecně neumožňuje specifikovat typ sekvence uvnitř sekvenčního souboru. Pro některé programy, jako je dále diskutovaný MACAW (8.2.), se typ dat musí specifikovat pomocí přípony (extenze) textového FASTA souboru (`*.nt` je sekvence nukleotidů, `*.aa` sekvence aminokyselin),[11] jinde určuje typ dat uživatel prostřednictvím interaktivního rozhraní programu.

Příkladem poměrně jednoduchého formátu, který elegantně řeší problém určení typu sekvence, je formát **PIR/NBRF** (podle Protein Information Resource/National Biomedical Research Foundation; Obr.3.3.). který sice přijímá jen málo programů, ale stojí zde za zmínku přinejmenším proto, abychom se vyvarovali možné záměny s na první pohled podobným formátem FASTA. První řádka začíná rovněž znakem > (větší než), za nímž následuje dvouznakový kód určující typ sekvence (např. `P1` je protein, `DL` lineární DNA - *DNA-linear*), středník a zkrácený název sekvence (v případě záznamu z databáze přístupový kód – *accession*, viz 4.2). Druhá řádka obsahuje plný název a anotaci, od třetí řádky následuje sekvence, ukončená hvězdičkou.

```
>P1;CRAB_ANAPL
ALPHA CRYSTALLIN B CHAIN (ALPHA(B)-CRYSTALLIN).
  MDITIHNPLI RRPLFSWLAP SRIFDQIFGE HLQESELLPA SPSLSPFLMR
  SPIFRMPSWL ETGLSEMRLE KDKFSVNLDV KHFSPEELKV KVLGDMVEIH
  GKHEERQDEH GFIAREFNRK YRIPADVDPL TITSSLSLDG VLTVSAPRKQ
  SDVPERSIPI TREEKPAIAG AQRK*
```

Obr.3.3. Sekvence proteinu ve formátu PIR/NBRF. Mezery uvnitř řádků ani na jejich začátku (odsazení) nejsou povinné.

Konvenční formáty existují i pro jiné typy dat, než jsou samotné sekvence – např. krátké variabilní sekvenční motivy (7.1.), mnohočetná přiřazení (8.1.) nebo „evoluční stromy“ – *dendrogramy* (10.1.). Budeme se jim dále věnovat v příslušných kapitolách.

K tomu, abychom byli schopni uvést sekvenci do počítačově zpracovatelné podoby (tj. např. do formátu FASTA), nám v principu stačí libovolný textový editor. Zdůrazňuji, že pouze v principu – např. odstraňování mezer a číslic, které některé standardní databázové formáty obsahují (viz 4. kapitola) je práce poměrně nudná, a přesahuje-li délka sekvence několik málo

[10] Chceme-li např. zpracovávat FASTA soubory programem MACAW (viz 8.2.), měli bychom se v prvním řádku vyvarovat znaku | (svislé dělítko), nebo aspoň – v rozporu se striktními pravidly formátu – umístit název sekvence až za tento znak, protože program ignoruje vše, co znaku předcházelo. Tato poznámka se může zdát odtažitou, avšak pouze do chvíle, než zjistíme, že FASTA výstupy z databází problematický znak běžně obsahují. Jiné programy (např. MUSCA – viz 8.2.) jsou citlivé třeba na výskyt mezer (prázdných znaků) na konci řádků.

[11] Abychom mohli přípony snadno měnit, je radno ve Windows zrušit nastavení „skrýt přípony souborů známých typů“ a zaregistrovat si soubory `*.nt` a `*.aa` pro otvírání v textovém editoru.

stovek párů bází, je radno si ji automatizovat. O tom za chvíli; zdržme se nyní u zdánlivě samozřejmého úkonu, kterým je formátování sekvence v textovém editoru, protože právě zde mohou číhat úskalí. Zejména při použití rafinovanějších editorů, nežli je Poznámkový blok (v našich krajích nejčastěji MS Wordu) je třeba dbát na to, aby

- sekvenční soubory byly, bez ohledu na příponu, uloženy ve formátu „pouze text“, ať si editor jakkoli vyhrožuje možnou ztrátou formátování;
- při úpravách souboru nebyly používány tabelátory a jiné z hlediska cílových programů cizí znaky;
- v editoru byly vypnuty funkce „automatické opravy“ a „automatický text“, které mají sklon zasahovat do sekvencí a „opravovat“ v nich pravopis, i funkce „inteligentní vyjímání a vkládání“. Kritické je toto nastavení zejména tam, kde pracujeme se zápisem přiřazení (viz zejména kap. 5 a 8).

Existuje řada volně dostupných softwarových nástrojů pro **formátování sekvencí** a vzájemné převody mezi jednotlivými formáty. Z těch starších a zasloužilých možno jmenovat např. veřejně dostupnou službu *BCM Sequence Utilities* na serveru Baylor College of Medicine.[12] Jde však o úkon natolik základní a častý, že není tak docela slušné zatěžovat jím cizí servery. Soubor nástrojů pro převod sekvencí nejrůznějších formátů (včetně EMBL a GenBank) na FASTA je však též součástí mimořádně zdařilého freewarového balíku programů *The Sequence Manipulation Suite* – SMS (Stothard, 2000).

3.2. Mapování, restrikce, translace a klonování *in silic(i)o*

Sekvence DNA sama, byť i počítačově čitelná, lidskému uživateli mnoho neřekne. I když třeba nechceme teoreticky zkoumat evoluční souvislosti, v běžné laboratorní praxi se potřebujeme v sekvenci vyznat – tedy např.

- zjistit, zda náš fragment DNA kóduje protein;
- předpovědět, jak bude molekula DNA o známé sekvenci štěpena restrikčními endonukleázami;
- nakreslit restrikční mapu – ať již existujícího fragmentu DNA o známé sekvenci, nebo konstruktu, který teprve hodláme vytvořit;
- navrhnout primery pro amplifikaci úseku známé sekvence pomocí PCR.

Ve všech těchto situacích můžeme opět využívat volně dostupné softwarové nástroje, často fungující na veřejně dostupných internetových serverech. Pokud však zmíněné úkony vykonáváme velmi často, měli bychom si, pokud to je rozumným způsobem možné, pořídit vlastní kopie programového vybavení – ať už instalované na uživatelských počítačích, nebo na serverech v místní síti. Autoři původního softwaru či provozovatelé veřejných serverů mnohdy nabízejí instalační verze svých programů zdarma ke stažení – nejen z velkorysosti, ale také proto, aby odlehčili vlastní síti od přemíry návštěvníků využívajících veřejnou službu. Lokální kopie softwaru nás navíc činí nezávislými na prostupnosti sítě, která ve špičce může být omezením.

V následující pasáži upozorňuji na několik vybraných osvědčených síťových zdrojů. Nepodávám zde podrobnější návody „jak se co dělá“. Jde mi spíše o nasměrování uživatele na

[12] Odkazy na veškeré zmiňované online nástroje naleznete na `http://kfrserver.natur.cuni.cz/ bioinfo.html`; pozornosti čtenáře lze doporučit též řadu odkazů dostupných prostřednictvím `http://bioinformatics.org`.

příslušný nástroj – všechny jsou opatřeny dost kvalitní nápovědou, která i začátečníkovi umožní samostatnou orientaci.

Potřebujeme-li **vytvářet restrikční mapy či překládat sekvence DNA do proteinu** jen občas, můžeme spoléhat na zdroje ve světové pavučině WWW.[13] Užitečný soubor nástrojů poskytují např. BCM Sequence Utilities na stránkách Baylor College of Medicine, kde najdeme i programy pro formátování sekvencí či odvození reverzně komplementárního řetězce DNA. Restrikční mapy lze vytvářet např. též na stránkách programu WebCutter a řady dalších. Další programy pro zjištění kódující kapacity DNA a překlad do sekvence aminokyselin poskytují např. stránky významného, původně švýcarského a nyní mezinárodního, bioinformatického serveru ExPASy (*Expert Protein Analysis System*; viz též kap. 7).

Podobu veřejně dostupných bioinformatických zdrojů v posledních letech významně změnil **projekt `bioinformatics.org`**, prezentovaný na stejnojmenných webových stránkách. I když jeho cílem je především vytvoření otevřeného fóra k výměně nápadů a zkušeností programátorů zaměřených na bioinformatiku, nabízí též některé velmi zajímavé nástroje pro běžného uživatele. Webový portál **SeWeR** (*Sequence analysis using Web Resource*; Basu, 2001) nabízí společné a velmi přehledné uživatelské rozhraní pro programy na řadě externích serverů, poskytujících služby od restrikčního mapování a translace až po značně pokročilé analytické nástroje.

Na serveru `http://bioinformatics.org` lze nalézt též odkaz na pozoruhodný softwarový balík – **The Sequence Manipulation Suite** (SMS; Stothard, 2000) – obsahující sadu nástrojů pro formátování, mapování, translaci a prezentační tisk sekvencí. Nástroje SMS prakticky pokrývají všechny běžné potřeby probírané v této kapitole, s výjimkou kreslení grafických map a návrhu PCR primerů. Celý balík je přitom napsán formou sady webových `*.html` stránek a skriptů v jazyce Java, což zaručuje kompatibilitu s běžnými operačními systémy, a je volně zdarma a bez registrace stažitelný pro použití na serverech i na vlastním počítači uživatele.[14] Pro lokální použití přitom nejsou nezbytná administrátorská práva (program nezapisuje na disk a dobře funguje třeba i z CD), a jeho uživatelské rozhraní je velmi přehledné a „samovysvětlující“. Proto lze balík SMS vřele doporučit i začátečníkům jako zcela základní nástroj; místní instalace je výhodná mj. i proto, že umožňuje „osahat“ si program důkladně třeba doma, bez nutnosti připojení k internetu.

Na tomtéž serveru sídlí též nástroj pro **konstrukci grafických map** plazmidů a jiných fragmentů DNA bez znalosti sekvence, pouze na základě známé velikosti fragmentů DNA – **Savvy**. Jeho výstup v grafickém formátu `*.svg` (*Scalable Vector Graphics*) vyžaduje instalaci pluginu firmy Adobe. Savvy však nedovede vytvořit mapu ze známé sekvence. Zdarma (aspoň zatím) lze ale ze stránek `http://www.acaclone.com` získat program **pDRAW32**,[15] který navíc dokáže i simulovat výsledky gelové elektroforézy fragmentů DNA

[13] Odkazy na stránkách kursu bioinformatiky – viz předchozí poznámka.
[14] SMS server na PřF UK je dostupný ze stránek kursu bioinformatiky.
[15] Tento program, jako jediný zde zmiňovaný, není freeware, nýbrž „beerware“, což znamená, že potkáte-li náhodou někde autora – dánského molekulárního biologa Kjelda Olesena, měli byste ho pozvat na pivo (případně i na dvě – jedno za mne). Po instalaci se pDRAW32 pravidelně dožaduje upgradu a bez něj přestává fungovat, což naznačuje, že autor si nechává otevřená zadní vrátka pro případné budoucí zpoplatnění.

a kreslit mapy produktů „virtuálního klonování“.[16] Nástroj pro kreslení a anotaci map plazmidů je rovněž součástí programu BioEdit, který bude podrobněji probírán v 8. kapitole.

Program pDRAW32 obsahuje též jednoduchý nástroj pro poslední ze zde probíraných běžných praktických úkonů – **návrh primerů pro PCR**. Pokud bychom měli brát v úvahu i optimalizaci místa nasednutí primeru, představoval by návrh primerových sekvencí již poměrně složitou analýzu. V praxi se však zpravidla omezujeme na predikci teploty tání a případných nežádoucích sekundárních struktury (vlásenek) a dimerů oligonukleotidů, jejichž sekvenci dodal uživatel. Uživatelské rozhraní analýzy primerů v pDRAW32 však není nejpříjemnější, a i pro rutinní práci stojí za to používat specializované, a tudíž lépe vyladěné, veřejně dostupné nástroje na webových serverech, jako je například program **NetPrimer** na stránkách firmy Premier Biosoft (`http://www.premierbiosoft.com/netprimer/`).

Až dosud jsme se zaměřili na práci s izolovanými sekvencemi, jejichž původ nás nezajímal. Přistupme nyní k tématice, která je jádrem vlastní bioinformatické práce – ať již za bioinformatiku považujeme jen tu „vysokou“, nebo i drobnou domácí výrobu. Řeč bude o využívání sekvenčních databází a o přístupu k téměř bezbřehým možnostem, které v dnešní době skýtají veřejné zdroje dat.

[16] Vznešeněji zní „klonování *in silico*“ podle vzoru *in vivo* a *in vitro* (a proti duchu latiny, který by velel užívat spojení *in silicio*, protože křemík je *silicium*). Spojení „virtuální klonování“ je ale možná výstižnější, neb poukazuje též na skutečnost, že rukama jsme dosud na daném konstruktu nic neudělali.

4. Veřejně dostupné zdroje dat

Za svůj současný rozmach bioinformatika možná vděčí i tomu, že základy oboru byly položeny v době, která se nám dnes může jevit jako idylická a idealistická. Vědecké úsilí – snad s výjimkou vojenského výzkumu a bezprostředně komerčních aplikací – bylo ještě počátkem 80. let minulého století vnímáno jako záležitost především veřejná (a do značné míry financováno z veřejných rozpočtů nebo alespoň z neziskových zdrojů), a měřítkem kvality výzkumu byly publikace, nikoli patenty. Neochota volně sdílet primární data, na nichž publikace stojí, mohla svědčit toliko o pochybné kvalitě oněch dat a tedy i týmu, který je vyprodukoval, a nikoli o snad tom, že si firma, která výzkum financuje, chce udržet kontrolu nad oběhem výsledků, které by mohly mít komerčně zajímavý dopad. Asi jen díky této atmosféře bylo vůbec možné, že se volné sdílení primárních sekvenčních dat stalo standardní praxí, od níž (naštěstí) snad již nelze ustoupit. Významnější časopisy dnes vyžadují, aby autoři, kteří publikují sekvenční data, tato data volně zpřístupnili odborné veřejnosti.

V případě sekvencí DNA se to standardně děje prostřednictvím tří vzájemně provázaných **databází** GenBank/EMBL/DDBJ, o kterých bude ještě řeč níže (4.1.); kromě nich však existují dnes již desítky dalších veřejně dostupných zdrojů dat[17] specializovanější povahy – např. databáze sekvencí odvozených od virů, repetitivních elementů, regulačních sekvencí, variant genů pro lidské imunoglobuliny a podobně. Krom toho lze nalézt i množství dalších databází zahrnujících kromě sekvencí (nebo výhradně) také biologická data nesekvenční povahy, například údaje o expresi genů, epigenetických modifikacích chromatinu, sestřihu mRNA, posttranslačních úpravách proteinů, aktivních místech enzymů. Za zmínku stojí též specializované databáze zaměřené na detailní dokumentaci specifických genových rodin či jednotlivých experimentálně studovaných organismů. Počet zavedených veřejně dostupných databází v současnosti přesahuje 600, a vzhledem k bouřlivému tempu sekvenování neustále narůstá (Galperin, 2004). Přehled aktuálního stavu bývá jednou ročně zveřejňován ve specializovaném čísle časopisu Nucleic Acids Research, spolu se shrnujícími články o vybraných významnějších databázích.

Obecně rozlišujeme **databáze moderované (*curated*) a nemoderované**. Do nemoderovaných databází může přispívat kdokoli – jedinou podmínkou je, že data jsou vhodného typu (tj. sekvence v případě sekvenčních databází) a ve správném strojově čitelném formátu. Správce databáze do dat nijak nezasahuje, ani nekontroluje kvalifikaci a kompetenci přispěvatele, nanejvýš poskytuje přispěvatelům nástroje pro kontrolu běžných chyb, jako je třeba přítomnost kontaminujících vektorových sekvencí. Zveřejnění výsledků v nemoderované databázi je tedy obdobou „papírové“ nerecenzované publikace. Naproti tomu moderované databáze přijímají data výběrově podle kritérií stanovených správcem databáze – u nesekvenčních databází může být podmínkou např. předchozí zveřejnění experimentálních výsledků v recenzovaném časopise, popřípadě správce databáze vede vlastní recenzní řízení. Příkladem moderované databáze je historicky významná databáze experimentálně ověřených proteinových sekvencí Swissprot, podrobněji probíraná v části 4.5.

Vzhledem k zaměření této práce se nadále budeme věnovat především veřejně dostupným zdrojům sekvenčních dat, jak nukleotidových, tak proteinových; s jinými databázemi, jako jsou např. databáze trojrozměrných struktur proteinů, se budeme seznamovat průběžně.

[17] „Veřejnou dostupností“ rozumějme dostupnost prostřednictvím Internetu komukoli zdarma a bez nutnosti registrace.

4.1. „Velká trojka“primárních databází nukleotidových sekvencí – GenBank, EMBL, DDBJ

Standardem ve zveřejňování nukleotidových sekvencí je trojice **primárních databází** mezinárodního konsorcia (*International Nucleotide Sequence Database Collaboration*), zahrnujícího americkou databázi **GenBank** (Benson et al., 2004), evropskou **EMBL** (*European Molecular Biology Laboratory Data Library*; Kulikova et al., 2004) a japonskou **DDBJ** (*DNA Data Bank of Japan*; Miyazaki et al., 2004). Tyto databáze si denně vyměňují veškeré změny v datech a prakticky se neustále vzájemně zálohují. Jejich obsah je tedy až na přírůstky z posledních hodin totožný. Vzhledem k exponenciálně rostoucí světové produkci sekvenčních dat, z nichž převážná většina dříve či později skončí v databázích Velké trojky, je údržba primárních databází úctyhodným úkolem, protože struktura databáze, software i hardware musejí být dimenzovány s ohledem na vývoj v dohledné budoucnosti. Přitom v současnosti se objem sekvenčních dat zdvojuje s „generační dobou“ přibližně dva roky. V primárních databázích bylo v září 2003 uloženo 27,2 x 10^6 záznamů o zhruba 33 x 10^9 párů bází, pocházejících z přibližně 150 000 různých organismů (Kulikova et al., 2004).[18]

Primární databáze jsou přístupné prostřednictvím webových rozhraní (viz 4.4); kromě toho lze prostřednictvím FTP bezplatně získat a lokálně instalovat celou databázi ve stavu odpovídajícím určitému datu (tzv. **vydání** – *release*, publikované v případě EMBL 4x ročně, u GenBank každé dva měsíce). Význam to má zejména pro pracoviště, která se zabývají vývojem a aplikací nových či mimořádně výpočetně náročných prohledávacích algoritmů, které nelze používat přes webové rozhraní. Změny v databázi od posledního vydání jsou označovány jako **přírůstky** (*updates*).

Stinnou stránkou obrovského objemu dat procházejících databázemi Velké trojky je to, že databáze nutně musejí být nemoderované; pro konkrétní databázový záznam tedy neexistuje žádná záruka, že data jsou kvalitní. Krom toho tyto databáze fungují jako **depozitář** (*repository*) nukleotidových sekvencí, což znamená, že právo upravovat záznamy v databázi nebo dokonce záznam odstranit má pouze původní autor, a to i v případě, že bylo prokázáno že data jsou mylná – např. sekvence pocházející z mikrobiální kontaminace tkáňových kultur anotovaná jako savčí či sekvence houbového patogena rostlin popsaná jako rostlinná. S údaji z primárních databází je proto nutno zacházet obezřetně, zejména pokud nejsou podloženy řádnou publikací.

V primárních databázích se můžeme setkat s několika **typy datových záznamů**. Následující výčet shrnuje jen ty nejběžnější:

- standardní originální nukleotidové sekvence získané pokud možno kvalitním sekvenováním fragmentů genomové DNA či cDNA získané reverzní transkripcí mRNA
- sekvence EST (*expressed sequence tags*) – tj. částečné sekvence konců jinak necharakterizovaných cDNA, které jsou obvykle nižší kvality než sekvence standardní
- dosud neposkládané a neanotované „surové“ sekvence ze sekvenování genomů (HTGS – *high throughput genome sequencing*)
- referenční sekvence již poskládaných a více či méně anotovaných kompletních genomů

[18] Nezanedbatelnou část z nich však zřejmě tvoří „nekultivované environmentální vzorky“ planktonních a půdních mikroorganismů.

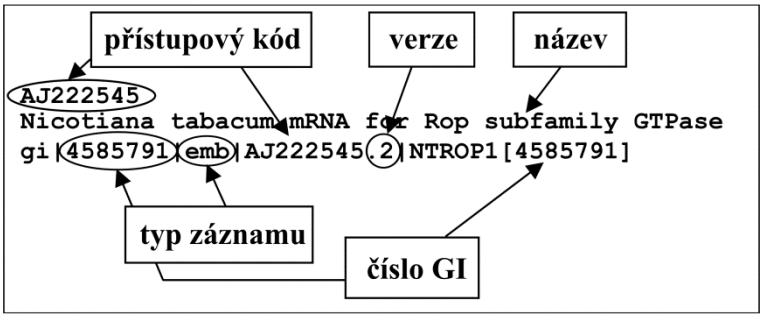
- sekvence anotované jinými než původními autory (TPA – *third party annotation* – viz 4.3.).

Součástí anotace mnoha záznamů v primárních databázích jsou také **předpokládané sekvence proteinů** kódovaných sekvenovaným fragmentem DNA, dodané autory databázových záznamů. Spojení mezi původní DNA sekvencí a jejím překladem přitom samozřejmě zůstává v databázi zachováno. Tak v GenBank je sekvence DNA zadávána a interně uchovávána společně s aminokyselinovým překladem v rámci jediného záznamu typu „nuc-prot". Sekvence těchto hypotetických proteinů vytvářejí odvozenou databázi (trEMBL – *translated EMBL*, případně GenPept), která je strukturována podobně jako primární databáze nukleotidová a kterou lze rovněž prohledávat.

4.2. Identifikační čísla, verze a typy datových souborů

Orientaci v primárních databázích umožňuje jednotný **systém jedinečných identifikátorů** databázových záznamů, sdílený mezi všemi třemi databázemi Velké trojky.

Každý datový záznam obdrží okamžitě po zařazení do databáze tzv. **přístupový kód** (*accession*), který se skládá z proměnlivého počtu písmen a číslic (podle toho, kdy a přes kterou primární databázi byl záznam přijat). Přístupový kód je jakýmsi ekvivalentem rodného čísla záznamu, zůstává nezměněn po celou dobu jeho existence a umožňuje kdykoli příslušný databázový záznam vyhledat. Současně s publikací v GenBanku pak záznam získá rovněž jedinečné **číslo GI** (*GenBank Identifier*). Mezi přístupovým kódem a číslem GI není zjevný vztah, podobně jako mezi rodným číslem a číslem občanského průkazu (Obr.4.1.).



Obr.4.1. Příklad identifikátoru („hlavičky") databázového záznamu pro sekvenci cDNA NtRop1 (formát GenBank; viz též Obr.4.2.).

Autoři záznamu (a nikdo jiný) mají možnost záznam dodatečně upravovat, takže se časem v databázi mohou vystřídat různé **verze**. První verze zaslaná do databáze má pořadové číslo 1. Číslo verze se uvádí jako „extenze" přístupového kódu, oddělená tečkou (Obr.4.1.). Změna verze má rovněž za následek nepredikovatelnou změnu čísla GI. Starší verze se uchovávají, dokonce i v případě, že autoři s ostudou stáhli příslušnou publikaci i se sekvencí (nelze nevzpomenout výrok M. Bulgakova, že rukopisy nehoří). V EMBL jsou staré záznamy přístupné prostřednictvím specializované databáze verzí (Leinonen et al., 2004), v GenBanku přes přímý odkaz v nové verzi a přes zvláštní odkaz na historii revizí (příklad viz Tab.4.1). Neplatné („mrtvé") verze v GenBanku, tj. záznamy, které byly buď nahrazeny novější verzí, nebo staženy, jsou opatřeny upozorněním na tento stav, a nedá se v nich vyhledávat.

„Hlavička“ záznamů v databázi GenBank obsahuje také informaci o typu dat a cestě, kudy se do databáze dostala – tak `emb` znamená záznam původem z EMBL, `gb` GenBank, `dbj` záznam z DDBJ, `dbest` záznam typu EST a podobně. Původ dat je obdobně vyznačen i u odvozených databází proteinových, které bývají pro účely prohledávání často spojovány i s jinými specializovanými databázemi, např. Swissprot či PIR (viz 4.5).

Tab.4.1. Historie revizí záznamu s přístupovým kódem AJ222545 (viz též Obr. 4.1. a 4.2.)

GI	Version	Update Date	Status	*Poznámka*
4585791	2	Aug 8 2003 8:25 AM	Live	*Aktuální verze*
4585791	2	Dec 14 2002 12:41 AM	Dead	*Změna formátu záznamu či databáze (obsah a GI zůstává)*
4585791	2	Dec 7 2002 12:04 AM	Dead	
4585791	2	Oct 17 2002 5:25 AM	Dead	
4585791	2	Aug 3 1999 12:16 AM	Dead	
4585791	2	Apr 16 1999 2:50 PM	Dead	
4585791	2	Apr 16 1999 1:40 PM	Dead	*Autorská oprava sekvence – změna verze a GI*
2598008	1	Mar 9 1999 6:24 AM	Dead	*Změna formátu záznamu (obsah a GI zůstává)*
2598008	1	Nov 8 1997 3:58 AM	Dead	*Původní vložení záznamu do databáze*

Vzhledem k tomu, že primární databáze jsou nemoderované, může se stát, že tentýž úsek DNA nezávisle sekvenuje více skupin, a v databázi se pak ocitne v několika podobách s různými přístupovými kódy a často i pod různými názvy. Kromě přímého porovnání sekvencí (viz kap. 5 a 6) nám informace o identitě opakovaně sekvenovaných fragmentů může poskytnout i autorská anotace, zejména u novějších záznamů týkajících se organismů, jejichž genom již byl kompletně sekvenován a kde existuje systém konvenční notace chromozomálních lokusů. Tak například pro *Arabidopsis thaliana* je každému teoreticky předpovězenému otevřenému čtecímu rámci přiřazen tzv. **AGI kód** (od *Arabidopsis Genome Initiative*) ve formátu AtXgYYYYY, kde X je číslo chromozómu (v rozmezí 1 – 5) a Y je pětimístná souřadnice genu v rámci chromosomu. Vztah mezi AGI kódem a konvenčním názvem genu, pokud tento existuje, lze zjistit v některé ze specializovaných databází zaměřených na *Arabidopsis* (viz 4.5).

4.3. Jak se data dostanou do databází?

Jak již bylo řečeno, původní data do primárních databází může uložit kdokoli, pokud dodrží formální požadavky na formát datových souborů. Formální stránku si správci databází zajišťují pomocí specializovaných softwarových nástroje pro přípravu vstupních souborů. Jak EMBL, tak GenBank používají stejné **vstupní nástroje** – webové rozhraní **Webin** a volně stažitelný lokální program **Sequin** k instalaci na vlastní počítač uživatele. Oba nástroje nabízejí celkem samovysvětlující (ovšem samozřejmě anglickou) sérii formulářů k vložení údajů o autorech, publikaci, zdroji sekvenované DNA a jejích vlastnostech. Samotná sekvence – jak DNA, tak i předpokládaných proteinových produktů – se zadává z předem připravených souborů. Zejména pro nezkušeného uživatele je radno nainstalovat si lokální program, protože na rozdíl od webového rozhraní umožňuje přípravu databázového vstupu kdykoli přerušit a případné chybějící údaje dohledat, upravit vstupní soubory a podobně. Výstupem lokálního programu pak je soubor ve speciálním formátu, který lze zaslat správci databáze e-mailem na adresu uvedenou na webových stránkách databáze (přístupných prostřednictvím `http://www.ebi.ac.uk` pro EMBL, `http://www.ncbi.nlm.nih.gov` pro GenBank). Autor obratem obdrží potvrzení o přijetí souboru a přístupový kód právě zaslané sekvence.

Uložení dat v databázích Velké trojky je dokonce **podmínkou přijetí publikace** u řady časopisů (prakticky u všech impaktových). Jako důkaz toho, že sekvence byla uložena do databáze, slouží odkaz na přístupový kód v článku určeném k publikaci. Vzhledem ke kompetitivní povaze současné vědy ovšem nemusí být vždy radno zveřejňovat data dosud nepublikovaná. Databáze proto poskytují možnost zaslanou sekvenci uložit bez zveřejnění (*on hold*). Sekvence zůstává neveřejná, dokud autor neoznámí, že smí být zveřejněna, popřípadě do data, které autor musí určit, pokud nesouhlasí s okamžitým zveřejněním.[19]

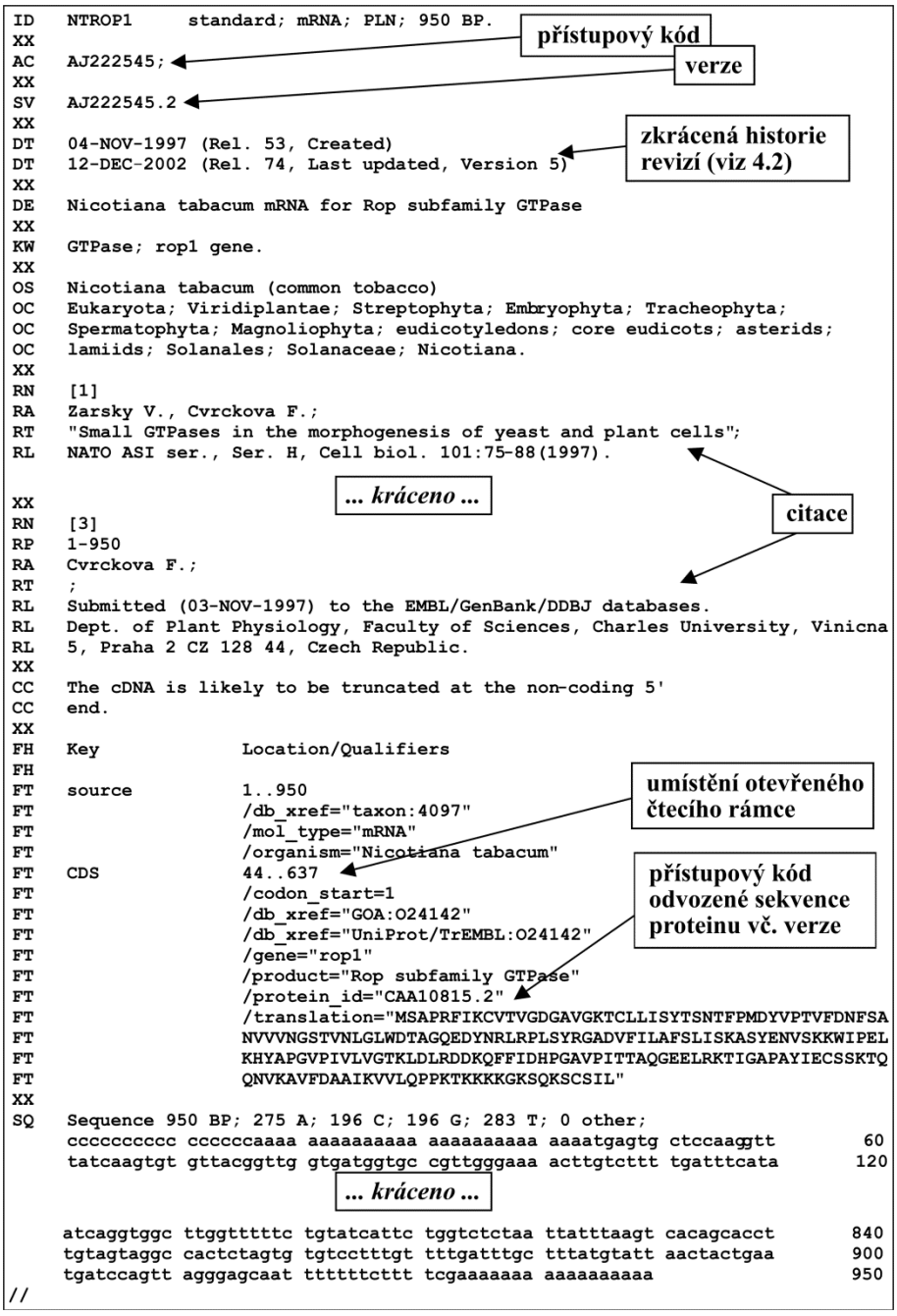
Zvláštní pravidla platí pro typ záznamu označovaný jako **TPA** čili „anotace třetí stranou" (*third party annotation*). Jde o záznamy, které nepřinášejí novou sekvenční informaci, ale upřesňují anotaci sekvencí již existujících, uložených do databází jinými autory – např. umístění promotorů, transkribovaných oblastí či exonů a intronů. Na rozdíl od původních sekvencí nejsou TPA záznamy zveřejňovány automaticky, ale pouze po řádné publikaci práce předkládající řádný důvod pro navrhovanou změnu anotace. Databáze Velké trojky začaly přijímat TPA vstupy v r. 2003, s poměrně volnou definicí toho, co lze přijmout jako publikované podklady – v počátečním období se nepřijímaly pouze výstupy automatických programů pro predikci struktury genů či regulačních míst (Benson et al., 2004; Kulikova et al., 2004), zhruba v polovině roku 2004 byl dohodou vědeckých poradců konsorcia databází (*Scientific Advisory Board*) zaveden striktní požadavek na přímý experimentální důkaz navrhované anotace, a záznamy, které tento požadavek nesplňují, včetně např. anotací odvozených na základě porovnání se známou sekvencí blízce příbuzného druhu, byly přesunuty mezi „mrtvé" databázové položky.

4.3.1. Anatomie databázového záznamu

Strukturu konkrétní databázové položky si ukážeme na dvou příkladech. Prvním je sekvence malé GTPázy NtRop1 z pylu tabáku, která byla klonována při vyhledávání pylově specifických či v pylu exprimovaných malých GTPáz pomocí RT-PCR (Žárský a Cvrčková, 1997), a její sekvence byla posléze upřesněna přesekvenováním sporného úseku před definitivním uložením do databáze, doprovázeným publikací anotace genu (Cvrčková a Žárský, 1999). Oprava sekvence se projevila též v historii revizí databázového záznamu, která již byla uvedena jako příklad v části 4.2. Obr. 4.2. ukazuje komentované plné znění databázového záznamu ve formátu EMBL.

Druhý příklad (Obr.4.3.) ilustruje na příkladu hypotetické sekvence 19 kDa podjednotky aktin-nukleujícího komplexu Arp2/3 *Arabidopsis thaliana* (AtArc19) strukturu záznamu TPA v databázi GenBank. Gen pro AtArc19 byl sice sekvenován spolu s převážnou většinou genomu *Arabidopsis* (The Arabidopsis Genome Initiative, 2000), avšak nebyl v genomové sekvenci náležitě rozpoznán (některé z jeho exonů byly dokonce mylně přičleněny k sousednímu genu). Jeho odhalení umožnilo až porovnání se sekvencemi charakterizovaných ortologních genů rýže a kukuřice (Cvrčková et al., 2004a). Vzhledem k výše zmíněné změně kritérií pro zařazování záznamů do databáze TPA však uvedený záznam není zařazen v hlavní prohledávatelné databázi, a je dostupný pouze prostřednictvím přístupového kódu.

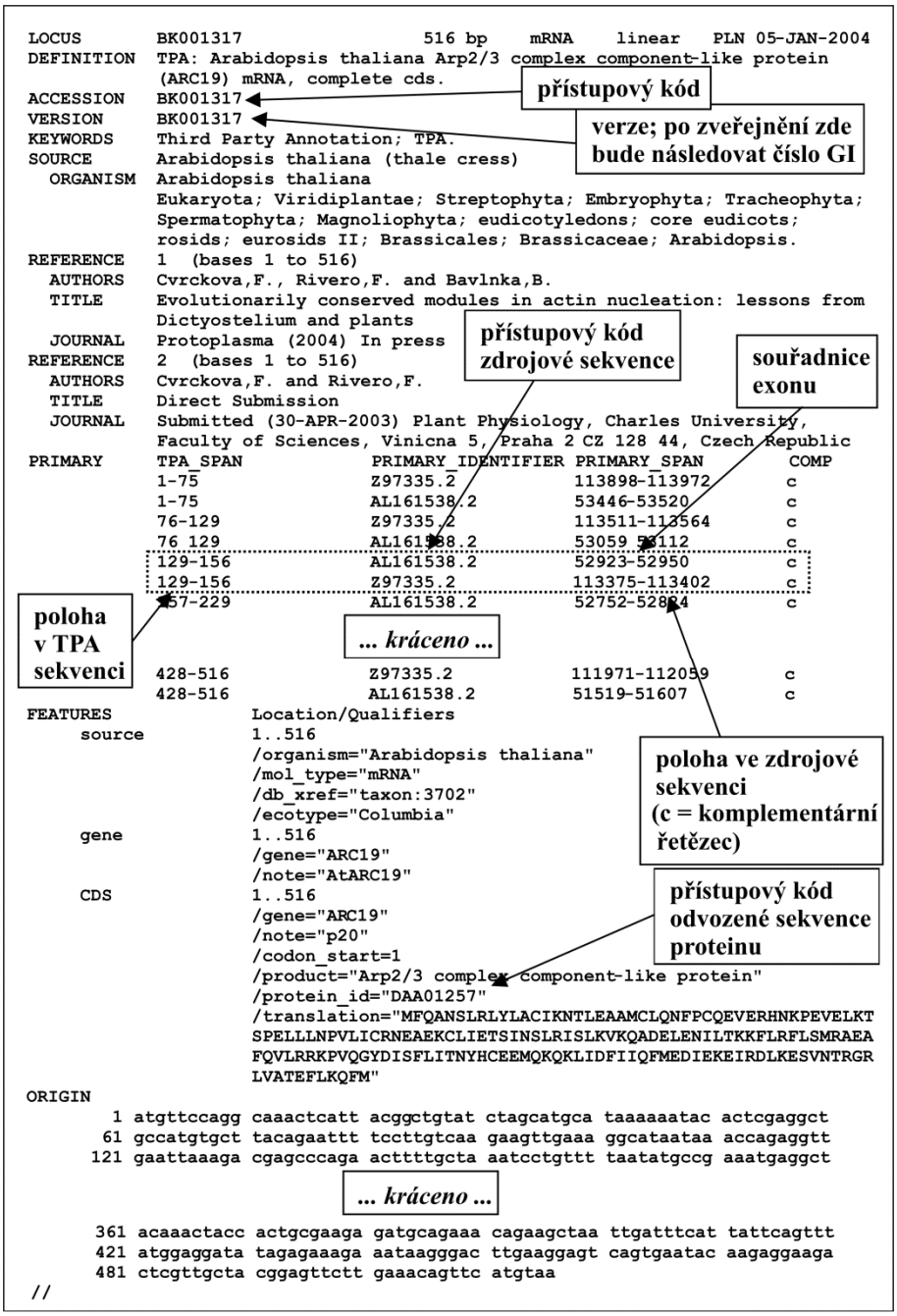
[19] Doba neveřejného uchovávání bývá omezena na nejvýše 1 rok jakožto pojistka proti zapomnětlivosti autorů, autor však může požádat o prodloužení.



Obr.4.2. Výpis záznamu z EMBL databáze.

4.4. Jednoduché vyhledávání v databázích Velké trojky

Ukládání původních dat do databází bylo, je a vždy bude záležitostí poměrně úzké části aktivní výzkumné komunity. Naproti tomu získávat data z databází a dále je analyzovat může kdokoli, pokud je vybaven počítačem napojeným na internet. Libovolný uživatel si v extrémním případě může opatřit třeba i kompletní vydání celé databáze, i když toto počínání má smysl jen v nemnoha specifických situacích (viz 4.1). Praktický uživatel, který není profesionálním bioinformatikem, si zpravidla potřebuje v databázi najít (a pak popřípadě okopírovat) pouze určité záznamy podle předem zvolených kritérií.



Obr.4.3. Výpis záznamu Third Party Annotation (TPA) z GenBank databáze v podobě, v jaké jej GenBank zasílá ke korektuře (formát GenBank flatfile).

K vyhledávání a stahování záznamů z databází slouží webové **uživatelské rozhraní**. Na rozdíl od obsahu, který je v rámci Velké trojky pokud možno zcela sjednocen, uživatelské rozhraní je pro každou databázi specifické. V našich poměrech typický uživatel (neznalý japonštiny) přistupuje k databázím Velké trojky buď prostřednictvím rozhraní **SRS** (*Sequence Retrieval System*) databáze EMBL na Evropském ústavu pro bioinformatiku (*European Bioinformatics Institute*; `http://www.ebi.ac.uk`) nebo prostřednictvím rozhraní **Entrez** databáze GenBank v americkém Národním centru pro biotechnologické informace (*National Center for Biotechnology Information*; `http://www.ncbi.nlm.nih.gov/Entrez/`). S troškou nadsázky by se dalo říci, že uživatelské rozhraní dává databázi osobitou tvář. Obsahově ovšem velké rozdíly nejsou: obě rozhraní dovolují vyhledávat záznamy na základě

přístupových kódů, klíčových slov v názvech a anotacích, jmen autorů a literárních citací. Umožňují též putovat mezi záznamy, případně mezi databázemi, podle křížových odkazů (např. mezi sekvencí DNA a odvozeného proteinu či mezi jednotlivými sekvencemi publikovanými v téže práci). Poskytují samozřejmě také přístup k pokročilejším nástrojům, např. prohledávání na základě podobnosti sekvencí samotných (kap. 6).

Konkrétní provedení se mezi SRS a Entrez výrazně liší. Rozhraní SRS skýtá výrazně více nástrojů než Entrez (včetně možnosti zavést si účet na serveru a *de facto* si postavit vlastní dílčí databázi), není však – právě pro obsáhlou nabídku sofistikovaných nástrojů – příliš přehledné. I když v poslední době doznalo rozhraní v tomto směru výraznou změnu k lepšímu, často používané nástroje se někdy obtížně hledají, mimo jiné právě proto, že stránky SRS procházejí neustálým vývojem. Rozhraní Entrez poskytuje sice méně funkcí, avšak je již po mnoho let stabilně uspořádané, jednodušší a vůči začátečníkům přívětivější – prakticky všechny běžně používané nástroje jsou rychle a snadno dostupné z jednoho místa – z „rolety“, na níž lze navolit databázi a hledané parametry. Pro nezkušeného uživatele je lze jen doporučit.[20]

Rozhraní NCBI Entrez umožňuje nejen přístup k databázi GenBank a přidruženým sekvenčním databázím (např. odvozená databáze sekvencí proteinů GenPept či samostatné proteinové databáze konsorcia Uniprot – viz 4.5), ale i prohledávání dalších zdrojů. Většina z nich je zaměřena spíše na medicínu a lidskou genetiku, což je zcela pochopitelné vzhledem k institucionálnímu zakotvení v Národní lékařské knihovně Spojených Států (*National Library of Medicine*). Jsou však mezi nimi i databáze s informacemi zásadní důležitosti i pro nelékařské biologické obory – např. databáze map některých sekvenovaných prokaryotních i eukaryotních genomů (včetně *Arabidopsis*) s odkazy na referenční sekvence v databázích.

Výsadní postavení mezi nesekvenčními databázemi dostupnými přes Entrez má **databáze literatury PubMed**, která je sice postavena na základě původně lékařské databáze Medline, avšak pokrývá široké pole experimentálních biologických oborů (nechybí zde ani časopisy vyhraněně rostlinného zaměření). Prostřednictvím PubMed lze získat plné citace prací od 60. let 20. století, od 80. let většinou včetně abstraktů a často i s odkazem na elektronické verze celých článků, někdy dostupné i zdarma.

Entrez poskytuje rovněž přístup k databázi experimentálně zjištěných **trojrozměrných struktur** proteinů (jádrem je databáze PDB, která disponuje rovněž vlastním webovým rozhraním nezávislým na Entrez na adrese `http://www.rcsb.org/pdb/`). Databáze struktur je prohledávatelná prostřednictvím klíčových slov, případně přístupových kódů, a je propojena s databází konzervovaných proteinových domén.

Zajímavou součástí nástrojů Entrez pro prohledávání sekvenčních databází, databáze PubMed i databáze struktur je možnost zobrazení **„příbuzných objektů“** pro libovolný databázový záznam. Jde-li o záznam sekvenční, lze přes nabídku „Links“ získat odkazy na citace v PubMed i seznam předem vyhledaných „příbuzných sekvencí“.[21] Záznamy z databáze literatury jsou naopak propojeny s odkazy na sekvence publikované v příslušných článcích, a na seznam „příbuzných článků“, vybraných a seřazených na základě kombinace kritérií – např. shodných klíčových slov, jmen autorů či výskytu slov v abstraktech.[22]

[20] Je pozoruhodné, jak celkový dojem z pojetí evropského a amerického poměrně dobře odpovídá vžitým představám o kulturních rozdílech mezi „starým“ a „novým“ kontinentem.
[21] O kritériích vyhledávání více v kap. 5 a 6.
[22] Využití slov z abstraktu vedlo již v časném období existence systému Entrez k několika neplánovaným a překvapivým důsledkům. Pro všechny kromě autorů programu asi spíše úsměvné

4.5. Specializované databáze včetně genomových

Z desítek sekvenčních databází mimo Velkou trojku stojí za zvláštní pozornost moderovaná **databáze proteinových sekvencí Swissprot** (Bairoch et al., 2004), která se vyznačuje důrazem na zvláště podrobnou manuální anotaci začleněných proteinů. Na rozdíl od nemoderovaných databází, jako je Velká trojka, se přitom na anotaci podílejí nejen původní autoři sekvencí, ale kolektiv více než sta expertů, kteří shromažďují připomínky od uživatelů a aktivně vyhledávají relevantní publikace. Po kritickém zhodnocení pak začleňují do jednotlivých databázových vstupů dobře doložené údaje týkající se např. posttranslačních modifikací či vazebných míst proteinů.

Vztah Swissprotu k Velké trojce je v mnoha ohledech nadstandardní – podílí se například na jednotném systému přístupových kódů a GI čísel. Některá přístupová rozhraní, jako třeba NCBI Entrez, ji zahrnují do základního neredundantního souboru, a v tomto ohledu je Swissprot svého druhu součástí Velké trojky (tři mušketýři ostatně také byli čtyři). Swissprot, trEMBL a americká databáze PIR (*Protein information resource*), fungující podobně jako Swissprot, se v r. 2002 propojily v rámci iniciativy **Uniprot** (Apweiler et al., 2004) a sdílejí nyní data podobným způsobem jako primární databáze Velké trojky.

Pro publikaci genomových sekvencí platí stejné zvyklosti jako pro zveřejňování jiných původních sekvenčních dat, a i genomové sekvence si tedy časem najdou cestu do databází Velké trojky. Stavba těchto databází však umožňuje pouze omezenou anotaci – standardní formát databázového záznamu nedovoluje začlenit třeba obsáhlejší údaje o funkci genových produktů, o fenotypu mutací či o regulaci genové exprese (např. data z mikroarrayů). Specializované **genomové databáze** pro jednotlivé organismy umožňují právě takovou rozšířenou anotaci. Dalším důvodem pro jejich existenci je i to, že sekvenování kompletních genomů je záležitost poměrně zdlouhavá, a laboratoře, které se na sekvenování podílejí, často v zájmu získání pomoci s anotací zpřístupňují vědecké komunitě i neanotované nebo dokonce nesestavené „polotovary“, které v takové nehotové podobě ani nemají v úmyslu zadávat do primárních databází.

Sekvenovaných genomů průběžně přibývá, a nemá cenu pokoušet se zde o vyčerpávající výčet databází. Uvedu jen několik příkladů, které spojuje mj. vztah k rostlinné biologii obecně a biologie *Arabidopsis* zvlášť. Již zmiňované rozhraní NCBI Entrez poskytuje přístup ke **genomové databázi asociované s GenBank**, pokrývající genomy, jejichž referenční sekvence již byla uložena v GenBank. Obsah databáze ovšem jen o málo překračuje standardní anotaci GenBank – asi nejvýznamnějším přínosem jsou přehledné grafické mapy genomů, ve kterých se snadno vyhledává. Jako příklad databáze, jejíž obsah překračuje pouhý uspořádaný výčet sekvencí, můžeme uvést databázovou sekci webových stránek Venterova Ústavu pro výzkum genomů **TIGR** (*The Institute for Genomic Research*, `http://www.tigr.org`), s podsekcemi věnovanými především genomům prokaryotních mikroorganismů, ale i genomovým či EST projektům některých eukaryot včetně nálevníka *Tetrahymena*, huseníčku (*Arabidopsis*), rýže, kukuřice, borovice, tolice (*Medicago truncatula*) a bramboru. Databázi genomu *Arabidopsis* lze prohledávat např. podle konvenčních názvů genů, AGI kódů chromozómových lokusů, či podle klonů sekvenovaných

---

bylo zjištění, že pokud výchozímu článku chyběl abstrakt, jevily se jako „příbuzné“ všechny články bez abstraktu – program hlásil 100 % shodu „abstraktu“, který zněl *„No abstract available“*. Nejméně jeden poměrně významný mezinárodní skandál však vypukl poté, co jistý autor nalezl přes „příbuzné články“ plagiát vlastního textu, také na základě 100 % shody abstraktu, tentokrát přítomného.

mezinárodním konsorciem; pro každý lokus pak je možné zobrazit např. synonyma, klonované cDNA, predikci struktury genu (sestřihu) různými metodami, i příbuzné sekvence.

Ještě více údajů může být uloženo v databázi zaměřené výhradně na jediný organismus. Pro *Arabidopsis* jsou takovýmto ústředním zdrojem stránky **TAIR** (*The Arabidopsis Information Resource,* `http://arabidopsis.org`), které kromě shora uvedených dat shromažďují i údaje o genové expresi (včetně přístupu k datům z mikroarrayových experimentů), fenotypy mutantů, relevantní literaturu a adresář vědců, kteří se zabývají výzkumem *Arabidopsis*. Od konce r. 2004 tato databáze také přebírá péči o průběžnou aktualizaci genomových dat, protože databáze TIGR již dále není aktualizována.

Na rozhraní mezi informačními zdroji zaměřenými na sekvence biologických makromolekul a literárními databázemi stojí poněkud nešťastně nazvaný, avšak myšlenkou a předpokládanými budoucími výstupy zajímavý projekt „ontologie genů" **GO** (*Gene Ontology*, `http://www.geneontology.org`; The Gene Ontology Consortium, 2001). Jeho cílem je vytvořit definovaný slovník „klíčových slov" (*GO terms*), která by mohla být zařazena do anotací genů v sekvenčních databázích, což by umožnilo snadné (a automatizovatelné) třídění genů podle toho, na jakých buněčných procesech se genový produkt podílí, jaká je jeho biochemická funkce a kde je v buňce lokalizován.[23]

4.6. Databáze propojené se sbírkami biologických materiálů

Zvláštním případem jsou databáze, které jsou nejen zdrojem dat, ale i katalogem biologických materiálů dostupných vědecké komunitě (pro nekomerční účely zpravidla za únosný manipulační poplatek, nebo zcela výjimečně i zdarma). Pro *Arabidopsis* existují dvě **centrální sbírky** (depozitáře) biologických materiálů, především geneticky definovaných semen a DNA klonů – americká ABRC (*The Arabidopsis Biological Resource Center*, přístup přes TAIR) a britská NASC (*Nottingham Arabidopsis Stock Center*). Obsah obou sbírek se do značné míry překrývá a očekává se, že duplikované materiály budou evropští uživatelé objednávat v Nottinghamu, kdežto američtí v ABRC. Přístup do katalogů centrálních sbírek je otevřený komukoli, objednávání pouze registrovaným uživatelům (registrace sama je zdarma). Krom těchto sbírek existují i další, dílčí kolekce materiálů spojených s konkrétními projekty (např. japonská sbírka cDNA RIKEN), často s obdobnými podmínkami přístupu.

Katalogy sbírek však obsahují pouze informace o názvech linií a klonů a o předchozí historii jejich objednávání. Uživatel musí předem vědět, co si chce objednat. Může to zjistit například z publikovaných prací – ovšem za předpokladu, že usiluje o získání klonu, na kterém už někdo pracuje. Obvyklejší však je situace, kdy uživatel hledá materiál dosud necharakterizovaný a uložený do sbírky v rámci nějaké hromadné analýzy - třeba cDNA k určitému lokusu (často i neúplně charakterizovaný klon, z jehož konce byla získána sekvence EST) nebo mutantní linii, která ve zvoleném lokusu nese inzerci T-DNA či transpozónu.[24] Takové materiály, pokud jsou

[23] Na toto úsilí se můžeme – v dobrém i zlém – dívat jako na o generaci mladší a současnému stavu vědění bližší alternativu standardizovaného systematického názvosloví enzymů. Ve své době bylo třídění enzymů podle E.C. (*Enzyme Comission*) čísel nepochybně pokrokem, a ostatně stojí za to stále databázi E.C. čísel udržovat (viz `http://www.expasy.org/enzyme/`). Hranice tohoto přístupu se staly zřejmými až poté, co se ukázalo, jakého bohatství biologických procesů se účastní např. proteinkinázy, souhrnně řazené do třídy E.C. 2.7.1.37.

[24] U semenných rostlin totiž nefunguje homologní rekombinace vnesené DNA do chromozómu, a proto nelze použít běžné postupy reverzní genetiky, jak je známe u savců či kvasinek. Jako náhradní

spojeny se sekvenční informací (EST či celá sekvence u cDNA, sekvence rozhraní vnesené inzerce u mutantních rostlin), lze samozřejmě „zamapovat“ na genom a orientovat se v nich podle umístění na genomové mapě. To umožňuje služba **T-DNA Express** na serveru `http://signal.salk.edu`, která propojuje podrobnou mapu genomu, sekvenční informace o dostupných materiálech a přímé odkazy na katalogy sbírek.

řešení se osvědčila inzerční mutageneze pomocí T-DNA, následovaná sekvenováním hranic inzerce u celých kolekcí rostlin, v nichž pak lze vyhledávat mutanty ve zvoleném genu.

5. Posuzování podobnosti sekvencí

Porovnávání dvou či více sekvencí a zjišťování míry jejich vzájemné podobnosti je centrálním tématem praktické bioinformatiky. Již jsme se s ním setkali v předchozí kapitole – vyvoláme-li z databáze prostřednictvím specializovaného odkazu „příbuzné sekvence“, využíváme již výsledky nějakého vyhledávacího postupu založeného na posuzování podobnosti sekvencí, i když třeba vůbec nevíme, jak se to dělalo.[25] Dříve než se pustíme do praktického prohledávání databází podle kritéria podobnosti s námi vybranou sekvencí, zastavme se u otázky, jak vůbec lze stanovit kvantitativní míru podobnosti dvou sekvencí.

5.1. Modelový postup stanovení podobnosti

Právě vyslovené otázce by možná měla předcházet jiná: za jakých předpokladů vůbec v případě **sekvencí digitálních znaků** můžeme hovořit o podobnosti? Dva řetězce digitálních znaků, například počítačové programy, totiž mohou být toliko totožné nebo nestejné: liší-li se byť i jen v jediném znaku, už jsou to různé programy. Považujeme-li však digitální znaky za symboly, zastupující **tělesné objekty**, například nukleotidy či aminokyseliny, o podobnosti už smysl hovořit má. Můžeme třeba konstatovat, že pokud jde o tvar, velikost a náboj molekuly, isoleucin je podobnější leucinu než histidinu. Tvrzení, že „I je blíž k L než k H“ má smysl právě jen jako zkratka v tomto či podobném kontextu (bližší rozbor viz Cvrčková a Markoš, 2005). Využití digitálních symbolů pak umožňuje podobnost nejen kvalitativně zjišťovat, ale i kvantitativně měřit, přijmeme-li určité předpoklady např. o vzájemných vztazích jednotlivých aminokyselin.

Má-li měření podobnosti mít smysl, musíme být schopni rozlišit, které sekvence jsou si doopravdy podobné. V ideálním případě tedy hledáme postup, který by identickým sekvencím přiřadil podobnost maximální, a dvěma náhodně vybraným náhodným sekvencím podobnost minimální, kterou bychom pro praktické účely mohli považovat za „nepodobnost“. Již ze složitosti této formulace lze tušit, že s „nepodobným“ koncem spektra bude potíž. Pokusme se proto nejprve navrhnout postup, který by nám umožnil změřit, či aspoň porovnat, podobnost sekvencí, které sice nejsou identické, ale jsou si velmi blízké (tj. mají stejnou délku a liší se pouze v nevelké frakci aminokyselin či nukleotidů). To znamená, že se budeme zabývat **homologními** sekvencemi vzniklými poměrně nedávnou evoluční divergencí ze společného předka.[26] Od tohoto postupu pak můžeme dále extrapolovat směrem k sekvencím různé délky či k sekvencím, které se výrazněji liší.

Obecný postup stanovení míry podobnosti dvou sekvencí by mohl vypadat např. takto (Obr.5.1.):

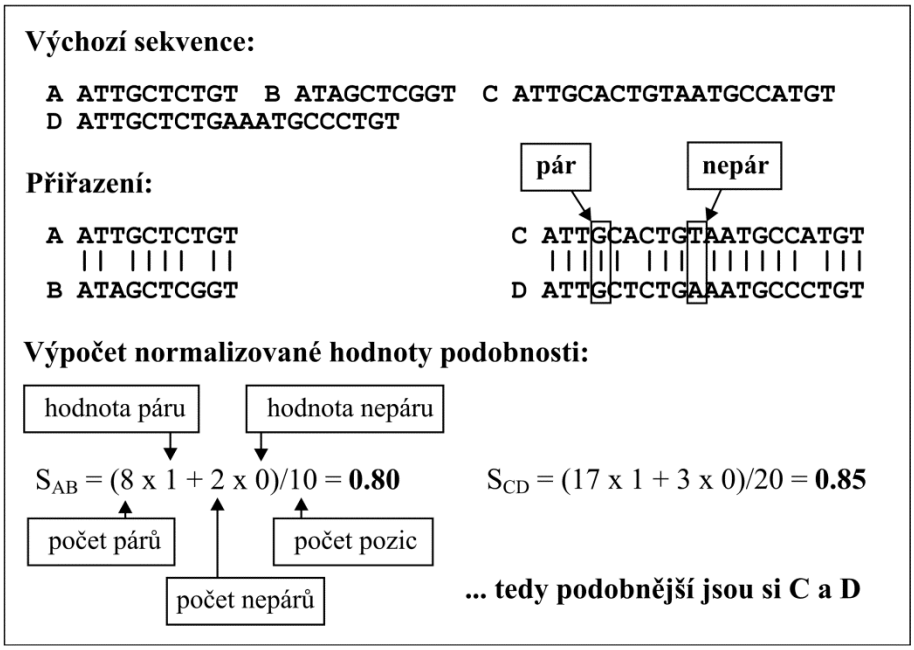
- Sekvence po celé délce přiložíme k sobě, tj. zapíšeme je do dvou pod sebou umístěných řádků tak, aby identické pozice (báze či aminokyseliny) ležely pod sebou.[27] Takovému zápisu se říká **přiřazení (*alignment*)**.

[25] „Podobnost“ a „příbuznost“ budeme prozatím chápat jako synonyma, ale časem se pokusíme oba termíny vymezit pečlivěji (viz kap. 10).

[26] Typickým příkladem jsou třeba sekvence izoforem určitého enzymu z příbuzných organismů – představme si třeba malou podjednotku Rubisco z *Arabidopsis* a z kukuřice. Podrobnější rozbor pojetí homologie a různých jejích variant viz kap. 10.1.

[27] Na počítači musíme používat neproporcionální písmo, t.j. písmo, které má všechny znaky stejně široké (např. `Courier`).

- Vypočteme **celkovou hodnotu (*score*) podobnosti** tak, že sečteme hodnoty podobnosti všech jednotlivých pozic přiřazení. Hodnoty podobnosti stanovíme podle předem zvolených kritérií (viz 5.3). V nejjednodušším případě např. můžeme jakékoli identické dvojici pozic (pár -*match*) přidělit hodnotu 1 a jakékoli neidentické dvojici (nepár - *mismatch*) hodnotu 0.[28]
- Chceme-li porovnávat míru podobnosti různě dlouhých dvojic sekvencí, vydělíme celkovou hodnotu podobnosti délkou přiřazení (**normalizace hodnoty podobnosti**).



Obr.5.1. Modelový postup stanovení odpovědi na otázku, zda jsou si vzájemně podobnější sekvence A a B nebo C a D.

Prakticky používané algoritmy, implementované v programech, které budeme používat, si můžeme představovat jako „černé skříňky“, ve kterých bydlí nějaká varianta právě uvedeného postupu.

5.2. Globální a lokální přiřazení

Chceme-li právě nastíněný postup rozšířit na sekvence odlišné délky, nebo i na sekvence stejné délky, které se významněji liší, sama konstrukce přiřazení se stává netriviální záležitostí. Existuje řada programů, které přiřazení dovedou sestavit (jeden příklad viz níže), a prakticky každá metoda prohledávání databází podle míry podobnosti (kap. 6) nějakým způsobem otázku automatizace přiřazení řeší. Přitom bychom si však měli být vědomi toho, že **konstrukce přiřazení zahrnuje rozhodnutí, jež v principu nelze algoritmizovat**, a tedy ani svěřit výpočetní technice. Podaří-li se nám někdy takovým rozhodnutím zdánlivě vyhnout, je to jen proto, že to za nás udělal autor softwaru, který používáme.

[28] Za krásný neologismus „nepár“ vděčím jistému prodavači, který se mne pokoušel přesvědčit, že při té slevě nevadí, že jedna bota je uvnitř šedá a druhá žlutá.

Jedno z těchto rozhodnutí se týká otázky, jak naložit s oblastmi sekvence, které vzdorují přiřazení – buďto jsou natolik odlišné, že se v nich „nechytíme“, nebo prostě přebývají (delece či inzerce v jedné z přiřazovaných sekvencí). Zde je na nás, zda se pokusíme sekvence přiřadit po celé délce i za cenu vnášení mezer do jedné z nich či do obou (**globální přiřazení**), nebo zda se omezíme na přiřazování jednoznačných úseků a skončíme tam, kde se sekvence rozcházejí nad únosnou míru (**lokální přiřazení**). Nepřiřazené úseky sekvencí můžeme totiž často v následných analýzách ignorovat (viz 10.4.1.). Pokoušet se o globální přiřazení má cenu jen u sekvencí, které jsou si vzájemně blízké – tj. nepodlehly v průběhu evoluce přestavbám a lze je vzájemně přiřadit s vnesením poměrně malého počtu nedlouhých mezer (Obr.5.2); u sekvencí vzdálenějších však produkuje artefakty. Naproti tomu lokální přiřazení je vhodné i pro sekvence složené ze „stěhovavých“ domén (viz Obr.5.4.), a v případě sekvencí blízce příbuzných přitom konverguje k výsledkům velmi podobným přiřazení globálnímu. Zdálo by se tedy, že globální přiřazení snad ani nestojí za to, abychom se jím zabývali, nebýt toho, že je základem některých metod pro konstrukci přiřazení mnohočetných (tj. přiřazení zahrnujících více než dvě sekvence – kap. 8).

Globální přiřazení

```
SLAV----------APATNIK-------PIQNYR-I------AKSETQRYMVIE
SLAVYTYIEFVRANAPATNIKSECVRAAPIQNYRRVEHVRATAKSETQRYMVIE
```

Lokální přiřazení

```
SLAVYTYIEFVRANAPATNIKSECVRAAPIQNYRRVEHVRATAKSETQRYMVIE
-------------NAPATNIKSECVRA-PIQNYRRVEHVRA-------------
```

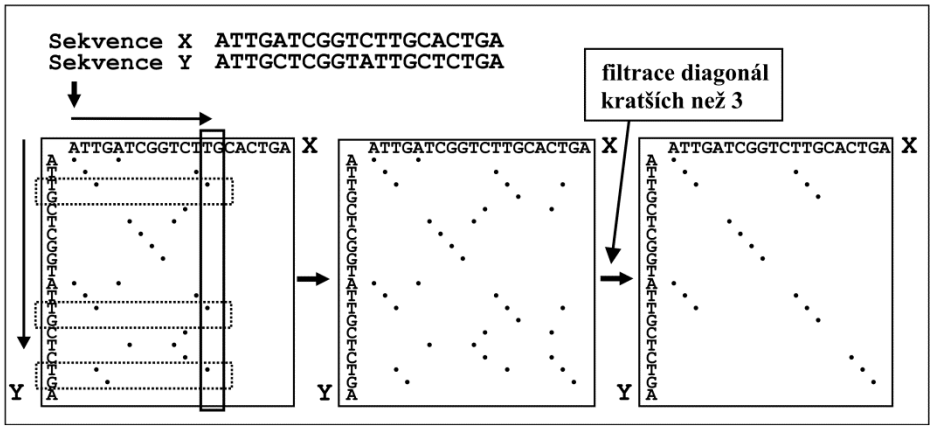
Obr.5.2. Příklad globálního a lokálního přiřazení.

Dobrou pomůckou pro volbu typu přiřazení, a vůbec pro posouzení, zda danou dvojici sekvencí má smysl přiřazovat, je grafická mapa vzájemné podobnosti sekvencí čili **bodový diagram (*dotplot*)**. Bodový diagram vyjadřující podobnost dvou sekvencí *X* a *Y* můžeme (opět v hrubém přiblížení typu „černé skříňky“) získat například následujícím postupem (Obr.5.3.):

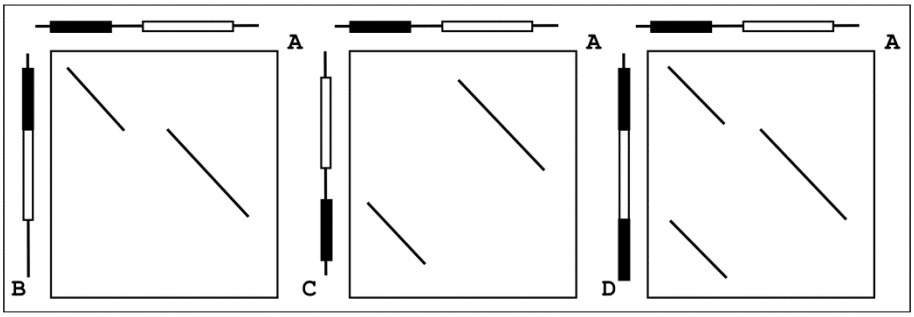
- Souřadnice (pořadová čísla) bází či aminokyselin v sekvencích vyneseme na dvě vzájemně kolmé osy (*X* na osu x, *Y* na osu y) tak, že pozici 1 v sekvenci *X* odpovídá pozice 1 v sekvenci *Y*.
- Vezmeme „okénko“ zvolené délky (např. 2 písmena) a posouváme je po sekvenci *X* od polohy 1 na konec v krocích po 1 písmenu.
- Pro každou pozici okénka uděláme tečku všude, kde okénko „našlo“ stejnou dvojici písmen v sekvenci *Y*.
- V následném kroku můžeme pro přehlednost odfiltrovat (tj. smazat) všechny tečky, které nejsou součástí diagonál delších než nějaká předem stanovená minimální hodnota.

Podobnost sekvencí *X* a *Y* se nám v diagramu projeví jako diagonála (přerušovaná v místě nepárů). Z pohledu na diagram můžeme okamžitě odhalit přítomnost a polohu oblastí vzájemné podobnosti, ale i inzercí, delecí, translokací a opakování v rámci jednotlivých sekvencí – a také posoudit, zda dotyčná dvojice sekvencí je vůbec vhodná pro konstrukci globálního přiřazení (Obr.5.4.). Pokud sekvence prošly přestavbami doménové struktury, musíme se buď spokojit s přiřazením lokálním, nebo vybrat pouze oblast odpovídající jedné celistvé doméně; pro ni pak lze udělat i přiřazení globální.[29]

[29] Smysl to má např. při konstrukci mnohočetných přiřazení pro fylogenetickou analýzu.



Obr.5.3. Konstrukce bodového diagramu.



Obr.5.4. Příklady bodových diagramů pro sekvence, které prošly přestavbami doménové struktury. Jen pro dvojici A a B lze sestrojit smysluplné globální přiřazení.

Algoritmy, které se používají pro praktickou konstrukci přiřazení, dovedou vytvořit buď přiřazení lokální, nebo globální. Rozhodnutí mezi lokálním a globálním přiřazením tedy musíme učinit již na úrovni volby programu; u párového přiřazení (*pairwise alignment, alignment of two sequences*) se prakticky setkáváme pouze s přiřazením lokálním. Strategii nejjednoduššího možného postupu **konstrukce lokálního přiřazení** si můžeme představit zhruba následovně:

- Shora uvedeným postupem sestrojíme bodový diagram.
- Zvolíme diagonálu, která se má stát jádrem přiřazení (pokud nás nevede zvláštní zájem o konkrétní oblast sekvence, o níž už něco víme, volíme diagonálu s nejvyšší *nenormalizovanou* hodnotou podobnosti).
- Tuto diagonálu postupně rozšiřujeme po 1 zbytku, a v každém kroku vypočteme hodnotu podobnosti. Pokračujeme tak dlouho, dokud s přidáváním dalších zbytků hodnota podobnosti buď roste, nebo klesá méně než o předem stanovený mezní rozdíl.

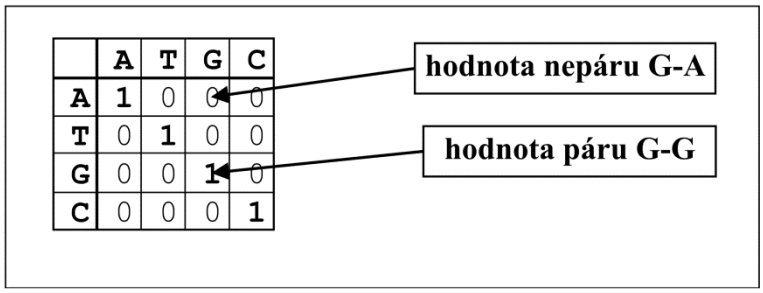
Varianty tohoto postupu mohou zahrnovat omezené zavádění mezer, například při propojování sousedících diagonál, jejichž konce jsou vzdáleny méně než nějaká předem stanovená mezní hodnota.

Praktickým příkladem obecně dostupného programu, který dokáže vytvořit bodový diagram a lokální přiřazení dvou uživatelsky zadaných sekvencí a dovoluje i manipulaci s empirickými parametry (viz níže) je **program BLAST2**, přístupný prostřednictvím stránek NCBI (`http://www.ncbi.nlm.nih.gov/BLAST/`; viz též 6.2). Algoritmus používaný tímto programem zhruba odpovídá právě uvedenému schématu a je variantou části dnes již klasického prohledávacího postupu BLAST (*Basic Local Alignment Search Tool*; Altschul et al., 1990), který bude blíže popsán v následující kapitole (6.2).

5.3. Empirické parametry – substituční matice a cena mezer

Volba druhu přiřazení není jediným krokem, kde do hry vstupují uživatelská rozhodnutí. Dojde na ně i při vlastním **výpočtu hodnoty podobnosti**, kde musíme nejprve zvolit „váhu“ všech možných kombinací (párů i nepárů) aminokyselin, popřípadě nukleotidů. Pokud jsme při konstrukci přiřazení zaváděli mezery, musíme se rovněž rozhodnout, jakou „váhu“ či „cenu“ mezerám přisoudíme. Od toho se pak bude odvíjet další rozhodování, totiž zdali se nám vůbec zavádění mezer vyplatí, nebo zda raději místo zavedení mezery lokální přiřazení ukončíme.

Váhy všech možných párů i nepárů aminokyselin či nukleotidů udává **substituční matice** (*substitution matrix*, *scoring matrix*). Substituční matice je čtvercová tabulka, jejíž řádky i sloupce odpovídají jednotlivým symbolům v sekvenci. Číselná hodnota na průsečíku řádku a sloupce odpovídá příspěvku příslušné kombinace symbolů k celkové hodnotě podobnosti (Obr.5.5. a Tab.5.1.). Hodnoty přisouzené jednotlivým nepárům nezávisí na pořadí symbolů (nepár G-A má stejnou hodnotu jako nepár A-G), a matice je tudíž symetrická podle diagonály.



Obr.5.5. Nejjednodušší substituční matice, použitá pro výpočet podobnosti nukleotidových sekvencí v Obr.5.1.

Konkrétní hodnoty v matici odrážejí pozorování ze skutečných sekvencí. V případě proteinů jsou to jednak pozorované frekvence záměn konkrétní aminokyseliny v rámci známých souborů homologních sekvencí, jednak frekvence výskytu jednotlivých aminokyselin. Spárovaná vzácná aminokyselina, např. cystein nebo tryptofan, má větší váhu než aminokyselina častá, popřípadě aminokyselina kódovaná mnoha kodony, jako třeba serin. Je nutno zdůraznit, že hodnoty, přisouzené jednotlivým kombinacím, nezávisí na sekvenčním kontextu – evoluční konzervace klíčového serinového zbytku v aktivním centru enzymu nepřispívá k celkové hodnotě podobnosti o nic víc než „konzervace“ serinového zbytku ve vysoce variabilní a pohyblivé smyčce sporného funkčního významu. Číselné hodnoty v maticích jsou stanoveny empiricky z „reprezentativních“ či „typických“ souborů sekvencí tak, aby jejich aplikace vedla k výsledkům, které pokud možno dávají smysl. Především pro proteiny **neexistuje „jediná správná“ substituční matice**. Pro sekvence nukleových kyselin se však prakticky používají varianty jediné matice, tzv. **matice IUPAC** čili matice identity (*identity matrix*), která všem párům přiřazuje konstantní kladnou hodnotu, a všem nepárům rovněž konstantu (nulovou či zápornou).

U proteinů je situace složitější, protože tam bylo až dosud sestaveno několik sérií matic. Historicky nejstarší, a dosud poměrně často používané, jsou **matice PAM** (*point accepted mutation*), vytvořené Margaret Dayhoffovou a spolupracovníky již na sklonku 70. let minulého století na základě analýzy globálních přiřazení sady blízce příbuzných sekvencí. Konstrukce matic PAM vychází z empirického stanovení frekvence jednotlivých specifických

záměn. V rámci modelové sady sekvencí byly pro každou aminokyselinu zjištěny frekvence všech možných záměnových mutací v situaci, kdy mutace postihují 1 % zbytků v sekvenci. Na základě takto změřených hodnot specifických mutačních rychlostí byla sestrojena matice PAM1, vynásobením této matice jí samotnou PAM2 atd. až po PAM250, která odpovídá stavu, kdy na stoaminokyselinový úsek připadá již 250 mutací (Hansen, 2001; Pearson, 2001). Vzhledem k tomu, že mutace postihují v ideálním případě náhodně vybrané zbytky, zůstává i v tomto případě zachováno přibližně 20 % zbytků nemutovaných.[30] Novější obdobou série PAM, sestrojenou z většího souboru výchozích dat, jsou **matice Jones-Taylor-Thorntonovy** (JTT) a **Gonnetovy**. Pro číslování Gonnetových matic platí totéž co pro PAM – GONNET250 odpovídá PAM250.

Tab. 5.1. Příklad reálné substituční matice pro proteiny - BLOSUM80. Kódy aminokyselin viz Tab. 3.1.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	5	-2	-2	-2	-1	-1	-1	0	-2	-2	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-1	-1	-6
R	-2	6	-1	-2	-4	1	-1	-3	0	-3	-3	2	-2	-4	-2	-1	-1	-4	-3	-3	-2	0	-1	-6
N	-2	-1	6	1	-3	0	-1	-1	0	-4	-4	0	-3	-4	-3	0	0	-4	-3	-4	4	0	-1	-6
D	-2	-2	1	6	-4	-1	1	-2	-2	-4	-5	-1	-4	-4	-2	-1	-1	-6	-4	-4	4	1	-2	-6
C	-1	-4	-3	-4	9	-4	-5	-4	-4	-2	-2	-4	-2	-3	-4	-2	-1	-3	-3	-1	-4	-4	-3	-6
Q	-1	1	0	-1	-4	6	2	-2	1	-3	-3	1	0	-4	-2	0	-1	-3	-2	-3	0	3	-1	-6
E	-1	-1	-1	1	-5	2	6	-3	0	-4	-4	1	-2	-4	-2	0	-1	-4	-3	-3	1	4	-1	-6
G	0	-3	-1	-2	-4	-2	-3	6	-3	-5	-4	-2	-4	-4	-3	-1	-2	-4	-4	-4	-1	-3	-2	-6
H	-2	0	0	-2	-4	1	0	-3	8	-4	-3	-1	-2	-2	-3	-1	-2	-3	2	-4	-1	0	-2	-6
I	-2	-3	-4	-4	-2	-3	-4	-5	-4	5	1	-3	1	-1	-4	-3	-1	-3	-2	3	-4	-4	-2	-6
L	-2	-3	-4	-5	-2	-3	-4	-4	-3	1	4	-3	2	0	-3	-3	-2	-2	-2	1	-4	-3	-2	-6
K	-1	2	0	-1	-4	1	1	-2	-1	-3	-3	5	-2	-4	-1	-1	-1	-4	-3	-3	-1	1	-1	-6
M	-1	-2	-3	-4	-2	0	-2	-4	-2	1	2	-2	6	0	-3	-2	-1	-2	-2	1	-3	-2	-1	-6
F	-3	-4	-4	-4	-3	-4	-4	-4	-2	-1	0	-4	0	6	-4	-3	-2	0	3	-1	-4	-4	-2	-6
P	-1	-2	-3	-2	-4	-2	-2	-3	-3	-4	-3	-1	-3	-4	8	-1	-2	-5	-4	-3	-2	-2	-2	-6
S	1	-1	0	-1	-2	0	0	-1	-1	-3	-3	-1	-2	-3	-1	5	1	-4	-2	-2	0	0	-1	-6
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-2	-1	-1	-2	-2	1	5	-4	-2	0	-1	-1	-1	-6
W	-3	-4	-4	-6	-3	-3	-4	-4	-3	-3	-2	-4	-2	0	-5	-4	-4	11	2	-3	-5	-4	-3	-6
Y	-2	-3	-3	-4	-3	-2	-3	-4	2	-2	-2	-3	-2	3	-4	-2	-2	2	7	-2	-3	-3	-2	-6
V	0	-3	-4	-4	-1	-3	-3	-4	-4	3	1	-3	1	-1	-3	-2	0	-3	-2	4	-4	-3	-1	-6
B	-2	-2	4	4	-4	0	1	-1	-1	-4	-4	-1	-3	-4	-2	0	-1	-5	-3	-4	4	0	-2	-6
Z	-1	0	0	1	-4	3	4	-3	0	-4	-3	1	-2	-4	-2	0	-1	-4	-3	-3	0	4	-1	-6
X	-1	-1	-1	-2	-3	-1	-1	-2	-2	-2	-2	-1	-1	-2	-2	-1	-1	-3	-2	-1	-2	-1	-1	-6
*****	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	1

PAM matice pro vzdálenější sekvence (tj. s vyššími čísly) byly tedy získány extrapolací. Druhá často používaná série **matic BLOSUM** byla naopak odvozena přímo z empirických dat na základě lokálního přiřazení konzervativních domén méně blízce příbuzných sekvencí. Také matice BLOSUM jsou číslovány – čím vyšší číslo, tím vyšší frakce konzervativních zbytků ve výchozím souboru sekvencí (Hansen 2001). Pro praktickou aplikaci je důležité pamatovat si pravidlo, že **čím podobnější jsou si sekvence proteinů, tím nižší hodnoty PAM, případně vyšší hodnoty BLOSUM** bychom měli volit, a naopak. Matice s nízkým PAM a s vysokým BLOSUM nejsou ale volně zaměnitelné, protože byly odvozeny na základě zcela jiných

[30] Hodnotu 20 % identických zbytků lze rovněž považovat za dolní mez podobnosti, která ještě o něčem vypovídá: s dalšími mutacemi se sekvence stává prakticky náhodnou, avšak její podobnost se sekvencí výchozí klesá už jen velmi pomalu (Chothia a Lesk, 1986; Wood a Pearson, 1999).

empirických dat. Používat bychom je měli pokud možno na data podobná těm původním, proto nízké PAM preferujeme v situacích, kde za podobnost „může“ krátká evoluční vzdálenost (malý počet mutací) oddělující zkoumané sekvence, kdežto vysoké BLOSUM tam, kde předpokládáme významnou podobnost „ostrůvků“ v jinak divergentních sekvencích např. v důsledku selekce či evolučních omezení (*constraints*).

O **ceně mezer** (*gap cost*, *gap penalty*) rozhodují zpravidla dva parametry – cena za otevření mezery (*gap open penalty*) a v absolutní hodnotě obvykle výrazně nižší cena za prodloužení již existující mezery o 1 zbytek (*gap extension penalty*). Rozdíl těchto cen, které výslednou hodnotu podobnosti vždy snižují, odráží skutečnost, že pokud již v sekvenci mezera vznikla (ať již delecí nebo inzercí), její prodloužení už je změnou poměrně nepodstatnou a mnoho na konečné míře podobnosti nezmění.

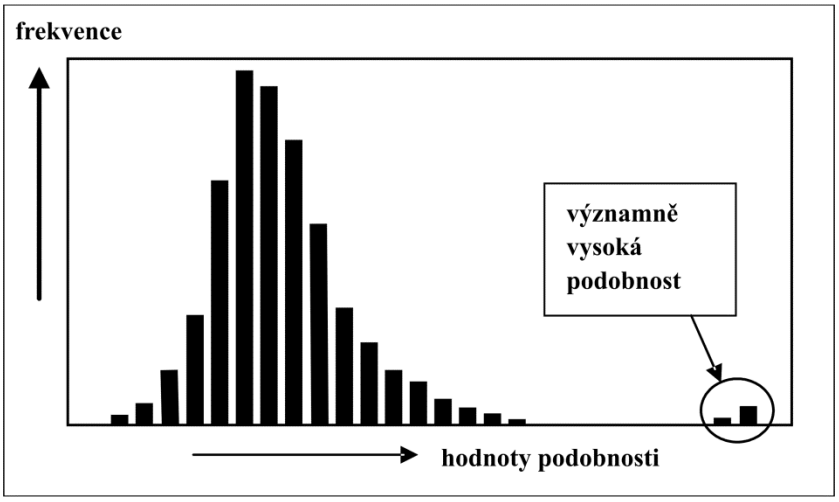
Nejen v případě programů pro konstrukci přiřazení a stanovení míry podobnosti sekvencí bývají **výchozí hodnoty empirických parametrů** (matice, ceny za mezeru) v programech nastaveny tak, aby ve většině případů vedly k „rozumným“ výsledkům (matice zpravidla na kompromisní hodnotu pro sekvence, které nejsou ani příliš blízké, ani příliš vzdálené). Nezkušený uživatel většinou neučiní chybu, ponechá-li nastavení programu ve výchozím stavu. Vždy však stojí za to zvážit, s jakými daty pracujeme: zajímá-li nás např. evoluce malé podjednotky Rubisco u semenných rostlin, můžeme si dovolit změnit výchozí matici na PAM40 či PAM20, kdežto v případě porovnávání globinu archeí a eukaryot volíme raději PAM250. Také vypadají-li výsledky podezřele (přiřazení se skládá z mnoha krátkých úseků oddělených mezerami v jedné i druhé sekvenci, nebo naopak zjevně podobné úseky sekvence nejsou navzájem přiřazeny), vyplatí se s hodnotami parametrů experimentovat.

6. Prohledávání databází podle podobnosti se známou sekvencí

Nyní, když jsme již získali představu o tom, jak můžeme chápat (a stanovovat) podobnost ve světě digitálních sekvencí, můžeme přistoupit k vlastnímu vyhledávání v databázích. Podobně jako v předchozí kapitole začneme tím, že se pokusíme formulovat zcela obecný algoritmus, který povede k cíli – totiž především ke zjištění, které sekvence ve zkoumané databázi jsou nejpodobnější námi zvolené sekvenci, kterou budeme nadále označovat jako **dotaz (*query*)**. Současně by bylo výhodou, kdybychom dokázali posoudit, zda ty nejpodobnější sekvence jsou dotazu vskutku podobné víc než jen náhodně – tedy **statisticky vyhodnotit** výsledky vyhledávání.

Předpokládejme nejprve, že máme k dispozici neomezené výpočetní kapacity a že na výsledek nespěcháme. V tom případě zajisté sekvence příbuzné našemu dotazu nalezneme tak, že:

- Pro každou sekvenci z databáze sestrojíme bodový diagram podobnosti s dotazem a z něj odvodíme co nejdelší lokální přiřazení (viz 5.2).[31]
- Pro každé lokální přiřazení vypočteme celkovou (nenormalizovanou) hodnotu podobnosti s dotazem.
- Všechny sekvence v databázi seřadíme podle hodnot podobnosti s dotazem; tím jsme již zjistili, která sekvence v databázi je dotazu nejbližší.
- Zaznamenáme distribuci zjištěných hodnot podobnosti dotazu s jednotlivými sekvencemi z databáze a zjistíme, jakou křivku rozdělení tyto hodnoty sledují. Hodnoty podobnosti, které leží výrazně mimo křivku odpovídající nahodilé distribuci, odpovídají sekvencím, jejichž míra shody s dotazem je zřejmě větší než nahodilá (Obr.6.1.)



Obr.6.1. Hypotetický příklad distribuce hodnot podobnosti.

Uvedený postup je poněkud naivním a zejména ve statistické části značně zjednodušeným příkladem **přesného** prohledávacího algoritmu, který by databázi prohledal způsobem vskutku vyčerpávajícím. Nevýhodou takového algoritmu je nutnost sestrojovat přiřazení pro všechny sekvence v databázi a z toho plynoucí pomalost. Proto byly přesné algoritmy v praxi

[31] Prohledáváme obvykle databáze, které obsahují i sekvence genomové či sekvence neúplných fragmentů molekul. Nemá tedy smysl zkoumat přiřazení globální.

do značné míry vytlačeny postupy **heuristickými**, které prohledávání výrazně zrychlují, byť i za cenu určité újmy na přesnosti.

I přes hardwarovou náročnost existují prakticky použitelné (byť i pomalé) implementace přesných prohledávacích algoritmů. Příkladem může být program SSEARCH (součást standardního instalačního balíku programu FASTA, o němž bude řeč dále), určený k prohledávání proteinových databází proteinovým dotazem, popřípadě nukleotidových databází dotazem nukleotidovým. Jeho heuristickým protějškem je algoritmus FASTA (Pearson a Lipman, 1988), se kterým jsme se již setkali jakožto s „dárcem" jména jednoho z běžných datových formátů (kap. 3), a na který se nyní zaměříme podrobněji, abychom si jeho obdoby mohli napříště dosazovat za modelový „obsah černé skříňky" zastupující heuristické prohledávací algoritmy.

6.1. FASTA jakožto modelový heuristický algoritmus

Jedním z časově nejnáročnějších kroků výše nastíněného prohledávacího postupu je konstrukce a optimalizace lokálních přiřazení. Přitom pro řadu sekvencí lze tento krok s minimálním rizikem vynechat – stačilo by, kdybychom uměli zhruba odhadnout míru podobnosti s dotazem a vyloučit sekvence, u nichž je krajně nepravděpodobné, že třeba i po důkladném „ladění" přiřazení dosáhnou vysokých hodnot podobnosti. Rovněž přiřazení samo není pro všechny sekvence nutno optimalizovat do posledního detailu (tj. do stavu, kdy žádná změna nezvýší hodnotu podobnosti). Pro vyloučení vzdálenějších sekvencí stačí, když se optimálnímu stavu přiblížíme, a jen do zbývajících „kandidátů" pak investujeme čas nezbytný k opravdu pečlivému vypracování přiřazení.

Uvedené principy lze demonstrovat na příkladu **algoritmu FASTA** (Pearson a Lipman, 1988; Pearson, 2000), který sice v poslední době zůstává poněkud ve stínu výkonnějších metod, ale ve své době přinesl zásadní průlom. Již jeho předchůdce, program FASTP vyvinutý stejnou pracovní skupinou v polovině 80. let minulého století, zrychlil prohledávání databází oproti přesným algoritmům o dva řády. Mladšímu čtenáři je však třeba připomenout, že až publikace programu FASTA přišla v době, kdy využití výpočetní techniky přestalo být záležitostí elitního klubu specialistů a výkonné počítače se staly dostupnými i pro lepší experimentálně biologická pracoviště.[32]

Nechceme-li zacházet do matematické teorie, můžeme říci, že klíčovým krokem ke zrychlení bylo připustit si, že k prvnímu hrubému odhadu míry podobnosti sekvencí v databázi s dotazem vlastně ani není potřeba sestrojovat přiřazení. K první filtraci databáze, tedy k vyloučení sekvencí, kterými nestojí za to se zabývat, může stačit samotný bodový diagram.

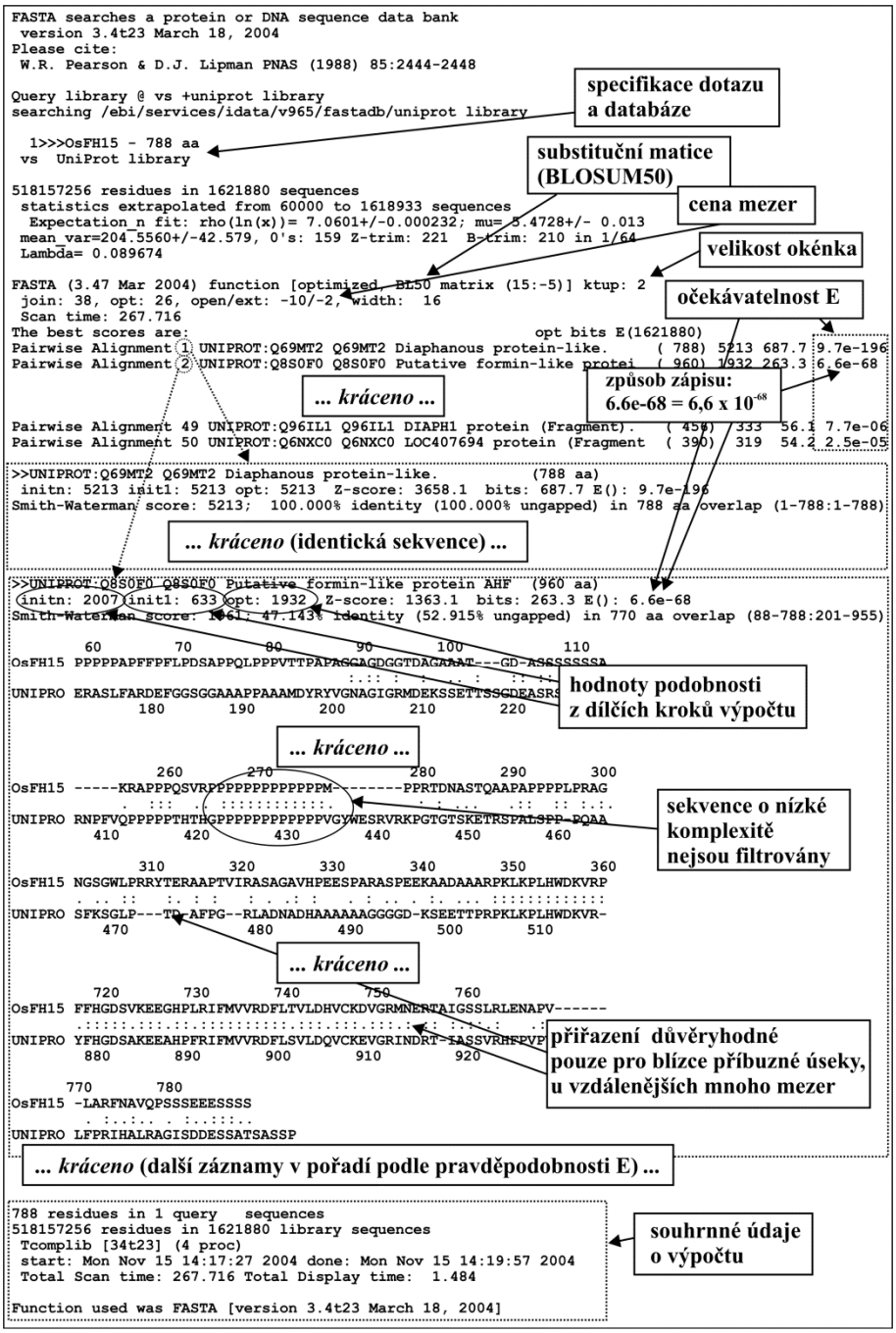
[32] Nesmíme si ovšem představovat dnešní stolní PC či dokonce notebooky. Poměrně bohaté instituce typu Rockefeller University či vídeňského Biozentra na začátku 90. let pro běžné kancelářské práce disponovaly PC s procesorem 286 a operačním systémem DOS či „klasickými" počítači Apple s černobílou obrazovkou. K „vážným" výpočtům, jako třeba prohledávání databází, sloužily sálové počítače operačního systému Unix či dokonce nyní již polozapomenutého VMS se sítí textových (negrafických) terminálů, třeba legendární VT102. Databáze samozřejmě musela být místní, tedy kompletní vydání EMBL či GenBanku, pravidelně updatované na páskách. Internet sice již existoval, ale k datům se přistupovalo pouze pomocí protokolu ftp, který přežívá dodnes, a gopher, nyní na pokraji – ne-li za pokrajem – vyhynutí (viz Cvrčková, 1994). První grafické webové prohlížeče pro protokol http – např. NCSA Mosaic pro Macintosh – do našich krajů dorazily až kolem r. 1994.

Vlastní **postup vyhledávání podobných sekvencí** pomocí algoritmu FASTA si můžeme představit přibližně takto (Pearson a Lipman, 1988):

- Pro každou sekvenci z databáze a dotaz je nejprve sestrojen bodový diagram s velikostí „okénka“ (*ktuple*) o předem zvolené délce (pro sekvenci aminokyselin k = 1 nebo 2, pro nukleotidovou k = 6 – 8). V něm je pak nalezen předem určený počet „nejlepších“ diagonál, tedy diagonál o co největším počtu identit.
- Pro tyto nejdelší diagonály je s využitím předem zvolené substituční matice vypočtena nenormalizovaná hodnota podobnosti (*init1*). Nedosahuje-li hodnota podobnosti nejlepší diagonály alespoň předem stanovené mezní hodnoty (*cutoff*), program se danou sekvencí z databáze přestane zabývat.
- Pokud hodnota podobnosti *init1* pro nejlepší diagonálu překročí mezní hodnotu, program prozkoumá, zda s touto diagonálou nesousedí diagonála jiná (to znamená, zda zkoumanou diagonálu nelze propojit s jinou diagonálou zavedením mezer nepřesahujících předem zvolenou délku). Jestliže ano, program obě diagonály propojí a vypočte novou hodnotu podobnosti *initn* jako součet hodnot *init1* pro výchozí diagonály, od něhož byla za každou mezeru odečtena konstanta.[33] Není-li s čím propojovat, pak *initn* = *init1*.
- Jestliže hodnota podobnosti *initn* přesáhne zvolenou mezní hodnotu (*threshold*), program teprve pro příslušnou diagonálu (či propojené diagonály) poctivě sestrojí co nejdelší lokální přiřazení a znovu vypočte optimalizovanou hodnotu podobnosti *opt* za použití zvolené substituční matice a ceny mezer.
- Hodnoty podobnosti *opt* jsou průběžně zaznamenávány a je stanovena jejich distribuce, přesněji řečeno distribuce z nich odvozených normalizovaných hodnot označovaných jako *Z-score*, případně (poněkud odlišné počítaných) hodnot *bit-score*. U starších verzí programu byl součástí výstupu i velmi názorný histogram hodnot Z-score, nyní se výstup skládá ze seznamu nalezených sekvencí (*hits*), série přiřazení mezi dotazem a nejbližšími nalezenými sekvencemi (zpravidla pro sekvence, jimž odpovídají hodnoty *Z-score* vzdálené od průměru o více než 7 směrodatných odchylek), a údajů o statistickém hodnocení (*bit-score*). Pro nalezené „významné“ sekvence jsou uvedeny též hodnoty *init1*, *initn* a *opt*, a zejména **hodnota „očekávatelnosti“ E** (*expectancy*), která vyjadřuje počet sekvencí vykazujících stejnou podobnost s dotazem, které by měly být nalezeny v databázi stejné velikosti, sestávající z náhodných sekvencí (Obr.6.2.).

Algoritmus FASTA je heuristický, nikoli přesný, protože selektuje sekvence hodné bližšího zkoumání na základě „kvalifikovaného odhadu“ – nezabývá se sekvencemi, u nichž nenašel ani jednu diagonálu s *init1* vyšším nebo rovným mezní hodnotě *cutoff*. Může se tedy dopustit **falešně negativních výsledků** tím, že zmešká sekvenci významně podobnou dotazu, pokud podobnost spočívá v přítomnosti mnoha velmi krátkých úseků identity. Dalším možným zdrojem falešně negativních výsledků jsou sekvence, které sdílejí několik nevalně konzervovaných domén v proměnlivém pořadí, a FASTA je sice možná i najde na základě jedné z domén, ale neshledá podobnost statisticky významnou, protože umí počítat pouze s nejvýše jedním homologním úsekem na každou sekvenci z databáze. Falešně negativní nálezy způsobené těmito příčinami však nepředstavují vážný problém: čím dál do evoluční minulosti sahá příbuznost homologních sekvencí, tím méně si bývají podobné, a musíme si zvyknout na skutečnost, že v jisté vzdálenosti už se nám příbuzenství ztratí za horizontem.

[33] Tento krok odlišuje FASTA od předchozí verze – FASTP.



Obr.6.2. Příklad výstupu programu FASTA.

Horší však je možnost získání **falešně pozitivních výsledků**, tedy nacházení sekvencí, které dotazu příbuzné nejsou. V právě uvedené standardní podobě se v tomto ohledu FASTA potýká s poměrně nepříjemným problémem, pokud sekvence dotazu obsahuje **úseky o nízké komplexitě**, v typickém případě oblasti extrémně bohaté jednou aminokyselinou. Ilustrovat to lze na příkladu kvasinkové proteinkinázy kódované genem *CLA4* (Cvrčková et al., 1995). Tento protein obsahuje N-koncovou doménu extrémně bohatou glutaminem a C-koncovou doménu proteinkinázovou. Gen byl klonován komplementací kvasinkové mutace, a to ještě před publikací kompletní sekvence genomu *Saccharomyces cerevisiae* (Goffeau et al., 1996). V průběhu sekvenování dílčích fragmentů klonovaného genu byla nehotová sekvence průběžně porovnávána s tehdy dostupnými databázemi za pomoci programu FASTA. Poměrně dlouho přitom glutaminem bohatá N-terminální doména „zastiňovala“ doménu

proteinkinázovou, která byla dokonce odhalena vizuální inspekcí dřív, než ji program FASTA uznal za průkaznou![34] Pozdější verze FASTA tomuto problému čelí buď aplikací dodatečných filtrů na výsledky (implementace ve starších verzích softwarového balíku GCG; Womble, 2000), nebo předběžnou filtrací dotazu vedoucí k tomu, že úseky o nízké komplexitě nepřispívají k *init1* (Pearson 2001).

Algoritmus FASTA je sice výrazně rychlejší než přesné prohledávací postupy, avšak stále je poměrně **náročný na výkonnost a čas počítače**. Hodí se proto spíše k provozování na lokálních databázích či v sítích s omezeným počtem registrovaných či platících uživatelů, a poměrně zřídka se s ním setkáváme jako s veřejně dostupnou službou. V takových případech mívá často přístup k omezeným databázím, jako jsou např. databáze genomové (tak třeba na stránkách TAIR – kap. 4.5 – je takto zpřístupněno prohledávání genomu *Arabidopsis*). FASTA s kompletním přístupem k primárním databázím je k dispozici například na serveru Evropského bioinformatického institutu (`http://www.ebi.ac.uk/fasta33/`). Existují **verze programu** určené k prohledávání proteinových databází proteinovým dotazem, nukleotidových databází nukleotidovým dotazem, i verze „přeložené“, kde dotaz a databáze představují sekvence různých typů a FASTA porovnává proteinové sekvence s překlady sekvencí nukleotidových (viz dále – 6.3).

6.2. BLAST

V současnosti zřejmě nejrozšířenějším heuristickým algoritmem pro prohledávání velkých databází je **algoritmus BLAST** (*Basic Local Alignment Search Tool*; Altschul et al., 1990; Gish a States, 1993; Altschul et al., 1997). Metoda BLAST představuje další zásadní zrychlení. Již první verze byla zhruba 10 x rychlejší než FASTP na téže databázi a hardwaru (Altschul et al., 1990), a autor „konkurenčního“ programu FASTA sám přiznává, že BLAST dokáže za jistých okolností zrychlit prohledávání velkých proteinových databází oproti FASTA zhruba 6x (Pearson 2001). Postup prohledávání databáze dnes nejčastěji používaným algoritmem BLAST 2. generace si můžeme představit přibližně takto (Altschul et al., 1990; Altschul et al., 1997; Hansen 2001; Pearson 2001):

- Pro sekvenci dotazu program nejprve vytvoří „slovník“ (*index*) výskytu všech přítomných kombinací symbolů („slov“) o předem zvolené délce (obvykle v rozmezí 2-3 symboly pro aminokyseliny a 10-11 symbolů pro nukleotidy). Pro každé slovo (*word*) ze slovníku a pro všechny jeho deriváty vzniklé záměnou jediného symbolu jsou nejprve zjištěny hodnoty podobnosti (se sebou samým i s deriváty) za použití zvolené substituční matice. V dalším kroku pak jsou ze slovníku vypuštěna všechna slova (ať už původní, nebo odvozená), jejichž výskyt v páru přiřazených sekvencí přispívá k výsledné hodnotě podobnosti hodnotou menší než je jistá předem zvolená mezní hodnota T (*threshold*).
- V následujícím kroku program prohledává všechny sekvence databázi a eviduje pro každou sekvenci výskyt slov ze slovníku. V principu by bylo možné připravit předem jednou provždy „adresář“ výskytu všech možných slov dané délky v databázi a místo databáze samotné prohledávat pouze tento adresář. V praxi se však ukazuje, že bereme-li v úvahu i takové technické aspekty, jako je rychlost přístupu k disku, je při vhodné volbě vlastního prohledávacího algoritmu rychlejší postupně prohledávat celou databázi. Pokud program v některé databázové sekvenci nenajde na téže

[34] Naopak program BLAST, o němž bude řeč dále, s její identifikací neměl nejmenší potíže.

diagonále nejméně dvě slova ze slovníku (*hits*) oddělená vzdáleností menší či rovnou než je předem stanovená „délka okna“ A, sekvenci opustí.

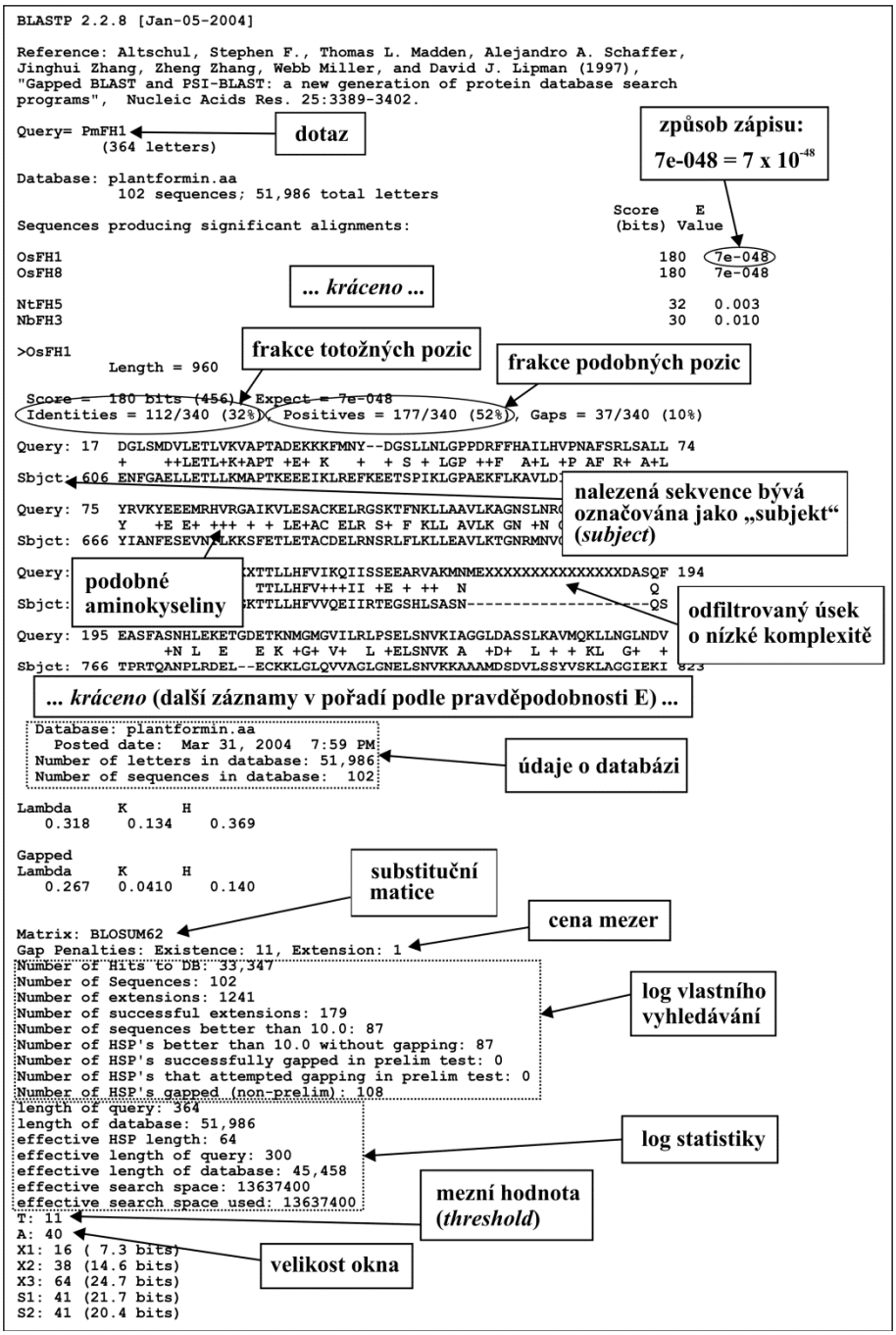
- Pro každou takto nalezenou dvojici „významných“ slov v těsné vzájemné blízkosti program slova propojí a použije jako jádro lokálního přiřazení, které pak oběma směry rozšiřuje (v novějších verzích i s možností zanesení mezer), dokud hodnota podobnosti nezačne klesat. Pokud zjištěná hodnota podobnosti (*score*) dosáhne alespoň předem zvolené mezní hodnoty (*cutoff*), program přiřazení zaznamená jako tzv. HSP (*high scoring pair*), jinak přiřazení opustí. Pro jednu databázovou sekvenci může BLAST najít i více HSP.[35]
- Pro každou databázovou sekvenci obsahující HSP pak program vypočte úhrnnou normalizovanou hodnotu podobnosti ze všech HSP, která je pak použita ve statistickém hodnocení podobně jako u algoritmu FASTA. Výstup programu obsahuje seznam sekvencí s nejlepšími HSP spolu s **hodnotami očekávatelnosti E** (*expectancy*, E-value), které odrážejí předpokládaný počet sekvencí o stejné nebo lepší podobnosti s dotazem ve stejně velké databázi složené z náhodných sekvencí. Za skutečně průkazné lze v případě proteinů zpravidla považovat hodnoty E nižší než 0,001, program však zpravidla uvádí i několik sekvencí s hodnotami vyššími. Následují přiřazení nejlepších HSP a úhrnná statistika úlohy (Obr.6.3.).

Hlavním zdrojem zrychlení (a také dodatečným zdrojem přibližnosti ve srovnání s FASTA) je první krok, který bychom mohli přirovnat k odfiltrování málo informativních slov, jako jsou třeba spojky a ukazovací zájmena, v textových vyhledávačích pro běžné „nevědecké“ použití, například Google. K rychlosti a relativní nenáročnosti programu však přispívá i vlastní algoritmus prohledávání databáze ve 2. kroku. V konečném důsledku BLAST dokáže „růst s databází“ podstatně lépe než FASTA, a je mimořádně vhodný i k provozování jako veřejně dostupná vyhledávací služba. Prakticky všechny veřejně přístupné databáze sekvencí dávají k dispozici server s programem BLAST, dostupný prostřednictvím webového rozhraní. Verze programu BLAST provozovaná a nadále vyvíjená na americkém Národním centru pro biotechnologické informace, tzv. **NCBI BLAST** (McGinnis a Madden, 2004),[36] se stala dnes již standardním webovým nástrojem k prohledávání rozsáhlých databází.

NCBI BLAST (`http://www.ncbi.nlm.nih.gov/BLAST/`) je opatřen přehledným **grafickým rozhraním**, přístupným prostřednictvím webového prohlížeče, a obsáhlou dokumentací. Rozhraní umožňuje celkem jednoduchý a intuitivní výběr varianty programu vhodné pro konkrétní typ dotazu i databáze (viz 6.3) i omezení prohledávané databáze např. na sekvence odvozené z jediného organismu. Nabízí také možnost využití postupu používaného programem BLAST pro konstrukci přiřazení dvou uživatelem vybraných sekvencí. Rovněž výstupy programu jsou obohaceny o grafickou prezentaci ilustrující umístění podobných úseků v rámci nalezené databázové sekvence a dotazu i míru podobnosti (pomocí barevného kódování), a nalezené sekvence lze přímo stáhnout z databáze prostřednictvím webových odkazů. Přednastavené parametry zahrnují i filtraci oblastí o nízké komplexitě, takže se lze snadno vyvarovat tohoto běžného zdroje falešně pozitivních výsledků (filtr však je možné vypnout).

[35] Zde je významný rozdíl oproti algoritmu FASTA, který pro každou databázovou sekvenci eviduje nejvýše jedno přiřazení. BLAST má tedy větší šanci najít sekvence, které prošly přestavbami doménové struktury a liší se pořadím domén (např. v situaci odpovídající prostřední a pravé variantě v obrázku 5.3).

[36] Citovaná práce vyšla ve zvláštním svazku časopisu Nucleic Acids Research, věnovaném veřejně dostupným webovým serverům (*web server issue*). Toto zvláštní číslo je publikováno každoročně, podobně jako svazek věnovaný databázím (viz 4. kapitola).



Obr.6.3. Příklad výstupu programu BLAST – výsledky prohledávání databáze rostlinných forminů sekvencí cDNA z Physcomitrella patens v lokální verzi programu BLAST.

Výsledky automatizovaného prohledávání databáze GenBank algoritmem BLAST jsou rovněž základem výběru „**příbuzných sekvencí**“ v rámci rozhraní Entrez (4.4.). Tyto příbuzné sekvence jsou přístupné přes odkaz Links v seznamu databázových záznamů (odkazy *Related sequences* a *BLink*) a lze je použít pro rychlou orientaci např. v taxonomické distribuci studovaného proteinu.

Ze stránek NCBI je rovněž možno zdarma získat instalační soubory pro **lokální BLAST**, instalovaný na uživatelském počítači. Lokální BLAST vyžaduje lokální databázi, nemá grafické rozhraní a lze jej ovládat pouze zadáváním příkazů z příkazového řádku v okně MS-DOS či jeho emulace. Alespoň na první pohled tedy výhody instalace takového programu

nemusejí být zřejmé. I řadovému uživateli se však lokální BLAST může hodit nejméně ve dvou poměrně běžných situacích. Jednou z nich je prohledávání předběžných či nehotových sekvencí (např. sekvencí nedokončených genomů či rozhraní T-DNA inzertů), které sice dosud nebyly začleněny do primárních databází, avšak autoři je dávají k dispozici ke stažení, obvykle bez jakýchkoli prohledávacích nástrojů.

Druhým možným použitím lokálního BLASTu jsou situace, kdy potřebujeme stanovit míru vzájemné podobnosti uvnitř nějaké sbírky sekvencí (např. u všech známých členů konkrétní genové rodiny), a přitom z těchto sekvencí nemůžeme sestrojit standardní dendrogram (viz 10. kapitola), protože některé z nich jsou neúplné. Jako příklad takového postupu může sloužit třeba „mapování" EST a krátkých fragmentů cDNA kódujících homology forminů z řady nemodelových rostlin na dendrogram sestrojený z kompletních sekvencí z *Arabidopsis* a rýže (Cvrčková et al., 2004b). Neúplné sekvence byly použity jako dotaz k prohledávání malé lokální databáze, sestávající pouze proteinových sekvencí rostlinných forminů, ať již kompletních, nebo neúplných. Z výstupů (viz Obr.6.3.) byla pro každý fragment identifikována nejbližší sekvence. Vzhledem k tomu, že fragmenty byly různé délky a ne všechny sekvence v databázi byly kompletní, nebylo však možno přímo porovnávat míru příbuznosti sekvencí na základě hodnot pravděpodobnosti E – jako měřítko příbuznosti musely posloužit přímo hodnoty udávající frakci identických aminokyselinových pozic.

Právě uvedený příklad již poukazuje na to, že BLAST nemusí sloužit jen k vyhledávání sekvencí. BLAST je totiž také **nástrojem statistickým**. Hodnoty pravděpodobnosti E mohou posloužit jakožto měřítko statistické významnosti pro sekvence nalezené jinou cestou, která statistické hodnocení nedovoluje – třeba pomocí vyhledávání krátkých, uživatelsky definovaných signatur funkčních domén. Příkladem může být ověření nálezu rostlinných homologů dříve charakterizovaných aktin-nukleujících proteinů z rodiny forminů pomocí zpětného vyhledávání BLASTem. Potenciální homology forminů byly identifikovány na základě přítomnosti konzervativního motivu, který je přítomen ve většině živočišných a houbových příslušníků genové rodiny, avšak sestává z pouhých pěti aminokyselin v definovaných rozestupech (Cvrčková, 2000; viz též 8.5.). Tak krátký motiv se v sekvencích může vyskytovat i nahodile, takže samotná jeho přítomnost by ještě nebyla postačujícím důkazem, že nalezené sekvence vskutku patří do rodiny forminů. Prohledávání neredundantní databáze proteinových sekvencí (tj. kombinace překladů Velké trojky a Uniprotu) BLASTem s jednou z nalezených sekvencí jakožto dotazem však identifikovalo známé živočišné forminy s nízkými (tedy vysoce průkaznými) hodnotami pravděpodobnosti E, což stačilo jako důkaz, že nalezené rostlinné sekvence vskutku kódují forminy.

6.2.1. PSI-BLAST a pozičně specifické substituční matice (PSSM)

Zajímavým rozšířením možností algoritmu BLAST, přístupným například prostřednictvím serveru NCBI, je PSI-BLAST – **pozičně specifický iterovaný BLAST** (*position-specific iterated BLAST*, Altschul et al., 1997). PSI-BLAST, dostupný pouze ve verzi pro proteinové sekvence, umožňuje identifikovat vzdálenější homology dotazu a tedy posunout horizont směrem k hlubší evoluční minulosti.

Při práci se standardní verzí BLASTu si někdy můžeme všimnout, že program zřejmě nachází na spodním konci seznamu HSP **„vzdálené příbuzné"** dotazu, ale přisuzuje jim poměrně vysoké a často nesignifikantní hodnoty pravděpodobnosti E. Pohled na výstup

BLASTu však může odhalit, že tyto sekvence ze spodního konce seznamu výsledků sdílejí konzervované sekvenční motivy, které by mohly být dokonce kandidáty na funkční jádro nějaké evolučně staré domény. Pokud standardní BLAST těmto sekvencím přiřadí nesignifikantní hodnoty E, je na místě se obávat, že jde o falešně negativní výsledek. Kdybychom však uměli brát v úvahu nejen hodnoty podobnosti z párového přiřazení mezi sekvencí z databáze a dotazem, ale i existenci vzájemné podobnosti mezi sekvencemi nalezenými v databázi, mohli bychom odhalit i takovéto vzdálené příbuzné – a právě tuto možnost poskytuje PSI-BLAST.

Klíčovým krokem pro vyhledávání vzdálených příbuzných programem PSI-BLAST je konstrukce a následné používání zvláštního typu substituční matice (viz 5.3), totiž **pozičně specifické substituční matice** (PSSM, *position specific substitution matrix*). Na rozdíl od dosud probíraných matic, které byly použitelné pro jakékoli přiřazení a u nichž hodnoty podobnosti přidělované jednotlivým kombinacím aminokyselin nezávisely na konkrétním sekvenčním kontextu, matice PSSM je konstruována vždy znovu pro každé jednotlivé konkrétní přiřazení a tatáž aminokyselina v různých pozicích může mít různou „váhu“. PSSM totiž bere v úvahu empiricky zjištěné frekvence záměn jednotlivých aminokyselin ve specifických, uživatelem vybraných sekvencích v rámci seznamu HSP – sdílí-li konkrétní aminokyselinu v konkrétní pozici několik sekvencí, její váha se zvyšuje.[37]

Postup prohledávání databáze metodou PSI-BLAST lze shrnout následovně:

- Prvním krokem je standardní BLAST (rozhraní NCBI BLAST dokonce umožňuje „přesednout“ ze standardního BLASTu na PSI-BLAST až na úrovni formátování výsledků zaškrtnutím volby „*Format for PSI-BLAST*“). Ve výstupu tohoto kroku, označovaného jako **1. iterace**, pak uživatel zvolí sekvence, z nichž bude v následujícím kroku (iteraci) vypočítána PSSM. Program sám nabízí předvolbu (*default*), v níž jsou vybrány všechny sekvence s hodnotou podobnosti nad jistou hranicí (*threshold*). Vyplatí se však seznam výsledků prohlédnout a vybrat pouze ty sekvence, které odpovídají jedné doméně. V případě podezření na přítomnost divergentní domény obsahující krátké konzervativní motivy naopak stojí za to přibrat i sekvence, které sice 1. iterace označila jako nesignifikantní, avšak vizuální inspekcí v nich lze zmíněné konzervativní motivy najít.
- Po zvolení sekvencí pro výpočet PSSM je spuštěna **druhá iterace**, tj. další kolo prohledávání, v němž program nejprve sestrojí PSSM a pak ji použije k prohledávání databáze. K zobrazení výsledků je nutno použít formátovací dialog, v němž také můžeme zvolit, zda se má zobrazit standardní výstup nebo PSSM pro další použití (v textovém formátu, takže ji lze i uložit). Zvolíme-li zobrazení PSSM, můžeme kdykoli navázat další iterací tak, že na výchozí stránce PSI-BLASTu zadáme jakožto dotaz místo sekvence samu PSSM. Ve výsledcích již je při výpočtu hodnoty E použita PSSM; graficky jsou také odlišeny nové sekvence od sekvencí nalezených v předcházejícím kole. Uživatel opět může navolit výběr sekvencí pro konstrukci PSSM a spustit následující iteraci.
- Chceme-li databázi prohledat skutečně vyčerpávajícím způsobem, celý **postup opakujeme** tak dlouho, dokud nepřestanou přibývat nové položky signifikantních výsledků.

[37] V tomto ohledu je PSSM obdobou *profilů*, o kterých bude řeč v kapitole 7.1.

6.3. Volba metody a parametrů pro konkrétní situace

Vnímavý čtenář si už mohl povšimnout, že ve většině dosud probíraných situací byly konkrétní případy zaměřeny na porovnávání sekvencí aminokyselin a nikoli nukleotidů. Prohledávací programy, jako je BLAST nebo FASTA, ovšem dovedou pracovat s oběma typy dat, a dokonce i s jejich kombinacemi; je však nutno zvolit správnou **variantu příslušného prohledávacího algoritmu** (Tab.6.1.).

Tab. 6.1. Přehled programů pro vyhledávání homologů v sekvenčních databázích. Doporučené programy jsou označeny tučně.

<table>
<tr><td rowspan="2">Dotaz
(query)</td><td colspan="2">Databáze</td></tr>
<tr><td>DNA</td><td>Protein</td></tr>
<tr><td rowspan="2">DNA</td><td>FASTA
BLASTN
SSEARCH</td><td rowspan="2">BLASTX
FASTX
FASTY</td></tr>
<tr><td>TBLASTX</td></tr>
<tr><td>Protein</td><td>TBLASTN
TFASTX</td><td>BLASTP
PSI-BLAST
FASTA
SSEARCH</td></tr>
</table>

Na uživateli navíc je, aby zvážil, zda v konkrétním případě použít metodu FASTA, BLAST (popřípadě PSI-BLAST), či dokonce nějakou metodu přesnou, a zda zůstat při nastavení nabízeném autory programu (*default*) nebo volit nastavení vlastní. Na rozdíl od předchozího kroku však zde nejde o striktní volbu jediného přípustného nastavení: nevhodná kombinace dotazu a databáze se vám odmění chybovým hlášením, avšak mnohdy se nelze jednoduše rozhodnout, zda je pro konkrétní úkol vhodnější program z rodiny BLAST nebo FASTA. V bezproblémových případech ostatně oba postupy dospějí k principiálně stejným a rovnocenným výsledkům (a naopak: **shoda výsledků různých metod** je nejlepším dokladem toho, že výsledky jsou vskutku jednoznačné).

Důvodem našeho dosavadního soustředění na proteinové sekvence je skutečnost, že **analýza sekvencí DNA** je úkolem poměrně řídkým. Zabýváme-li se evoluční historií genů kódujících proteiny, popřípadě sekvencemi proteinů samotných, dáváme v převážné většině případů přednost vyhledávání na základě proteinových sekvencí. Sekvence DNA, i kódující, totiž podléhají rychlé evoluční změně – v důsledku existence synonymních kodonů je teoreticky možné, aby se dvě sekvence DNA lišily téměř o třetinu a přitom kódovaly neodlišitelný protein. Vzhledem k tomu, že se kvalitní sekvence DNA bez variabilních pozic skládá z pouhých čtyř druhů symbolů, mohou současně zcela nepříbuzné (náhodné) sekvence sdílet až 25 % pozic. Horizont, za nímž se všechny stopy příbuznosti sekvencí odvozených ze společného předka ztrácejí v nerozlišitelnosti, leží tedy poměrně blízko.

Situací, kdy nezbývá než vyhledávat **za použití sekvence DNA jako dotazu**, není mnoho. S vyhledáváním **v databázích sekvencí DNA** se setkáme především v následujících případech:

- Hledání homologů oblastí genomu nekódujících proteiny (např. oblastí kódujících nepřekládanou RNA či vůbec nepřepisovaných). Jsme ovšem odkázáni pouze na sekvence, které jsou si blízce příbuzné.
- Vyhledávání sekvencí DNA identických či téměř identických dotazu. Může to být třeba proto, že chceme zjistit, zda už náš fragment někdo nesekvenoval, případně zda

se ke zkoumané cDNA v databázi najde chromozomální sekvence. Sem spadá též rekonstrukce evoluce sekvencí velmi blízkých (například v rámci vnitropopulační dynamiky), kde na úrovni proteinu nenacházíme prakticky žádné rozdíly.

V právě uvedených případech se jako standardní nástroje používají příslušné varianty algoritmu BLAST nebo FASTA (**BLASTN**, **FASTA** s volbou nukleotidového dotazu a databází). Podle W. Pearsona, jednoho z autorů algoritmu FASTA, je porovnávání nukleotidových sekvencí jednou z nemnoha situací, kdy FASTA svou citlivostí, i když nikoli rychlostí, předčí BLAST (Pearson, 2004). U neproblematických sekvencí však jsou rozdíly prakticky zanedbatelné, a s BLASTem se dá dobře vystačit. Zcela přesný, nikoli heuristický, algoritmus SSEARCH, který lze v principu také použít, je sice výrazně citlivější, ale nejméně 10 x pomalejší , a praktický uživatel se bez něj zpravidla dobře obejde. SSEARCH je součástí standardní distribuce FASTA a některých komerčních programových balíků, např. GCG (Womble, 2000).

Tam, kde potřebujeme prohledávat databáze pomocí **proteinového dotazu**, neučiníme nikdy chybu, pokud použijeme algoritmus BLAST (v případě **proteinových databází** v provedení **BLASTP**, případně **PSI-BLAST** pro vzdálenější sekvence). V principu sice lze použít i program FASTA, je však pomalejší a u vzdálenějších sekvencí má sklon vytvářet mírně podezřelá přiřazení s řadou mezer (viz Obr.6.2.), což vede ke zvýšenému riziku falešně pozitivních nálezů.

Chceme-li si však být jisti, že nám neunikly žádné dostupné sekvence homologní dotazu, nemůžeme spoléhat pouze na existující proteinové databáze, nejen primární (Uniprot), ale ani přeložené (GenPept, trEMBL). V přeložených databázích jsou totiž uvedeny pouze překlady nukleotidových sekvencí dodané autory původních sekvenčních či TPA záznamů, a u neanotovaných (či špatně anotovaných) úseků DNA nejsou proteinové překlady uvedeny vůbec, nebo jsou nekompletní. Tzv. „přeložené“ varianty BLASTu či FASTA (*translated searches*)[38] – TBLASTN a TFASTX – dovedou prohledávat **proteinovým dotazem nukleotidovou databázi**, přesněji řečeno databázi, jejímž obsahem jsou automaticky vytvořené překlady všech sekvencí nukleotidové databáze ve všech šesti čtecích rámcích. Preferovaným nástrojem bývá s ohledem na právě zmíněné výhody BLASTu na proteinových datech program **TBLASTN**. Vzhledem ke schopnosti algoritmu BLAST detekovat i několik homologiích úseků v rámci jedné sekvence TBLASTN snadno nachází např. exony kódujícího genu roztroušené po více čtecích rámcích.

Představit si situaci, kdy bychom potřebovali prohledávat opačnou kombinaci dat – tedy **proteinovou databázi nukleotidovým dotazem** – nemusí být úplně snadné. Stačí si však uvědomit, že sekvence DNA z prvního čtení (jako například mnohé EST) bývají nevalné kvality a mohou obsahovat posuny čtecího rámce. Jak BLAST, tak FASTA má varianty pro tento typ vyhledávání, a v různých konkrétních situacích lze najít důvody pro preferenci jednoho či druhého postupu. Varianty algoritmu FASTA – **FASTX** a pomalejší, avšak citlivější **FASTY**, určený pro zvláště špatné sekvence – sice vyžadují znalost orientace dotazu, ale zohledňují posuny čtecího rámce již v průběhu hledání a jsou údajně citlivější než **BLASTX**, který dotaz nejprve přeloží ve všech šesti čtecích rámcích a pak teprve vzniklou sekvencí prohledává databázi (Pearson 2004). Neznáme-li orientaci sekvenovaného fragmentu, je BLASTX nejvhodnějším programem.

[38] Přesnější by sice bylo označovat tyto varianty jako „překládající“, držím se však termínu používaného v anglické literatuře (*translated*, nikoli *translating*)

Rodina programů BLAST obsahuje i verzi **TBLASTX**, která je určena k prohledávání všech možných překladů nukleotidové databáze všemi možnými překlady nukleotidového dotazu. Formálně tedy vlastně jde o „prohledávání nukleotidové databáze nukleotidovým dotazem“, ale fakticky lze tento program chápat spíše jako rozšíření BLASTX o přístup k překladům neanotovaných nukleotidových sekvencí.

Na volbu vhodného programu pak navazuje **výběr konkrétních hodnot parametrů programu**, jako je substituční matice či ceny mezer. Obecně lze říci, že pro první pokus je nejlépe spolehnout na autorské přednastavení (*default*), a že velmi často s přednastavením i vůbec vystačíme. Pro vyhledávání vzdálenějších příbuzných dotazu má však někdy cenu použít matici s nižší hodnotou BLOSUM nebo vyšší hodnotou PAM, změny cen za zavedení či prodloužení mezery nemívají u proteinů významnější dopad. Pokud BLAST nic nenalezne, jednou z příčin může být i filtr sekvencí o nízké komplexitě, který je v takovém případě radno vypnout.

V případě NCBI BLASTu se vyplatí věnovat pozornost **výběru prohledávané databáze** – např. hledáme-li rostlinné homology nějaké sekvence, stojí za to omezit databázi podle organismu na „*Viridiplantae*“. Přínos je hned dvojí – jednak se zmenšením databáze hledání výrazně zrychlí, jednak se výstup stane přehlednějším. Program ovšem umožňuje i dodatečné omezení výstupu na vybranou skupinu organismů v rámci možností formátování.

6.4. Prohledávání strukturních databází sekvenčním dotazem

Speciální případ prohledávání databází představuje hledání odpovědi na otázku, zda je vybraná proteinová sekvence (dotaz) příbuzná nějakému proteinu o známé **trojrozměrné strukuře**. Na rozdíl od dosud popisovaného vyhledávání na základě „pouhé“ podobnosti sekvencí, které bralo v úvahu jednotlivé aminokyseliny buď zcela mimo kontext (v případě použití předem stanovených substitučních matic) nebo v kontextu lineární sekvence (pozičně specifická substituční matice – PSSM), bereme zde v úvahu i pravděpodobnost, s jakou řetězec aminokyselin odpovídající sekvenci dotazu může zaujmout sekundární, popřípadě terciární strukturu podobnou řetězci z databáze. Výsledky takového prohledávání pak mohou sloužit jako základ pro modelování prostorové struktury proteinů (viz kap. 8.3.).

Nejjednodušší způsob, jak získat informaci o příbuzenství mezi naší sekvencí (dotazem) a nějakou charakterizovanou strukturou v databázi, je využít dostupných strukturních dat o nejbližších příbuzných nalezených např. metodou BLAST. Vzhledem k tomu, že dosud všechny známé proteiny zřejmě lze rozčlenit do pouhých několika tisíc strukturních tříd (*folds*), a většina struktur spadá do několika set hojněji zastoupených tříd (Chothia, 1992; Liu et al., 2004), je dost pravděpodobné, že se dotaz i na základě pouhého porovnávání sekvencí podaří aspoň zhruba zařadit do nějaké strukturní třídy. Rozhraní Entrez nabízí u mnoha sekvenčních záznamů výsledky předem provedeného vyhledávání homologních sekvencí pomocí varianty BLASTu (odkaz *BLink*), jejichž součástí je i odkaz na dostupné trojrozměrné struktury (*3D Structures*). Podobně jako v případě předem vyhledaných „příbuzných sekvencí“ však tyto „prefabrikované výsledky“ mohou sloužit pouze k první orientaci, a zdaleka je nelze pokládat za vyčerpávající.

Chceme-li si příbuzné struktury vyhledat sami, využíváme zpravidla metod založených na tzv. „navlékacích“ (*threading*) algoritmech. První krok prohledávání se odehrává na úrovni sekvencí a předpovězených sekundárních struktur (viz též 7.5.). V případě zjištění přijatelné

míry sekvenční příbuznosti pak navazuje pokus o „navlečení“ dotazu na empiricky známou strukturu a vyhodnocení fyzikálně chemické přijatelnosti a pravděpodobnosti takové struktury. I když prohledávání strukturních databází je záležitost náročná, existuje několik serverů dostupných i neplatící akademické veřejnosti a využívajících různé algoritmy; často však tyto servery fungují dávkovým (tj. nikoli interaktivním) způsobem a dodávají výsledky prostřednictvím e-mailu, třeba i několik hodin po zadání.

Tak program **3D-PSSM** (Kelley et al., 2000) na serveru Imperial College v Londýně (`http://www.sbg.bio.ic.ac.uk/~3dpssm/`) umožňuje prohledávat databázi SCOP (Andreeva et al., 2004) – tedy sbírku struktur „vyššího řádu“ (*folds*), odvozenou z primární databáze PDB – postupem, jehož první krok je založen na algoritmu PSI-BLAST. Citlivost prohledávání se zvýší, pokud programu jakožto dotaz nabídneme nikoli jednu sekvenci, ale několik sekvencí vzájemně příbuzných ve formě mnohočetného přiřazení (viz 7.1. a 8. kapitola), což usnadní odvození „trojrozměrné pozičně specifické substituční matice“, v níž jedním z parametrů je i predikovaná sekundární struktura. Jako příklad užití tohoto programu nám může posloužit např. identifikace výchozího modelu pro predikci trojrozměrné struktury nově identifikované domény příbuzné antionkogenu Pten v jedné ze skupin rostlinných forminů (Cvrčková et al., 2004b); viz též 7.5. a 8.3.). Modernější variantou je program PHYRE, zatím ve stadiu testování (`http://www.sbg.bio.ic.ac.uk/~phyre/`). Iterativní postup vyhledávání, příbuzný PSI-BLASTu, používá k vyhledávání možných blízkých struktur, které pak mohou posloužit jako templáty pro „navlékání“, také modelovací program **CPHmodels** (Lund et al., 2002) dostupný na serveru Dánské technické univerzity (`http://www.cbs.dtu.dk/services/CPHmodels/`). Podrobněji bude diskutován v části 8.3.

Další možnosti identifikace příbuzných struktur poskytuje např. program **FUGUE** (Shi et al., 2001), který rovněž pracuje s nějakou obdobou pozičně specifických substitučních matic a bere v úvahu i pozičně specifickou váhu mezer. Na serveru univerzity v Cambridgi (`http://www-cryst.bioc.cam.ac.uk/~fugue/`) umožňuje prohledávat databázi strukturních tříd HOMSTRAD (*Homologous Structure Alignment Database*, Stebbings a Mizuguchi, 2004). Podle našich zkušeností se výsledky 3D-PSSM a FUGUE poměrně dobře shodují (viz Cvrčková et al., 2004b).

Metoda **SAM-T99** (Karplus a Hu, 2001) je založena na odlišném algoritmu a má někdy sklon produkovat falešně negativní výsledky. Sami autoři v manuálu na stránkách serveru SAM-T99 (`http://www.cse.ucsc.edu/research/compbio/HMM-apps/`) připouštějí, že jejich cílem bylo především vyvarovat se výsledků falešně pozitivních, a že při volbě parametrů usilovali o maximální selektivitu i za cenu snížené citlivosti. Novější variantou je algoritmus SAM-T02, dostupný na stejném serveru.

Vzhledem k tomu, že různé programy využívají různé algoritmy, není nijak překvapující, že se jejich výsledky liší. I zde, podobně jako při prostém prohledávání sekvenčních databází, platí že **dva je víc než jeden** – tedy že hodnotu výsledků výrazně zvyšuje shoda několika programů, především pokud jsou založeny na odlišném principu.

7. Hledání známých motivů v proteinových sekvencích

Až dosud jsme se na dostupné údaje o sekvencích proteinů dívali jako na jediný obrovský „balík“ dat, v němž sice můžeme podle libosti vyhledávat, avšak nemáme k dispozici nic než sekvence samotné a jejich autorské anotace. Dosavadní analýzy sekvenčních dat, na kterých se podílela celá vědecká komunita, však vyprodukovaly obrovský **tezaurus znalostí**, z něhož můžeme čerpat podobně jako z dat shromážděných v sekvenčních databázích.

Dobrým vstupem do dostupných údajů o proteinových sekvencích a jejich strukturní a funkční interpretaci je server **ExPASy** (*Expert Protein Analysis System*, `http://www.expasy.org`), který je historicky spjat se Švýcarským ústavem pro bioinformatiku a s databází Swissprot (viz 4.5.). V sekci nástrojů pro proteomiku poskytuje vědecké veřejnosti soubor desítek místních i vzdálených programů pro nejrůznější analýzy proteinových sekvencí. Možnosti sahají od prostého stanovení aminokyselinového složení a izoelektrického bodu (pI) přes predikci molekulové hmotnosti proteolyticky získaných fragmentů a identifikaci proteinů pomocí hmotnostní spektroskopie až po predikci buněčné lokalizace, posttranslačních modifikací a sekundární i terciární struktury.

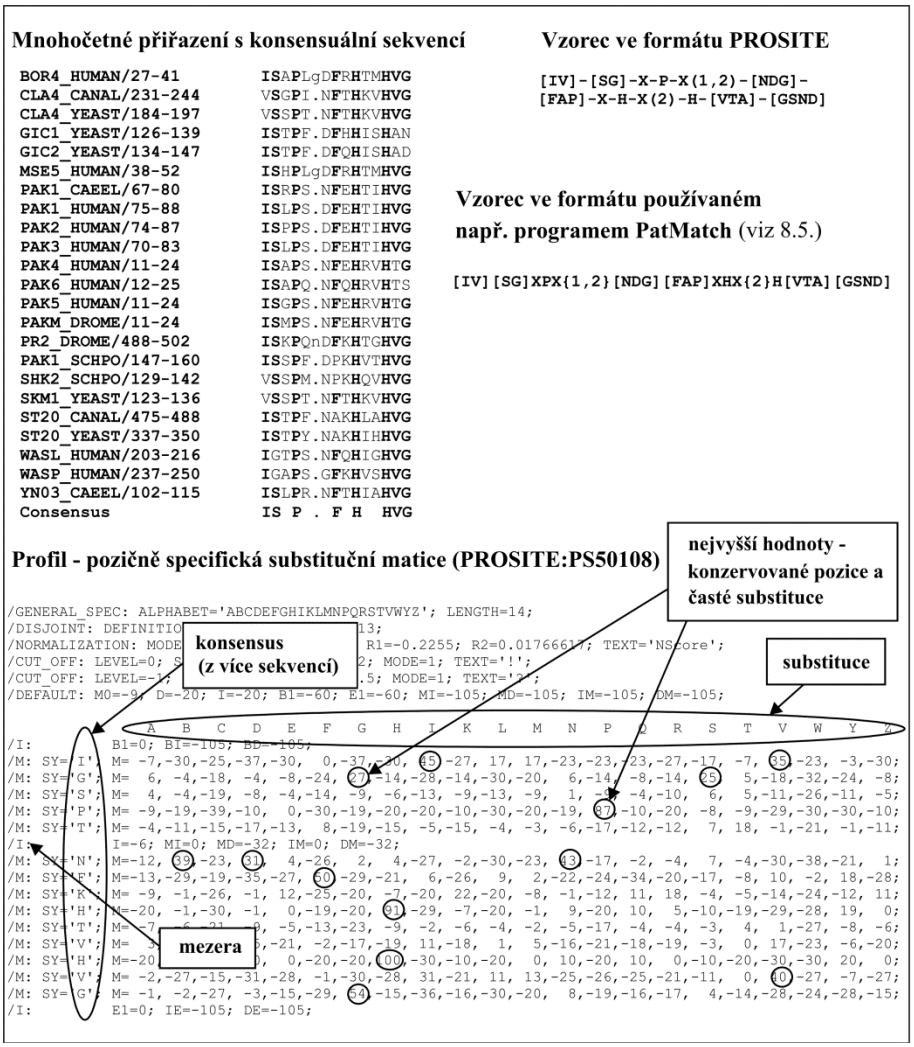
Na serveru ExPASy sídlí také jedna z nejstarších veřejně dostupných databází sekvenčních motivů – **PROSITE** – která nám nyní může posloužit jako dobrý zdroj příkladů. Nejprve se však budeme věnovat motivům z obecnějšího hlediska.

7.1. Mnohočetné přiřazení a způsoby zápisu sekvenčních motivů

Jako **sekvenční motivy** budeme nadále označovat oblasti aminokyselinových sekvencí sdílené nějakou skupinou proteinů. Ve většině prakticky významných případů je „skupinou“ míněna evolučně spřízněná rodina homologních proteinů spjatých společným původem, a jde tedy o motivy evolučně konzervované. Můžeme se však setkat i s jednoduchými motivy vzniklými často konvergencí (např. fosforylační místa či jiná místa posttranslačních modifikací, signální peptidy a jiné motivy určující buněčnou lokalizaci proteinů).

Motiv lze definovat a zapsat různými způsoby, podobně jako samotnou sekvenci (viz 3.1.). Ve všech případech však vycházíme z **mnohočetného přiřazení** (*multiple alignment*) souboru sekvencí, které motiv sdílejí. V mnohočetném přiřazení, podobně jako u přiřazení jednoduchého (viz 5.1.), jsou sekvence zapsány nad sebou tak, aby se na sobě odpovídajících pozicích vyskytovaly pokud možno stejné nebo podobné aminokyseliny (Obr.7.1.). Konstrukce mnohočetného přiřazení sama o sobě je netriviální záležitost; bude jí věnována následující kapitola.

Mnohočetné přiřazení samo sice nese veškerou informaci použitou k vymezení motivu, k praktickému používání však se tento formát příliš nehodí. Snáze se pracuje se zápisy, které v sobě již nesou nějaké předběžné hodnocení – na kterých pozicích záleží víc a na kterých méně, které jsou invariantní a které proměnlivé. Jednou z nejjednodušších možností vyjádření těchto faktorů je **konsensuální sekvence** (*consensus sequence*) – tedy sekvence, která udává invariantní či jednoznačně většinové pozice a jejich rozestupy (Obr. 7.1.). Stanovuje se vždy pro předem určenou míru shody – např. 60 % – i když ne vždy se tato míra v publikacích uvádí. Konsensuální sekvence obsahuje pouze údaje o většinových aminokyselinách na konzervativních pozicích, ne však o jejich frekvenci či o nejčastějších alternativách; problematické je rovněž zacházení s úseky proměnlivé délky.



Obr.7.1. Různé zápisy konzervativního sekvenčního motivu. Uvedený příklad odpovídá části souboru sekvencí definujících tzv. CRIB doménu vážící malé GTPázy třídy Rho, PROSITE: PS50108.

Část těchto nedostatků obchází zápis pomocí **vzorců** (*patterns*). Vzorce umožňují pro variabilní pozice uvádět více alternativ (neobsahují však informaci o frekvenci jednotlivých aminokyselin na proměnlivých pozicích). Dovolují také zápis úseků proměnlivé délky. Vzorce lze psát různými způsoby, podobně jako sekvence samotné. Výsadní postavení, srovnatelné s postavením formátu FASTA pro jednoduché sekvence, zaujímá formát PROSITE, nazvaný podle stejnojmenné databáze (Tab.7.1. a Obr.7.1.). Existují však i jiné možnosti, a zejména programy pro prohledávání databází uživatelsky navrženými vzorci (viz 8.5.) jich hojně využívají (jeden příklad je uveden na Obr.7.1.).

Tab.7.1. Konvence zápisu sekvenčních motivů pomocí vzorců ve formátu PROSITE. Aminokyseliny označujeme standardními IUPAC kódy (viz Tab. 1.1.), jednotlivé pozice jsou odděleny pomlčkami.

Příklad vzorce		<M-[ILV]-{KR}-S(2)-Y-X(2,4)-G-R-X-W
Symbol	**Příklad**	**Význam**
[ ]	[ILV]	nebo (isoleucin, leucin nebo valin)
{ }	{KR}	nikoli (cokoli kromě lysinu nebo argininu)
(n)	S(2)	počet (dva seriny za sebou)
(m,n)	X(2,4)	proměnlivý počet (dvě až čtyři libovolné aminokyseliny)
<	<M	začátek (methionin na N-konci)
>	C>	konec (cystein na C-konci)

Složitějším, avšak informačně bohatším způsobem zápisu jsou **profily** (*profiles*), na které můžeme pohlížet jako na obdobu pozičně specifických substitučních matic (PSSM), sestavených zvlášť pro daný motiv. Profily nesou informaci o pravděpodobnosti různých záměn kterékoli konkrétní aminokyseliny v rámci motivu v podobě „váhy" přiřazované jednotlivým symbolům v příslušných pozicích: čím stabilnější pozice, tím vyšší příspěvek konzervativní aminokyseliny k úhrnné hodnotě podobnosti zkoumané sekvence s profilem (viz Obr.7.1.).

7.2. Databáze známých motivů

Na rozdíl od primárních sekvencí, které jsou soustředěny do několika málo vzájemně dobře propojených databází (viz 4. kapitola), databáze známých sekvenčních motivů představují značně roztříštěný soubor zdrojů, které se zčásti překrývají, avšak nejsou, až na poměrně nové výjimky, konzistentním způsobem provázány.[39] Aktuální přehled stavu databází je každoročně zveřejňován v databázovém čísle časopisu Nucleic Acids Research – v posledním vydání se uvádí téměř 30 databází proteinových motivů či konzervativních domén (Galperin, 2004). Neexistuje také univerzálně přijímaný požadavek na ukládání objevených motivů do veřejně dostupných databází; prakticky všechny **databáze motivů jsou moderované**. Zodpovědnost za naplnění databáze záznamy leží tedy na skupině, která databázi spravuje, a která se zpravidla aktivně angažuje ve výzkumu evoluce proteinových sekvencí či ve vývoji prohledávacích algoritmů.

Zde se soustředíme na několik reprezentativních příkladů – zejména na databáze velké, často navštěvované, opatřené zajímavým vyhledávacím rozhraním či jinak pozoruhodné. Historicky výsadní postavení (přinejmenším v Evropě) má již zmiňovaná **databáze PROSITE** (Falquet et al., 2002; Hulo et al., 2004). PROSITE obsahuje v zásadě dva typy záznamů – vlastní zápisy motivů ve formě vzorců nebo profilů s minimální anotací (v souborech označených přístupovým kódem ve formě PSxxxxx, kde xxxxx je pětimístné číslo) a soubory rozšířené anotace (v souborech PDOCyyyyy, kde yyyyy je pětimístné číslo bez přímého vztahu k číslu odpovídajících vzorců či profilů). Soubory anotace jsou často společné pro několik vzorců či profilů - například katalytická doména proteinkinázy je definována celkem 3 vzorci (PS00107, PS00108 a PS00109), jedním profilem (PS50011) a jednou společnou anotací PDOC00100. Anotace obsahuje mimo jiné informace o **selektivitě a specifitě** jednotlivých vzorců či profilů – tedy jednak o tom, jakou část popisované genové rodiny vzorec zachytí a zda případně některé známé členy rodiny mine (falešně negativní výsledky), či zda a jak často nachází sekvence nepříbuzné (falešně pozitivní).

Databáze PROSITE je dostupná prostřednictvím webového rozhraní na `http://www.expasy.org/prosite/` a může být samozřejmě prohledávána jak pomocí klíčových slov, tak sekvenčním dotazem (pomocí programu ScanPROSITE).

V odlišných formátech, ale na podobném principu fungují další databáze, jako je sbírka „otisků prstů" proteinových domén **PRINTS** (Attwood et al., 2002), kolekce konzervativních domén **ProDom** (Corpet et al., 2000), databáze „proteinových rodin" **Pfam** (*protein families*; Bateman et al., 2004), a další. Zvláštností databáze Pfam a s ní spojených vyhledávacích nástrojů je důsledné využití poměrně komplikovaného, avšak v práci se

[39] Tak v případě databáze PROSITE zprostředkují spojení s dalšími databázemi pouze křížové odkazy ze vstupní stránky.

sekvencemi dnes dosti často používaného matematického postupu založeného na tzv. skrytých Markovovských modelech (*hidden Markov models*, HMM).[40]

Právě uvedené tři databáze se v roce 1999 staly základem „střechového" konsorcia **InterPro**, které se posléze rozšířilo o další zdroje (Mulder et al., 2003) včetně databáze SMART, která bude podrobněji diskutována níže (7.3.). Databáze konsorcia InterPro mohou být prohledávány také pomocí standardizovaných termínů ze „slovníku" GO (viz 4.5.). V poslední době bylo propojení mezi GO termíny a databází PROSITE využito např. k vývoji metod pro **automatizovanou predikci** možných funkčních souvislostí nově objevených sekvenčních motivů (Lu et al., 2004). Do zdrojů InterPro lze vstoupit např. přes domovskou stránku na Evropském bioinformatickém institutu (`http://www.ebi.ac.uk/interpro/`) nebo prostřednictvím dílčích databází konsorcia (včetně PROSITE).

Z databází, které v současnosti nejsou součástí InterPro, stojí za zmínku především **BLOCKS** (`http://blocks.fhcrc.org`, Henikoff et al., 2000). Tato databáze se sice zdá být za zenitem svého rozkvětu koncem 90. let,[41] avšak zasluhuje pozornost jak pro poměrně neobvyklý formát dat, tak i pro bohatou nabídku uživatelských nástrojů dostupných z její domovské stránky. Motivy jsou zde uloženy ve formě „bloků", tedy lokálních mnohočetných přiřazení prostých mezer, a autoři dávají k dispozici i program BLOCKMAKER k tvorbě vlastních bloků z uživatelem dodaných sekvencí bez potřeby předchozí konstrukce přiřazení (viz též 8.2.).

Vlastní databázi konzervativních sekvenčních motivů, až dosud mimo konsorcium InterPro, vyvíjí a udržuje také pracovní skupina spojená s informačními zdroji NCBI (viz 4.4.). Součástí nabídky dostupné prostřednictvím rozhraní Entrez je databáze konzervativních domén **CDD** (*Conserved Domains Database*; Marchler-Bauer et al., 2003), kterou lze prohledávat buď přímo pomocí klíčových slov, nebo sekvenčním dotazem za použití programu BLAST. Prohledávání databáze CDD je dokonce standardně spuštěno jako součást jakéhokoli hledání programem BLASTP na NCBI (viz 6.3.), uživatel ovšem tuto funkci může vypnout.

---

[40] Podrobnější diskuse tohoto algoritmu, který vychází z analýzy pravděpodobnosti přechodu mezi hypotetickými stavy studovaného systému v rámci lineární posloupnosti (např. pár-nepár-delece-inzerce), by přesahovala zaměření této práce. Právě vyslovený výrok lze chápat i tak, že autorka sama nemá tak docela jasno; připouštím, že tomu tak je, a útěchou mi může být snad jen to, že v tom nejsem sama (viz též Hansen, 2001). Dokumentace dostupná ze serveru SMART i z jiných serverů využívajících nástroje založené na HMM má buď povahu odkazu na technická matematická pojednání, nebo prostě konstatuje, že HMM dělá něco jako PSI-BLAST, ale lépe (viz např. dokumentace databáze TIGRFam na `http://www.tigr.org`); naopak se ovšem dá také říci, že PSI-BLAST a jiné na profilech založené prohledávací postupy jsou vlastně zjednodušené, heuristické varianty HMM. Nechci-li se dopustit plagiátu, odkazuji raději čtenáře, kteří se dosud nedali odradit, na instruktivní, i když dosti obsáhlý, výklad na webových stránkách prof. R.D. Boyla z univerzity v Leedsu (`http://www.comp.leeds.ac.uk/roger/`), jejichž autorovi se poměrně dobře podařilo vyložit princip HMM na modelovém případu poustevníka předpovídajícího počasí na základě stavu kousku mořské řasy.

[41] Skupina, která ji vyvinula, nyní soustřeďuje své úsilí na rozvoj výše zmíněné databáze ProDom v rámci InterPro.

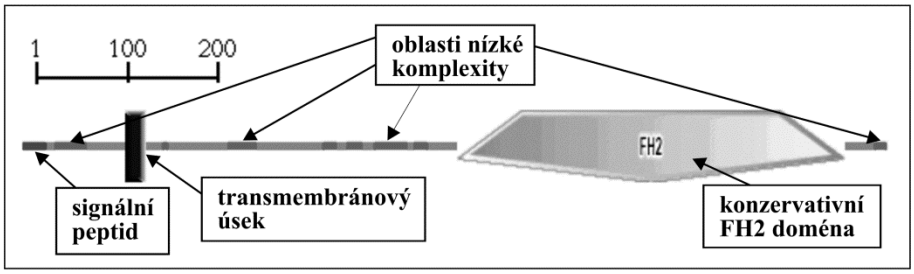
7.3. SMART

Soubor nástrojů pro analýzu struktury proteinů **SMART** (*Simple Modular Architecture Research Tool*; Schultz et al., 1998; Letunic et al., 2002), dostupný na `http://smart.embl-heidelberg.de`, sice obsahuje vlastní databázi konzervativních motivů (nyní začleněnou do konsorcia InterPro), ale poskytuje i další možnosti, zdaleka přesahující pouhé prohledávání databáze sekvenčních motivů.

Jak již naznačuje název, SMART se snaží poskytnout uživatelsky jednoduché rozhraní pro přístup k několika nástrojům poskytujícím informace o různých aspektech doménové či „modulární" struktury proteinů. Aplikací nástrojů z kolekce SMART na uživatelskou sekvenci (dotaz) lze v jediném kroku získat odpovědi na následující otázky:

- Obsahuje dotaz již **známé konzervativní sekvenční motivy či domény**? Uživatel přitom může zvolit, zda prohledávat pouze databázi SMART, nebo i Pfam a databázi „vzdálených homologů" (*outlier homologues*). Prohledávací program HMMER je založen na principu HMM.
- Jsou součástí dotazu oblasti **o nízké sekvenční komplexitě**, popřípadě úseky náchylné k vytváření struktur typu *coiled coil*?
- Obsahuje sekvence dotazu **opakované (duplikované) úseky**?
- Je protein kódovaný dotazem lokalizován do membrány nebo do necytoplazmatického buněčného kompartmentu – tj. jsou v něm přítomny **signální sekvence** pro sekreci či vložení do membrány, popřípadě **transmembránové úseky**?
- Existují v databázích i jiné **sekvence se stejným uspořádáním domén**?

Odpovědi na právě uvedené otázky přitom SMART podává poměrně rychle a ve velmi přehledné grafické podobě (Obr.7.2.) spolu se statistickými údaji (hodnotami pravděpodobnosti falešně positivního výsledku E – viz 6.2.) a přímými odkazy na dokumentaci nalezených konzervativních domén.



Obr.7.2. Příklad grafického výstupu programu SMART pro rostlinný formin třídy I (AtFH6). Skutečný výstup je „klikací" a jednotlivé graficky odlišené úseky slouží jako odkazy na bližší informace o příslušném sekvenčním či strukturním elementu.

Spojení takto různorodých nástrojů způsobem **„vše v jednom"** s jednoduchým uživatelským rozhraním ovšem má i své nevýhody – především omezenou či přímo nulovou možnost zásahu do přednastavení jednotlivých hledacích programu, a u některých metod (např. pro hledání signálních peptidů a transmembránových segmentů) také velmi zestručněný výstup. SMART je ovšem vynikajícím nástrojem pro rychlou orientaci ve zkoumané proteinové sekvenci, a zejména v případě detekce domén jsou jeho výsledky i dobře statisticky podložené a tudíž publikovatelné. Nelze však předpokládat, že by do budoucna zcela vytlačil nástroje specializovanější. U některých z nich se nyní zastavíme.

7.4. Predikce buněčné lokalizace proteinů

Hlubší pohled na metody, které SMART sice používá, ale nedovolí jim nahlédnout pod pokličku, může začít u predikce **směrování (adresování) proteinů** do jednotlivých buněčných kompartmentů. Zaměřme se nejprve na dva programy sídlící na serveru Dánské technické univerzity (`http://www.cbs.dtu.dk/services/`), které jsou do značné míry komplementární.[42]

První z nich, program **SignalP** (Nielsen a Krogh, 1998), zjišťuje, zda N-konec proteinu obsahuje signální sekvenci pro protažení membránou endoplazmatického retikula (tj. **sekreční signál**). Kombinuje přitom metodu HMM a virtuální neuronovou síť;[43] výsledky obou přístupů předkládá v grafické podobě odděleně, spolu s hodnotícím shrnutím. Součástí výstupu je i odhad pravděpodobnosti, s jakou N-koncový úsek proteinu bude odštěpen (tj. poslouží jako řádný signální peptid), a s jakou zůstane spojen se zbytkem proteinu a bude tedy fungovat jako „kotva" integrálního membránového proteinu.

Program **TMHMM** (Krogh et al., 2001) detekuje **transmembránové úseky** s využitím metody založené na HMM. Součástí grafického výstupu je i předpověď topologického uspořádání transmembránového proteinu – tedy odhad, které části proteinu budou v cytoplasmě a které venku. Spolehlivost této predikce je však sporná v případě proteinů s N-terminálním signálním peptidem či membránovou kotvou, pro které někdy vychází obrácená lokalizace. Tak například pro některé rostlinné forminů třídy I (Cvrčková, 2000) program předpovídá krajně nepravděpodobnou extracytoplazmatickou lokalizaci konzervativní FH2 domény nukleující aktin.

Poněkud odlišný algoritmus ke stejnému účelu – predikci transmembránových úseků – využívá program **DAS** (`http://www.sbc.su.se/~miklos/DAS/`, novější verze na `http://mendel.imp.univie.ac.at/sat/DAS/DAS.html`), jehož využití však omezuje skutečnost, že je optimalizován pouze pro prokaryotické sekvence.

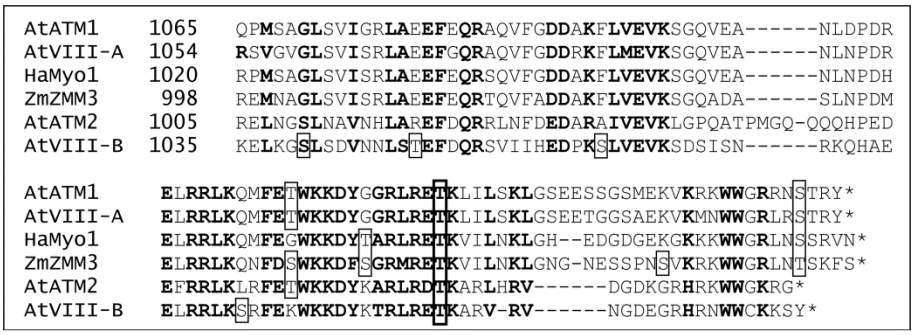
7.5. Vyhledávání dalších motivů v proteinových sekvencích

Funkčně významné motivy v proteinových sekvencích se ovšem neomezují jen na konzervativní domény a lokalizační signály. Důležitou skupinou jsou též sekvenční motivy podléhající **posttranslačním modifikacím**. Často jsou přitom definovány jen velmi krátkým konsensem zasazeným ve velmi proměnlivém okolí. Na jejich možnou biologickou úlohu můžeme proto někdy usuzovat až na základě zjištěné evoluční konzervace, jako v případě možného fosforylačního místa pro proteinkinázu A v rostlinných myozinech třídy VIII (Baluška et al., 2001; Obr.7.3.). Konvenční vyhledávací metody založené na vzorcích či profilech mají v takových případech sklon nacházet mnoho falešných pozitivů. Tak třeba **fosforylační místa** běžných proteinkináz jsou sice zachycena vzorcem v databázi PROSITE, ale tyto vzorce jsou provázeny varováním, že jde o motivy často nacházené a ne nutně funkčně významné.

---

[42] Na uvedené adrese sídlí i další specializované programy jak pro predikci adresování proteinů, tak i pro hledání cílových míst posttranslačních modifikací, a čtenáři vřele doporučuji, aby se tam kolem sebe řádně rozhlédl.

[43] Netechnický výklad principu fungování „neuronových sítí" viz Thagard (1996/2001, kap.7).



Obr.7.3. Identifikace konzervativního fosforylačního místa v rostlinných myosinech třídy VIII (podle Baluška et al., 2001). Fosforylační místa pro různé serin/threoninové kinázy jsou označena rámečky, konzervativní místo rozpoznávané proteinkinázou A rámečkem zesíleným. K vyhledání fosforylačních míst byl použit nyní již nepřístupný program pro prohledávání PhosphoBase (Kreegipuu et al., 1999).

Jako vhodnější pro vyhledávání motivů podobného typu se jeví např. postupy založené na principu virtuálních neuronových sítí „natrénovaných" na vzorových experimentálně ověřených datech. Příkladem je program **NetPhos** (Blom et al., 1999) pro vyhledávání fosforylačních míst, dostupný na již zmiňovaném serveru `http://www.cbs.dtu.dk/services/`. Jako zdroj výchozích dat pro optimalizaci programu posloužila PhosphoBase (Kreegipuu et al., 1999), databáze experimentálně stanovených fosforylačních míst, nyní začleněná do širší databáze PhosphoELM (Diella et al., 2004).[44]

Ne všechny sekvence předurčující proteiny k posttranslační modifikaci jsou takto obtížně vymezitelné. Tak třeba signálem pro **degradaci proteinu** zprostředkovanou proteazomem, kterou můžeme pokládat za extrémní případ posttranslační modifikace, je poměrně jednoduše definovaný „PEST" motiv bohatý prolinem, kys. glutamovou, serinem a threoninem (Rechsteiner a Rogers, 1996). Specializovaný nástroj k vyhledávání PEST motivů **PESTFind** v uživatelských sekvencích je k dispozici na serveru ve vídeňském Biozentru (`https://emb1.bcc.univie.ac.at/`).

Rozšíříme-li pojem motivu i na další specifická uspořádání primární struktury, můžeme za svého druhu motiv považovat i **opakování v rámci sekvence proteinu**. Takováto opakování účinně a vcelku rychle detekuje program **RADAR** (*Rapid automatic detection and alignment of repeats*, Heger a Holm, 2000), dostupný na `http://www.ebi.ac.uk/Radar/`.

Poslední téma, kterého se v této kapitole dotkneme, není již v pravém smyslu vyhledáváním sekvenčních motivů. Rozšíříme-li však pojem motivu i na elementy prostorového uspořádání proteinového řetězce podmíněného poměrně variabilní sekvencí, můžeme sem zařadit i predikci **sekundární struktury** proteinů. K tomuto účelu byla vyvinuta řada algoritmů, které jsou často dostupné i na veřejných serverech.

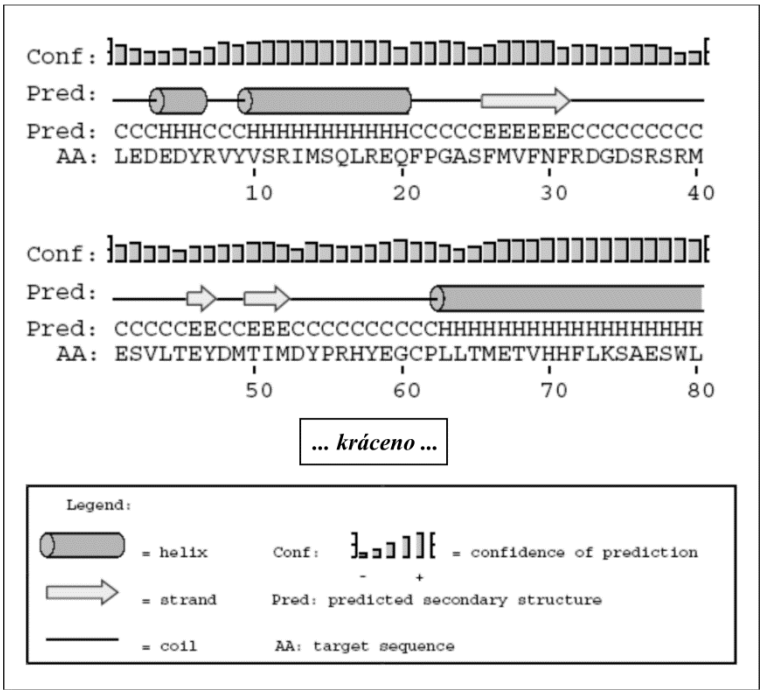
Tak predikce sekundární struktury programem **PSIPRED** (Jones, 1999; McGuffin et al., 2000) na `http://bioinf.cs.ucl.ac.uk/psipred/` využívá postupu založeného variantě PSSM.[45] Praktické začlenění předpovědi sekundární struktury do **strategie**

[44] Na rozdíl od původní PhosphoBase není PhosphoELM dosud prohledávatelná sekvenčním dotazem (viz Obr. 7.3.), i když na této možnosti se údajně pracuje.

[45] Tento program sdílí uživatelské rozhraní s nástrojem na predikci transmembránových úseků proteinu. Zadání zpracovává dávkovým způsobem a výsledky dodává prostřednictvím e-mailu.

analýzy proteinových sekvencí si můžeme demonstrovat právě na konkrétním příkladu využití tohoto programu (Cvrčková et al., 2004b):

- U většiny příslušníků genové rodiny rostlinných forminů třídy II byla nalezena nová, doposud necharakterizovaná doména, u níž BLAST (viz 6.2.) odhalil významnou příbuznost s doménou sdílenou některými proteinfosfatázami, lipidfosfatázami, tensinem a antionkogenem MMAC1/Pten. V některých proteinech byl „sekvenční motiv příbuzný proteinfosfatázám“ nalezen i pomocí SMARTu.
- Pten byl také identifikován jako pravděpodobný nejbližší „strukturní soused“ příslušné domény u několika vybraných rostlinných forminů metodou 3D-PSSM (viz 6.4.).
- K dalšímu ověření byla pro rostlinné forminy získána predikce sekundární struktury metodou PSIPRED, která indikovala rozložení α-helikálních a β-listových úseků velmi podobné experimentálně charakterizované struktuře proteinu Pten (příklad části výstupu pro jeden z analyzovaných proteinů viz Obr.7.4.).
- Na základě právě uvedených výsledků byla struktura proteinu Pten použita jako výchozí vzor (templát) pro konstrukci trojrozměrného modelu nově nalezené domény (viz 8.3.)



Obr.7.4. Příklad výstupu programu PSIPRED.

Predikce sekundární struktury poněkud odlišným algoritmem (Rost et al., 1994) je též součástí široké nabídky metod pro analýzu proteinových sekvencí na serveru **PredictProtein** (`http://www.embl-heidelberg.de/predictprotein/`). Predict Protein však poskytuje i řadu dalších služeb, od prohledávání databází až po predikci trojrozměrné struktury na základě „navlékání“ na homologní sekvenci (viz 8.3.). Výsledky zpracování zadaných úloh jsou zasílány e-mailem, jak už bývá u takto náročných výpočtů běžné.

Podobně jako v případě vyhledávání vzdáleně příbuzných sekvencí i u sekundárních struktur platí, že dvě metody jsou více než jedna. Proto je výhodné používat servery, které umožňují stanovit sekundární strukturu jako **konsensus z více algoritmů**. Tuto možnost poskytuje např. server **NPSA** (Deleage et al., 1997) na `http://npsa-pbil.ibcp.fr` v Lyonu, pozoruhodný mimo jiné svým interaktivním rozhraním. Ještě více možností poskytuje (ovšem za cenu dávkového zadávání úloh) server **Jpred** (Cuff et al., 1998), jehož nabídka programů

neustále roste (http://www.compbio.dundee.ac.uk/~www-jpred/). V současnosti již umožňuje brát při predikci sekundární struktury v úvahu i evoluční konzervaci jednotlivých pozic v proteinu – pokud uživatel nedodá homologní sekvence, program si je pomocí BLASTu najde sám.

8. Konstrukce a interpretace mnohočetného přiřazení

S pojmem **mnohočetné přiřazení (*multiple alignment*)** jsme se již setkali v předchozí kapitole (7.1.), kde jsme se rovněž seznámili s jednou z možných jeho interpretací – totiž s mnohočetným přiřazením jakožto prostředkem definice evolučně konzervovaných či konvergentních sekvenčních motivů. Mnohočetné přiřazení však bývá běžně používáno nejen k porovnávání proteinových sekvencí za účelem vyhledávání konzervativních domén či následných evolučních analýz (viz 10. kapitola), ale i k dalším účelům. Tak třeba mnohočetné přiřazení sekvencí DNA je základem metod pro sestavování delších fragmentů z primárních sekvenčních dat, až po celé genomové sekvence (viz např. Fleischmann et al., 1995). Rovněž při predikci struktury genů, jejichž transkript podléhá sestřihu, vycházíme často z mnohočetných přiřazení na úrovni nukleotidových sekvencí (viz 9. kapitola).

V předchozí kapitole jsme se zabývali výhradně mnohočetnými přiřazeními, která již sestrojil někdo jiný. Mohli jsme proto dosud ponechat stranou otázku, odkud se mnohočetné přiřazení bere – proces jeho tvorby nám byl až dosud černou skříňkou. Je však již na čase tento stav změnit, protože **praktická konstrukce mnohočetných přiřazení** – ať už proteinových, nebo nukleotidových sekvencí – je běžnou součástí každodenní bioinformatické práce.

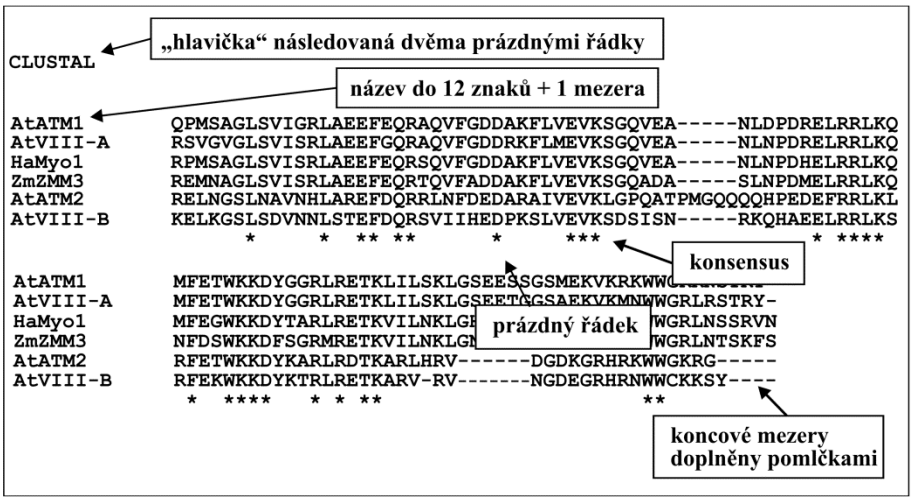
8.1. Zápis mnohočetného přiřazení

Dříve než se budeme věnovat vlastní konstrukci mnohočetných přiřazení, měli bychom se opět, podobně jako v případě sekvencí (kap. 3.1.) a sekvenčních motivů (kap. 7.1.), seznámit s nejběžnějšími **formáty souborů** používanými k zápisu přiřazení – tedy především s formáty, se kterými dovedou pracovat níže diskutované programy, a které si přitom můžeme troufnout napsat i ručně. Teoreticky sice lze mnohočetné přiřazení vytvářet a zpracovávat i v textovém editoru, v praxi se to však z pochopitelných důvodů nedělá. I v případech, kdy přiřazení tvoříme či upravujeme tzv. ručně (viz 8.2.1.), využíváme ve skutečnosti specializovaných programů, které umožňují alespoň jednoduše vzájemně posouvat bloky sekvencí. Tyto programy však vyžadují formátovaná **vstupní data**, která sice mohou sestávat z prostých sekvencí třeba ve formátu FASTA (viz 3.1.), ale mohou již mít i podobu již existujícího rozpracovaného přiřazení (zejména aplikujeme-li na náš soubor sekvencí postupně několik programů).

Pro lidského pozorovatele pravděpodobně nejnázornějším a nejjednodušším formátem mnohočetného přiřazení je **prostý zápis** vzájemně odpovídajících pozic pod sebe s tím, že na začátku každé řádky jsou uvedeny názvy sekvencí (a případné další údaje, jako je třeba pozice fragmentu v rámci delší zdrojové sekvence) a mezery jsou vyznačeny pomlčkami (viz např. Obr.7.3. a Obr.8.8.). Pro strojové zpracování však takový zápis sám o sobě nestačí – bylo by dosti složité přivést program k tomu, aby poznal, kde končí název a kde začíná sekvence, a možným zdrojem zmatků je i případné rozdělení přiřazení do více řádků (bloků). Proto se v praxi používají poněkud přísněji definované formáty, které však lze z právě uvedeného velmi snadno odvodit.

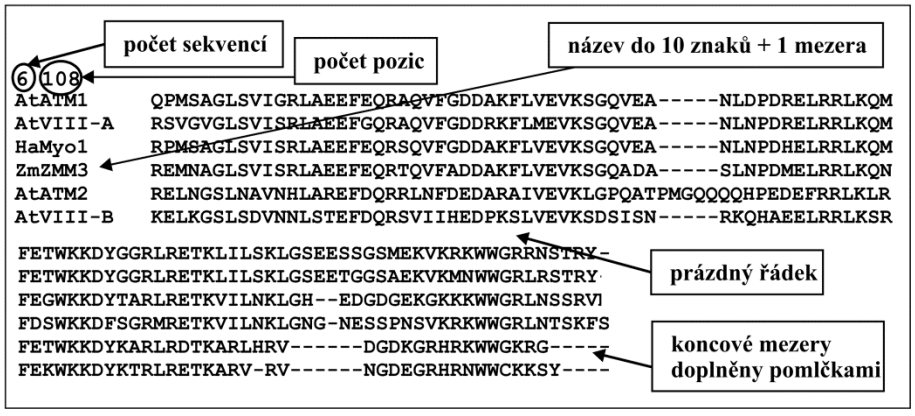
Jedním z nejjednodušších je formát **CLUSTAL** (někdy též `*.aln`, viz Obr.8.1.), nazvaný po stejnojmenném programu, který bude podrobněji diskutován v části 8.2.2. Od prostého zápisu ve výše uvedeném smyslu se liší pouze definovanou délkou názvů sekvencí a rozestupem řádků, hlavičkou, která musí začínat slovem CLUSTAL a může pokračovat krátkým komentářem (do 1 řádku), a přítomností řádku vyznačujícího konsensuální pozice pomocí

volitelných symbolů (vždy ***** pro pozici absolutně konzervovanou, někdy též **:** a **.** pro pozice s různě konzervovaným většinovým zastoupením). Programy, které vyžadují formát CLUSTAL, zpravidla nekontrolují obsah konsensuálního řádku, ale pouze jeho přítomnost, a tolerují i „konsensuální řádek" prázdný nebo obsahující pouze několik mezer. Uživatel tedy může vyrobit přijatelný, byť i formálně ne zcela správný, vstupní soubor i zcela ručně v textovém editoru.



Obr.8.1. Mnohočetné přiřazení ve formátu CLUSTAL (stejné sekvence jako v Obr.7.3.).

Podobně jednoduchý je i jeden z formátů zavedených spolu s fylogenetickým programem PHYLIP (viz 10.4.1.), označovaný střídavě jako **PHYLIP(i)**, případně PHYLIP 4 či PHYLIP bez přívlastků (často s příponou `*.phy`, viz Obr.8.2.). Na rozdíl od předchozího formátu tento nelze „ošidit" – programy, které soubory ve formátu PHYLIP přijímají, kontrolují správnost uvedených údajů o počtu a délce sekvencí.[46]



Obr.8.2. Mnohočetné přiřazení ve formátu PHYLIP(i). Tatáž data jako Obr.8.1. a Obr.7.3.

Poněkud méně přehlednou, ale v některých situacích výhodnější variantou právě uvedených „prostřídaných" (*interleaved*) formátů jsou formáty, v nichž jsou jednotlivé sekvence i s mezerami uváděny odděleně za sebou (formáty postupné – *sequential*). Takový zápis totiž usnadňuje např. výběr pouze některých sekvencí pro další zpracování nebo postupné

[46] Délku sekvence lze snadno zjistit třeba za použití nástroje na počítání znaků v programu MS Word. Opět však je třeba připomenout nezbytnou ostražitost při používání Wordu při práci se sekvencemi a zejména na nezbytnost vypínání funkce automatických oprav (viz 3.1.).

přidávání sekvencí do přiřazení. Asi nejjednodušší z těchto formátů je nám již dobře známý formát **FASTA**, obohacený ovšem o mezery (Obr.8.3.; často s příponou `*.fst`, `*.fas` či `*.fsa`; viz též 3.1.). Podobný formát **PHYLIP(s)**, označovaný někdy též jako PHYLIP 3.2 (rovněž s příponou `*.phy`) získáme „rozpáráním" přiřazení ve formátu PHYLIP(i) podle sekvencí (Obr.8.4.). I u postupných formátů programy trvají na dodržení stejné délky sekvencí v přiřazení – přečnívající konce se doplňují pomlčkami.

```
>AtATM1
QPMSAGLSVIGRLAEEFEQRAQVFGDDAKFLVEVKSGQVEA-----NLDPDRELRRLKQM
FETWKKDYGGRLRETKLILSKLGSEESSGSMEKVKRKWWGRRNSTRY-
>AtVIII-A
RSVGVGLSVISRLAEEFGQRAQVFGDDRKFLMEVKSGQVEA-----NLNPDRELRRLKQM
FETWKKDYGGRLRETKLILSKLGSEETGGSAEKVKMNWWGRLRSTRY-
>HaMyo1
RPMSAGLSVISRLAEEFEQRSQVFGDDAKFLVEVKSGQVEA-----NLNPDHELRRLKQM
FEGWKKDYTARLRETKVILNKLGH--EDGDGEKGKKKWWGRLNSSRVN
>ZmZMM3
REMNAGLSVISRLAEEFEQRTQVFADDAKFLVEVKSGQADA-----SLNPDMELRRLKQN
FDSWKKDFSGRMRETKVILNKLGNG-NESSPNSVKRKWWGRLNTSKFS
>AtATM2
RELNGSLNAVNHLAREFDQRRLNFDEDARAIVEVKLGPQATPMGQQQQHPEDEFRRLKLR
FETWKKDYKARLRDTKARLHRV------DGDKGRHRKWWGKRG-----
>AtVIII-B
KELKGSLSDVNNLSTEFDQRSVIIHEDPKSLVEVKSDSISN-----RKQHAEELRRLKSR
FEKWKKDYKTRLRETKARV-RV------NGDEGRHRNWWCKKSY----
```

Obr.8.3. Mnohočetné přiřazení ve formátu FASTA. Tatáž data jako Obr.8.1., 8.2 a 7.3.

```
6 108
AtATM1
QPMSAGLSVIGRLAEEFEQRAQVFGDDAKFLVEVKSGQVEA-----NLDPDRELRRLKQM
FETWKKDYGGRLRETKLILSKLGSEESSGSMEKVKRKWWGRRNSTRY-
AtVIII-A
RSVGVGLSVISRLAEEFGQRAQVFGDDRKFLMEVKSGQVEA-----NLNPDRELRRLKQM
FETWKKDYGGRLRETKLILSKLGSEETGGSAEKVKMNWWGRLRSTRY-
HaMyo1
RPMSAGLSVISRLAEEFEQRSQVFGDDAKFLVEVKSGQVEA-----NLNPDHELRRLKQM
FEGWKKDYTARLRETKVILNKLGH--EDGDGEKGKKKWWGRLNSSRVN
ZmZMM3
REMNAGLSVISRLAEEFEQRTQVFADDAKFLVEVKSGQADA-----SLNPDMELRRLKQN
FDSWKKDFSGRMRETKVILNKLGNG-NESSPNSVKRKWWGRLNTSKFS
AtATM2
RELNGSLNAVNHLAREFDQRRLNFDEDARAIVEVKLGPQATPMGQQQQHPEDEFRRLKLR
FETWKKDYKARLRDTKARLHRV------DGDKGRHRKWWGKRG-----
AtVIII-B
KELKGSLSDVNNLSTEFDQRSVIIHEDPKSLVEVKSDSISN-----RKQHAEELRRLKSR
FEKWKKDYKTRLRETKARV-RV------NGDEGRHRNWWCKKSY----
```

Obr.8.4. Mnohočetné přiřazení ve formátu PHYLIP(s). Tatáž data jako Obr.8.1.,8.2.,8.3. a 7.3.

Jak již bylo řečeno, k úpravám formátu datových souborů lze používat jakýkoli textový editor.[47] Chceme-li se však vyhnout pracné a monotónní činnosti, a současně snížit riziko výskytu chyb v souborech, je výhodou využít softwarové **nástroje na konverzi formátů** mnohočetných přiřazení. První vstupní soubor ale musíme tak jako tak vyrobit sami buď ručně, nebo konstrukcí přiřazení *de novo* např. ze sekvencí ve formátu FASTA ve vhodném programu.

[47] Při náležité opatrnosti je dosti dobrým nástrojem MS Word, který dovoluje i výběr a přesuny vertikálních bloků textu kombinací Alt + levá myš.

Některé níže probírané programy (např. BioEdit) dovolují import dat v jednoduchých formátech, jako je FASTA či CLUSTAL, a jejich export v řadě formátů dalších, i mnohem komplikovanějších než jsou právě popisované. Specializovaný jednoúčelový nástroj na konverzi formátů **ForCon** (Raes a Van de Peer, 1999) je rovněž součástí novějších verzí balíčku fylogenetických programů TreeCon (Van de Peer a De Wachter, 1994).[48]

8.2. Ruční a automatizované postupy konstrukce mnohočetného přiřazení

Mnohočetná přiřazení sestavujeme zpravidla buď proto, abychom na základě evoluční konzervace usuzovali na **funkční význam konkrétních pozic** (např. identifikovali aktivní místa enzymů nebo regulačně významná místa posttranslačních modifikací – viz Obr.7.3.), nebo abychom na jejich základě vyvozovali **hypotézy o evoluční minulosti** studované genové rodiny.

V obou případech však má smysl přiřazovat pouze sekvence, které jsou si vskutku příbuzné (podobně jako v případě prostého přiřazení dvou sekvencí – viz kap. 5.2.).[49] Na nás tedy je, abychom rozhodli, zda sekvence ve zkoumaném souboru splňují podmínku příbuznosti po celé délce (a lze na ně tedy aplikovat metody pro konstrukci **přiřazení globálního**), nebo zda sdílejí pouze izolovanou doménu či domény, a musíme se tudíž spokojit s **přiřazením lokálním**. Vodítkem k takovému rozhodnutí může být třeba párové porovnání sekvencí (každá s každou). Existují však i metody, které umožňují přímo vyhledávat bloky lokální podobnosti většího počtu sekvencí (viz níže).

Obecnou strategii konstrukce mnohočetného přiřazení bychom, bez ohledu na konkrétní software, mohli shrnout zásadou, že postupujeme **vždy od bližších sekvencí ke vzdálenějším**. To znamená, že nejprve spojujeme dvojice sekvencí, které jsou si vzájemně nejpodobnější v globálním přiřazení, a úseky, které vykazují nejvyšší podobnost v přiřazení lokálním. Vzdálenější sekvence a úseky pak postupně „nabalujeme“ na tuto kostru. V principu ovšem lze přiřadit téměř cokoli k čemukoli, zejména pokud připustíme zavádění krátkých mezer, a proto je důležité vědět, kdy přestat. Dobrým indikátorem upadající kvality přiřazení je právě přítomnost většího počtu mezer blízko u sebe.

V následujícím výkladu se budeme věnovat především **proteinovým sekvencím**; základní strategie i některé probírané programy však jsou použitelné i pro sekvence nukleotidové.

8.2.1. CLUSTAL jako modelový automatizovaný postup

Nutnou podmínkou úspěšné automatizace tvorby mnohočetného přiřazení je vytvoření spolehlivého **algoritmu**. Jak jsme již viděli, ani pro přiřazení pouhých dvou sekvencí není

[48] Program TreeCon je poměrně starý, napsaný původně pro Win98 a WinNT a v novějších verzích Windows, zejména Windows XP, se chová problematicky. Dle zkušeností autorky ForCon pod WinXP SP2 funguje, pokud je celý balík TreeCon nainstalován v adresáři `C:\treeconw` na systémovém disku (tedy nikoli v Program Files). ForCon také někdy po skončení zanechává v paměti neukončené procesy, které je radno ukončit ručně, protože mohou vyvolávat konflikty např. s MS Office.

[49] Kritériem příbuznosti může být např. dostatečně nízká hodnota E stanovená programem BLAST (viz 6.2.).

obecné řešení této úlohy triviální (kap. 5.). U mnohočetného přiřazení pak přistupují další problémy, zejména dramatický nárůst výpočetní náročnosti (zhruba kvadraticky s počtem sekvencí). Zejména s ohledem na tento jev mají prakticky všechny reálně použitelné algoritmy heuristickou (přibližnou) povahu, podobně jako je tomu v případě prohledávání databází (kap. 6.).

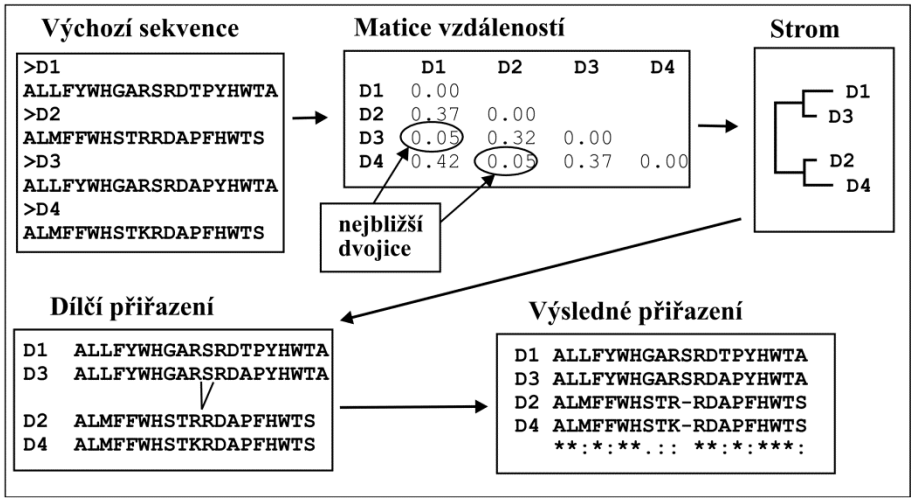
Konkrétní příklad strategie pro plně automatickou konstrukci mnohočetného přiřazení si můžeme ukázat na příkladu algoritmu CLUSTAL, na němž je založen často používaný a všeobecně dostupný program **Clustal W** (Thompson et al., 1994). S tímto programem se můžeme setkat v řadě verzí pro různé operační systémy, které se liší uživatelským rozhraním, přednastavenými (*default*) parametry, a výběrem parametrů, které uživatel může interaktivně měnit. Některé varianty jsou dostupné prostřednictvím veřejných webových serverů. Pro soustavnou práci však nepochybně stojí za to opatřit si lokální kopii programu; lze doporučit např. freewarový balíček **Clustal X** (Thompson et al., 1997), což je Clustal W opatřený grafickým rozhraním pro Windows a obohacený o „nadstavbové“ možnosti – formátování vstupů a výstupů, konstrukci dílčích přiřazení z části sekvencí a jednoduché fylogenetické analýzy (viz též kap. 10). Clustal X přijímá vstupní data v několika formátech včetně FASTA z textového souboru o libovolné extenzi (pro potřeby fylogenetických analýz či revize lze importovat i již existující přiřazení, např. ve formátu CLUSTAL).

Postup používaný při konstrukci mnohočetného přiřazení metodou CLUSTAL lze shrnout do následujících kroků (Obr.8.5.):

- Program nejprve vzájemně porovná všechny možné dvojice vložených sekvencí a z výsledků sestrojí **matici párových normalizovaných hodnot vzdálenosti**. Pro vlastní stanovení párových hodnot vzdálenosti může uživatel zvolit buď rychlou, ale méně přesnou metodu, která vychází z hodnot podobnosti (*score*) odpovídajících nejdelším diagonálám bodového diagramu (viz kap. 5), nebo pomalejší, avšak přesnější postup, který zahrnuje optimalizaci párového přiřazení pro každou dvojici sekvencí. Tento krok je z celého postupu nejpomalejší a jeho trvání roste s počtem sekvencí kvadraticky (pro N sekvencí je nutno provést N(N-1)/2 porovnání). Proto některé serverové varianty CLUSTALu nedovolují používat pomalejší přesnou metodu konstrukce párových přiřazení.
- Na základě matice podobností program provede shlukovou analýzu a sestrojí **„vůdčí strom“** (*guide tree*) – tedy dendrogram, který postupně sdružuje vložené sekvence podle míry vzájemné podobnosti od nejbližších ke vzdálenějším.[50]
- Nakonec program sestrojí **vlastní přiřazení** tak, že nejprve vytvoří párová globální přiřazení těch sekvencí, které jsou ve vůdčím stromu bezprostředními sousedy. V následujících krocích postupně přiřazuje další sekvence (popřípadě párová či mnohočetná přiřazení) v pořadí daném vůdčím stromem.

CLUSTAL vytváří **globální přiřazení** (viz 5.2) se všemi jeho výhodami i nedostatky. Z toho vyplývá, že je principiálně vhodný pouze pro práci se sekvencemi, které lze vzájemně přiřadit po celé délce. Pro takové sekvence však CLUSTAL rychle a jednoduše napoprvé poskytne důvěryhodný výsledek, jehož součástí může být i matice vzdáleností.[51]

[50] Ve skutečnosti CLUSTAL používá postup o něco složitější, než ukazuje Obr. 8.5. - totiž metodu NJ (*neighbor joining*), která bude diskutována v 10. kapitole (10.2.).
[51] Clustal X umožňuje i export vůdčího stromu; ten však postrádá statistické podložení a neměl by být v publikacích prezentován (viz 10.3.).



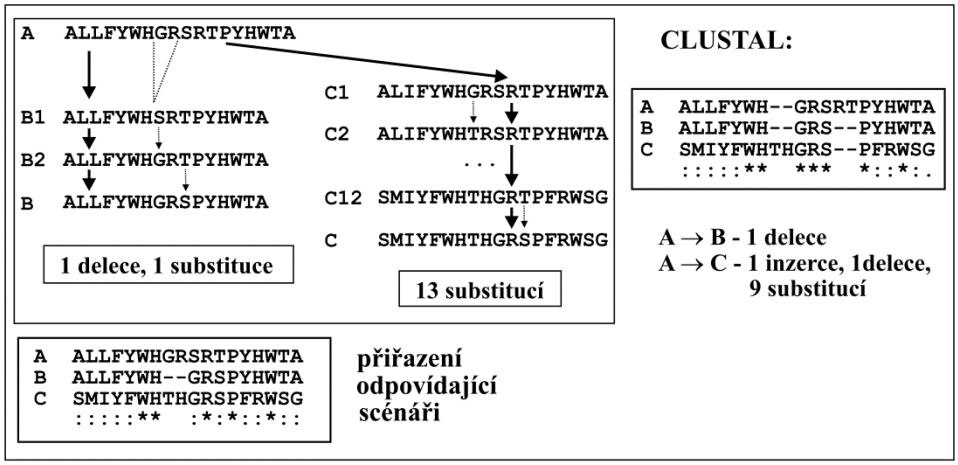
Obr.8.5. Postup konstrukce mnohočetného přiřazení algoritmem CLUSTAL.

Dobrými kandidáty na rutinní používání CLUSTALu jako preferované metody jsou sekvence příslušníků dobře evolučně konzervovaných genových rodin s minimem mezer a více než polovinou konzervovaných pozic. Pokud studované sekvence v průběhu evoluce podlehly výměnám domén či přestavbám jejich pořadí, lze CLUSTAL použít pouze na předem vybrané konzervativní domény.

Postup vlastní konstrukce přiřazení v posledním kroku s sebou nese nevyhnutelný zdroj občasných, avšak dosti nepříjemných **artefaktů**. Příčinou je skutečnost, že CLUSTAL umí mezery pouze zavádět, ne však odstraňovat. Přitom již při tvorbě párových přiřazení nejbližších dvojic se program občas musí rozhodovat mezi prakticky rovnocennými variantami umístění mezery. Případný „omyl“ sice může vyjít najevo v dalších krocích, avšak vzhledem k tomu, že **mezery jsou navždycky**,[52] jej nelze odstranit, toliko kompenzovat zavedením dalších mezer (Obr.8.6.). Výskyt většího počtu mezer blízko u sebe bychom tedy měli vždy považovat za varování. Výskyt „mezerových“ artefaktů lze omezit (i když ne úplně vyloučit) vhodnou volbou uživatelského nastavení (viz níže), a Clustal X dokonce umožňuje „přepočítat“ a znovu přiřadit vybrané úseky sekvencí se změněnými parametry. Přesto nám však někdy nezbývá než „doladit“ přiřazení ručně; tento postup je obecně přijímaný (Barton, 2001), samozřejmě s tím, že se sluší řádně jej uvést v popisu metod.

Většina implementací CLUSTALu umožňuje **uživatelské nastavení** některých parametrů. Patří mezi ně zejména volba rychlého nebo pomalého algoritmu v prvním porovnávacím kroku a volba empirických parametrů (substituční matice, cena mezer) jak pro počáteční párové porovnávání, tak i pro závěrečný krok vlastní konstrukce přiřazení mnohočetného. Za pozornost stojí, že pouze pro první krok – párové porovnání sekvencí – uživatel vybírá konkrétní substituční matici. Pro závěrečnou fázi však volí jen sérii matic (GONNET, PAM, BLOSUM) s tím, že program sám vybere matici přiměřenou zjištěné podobnosti konkrétního páru sekvencí.

[52] Ian Fleming nechť se v hrobě neobrací... vím samozřejmě, že něco takového říkal o démantech!



Obr.8.6. Hypotetický scénář evoluce sekvencí B a C z A a přiřazení vytvořené CLUSTALem – příklad artefaktu vzniklého nemožností odstranit jednou zavedené mezery.

8.2.2. Poloautomatické a ruční metody

Navzdory existenci řady algoritmů pro konstrukci mnohočetného přiřazení, z nichž CLUSTAL ještě patří mezi ty jednodušší, **ruční konstrukce přiřazení** (nebo aspoň jeho ruční modifikace) je dodnes uznávaným postupem (Barton, 2001; Brinkman a Leipe, 2001).[53] S vývojem automatických či poloautomatických metod ovšem význam ruční konstrukce přiřazení poněkud upadá; ještě v roce 1995 však mohla být pouze na základě ručně sestrojeného přiřazení, vyrobeného doslova tužkou na papíře, zjištěna např. přítomnost PH (*pleckstrin homology*) domény v sekvenci kvasinkové proteinkinázy Cla4p (Cvrčková et al., 1995).

Konstrukce přiřazení tužkou na papíře, nebo třeba v textovém editoru, ovšem není právě praktická; lépe je využít nějaký specializovaný program, který dokáže přinejmenším přesouvat celistvé bloky zasahující několik sekvencí, aniž by se zbytek přiřazení rozsypal. Příkladem takového programu je zdarma distribuovaný software **BioEdit** (Hall, 1999), který kromě dobrého editoru na ruční konstrukci a úpravy mnohočetných přiřazení nabízí řadu dalších funkcí. Editor sám má sice ne zcela intuitivní grafické rozhraní, na které si uživatel musí zvyknout, avšak umožňuje upravovat samotnou sekvenci v průběhu práce. To může být výhodou třeba při skládání fragmentů nukleotidových sekvencí z primárních elektroforetogramů, kde o obsazení nejednoznačných pozic lze často rozhodnout až po porovnání nezávislých sekvenovacích reakcí. I při práci s proteinovými sekvencemi je někdy výhodou možnost „ořezání“ nepřiřaditelných translokovaných domén (viz též 10.4.). Editor rovněž umožňuje „obarvit“ sekvenci podle prostorových a chemických vlastností aminokyselin, jak je ostatně u programů této třídy obvyklé; barevné odlišení

[53] Barton (2001, s.218) přímo uvádí, že „*každý soubor sekvencí přináší jedinečné biologicky zaměřené otázky, a konstrukce skutečně kvalitního přiřazení musí být vedena zkušeností.*“, a doporučuje využívat pokud možno srovnání s publikovanými výsledky práce kvalifikovaných expertů. Obecný problém, do jaké míry a za jakých předpokladů je tvorba přiřazení v principu algoritmizovatelná, viz též Cvrčková a Markoš (2005).

aminokyselinových zbytků pak může zcela zásadně napomoci odhalování nesrovnalostí v přiřazení, zejména pokud se naučíme dívat se na sekvence rozostřeně (Obr.8.7.).

Obr.8.7. Barevné odlišení jednotlivých skupin aminokyselin může výrazně napomoci při hledání pravidelností. Sloupce konzervativních pozic nejsnáze vyniknou, podíváme-li se na přiřazení rozostřeně tak, aby sekvence právě přestaly být čitelné (dole). Převzato z Cvrčková a Markoš (2005).

BioEdit přijímá **vstupní data** a ukládá soubory v řadě běžných formátů včetně FASTA a obou variant formátu PHYLIP. Umí číst (ale ne zapisovat) i soubory ve formátu CLUSTAL, takže je vhodný i k dodatečným ručním úpravám přiřazení vyrobeného např. pomocí programu Clustal X. BioEdit dokáže také propojit dvě existující přiřazení v jedno prostřednictvím sdílené sekvence, což je výhodou zejména při práci na velkých projektech, jako jsou analýzy genových rodin s více podrodinami. Některé funkce programu jsou specifické pro nukleotidové sekvence (BioEdit zahrnuje i jednoduchý editor na **konstrukci grafických map** plazmidů). Zajímavá je možnost importovat nukleotidové sekvence přímo z primárních elektroforetogramů dodaných sekvenátorem (formát `*.abi` nebo `*.ab1`, viz 3.1.), a začlenění programu pro vyhledávání překryvů jednotlivých sekvenovacích běhů a konstrukci konsensuální sekvence **CAP** (*contig assembly program*).

Právě poslední vzpomínaná funkce je příkladem **„zásuvného modulu“** (*plugin*) – totiž původně nezávislého programu, který byl začleněn do BioEditu a opatřen intuitivním uživatelským rozhraním, které umožňuje snadný import právě otevřeného souboru a export výsledků zpět do editoru přiřazení. BioEdit obsahuje bohatou nabídku předkonfigurovaných zásuvných modelů a umožňuje uživatelskou instalaci dalších. Součástí standardní nabídky je např. BLAST (s možností konfigurace vlastní lokální databáze či prohledávání databáze NCBI prostřednictvím Internetu) a některé fylogenetické nástroje z nabídky softwarového balíku PHYLIP (viz 10.4.). Z hlediska tématu této kapitoly zvláště významné je to, že jedním

ze standardních (předinstalovaných) zásuvných modulů BioEditu je také **Clustal W**.[54] Verze CLUSTALu začleněná do BioEditu sice poskytuje o něco chudší možnosti než implementace v programu Clustal X (neumožňuje například revizi části přiřazení se změněnými parametry), ale pro většinu prakticky významných situací bohatě postačí. Uživatel se tedy může těšit z výhod obou cest konstrukce přiřazení – může totiž nejprve vytvořit výchozí přiřazení automatizovaným algoritmem CLUSTAL a pak je dále ručně upravovat.

Zcela jinou strategii kombinace automatických a ručních postupů představuje sice již poněkud zastaralý, ale v některých ohledech dosud nepřekonatelný program **MACAW** (*Multiple Alignment Construction and Analysis Workbench*; Schuler et al., 1991; Lawrence et al., 1993), který umožňuje vyhledávat a interaktivně editovat **bloky lokální podobnosti** studovaných sekvencí. Hodí se především pro rychlé posouzení vzájemné podobnosti sekvencí, o nichž nic nevíme, pro vyhledávání jednotlivých konzervovaných domén v rámci složitějších studovaných proteinů, a také třeba pro rychlé mapování exonů transkriptu na genomovou sekvenci. Jeho silnou stránkou jsou totiž situace, kdy zákonitě selhávají postupy založené na globální podobnosti a globálním přiřazení.

Práce v programu MACAW začíná založením **projektu**, u něhož je nutno hned zpočátku zadat typ sekvencí (DNA/RNA/protein) a substituční matici.[55] Vstupní data lze buď přímo vložit ze schránky (*clipboard*) nebo importovat z textového souboru ve formátu FASTA s tím, že program trvá na extenzi `*.nt` pro nukleotidové sekvence a `*.aa` pro sekvence aminokyselinové (viz též 3.1.). Pozdější editování sekvencí není možné, lze však celé sekvence odstraňovat a nové (případně pozměněné) přidávat. Vložené sekvence se objeví v centrálním grafickém okně (*Schematic*) jako schematické obdélníčky bez vypsaných pozic (ty lze kdykoli zviditelnit otevřením okna přiřazení – *Alignment*), a lze nastavit jejich barevné stínování dle vzájemné podobnosti. Jakmile uživatel pomocí myši vybere nejméně dvě sekvence či jejich části,[56] může spustit funkci vyhledávání bloků podobnosti ve vybraných úsecích (*Alignment – Search for blocks*). Má přitom k dispozici dvě metody. Jednou z nich je prosté **vyhledávání přiřaditelných bloků** sdílených dvěma či více sekvencemi a dosahujících alespoň minimální hodnoty podobnosti (minimální počet sekvencí i hodnotu podobnosti lze nastavit), druhou je tzv. **Gibbsovo vzorkování** (*Gibbs sampler*), které identifikuje krátké a roztroušené motivy sdílené všemi zkoumanými sekvencemi. Výsledky vyhledávání jsou opatřeny odhadem statistické významnosti v podobě hodnoty pravděpodobnosti E srovnatelného náhodného výsledku. Uživatel může nalezený blok na základě vizuální inspekce sekvencí zúžit nebo rozšířit, a zejména může sekvence v rámci bloku „provázat“ (*Link*), takže se nadále budou pohybovat spolu; mnohočetné přiřazení vzniká právě postupným provazováním bloků. Bloky lze definovat a provazovat také ručně, i v okně přiřazení. Výsledky mohou být exportovány do textového souboru (bez barvy a grafiky) nebo na tiskárnu, kterou ovšem může být třeba Acrobat Distiller.

MACAW může vzácně generovat **statistické artefakty**. Pokud zkoumaný soubor sekvencí obsahuje na srovnatelných pozicích různé a vzájemně nepříbuzné domény, z nichž každá sama o sobě zajistí statistickou průkaznost nalezeného bloku, může se stát, že program ohlásí „kombinovaný blok“ ze dvou vzájemně nepodobných domén. Tento artefakt lze ovšem snadno odhalit jednak vizuální inspekcí, jednak tím, že eliminace jedné z přispívajících domén výrazně zvýší průkaznost.

[54] Pozor, ve verzích BioEditu starších než 7.0.1. (srpen 2004) tato funkce nepracuje pod Windows XP.
[55] Pokud tyto hodnoty nezadáme, program si pamatuje naposledy použité nastavení a se zárukou se zhroutí v případě vložení nesprávného typu sekvence.
[56] K výběru nesouvislého bloku („ob sekvenci“) slouží pravé tlačítko myši.

Nevýhodou programu MACAW je také poněkud „exotický“ výstupní formát, který neodpovídá běžným standardům a pro případné další strojové zpracování musí být ručně editován do formátu CLUSTAL či PHYLIP(i),[57] omezená kapacita co do počtu i délky sekvencí, a občas záludná nestabilita (je nutno dávat pozor, aby přiřazení po celou dobu zůstávalo vnitřně konzistentní a abychom nějaký blok neprovázali „křížem“, a vyplatí se často ukládat mezivýsledky).

8.2.3. Konstrukce přiřazení pomocí programů na veřejných serverech

Konstrukce dobrého mnohočetného přiřazení je, dokonce i za použití vhodných heuristických metod, úkolem **náročným na čas** uživatelský i hardwarový, a pokud se této činnosti chceme věnovat systematicky, zajisté stojí za to pořídit si lokální software. Přesto však existují veřejné servery, poskytující tuto službu. Využívají často velmi sofistikovaných a poměrně neobvyklých postupů a mohou být nápomocné při odhalování ne zcela zřejmých společných rysů sekvencí, které by programy jako CLUSTAL nebo MACAW mohly zmeškat. Uživatel ovšem musí být připraven na nikoli nejrychlejší odezvu, popřípadě i na doručení výsledků e-mailem.

S některými takovými programy jsme se ostatně již setkali. Prohledávací metoda **PSI-BLAST** (6.2.1.) zahrnuje *de facto* konstrukci mnohočetného přiřazení poněkud neobvyklého druhu – na rozdíl od všech ostatních zde probíraných případů je zde totiž délka přiřazení dána délkou dotazu (*query*), do něhož se nesmějí zavádět mezery – a také poněkud jednoúčelového využití. Program **BLOCKMAKER** na serveru databáze BLOCKS (7.2.) poskytuje nástroj pro vyhledávání krátkých konzervativních úseků (domén) v uživatelem zadaných sekvencích a tvorbu lokálního přiřazení.

Z hlediska snadné dostupnosti, jednoduchého rozhraní, zajímavých výstupů a poměrně rychlé odezvy stojí za zmínku další dva programy dostupné prostřednictvím veřejných serverů. Algoritmus **T-Coffee** (*Tree-based Consistency Objective Function for Alignment Evaluation*, Notredame et al., 2000), dosažitelný např. prostřednictvím odkazu ze stránek Evropského bioinformatického institutu, představuje zajímavou alternativu CLUSTALu – totiž plně automatizovaný postup pro konstrukci globálního mnohočetného přiřazení, sice pomalejší než CLUSTAL, avšak odolnější vůči „mezerovému“ artefaktu. Podle autorů programu i podle osobní zkušenosti T-Coffee dává rozumně vypadající výsledky i pro sekvence, které jsou pro CLUSTAL již problematické (kolem 30 % identických pozic). Uživatel se ovšem musí smířit s tím, že program se chová jako černá skříňka bez možnosti uživatelského zásahu do nastavení. T-Coffee přijímá vstupní data ve formátu FASTA a vrací výstup v několika textových formátech (včetně CLUSTAL a PHYLIP) i graficky.

K srovnatelným výsledkům jako T-Coffee při významně menších nárocích na strojový čas lze dospět pomocí novějšího programu **MUSCLE** (*Multiple Sequence Comparison by Log-Expectation*, Edgar, 2004), který je dostupný jak v serverové verzi, tak i ke stažení a vlastní

[57] Alternativou je „vypreparovat“ sekvence z mezerami, použitelné jako základ přiřazení ve formátu FASTA nebo PHYLIP(s), přímo ze souboru projektu (`*.mcw`), který je v textovém formátu. Sekvence se skrývají v řádcích nadepsaných „`gap-seq`“, jejich názvy se nacházejí na začátku souboru a jsou označeny identifikačním klíčem „`id`“, jehož hodnota odpovídá „`seq-id`“ u sekvencí samotných.

instalaci na `http://www.drive5.com/muscle/`, a v lokální verzi dovoluje i změny přednastavených parametrů.

Zvláště pro konstrukci přiřazení vzdáleně příbuzných sekvencí byl v laboratořích firmy IBM vyvinut algoritmus **MUSCA** (Parida et al., 1998; Parida et al., 1999),[58] dostupný na adrese `http://cbcsrv.watson.ibm.com/Tmsa.html`. Pracuje dvoustupňově, prvním krokem je vyhledávání konzervativních motivů způsobem, který trochu připomíná Gibbsovo vzorkováni (aspoň co se strategie týče), druhým vlastní konstrukce přiřazení. Vstupní data MUSCA přijímá ve formátu FASTA (netoleruje však mezery uvnitř datového souboru, dokonce ani na konci řádků), a dovoluje uživatelskou volbu „příbuzenských rodin" aminokyselin. Program je pozoruhodně citlivý na dobře konzervované motivy z „roztroušených" aminokyselin, občas však – poněkud překvapivě – míjí zjevně homologní úseky.

8.3. Přiřazování trojrozměrných struktur a základy modelování struktury proteinů

Predikce **trojrozměrné struktury proteinů** na základě sekvence představuje poměrně zásadní problém, který v obecné rovině dosud není uspokojivě vyřešen. Důležitou součástí problému je i otázka, do jaké míry vůbec je v sekvenci obsažena informace o trojrozměrné struktuře proteinu.

Přehnaně optimistická extrapolace z omezeného souboru proteinů, které přežijí denaturaci a renaturaci (někdy dokonce i v rukou nezkušeného studenta v praktiku), vedla v 60. a 70. letech k představě, že v obecném případě sekvence aminokyselin bezvýhradně určuje prostorové uspořádání proteinu. Taková představa ostatně dobře zapadala do konvenčního obrazu živých bytostí jako extrémně složitých strojů, jejichž struktura (a z ní vyplývající funkce) je beze zbytku zakódována v DNA. Dnes však jsme si již vědomi některých omezení tohoto konvenčního pohledu, a víme také, že u mnohých proteinů k ustavení funkční struktury významně přispívá specifické vnitrobuněčné prostředí včetně specializovaných proteinů (podrobněji viz např. Kanehisa, 2000, s.20). Přesto však zřejmě můžeme bezpečně předpokládat, že **proteiny o podobné sekvenci budou mít sklon zaujímat podobné prostorové uspořádání**.

Přímým důsledkem tohoto předpokladu je výrazné usnadnění predikce trojrozměrných struktur proteinů v případech, kdy již známe experimentálně zjištěnou prostorovou strukturu proteinu o podobné sekvenci. V takovém případě totiž můžeme použít experimentálně zjištěnou strukturu jako „formu" – **templát**, podle kterého nejprve virtuálně „poskládáme" řetězec odpovídající zkoumané sekvenci a pak pouze drobnými úpravami vyřešíme případné strukturní či energetické konflikty (např. elektrostatické pnutí či prostorovou kolizi aminokyselinových zbytků). Postupy založené na právě popsaném principu bývají označovány jako **navlékací algoritmy** (*threading algorithms*). Na navlékání samotné se můžeme dívat jako na speciální typ konstrukce přiřazení – totiž „**přiřazení mezi sekvencí a strukturou**".[59] Navíc se dotkneme i otázky vzájemného přiřazování struktur jako takových –

[58] Podle autorů MUSCA znamená „jisté souhvězdí jižní oblohy, a také anagram zkratky *Constrained Multiple Sequence Alignment*".

[59] Při důsledném dodržování koncepce nastíněné v Cvrčková a Markoš (2005) by tato formulace mohla být právem považována za vnitřně rozpornou. Budeme ji však chápat jako stručné vyjádření přesnější, avšak méně praktické formulace „přiřazení mezi aminokyselinovým řetězcem o známé sekvenci a neznámé struktuře a řetězcem o známé sekvenci i struktuře".

výhodou totiž je, pokud pro danou úlohu máme k dispozici několik vzájemně příbuzných templátů a můžeme tedy využít znalosti konzervovaných oblastí struktury.

Podrobnější analýza těchto metod by přesahovala rámec této práce. Jako již několikrát, i zde se nyní spokojíme s nástinem j**edné z možností postupu** modelování struktury vybrané proteinové domény. Následující návod je rozumně použitelný hlavně pro krátké sekvence, je celkem snadno přístupný i začátečníkovi, a může poskytnout obecnější představu o používaných strategiích.

Popisovaný postup je založen na využití modelovací služby **SwissModel** (`http://swissmodel.expasy.org`) ve spojitosti se zdarma stažitelným klientským programem **DeepView**, který umožňuje uskutečnit první – časově poměrně náročné – kroky postupu na uživatelském počítači (Guex a Peitsch, 1997).[60] Obdobný, i když trochu jednodušší, postup byl použit např. pro modelování nově identifikované domény příbuzné antionkogenu Pten a tenzinu v rostlinných forminech třídy II (Cvrčková et al., 2004b), kde byl k dispozici jen jediný strukturní templát.

- Prvním, a pro úspěch modelování naprosto zásadním, krokem je **nalezení vhodného strukturního templátu** (nejlépe více než jednoho). Používané postupy byly diskutovány v kapitole 6.4. – lze doporučit například program 3D-PSSM či CPHmodels, je však radno vyzkoušet více metod. O vhodnosti templátu může kromě statistických hodnot dodaných hledacím programem vypovídat také třeba shoda předpovědi sekundární struktury pro zkoumanou sekvenci se známou strukturou templátu (viz 7.5.).
- Nalezené struktury dvou nejlepších templátů (ve formátu `*.pdb`) potom otevřeme v samostatných „vrstvách“ programu DeepView. V hlavním okně se struktury objeví přes sebe, k navigaci mezi vrstvami i k nastavení barev a způsobu zobrazení (postranní řetězce a podobně) slouží okno „*Control panel*“, k inspekci sekvencí a jejich přiřazení okno „*Alignment*“. Pokud je vše v pořádku (tj. pokud naše templáty neodpovídají různým fragmentům sekvence), vzájemná podobnost obou templátů se projeví již při vizuální inspekci. Nyní pomocí příkazu „*Fit – Iterative magic fit*“ **přiložíme templáty na sebe** v prostoru. V okně „*Alignment*“ se nám současně objeví přiřazení sekvencí, které odpovídá zobrazenému přiřazení struktur.
- V dalším kroku **importujeme zkoumanou sekvenci** ve formátu FASTA (příkazem „*SwissModel – Load raw sequence to model*“). Je velmi důležité importovat pouze tu část zkoumané sekvence, která odpovídá templátu. Vyplatí se ověřit si to nanečisto konstrukcí mnohočetného přiřazení mezi zkoumanou sekvencí a templáty, a v případě potřeby zkoumanou sekvenci „ořezat“. Je-li templát opravdu vhodný, podobá se zpravidla zkoumané sekvenci natolik, že mnohočetné přiřazení můžeme bez problémů sestrojit. Řetězec popsaný zkoumanou sekvencí se objeví v nové vrstvě v podobě jediného dlouhého α-helixu.
- Pomocí příkazu „*Fit – Fit raw sequence*“ **„navlečeme“ zkoumaný řetězec na templát** a vytvoříme hrubý výchozí model. Vizuální inspekcí přiřazení okna „*Alignment*“ můžeme ověřit, zda přiřazení vypadá rozumně, a případně i provést drobné úpravy. Vodítkem nám může být zvýraznění konfliktních pozic prostřednictvím nabídky

[60] Odkazy na další použitelné metody, včetně plně automatických, najde čtenář na webových stránkách přednášky M. Potockého v rámci kursu bioinformatiky (`http://kfrserver.natur.cuni.cz/bioinfo.html`). Martinovi Potockému děkuji též za návod, z něhož je odvozeno toto zpracování. Podrobné instrukce k obsluze programu DeepView čtenář najde na `http://swissmodel.expasy.org/spdbv/`.

„*Select*“ a zejména graf znázorňující energetickou náročnost navrženého uspořádání v rozbalovacím panelu okna „*Alignment*“. Křivka grafu vyjadřuje zhruba množství energie, kterou je nutno dodat, aby aminokyselina zaujala navrženou konformaci, a měla by být pokud možno pod nulovou hladinou (souvislá čára).

- V dalším kole úprav optimalizujeme „konfliktní“ **postranní řetězce** (opět zvýraznitelné příkazem z nabídky „*Select*“) pomocí příkazu „*Tools – Fix selected sidechains*“, který umožňuje vybrat varianty z knihovny rotačních izomerů („rotamerů“) jednotlivých aminokyselinových postranních řetězců.
- Pro konečnou optimalizaci odešleme ručně doladěný model ke konečné optimalizaci na **server SwissModel** („*SwissModel – Submit modelling request*“) a počkáme si na výsledek, který přijde elektronickou poštou ve formátu `*.pdb`.
- Konečným krokem je **vyhodnocení kvality navrženého modelu**, např. za pomoci programu WHAT IF (Vriend, 1990), který je, spolu s několika dalšími metodami, dostupný např. na serveru `http://biotech.ebi.ac.uk:8400`. Obsahuje-li výstupní zpráva upozornění na chyby, stojí za to provést podobnou kontrolu i pro templát – kontrolní program někdy upozorňuje na chyby i v empiricky zjištěných strukturách, a nemůžeme čekat, že náš model bude lepší než templát, podle něhož byl sestrojen.[61]

Nezanedbatelným aspektem konstrukce trojrozměrných modelů je jejich **grafická prezentace** pro publikační či přednáškové účely. Rozlišení přímých obrazových výstupů programu DeepView je omezené rozlišením obrazovky, což stačí nanejvýš na prezentaci na webu či v PowerPointu, ne však pro tisk. DeepView však dovoluje uložit zvolený pohled na molekulu ve formátu „scény“ čitelné freewarovým programem pro konstrukci fotorealistických obrazů trojrozměrných útvarů **PoVRay** (*Persistence of Vision Raytracer*, `http://www.povray.org`), s jehož pomocí lze pak generovat obrázky profesionální kvality.[62] K tomuto účelu lze využít rovněž specializovaný software, například volně dostupný program MolMol.

8.4. Interpretace mnohočetného přiřazení

Nyní, když už dovedeme sestrojit mnohočetné přiřazení několika sekvencí (nebo dokonce i jim odpovídajících struktur), je na místě ohlédnout se a zamyslet se nad tím, **na jaké otázky nám konstrukce přiřazení může pomoci získat odpověď**.

Je zřejmé, že přiřazení vůbec můžeme sestrojit pouze pro sekvence, které jsou si vzájemně podobné. U sekvencí delších než pouhých několik pozic pak vzájemná podobnost nutně vypovídá o společném evolučním původu. Přiřazení samo potom může posloužit jako východisko k **rekonstrukci evoluční historie** studované rodiny genů či proteinů. Tématika evolučních interpretací přiřazení (pro běžného uživatele spojená s dnes již tradiční ikonou „evolučního stromu“ – *dendrogramu*) však je natolik obsáhlá a složitá, že jí nebudeme dále zatěžovat tuto, i tak již dosti dlouhou, kapitolu. Raději se k ní vrátíme a podrobně ji probereme v samostatné 10. kapitole.

[61] Nemohu zde nevzpomenout úsloví pocházející od babičky mého ctěného kolegy a prvního učitele ve věcech strukturního modelování Mariana Novotného: *Kanárek sovu neulehne.*

[62] Zachce-li se vám, můžete povrch vaší molekuly třeba pozlatit, pokrýt srstí nebo opatřit texturou leštěného vřesového kořene. Asi by se zde však slušelo čtenáře varovat – PoVRay je záležitost poněkud návyková, asi tak, jako bývají návykové počítačové hry (spíš „intelektuálního“ než akčního ražení).

Již zde však stojí za to předestřít, že celá řada metod používaných k rekonstrukci evolučních historií z přiřazení vychází z předpokladu, že **všechny pozice ve sledovaných sekvencích jsou rovnocenné**, to znamená, že žádná pozice nepodléhá specifickým selekčním tlakům (Brinkman a Leipe 2001). U sekvencí kódujících proteiny, funkčně zapojené do života buňky, tento předpoklad prakticky nemůže být splněn. Od této skutečnosti ovšem můžeme dočasně odhlédnout a předpokládat, že „všechny pozice v sekvenci se vyvíjejí stejně rychle“, nebo že distribuce případných selekčních tlaků v rámci sekvence je nahodilá (což je docela rozumný předpoklad tam, kde nic určitějšího nevíme). Pak ale musíme výsledky fylogenetických analýz interpretovat ve světle právě přijatého předpokladu – nesmíme např. očekávat, že délka větví dendrogramu bude přímo odrážet dobu, která uplynula od oddělení sledovaných linií.

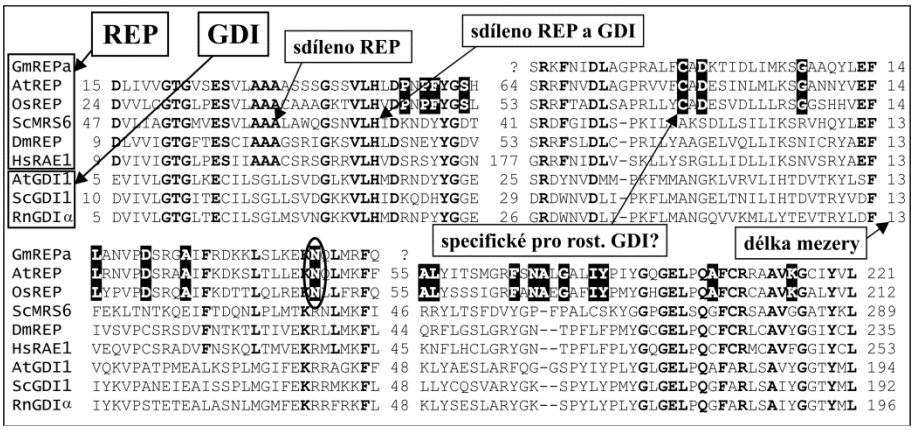
Jindy však můžeme naopak nerovnocennost jednotlivých pozic v rámci přiřazení učinit vlastním tématem studia. Z nápadné evoluční konzervace některých pozic v proteinu můžeme **usuzovat na funkční význam** příslušných aminokyselin – např. na jejich podíl na utváření aktivního místa enzymu či vazebného „rozhraní“ (*interface*) pro vzájemnou interakci proteinů. S tímto způsobem interpretace mnohočetného přiřazení jsme se ostatně již setkali v souvislosti s identifikací konzervativního fosforylačního místa specifického pro rostlinné myoziny třídy VIII, a tudíž možná funkčně zapojeného do některého buněčného procesu závislého na této „větvi“ rodiny myozinů (Obr.7.3.; Baluška et al., 2001).

Jako příklad poněkud složitější situace, k jejímuž pochopení mohla přispět analýza mnohočetného přiřazení, si můžeme uvést případ nadrodiny regulátorů malých GTPáz třídy Rab. Tato rodina proteinů společného evolučního původu zahrnuje jednak inhibitory disociace GDP (GDI – *GDP dissociation inhibitors*), jednak tzv. Rab escort proteiny (REP), které se podílejí na posttranslační prenylaci Rab GTPáz. Rostlinné homology GDI se podařilo naklonovat na základě funkce rostlinného proteinu v kvasinkových buňkách (Ueda et al., 1996; Žárský et al., 1997), kdežto v případě REP takový postup selhal. Přitom však víme, že v rostlinných genomech se vyskytují geny kódující proteiny z rodiny REP (Cvrčková et al., 2001). Možnou příčinu naznačila inspekce mnohočetného přiřazení vybraných zástupců rodin GDI a REP (Obr.8.8.). Odhalila totiž jak pozice sdílené oběma třídami proteinů, tak i pozice specifické buď pro GDI, nebo pro REP, a několik pozic společných pouze rostlinným REP.[63] Jedním z rostlinných specifik je záměna aspartátu za jeden z argininových zbytků, zcela konzervovaných u kvasinkových i živočišných REP. Hypotéza, že tento rozdíl mezi rostlinnými a nerostlinnými REP je příčinou nefunkčnosti rostlinného proteinu v kvasinkách i pozorovaných biochemických rozdílů v „chování“ REP různého původu *in vitro*, byla posléze experimentálně ověřena konstrukcí a charakterizací příslušných bodových mutantů (Hála et al., 2005).

Nemůžeme však vždy spoléhat na to, že mnohočetné přiřazení sekvencí odhalí **všechny funkčně významné rozdíly** v rámci rodiny vzájemně příbuzných proteinů. Tak v již několikrát zmiňovaném případě domény příbuzné Pten a tenzinu v rostlinných forminech třídy II jak mnohočetné přiřazení rostlinných a živočišných sekvencí, tak tvar trojrozměrného modelu rostlinné domény vypovídá o vzájemné příbuznosti mezi rostlinnými a živočišnými proteiny této strukturní třídy. Z porovnání sekvencí pak lze vyvodit, že rostlinná doména postrádá protein- a lipidfosfatázovou aktivitu, charakteristickou pro většinu živočišných

[63] Na obrázku jsou tučně jednak pozice sdílené všemi proteiny, jednak ty pozice společné všem zástupcům rodiny REP. Aminokyseliny společné pouze rostlinným REP jsou na černém pozadí.

homologů Pten.[64] Až následné modelování distribuce povrchového náboje na sestrojeném strukturním modelu však odhalilo předpokládané dramatické rozdíly v biochemických vlastnostech rostlinných a živočišných příslušníků studované genové rodiny (Cvrčková et al., 2004b).



Obr.8.8. Část mnohočetného přiřazení vybraných příslušníků nadrodiny regulátorů malých GTPáz třídy Rab. Zvýrazněna je specifická „signatura" rostlinných Rab escort proteinů. Upraveno z Cvrčková et al. (2001).

8.5. Odvozování vlastních konzervativních motivů a jejich použití v prohledávání databází

Podaří-li se nám ve zkoumané rodině proteinů nalézt **konzervativní sekvenční motivy**, charakteristické pro všechny příslušníky zkoumané rodiny proteinů, bude nás zřejmě dále zajímat, jaké další proteiny náš nově objevený motiv obsahují. Jde o úlohu komplementární vyhledávání předem charakterizovaných sekvenčních motivů z databáze (např. InterPro) v uživatelem dodané sekvenci (7. kapitola): tentokrát uživatel dodává krátký, někdy nespojitý a případně i variabilní sekvenční motiv (zapsaný zpravidla v podobě **vzorce** – *pattern* – viz 7.1.) jakožto dotaz k prohledávání databáze sekvencí.

Zvláště v případě variabilních uživatelských vzorců je takovéto prohledávání dosti náročné na výkon a čas počítače, a dostupné **veřejné servery** donedávna poskytovaly přístup k poměrně omezeným databázovým zdrojům. Jednou z prvních vyhledávacích služeb tohoto druhu se širší nabídkou databází byl server FindPattern na Švýcarském ústavu pro experimentální studium rakoviny ISREC (*Institut Suisse du Recherche Expérimentale du Cancer*), který umožňoval přístup k databázi SwissProt a využíval poměrně kuriózní, leč velmi výkonný po domácku vyrobený superpočítač (Jongeneel et al., 1997).[65] Nyní byla původní služba

[64] Zkratka Pten, zavedená v souvislosti s objevem lidského antionkogenu na chromosomu 10, který se posléze stal prototypem této genové rodiny, znamená *Phosphatase and tensin homolog on chromosome ten* (Li et al., 1997).

[65] Používaná technologie INSECT (*Inexpensive Networked Sequence Comparison Technology*) byla založena na síti 9 PC, která již ve své době (1997) odpovídala spíš průměrnému než špičkovému kancelářskému standardu (procesor Cyrix, 200 MHz). Tyto počítače byly vybaveny operačním systémem Linux a speciálním softwarem MOLLUSCS (*Modular Low-cost Linux-based Unified Sequence Comparison System*), který zajišťoval distribuci jednotlivých částí úlohy (t.j. prohledávání specifických úseků databáze) mezi jednotlivé procesory a sběr výsledků. Výkon této sestavy u řady úloh překonal i řádově dražší specializovanou víceprocesorovou pracovní stanici (Jongeneel et al. 1997).

FindPattern nahrazena širší nabídkou na serveru **MyHits** (Pagni et al., 2004) Švýcarského ústavu pro bioinformatiku (`http://myhits.isb-sib.ch`), kde lze prohledávat jak neredundantní databáze vzniklé překladem Velké trojky, tak i místní databázi trEST zkompilovanou z hypotetických proteinových překladů EST sekvencí. Server přijímá zadání (uživatelský motiv) ve formátu vzorce PROSITE a dovoluje omezit prohledávané databáze podle taxonomického zařazení zdrojového organismu.

Existují i specializované prohledávací programy zaměřené např. na konkrétní sekvenovaný genom. Tak pro *A. thaliana* je prostřednictvím serveru TAIR (`http://arabidopsis.org`) dostupný program **PatMatch** (Weng, 1998). Tento program byl úspěšně použit pro počáteční identifikaci rostlinných homologů forminů na základě přítomnosti „signatury FH2 domény“ – motivu L-x-x-G-N-x-M-N – v tehdy ještě nedosekvenovaném genomu *Arabidopsis* (Cvrčková, 2000). Mírně nepříjemnou vlastností programu PatMatch je to, že používá vlastní způsob zápisu vzorců, neodpovídající standardu PROSITE (viz Obr.7.1.).

Vzhledem k tomu, že uživatelem definované vzorce jsou obyčejně krátké, vyhledávací programy neposkytují **statistické hodnocení**, zda je výskyt hledaného motivu v databázové sekvenci nenáhodný. Ověření této skutečnosti je na uživateli, který si musí poradit, jak jen to jde. Nález motivu lze považovat za důvěryhodný, jestliže za použití nalezené sekvence jako dotazu v BLASTu identifikujeme alespoň některé ze sekvencí použitých k původnímu odvození motivu (s nízkou, tedy významnou, hodnotou E), a jestliže můžeme vytvořit dobré mnohočetné přiřazení mezi nalezenou sekvencí a sekvencemi, z nichž byl motiv odvozen (Cvrčková, 2000).

9. Analýza struktury a funkce genů

Nyní bychom se již mohli domnívat, že nám nic nebrání pokročit od konstrukce mnohočetného přiřazení proteinových sekvencí k fylogenetické analýze, jejímž výsledkem by byla interpretace přiřazení jakožto záznamu evoluční historie. Učiňme však nejprve zdánlivý krok stranou. V této kapitole se totiž zaměříme na **analýzu sekvencí DNA** (především DNA genomové) a na související metody, které vypovídají o funkčnosti jednotlivých úseků genomu.

Souvislost s kapitolou předchozí je zřejmá – budeme totiž do značné míry stavět právě na konstrukci mnohočetných přiřazení, tentokrát ve světě nukleotidových sekvencí. Zřejmou se záhy stane i souvislost s kapitolou následující, věnovanou evoluční interpretaci sekvencí. Kvalita jakýchkoli následných analýz je totiž nutně omezena **kvalitou vstupních dat**, a musíme se tedy snažit, abychom pracovali s co nejlepšími výchozími sekvencemi – tedy například abychom nezmeškali některé exony (nebo introny) sestřihovaných genů, nebo abychom nemíchali geny a pseudogeny.

9.1. Identifikace kódujících oblastí a predikce sestřihu

Prvním úkolem, kterým se zde budeme zabývat, je **predikce struktury proteinu kódovaného typickým eukaryotním genem**, jehož primární transkript podléhá sestřihu (*splicing*), tedy vlastně predikce struktury výsledné sestřižené mRNA. Budeme předpokládat, že známe sekvenci příslušného úseku chromozómu – ať již finální (ale ne nutně správně anotovanou), nebo předběžnou. V současné éře hromadného sekvenování genomů totiž značná část genomových sekvencí dostupných prostřednictvím databází (a prohledávatelných programy typu BLAST) spadá právě do kategorie sekvencí „předběžných“ – např. neanotované sekvence klonů dosud nezmapovaných na chromozómy v databázi HTGS (*high throughput genome sequencing* – viz 4.1.).

U typických genů kódujících proteiny lze při **vyhledávání hranic intronů a exonů** vycházet z toho, že hranice intronů[66] sice nejsou absolutně konzervovány, ale pro konkrétní organismus či skupinu organismů přece jen vykazují určité společné rysy.[67] Podobně lze identifikovat i oblast intronu, která odpovídá místu větvení (*branch point*) při tvorbě lariátu v průběhu sestřihu. Přítomnost všech tří sekvenčních elementů (začátek – místo větvení – konec) ve správném pořadí a v „rozumné“ vzdálenosti několika desítek až několik set bazí (nejkratší eukaryotické introny s výjimkou prvoků mají délku zhruba 30 bp) pak již umožňuje vymezit na základě genomové sekvence hranice intronu s pravděpodobností typicky převyšující 80 %. Je však nutno si uvědomit, že pravděpodobnost správné predikce pro gen jako celek klesá s rostoucím počtem intronů (Haas et al., 2002): tak 90 % pravděpodobnost správné predikce pro jediný intron znamená pouhých 60 % (0.9^5) pro gen s pěti introny. Proto je žádoucí kombinovat pokud možno výsledky několika různých predikčních metod a zejména brát v úvahu co nejvíce skutečných experimentálních dat (sekvence cDNA včetně EST i příbuzné cDNA – viz níže).

Prostřednictvím webových serverů je dostupná řada **programů pro predikci struktury mRNA**, zpravidla optimalizovaných pro vybrané modelové organismy. Následující výběr představuje

[66] 5′ konec se obvykle označuje jako „donor“ (*donor*), 3′konec jako „akceptor“ (*acceptor*) v souvislosti s mechanismem transesterifikační reakce při nejběžnějším typu sestřihu.

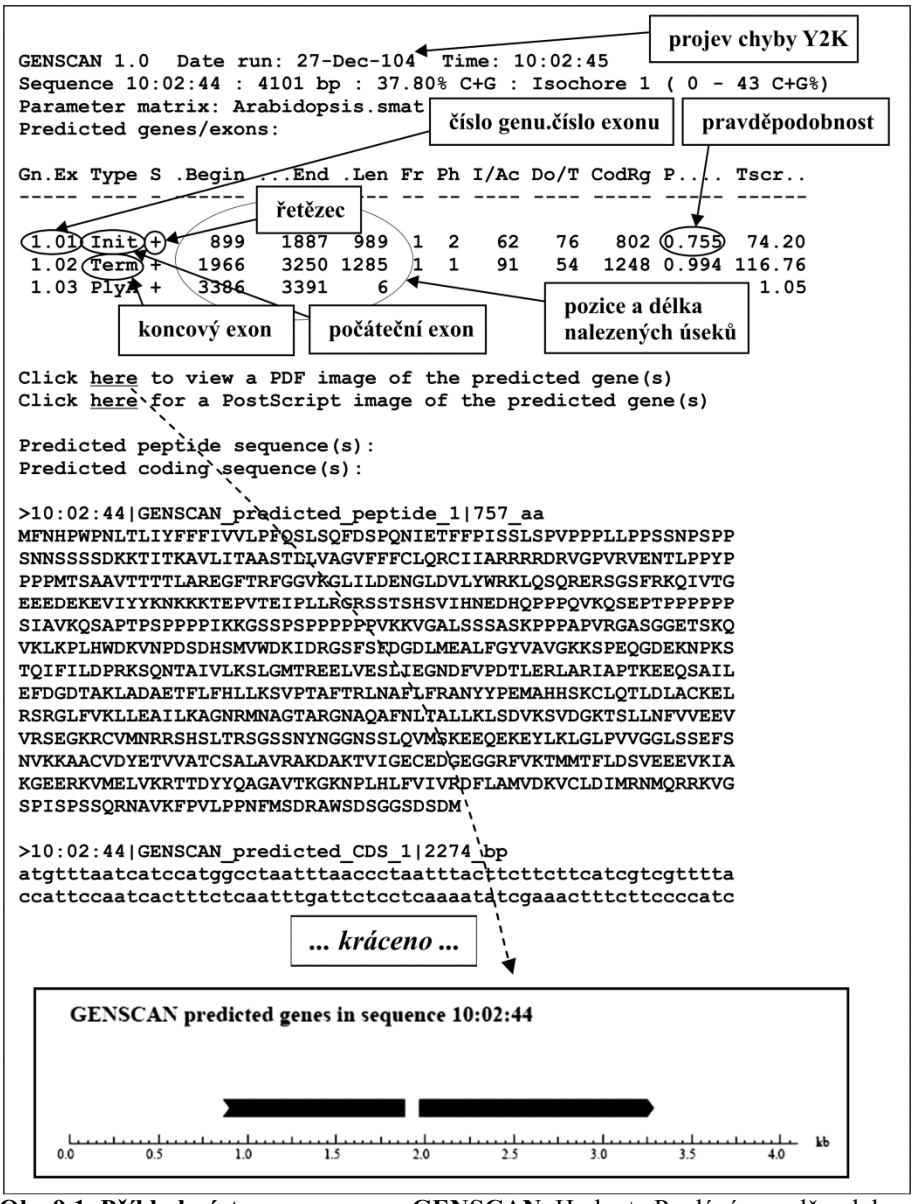
[67] Tak u *Arabidopsis* většina intronů začíná `GT...` a končí `...AG`, vzácněji mohou být introny vymezeny na 5′ konci sekvencí `AT...`, na 3′ konci sekvencí `...AC`.

základ, který stačí pro běžné použití uživateli zaměřenému především na rostlinné modely. Seřazen je podle snadnosti užití programu (což zahrnuje i rychlost odezvy serveru a jeho dostupnost) a přehlednosti výstupů. Zájemce o širší přehled a podrobnější popis odkazuji na publikovanou literaturu (Baxevanis, 2001b). Jak již bývá zvykem, na snadné a přehledné programy lze obvykle spoléhat jen v bezproblémových situacích. Ty složitější tedy jistě stojí za pokus přinejmenším tam, kde předpokládaný genový produkt nedává mnoho smyslu (např. postrádá dobře evolučně konzervovanou doménu, a přitom lze úseky odpovídající této doméně najít v domnělých intronech).

Program **GENSCAN** na serveru Massachussets Institute of Technology (`http://genes.mit.edu/GENSCAN.html`) byl původně vyvinut pro vyhledávání kódujících oblastí v obratlovčích, zejména lidských sekvencích (Burge a Karlin, 1997). Později byly doplněny i verze pro hmyz a rostliny – *Arabidopsis* a kukuřici (Burge, 1998). Zkušenosti z naší laboratoře poněkud překvapivě ukazují, že pro predikci struktury genů rýže je verze pro *Arabidopsis* zřejmě spolehlivější než ta „kukuřičná" (Cvrčková et al., 2004b). GENSCAN přijímá vstupní data v podobě čisté sekvence a podle uživatelské volby vrací buď jen predikovanou sekvenci proteinu, nebo jak sekvenci proteinu, tak i mRNA ve formátu FASTA, stručný souhrn nalezených exon-intronových rozhraní s údaji o kvalitě predikce a grafickou mapu genu (Obr.9.1.). Program určuje na základě přítomnosti charakteristických sekvenčních rysů pravděpodobnost, že daný úsek sekvence odpovídá exonu, intronu či nepřepisované oblasti. Vnitřní exony dovede predikovat spolehlivěji než exony počáteční a konečné; umí však odhalit i přítomnost většího počtu genů v předloženém chromozomálním úseku. GENSCAN neposkytuje alternativní sestřihové varianty, a vzhledem k tomu, že byl optimalizován na základě standardní sady genů o poměrně malém počtu intronů, autoři sami varují, že výsledky pro dlouhé geny s mnoha introny mohou být poněkud nejisté.

Rostlinná varianta programu **NetGene2**, NetPlantGene (Hebsgaard et al., 1996), byla vyvinuta přímo pro *Arabidopsis*. K dispozici je spolu s variantami pro jiné organismy na serveru Dánské technické univerzity (`http://www.cbs.dtu.dk`). NetPlantGene využívá dvoustupňového postupu založeného na vyhledávání konsensuálních sekvenčních motivů odpovídajících hranicím exon-intron a místům větvení s následnou filtrací výsledků. Ta zahrnuje vyhodnocení vzájemných poloh a vzdáleností nalezených motivů i nezávisle stanovené pravděpodobnosti, že predikované exony kódují protein. Vyhledávání motivů i predikce kódujících exonů využívá virtuální „neuronovou síť", „trénovanou" na standardní sadě rostlinných genů. Vstupem programu je samotná sekvence, výstup obsahuje vždy údaje pro oba řetězce DNA v podobě textové i grafické „mapy" exon-intronových rozhraní a míst větvení včetně údajů o spolehlivosti predikce (Obr.9.2.). Součástí výstupu tedy není sekvence predikované mRNA; uživatel si ji musí sestavit sám, a je na něm, jakým způsobem naloží např. s predikovanými sestřihovými variantami.[68]

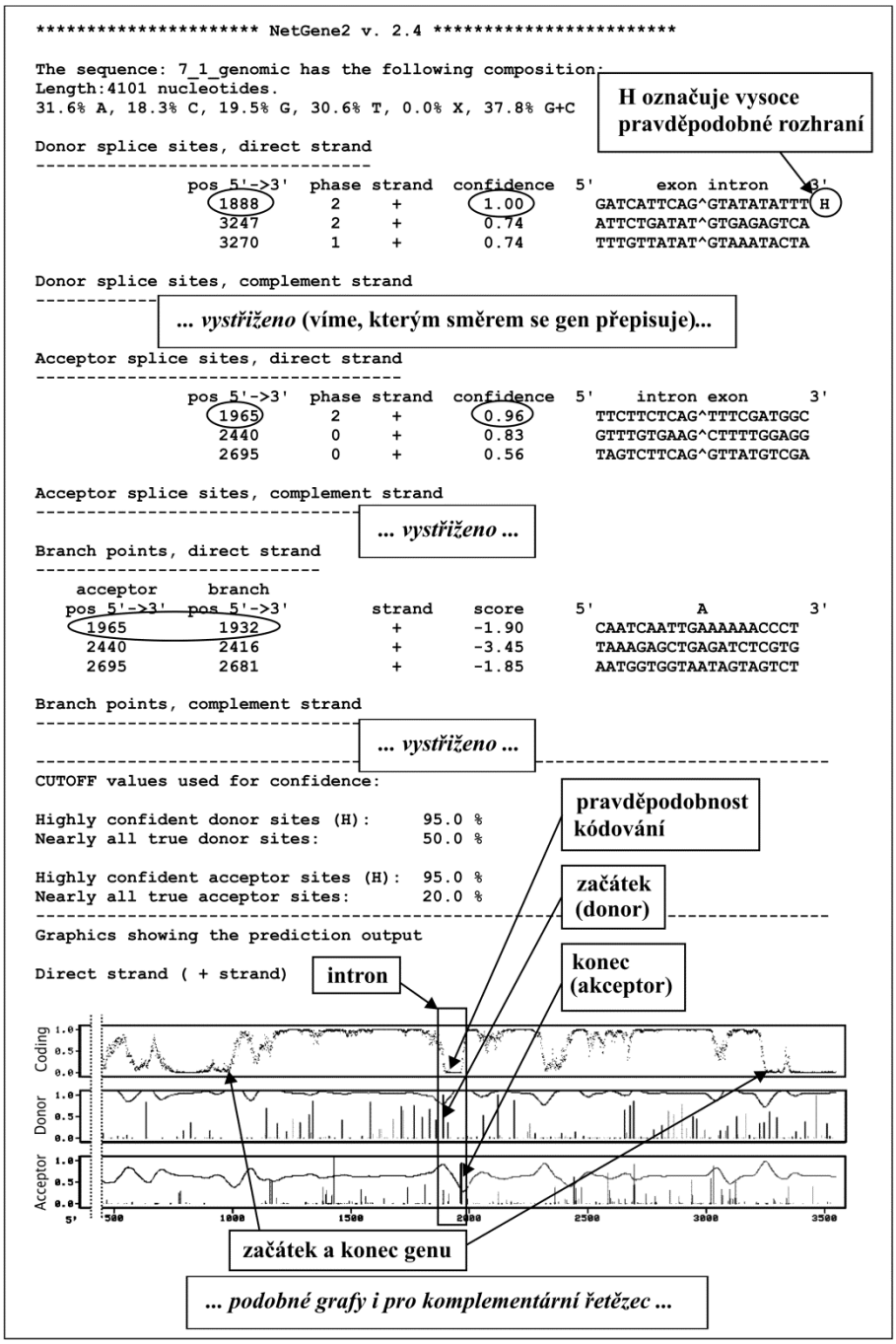
[68] Při zpracování výstupu programu NetGene je vhodné budovat predikovanou sekvenci mRNA „sestřihem" v textovém editoru (např. Notepad) a průběžně přitom ověřovat spojitost čtecího rámce translací. V obou krocích lze s výhodou využít nástrojů sady SMS (viz 3.2.) – zejména GroupDNA pro orientaci v sekvenci a Translate pro vlastní translaci (výhodou je, že Translate ignoruje čísla a mezery v sekvenci).



Obr.9.1. Příklad výstupu programu GENSCAN. Hodnota P udává pravděpodobnost správného výsledku.

Soubor programů **GeneBuilder** (Milanesi et al., 1999) je součástí kolekce bioinformatických nástrojů WebGene na italském Ústavu pro pokročilé biomedicínské technologie (*Istituto di Tecnologie Biomediche Avanzate*) v Miláně (`http://www.itba.mi.cnr.it/webgene/`). GeneBuilder slouží k predikci úplné struktury genů (včetně promotoru). Jeho součástí je i volitelný modul pro prohledávání databází EST, který umožňuje zahrnout do prováděných analýz případné experimentální podklady (částečné sekvence cDNA). Další volitelné moduly dovolují využít pro konstrukci modelu porovnání se zadaným homologním genem o známé struktuře, identifikovat promotory či polyadenylační místa, nebo dokonce označit možné chyby sekvenování či vyhledat v promotoru genu vazebná místa pro známé transkripční faktory. K dispozici jsou parametry pro několik vybraných obratlovců, *Drosophilu*, *Caenorhabditis*, *Arabidopsis* a *Aspergillus*. Výstupem je anotovaná sekvence genomové DNA ve formátu EMBL (Obr.9.3.), grafický výstup je volitelný. Predikovanou sekvenci mRNA a proteinu může uživatel získat například pomocí nástrojů z balíku SMS (viz 3.2.) EMBL Feature Extractor a EMBL Trans Extractor, případně si ji může postavit sám podobně jako v případě programu NetGene. Nevýhodou je někdy pomalá odezva, zejména ve špičce a při

využití náročnějších volitelných modulů (stojí za to využít možnosti zaslání výsledku e-mailem).



Obr.9.2. Příklad výstupu programu NetGene2 pro tutéž sekvenci jako Obr.9.1. Zvýrazněny jsou souřadnice nejpravděpodobnějšího intronu, který byl shodně predikován i programem GENSCAN (ten však uváděl polohu krajních bází okolních exonů, nikoli intronu).

Mohli bychom se domnívat, že teoretické metody pro predikci struktury sestřiženého transkriptu se stávají zbytečnými, máme-li k dispozici **experimentálně zjištěnou sekvenci cDNA** odpovídající zkoumanému genu. Mnohdy je tomu vskutku tak; empiricky stanovená sekvence cDNA je nepřekonatelným „zlatým standardem“, na jehož základě jsou vyvíjeny a ověřovány teoretické predikční postupy (Haas et al., 2002). Nesmíme však zapomínat, že ani tam, kde známe pro daný gen jen jedinou cDNA, nelze vyloučit existenci alternativních

sestřihových variant. Navíc knihovny cDNA (a z nich odvozené sekvence) mohou být kontaminovány nesestřiženými či částečně sestřiženými transkripty, fragmenty chromozomální DNA nebo dokonce hrubými klonovacími artefakty. Problematická může být konečně i kvalita samotného sekvenování cDNA (viz 9.1.1.). Pokud se nám empiricky zjištěná sekvence cDNA jeví jako podezřelá, vyplatí se tedy vrátit se k teoretickým metodám a s jejich pomocí prozkoumat možné alternativní produkty sestřihu. V případě zajímavých výsledků pak můžeme po předpokládaných transkriptech pátrat např. za pomoci RT-PCR.

Obr.9.3. Příklad výstupu programu GeneBuilder pro stejnou sekvenci jako Obr.9.1. a 9.2. V tomto případě dospěl Gene Builder k výsledkům dramaticky odlišným od předchozích dvou programů – našel totiž dva geny místo jednoho, a navíc podivný krátký peptid bez ATG.

Pro geny, jejichž úplná cDNA nebyla dosud klonována, můžeme někdy celou, jindy alespoň částečnou sekvenci rekonstruovat z krátkých fragmentů deponovaných v databázích jako tzv. **EST sekvence** (*expressed sequence tags*, viz 4.1.). Záznamy EST odpovídají zpravidla jedné sekvenovací reakci od konce klonu. Vzhledem k tomu, že sekvenované cDNA mohou být neúplné, lze v EST databázích najít i úseky odpovídající místům uvnitř studované sekvence, a při troše štěstí z nich poskládat částečnou nebo i celou sekvenci zralé mRNA. K **vyhledávání EST** lze využít např. program BLASTN (viz 6.3.) s chromozomální sekvencí jakožto dotazem. Překrývající se **fragmenty EST lze sestavit** třeba pomocí programu BioEdit (viz 8.2.2.) s využitím zásuvného modulu CAP pro vyhledávání překryvů a odvozování konsensuální sekvence zahrnující všechny překrývající se fragmenty (*contig*).

Právě uvedený postup může vést k problémům tam, kde dostupné EST odpovídají různým sestřihovým variantám (nebo nesestřiženým či částečně sestřiženým transkriptům), a už vůbec se nehodí k **mapování exonů na genom** či k vzájemnému porovnávání predikcí získaných různými metodami. Pro tyto účely lze využít Clustal W jakožto zásuvný modul BioEditu (viz 8.2.2.). Musíme ovšem specifikovat, že pracujeme se sekvencemi DNA (`/TYPE=DNA`), a je rovněž vhodné nahradit přednastavenou „cenu za prodloužení mezery" (*gap extend*) pro párové i mnohočetné přiřazení nulou. I tak se ale asi nevyhneme poměrně zdlouhavému ručnímu editování vzniklého přiřazení v BioEditu. Příjemněji lze uvedené úkoly řešit za pomoci programu MACAW (8.2.2.), i když nám pak nezbude než „dobývat" výslednou predikci ručně z exportovaného výstupu v textovém editoru (doporučuje se přitom průběžně kontrolovat předpokládané translační produkty např. pomocí nástroje SMS Translate).

MACAW je vhodný i k porovnávání sekvencí predikovaných mRNA se sekvencemi odvozenými od blízce příbuzných **homologních genů**, které mohou poskytnout dobré vodítko např. při rozhodování mezi různými sestřihovými variantami. Uživatel tak vlastně ručně a interaktivně uskutečňuje jakousi obdobu postupu, který v automatizované podobě provádí GeneBuilder – ale může přitom současně do konstrukce modelu genu vložit i obtížně algoritmizovatelné či vůbec nealgoritmizovatelné biologické a biochemické vědomosti a vhledy.

9.1.1. Jak kvalitní je anotace v databázích?

Pro současnou dobu je charakteristický dramatický nárůst objemu dat ve veřejně dostupných databázích i počtu genomových projektů v nejrůznějších stadiích vývoje – od zveřejněných neposkládaných primárních sekvenčních záznamů v databázové sekci HTGS či v specializovaných databázích mimo „Velkou trojku" až po „plně anotované" genomy dostupné prostřednictvím databáze „referenčních sekvencí" (4.2.). Je však třeba předestřít, že dodnes **žádný genomový projekt nebyl zcela dokončen** v tom smyslu, že by byla vydána nějaká definitivní a úplná anotace, na níž už se nikdy nic nebude měnit. I u „dokončených" genomů proces zpřesňování anotací průběžně pokračuje, a postupně jsou vydávány nové a nové verze.

Dobrou představu nám může poskytnout ohlédnutí za nedávným vývojem **sekvenování a anotace genomu dnes již klasického rostlinného modelu – *Arabidopsis thaliana***. První verze anotované genomové sekvence *Arabidopsis* byla zveřejněna již roku 2000 (The Arabidopsis Genome Initiative, 2000). Během uplynulých čtyř let pak byly postupně vydáno již pět „velkých" verzí (*release*) anotace a několik menších aktualizací, číslovaných podobně, jako tomu bývá u softwaru.[69] Vývoj anotace odráží jednak systematické mapování dříve neznámých funkčních elementů, jako jsou např. geny kódující malé regulační RNA (microRNA), jednak zpřesňování predikce genů kódujících proteiny. Hlavním zdrojem změn v anotaci kódujících genů je začlenění dat z přímého sekvenování kompletních cDNA a EST; pro systematické zapracování těchto údajů jsou již dokonce vyvíjeny automatizované postupy (Haas et al., 2002; Haas et al., 2003). Při zpřesňování anotace jsou brány v úvahu i sekvence homologních genů.

Podle nedávných odhadů (Haas et al., 2003) vedlo zapracování dat z cDNA a EST ke změnám anotace zhruba poloviny genů, které byly původně popsány pouze na základě „ručního" vyhodnocení teoretických predikcí sestřihu. Je však obtížné z dostupných dat zjistit, jak často se změny týkaly vlastní kódující sekvence, a kdy zapracování sekvencí cDNA vedlo „pouze"

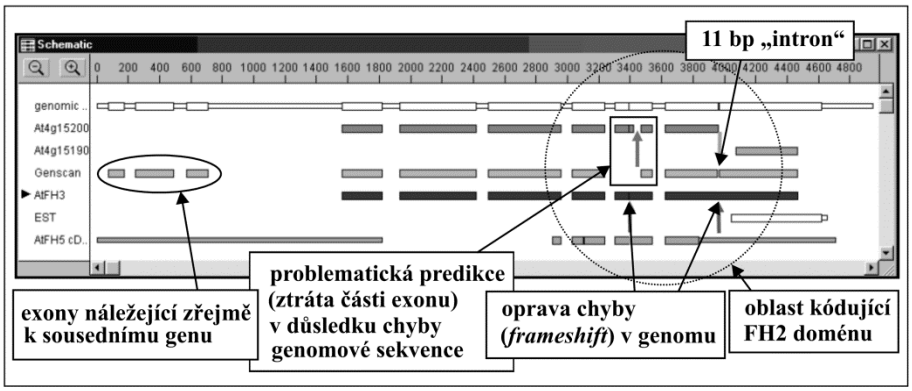
ke zpřesnění struktury nekódujících začátků a konců mRNA. Názornější představu o charakteru změn možná poskytnou konkrétní **příklady podrobněji analyzovaných genových rodin**, jejichž anotace byla postupem času revidována s využitím všech dostupných údajů, s dosti významným podílem ruční práce.

Tak v případě rodiny **rostlinných homologů forminů**, která u *Arabidopsis* zahrnuje 21 chromozomálních lokusů, spadajících do dvou tříd, bylo z 11 forminů třídy I těsně před zveřejněním první verze genomu v databázi zastoupeno 8 genů, a jen tři z nich byly anotovány (Cvrčková, 2000). O dva roky později již bylo anotováno všech 21 genů (Deeks et al., 2002). Následná podrobná analýza, při níž byly brány v úvahu všechny dostupné sekvence cDNA a EST i homologní geny (včetně genů z jiných rostlinných druhů), však vyvolala poměrně zásadní revizi predikovaných sekvencí mRNA (a tedy i kódovaných proteinů) pro dva z 11 forminů třídy I a pro 8 z 10 forminů třídy II; navíc se ukázalo, že jeden z genů asi kóduje dva proteiny, a možná jde vlastně o dva sousedící geny (Cvrčková et al., 2004b). Zjevný (ačkoli statisticky asi těžko doložitelný) rozdíl mezi třídami zřejmě souvisí s tím, že geny pro forminy třídy I mají výrazně méně intronů (1 až 5) než geny pro forminy třídy II (10 až 23), projevuje se zde tedy již zmiňovaná kumulace chyb při predikci struktury mRNA.

Povahu běžných chyb, s nimiž se můžeme v anotacích genů setkat, lze ilustrovat příkladem jednoho z problematických forminů třídy I – AtFH3 (Obr.9.4.), i když dlužno přiznat, že koncentrace chyb v tomto konkrétním genu je vyšší, než bývá zvykem. Oblast chromozómu 4, v níž tento gen sídlí, byla původně anotována jako dva lokusy – At4g15200 a At4g15190, přičemž pouze pro At4g15190 byla v databázi zachycena částečná cDNA (EST). Predikované lokusy však zjevně kódují různé části vysoce konzervativní FH2 domény. Podezření, že jde o uměle oddělené fragmenty jednoho genu, dále potvrzuje jednak skutečnost, že k oběma lokusům a úseku „mezi nimi“ lze dobře přiřadit experimentálně stanovenou sekvenci cDNA blízce příbuzného genu AtFH5, jednak teoretická predikce struktury mRNA programem GENSCAN. Ta totiž rovněž oba lokusy propojuje; předpokládá však vystřižení intronu o délce 11 bp, který je zřejmě příliš krátký na to, aby byl skutečně sestřihován.Nejkratší dokumentované introny, známé pouze u protozoí, totiž mají délku 13 bp (Deutsch a Long, 1999). Za pravděpodobnější proto lze (také vzhledem k GC-bohaté sekvenci příslušné oblasti) považovat chybu v genomové sekvenci, která vedla k narušení čtecího rámce, z něhož v příslušné oblasti není vystřihováno nic. Druhá podobná chyba (opět v GC-bohaté oblasti) byla opravena přímo na základě lokálního přiřazení mezi dobře konzervovanými úseky genů AtFH3 a AtFH5; původní anotace genomu i GENSCAN v této oblasti vynechávaly část velmi konzervativního exonu. Přes veškeré úsilí se dosud zřejmě nepodařilo najít počáteční exon genu, neznáme tedy polohu ATG a sekvenci N-konce kódovaného proteinu. Příčina může souviset s poměrně speciální regulací exprese tohoto genu, který je podle transkriptomických dat (viz 9.2.) zřejmě exprimován výhradně v pylu.

Vysoká frekvence chyb v anotaci genové rodiny kódující homology forminů by mohla souviset s tím, že experimentální charakterizace rostlinných forminů je teprve v počátcích. Publikované experimentální studie se zatím týkají pouhých tří genů z 21 (Banno a Chua, 2000; Cheung a Wu, 2004; Favery et al., 2004; Ingouff et al., 2005). Podobně je tomu však i u důkladněji charakterizovaných genových rodin. Tak v rámci kritické analýzy genové rodiny **rostlinných fosfolipáz D** (PLD) byla revidována anotace čtyř z 12 genů kódujících PLD u *Arabidopsis*, a pro další gen byl navržen alternativní vzorec sestřihu. Přitom byly v publikovaných sekvencích cDNA z šesti experimentálně charakterizovaných genů zjištěny nejen bodové mutace a jednonukleotidové delece a inzerce vedoucí k posunu čtecího rámce, ale dokonce i zřejmé klonovací artefakty složené z fragmentů cDNA odpovídajících dvěma

různým genům, v jednom případě dokonce zcela nepříbuzným a umístěným na různých chromozómech (Eliáš et al., 2002).



Obr.9.4. Schéma revize struktury lokusu At4g15200, který kóduje homolog forminu AtFH3 (viz Cvrčková et al., 2004b). Obrázek je založen na výstupu programu MACAW, výsledná predikovaná cDNA AtFH3 je označena šipkou vlevo.

Můžeme tedy shrnout, že dokonce i u poměrně dobře charakterizovaných genomů, jako je *Arabidopsis*, by **podrobná revize genové struktury** na základě všech dostupných údajů měla předcházet jakýmkoli dalším analýzám, zejména fylogenetickým. Pro genomy, jejichž sekvenování či anotace ještě nedospěly srovnatelného stadia, se tento požadavek samozřejmě stává ještě naléhavějším.

9.2. Analýza genové exprese

Analýza dostupných údajů o **časové a prostorové regulaci genové exprese** je poměrně častým úkolem bioinformatické praxe. I pokud nás zajímá spíše fylogenetická než funkční stránka zkoumaných genů, měli bychom být přinejmenším schopni odlišit exprimované geny od pseudogenů. Znalost časového a prostorového uspořádání genové exprese kromě toho může zásadně přispět k pochopení funkce studovaných genů. Umožňuje totiž například vybrat v rámci genové rodiny ty geny, které by díky podobnému „vzoru“ (*pattern*) exprese mohly vykazovat funkční redundanci, a naopak odhadnout, kde a za jakých podmínek by se mutace jednotlivého genu mohla fenotypově projevit.

O regulaci exprese konkrétního genu mohou vypovídat datové zdroje dvojího typu. Prvním jsou přímo experimentálně stanovené údaje o množství transkriptu studovaného genu v konkrétních situacích (orgán, tkáň či pletivo, kultivační podmínky ...), získané v rámci **transkriptomických studií**. Druhým typem dat pak jsou údaje o přítomnosti **vazebných míst pro známé transkripční regulátory** v sekvenci genu, zejména jeho promotoru.

Vyhledávání **vazebných míst pro transkripční regulátory** však představuje spíše doplněk k spolehlivějším metodám analýzy genové exprese. Má smysl především tam, kde již máme podezření na konkrétní regulaci a chceme např. předběžně vymezit oblast promotoru, která stojí za bližší experimentální studium. Sekvenční motivy rozpoznávané transkripčními faktory jsou totiž obvykle krátké a jejich funkčnost může být podmíněna i dalšími podmínkami, zejména konformací chromatinu. Musíme se smířit s tím, že řada „vazebných míst“ nalezených na základě prosté shody sekvence může tedy představovat falešné pozitivy; a toto riziko s nárůstem počtu známých motivů dále roste (Prestridge, 1991).

S jedním z programů schopných vyhledávat vazebná místa v promotorech jsme se již setkali – systém **GeneBuilder** (9.1.) totiž zahrnuje volitelnou možnost prohledávání databáze profilů regulačních míst algoritmem MatInspector (Quandt et al., 1995); k dispozici je nastavení pro obratlovce, hmyz, houby a rostliny. Jednoduché prohledávání databáze vazebných míst TRANSFAC (Heinemeyer et al., 1998), opět s možností výběru cílového organismu, poskytuje i program **TFSEARCH** (`http://www.cbrc.jp/research/db/TFSEARCH.html`).[70]

Obě zmíněné databáze „obecných" vazebných míst pro transkripční faktory však jsou poměrně chudé na údaje o rostlinných sekvencích. Rozsáhlejší specializovanou rostlinnou databázi **PLACE** (*Plant cis-acting regulatory DNA elements*, Higo et al., 1999) lze pomocí programu SignalScan (Prestridge, 1991) prohledávat na adrese `http://www.dna.affrc.go.jp/PLACE/signalscan.html`. Program **NsiteP** na komerčních stránkách `http://www.softberry.com` prohledává nepublikovanou firemní databázi rostlinných regulačních sekvencí RegSiteP; pro nekomerční využití je dostupný zdarma a bez registrace.

Na rozdíl od dosud probíraných zdrojů, které zpravidla sdružovaly data pro širší spektrum organismů, veřejně dostupné **databáze výsledků transkriptomických studií** jsou specifické pro jeden či úzkou skupinu biologických modelů a nejsou centralizovány. Nebudu se zde pokoušet o vyčerpávající přehled celé taxonomické šíře; uvedu pouze několik „startovních bodů" pro **modelový případ** konkrétní skupiny organismů – totiž vyšší rostliny, zejména *Arabidopsis*. Zájemce o jiné organismy odkazuji zejména na „rozcestník" veřejně dostupných transkriptomických databází na `http://genome-www.stanford.edu` (*Stanford Microarray Database*) a na veřejné depozitáře transkriptomických dat – databázi Arrayexpress na serveru EBI (`http://www.ebi.ac.uk/Databases/microarray.html`) a Gene Expression Omnibus na NCBI (`http://ncbi.nlm.nih.gov/GEO/`).

Nebudu se také zabývat celou šíří problematiky transkriptomických dat. Analýza genové exprese v měřítku pokrývajícím celý genom, založená na kvantitativním měření hybridizace značené celkové cDNA či cRNA (komplementární RNA) ke stovkám až tisícům genově specifických DNA či oligonukleotidových sond na **mikroarrayích**, je dnes již samostatným podoborem molekulární biologie i bioinformatiky ve smyslu širší definice oboru (viz 1. kapitola), a je tématem řady monografií a nejméně jedné velmi dobré učebnice (Stekel, 2003). Získávání primárních transkriptomických dat a jejich analýza však již jednoznačně spadá do oblasti „vysoké bioinformatiky" a překračuje rámec tématiky tohoto textu. Na našem modelovém rostlinném systému si proto pouze ukážeme jeden z možných způsobů, jak z veřejně dostupných transkriptomických dat poměrně jednoduše získat informace o expresi vybraných konkrétního genu.

Pro *Arabidopsis* existuje několik veřejně dostupných databází výsledků transkriptomických studií; všechny jsou přístupné prostřednictvím stránek TAIR (`http://arabidopsis.org`). Zvláštní pozornost si mezi nimi zasluhuje kolekce výsledků standardizovaných měření genové exprese na mikroarrayích (čipech) systému Affymetrix **NASCArrays** (*Nottingham Arabidopsis Stock Centre Arrays,* Craigon et al., 2004), přímo dostupná na adrese `http://affymetrix.arabidopsis.info`. Procedura přípravy a hodnocení mikroarrayů zavedená firmou Affymetrix je totiž výjimečně dobře standardizována a dovoluje i přímé porovnávání experimentů z různých laboratoří (včetně sdílení kontrol), což o jiných systémech rozhodně nelze říci. Kromě prohlížení

předzpracovaných a anotovaných primárních dat a jejich stahování pro další analýzy poskytuje databáze NASCArrays několik poměrně zajímavých **nástrojů pro vyhledávání genově specifických dat**:

- Nástroj **„Spot History“** prezentuje distribuci relativních hodnot exprese zvoleného genu ve formě histogramu, na němž lze snadno identifikovat datové sady (sklíčka), z nichž pocházejí extrémní hodnoty. Uživatel tedy např. ihned pozná, kde a kdy je exprese studovaného genu mimořádně zvýšená či snížená.
- **„Gene Swinger“** přímo vybírá experimenty (tedy dvojice sklíček „kontrola – pokus“), v nichž je uživatelem vybraný gen nejvýrazněji regulován.
- Volba **„Two gene scatter plot“** vytváří bodový diagram, na němž jsou na vzájemně kolmých osách vyneseny hodnoty exprese ze všech datových sad pro dva uživatelem vybrané geny. Z grafu lze posoudit, zda geny mohou být koregulovány, a případně vybrat sklíčka, na nichž se chovají neobvykle.
- Pomocí **„Bulk Gene Download“** lze pro zadaný gen stáhnout a uložit naměřené hodnoty hladin exprese ze všech experimentů pro další zpracování.

Velmi zajímavou nabídku nástrojů pro analýzu širšího souboru mikroarrayů systému Affymetrix, zahrnující mimo jiné také úplnou kolekci NASCArrays, nabízí v poslední době také server **Genevestigator** (`https://www.genevestigator.ethz.ch`), který však je zdarma pouze pro nekomerční akademické využití a vyžaduje registraci. Zvláštní pozornost si zde zasluhuje zejména možnost konstrukce „virtuálního Northern blotu“ a funkce Meta-Analyzer, která slouží k hledání společně regulovaných genů v rámci uživatelsky zadaného výběru (např. genové rodiny).

Vzhledem k tomu, že oligonukleotidové sondy čipů Affymetrix pokrývají téměř všechny predikované geny *Arabidopsis*, máme poměrně velkou naději, že údaje pro gen, který nás zajímá, najdeme. Někdy tak můžeme dokázat, že náš gen vykazuje aspoň v nějaké situaci nenulovou expresi, i když pro něj dosud nebyla klonována ani částečná cDNA, a ujistit se tak, že pracujeme se skutečným genem a ne s pseudogenem.

10. Studium příbuzenských vztahů biologických sekvencí

Sekvence biologických makromolekul poskytují bohatý zdroj informací o příbuzenských vztazích mezi organismy i jejich geny. V této kapitole se budeme věnovat alespoň počátečním krokům cesty, která vede od souboru sekvencí k výpovědi o jejich evoluční historii, obvykle ztělesněné **„genealogickým stromem“** čili *dendrogramem* v duchu nejlepší haeckelovské tradice. Fylogenetická analýza (nejen) sekvencí je téma obsáhlé, daleko překračující prostor, který jí zde můžeme věnovat; nezbývá, než zůstat vskutku u základů.[71] Ty by měly pro začátek stačit uživateli, který není specializovaným evolučním biologem, ale potřebuje „pouze“ sestrojit dendrogram studovaných sekvencí, vyvodit z něj biologicky smysluplné závěry a obstát v seriózně vedeném recenzním řízení.[72]

Připomeňme si nejprve některé obecnější souvislosti.

- Sekvence makromolekul zdaleka nejsou jediným zdrojem informací o vzájemné příbuznosti organismů či biologických pochodů. Toto konstatování se může zdát triviálním – přece víme, že Darwin neměl ani tušení o existenci DNA. I dnes však lze dokumentovat nejen „klasické“ taxonomické studie, ale i práce týkající se **evoluční konzervace buněčných dějů** na molekulární úrovni, jinými než sekvenčními údaji.

To je případ třeba řady dílčích farmakologických studií,[73] ale i prací tak zásadních, jako byl „nobelovský“ biochemicko-imunologický důkaz společné podstaty centrálních regulátorů buněčného cyklu (Labbe et al., 1988; Gautier et al., 1988).

- U současných sekvencí nemůžeme přímo pozorovat „příbuzenské vztahy“, ale jen **vztahy podobnosti** (viz 5. kapitola; Petsko, 2001; Cvrčková a Markoš, 2005). Podobnost struktury však sama o sobě není důkazem příbuznosti genealogické. Genealogická příbuznost je „pouze“ často nejjednodušším možným, a v případě molekulárních dat až na výjimky (krátké motivy vzniklé konvergencí) zřejmě jediným správným, vysvětlením pozorované příbuznosti strukturní. Některé metody používané k rekonstrukci genealogie molekulárních sekvencí společný původ zkoumaných sekvencí přímo předpokládají (např. „*maximum parsimony*“ – viz 10.2.), jiné (distanční metody) se mohou obejít i bez předpokladu genealogické souvislosti (aspoň pokud bychom zvolili „genealogicky nezatíženou“ substituční matici, např. matici identity – viz 5.3.), ba i bez předpokladu existence evolučního procesu. I u nich však interpretace výsledků může záviset na přijetí dalších dosti specifických předpokladů, týkajících se např. druhu a tempa změn, k nimž v průběhu evoluce docházelo, které přijetí předpokladu genealogické souvislosti vyžadují.
- Předpoklad, že studované vzájemně příbuzné sekvence vznikly evolučním procesem ze společného předka, ovšem neznamená, že tato evoluce musela probíhat darwinovsky – tj. působením selekčního procesu na nahodilé mutace. Ostatně mezi „naivního školního darwinismu“ jsme si vědomi do značné míry právě díky genomovým projektům a následným bioinformatickým i experimentálním studiím. Naopak, řada používaných metod obsahuje předpoklad **neutrální evoluce** v nepřítomnosti selekčního tlaku (proto jsou vděčným objektem zájmu molekulárních taxonomů nekódující sekvence, např. „mezerníky“ lokusů kódujících ribozomální RNA). Data, která tento předpoklad nesplňují, sice můžeme analyzovat, ne všechny výstupy analýzy však poskytují biologicky smysluplnou výpověď.
- Předpoklad **společného předka** však je nutno brát *velmi* vážně v tom smyslu, že nestačí deklarovat, že koneckonců všechny nás spojuje pramáti Lucinka (LUCA – *Last universal common ancestor*). Aby výsledky dávaly smysl, musí být celý soubor zkoumaných sekvencí *monofyletický* (tedy propojený poměrně nedávným společným předkem), nesmí tedy např. obsahovat subgenové domény, které přivandrovaly přesunem exonů či horizontálním přenosem genetické informace jen do některých sekvencí studované rodiny.[74]

Můžeme tedy předběžně shrnout, že v molekulární fylogenetice platí snad více než kde jinde výstižný výrok Andrey Hansen, že „*programy [...]* ***vždy*** *dávají výsledky, ale je na uživateli, aby rozhodl, zda výsledek dává také smysl*“.[75]

10.1. Základní pojmy

Metody molekulární fylogenetiky zde budeme chápat poněkud zúženě jako **postupy, které využívají sekvenčních dat k rekonstrukci genealogických vztahů** mezi organismy, případně jejich geny. Odhlédneme tedy pro účely tohoto výkladu od metod vycházejících z jiných než sekvenčních dat (např. hodnocení přítomnosti restrikčních míst, specifických fragmentů amplifikovaných PCR či jiných znaků); zbavíme se tím rovněž možných problémů pramenících z toho, že u nesekvenčních dat může být obtížné poznat skutečnou **homologii**, která je výsledkem společného evolučního původu, od **analogie** vzniklé konvergencí. Pro účely molekulární systematiky organismů je žádoucí, aby studované sekvence byly

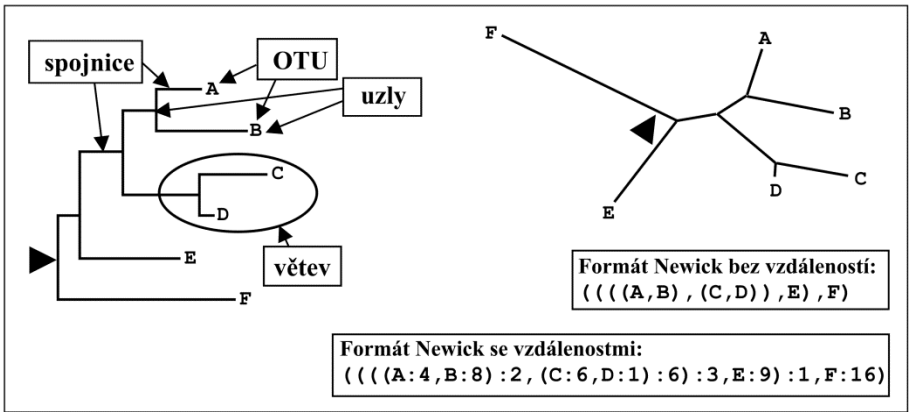
homologní specifickým způsobem – totiž **ortologní**. Ortologní geny existují v genomech v jediné kopii, která u všech zkoumaných organismů vykonává tutéž funkci. Řada evolučně konzervovaných genů však není ortologní, nýbrž **paralogní**: proběhly v průběhu evoluce jednou či několika duplikacemi následovanými rozrůzněním struktury i funkce, a někdy i ztrátou některých kopií (viz např. Sonnhammer a Koonin, 2002). Přinejmenším pro některé skupiny organismů (namátkou třeba obratlovce a vyšší rostliny) je určení ortologů a paralogů z principu problematické, protože celé jejich genomy prošly sérií genových duplikací a ztrát jednotlivých kopií genů.

Genová duplikace je dnes chápána především jako zdroj nových genů s novými funkcemi. Ne vždy však je duplikace následována zásadní proměnou funkce genu. Duplikované geny si často zachovávají jistou míru funkčního překryvu, který se za určitých podmínek může projevovat jako **„redundance“**, ačkoli v některých biologicky významných situacích jemné rozdíly mezi paralogními geny přispívají k optimalizaci funkce organismu (Nasmyth et al., 1991; Brookfield, 1997). Naopak změna funkce nemusí být vždy provázena dramatickou přestavbou struktury genu či proteinu. L. Hartwell se spolupracovníky (1999) formulovali hypotézu **„modulární biologie“**, podle níž buňky lze chápat jako soustavu multiproteinových modulů, které často využívají evolučně konzervovaných molekul. Dobrým příkladem modulu je třeba signální dráha spojená s malými GTPázami, využívaná v řadě buněčných kontextů od exocytózy až po organizaci aktinového cytoskeletu. Můžeme snad i zobecnit, že k vývoji nových funkcí zřejmě přispívají zejména změny vstupních a výstupních prvků modulů, které zprostředkovávají jeho propojení s jinými moduly i vnějším prostředím (Csete a Doyle, 2002; na konkrétním příkladu malých GTPáz ukázáno též v Cvrčková et al. 2004a).

Ve standardním kontextu systematické biologie, popřípadě taxonomie, jsou předmětem studia zpravidla biologické druhy či jinak specifikované populace organismů – „**operační taxonomické jednotky**“ (**OTU**, *Operational Taxonomic Units*), reprezentované sekvencemi vybraných vzájemně ortologních genů z vybraných jedinců. Metody molekulární fylogenetiky však lze použít i k analýze paralogních genových rodin, zejména k predikci možných vztahů redundance či naopak funkčních rozdílů mezi jejich členy (Brinkman a Leipe 2001; viz též příklady v kap. 10.5), i když k tomuto účelu původně nebyly míněny. V takových příkladech pak v roli OTU vystupují geny samotné, bez ohledu na jejich organizmální původ - věnujeme se vlastně „systematice genů“.

Cílem našeho snažení bude konstrukce „genealogického stromu“ studovaných sekvencí čili dendrogramu. **Dendrogram** (Obr.10.1) je graf, který spojuje OTU prostřednictvím **větví** (*branches*) s **uzly** (*nodes*), které reprezentují hypotetické společné předky propojených OTU a jsou navzájem dalšími spojnicemi propojeny s předchozí „generací“ společných předků, reprezentovaných uzly hlouběji uvnitř stromu. Předpokládáme, že geny se zmnožují duplikací a „konvenční“ taxony odštěpováním populací; větvení dendrogramu je tedy **binární**. Sice se někdy můžeme setkat s dendrogramem, v němž se z jednoho bodu odštěpuje několik větví (multifurkace), bývá to však obvykle grafické vyjádření toho, že dostupná data a používané metody nedovolí rozlišit pořadí odvětvování, o němž i nadále předpokládáme, že bylo binární. Všechny spojnice vycházející z uzlu a všechny uzly na nich ležící (včetně koncových OTU, které lze považovat za zvláštní druh uzlů) tvoří „větev vyššího řádu“ čili **klad** (*clade*). Reprezentují-li OTU konvenční taxonomické jednotky a nikoli geny, klad odpovídá monofyletickému taxonu. **Délka větví** odráží „evoluční vzdálenost“ uzlů, kterou si můžeme představit jako počet mutací, který odděluje uzly propojené danou spojnicí. Z praktických důvodů se dendrogramy často uvádějí v pravoúhlé grafické reprezentaci (Obr.10.1. vlevo). Přehlednější, avšak na výpočetní a grafické provedení náročná je „hvězdicovitá“ forma

dendrogramu (Obr. 10.1. vpravo). Má-li délka větve odrážet **čas** uplynulý od oddělení linií vycházejících z uzlu, musí být splněny další předpoklady – např. stálá a pro všechny sekvence stejná frekvence mutací, což je zejména u sekvencí kódujících proteiny a podléhajících selekčnímu tlaku málokdy splněno. Proto se někdy v dendrogramech délka větví neuvádí a zajímá nás pak pouze **topologie** „stromu“ (viz Obr.10.2.). Takový strom se někdy označuje jako kladogram, i když tentýž termín se v systematické biologii používá v poněkud odlišném významu.



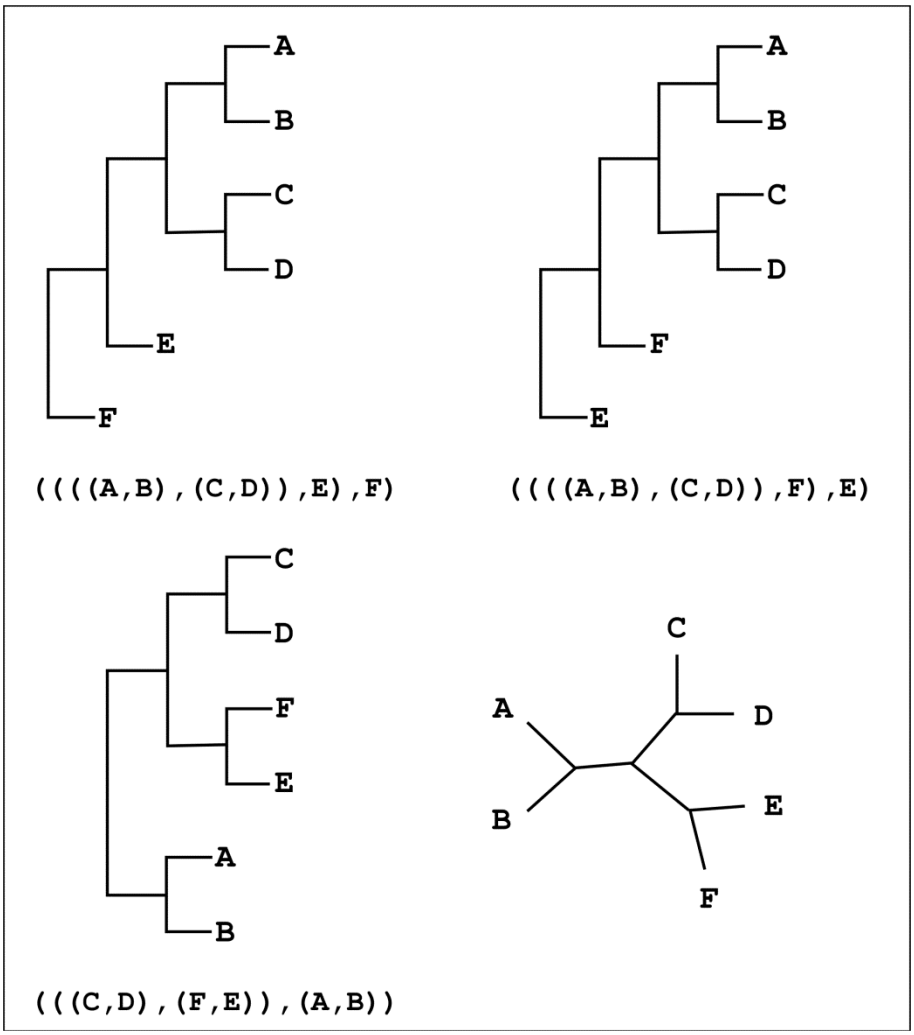
Obr.10.1. Dendrogram a jeho zápis v newickovském formátu. U dendrogramu vlevo k délce větví přispívají pouze vodorovné úsečky, nikoli svislé spojovací čáry. Tučná šipka odpovídá poloze kořene stromu v případě, že OTU F je outgroup.

Umístění společného hypotetického předka všech OTU čili **kořene stromu** (*root*) představuje závažný problém. Nejjednodušším řešením je začlenit do analýzy tzv. ***outgroup*** (dle J. Flegra „vnějšák“) – sekvenci, která je sice homologní, avšak vzdálená ostatním studovaným vzorkům. Může to být např. ortolog z fylogeneticky vzdáleného organismu (zkoumáme-li genovou rodinu z rostlin, můžeme použít jako outgroup homologní živočišný gen), nebo i vhodně zvolená sekvence paralogní (zabýváme-li se malými GTPázami rodiny Rab, použijeme k zakořenění stromu jinou malou GTPázu, třeba Rho nebo Ras). Kořen **zakořeněného** (*rooted*) stromu se nachází na větvi, která spojuje outgroup a zbytek stromu. Není-li k dispozici vhodná sekvence, nezbývá než sestrojit, případně interpretovat strom jakožto **nezakořeněný** (*unrooted*). Při stálé rychlosti evoluce všech linií bychom mohli kořen nezakořeněného stromu umístit do bodu stejně vzdáleného od všech koncových OTU, prakticky to však může být dost problematické.

Grafická forma dendrogramu (ba ani ta zjednodušená pravoúhlá) se příliš nehodí k tomu, aby se na ní či z ní cokoli počítalo; podstatně snazší je pracovat s daty zapsanými pomocí standardní sady znaků. Všechny běžné fylogenetické programy dovedou zacházet s daty v tzv. **newickovském formátu** (*Newick format*), který zachycuje vzájemné propojení uzlů pomocí závorek a umožňuje i zápis délky větví (viz též Obr.10.1.).[76] Dendrogram v grafické podobě i v newickovském formátu může vypadat jako zakořeněný i pro nezakořeněné stromy – uživatel musí vědět, že v některých situacích má kořen ignorovat.

Je snadné nahlédnout, že libovolná větev vyššího řádu může rotovat kolem kterékoli větve vnitřní. Proto je jakýkoli dendrogram možno správně nakreslit (a dokonce i zapsat v newickovském formátu) více než jedním způsobem. Zejména u nezakořeněných stromů se varianty mohou na první pohled i značně lišit (Obr.10.2.). Rozpoznání identity **různých forem téhož stromu** vyžaduje jistou dávku cviku, proto bychom neměli být hned příliš znepokojeni, když nám různé metody konstrukce dendrogramu (nebo dokonce opakování

téhož postupu) poskytnou „různé“ výsledky. Při vizuálním porovnávání stromů mohou pomoci programy, které dovedou graficky „přeskládat“ dendrogram ve formátu Newick; patří mezi ně např. níže probíraný NJ Plot (viz 10.4.2.).



Obr.10.2. Různé zápisy téhož (nezakořeněného) stromu. Jen stromy v dolní řadě jsou nezakořeněné i formálně.

10.2. Metody konstrukce dendrogramů

Metody konstrukce dendrogramů byly vyvinuty pro práci s OTU charakterizovanými souborem znaků. **Znaky** mohou mít nejrůznější povahu – od přetrhovaně lichozpeřených listů přes výskyt specifického vzorce restrikčních míst až po přítomnost serinu na pozici 129 konkrétního genu. S jednotlivými pozicemi v sekvenci tedy můžeme zacházet jako se samostatnými digitálními znaky, které nabývají několika jasně vymezených stavů. Toho využívají **znakové metody** (*character state methods*) konstrukce dendrogramů. Setkali jsme se však již i s budováním dendrogramu na zcela jiném principu: připomeňme si konstrukci mnohočetného přiřazení metodou CLUSTAL (8.2.1.), jejíž součástí byl výpočet „vůdčího stromu“ na základě matice vzdáleností mezi přiřazovanými sekvencemi. Dendrogram tedy můžeme postavit i na základě hodnot celkové vzájemné (ne)podobnosti (vzdálenosti) mezi zkoumanými sekvencemi; v takovém případě hovoříme o **metodách vzdálenostních** (*distance matrix methods*).

Ať již používáme metody znakové nebo vzdálenostní, vlastní postup vedoucí k výslednému dendrogramu může být založen na jednom ze dvou principů. Buď se snažíme pomocí nějaké algoritmické metody **sestrojit** jediný „nejlepšího možný“ strom, nebo **vyhledáváme** nejlepší z množiny všech možných dendrogramů, které by se na základě vstupních dat daly postavit.[77] Vyhledávací postupy mají oproti konstrukčním tu výhodu, že jejich výstupem může být ne jeden dendrogram, ale celé pole „kandidátů“; v případě problematických dat to dovoluje např. vytvořit konsensus z několika nejlepších stromů. Naopak nevýhodou vyhledávacích metod je jejich veliká výpočetní náročnost, protože **počet možných dendrogramů**, a tedy i velikost prohledávaného „prostoru“, s rostoucím počtem OTU velmi dramaticky narůstá (Tab.10.1.). Pro ***n*** sekvencí můžeme sestrojit

$$N = [(2n - 3)!]/[2^{n-1}(n-1)!]$$

různých zakořeněných stromů; vzhledem k tomu, že na nezakořeněný strom můžeme hledět jako na strom, který má o jednu větev (kořen) méně, odpovídá počet možných nezakořeněných stromů počtu zakořeněných stromů pro $n-1$ sekvencí (Felsenstein 2004). Prakticky se proto často, podobně jako v případě prohledávání velkých databází, používají **heuristické metody**, které namísto celého prostoru možností prohledávají jen jeho „nejnadějnější“ část, i za cenu nepříliš velkého rizika chyby.

Tab.10.1. Počet možných dendrogramů pro daný počet sekvencí.

Počet OTU	Počet zakořeněných stromů	Počet nezakořeněných stromů
1	1	1
2	1	1
3	3	1
4	15	3
5	105	15
6	945	105
7	10 395	945
8	135 135	10 395
9	2 027 025	135 135
10	34 459 425	2 027 025
11	654 729 075	34 459 425
12	13 749 310 575	654 729 075
...	...	...
20	8×10^{21}	2×10^{20}

Soustřeďme se nyní na několik vybraných metod konstrukce či optimalizace dendrogramů, které jsou přístupné i nespecializovanému a matematicky nepříliš vzdělanému uživateli (tím odpadá např. celá skupina metod založených na Bayesovských algoritmech a Markovovských řetězcích). Výběr byl řízen také volnou dostupností softwaru pro jejich praktické použití.

Nejjednodušší metodu **konstrukce dendrogramu z matice vzdáleností**, odpovídající algoritmu **UPGMA (*Unweighted Pair Group Method with Arithmetic mean*)** si můžeme představit zhruba takto:

- Vypočteme vzdálenosti mezi všemi možnými dvojicemi sekvencí a sestrojíme **matici vzdáleností**. Hodnotu vzdálenosti můžeme získat např. tak, že zjistíme normalizovanou hodnotu podobnosti (kap. 5) a odečteme ji od jedničky.

- V matici vzdáleností určíme nejnižší hodnotu a tedy **pár sekvencí, které jsou si nejbližší**. Tyto sekvence spojíme prostřednictvím dvou větví a uzlu – hypotetického společného předka. V matici nahradíme dvojici řádků a sloupců odpovídající propojenému páru jedním řádkem a sloupcem pro právě vytvořený uzel. Hodnoty pro uzel vypočteme jako aritmetický průměr hodnot pro výchozí dvojici OTU.
- Takto jsme získali **matici, která obsahuje o jednu OTU méně** než matice výchozí. Nyní celý postup opakujeme, dokud nepropojíme všechny OTU.

Metoda UPGMA je dost názorná na to, aby zabydlela naši pomyslnou světle šedou skříňku. Je to ale metoda založená čistě na analýze podobnosti fenotypů – vůbec nevyžaduje předpoklad evoluce, a tudíž nebere v úvahu ani takové faktory, jako je rozdílná rychlost evoluce (rozdílné množství mutací) v jednotlivých větvích. Pro praktickou analýzu sekvencí se proto příliš nehodí, protože právě při nestejné evoluční rychlosti jednotlivých větví generuje artefakty. Tomuto problému se vyhýbá metoda **minimální evoluce (ME)**, která se snaží minimalizovat součet délky větví; je však poměrně výpočetně náročná. Její aproximací však je asi nejčastěji používaná vzdálenostní metoda konstrukce dendrogramu, **NJ (*neighbor-joining***; Saitou a Nei, 1987; OOta a Saitou, 1999). Algoritmus NJ vychází z konstrukce hvězdicovitého „nerozlišeného" stromu, v němž z jediného centrálního uzlu (multifurkace) vycházejí větve ke všem koncovým OTU. Postup je, podobně jako v případě UPGMA, iterativní – algoritmus postupně spojuje dvojice uzlů tak, aby vždy co možná minimalizoval součet délek větví. Spojená dvojice je v každém kroku nahrazena hypotetickým předkem, což sníží počet větví („paprsků hvězdice") o jednu, a tento postup se opakuje, dokud nezbude dendrogram, který obsahuje pouze bifurkace.

Strom založený pouze na porovnání fenotypů označujeme jako **fenogram**, strom, který se snaží být i rekonstrukcí evoluční historie, jako **fylogram**. Metoda UPGMA tedy produkuje fenogram, ME a NJ fylogram. Rovněž následující skupina metod je určena ke konstrukci fylogramů.

Znakové metody zacházejí se sekvencí jako se souborem nezávislých znaků, které odpovídají jednotlivým bázím nebo aminokyselinám. Metody ze skupiny ***maximum parsimony* (MP)** hledají dendrogram odpovídající minimálnímu úhrnnému počtu mutací nezbytných k dosažení pozorovaného stavu.[78] Výsledný strom vzniká jako konsensus stromů pro jednotlivé pozice; „nejlepších" stromů může být i víc. V úvahu jsou přitom brány pouze „fylogeneticky informativní pozice", to jest ty pozice, které umožňují konstrukci jednoznačných stromů; program tedy využívá jen zlomku dostupných dat. Metody MP se používají poměrně málo, protože jsou dost náchylné k artefaktům, jestliže od evolučního oddělení zkoumaných sekvencí uplynula dlouhá doba, a tudíž se nahromadilo mnoho mutací (artefakt vzájemného přitahování dlouhých větví – *long branch attraction*). Citlivé jsou též na rozdílnou rychlost evoluce v jednotlivých liniích (Brinkman a Leipe 2001, Felsenstein 2004).

Z hlediska spolehlivosti výsledku asi nejlepší z „klasických" metod je hledání „nejvěrohodnějších stromů" – metoda ***maximum likelihood* (ML**; Felsenstein, 1981). Metoda ML prohledává všechny možné dendrogramy (případně při použití heuristických postupů jen jejich část), a pro evoluční scénář odpovídající každému z těchto stromů stanovuje pravděpodobnost, s jakou mohl generovat soubory znaků, odpovídající vloženým datům. Bere přitom, na rozdíl od metody MP, v úvahu všechny pozice. Výsledek si můžeme představit zhruba jako MP strom, v němž je počet mutací na jednotlivých větvích přizpůsoben délce větví. Podstatnou nevýhodou této metody je její velká výpočetní náročnost – i při použití dobrých heuristických aproximací se výpočet pro soubory čítající několik desítek sekvencí

(včetně náležitého vyhodnocení třeba metodou bootstrapping – viz 10.3.) stává řadovému uživateli nedostupným.

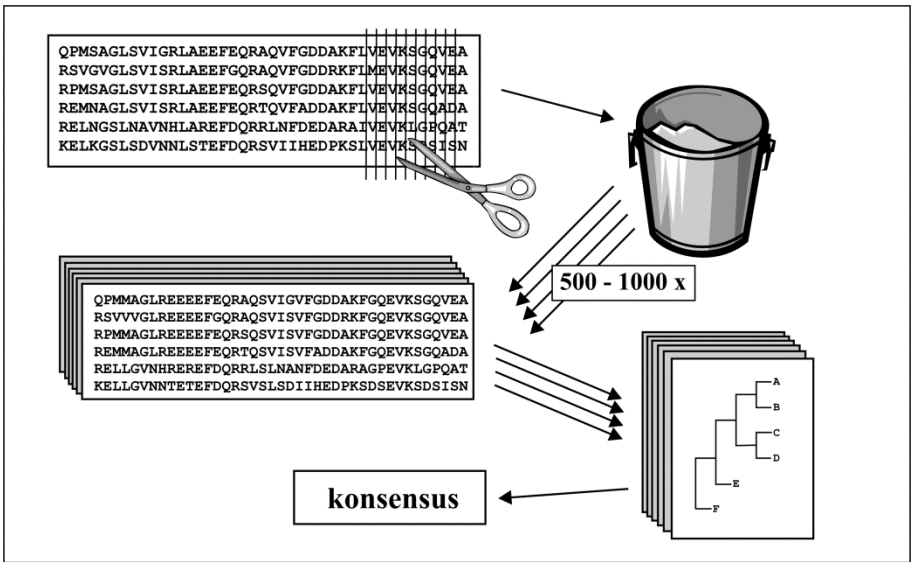
I v případě konstrukce dendrogramů platí, že dva je víc než jeden, a že pokud to lze, měli bychom studovaná data zpracovat **více metodami**. Ne všem výsledkům však můžeme přisoudit rovnocennou váhu, a rozhodnutí, zda dané metodě budeme věřit, může do jisté míry záviset i na tom, jak vypadá výsledný dendrogram. Neznamená to, že vyhodíme výsledky, které se nám nelíbí: pouze např. nebudeme věřit výsledkům MP, pokud dendrogram vypadá jako kytice dlouhých větví spojených blízko u kořene, protože víme, že právě na takových datech uvedená metoda trpí artefakty z přitahování dlouhých větví. Prakticky vždy stojí za to sestrojit dendrogram metodou NJ, a pokud velikost úlohy dovolí, použít také metodu ML.

10.3. Hodnocení kvality dendrogramu

Ať už jsme dendrogram získali jakkoli, měli bychom se vždy pokusit nějak vyhodnotit míru jeho důvěryhodnosti, než jej budeme interpretovat nebo dokonce prezentovat. Asi nejčastěji se k tomu používají **vzorkovací metody**, které testují odolnost (robustnost) jednotlivých větví dendrogramu vůči změně („poškození“) vstupních dat. Nejběžněji používaný postup, ***bootstrapping***, si můžeme ve zjednodušené podobě představit takto (Obr.10.3.):

- Výchozí mnohočetné přiřazení nastříháme na svislé proužky o šířce jedné pozice, proužky nasypeme do nádoby a dobře zamícháme.
- Z nádoby vylovíme jeden proužek, opíšeme jej, vrátíme zpět a opět zamícháme. Takto vylosujeme a opíšeme tolik proužků, kolik mělo výchozí přiřazení pozic. Získáme tak „mnohočetné přiřazení“ (**vzorek**), v němž jsou pozice z výchozího přiřazení promíchány a navíc některé jsou zastoupeny vícekrát a jiné chybí. Zůstává však zachována identita OTU – pozice, které byly nad sebou, zůstávají nad sebou a pořadí řádků se nemění.[79]
- Právě popsaný postup opakujeme, dokud nevyrobíme dostatečný počet vzorků – pokud možno nejméně 500, lépe 1000.[80]
- Pro každý vzorek sestrojíme zvolenou metodou dendrogram.
- Nakonec vypočteme ze všech získaných stromů **konsensus** a pro každou vnitřní větev stanovíme, v jaké frakci vzorků byla nalezena. (Můžeme to chápat tak, že kterákoli vnitřní větev rozděluje strom na dvě části; počítáme vždy frakci vzorků, v nichž byly od sebe odděleny tytéž dvě skupiny OTU jako v konsensuálním stromu.) Vypočtené **hodnoty bootstrapové podpory** (*bootstrap values*) se často konvenčně zapisují na „periferní“ konce větví dendrogramu – znamenají totiž současně frakci případů, v nichž příslušný uzel sdružuje tytéž OTU jako v konsensuálním stromu (viz Obr.11.1. – 11.3.).

Hodnoty bootstrapové podpory vypovídají o **robustnosti** struktury dendrogramu, tedy o tom, jak snadné či obtížné je tuto strukturu změnit přidáním či ubráním dat stejného charakteru, jako byla ta výchozí. Zřejmě je však nelze interpretovat přímo jako „pravděpodobnost, že daná větev je správná“. Za spolehlivé se zpravidla považují větve s hodnotami bootstrapové podpory 90 – 100 %, naopak hodnoty pod 50 % bychom neměli brát příliš vážně. U nízkých hodnot bootstrapové podpory si nepomůžeme ani tehdy, když danou větev získáme několika metodami. Jak výstižně formuloval J. Felsenstein v manuálu k softwarovému balíku PHYLIP (1989), dvakrát nezávisle získaná bootstrapová podpora 51 % se příliš neliší od dvakrát nezávisle získaných 49 %, za kterými by nám nestálo za to se ohlížet.



Obr.10.3. Princip konstrukce a hodnocení dendrogramu s využitím bootstrappingu.

Kromě testování robustnosti dendrogramu vzorkovacími metodami zejména u heuristických postupů stojí za to vyzkoušet, jak struktura stromu reaguje na **permutaci** (změnu pořadí) OTU v zadávaném přiřazení. Některé programy dovolují permutaci sekvencí (*jumble*) již při vlastním výpočtu s tím, že výsledný konsensuální strom zahrnuje jak bootstrapové vzorky, tak i permutace pořadí druhů.

10.4. Praktická konstrukce dendrogramu

Soustřeďme se nyní na typickou situaci molekulárního biologa, který si chce „udělat pořádek" v početné genové rodině, u níž zná zástupce z několika modelových organismů. Neví však, kolik z těchto zástupců představuje dávné paralogní linie, jejichž funkce se hluboce různí, a které jsou „pouze" produkty poměrně nedávných genových duplikací. Dobrou cestou k získání odpovědi na tyto otázky je konstrukce stromu; prakticky tedy jde o to, jak vytvořit dendrogram pro několik desítek nepříliš dlouhých sekvencí proteinů.[81]

První, co vždy stojí za to spočítat, je **NJ strom s vysokým počtem cyklů bootstrapového vzorkování**. Pro jistotu bychom se však měli pokusit i o jiné postupy – jistě ML, dovoluje-li to velikost souboru, a pokud z NJ nevycházejí příliš dlouhé větve, také případně i MP.

10.4.1. Příprava výchozích dat

Zásadním pravidlem, které má téměř povahu zákona (a takových v biologii moc není), je že **kvalita dendrogramu je podmíněna kvalitou přiřazení a ta zase kvalitou vstupních sekvenčních dat**.[82] Náš strom nemůže být lepší než data, na kterých je založen, a proto si na datech musíme dát záležet. Samozřejmým předpokladem je, že sekvence, jimiž se zabýváme, jsou pokud možno správné (u proteinových sekvencí odvozených z genomu by vždy měla být provedena revize struktury genu – viz 9. kapitola) a že jsme z nich mnohočetné přiřazení sestrojili s náležitou pečlivostí (8. kapitola).

Mnohočetné přiřazení, jakkoli optimalizované, však obvykle nemůžeme použít ke konstrukci dendrogramu bez dalších úprav. Především při použití znakových metod totiž většinou tiše předpokládáme, že se sekvence vyvíjely prostřednictvím **bodových mutací**. Dokonce i u některých metod vzdálenostních je zacházení s jinými než substitučními mutacemi problematické. Zatímco pro výpočet vzdálenosti sekvencí, které hromadily pouze bodové mutace, můžeme použít některou ze standardních substitučních matic (5.3.), už s cenou za otevření a prodloužení mezery mohou nastat potíže, zejména jestliže pracujeme s bootstrapovými vzorky. Už vůbec si pak neumíme poradit se situací, kdy některá ze studovaných sekvencí podlehla rozsáhlejším přestavbám, třeba vložení transpozonu či ztrátě nebo naopak zisku exonu.

Proto je nezbytným krokem **selekce („filtrace“) dat** – musíme ručně vybrat jen ty oblasti přiřazení, které uvedený předpoklad splňují (Brinkman a Leipe 2001). Zejména musíme ořezat ty úseky, v nichž se nám sekvence buď nepodařilo přiřadit, nebo přiřazení vypadá podezřele – například obsahuje vícero krátkých mezer. Zbude-li nám dostatek pozic pro analýzu, neuškodí, když vypustíme dokonce všechny pozice, v nichž aspoň jedna ze sekvencí přiřazení obsahuje mezery (delece či inzerce několika málo znaků však někdy musíme tolerovat – kdybychom je vyloučili, nic by nám z dat nezbylo). Zbývající dobře přiřazené úseky, které zpravidla odpovídají konzervativním doménám, pak prostě pospojujeme. Takový zásah do dat může na první pohled vypadat jako nezdůvodnitelná svévole, avšak opak je pravdou. Kdybychom totiž v datovém souboru nechali úseky, které nevyhovují předpokládanému modelu evoluce (hromadění bodových mutací), pracovali bychom s kontaminovanými daty, což by zřejmě zkreslilo výsledky. Naopak vyhodíme-li nedopatřením něco, co mělo zůstat (třeba proto, že jsme příliš důvěřovali CLUSTALu a nechali jsme si do přiřazení zanést víc mezer, než bylo zdrávo), nestane se zase tak velké neštěstí, pokud nám ještě zbyl dostatek dat – pravděpodobně nanejvýš o něco poklesne průkaznost výsledku. Ostatně při bootstrapové analýze vlastně také nahodile vybranou část dat záměrně vylučujeme, a sledujeme, nakolik to strom ovlivní; lze očekávat, že přinejmenším větve s vysokou bootstrapovou podporou neohrozí ztráta nevelké části dat, o nichž máme pochybnosti.[83]

10.4.2. Metoda NJ (*neighbor-joining*)

Ke konstrukci dendrogramu metodou NJ můžeme využít několik volně dostupných programů. My se zde zaměříme především na postup využívající nástrojů z dnes již klasické Felsensteinovy sady programů PHYLIP. Dotkneme se však i dvou modernějších, rychlejších, z hlediska obsluhy jednodušších, ale zato méně názorných metod výpočtu NJ dendrogramů.
Softwarový balíček PHYLIP (*Phylogeny Inferrence Package*; Felsenstein, 1989; Felsenstein, 1996) obsahuje několik desítek programů, určených ke konstrukci a vyhodnocování fylogenetických stromů na základě dat nejrůznějšího typu. Programy pro práci se sekvencemi DNA a proteinů jsou pouze zlomkem celé nabídky. Celou sadu programů, která představuje vskutku klasické dílo molekulární fylogenetiky, a přitom její vývoj pokračuje dodnes, lze získat zdarma z autorových stránek (`http://evolution.genetics.washington.edu/phylip.html`) i s velice důkladnou a přehledně napsanou dokumentací. Registrace se sice vyžaduje, ale je spíš slušností než nezbytností.

Jistou nevýhodou je uživatelské rozhraní a vůbec celý **způsob práce s programy sady PHYLIP**, který je dědictvím dob z hlediska vývoje výpočetní techniky dosti dávných, a

u nezvyklého uživatele, zhýčkaného prací ve Windows XP, může vyvolávat husí kůži. Naopak uživatel, který rád vidí „točit kolečka“ a ve kterém vyvolávají nostalgické rozněžnění mechanické šicí stroje, Warburgův respirometr nebo třeba technické finesy spřáhel tramvají, bude potěšen, mimo jiné i proto, že součástí distribuce PHYLIPu je plné znění zdrojového kódu v jazyce C! Nemusíme si z něj naštěstí sami kompilovat spustitelné programové soubory, protože ty již jsou připraveny ke stažení, dokonce ve verzích pro různé operační systémy (Unix, Linux, Windows, Mac OS). Instalace je na dnešní poměry nebývale jednoduchá, a vlastně ani není instalací v pravém slova smyslu – stačí rozbalit všechny stažené programové soubory do jednoho adresáře, do kterého máme právo zápisu a ve kterém budeme (proti dnes běžným zvyklostem) držet pospolu programy i data.

Všechny programy, které ke konstrukci NJ stromu budeme potřebovat, běží v okně příkazového řádku (*command prompt*, emulace MS-DOS), používají velmi podobné negrafické **uživatelské rozhraní**, umožňující ovládání pomocí výběru z menu prostřednictvím jednopísmenových příkazů (Obr.10.4.), a dodržují konvenční názvy vstupních a výstupních souborů. Vstupní data se vždy nacházejí v souboru nazvaném `infile` (kromě stromů ve formátu Newick, které sídlí v souboru `intree`); výstupní soubory se jmenují `outfile` (textové soubory jiné než stromy), `outtree` (stromy), popřípadě `plotfile` (grafické soubory).[84]

Vlastní konstrukce NJ stromu pro sadu proteinových sekvencí probíhá takto:

- Připravíme **vstupní data** („ořezané“ mnohočetné přiřazení) v podobě textového souboru ve formátu PHYLIP(i) (viz 8.1.). Tento soubor zkopírujeme do adresáře s programy sady PHYLIP a přejmenujeme na `infile` (bez přípony).
- Ze vstupních dat odvodíme **sadu bootstrapových vzorků** (nejlépe aspoň 500) za použití programu SEQBOOT, který spustíme poklepáním na jeho ikonu. Parametry můžeme nastavit podle legendy Obr.10.4.; význam dalších volitelných nastavení tohoto i ostatních programů je důkladně popsán v dokumentaci. Program nám vytvoří soubor `outfile`. Nyní z adresáře odstraníme (přejmenujeme či přestěhujeme) původní `infile` a pak `outfile` přejmenujeme na `infile`. Veškeré meziprodukty stojí za to nějakou dobu uchovávat – tytéž bootstrapové vzorky můžeme použít i pro konstrukci dendrogramu jinou metodou.
- Pro všechny vzorky spočítáme **matici vzdáleností** programem PROTDIST; musíme přitom použít volbu „*multiple data sets*“ a znovu zadat počet vzorků. Můžeme si také vybrat substituční matici. Tento krok je nejpomalejší, jeho trvání roste s druhou mocninou počtu sekvencí. Při konstrukci dendrogramu ze sekvencí DNA bychom postupovali obdobně, ale použili bychom program DNADIST; po skončení opět odstraníme `infile` a pak přejmenujeme `outfile` na `infile`.
- Pomocí programu NEIGHBOR **vypočteme NJ stromy** pro každý vzorek. Opět musíme zadat počet vzorků jako v předchozím kroku; navíc můžeme specifikovat, zda se má permutovat pořadí sekvencí (implicitní nastavení, doporučeno) a zda strom má být zakořeněný (pokud ne, program bere první sekvenci jakožto formální „outgroup“). Mohli bychom také namísto metody NJ zvolit UPGMA, ale už víme, že chceme-li získat fylogram a ne fenogram, není to radno. Výstupem jsou tentokrát dva soubory – `outfile` a `outtree`. Odstraníme `outfile` a přejmenujeme `outtree` na `intree`.
- Konečný **konsensuální strom a hodnoty bootstrapové podpory** získáme pomocí programu CONSENSE (měli bychom nastavit kořen či „nezakořeněnost“ stromu stejně jako v předchozím kroku). Výstupní soubor `outfile` obsahuje výsledný strom

včetně hodnot podpory v textovém formátu; v souboru `outtree` je strom ve formátu Newick.

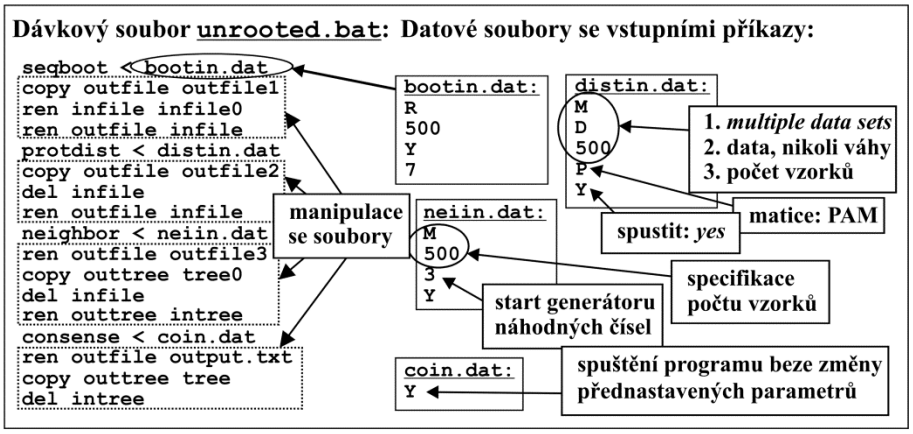
Pokud NJ stromy konstruujeme často, stojí za to celý postup automatizovat pomocí **dávkového souboru** MS-DOS (*MS-DOS batch file*), který postupně vyvolává uvedené procedury a provádí veškeré nutné manipulace se soubory (Obr.10.5). Dávkový soubor napíšeme v textovém editoru, umístíme do adresáře s programy PHYLIP a daty a opatříme příponou `*.bat`, čímž se stane spustitelným. Příkazy, které bychom měli zadávat interaktivně z klávesnice, musíme pro každý program zvlášť zapsat do textového souboru (s libovolnou příponou – v uvedeném příkladu `*.dat`) přesně v tom pořadí, v jakém bychom je zadávali ručně.

```
Bootstrapping algorithm, version 3.6a2.1

Settings for this run:
  D      Sequence, Morph, Rest., Gene Freqs?  Molecular sequences
  J  Bootstrap, Jackknife, Permute, Rewrite?  Bootstrap
  B      Block size for block-bootstrapping?  1 (regular bootstrap)
  R                      How many replicates?  100
  W               Read weights of characters?  No
  C                 Read categories of sites?  No
  F     Write out data sets or just weights?  Data sets
  I             Input sequences interleaved?  Yes
  0      Terminal type (IBM PC, ANSI, none)?  (none)
  1       Print out the data at start of run  No
  2     Print indications of progress of run  Yes

  Y to accept these or type the letter for one to change
```

Obr.10.4. Uživatelské rozhraní programů ze sady PHYLIP – příklad programu SEQBOOT, který ze zadaného mnohočetného přiřazení generuje sadu bootstrapových vzorků. Chceme-li např. změnit počet vzorků na 500, zadáme `R`, na následující otázku po počtu replikací zadáme `500` a parametry schválíme zadáním `Y`. Program si poté vyžádá zadání libovolného lichého čísla k odstartování generátoru náhodných čísel (*random number seed*), zadáme třeba `7`.



Obr.10.5. Výpis dávkového souboru pro automatizaci konstrukce nezakořeněného NJ stromu pomocí programového balíku PHYLIP postupem uvedeným v textu. Výchozí přiřazení (ve formátu PHYLIP) se zadává v souboru `infile`, výstup se objeví jako `output.txt` (textový formát) a `tree` (formát Newick). Posloupnost příkazů v souboru `bootin.dat` přesně odpovídá postupu z legendy k Obr.10.4.

Grafickou prezentaci stromu můžeme zhotovit několika způsoby. PHYLIP sám nabízí dva programy na kreslení stromů – DRAWTREE pro nezakořeněné a DRAWGRAM pro zakořeněné dendrogramy; strom ve formátu Newick do nich zadáváme prostřednictvím souboru `intree`.[85] Oba dovedou vyprodukovat výstup (`plotfile`) v řadě formátů včetně

„obecného postscriptu“, který lze importovat do některých kreslicích programů (např. CorelDraw) pro další grafické úpravy. K dispozici jsou také bitmapové formáty `*.pcx` a `*.bmp` o omezeném rozlišení a stará verze „scén“ pro PoVRay (viz 8.3.), který vytváří obrazy sice graficky působivé, ale pro použití v publikaci možná až příliš extravagantní.[86] Pro malé dendrogramy možná vůbec nejjednodušší, i když nepříliš elegantní, je obkreslit v nějakém grafickém editoru (jako je třeba CorelDraw, nebo přinejhorším i kreslicí funkce MS Wordu) textový strom ze souboru `outfile`. Hodnoty bootstrapové podpory musíme však v každém případě dopsat ručně podle souboru `outfile`, popřípadě vyjádřit vhodnými symboly na uzlech stromu (viz Obr.11.1 až 11.3). Strom ve formátu Newick lze také použít jako vstup pro mnohé další programy s širším výběrem grafických možností; dva z nich budou diskutovány dále.

Konstrukce NJ stromu pomocí programů sady PHYLIP je velmi názorná, dovoluje sledovat všechny kroky a nastavovat řadu parametrů, avšak je poměrně **náročná na ruční práci uživatele i čas**. Zejména v systému Windows programy zdaleka nevyužívají všech možností počítače a jsou dosti pomalé (viz Tab.10.2.). Proto zde zmíním aspoň dva z programů, které nabízejí rychlejší a z hlediska obsluhy jednodušší alternativy.

S prvním z nich jsme se již setkali v kapitole 8.2.1. – jednoduché, avšak velmi výkonné nástroje pro konstrukci NJ stromu totiž obsahuje **Clustal X** (Thompson et al., 1997). Metoda CLUSTAL vytváří mnohočetné přiřazení na základě „vůdčího stromu“ získaného právě metodou NJ, takže dendrogram touto metodou spočítat dovede. Ke konstrukci dendrogramu můžeme využít jak přiřazení čerstvě vytvořené metodou CLUSTAL v samotém programu Clustal X, tak i přiřazení vytvořené jiným programem a importované ve formátu `*.aln` či FASTA (viz 8.1.). Nemusíme se zabývat ani odstraňováním mezer, protože program pro výpočet dendrogramu dovede ignorovat všechny pozice s mezerami (přepínač v menu *Trees – Exclude positions with gaps*). Sama konstrukce stromu je velmi jednoduchá – stačí vybrat příkaz *Trees – Bootstrap N-J tree* a zvolit počet bootstrapových vzorků a libovolné liché číslo pro odstartování generátoru náhodných čísel. Ostatní parametry (např. substituční matici) nastavujeme přes nabídku *Alignment – Alignment parameters*. Výstupem je soubor s příponou `*.phb`, který obsahuje dendrogram v rozšířené verzi formátu Newick, připouštějící i zápis hodnot bootstrapové podpory (v hranatých závorkách u každé větve).[87] Konstrukce stromu programem Clustal X je velice rychlá (Tab.10.2.), nevýhodou je poměrně malá možnost změny přednastavených parametrů a nepříliš podrobná dokumentace (aspoň ve srovnání s PHYLIPem). Nelze se např. snadno dopátrat toho, zda program permutuje pořadí sekvencí v bootstrapových vzorcích a zdali si výpočet nezjednodušuje heuristicky, jak by rychlost naznačovala.

Distribuce programu Clustal X obsahuje i dva samostatné programy pro **vizualizaci dendrogramů** ve standardním newickovském formátu i v jeho rozšířené verzi s hodnotami bootstrapové podpory; jsou tedy vhodné i pro prezentaci stromů vytvořených třeba pomocí sady PHYLIP. Oba programy dovedou zapisovat grafické soubory ve formátu Postscript (`*.ps`), který lze dále editovat běžnými komerčními grafickými programy. Program UNROOTED vytváří nepříliš pohledné hvězdicovité stromy podobné těm z programu DRAWTREE (včetně toho, že neumí vypsat hodnoty podpory jednotlivých větví). Program **NJPLOT** však poskytuje mnohem víc možností – nejen že do obrázku píše i hodnoty bootstrapové podpory, ale dovoluje i „přestavbu“ stromu otáčením větví (Obr.10.2.) a interaktivní změnu outgroup (příkazy *Swap nodes* a *New outgroup*).

Zajímavý kompromis mezi bohatými možnostmi uživatelského nastavení sady PHYLIP a rychlostí a příjemným grafickým rozhraním programu Clustal X poskytuje **Treecon** (Van de Peer a De Wachter, 1994). Přijímá vstupní data v několika formátech (včetně obou verzí formátu PHYLIP) a navíc obsahuje užitečný nástroj na konverzi formátů mnohočetného přiřazení ForCon (viz 8.1.). Postup práce s programem prakticky sleduje pořadí tlačítek v interaktivním grafickém menu:

- V prvním kroku (***Distance estimation***) vytvoříme bootstrapové vzorky a spočteme matice vzdáleností. Je nutné specifikovat typ dat a formát souboru, vybrat ze vstupního přiřazení sekvence, které budou zahrnuty do analýzy (obvykle všechny) a v dalším kroku vyvolat nabídku „*Bootstrap analysis – Yes*“, která otevře dialogové okno k výběru počtu bootstrapových vzorků. Program také dovoluje volit z několika různých postupů výpočtu vzdálenosti, a umožňuje dokonce brát v úvahu mezery (jistější však asi je mezery vypustit).
- Následující krok (***Infer tree topology***) je přibližně obdobou programu NEIGHBOR; opět musíme určit počet bootstrapových vzorků.
- V dalším kroku (***Root unrooted trees***), který zhruba odpovídá programu CONSENSE, program vyžaduje opětovné zadání počtu bootstrapových vzorků a volbu „outgroup“ dokonce i pro nezakořeněný strom (připomeňme si, že i PHYLIP zachází s nezakořeněnými stromy jako se zakořeněnými, uživatel však nemá na vybranou – roli outgroup vždy hraje první sekvence v přiřazení).
- Posledním krokem je vizualizace stromu (***Draw phylogenetic tree***). Nevýhodou programu Treecon je, že používá vlastní formát zápisu stromů `*.trc`, nekompatibilní s formátem Newick. Umí však stromy ve formátu Newick čili „New Hampshire bracket format“ otvírat a editovat, přičemž poskytuje více možností než NJPLOT. Vytvořený strom otevřeme příkazem „*File – Open – (new) tree*“ a po případné grafické úpravě jej můžeme uložit ve formátu `*.trc`. Treecon umí jen posílat výstupy na tiskárnu či na obrazovku a kopírovat výstup z obrazovky (o nevalném rozlišení) do schránky (*Clipboard*) příkazem „*File – Copy*“. Tento nedostatek lze nepříliš elegantně obejít tím, že na fiktivní port `FILE:` instalujeme postscriptovou tiskárnu, „odchytíme“ do souboru výstup ve formátu PostScript a ten pak importujeme do vhodného grafického editoru.[88]

10.4.3. Další metody konstrukce dendrogramů

Navzdory již zmiňovaným nevýhodám někdy stojí za to počítat dendrogram také metodou ***maximum parsimony*** **(MP)**, i když není příliš vhodná pro sekvence s proměnlivou frekvencí mutací jednotlivých pozic (což je obvyklé u proteinů). Pracujeme-li však s početnými soubory sekvencí, pro které nelze hledat nejvěrohodnější strom (viz níže), je metoda MP prakticky jedinou jednoduše dostupnou možností, jak dospět k dendrogramu jinak než algoritmem NJ. I když je MP pro některá data výrazně méně spolehlivá než NJ, lze při dobré bootstrapové podpoře pokládat shodu mezi NJ a MP za dobrý doklad správnosti dendrogramu, i když zřejmě jen pro stromy, které neobsahují shluky dlouhých větví. Naopak tam, kde máme podezření, že od divergence studovaných sekvencí uplynula dlouhá doba a došlo k mnoha evolučním změnám, bychom se měli používání metody MP vyvarovat.

Sada programů **PHYLIP** obsahuje nástroje na konstrukci MP stromů pro sekvence DNA (DNAPARS) i proteinů (PROTPARS). Průběh výpočtu je obdobný jako pro NJ dendrogram (10.4.2.), až na to, že stanovení matic vzdáleností a následná konstrukce dendrogramů z bootstrapových vzorků jsou nahrazeny jediným krokem – výpočtem dendrogramů metodou MP.

Postupujeme tedy tak, že v uvedeném pořadí aplikujeme programy SEQBOOT (příprava bootstrapových vzorků), DNAPARS nebo PROTPARS (výpočet MP stromů pro všechny vzorky) a CONSENSE (stanovení konsensuálního stromu a hodnot bootstrapové podpory). Metodu lze automatizovat pomocí dávkových souborů podobně jako v předchozím případě, a také trvání výpočtu je srovnatelné s konstrukcí dendrogramu metodou NJ (Tab.10.2.).

Tab.10.2. Porovnání vybraných metod konstrukce dendrogramu pro soubor 10 sekvencí aminokyselin vybraných členů nadrodiny regulátorů malých GTPáz REP/GDI (viz též kap. 8.4.) o délce 365 znaků. Výpočet proběhl na PC s operačním systémem Windows XP, procesorem Pentium 4 (2,66 GHz), 512 MB RAM, ve všech případech byl použit bootstrap s 500 cykly vzorkování. V případě označeném hvězdičkou byla použita volitelná přesnější, ale pomalejší metoda.

Metoda	**Program**	**Trvání výpočtu**	**Konsensuální strom** (větve s podporou nad 80 % tučně, nad 90 % bíle na černém pozadí)
NJ	PHYLIP/NEIGHBOR	9 min	(((**(((A,B),(C,D)),(E,F))**),(G,H)),I),J)
NJ	TreeCon	1 min	(((**(((A,B),(C,D)),(E,F))**),(G,H)),I),J)
NJ	Clustal X	méně než 30 s	(((**(((A,B),(C,D)),(E,F))**),(G,H)),I),J)
MP	PHYLIP/PROTPARS	4 min	(((((A,B),(C,D)),F),E),(G,H)),I),J)
ML*	PHYLIP/PROML	2 h 53 min	(((**((A,B)**,(C,D)),(E,F))**,(G,H)**),I),J)
ML	PHYLIP/PROML	1 h 7 min	(((**((A,B)**,(C,D)**)**,(E,F))**,(G,H)**),I),J)
ML	PHYML	57 min	(((**((A,B)**,(C,D))**,(E,F))**,**(G,H)**),I),J)

Ideální je, můžeme-li kromě NJ dendrogramu vypočítat také „nejvěrohodnější“ strom metodou ***maximum likelihood*** **(ML)**. Ne vždy je to prakticky proveditelné, protože časová náročnost výpočtu dramaticky roste s počtem sekvencí, a mírně i s jejich délkou (Tab.10.3.). U velkých souborů se musíme často smířit s výrazně nižším počtem bootstrapových vzorků, než je obvyklých několik set, a u hodně velkých souborů čítajících několik set sekvencí ML dendrogram na běžném PC prostě nespočteme. Rozhodně se vždy vyplatí vyzkoušet, jak dlouho trvá výpočet pro *jednu* sadu sekvencí, dřív než spustíme analýzu se sadou bootstrapových vzorků.

Tab.10.3. Porovnání rychlosti výpočtu jednoho ML dendrogramu (bez bootstrapu) pro různý počet sekvencí. Použitý hardware byl stejný jako pro Tab.10.2., výpočet probíhal na pozadí (což je sřejmě důvodem proměnlivé rychlosti). Přiřazení o délce 118 pozic bylo vybráno z dat použitých ke konstrukci dendrogramu FH2 domén v (Cvrčková et al., 2004b). Ve sloupci označeném hvězdičkou bylo analyzováno přiřazení poloviční délky, získané odstřižením poloviny každé ze sekvencí ve výchozím souboru.

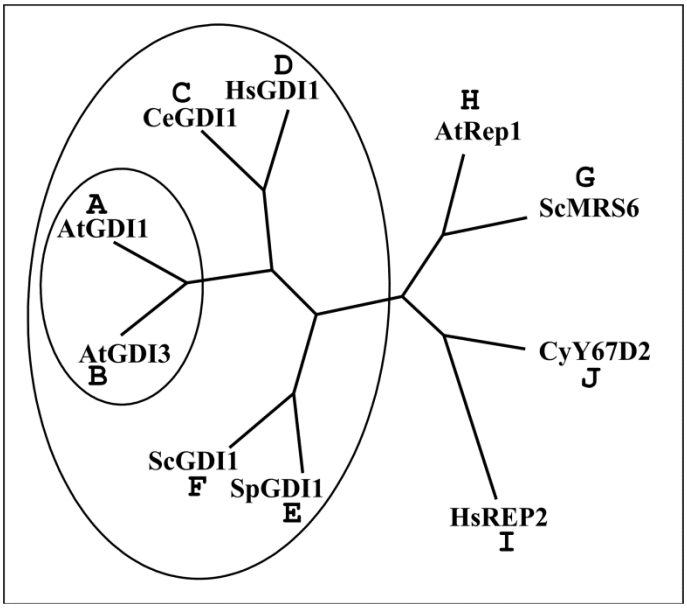
Metoda	**10 sekvencí**	**15 sekvencí**	**30 sekvencí**	**30 sekvencí***	**60 sekvencí**
PROML (pomalu, přesně)	20 s	45 s	7 min 35 s	4 min 21 s	35 min 12 s
PROML (rychle, přibližně)	10 s	27 s	1 min 1 s	1 min 2 s	4 min 21 s
PHYML	5 s	7 s	21 s	13 s	46 s

K praktickému vyhledávání ML dendrogramu můžeme opět použít programy ze sady **PHYLIP** – DNAML pro DNA a PROML pro proteiny. [89] Postup jejich použití je stejný jako v případě výše popsané konstrukce MP stromu, pouze v popsané proceduře použijeme DNAML, popřípadě PROML, namísto DNAPARS či PROTPARS. Oba programy využívají heuristických postupů a umožňují zvolit buď rychlé, ale méně přesné, nebo přesné, avšak pomalejší provedení. I při použití „rychlé“ metody však výpočet trvá výrazně déle než u ostatních dosud zmíněných metod.

Výraznou úsporu času přináší heuristický postup používaný programem **PHYML** (Guindon a Gascuel, 2003), dostupným na `http://atgc.lirmm.fr/phyml/` dokonce i v online verzi (viz Tab.10.2 a Tab.10.3.). Program navenek vypadá podobně jako nástroje sady

PHYLIP a je s nimi kompatibilní; lze jej tedy „nakrmit“ třeba daty vytvořenými programem SEQBOOT a z výstupního souboru vyrobit konsensuální strom pomocí CONSENSE. Autoři sami varují před možnými artefakty při analýze sekvencí, které obsahují oblasti o dramaticky odlišné evoluční historii (jako příklad uvádějí zřetězené sekvence několika genů, které se běžně používají ve fylogenetice organismů, ale totéž riziko hrozí i u sekvencí genů, které vznikly spojením několika domén).

V praxi je někdy **volba metody** politováníhodně snadná: jediné, co stojí za to počítat pro hodně divergentní sekvence, je NJ a ML, a jediné, co technicky *lze* spočítat pro velkou sadu dat, je NJ a MP, takže rádi zůstaneme u metody *neighbor-joining*. Máme-li štěstí, můžeme porovnat všechny tři metody. V tabulce 10.2. je velmi stručně shrnut jeden takový případ – totiž analýza nadrodiny regulátorů malých GTPáz REP/GDI (viz též Žárský et al., 1997).[90] Zjednodušený zápis topologie dendrogramu s vyznačením větví s vysokou bootstrapovou podporou ukazuje na dobrou shodu téměř všech použitých postupů – s jedinou výjimkou (metoda MP, u níž se lze obávat nepřesností) je větvení stromu stejné, liší se jen hodnoty bootstrapové podpory. Všechny metody se také shodují na velmi vysoké bootstrapové podpoře dvou větví. Jedna z nich odděluje (bereme-li strom jako nezakořeněný) skupinu genů A až F od genů G až J a vymezuje tedy vzájemně rodiny REP a GDI, druhá spojuje dohromady sekvence A a B, které zřejmě vznikly nedávnou duplikací genu pro GDI v *Arabidopsis* (Obr.10.6.).



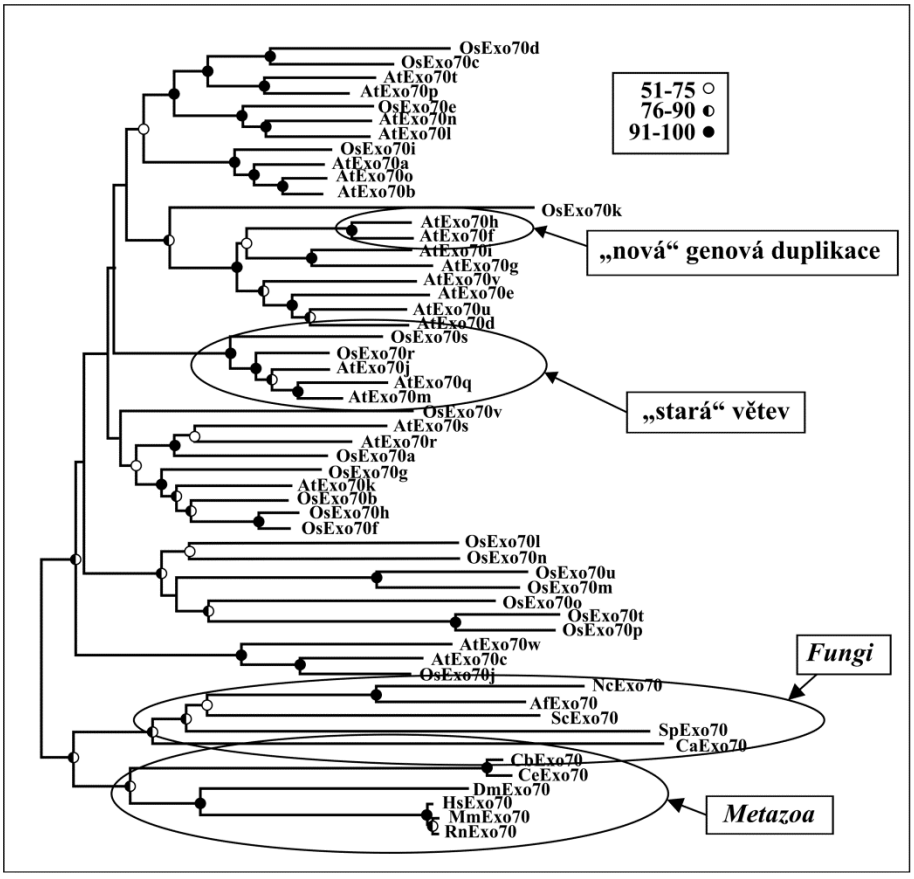
Obr.10.6. Topologie většinového konsensuálního stromu z tabulky 10.2. s vyznačením větví, které získaly více než 90 % podporu všemi metodami. Zobrazen je nezakořeněný NJ strom sestrojený pomocí programů sady PHYLIP.

Uzavřeme nyní naši cestu za fylogenetickým stromem, která začala vlastně již „hrátkami“ s formáty sekvencí a prohledáváním databází v počátečních kapitolách, třemi konkrétními příklady z naší laboratoře. Všechny tři se týkají poněkud nestandardní aplikace fylogenetických metod – totiž jejich využití ke studiu příbuzenských vztahů ve **velkých genových rodinách** s řadou vzájemně paralogních členů.

11. Tři příklady na závěr

Soustřeďme se na závěr na tři příklady bioinformatické analýzy velkých genových rodin u krytosemenných rostlin. Pokusíme se na nich demonstrovat nejen postup od identifikace genů až po konstrukci a interpretaci dendrogramu, ale i některé **biologické problémy**, u nichž mohou bioinformatické přístupy přispět jak k formulaci řešitelných otázek, tak i k hledání experimentálních cest směřujících k odpovědím.

Prvním příkladem je vcelku přímočará analýza rodiny **rostlinných homologů proteinu Exo70** – jedné z podjednotek komplexu Exocyst, který se u hub a živočichů účastní konečných kroků exocytózy a zřejmě se podílí mj. na určení místa v rámci buněčného povrchu, kde má dojít k fúzi sekretorických váčků (Obr.11.1.). Rostlinné homology všech podjednotek Exocystu byly identifikovány v databázích metodou BLAST. Po revizi genové struktury byl zkonstruován dendrogram metodou NJ pro všechny homology z *Arabidopsis* a dostupnou část genů z rýže (Cvrčková et al. 2001, Eliáš et al., 2003).

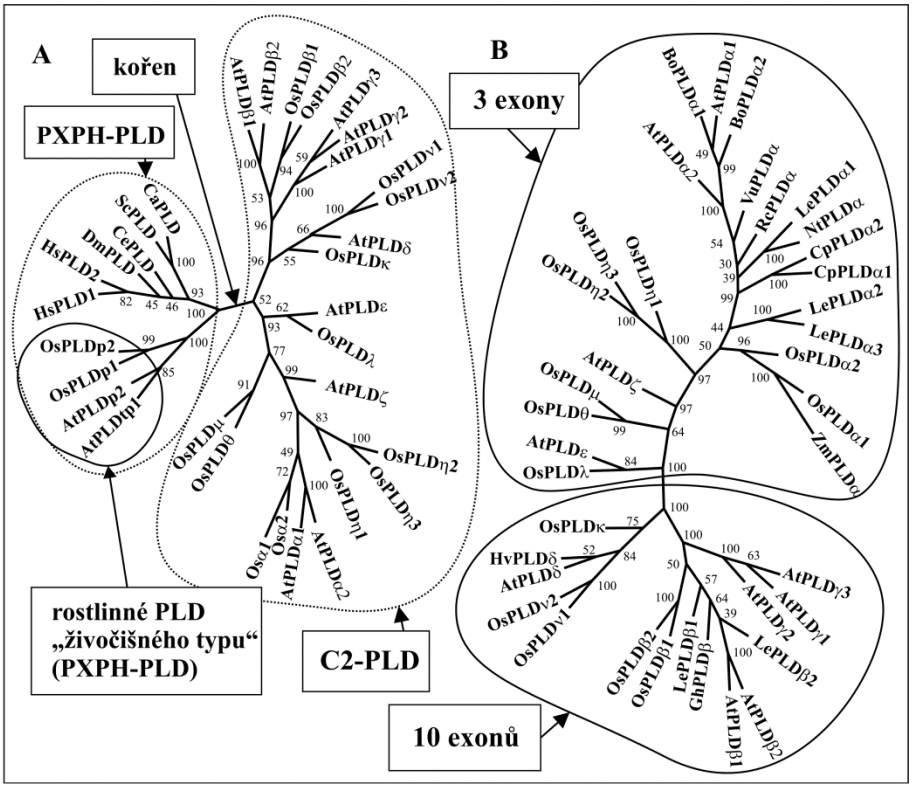


Obr.11.1. Nezakořeněný NJ dendrogram rostlinných příslušníků genové rodiny Exo70 a příbuzných proteinů z hub a živočichů (podle Eliáš et al., 2003). Symboly v uzlech stromu vyjadřují bootstrapovou podporu v procentech (z 500 vzorků); At – *Arabidopsis thaliana*, Os – *Oryza sativa*, Af – *Aspergillus flavus*, Nc – *Neurospora crassa*, Sc – *Saccharomyces cerevisiae*, Ca – *Candida albicans*, Cb – *Caenorhabditis briggsae*, Ce – *C. elegans*, Dm – *Drosophila melanogaster*, Hs – *Homo sapiens*, Mm – *Mus musculus*, Rn – *Rattus norvegicus*.

Interpretace výsledků analýzy rodiny Exo70 je celkem neproblematická, zejména vzhledem k tomu že studovaný protein sestává z jediné domény. Překvapivé a vzrušující bylo jen zjištění, že rostlinné genomy obsahují celou řadu kopií genů kódujících proteiny homologní Exo70, zatímco živočichové i houby si vystačí s jedinou kopií na genom. Konstrukce

dendrogramu (Obr.11.1.) pak jasně prokázala, že **série genových duplikací** u rostlin zřejmě započala dosti dávno, ještě před oddělením jednoděložných a dvouděložných, protože lze identifikovat dobře podpořené genové podrodiny obsahující jak geny z *Arabidopsis*, tak jejich protějšky z rýže. Zmnožování genů však pokračovalo i poté, takže v některých podrodinách nacházíme stopy celé série duplikačních událostí, podobně jako v případě jiných podobně početných genových rodin (namátkou můžeme vybrat třeba rostlinné malé GTPázy třídy Rop (Li et al., 1998), nebo níže podrobněji diskutované fosfolipázy D a forminy). Výsledkem je stav, kdy sice můžeme jednotlivé geny zařadit do specifických podrodin, avšak v rámci těchto podrodin již nelze přímo určit, které geny různých druhů jsou vzájemně ortologní.

Podobný obraz poskytuje i případ genové rodiny **rostlinných fosfolipáz D (PLD)** – enzymů, které zřejmě hrají klíčovou úlohu v řadě signálních drah, z nichž některé se podílejí na morfogenezi buňky (Eliáš et al., 2002). [91] Situace zde však je složitější v tom smyslu, že protein obsahuje kromě evolučně konzervativní domény charakteristické pro PLD také sekvenčně variabilní oblasti, ve kterých se mohou nacházet další konzervativní motivy. Dendrogramy (Obr.11.2.) byly sestrojeny na základě té části sekvence, pro kterou bylo možno vytvořit jednoznačné přiřazení (viz 10.4.1.).



Obr.11.2. Fylogeneze genové rodiny rostlinných fosfolipáz D (PLD) a jejich vztah k fosfatidylcholinspecifickým PLD (PC-PLD) živočichů a hub (podle Eliáš et al., 2002). Oba stromy jsou nezakořeněné, sestrojené metodou NJ, čísla uvádějí % bootstrapové hodnoty (z 500 vzorků); poloha kořene v levé části byla zjištěna konstrukcí zakořeněného stromu s bakteriální PLD jakožto outgroup. Levý strom (A) obsahuje pouze sekvence z *Arabidopsis*, rýže a vybraných živočišných a houbových druhů a demonstruje vztah rostlinných PLD k živočišným a jejich členění na dvě skupiny, pravý (B) představuje podrobnější analýzu třídy C2-PLD, specifické pro rostliny. At – *Arabidopsis thaliana*, Bo – *Brassica oleracea*, Ca – *Candida albicans*, Ce – *Caenorhabditis elegans*, Cp – *Craterostigma plantagineum*, Dm – *Drosophila melanogaster*, Gh – *Gossypium hirsutum*, Hs – *Homo sapiens*, Hv – *Hordeum vulgare*, Le – *Lycopersicon esculentum*, Nt – *Nicotiana tabacum*, Os – *Oryza sativa*, Sc – *Saccharomyces cerevisiae*, Rc – *Ricinus communis*, Vu – *Vigna unguiculata*, Zm – *Zea mays*.

Z analýzy dendrogramu zahrnujícího PLD z *Arabidopsis* a rýže a vybrané houbové a živočišné enzymy (Obr.11.2. vlevo) je zřejmé, že rostlinné PLD tvoří dvě dobře oddělené podrodiny, které se vyznačují specifickým **uspořádáním domén**. Jedna z nich zahrnuje všechny doposud experimentálně charakterizované rostlinné proteiny a vyznačuje se přítomností N-koncové C2 domény, která váže fosfolipidy (proto pro ně bylo navrženo označení C2-PLD). Druhá větev zahrnuje jak živočišné a houbové proteiny, tak i poměrně nepočetné a až dosud nepopsané zástupce rostlinné. Vzhledem k přítomnosti dvou různých domén vážících fosfolipidy – PX (*phox*) a PH (*pleckstrin homology*) – pro ně bylo navrženo označení PXPH-PLD.

Podrobnější analýza větší, specificky rostlinné rodiny C2-PLD se zahrnutím sekvencí z dalších druhů (Obr.11.2. vpravo) odhalila stopy většího počtu genových duplikací (včetně některých zřejmě poměrně nedávných). To prakticky vyloučilo identifikaci vzájemně ortologních proteinů z různých druhů, podobně jako v předchozím příkladu. S vysokou bootstrapovou podporou (500 z 500 vzorků) byly však určeny dvě podtřídy C2-PLD, jejichž rozdělení zjevně předcházelo rozštěpení rostlinné říše na předky dnešních jednoděložných a dvouděložných. Dlouhodobě oddělenou evoluční historii obou skupin přitom nezávisle dokládá i porovnání **exon-intronové struktury genů**: v jednom případě lze strukturu všech genů podtřídy odvodit z hypotetického předka s třemi kódujícími exony, v druhém pak měl výchozí (ancestrální) gen exonů osm.

Poslední příklad – rekonstrukci evoluční historie **rostlinných forminů** (Cvrčková et al., 2004b) – zde uvádím také proto, aby se dlouhá řada setkávání s touto genovou rodinou v souvislosti s předchozími tématy dočkala důstojného zakončení. [92] Představuje vlastně svého druhu souhrn a rekapitulaci témat se kterými jsme se již setkali v předchozích dvou příkladech:

- Forminy jsou proteiny se značně **proměnlivou doménovou stavbou**. Všechny sdílejí prakticky jen doménu FH2, proto ke konstrukci fylogenetického stromu použita právě tato doména.
- Dendrogram (Obr.11.3.) člení všechny rostlinné FH2 domény do dvou **evolučně starých kompaktních větví** (třídy I a II) s vysokou bootstrapovou podporou. Podobně dobře podpořená je i rodina kvasinkových forminů, zatímco nepočetní zástupci živočišných forminů zařazení do analýzy jsou roztroušeni po dlouhých větvích nejasného postavení (avšak zjevně vně obou tříd rostlinných i větve kvasinkové). Nedávno byla publikována komplementární analýza rodiny FH2 domén, soustředěná na živočišné proteiny (Higgs a Peterson, 2005); navzdory poněkud diskutabilní metodologii a interpretaci také v ní byla nalezena dobře podpořená větev sdružující kvasinkové FH2 domény.
- Podobně jako u Exo70, i zde lze v rámci obou hlavních tříd rostlinných forminů identifikovat **dobře definované podtřídy** obsahující zástupce z dvouděložných i jednoděložných rostlin (v obrázku označeny malými písmeny). V rámci podtříd však docházelo k dalším **genovým duplikacím**, z nichž některé jsou poměrně nedávné (např. na chromosomu V *Arabidopsis* byl nalezen shluk šesti vzájemně blízce příbuzných genů). Tyto recentní duplikační události znemožňují jednoznačnou identifikaci ortologů nejen mezi jednoděložnými a dvouděložnými rostlinami, ale dokonce i v rámci dvouděložných (*Arabidopsis* – tabák).
- Členění rostlinných forminů na dvě třídy lze opět podpořit i porovnáním **struktury genů** – zatímco forminy třídy I jsou kódovány geny s relativně nízkým počtem exonů (2-6), geny třídy II sestávají z minimálně 11, ale až 24 kódujících exonů (což ovšem výrazně komplikuje predikci struktury mRNA a proteinu).

- Dvě třídy rostlinných forminů, identifikované na základě analýzy sekvencí FH2 domén, se liší také celkovou **doménovou stavbou**: zatímco pro třídu I je charakteristická přítomnost N-terminální membránové kotvy a transmembránové domény (nalezena u 22 z 24 analyzovaných zástupců), forminy třídy II často obsahují dříve neznámou doménu příbuznou antionkogenu Pten (9 ze 16 zástupců; je pozoruhodné, že 5 ze 7 forminů třídy II, které tuto doménu nemají, je členy již zmíněného shluku genů na chromozomu V *Arabidopsis*).

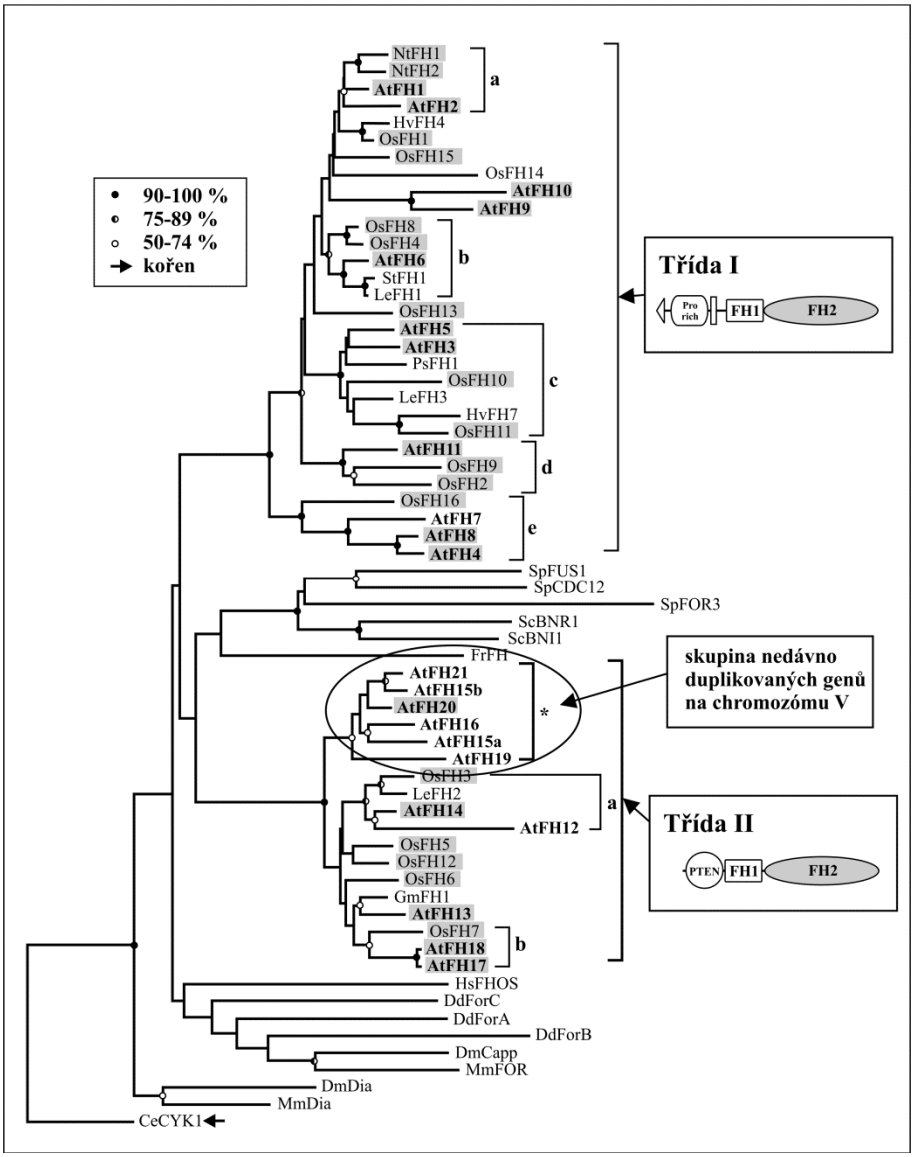
Všechny tři příklady nás vedou k téže obecné otázce: **proč mají** (aspoň krytosemenné) **rostliny v některých genových rodinách tolik různých paralogních genů**, zatímco jiné eukaryotní organismy (včetně nás obratlovců) si vystačí s jednou či nejvýše dvěma kopiemi? Zřejmě to nebude jen důsledek obecné náchylnosti rostlinných genomů k opakovaným genovým duplikacím. U mnoha evolučně konzervovaných genů totiž i rostliny disponují pouze jednou či dvěma variantami, podobně jako ostatní eukaryota. V jedné kopii přitom u rostlin nacházíme i geny kódujících proteiny zřejmě úzce funkčně spjaté s košatými genovými rodinami, například další podjednotky komplexu Exocyst. Ojedinělé duplikace, vedoucí ke vzniku dvou či tří mírně divergovaných kopií, bychom snad mohli svádět na náhodu. Proč je ale u rostlin ne-li preferováno, tedy aspoň tolerováno masivní zmnožení některých konkrétních genů, produkující i víc než dvě desítky variant i v poměrně malém genomu *Arabidopsis*? A proč se tato zdánlivá výjimka z přísné darwinovské ekonomie týká právě těchto genů, a ne jiných?

Pokud jde o výběr genů, jednoho z možných směrů hledání odpovědi jsme se již dotkli v souvislosti s představou buňky jakožto sítě vzájemně propojených modulů (kap. 10.1.): více proměnlivosti můžeme očekávat u molekul, které zprostředkují **komunikaci mezi jednotlivými evolučně konzervovanými signálními drahami**, než u vnitřních prvků těchto drah. Všechny tři diskutované případy by vskutku mohly spadat do této kategorie – Exo70 je možná podjednotkou Exocystu zodpovědnou např. za interakci s cytoskeletem (Wang et al., 2004), PLD produkuje fosfatidovou kyselinu, na kterou můžeme pohlížet jako na buněčného „druhého posla“ ovládajícího např. také procesy buněčné morfogeneze (i u rostlin – viz Potocký et al., 2003), forminy zřejmě zprostředkují integraci řady regulačních vstupů ovlivňujících nukleaci aktinových vláken (viz (Cvrčková et al., 2004a).

Nastíněný směr uvažování však nevysvětluje, **proč ke zmnožení příslušných genů došlo u rostlin**, ale ne u živočichů (o zbytku eukaryot toho totiž zatím moc nevíme). Nabízejí se přinejmenším dvě poněkud spekulativní vysvětlení, která propojuje představa, že zmnožení kopií genů představuje poměrně účinný nástroj k zajištění „jemného ladění“, zajišťujícího optimalizaci funkce v proměnlivém prostředí (Nasmyth et al., 1991):

- Rostliny jakožto **přisedlé organismy** neschopné lokomoce musejí být schopny víceméně normální funkce v daleko širším rozmezí podmínek než organismy, které dovedou nepřízni prostředí utéci.
- Podobnou úvahu lze uplatnit i v mikroměřítku. Rostlinné buňky jsou obklopeny pevnou **buněčnou stěnou**, která je těsně propojena s cytoplazmatickou membránou prostřednictvím mnoha různých transmembránových proteinů; některé z nich zprostředkují i další propojení s cytoskeletem. Proto je zejména pohyb molekul v kortikálních strukturách buňky omezen ve srovnání s mnohem pohyblivějšími buňkami živočišnými. U rostlin lze tedy očekávat mnohem větší heterogenitu nitrobuněčného prostředí především v kortikální vrstvě cytoplasmy, a tedy větší pestrost lokálních podmínek, na něž jsou jednotlivé izoformy proteinů optimalizovány. Lze si představit, že zvláště významnou úlohu v rozrůzňování lokálních podmínek může hrát polární povaha prostředí v pletivech mnohobuněčného organismu, a že

zmnožování genů tedy mohlo jít ruku v ruce s vývojem **mnohobuněčnosti** rostlinného typu. Připomeňme si ostatně, jak významný podíl v evoluci mnohobuněčnosti živočišné zřejmě hrála diverzita molekul buněčných povrchů.



Obr.11.3. Nezakořeněný NJ dendrogram forminů, sestrojený na základě sekvencí konzervativní FH2 domény (podle Cvrčková et al., 2004b). At – *Arabidopsis thaliana*, Ce – *Caenorhabditis elegans*, Dd – *Dictyostelium discoideum*, Dm – *Drosophila melanogaster*, Fr – *Fugu rubripes*, Gm – *Glycine max*, Hs – *Homo sapiens*, Hv – *Hordeum vulgare*, Le – *Lycopersicon esculentum*, Mm – *Mus musculus*, Nt – *Nicotiana tabacum*, Os – *Oryza sativa*, Ps – *Pisum sativum*, Sc – *Saccharomyces cerevisiae*, Sp – *Schizosaccharomyces pombe*, St – *Solanum tuberosum*. U obou podrodin rostlinných forminů je schématicky znázorněno charakteristické uspořádání domén.

Obě hypotézy jsou v principu **testovatelné**, a to jak standardními experimentálními postupy molekulární biologie, tak i pomocí přístupů, které bychom v rámci pojetí zavedeného v 1. kapitole řadili do „vysoké bioinformatiky“. Zajímavé výsledky především ve vztahu ke druhé hypotéze by mohla přinést například důkladná transkriptomická analýza chování početných genových rodin během ontogeneze či v širokém rozmezí kultivačních podmínek; zatím však zřejmě neexistuje software, který by umožnil sledovat osud konkrétních, předem vybraných genů v transkriptomickém experimentu!

Nás jakožto praktické molekulární biology budou v bezprostřední budoucnosti zajímat spíš tradiční **„mokré“ experimentální přístupy** (ke kterým řadím i metody tak sofistikované, jako je analýza exprese vybraných genů pomocí kvantitativní RT-PCR). Výsledky bioinformatických analýz budou zřejmě hrát nezastupitelnou roli v plánování experimentů, a to nejen v poměrně triviálních, avšak důležitých úkolech, jako je návrh primerů pro PCR a klonovacích strategií či identifikace dostupných mutantů ve veřejných sbírkách. U početných genových rodin lze totiž na základě fylogenetické analýzy a veřejně dostupných transkriptomických dat provést třeba předběžný výběr genů, na které se má soustředit experimentální práce. Kritériem může být třeba absence recentních genových duplikací v dané podrodině a z toho pramenící nižší pravděpodobnost redundance se sesterskými geny, popřípadě uspořádání exprese (je s výhodou, pokud se nám podaří najít situaci, v níž je zkoumaný gen jediným exprimovaným příslušníkem rodiny).

Nástroji pro řešení úkolů právě zmíněného druhu by měl být čtenář, který se dopracoval až sem, alespoň v principu vybaven; teď je už řada na něm. Na text, který jste právě dočetli, se můžete dívat jako na turistickou mapu nepříliš podrobného měřítka. Můžete se s ní bez obav pustit do krajiny, jste-li smířeni s tím, že vás krajina občas překvapí – někdy absencí lávky přes bahnitý potůček nebo plotem přehrazujícím cestu, jindy naopak idylickým zákoutím či útulnou hospůdkou. Podobně je tomu i s virtuální „krajinou“ bioinformatických metod: vede přes ni mnoho cest, a leccos z toho, co jste našli zde, se dá udělat i jinak. Leckdy jistě i lépe, nebo aspoň snáze či rychleji. Přeji vám tedy mnoho štěstí v hledání vlastních cestiček!

Praha a Strakonice, podzim 2004 – leden 2005 Fatima Cvrčková

Vytáhl jsem si gen z genomové databáze, analyzoval jsem jeho sekvenci pomocí počítačového programu, odeslal jsem rukopis po internetu a článek mi vyšel v online časopisu. Celá tahle zkušenost ve mně zanechala jakýsi pocit prázdnoty.

12. Slovníček a seznam zkratek

3D-PSSM	metoda vyhledávání trojrozměrných struktur, jimž by mohla být příbuzná uživatelská sekvence – dotaz (6.4.)
accession	viz přístupový kód
alignment	viz přiřazení
BioEdit	program pro manuální konstrukci mnohotného přiřazení, s možností zapracování výsledku automatizovaného přiřazení programem CLUSTAL (8.2.2.)
BLAST	metoda a program pro vyhledávání lokálních podobností mezi uživatelsky zadanou sekvencí – dotazem – a jinou sekvencí či skupinou sekvencí, např. celou databází (*Basic local alignment search tool*) (6.2.)
BLOSUM	viz substituční matice
bootstrapping	v naší souvislosti nejběžnější metoda statistického zhodnocení dendrogramů, založená na porovnání velké série „pseudovzorků“ odvozených od výchozích dat (10.3.)
branch	viz větev
clade	viz větev
CLUSTAL	algoritmus pro automatizovanou konstrukci mnohotného přiřazení (8.2.1.)
Clustal X	volně šiřitelný program pro Windows obsahující implementaci algoritmu CLUSTAL a nástroje pro konstrukci a prezentaci dendrogramu metodou NJ (8.2.1., 10.4.2.)
CONSENSE	program ze sady PHYLIP, určený k výpočtu konsensuálního dendrogramu (10.4.2.)
dendrogram	čili strom - schéma, které vyjadřuje příbuzenské vztahy (v naší souvislosti mezi sekvencemi) tak, že jednotlivé spojnice (větve) propojují příbuzné OTU (zkoumané či hypotetické) s uzly, které představují jejich hypotetické společné předky (10.1.)
distanční metody	metody konstrukce dendrogramu vycházející z matice vzájemných vzdáleností hodnocených OTU (např. sekvencí); každá OTU vystupuje jakožto celek, který je charakterizován jen vzdálenostmi vůči všem ostatním; viz též znakové metody (10.2.)
dotaz	*query* – uživatelsky zadaná sekvence, jejíž příbuzné hledáme v databázi; to, co jsme našli, bývá označováno jako subjekt (*subject*)
FASTA	někdy též FastA; jednak celkem univerzální a jednoduchý formát pro zápis sekvenčních dat (3.1.), jednak nyní již méně běžná, avšak stále užívaná metoda vyhledávání sekvencí v databázích na základě podobnosti (6.1.)
ForCon	program pro konverzi formátů mnohočetného přiřazení, distribuovaný v rámci softwarového balíčku TreeCon (10.4.3.)
GenScan	poměrně spolehlivý a rychlý program pro predikci sestřihu mRNA (9.1.)
GDI	*GDP Dissociation Inhibitor*, kofaktor malých GTPáz evolučně příbuzný REP; zmiňován v řadě příkladů na různých místech v textu
Gonnet	viz substituční matice
HMM	skryté Markovovské modely (*hidden Markov models*), teoretický základ některých algoritmů pro analýzu sekvencí (7.2.)
IUPAC	Mezinárodní unie pro čistou a aplikovanou chemii (*International Union for Pure and Applied Chemistry*); přeneseně též standardizované zkratky pro zápis sekvencí monomerů biologických makromolekul (3.1.), v terminologii programu MACAW také substituční matice používaná při práci se sekvencemi DNA a RNA (viz 5.3. a 8.2.2.)
JTT	viz substituční matice
klad	krom triviálního významu (opak záporu) též *clade* – viz větev
MACAW	poněkud zastaralý, avšak uživatelsky (většinou) velmi příjemný program pro manuální konstrukci mnohotného přiřazení s možností automatizovaného vyhledávání lokální podobnosti sekvencí a statistického zhodnocení (*Multiple alignment construction and analysis workbench*) (8.2.2.)
ME	*minimum evolution*, distanční metoda konstrukce dendrogramu, která se snaží minimalizovat součet délky větví stromu; je poměrně výpočetně náročná a v praxi bývá místo ní užívána spíše metoda NJ, která je její dosti dobrou aproximací (10.2.)
ML	*maximum likelihood*, znaková metoda hledání „nejvěrohodnějšího“ dendrogramu, považovaná za jeden z nejpřesnějších dostupných postupů; její použití však omezuje velká náročnost na strojový čas a výkon počítače (10.2.)
MP	*maximum parsimony*, znaková metoda konstrukce „nejúspornějšího“ dendrogramu založená na minimalizaci počtu mutací nezbytných k dosažení pozorovaného stavu; je velmi rychlá, avšak využívá pouze část dat a mnohdy nevyniká spolehlivostí (10.2.)
NEIGHBOR	program ze sady PHYLIP, určený ke konstrukci dendrogramů metodou NJ (10.4.2.)

NetGene	program pro predikci sestřihu mRNA s možností manuálního výběru sestřihových alternativ a lehce nepříjemným výstupním formátem
Newick	standardní formát zápisu dendrogramů (název podle restaurace, v níž byl vyvinut), někdy označován též jako „závorkový formát New Hampshire“ (*New Hampshire bracket format*; 10.2.)
NJ	*neighbor-joining*, nejčastěji používaná distanční metoda konstrukce dendrogramů (10.2., 10.4.2.)
NJPlot	program pro grafickou prezentaci dendrogramů, který je součástí balíčku Clustal X (10.4.2.)
node	viz uzel
OTU	operační taxonomická jednotka (*Operational Taxonomic Unit*) čili „to, co sedí na konci větví dendrogramu, popřípadě i na jeho uzlech“; v tradiční fylogenetice jsou OTU zpravidla taxony organismů různého systematického postavení, pojem však lze rozšířit tak, aby zahrnoval i jednotlivé geny bez ohledu na organizmální původ (10.1.)
outgroup	neboli „vnějšák“ (podle J. Flegra) – OTU evolučně vzdálená zbytku analyzovaného souboru, která se do analýzy přibírá za účelem stanovení pozice hypotetického společného předka čili „kořene“ stromu (10.1.)
PAM	viz substituční matice
pDRAW32	hezký program na grafickou reprezentaci sekvencí (3.2)
PHYML	program pro konstrukci stromů heuristickou aproximací metody ML, výrazně rychlejší než PHYML a kompatibilní se sadou PHYLIP (10.4.3.)
PHYLIP	balík programů pro rekonstrukci fylogeneze z dat různých typů včetně sekvenčních (*Phylogeny inference package*; 10.4.2.)
PROML	program ze sady PHYLIP, určený ke konstrukci stromů metodou ML (10.4.3.)
PROSITE	jednak databáze konzervovaných motivů v proteinech, jednak standardní formát zápisu sekvenčních motivů, původně zavedený v této databázi (7.1, 7.2.)
PROTDIST	program ze sady PHYLIP, určený k výpočtu matice vzdáleností, která může být dále použita ke konstrukci dendrogramu distančními metodami, jako je NJ nebo UPGMA (10.4.2.)
PROTPARS	program ze sady PHYLIP, určený ke konstrukci dendrogramu metodou MP (10.4.3.)
přiřazení	*alignment*, zápis, v němž jsou vzájemně podobné sekvence uvedeny v řádkách nad sebou tak, aby si pozice umístěné bezprostředně nad sebou pokud možno odpovídaly (v případě potřeby lze do sekvencí zavést mezery); je-li sekvencí v přiřazení více než dvě, hovoříme o mnohotném přiřazení (*multiple alignment*); viz zejména 5.2., 7.1. a 8. kapitola
přístupový kód	*accession*, identifikační kód datového souboru, sdílený mezi databázemi Velké trojky (4.2.)
PSSM	pozičně specifická substituční matice vytvořená např. programem PSI-BLAST (viz 6.2.)
query	viz dotaz
REP	*Rab Escort Protein*, kofaktor malých GTPáz evolučně příbuzný GDI; zmiňován v řadě příkladů na různých místech v textu
Savvy	program pro konstrukci map plazmidů (3.2.)
SEQBOOT	program ze sady PHYLIP, sloužící ke generování bootstrapových vzorků z dané sady dat (10.3.)
SignalP	program k vyhledávání signálních peptidů a membránových kotev (7.4.)
SMART	nástroj k vyhledávání konzervativních domén v proteinových sekvencích, s možností vyhledávání dalších motivů, jako jsou např. signály pro membránovou lokalizaci; též databáze, na níž tento nástroj pracuje (7.3.)
SMS	v naší souvislosti soustava HTML/Java skriptů pro jednoduchou manipulaci se sekvencemi - *Sequence manipulation suite* (3.2.)
SPDBV	program pro vizualizaci trojrozměrných struktur proteinů, který je rovněž klientem pro síťovou službu modelování struktur SwissModel (*SwissProt Database Viewer*)
strom	viz dendrogram
subject	viz dotaz
substituční matice	tabulka, která přiděluje hodnoty (*score*) jednotlivým kombinacím aminokyselin v přiřazení pro potřeby výpočtu míry podobnosti; pro aminokyselinové sekvence se běžně používají matice PAM, JTT, Gonnet a BLOSUM, pro sekvence nukleotidové matice identity čili IUPAC (5.3.)
TMHMM	nástroj k vyhledávání transmembránových segmentů proteinu (7.4.)
TPA	anotace sekvence v databázi poskytnutá „třetí stranou“ – tedy někým, kdo není ani původní autor, ani správce databáze (*third party annotation*)

TreeCon	uživatelsky jednoduchý a rychlý program pro konstrukci dendrogramů distančními metodami a jejich grafickou prezentaci (10.4.2.)
UPGMA	*Unweighted Pair Group Method with Arithmetic mean*, nejjednodušší distanční metoda konstrukce dendrogramu, nevhodná pro sekvence s různou rychlostí evoluce (10.2.)
uzel	*node*, bod spojení větví dendrogramu, který odpovídá hypotetické ancestrální OTU (10.1.)
větev	buďto spojnice dvou uzlů v dendrogramu (*branch*), nebo „větev vyššího řádu“ čili „klad“ (*clade*) zahrnující všechny potomky daného uzlu (10.1.)
WebGene	program k predikci sestřihu mRNA s bohatými možnostmi a poměrně nepříjemně strukturovaným výstupem
znakové metody	metody konstrukce dendrogramu, v nichž je každá OTU charakterizována souborem znaků, které považujeme za nezávislé; tak např. při konstrukci stromu z přiřazení sekvencí vystupuje každá pozice samostatně; viz též distanční metody (10.2.)

13. Literatura

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. a Lipman, D.J. (1990): Basic local alignment search tool. J.Mol.Biol. 215, 403-410.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., et al. (1997): Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389-3402.

Andreeva, A., Howorth, D., Brenner, S.E., Hubbard, T.J., Chothia, C., et al. (2004): SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Res. 32, D226-D229.

Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., et al. (2004): UniProt: the Universal Protein knowledgebase. Nucleic Acids Res. 32, D115-D119.

Attwood, T.K., Blythe, M.J., Flower, D.R., Gaulton, A., Mabey, J.E., et al. (2002): PRINTS and PRINTS-S shed light on protein ancestry. Nucleic Acids Res. 30, 239-241.

Bairoch, A., Boeckmann, B., Ferro, S. a Gasteiger, E. (2004): Swiss-Prot: Juggling between evolution and stability. Briefings in Bioinformatics 5, 39-55.

Baluška, F., Cvrčková, F., Kendrick-Jones, J. a Volkmann, D. (2001): Sink plasmodesmata as gateways for phloem unloading: Myosin VIII and calreticulin as molecular determinants of sink strength? Plant Physiol. 126, 39-46.

Banno, H. a Chua, N.H. (2000): Characterization of the arabidopsis formin-like protein AFH1 and its interacting protein. Plant Cell Physiol. 41, 617-626.

Barton, G.J. (2001): Creation and analysis of protein multiple sequence alignments. In: Bioinformatics. A practical guide to the analysis of genes and proteins., A.D.Baxevanis and B.F.F.Ouelette (ed.), Wiley-Interscience, New York., 215-232.

Basu, M.K. (2001): SeWeR: a customizable and integrated dynamic HTML interface to bioinformatics services. Bioinformatics 17, 577-578.

Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., et al. (2004): The Pfam Protein Families Database. Nucleic Acids Res. 32, D138-D141.

Baxevanis, A.D. (2001a): Bioinformatics and the internet. In: Bioinformatics. A practical guide to the analysis of genes and proteins., A.D. Baxevanis a B.F.F. Ouelette (ed.), Wiley-Interscience, New York., 1-17.

Baxevanis, A.D. (2001b): Predictive methods using DNA sequences. In: Bioinformatics. A practical guide to the analysis of genes and proteins., A.D.Baxevanis a B.F.F.Ouelette (ed.), Wiley-Interscience, New York., 233-252.

Baxevanis, A.D. a Ouelette, B.F.F., ed. (2001): Bioinformatics. A practical guide to the analysis of genes and proteins. Wiley-Interscience, New York.

Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. a Wheeler, D.L. (2004): GenBank: update. Nucleic Acids Res. 32, D23-D26.

Blom, N., Gammeltoft, S. a Brunak, S. (1999): Sequence- and structure-based prediction of eukaryotic protein phosphorylation sites. J.Mol.Biol. 294, 1351-1362.

Brinkman, F.S.L. a Leipe, D.D. (2001): Phylogenetic analysis. In: Bioinformatics. A practical guide to the analysis of genes and proteins., A.D. Baxevanis a B.F.F. Ouelette (ed.), Wiley-Interscience, New York., 323-358.

Brookfield, J.F.Y. (1997): Genetic redundancy. Adv.Genet. 36, 137-155.

Burge, C. (1998): Modeling dependencies in pre-mRNA splicing signals. In: Computational Methods in Molecular Biology, S. Salzberg, et al (ed.), Elsevier Science, Amsterdam., 127-163.

Burge, C. a Karlin, S. (1997): Prediction of complete gene structures in human genomic DNA. J.Mol.Biol. 268, 78-94.

Cennini, C. (1437/1946): Kniha o umění středověku (Il libro dell´ arte) A.D.1437. Překlad F. Topinka. Vladimír Žikeš, Praha.

Cheung, A.Y. a Wu, H. (2004): Overexpression of an Arabidopsis formin stimulates supernumerary actin cable formation from pollen tube cell membrane. Plant Cell 16, 257-269.

Chothia, C. (1992): Proteins. One thousand families for the molecular biologist. Nature 357, 543-544.

Chothia, C. a Lesk, A.M. (1986): The relation between the divergence of sequence and structure in proteins. EMBO J. 5, 823-826.

Corpet, F., Servant, F., Gouzy, J. a Kahn, D. (2000): ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. Nucleic Acids Res. 28, 267-269.

Counsell, D. (2004): The Bioinformatics FAQ. `http://bioinformatics.org/faq/`

Craigon, D.J., James, N., Okyere, J., Higgins, J., Jotham, J., et al. (2004): NASCArrays: a repository for microarray data generated by NASC's transcriptomics service. Nucleic Acids Res. 32, D575-D577.

Csete, M.E. a Doyle, J.C. (2002): Reverse engineering of biological complexity. Science 295, 1664-1668.

Cuff, J.A., Clamp, M.E., Siddiqui, A.S., Finlay, M. a Barton, G.J. (1998): Jpred: a consensus secondary structure prediction server. Bioinformatics 14, 892-893.

Cvrčková, F. (2000): Are plant formins integral membrane proteins? Genome Biology 1, research 001-007.
Cvrčková, F., Bavlnka, B. a Rivero, F. (2004a): Evolutionarily conserved modules in actin nucleation: lessons from Dictyostelium discoideum and plants. Protoplasma 224, 15-31.
Cvrčková, F., Eliáš, M., Hála, M., Obermeyer, G., a Žárský, V. (2001): Small GTPases and conserved signalling pathways in plant cell morphogenesis: from exocytosis to the Exocyst. In: Cell Biology of Plant and Fungal Tip Growth, A. Geitmann, et al (ed.), IOS Press, Amsterdam., 105-122.
Cvrčková, F. a Markoš, A. (2005): Beyond bioinformatics: can similarity be measured in the digital world? J.Biosem., *v tisku*.
Cvrčková, F., Novotný, M., Pícková, D. a Žárský, V. (2004b): Formin homology 2 domains occur in multiple contexts in angiosperms. BMC Genomics 5, 44.
Cvrčková, F. a Žárský, V. (1999): Ntrop1, a Tobacco (Nicotiana tabacum) cDNA encoding a Rho subfamily GTPase expressed in pollen (Accession No. AJ222545). (PGR99-079). Plant Physiol. 120, 634.
Cvrčková, F. (1994): Příspěvek k ekologii elektronického sysla. Vesmír 73, 348-350.
Cvrčková, F., De Virgilio, C., Manser, E., Pringle, J.R. a Nasmyth, K.A. (1995): Ste20-like protein kinases are required for normal localization of cell growth and for cytokinesis in budding yeast. Genes Dev. 9, 1817-1830.
Deeks, M.J., Hussey, P. a Davies, B. (2002): Formins: intermediates in signal transduction cascades that affect cytoskeletal reorganization. Trends Plant Sci. 7, 492-498.
Deleage, G., Blanchet, C. a Geourjon, C. (1997): Protein structure prediction. Implications for the biologist. Biochimie 79, 681-686.
Deutsch, M. a Long, M. (1999): Intron-exon structures of eukaryotic model organisms. Nucleic Acids Res. 27, 3219-3228.
Diella, F., Cameron, S., Gemünd, C., Lindig, R., Via, A., et al. (2004): Phospho.ELM: A database of experimentally verified phosphorylation sites in eukaryotic proteins. BMC Bioinformatics 5, 79.
Edgar, R.C. (2004): MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 32, 1792-1797.
Eliáš, M., Cvrčková, F., Obermeyer, G. a Žárský, V. (2001): Microinjection of guanine nucleotide analogues into lily pollen tubes results in isodiametric tip expansion. Plant Biol. 3, 489-492.
Eliáš, M., Drdová, E., Žiak, D., Bavlnka, B., Hála, M., et al. (2003): The exocyst complex in plants. Cell Biol.Int. 27, 199-201.
Eliáš, M., Potocký, M., Cvrčková, F. a Žárský, V. (2002): Molecular diversity of phospholipase D in angiosperms. BMC Genomics 3, 2.
Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C.J., et al. (2002): The PROSITE database, its status in 2002. Nucleic Acids Res. 30, 235-238.
Favery, B., Chelysheva, L.A., Lebris, M., Jammes, F., Marmagne, A., et al. (2004): Arabidopsis formin AtFH6 is a plasma membrane-associated protein upregulated in giant cells induced by parasitic nematodes. Plant Cell 16, 2529-2540.
Felsenstein, J. (1981): Evolutionary trees from DNA sequences: a maximum likehood approach. J.Mol.Evol. 17, 368-376.
Felsenstein, J. (1989): PHYLIP -- Phylogeny Inference Package (Version 3.2). Cladistics 5, 164-166.
Felsenstein, J. (1996): Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. Methods in Enzymology 266, 418-427.
Felsenstein, J. (2004): Inferring phylogenies. Sunderland, MA.
Fleischmann, R.D., Adams, M.D., White, O., Clayton, R.A., Kirkness, E.F., et al. (1995): Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. Science 269, 496-512.
Galperin, M.Y. (2004): The Molecular Biology Database Collection: 2004 update. Nucleic Acids Res. 32, D3-D22.
Gautier, J., Norbury, C., Lohka, M., Nurse, P. a Maller, J. (1988): Purified maturation-promoting factor contains the product of a Xenopus homolog of the fission yeast cell cycle control gene cdc2+. Cell 54, 433-439.
Gish, W. a States, D.J. (1993): Identification of protein coding regions by database similarity search. Nature Genetics 3, 266-272.
Goffeau, A., Barrell, B.G., Bussey, H., Davis, R.W., Dujon, B., et al. (1996): Life with 6000 genes. Science 274, 563-567.
Guex, N. a Peitsch, M.C. (1997): SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling. Electrophoresis 18, 2714-2723.
Guindon, S. a Gascuel, O. (2003): A simple, fast an accurate algorithm to estimate large phylogenies by maximum likelihood. Syst.Biol. 52, 696-704.
Haas, B.J., Delcher, A.L., Mount, S.M., Wortman, J.R., Smith, R.K.Jr., et al. (2003): Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies. Nucleic Acids Res. 31, 5654-5666.

Haas, B.J., Volfovsky, N., Town, C.D., Troukhan, M., Alexandrov, N., et al. (2002): Full-length messenger RNA sequences greatly improve genome annotation. Genome Biology 3, research0029.1-0029.12.
Hála, M., Eliáš, M., Žárský, V. (2005): A specific feature of the angiosperm Rab escort protein (REP) and evolution of the REP/GDI superfamily. J.Mol.Biol. 348, 1299-1313.
Hall, T.A. (1999): BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT. Nucl.Acids Symp.Ser. 41, 95-98.
Hansen, A. (2001): Bioinformatik: ein Leitfaden für Naturwissenschaftler. Birkhäuser, Basel, Boston, Berlin.
Hartwell, L.H., Hopfield, J.J., Leibler, S. a Murray, A.W. (1999): From molecular to modular cell biology. Nature 402 Supp, C47-C52.
Hebsgaard, S.M., Korning, P.G., Tolstrup, N., Engelbrecht, J., Rouze, P., et al. (1996): Splice site prediction in Arabidopsis thaliana DNA by combining local and global sequence information. Nucleic Acids Res. 24, 3439-3452.
Heger, A. a Holm, L. (2000): Rapid automatic detection and alignment of repeats in protein sequences. Proteins 41, 224-237.
Heinemeyer, T., Wingender, E., Reuter, I., Hermjakob, H., Kel, A.E., et al. (1998): Databases on transcriptional regulation: TRANSFAC, TRRD, and COMPEL. Nucleic Acids Res. 26, 364-370.
Henikoff, J.G., Greene, E.A., Pietrokovski, S. a Henikoff, S. (2000): Increased coverage of protein families with the blocks database servers. Nucleic Acids Res. 28, 228-230.
Higgs, H.N. a Peterson, K.J. (2005): Phylogenetic analysis of the formin homology 2 domain. Mol.Biol.Cell 16, 1-13.
Higo, K., Ugawa, Y., Iwamoto, M. a Korenaga, T. (1999): Plant cis-acting regulatory DNA elements (PLACE) database: 1999. Nucleic Acids Res. 27, 297-300.
Hulo, N., Sigrist, C.J., Le Saux, V., Langendijk-Genevaux, P.S., Bordoli, L., et al. (2004): Recent improvements to the PROSITE database. Nucleic Acids Res. 32, D134-D137.
Ingouff, M., Gerald, J.N.F., Guérin, C., Robert, H., Sørensen, M.B., Van Damme, D., Geelen, D., Blanchoin, L., Berger, F. (2005): Plant formin AtFH5 is an evolutionarily conserved actin nucleator involved in cytokinesis. Nature Cell Biol. 7, 374-380.
Jones, D.T. (1999): Protein secondary structure prediction based on position-specific scoring matrices. J.Mol.Biol. 292, 195-202.
Jongeneel, V., Junier, T., Iseli, Ch., Hofmann, K., a Bucher, P. (1997): INSECT and MOLLUSCS - supercomputing on the cheap. embnet.news 4(3), http://www.hgmp.mrc.ac.uk/embnet.news/vol4_3/insect.html.
Kanehisa, M. (2000): Post-genome Informatics. Oxford University Press, Oxford.
Karplus, K. a Hu, B. (2001): Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. Bioinformatics 17, 713-720.
Kelley, L.A., MacCallum, R.M. a Sternberg, M.J.E. (2000): Enhanced Genome Annotation using Structural Profiles in the Program 3D-PSSM. J.Mol.Biol. 299, 499-520.
Kreegipuu, A., Blom, N. a Brunak, S. (1999): PhosphoBase, a database of phosphorylation sites: release 2.0. Nucleic Acids Res. 27, 237-239.
Krogh, A., Larsson, B., von Heijne, G. a Sonnhammer, E.L. (2001): Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J.Mol.Biol. 305, 567-580.
Kulikova, T., Aldebert, P., Althorpe, N., Baker, W., Bates, K., et al. (2004): The EMBL Nucleotide Sequence Database. Nucleic Acids Res. 32, D27-D30.
Labbe, J.C., Lee, M.G., Nurse, P., Picard, A. a Doree, M. (1988): Activation at M-phase of a protein kinase encoded by a starfish homologue of the cell cycle control gene cdc2+. Nature 335, 251-254.
Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., et al. (1993): Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. Science 262, 208-214.
Leinonen, R., Nardone, F., Oyewole, O., Redaschi, N. a Stoehr, P. (2004): The EMBL sequence version archive. Bioinformatics 19, 1861-1862.
Letunic, I., Goodstadt, L., Dickens, N.J., Doerks, T., Schultz, J., et al. (2002): Recent improvements to the SMART domain-based sequence annotation resource. Nucleic Acids Res. 30, 242-244.
Li, H., Wu, G., Ware, D., Davis, K.R. a Yang, Z. (1998): Arabidopsis Rho-related GTPases: differential gene expression in pollen and polar localization in fission yeast. Plant Physiol. 118, 407-417.
Li, J., Yen, C., Liaw, D., Podsypanina, K., Bose, S., et al. (1997): PTEN, a putative protein tyrosine phosphatase gene mutated in human brain, breast, and prostate cancer. Science 275, 1943-1946.
Liu, X., Fan, K. a Wang, W. (2004): The number of protein folds and their distribution over families in nature. Proteins 54, 491-499.
Lu, X., Zhai, C., Gopalakrishnan, V. a Buchanan, B.G. (2004): Automatic annotation of protein motif function with Gene Ontology terms. BMC Bioinformatics 5, 122.

Lund, O., Nielsen, M., Lundegaard, C., a Worning, P. (2002): CPHmodels 2.0: X3M a Computer Program to Extract 3D Models. CASP5 Conference Proceedings, Abstract A 102.
Marchler-Bauer, A., Anderson, J.B., DeWeese-Scott, C., Fedorova, N.D., Geer, L.Y., et al. (2003): CDD: a curated Entrez database of conserved domain alignments. Nucleic Acids Res. 31, 383-387.
McGinnis, S. a Madden, T.L. (2004): BLAST: at the core of a powerful and diverse set of sequence analysis tools. Nucleic Acids Res. 32, W20-W25.
McGuffin, L.J., Jones, D.T. a Bryson, K. (2000): The PSIPRED protein structure prediction server. Bioinformatics 16, 404-405.
Milanesi, L., D'Angelo, D. a Rogozin, I.B. (1999): GeneBuilder: interactive in silico prediction of genes structure. Bioinformatics 15, 612-621.
Miyazaki, S., Sugawara, H., Ikeo, K., Gojobori, T. a Tateno, Y. (2004): DDBJ in the stream of various biological data. Nucleic Acids Res. 32, D31-D34.
Mulder, N.J., Apweiler, R., Attwood, T.K., Bairoch, A., Barrell, D., et al. (2003): The InterPro Database, 2003 brings increased coverage and new features. Nucleic Acids Res. 31, 315-318.
Nasmyth, K.A., Dirick, L., Surana, U., Amon, A. a Cvrčková, F. (1991): Some facts and thoughts on cell cycle control in yeast. Cold Spring Harb.Symp.Quant.Biol. 56, 9-20.
Nielsen, H. a Krogh, A. (1998): Prediction of signal peptides and signal anchors by a hidden Markov model. In: Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB 6), AnonymousAAAI Press, Menlo Park, California., 122-130.
Notredame, C., Higgins, D. a Heringa, J. (2000): T-Coffee: A novel method for multiple sequence alignments. J.Mol.Biol. 302, 205-217.
Novák, D. (2001): Příspěvek elektronické konference Záhady, podrubrika Psychotronika. http://www.zahady.cz/clanek_detail.php?id=55&r=1.
OOta, S. a Saitou, N. (1999): Phylogenetic relationship of muscle tissue deduced from superimposition of gene trees. Mol.Biol.Evol. 16, 856-867.
Pagni, M., Ioannidis, V., Cerutti, L., Zahn-Zabal, M., Jongeneel, C.V., et al. (2004): MyHits: a new interactive resource for protein annotation and domain identification. Nucleic Acids Res. 32, W332-W335.
Parida, L.A., Floratos, A., a Rigoutsos, I. (1998): MUSCA: An algorithm for constrained alignment of multiple data sequences. Proceedings 9th Workshop on Genome Informatics. Tokyo, Japan, December 1998.
Parida, L.A., Floratos, A. a Rigoutsos, I. (1999): An approximation algorithm for alignment of multiple sequences using motif discovery. J.Combin.Optim. 3, 247-275.
Pearson, W.R. (2000): Flexible sequence similarity searching with the FASTA3 program package. Methods Mol.Biol. 132, 185-219.
Pearson, W.R. (2001): Protein sequence comparison and protein evolution: Tutorial - ISMB2000. http://www.people.virginia.edu/~wrp/papers/ismb2000.pdf.
Pearson, W.R. (2004): Sequence similarity, protein sequence comparison and protein evolution: What BLAST does/why BLAST works. http://www.people.virginia.edu/~wrp/cshl02/pdf/wrp_ecg_l1_04.pdf.
Pearson, W.R. a Lipman, D.J. (1988): Improved tools for biological sequence comparison. Proc.Natl.Acad.Sci.U.S.A. 85, 2444-2448.
Petsko, G.A. (2001): Homologuephobia. Genome Biology 2, comment1002.1.
Potocký, M., Eliáš, M., Profotová, B., Novotná, Z., Valentová, O., et al. (2003): Phosphatidic acid produced by phospholipase D is required for tobacco pollen tube growth. Planta 217, 122-130.
Pratchett, T., Stewart, I., a Cohen, J. (1999/2005): Věda na Zeměploše. Překlad L. Hozák. TALPRESS, Praha. [Původní vydání: The Science of Discworld, Ebury Press, 1999.]
Prestridge, D.S. (1991): SIGNAL SCAN: a computer program that scans DNA sequences for eukaryotic transcriptional elements. CABIOS 7, 203-206.
Quandt, K., Frech, K., Karas, H., Wingender, E. a Werner, T. (1995): MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. Nucleic Acids Res. 23, 4878-4884.
Raes, J. a Van de Peer, Y. (1999): ForCon: a software tool for the conversion of sequence alignments. embnet.news 6, 10-12.
Rechsteiner, M. a Rogers, S.W. (1996): PEST sequences and regulation by proteolysis. Trends Biochem.Sci. 21, 267-271.
Rost, B., Sander, C. a Schneider, R. (1994): PHD - an automatic mail server for protein secondary structure prediction. CABIOS 10, 53-60.
Saitou, N. a Nei, M. (1987): The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol.Biol.Evol. 4, 406-425.
Schuler, G.D., Altschul, S.F. a Lipman, D.J. (1991): A workbench for multiple alignment construction analysis. Prot Struct Funct Genet 9, 180-190.

Schultz, J., Milpetz, F., Bork, P. a Ponting, C. (1998): SMART, a simple modular architecture research tool: Identification of signalling domains. Proc.Natl.Acad.Sci.U.S.A. 95, 5857-5864.
Shi, J., Blundell, T. a Mizuguchi, K. (2001): FUGUE: Sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. J.Mol.Biol. 310, 243-257.
Slack, J.M.W. (1999/2001): O vejcích a vědcích. Téměř pravdivé vyprávění ze života v biologické laboratoři. Překlad F. Cvrčková, Paseka, Praha. [Původní vydání: Egg and Ego. An almost true story of life in the biology lab. Springer-Verlag, New York, 1999.]
Sonnhammer, E.L. a Koonin, E.V. (2002): Orthology, paralogy and proposed classification for paralog subtypes. Trends Genet. 18, 619-620.
Stebbings, L.A. a Mizuguchi, K. (2004): HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database. Nucleic Acids Res. 32, D203-D207.
Stekel, D. (2003): Microarray bioinformatics. Cambridge University Press, Cambridge.
Stothard, P. (2000): The Sequence Manipulation Suite: JavaScript programs for analyzing and formatting protein and DNA sequences. Biotechniques 28, 1102-1104.
Thagard, P. (1996/2001): Úvod do kognitivní vědy - mysl a myšlení. Překlad A. Markoš, Portál, Praha. [Původní vydání: Mind. Introduction to Cognitive Science. MIT Press, Cambridge,MA, 1996]
The Arabidopsis Genome Initiative (2000): Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. Nature 408, 796-815.
The Gene Ontology Consortium (2001): Creating the gene ontology resource: design and implementation. Genome Res. 11, 1425-1433.
Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, F. a Higgins, D.G. (1997): The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Res. 24, 4876-4882.
Thompson, J.D., Higgins, D.G. a Gibson, T.J. (1994): CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22, 4673-4680.
Ueda, T., Matsuda, N., Anai, T., Tsukaya, H., Uchimiya, H., et al. (1996): An Arabidopsis gene isolated by a novel method for detecting genetic interaction in yeast encodes the GDP dissociation inhibitor of Ara4 GTPase. Plant Cell 8, 2079-2091.
Van de Peer, Y. a De Wachter, R. (1994): TREECON for Windows: a software package for the construction and drawing of evolutionary trees for the Microsoft Windows environment. Comput.Appl.Biosci. 10, 569-570.
Vriend, G. (1990): WHAT IF: a molecular modeling and drug design program. J.Mol.Graph. 8, 52-56.
Wang, S., Liu, Y., Adamson, C.L., Valdez, G., Guo, W., et al. (2004): The mammalian Exocyst, a complex required for exocytosis, inhibits tubulin polymerization. J.Biol.Chem. 279, 35958-35966.
Weng, S. (1998): PatMatch, Pattern Matching Software for Saccharomyces Genome Database and Arabidopsis thaliana Database. `http://genome-www2.stanford.edu/cgi-bin/AtDB/PATMATCH/nph-patmatch`.
Womble, D.D. (2000): GCG: The Wisconsin Package of sequence analysis programs. Methods Mol.Biol. 132, 3-22.
Wood, T.C. a Pearson, W.R. (1999): Evolution of protein sequences and structures. J.Mol.Biol. 291, 977-995.
Žárský, V. a Cvrčková, F. (1997): Small GTPases in the morphogenesis of yeast and plant cells. In: Molecular mechanisms of signalling and membrane transport, K.W.A.Wirtz (ed.), Springer-Verlag, Berlin, Heidelberg, New York., 75-88.
Žárský, V., Cvrčková, F., Bischoff, F. a Palme, K. (1997): At-GDI1 from Arabidopsis thaliana encodes a rab-specific GDI that complements the sec19 mutation of Saccharomyces cerevisiae. FEBS Letters 403, 303-308.
Zrzavý, J., Storch, D., a Mihulka, S. (2004): Jak se dělá evoluce. Paseka, Praha.

Ztracené a znovunalezené poznámky pod čarou

[69] V době, kdy tento text píši (prosinec 2004), má nejnovější verze číslo 5.0. Aktuální stav lze zjistit na adrese http://www.arabidopsis.org.

[70] Zmiňovaný server je pozoruhodný tím, že běží na masivně paralelním systému PAPIA (*Parallel Protein Information Analysis*), jehož základem je síť 64 PC propojených podobně, jako tomu bylo u historického systému INSECT (viz pozn. 65).

[71] Zájemci o hlubší vniknutí do problematiky doporučuji některou specializovanější práci; dobrým výchozím bodem je např. Brinkman (2001), velmi podrobným a vyčerpávajícím zdrojem pak Felsenstein (2004).

[72] Poslední požadavek je spíše výrazem zbožného přání – je s podivem, jak nekvalitní dendrogramy bývají občas publikovány i ve velmi slušných časopisech; výjimkou není ani úplná absence statistického hodnocení (viz 10.3.).

[73] Za všechny uveďme příklad z naší laboratoře – důkaz úlohy GTP-vazebných proteinů v řízení polarity pylové láčky (Eliáš et al., 2001).

[74] Teď se dopouštím pratchettovské „lži-dětem“ (Pratchett et al., 1999/2005): o událostech horizontálního přenosu genetické informace samozřejmě nevíme předem, přicházíme na ně často právě tak, že jsou jediným možným vysvětlením, proč výsledky fylogenetické studie nedávají smysl. Standardní způsob, jak si v takovém případě pomoci, pokud zrovna nechceme studovat horizontální přenos, je vyhodit podezřelou sekvenci a všechno přepočítat.

[75] *Die Programme [...] liefern **immer** ein Ergebnis. Der Anwender muß selbst entscheiden, ob dieses auch einen Sinn ergibt.* (Hansen, 2001, s. 40.)

[76] Je chybou nazývat tento formát „Newickovým formátem“. Název totiž pochází od jména restaurace v Durhamu, NH (proto se někdy hovoří též o „formátu New Hampshire“), kde se 24. června 1986 v průběhu jisté konference sešla „neformální komise sedmi molekulárních fylogenetiků“, aby tam pojídajíce humra vymysleli zmíněný způsob zápisu dendrogramů (Felsenstein 2004).

[77] Vzpomenutý příklad výpočtu vůdčího stromu programem CLUSTAL byl právě příkladem algoritmické konstrukce dendrogramu.

[78] MP metoda má několik variant, které se liší v předpokladech – některé připouštějí vratnost mutací, popřípadě opakovaný vznik evolučních novinek, a jiné nikoli.

[79] Jiné vzorkovací metody se liší právě v tomto kroku – např. metoda *jackknife* „losuje“ kratší vzorky bez opakování pozic.

[80] Pokud hledáme dendrogram metodou ML pro přiřazení většího počtu sekvencí (víc než 10), musíme zpravidla slevit.

[81] Kódující sekvence bývají totiž často natolik rozrůzněné, že na úrovni DNA bychom jen obtížně vyráběli výchozí přiřazení.

[82] Anglosasové to dovedou vyjádřit lapidárně: *rubbish in, rubbish out* (a za „*rubbish*“ by se dalo dosadit i kratší slovo).

[83] Plyne z toho také jistá útěcha pro situace, kdy nejsme schopni spolehlivě předpovědět strukturu mRNA. Pokud jsme v některém ze zkoumaných genů zmeškali exon, vznikne v mnohočetném přiřazení mezera a tu tak jako tak vyhodíme; podobně nevystřižený intron se projeví jako inzerce – tedy mezera v ostatních sekvencích přiřazení – se stejným důsledkem.

[84] Konvenčních názvů je víc, pro konstrukci NJ stromu ze sekvenčních dat ale tyto stačí.

[85] U obou programů uživatel musí zadat typ písma pro popis dendrogramu v souboru fontfile. Pravidelnému opakování tohoto úkonu se můžeme vyhnout tím, že kopii jednoho z nabízených písem (font1 až font6) přejmenujeme na fontfile.

[86] Soubor plotfile musíme před otevřením v PovRay opatřit příponou *.pov. Novější verze PoVRay dovedou výstup DRAWTREE a DRAWGRAMu interpretovat jen pokud je přimějeme, aby emulovaly verzi 3.1, takže do souboru musíme navíc jako první řádek přidat instrukci #version 3.1; (i se středníkem na konci).

[87] Výstup se zapisuje do stejného adresáře, ve kterém byla vstupní data, není tedy dobře spouštět program s daty na CD či v adresáři, který je pouze pro čtení.

[88] Windows XP již nenabízejí ovladače pro „obecnou postscriptovou tiskárnu“, musíme tedy vybrat tiskárnu konkrétní (osvědčil se např. HP LaserJet 6P/6MP PostScript).

[89] Pozor, tyto programy nesmíme zaměnit s pouze zdánlivě příbuznými DNAMLK a PROMLK, které hledají ML stromy za předpokladu konstantní rychlosti evoluce všech větví („molekulárních hodin“) a jsou vhodné pouze pro velmi speciální případy (např. analýza nekódujících sekvencí ve velmi rychle se vyvíjejících virových izolátech).

[90] Úplný datový soubor je k dispozici na http://kfrserver.natur.cuni.cz/bioinfo.html (modelová data pro úlohu 10 kursu PřF UK – ručně optimalizované přiřazení bez mezer). Zde jsou pro přehlednost místo názvů sekvencí použity jednopísmenné symboly: **A** (AtGDI1), **B** (AtGDI3), **C** (CeGDI1), **D** (HsGDI1), **E** (SpGDI1), **F** (ScGDI1), **G** (ScMRS6), **H** (AtREP1), **I** (HsREP2), **J** (CeY67D2). Sekvence A až F patří do podrodiny GDI, G až J do podrodiny REP.

[91] Setkáváme se s ním v tomto textu již podruhé – v části 9.1.1. byla již diskutována poměrně rozsáhlá revize predikované struktury některých genů, která konstrukci dendrogramů předcházela.

[92] Příslušníci rodiny forminů vystupovali mimo jiné v příkladech prohledávacích algoritmů v kap.6.2. a 6.4., analýz doménové struktury (7.3., 7.4., 7.5.), i v příkladech konstrukce a interpretace mnohočetných přiřazení v 8. kapitole (8.3., 8.4. a 8.5.). Jeden ze členů této genové rodiny dodal vybraně zapeklitý případ revize genové struktury v části 9.1.1., a jiný – bez nějakých zvláštních důvodů, prostě proto, že byl zrovna po ruce – „stojí modelem“ v jednom z prvních příkladů datových formátů (Obr.3.2.).