



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

IDENTIFIKACE PROTEINOVÝCH TUNELŮ S VYUŽITÍM MOLEKULÁRNÍCH DYNAMIK

IDENTIFICATION OF THE PROTEIN TUNNELS USING MOLECULAR DYNAMICS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR KOHOUT

VEDOUcí PRÁCE

SUPERVISOR

Ing. MILOŠ MUSIL, Ph.D.

BRNO 2022

Zadání diplomové práce



Student: **Kohout Petr, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Bioinformatika a biocomputing
Název: **Identifikace proteinových tunelů s využitím molekulárních dynamik**
Identification of the Protein Tunnels Using Molecular Dynamics
Kategorie: Bioinformatika
Zadání:

1. Prostudujte problematiku tunelů, molekulárních dynamik a nástroje Caver.
2. Naučte se základní nastavování, spouštění a zpracování molekulárních dynamik s využitím software Amber.
3. Upravte kód výpočetního jádra tak, aby dokázalo analyzovat tunely na více proteinových strukturách.
4. Upravte kód výpočetního jádra tak, aby bylo možné efektivně spouštět jednotlivé výpočty na superpočítači.
5. Navrhněte nové uživatelské rozhraní s důrazem na jednoduchost použití.
6. Implementujte nové uživatelské rozhraní v technologii React.

Literatura:

- Chovancova E, et al. CAVER3.0: a tool for the analysis of transport pathways in dynamic protein structures. PLOS computational biology. 2012, e1002708.
- Jurcik A, et al. CAVER Analyst 2.0: analysis and visualization of channels and tunnels in protein structures and molecular dynamics trajectories. Bioinformatics. 2018, 34, 3586-3588.
- Stourac J, et al. Caver Web 1.0: identification of tunnels and channels in proteins and analysis of ligand transport. Nucleic Acids Research, 2019, 47, 414-422.
- Vavra O, et al. CaverDock: a molecular docking-based tool to analyse ligand transport through protein tunnels and channels.
- Pinto G, et al. Fast screening of inhibitor binding/unbinding using novel software tool CaverDock. Frontiers in Chemistry. 2019, 7, 709.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Musil Miloš, Ing., Ph.D.**
Konzultant: Štourač Jan, LL
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 18. května 2022
Datum schválení: 11. října 2021

Abstrakt

Tato diplomová práce se zabývá analýzou proteinových struktur. Cílem je navrhnout Caver Web 2.0 – novou verzi webové aplikace, která začlení další vědecké nástroje a uživatelům umožní projít komplikovaný pracovní protokol za poskytnutí relevantních výsledků bez nutnosti hlubší znalosti integrovaných nástrojů. Vše bude zprostředkováno prostřednictvím jednoduchého a interaktivního uživatelského rozhraní. Aplikace rozšiřuje původní aplikaci Caver Web 1.0 o nové vlastnosti. Caver Web 1.0 je webový server vhodný pro identifikaci proteinových tunelů a kanálů, pro které umožňuje spustit analýzy transportu ligandů. Program se vyznačuje intuitivním a uživatelsky přívětivým rozhraním s minimem požadovaných vstupů od uživatele. Server je vhodný i pro výzkumníky bez pokročilých bioinformatických nebo technických znalostí. Jeho současná verze je ve vědecké komunitě dobře zavedená a velmi využívaná (35 000 dokončených výpočtů během dvou let provozu). Nejvýznamnějším omezením současné verze je možnost analyzovat pouze statickou strukturu, což často poskytuje neúplný biologický obraz. Proto bylo rozhodnuto o rozšíření nástroje o výpočet molekulárních dynamik, které poskytnou ucelený obraz na proměny proteinových struktur.

Abstract

This thesis focuses on the analysis of protein structures. The aim is to design Caver Web 2.0 – a new version of the web application that integrates additional scientific tools and allows users to go through a complicated workflow to provide relevant results without the need for a deeper knowledge of the integrated tools. Everything will be delivered through a simple and interactive user interface. The application extends the original Caver Web 1.0 application with new features. Caver Web 1.0 is a web server suitable for identifying protein tunnels and channels for which it allows to run ligand transport analyses. The program is characterized by an intuitive and user-friendly interface with minimum required input from the user. The server is suitable for researchers without advanced bioinformatics or technical knowledge. Its current version is well established and highly used in the scientific community (35,000 completed calculations in two years of operation). The most significant limitation of the current version is the ability to analyze only static structure, which often provides an incomplete biological picture. Therefore, it was decided to extend the tool to calculate molecular dynamics to provide a comprehensive picture of protein structure changes.

Klíčová slova

bioinformatika, webová aplikace, proteinové inženýrství, molekulární dynamiky, proteinové tunely, dokování ligandu

Keywords

bioinformatics, web application, protein engineering, molecular dynamics, protein tunnels, ligand docking

Citace

KOHOUT, Petr. *Identifikace proteinových tunelů s využitím molekulárních dynamik*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Miloš Musil, Ph.D.

Identifikace proteinových tunelů s využitím molekulárních dynamik

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Miloše Musila, Ph.D. Další informace mi poskytl Mgr. Jan Štourač. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Petr Kohout
16. května 2022

Poděkování

Za odborné vedení mé diplomové práce bych chtěl poděkovat Mgr. Janu Štourači, který mi kdykoli poskytl konzultace, zodpověděl mé dotazy a pomohl mi s každým problémem, se kterým jsem se setkal. Dále bych chtěl poděkovat Ing. Miloši Musilovi, Ph.D. za vedení mé diplomové práce v rámci fakulty. Za podporu, příjemné pracovní prostředí a podnětné připomínky chci poděkovat celému týmu Loschmidtových laboratoří.

Obsah

1	Úvod	2
2	Proteiny	4
2.1	Struktura proteinů	4
2.2	Funkční skupiny proteinů	5
2.3	Enzymy	6
2.4	Proteinové tunely a kanály	6
3	Molekulární dynamika	10
3.1	Kroky simulace	10
3.2	Silová pole (force field)	12
3.3	Pohyb částic	12
3.4	Výběr souborů a termostatů	16
3.5	Běh simulace	16
3.6	Aproximativní metody	17
4	Nástroj CAVER a CaverDock	19
4.1	CAVER – Nástroj pro analýzu transportních tunelů	19
4.2	CaverDock – Nástroj pro analýzu transportu ligandů	23
5	Caver Web 1.0	26
6	Návrh nové verze nástroje Caver Web	31
6.1	Existující řešení	31
6.2	Návrh nového workflow	31
6.3	Návrh rozšíření výpočetního jádra	34
6.4	Návrh nového uživatelského rozhraní	35
7	Implementace nástroje Caver Web	45
7.1	Architektura aplikace	45
7.2	Výpočetní jádro	45
7.3	Webová aplikace	49
7.4	Nasazení	52
8	Závěr	54
	Literatura	56

Kapitola 1

Úvod

Již od 18. století zažívá náš druh výrazný a poměrně rychlý technologický pokrok, který mu umožňuje prozkoumávat stále konkrétnější oblasti bezprostředně spjaté s naším každodenním životem. Jednou z těchto oblastí je vědomí o tom, jak funguje naše tělo a život kolem nás. S rozvojem lidské společnosti a technologií se vyvinuly metody, které odhalily a umožnily analyzovat nejzákladnější stavební prvky našeho těla včetně jejich funkcí.

Jednou z klíčových složek živých organismů na planetě jsou makromolekulární sloučeniny zvané proteiny, které v nich zastávají širokou plejádu důležitých funkcí. Slouží například jako stavební prvky, transportéry malých molekul, zásobárny nebo se podílí na imunitním systému či katalýze biochemických procesů. Nepřekvapí proto, že proteiny můžeme najít ve všech orgánech a částech lidského těla. Lze je nalézt ve vlasech (keratin) a svalech (aktin), kde ovlivňují náš vzhled a umožňují motoriku, nebo v mozku (myelin), kde zajišťují správné a rychlé vedení nervových vzruchů. Z tohoto důvodu jsou proteiny již dlouhou dobu v popředí zájmu vědců. Vědomosti získané zmapováním vlastností a interakcí mezi proteiny nám slouží nejenom k lepšímu pochopení fungování organismu, ale lze je využít též při tvorbě nových proteinů. Nové proteiny by mohly být aplikovány ve velkém množství odvětví lidské činnosti. Nejčastějším příkladem zájmových oblastí tzv. proteinových inženýrů je oblast vývoje nových léčiv či zvýšení produkce průmyslově využívaných proteinů.

Sektor proteinového inženýrství nabízí prostor pro aplikaci výpočetní techniky, která umožňuje značně urychlit a zlevnit samotný vývoj a studium vlastností proteinů. Důležitými funkcemi, které výpočetní technika poskytuje, jsou smysluplné reprezentace dat vědecké komunitě, umožnění základních i pokročilých analýz dat a zprostředkování predikce vlivu změn na budoucí vlastnosti látek. Dalším příkladem aplikace je výpočet molekulární dynamiky, které tak zpřístupňují jinak těžko v čase pozorovatelné děje na úrovni atomů. Vytvořený model navíc lépe prezentuje modelovaný systém a umožňuje přesnější analýzu proteinů.

Těchto užitečných funkcí, zobrazení dat, výpočtů a simulací lze docílit používáním specializovaných programů. Jejich používání však klade nové požadavky na proteinové inženýry, protože jednotlivé programy očekávají odlišné vstupní formáty a disponují celou řadou konfiguračních parametrů, které mohou mít kritický vliv na biologickou relevanci výsledků. Takže pro správné nastavení parametrů je nutné hlouběji studovat tyto nástroje a datové formáty, které používají. Každý program navíc většinou poskytuje pouze parciální funkci, takže je nutné je vhodně řetězit a kombinovat – například při provedení libovolné analýzy je potřeba získaná data vizualizovat dalším vhodným softwarem.

Představme si vědce – proteinového inženýra, který je expert v doméně strukturální biologie, a jeho činností je studium struktury proteinů. Vědec chce studovat enzym *e*, který

při vytvořených laboratorních podmínkách neprovádí předpokládanou katalytickou činnost v takovém rozsahu, v které by ji očekával. Disponuje protokoly z laboratoří, které udávají, jaké látky se v roztoku vyskytují – například látky: a , b a c , přičemž látka a je katalyzována enzymem e na látku b . Vědec z provedených experimentů předpokládá, že látka c může nějakým způsobem blokovat aktivitu enzymu, a chce toto tvrzení prověřit pomocí počítačových programů. Nejprve musí provést rešerši ohledně existujících nástrojů. Například pro zobrazení dat může využít nástroj PyMOL [20]. Dále ví, že daný enzym má své aktivní centrum uvnitř struktury, a proto chce studovat celou cestu, která do místa vede. Pro identifikaci takové cesty může použít nástroj Caver [14]. Současně ale ví, že kmitání proteinu může způsobit změnu vlastností cest vedoucích do jeho zkoumaného místa, a proto použije nástroj pro simulaci tohoto kmitání – Yasara [31]. Na závěr chce provést analýzu, jestli látky doputují až do aktivního místa, nebo se do něj nedostanou v důsledku působení látky c . K této analýze využije například nástroj CaverDock [55]. Po nalezení a instalaci začne jednotlivé programy spouštět a vyvstanou mu nové otázky a problémy typu: (i) neexistuje lepší reprezentace této struktury?; (ii) jak vybrat vhodný bod pro výpočet tunelu a kde leží samotné aktivní místo?; (iii) kde naleznou popis látek v roztoku a jak je předám nástroji? (iv) jak mohu spustit výpočet kmitání proteinu tak, abych mohl na svém stroji pracovat i na dalších problémech?

Vědec zjistil, že pro řešení jeho otázky je potřeba použít poměrně mnoho programů, které musí spouštět jeden po druhém. Svého záměru však nakonec po dlouhé analýze získaných výsledků dosáhl – zjistil, že látka c se váže v jediné přístupové cestě k aktivnímu místu a výrazně tak snižuje jeho aktivitu.

Podobné postupy a související problémy denně řeší i kolegové z Loschmidtových laboratoří, a proto vznikla myšlenka na vytvoření aplikace, která tuto činnost zautomatizuje a nebude po uživateli požadovat odborné znalosti používaných nástrojů. Na základě této myšlenky vznikla webová aplikace Caver Web, která se rychle stala populární napříč světovou vědeckou komunitou.

Cílem této diplomové práce je vývoj nové verze webové aplikace Caver Web [49], která poskytuje snadný přístup a ovládání bioinformatických nástrojů za účelem analýzy proteinových tunelů a kanálů. Skrze přívětivé uživatelské prostředí umožňuje smysluplně nastavit parametry těchto specializovaných nástrojů bez nutnosti jejich podrobné znalosti. Hlavním přínosem nové verze aplikace je jednoduché nastavení a zpřístupnění molekulárních dynamik, které do analýzy přidávají dosud chybějící hledisko přirozené dynamické povahy proteinů a díky kterým se výrazně zvýší důvěryhodnost a relevance nalezených tunelů.

V rámci diplomové práce bude uvedena a popsána problematika oblastí analýzy proteinových tunelů za využití molekulárních dynamik. V kapitole 2 bude vysvětlena problematika proteinů a proteinových tunelů. Bude popsána jejich struktura, složení a funkce. Pozornost bude soustředěna na katalytickou funkční skupinu proteinů (tzv. enzymy), především na detekci vnitřních prostor (aktivních míst) a cest (tunelů) vedoucích do těchto vnitroproteinových míst. V následující kapitole 3 bude představena molekulární dynamika. Bude vytyčena terminologie používaná v této oblasti. Poté budou popsány jednotlivé kroky simulace. Bude uvedena problematika silových polí a pohybu částic, u které budou představeny základní metody používané pro integraci pohybových rovnic. Závěrem kapitoly bude popsána volba ideální délky kroku simulace. Následně budou v kapitole 4 popsány nástroje CAVER a Caver Web. V kapitole 5 bude představena současná verze rozšiřované aplikace. V kapitolách 6 a 7 bude uveden návrh (včetně průzkumu existujících řešení) a implementace aplikace.

Kapitola 2

Proteiny

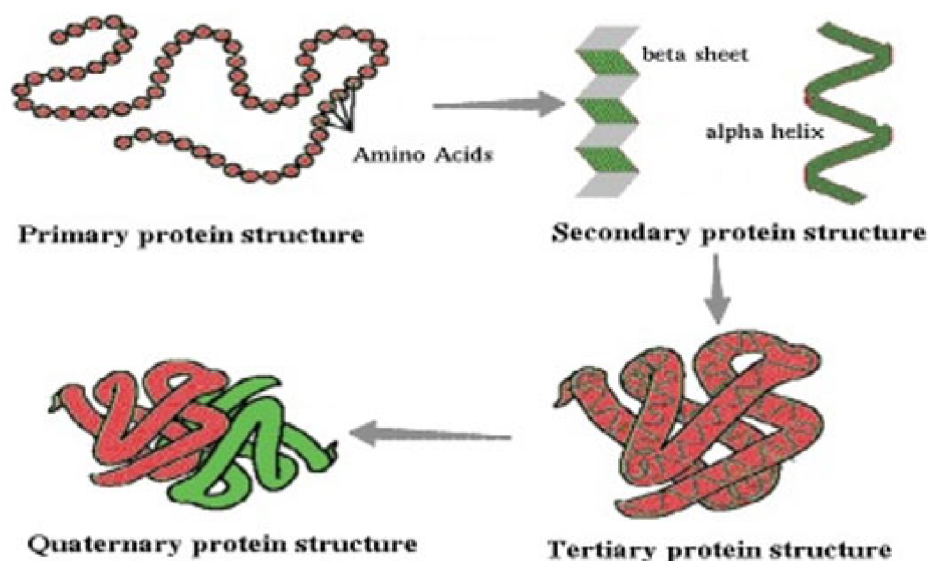
Proteiny jsou nedílnou součástí všech živých organismů na Zemi a tvoří většinu jejich pevné hmoty. Tato kapitola si klade za cíl uvést do problematiky proteinů. Bude vysvětlena základní struktura těchto makromolekul včetně principu jejich formování do složitých prostorových útvarů zvaných konformace. Dále budou uvedeny funkční skupiny proteinů, do kterých se proteiny řadí podle funkce, jíž v buňce zastávají. Poté bude detailně popsána katalytická skupina, u které se ve velké míře vyskytují tzv. aktivní místa, a proteinové tunely, které budou podrobně vysvětleny v závěru kapitoly. Zdrojem informací pro sepsání kapitoly, pokud není v textu uvedeno jinak, bylo využito literatury [6].

2.1 Struktura proteinů

Proteiny (také označované jako bílkoviny) jsou makromolekulární molekuly složené z jednoho, nebo více polypeptidových řetězců. Základní složkou polypeptidového řetězce jsou aminokyseliny. Tyto aminokyseliny se zřetěžují pomocí kovalentní peptidové vazby a jejich sekvence jednoznačně určuje konkrétní bílkovinu.

Sekvenci aminokyselin označujeme jako *primární* strukturu proteinu. Každá z bílkovinných aminokyselin se vyznačuje svými specifickými vlastnostmi. Při pozorování zaujímání prostorové konformace (uspořádání) nás především zajímá jejich (ne)polární a hydrofilní, respektive hydrofobní, charakter, který společně s (k) natáčením (benevolentní) polypeptidovou vazbou mezi aminokyselinami umožňuje svinutí tzv. polypeptidové kostry proteinu do 3D struktur (viz obrázek 2.1). Formování těchto struktur je zapříčiněno působením nekovalentních vazeb, které působí mezi vodíky (vodíkové můstky), ionty (iontové vazby) a mezi molekulami (van der Waalovy síly). Pokud studujeme prostorové uspořádání v části sekvence bílkoviny, hovoříme o *sekundární* struktuře proteinu a můžeme identifikovat dva prostorové útvary: alfa šroubovici (α -helix) a skládaný list (β -sheet). V případě analýzy celé sekvence proteinu hovoříme o *terciální* struktuře. Za *kvarterní* strukturu označujeme shluk dvou a více polypeptidových řetězců, které společně tvoří jednotný funkční celek (tzv. multimer) [61].

Unikátní primární struktura proteinů s působením sil nekovalentních vazeb předurčuje i unikátní prostorovou konformaci, která předurčuje jeho funkci.



Obrázek 2.1: Příklady primární, sekundární, terciální a kvartérní struktury proteinů (převzato z [18]).

2.2 Funkční skupiny proteinů

Jak již bylo zmíněno výše, proteiny jsou základními stavebními kameny živých organismů a plní v nich velké množství funkcí. Z tohoto důvodu se bílkoviny dělí do níže vypsanych skupin podle úlohy, kterou v organismu zajišťují. Poslední skupina proteinů bude podrobněji popsána v následující části této kapitoly.

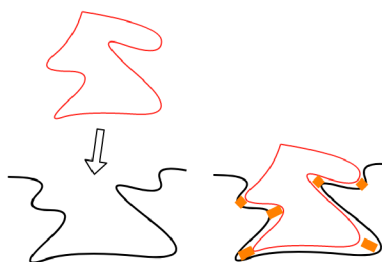
- Skupina **regulačních** bílkovin svým působením kontroluje expresi genů, čímž umožňuje organismu ovládat procesy, které se uvnitř něho dějí. Příkladem jsou bílkoviny, které ovlivňují tvorbu *laktázy*. Ta štěpí laktózu na sacharidy, které se dále zpracovávají v organismu za účelem zisku energie.
- Klasickým zástupcem **receptorových** proteinů, tedy proteinů zpracovávajících signály z vnějšího i vnitřního prostředí, je *rhodopsin*, který reaguje na světlo dopadající na sítnici.
- Skupina **signálních** bílkovin zajišťuje přenos signálu k zástupcům receptorové skupiny. Příkladem je *insulin*, jenž reguluje množství *glukosy* v těle.
- **Zásobní** proteiny ukládají zásoby látek pro budoucí zpracování. Zástupcem je například *ferritin*, který ukládá železo v játrech.
- Proteiny, které zajišťují pohyb organismu, označujeme za proteiny **pohybové**. Příkladem je *myosin*. Ten umožňuje přeměňovat energii v ATP na cíleně směřovaný pohyb v buňce.
- Další neméně důležitou skupinu tvoří proteiny **transportní**, které zajišťují přenos látek z jednoho místa na druhé. Příkladem je *hemoglobin*, jenž zásobuje organismus kyslíkem.

- **Strukturní** bílkoviny tvoří hmotu, ze které jsou sestaveny buňky a tkáně. Ve vlasech a nehtech se nalézá zřejmě nejznámější zástupce – *keratin*.
- Poslední skupinu tvoří katalytické proteiny známé také jako **enzymy**. Jejich specifickou vlastností je schopnost rušení a vytváření nových kovalentních vazeb. Příkladem enzymu je již výše zmíněný protein *laktázy*, který štěpí mléčný cukr laktózu na jednodušší a pro tělo zpracovatelnou glukózu.

2.3 Enzymy

Enzymy jsou nedílnou součástí každé buňky, která jich obsahuje velké množství. V buňce plní velice užitečnou funkci katalyzátorů chemických reakcí, kdy snižují potřebnou *aktivační energii*. Bez jejich přítomnosti by většina chemických dějů v buňce neprobíhala.

Toto prostorové uspořádání při detailnějším pohledu odkrývá nesčetné množství záhybů, výčnělků a kanálů, přičemž každý je tvořen specifickým souborem aminokyselin. Jejich soubory umožňují navázání ligandu vytvořením dostatečného množství nekovalentních vazeb pouze těm ligandům, které svým tvarem přesně odpovídají tvaru povrchu proteinu a jejich složení vytváří příslušné nekovalentní interakce. Tento princip je známý pod pojmem *ruka a rukavice* (nebo také princip *klíče a zámku*) a je vyobrazen na obrázku 2.2.



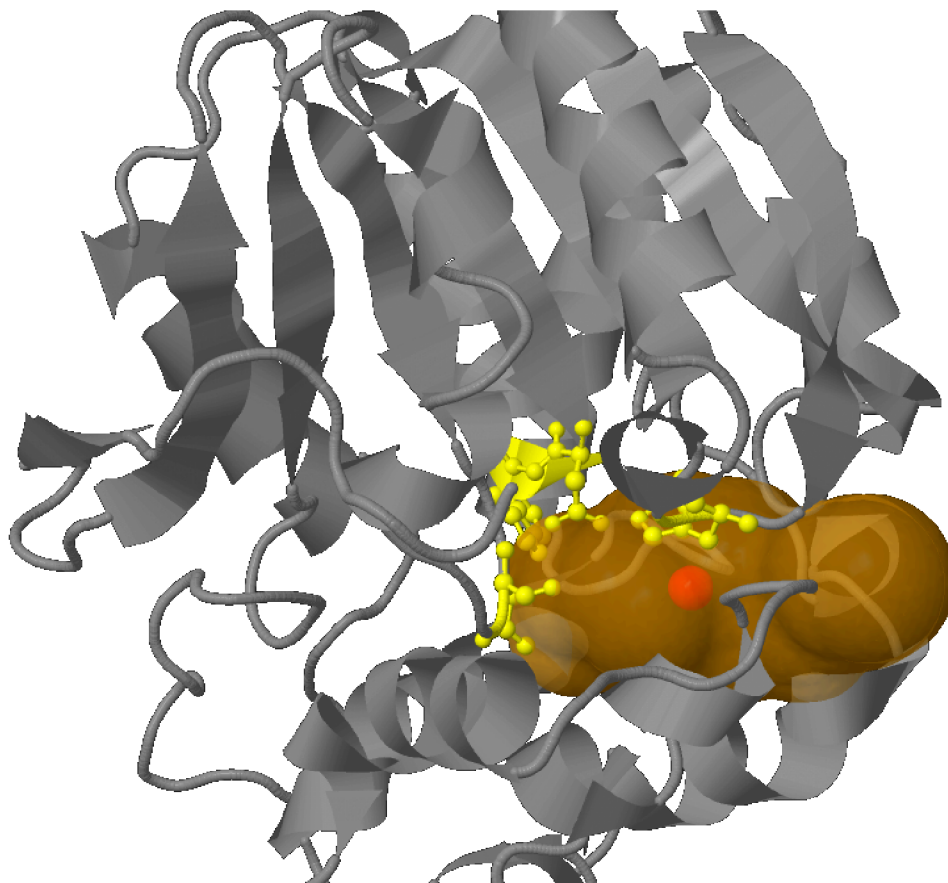
Obrázek 2.2: Znázornění vázání specifického ligandu na povrchu proteinu (*ruka a rukavice*)

Enzymy váží jeden, nebo více ligandů zvaných *substráty*. Substrát vázaný v tzv. *aktivním místě* (obrázek 2.3) přeměňují při tvorbě nebo rušení kovalentních vazeb na produkt. Po dokončení reakce je produkt díky pozměněné struktuře, která již nesplňuje podmínky pro vázání v aktivním místě, uvolněn do prostředí.

2.4 Proteinové tunely a kanály

Ve všech skupinách proteinů můžeme nalézt spousty zákrutů, otvorů a dlouhých slepých chodeb, které se nazývají *tunely*. Problematika těchto tunelů je důkladně studována především ve spojitosti s enzymy, které si v milionech let trvající evoluci vyvinuly a zformovaly svá aktivní místa právě na konci těchto slepých chodeb, aby byla chráněna před vlivem vnějšího prostředí. Informace pro tuto část kapitoly jsem čerpal z literatury [51].

Potenciálním důvodem přenosu aktivního místa do vnitra enzymů může být jejich snaha o plnou kontrolu okolí aktivního místa, která by na povrchu mohla být ovlivněna různými rozpouštědly z okolního prostředí. Aktivní místa jsou spojena s povrchem *tunely*, které zprostředkovávají výměnu látek mezi aktivním místem a vnějším prostředím a rozšiřují tak model *zámku a klíče* o prvek klíčové dírky. *Klíčová dírka* je v tomto případě tvořená tunelem určitého tvaru a její složení předurčuje enzymatickou funkci celého enzymu.



[Smol]

Obrázek 2.3: Příklad aktivního místa (oranžová kapsa) v proteinu *haloalkandehalogenázy*, který odbourává látku znečišťující životní prostředí [35]. V obrázku jsou (žlutě) vyznačeny tzv. *katalytická rezidua*, která umožňují katalýzu substrátu. Obrázek byl vytvořen prostřednictvím aplikace Caver Web [49].

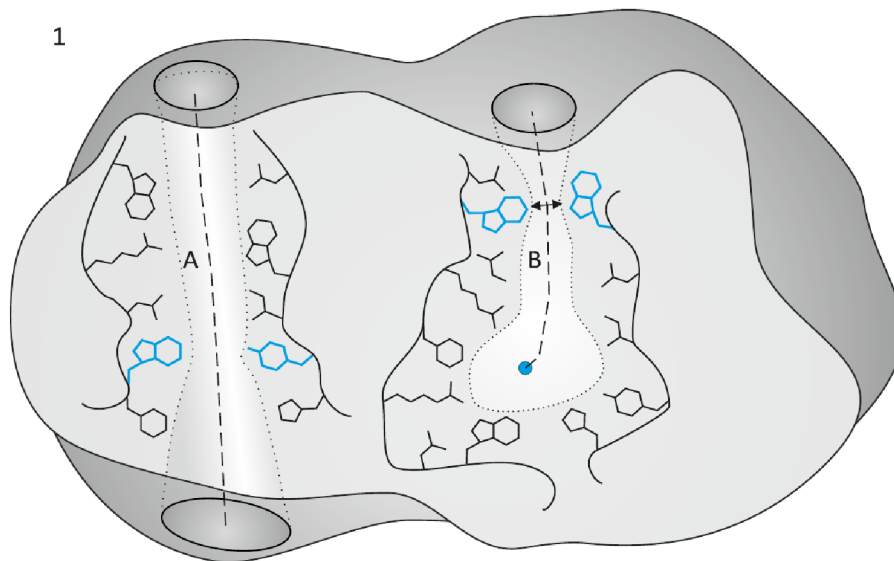
V literatuře se lze setkat s mnoha výklady termínu *tunel* (*tunnel*) a *kanál* (*channel*). V rámci výše zmíněné literatury a dále i v rámci celé diplomové práce pojem *tunel* označuje spojení mezi aktivním místem umístěným uvnitř enzymu a jeho povrchem nebo spojení mezi aktivními místy uvnitř enzymu či v proteinovém komplexu (multimeru). Pojem *kanál* označuje spojení různých částí povrchu proteinu, kterým může látka projít bez nutnosti transformace (obrázek 2.4).

Nejužší místo tunelu se nazývá *hrdlo*. Hrdlo tunelu je často ovládáno tzv. *bránou*, která toto místo může ještě více zužovat (uzavírat) a učinit tak tunel neprůchozí. Brána často otevírá tunel v určitém časovém intervalu. To znamená, že nabízí další regulační mechanismus pro přístup látek do aktivního místa [51].

Permanentní a přechodné tunely

Většina proteinů při prvotní analýze působí dojmem, že obsahuje pouze jeden *permanentní tunel*, který se v čase neuzavírá. Pokud bychom však tyto proteiny podrobili dlouhodobému pozorování s možností identifikace tunelů v reálném čase, zjistili bychom, že některé proteiny disponují dalšími otevřenými tunely jen v určitých okamžicích. Tunely, které jsou identifikovány jen ve vybraném čase, se nazývají *přechodné* (*transient*) (viz obrázek 2.5).

1



Obrázek 2.4: Proteinový kanál (A) a tunel (B) (převzato z [11]).

Otázka, která může vyvstat ve spojitosti s pojmy otevřených a zavřených tunelů, je požadavek na minimální průměr tunelu, aby byl klasifikován jako otevřený. Podle výše uvedené literatury [51] je za otevřený tunel považován takový tunel, který disponuje šířkou hrdla větší než 1.4 Å.

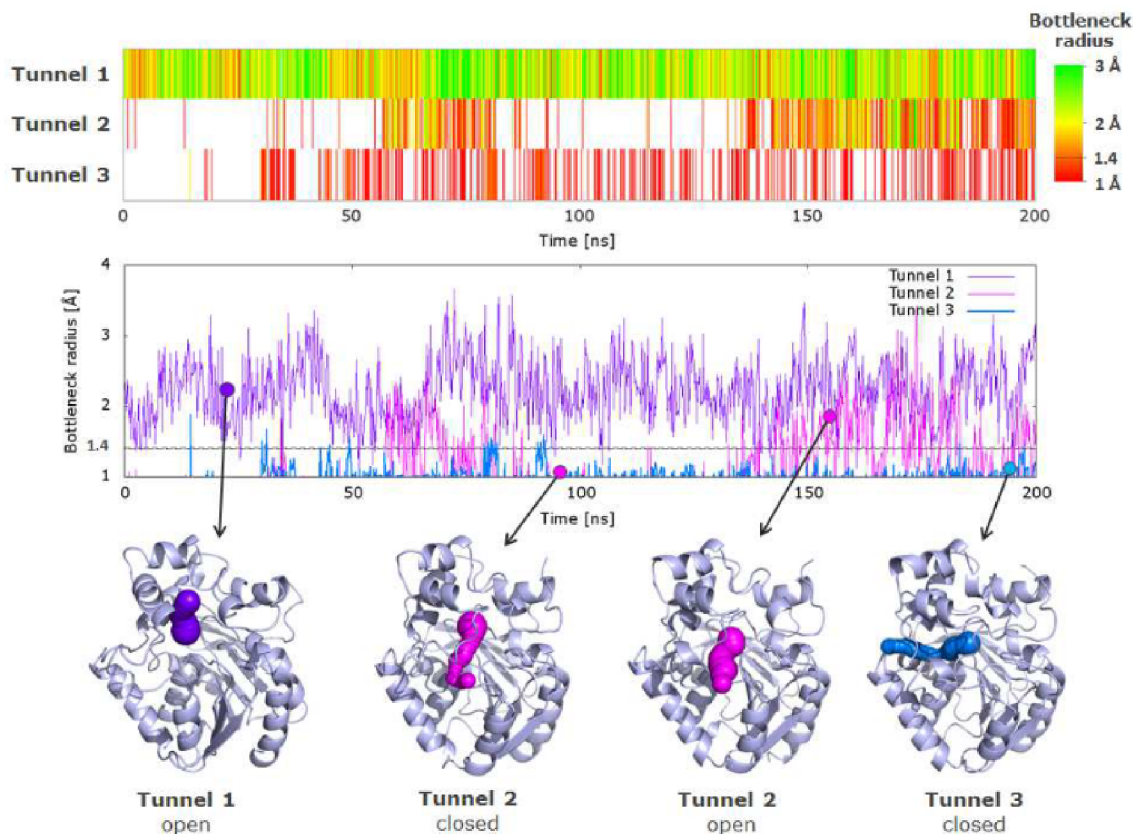
Jelikož v současnosti existují v naprosté většině záznamy pouze o statické podobě proteinů, je nutné pro sledování změn proteinové struktury v čase využít počítačové programy. Pro sledování změn struktury proteinů v čase se využívá výpočtu *molekulárních dynamik*, které budou podrobněji popsány v kapitole 3.

2.4.1 Experimentální metody

V současnosti neexistuje žádná metoda schopná přímo experimentálně studovat tunely a kanály. Informace o jejich vlastnostech se tak musí získávat pouze zprostředkovaně, což limituje jejich přímou využitelnost. Proto se tunely nejčastěji studují *in silico* s využitím výpočetních metod. Prvním krokem k jejich studiu je získání kvalitní informace o poloze jednotlivých atomů proteinu v prostoru. Pro tento úkol existují tři hlavní experimentální metody, které umožňují zachytit protein na atomární úrovni. První metodou je *rentgenová krystalografie*, druhou je *nukleární magnetická rezonance* a třetí *elektronová mikroskopie*.

Pokroky, kterých bylo dosaženo v posledních desetiletích v oblasti rentgenových technologií, umožnily charakterizovat struktury proteinů s velmi vysokým rozlišením. Při této metodě je protein vykryštalizován a podroben svazku rentgenového záření. Difraktovaný paprsek, který je zachycen snímačem, je poté analyzován za účelem určení polohy jednotlivých atomů proteinu. Proteiny jsou zachyceny v různých konformacích, ve kterých byly zachyceny v podobě krystalu. Díky tomu se získávají snímky různých proteinových konformací, které však nemusí odrážet trajektorie samotného proteinu [7].

Nukleární magnetická rezonance analyzuje protein, který je umístěn v rozpouštědle za vlivu silného magnetického pole. Struktura je podrobena zkoumání pomocí rádiových vln, které zachytí pozice atomů v proteinu. Na rozdíl od prvně jmenované metody nukleární magnetická rezonance umožňuje limitovaně studovat dynamiku (proměnlivost) proteinové



Obrázek 2.5: Ukázka permanentního (tunel 1), přechodného (tunel 2) a trvale uzavřeného (tunel 3) tunelu. Převzato z [51].

struktury. Možnost pozorovat molekulární dynamiky umožňuje ověřit přesnost jejich teoretických výpočtů a poskytuje příležitost k identifikaci přechodných tunelů.

Poslední zmíněnou metodou je elektronová mikroskopie. K zobrazení proteinu se využívá svazek elektronů a soustava elektronových snímačů. Výsledkem jsou 2D snímky, které se do podoby 3D struktury převádí pomocí specializovaných technik. Jednou z nich je například technika cryo-EM. Technika zobrazuje mnoho tisíc různých jednotlivých částic uchovaných v tenké vrstvě nekrytalického ledu. Předpokládá se, že tato zobrazení zachycují molekulu v nesčetných různých orientacích. Následný výpočetní postup, který se podobá počítačové tomografii, poskytne 3D mapu hustoty hmoty, ze které se získá 3D reprezentace proteinu [7].

Kapitola 3

Molekulární dynamika

Molekulární dynamika je metoda počítačové simulace pohybu atomů v závislosti na fyzikálních silách, které na ně působí. Simulace umožňují předpovídat chování molekul za určitých podmínek (teploty, tlaku, přítomnosti malých molekul a prostředí).

Historie výpočtu molekulárních dynamik sahá ke konci 50. let 20. století, kdy byl poprvé představen jejich koncept Alderem a Wainwrightem za účelem zachycení interakce mezi částicemi. V průběhu následujících desetiletích se tyto metody a poznatky postupně zlepšovaly a vytvořily tak jejich dnešní podobu a způsob využití. V současnosti nacházejí tyto simulace uplatnění v oblasti předpovídání dynamiky makromolekulárních struktur, především proteinů. Příkladem může být *formování struktury proteinu (protein folding)*, výpočet *volné energie* molekuly a studium *transportu látek* v proteinech [23].

Molekulární dynamiky slouží také k rozšíření teoretických znalostí o strukturách, které byly získány pomocí rentgenové krystalografie nebo nukleární magnetické rezonance ve statickém stavu. Můžeme o nich říct, že jsou tvořeny programem, který zaštiťuje simulační techniky pro pochopení principů uvnitř struktur na atomové úrovni [51].

Pro výpočet molekulárních dynamik se používají specializované softwary. Mezi nejpopulárnější nástroje patří AMBER, CHARM, GROMACS a NAMD. Ačkoli se jedná o velice optimalizované nástroje, stále není možné simulovat rozsáhlé a dlouhé dynamiky z důvodu velké náročnosti na hardware a výpočetní čas. Příkladem stroje, který je schopen provést výpočty simulující trvání pozorování stovky nanosekund nebo jednotky mikrosekund za využití velkého množství grafických jednotek, je superpočítač *Anton* [51].

V rámci kapitoly bude uvedena problematika molekulárních dynamik. Budou popsány jednotlivé kroky simulace skládající se z inicializace základních parametrů včetně silových polí. Dále se bude kapitola věnovat uvedení základních integračních metod, které se při výpočtu používají, a stanovení délky kroku simulace. Závěrem bude uveden příklad alternativního přístupu k výpočtu konformací proteinu.

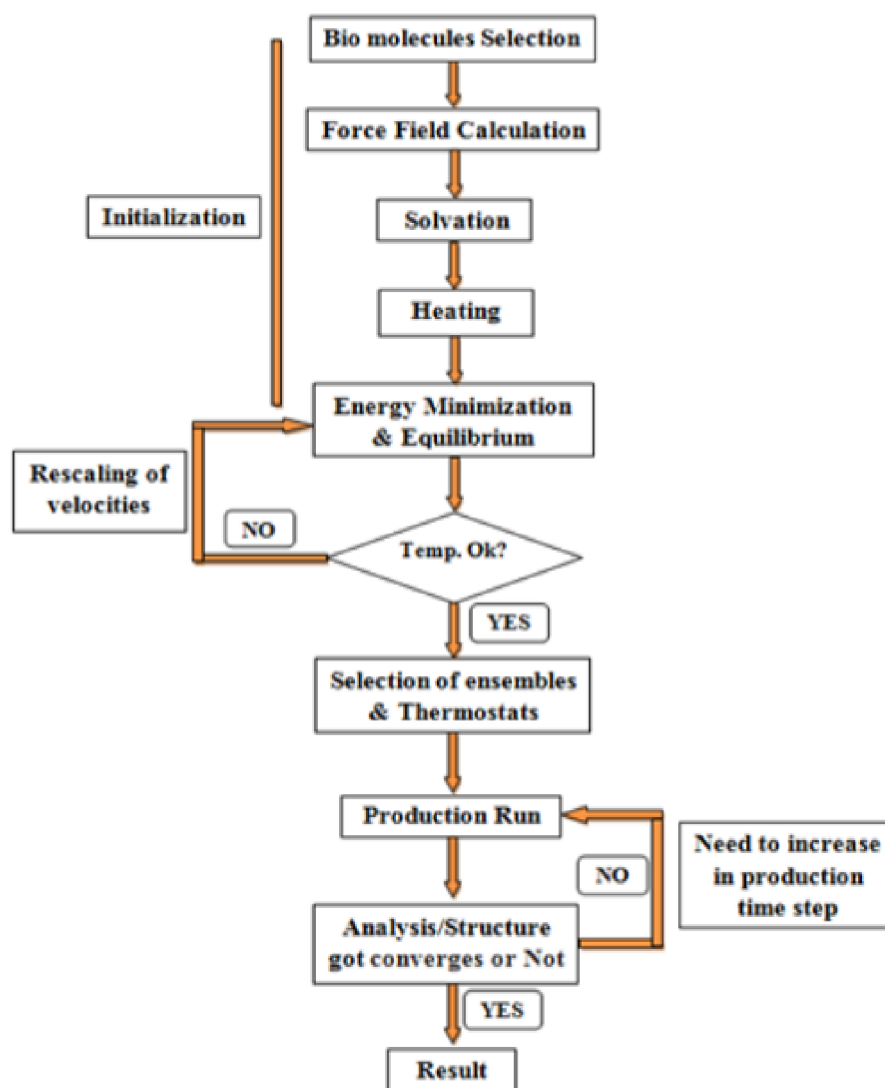
3.1 Kroky simulace

Simulační procedura obsahuje několik dílčích kroků, kterými jsou: *inicializace*, výpočet *silových polí (force field)*, kalkulace integrace *pohybových rovnic*, výběr *souborů (ensembles)* a termostatů (*thermostats*), běh simulace a analýza (viz obrázek 3.1). Prvním krokem je získání a příprava proteinové struktury. Využití je možné nejen experimentálně získané struktury, ale také výpočetní struktury získané například homologním modelováním.

Silová pole jsou definována pomocí datového souboru. Slouží k určení sil, kterými jednotlivé atomy disponují, a umožňují programu určit, k jakým interakcím bude v systému docházet. Problematika silových polí bude podrobně zachycena v následující části kapitoly. Současně s definicí silových polí dochází i k nasazení určitého modelu vody (příkladem může být SPC SPC/E a TIP3P), která poslouží jako rozpouštědlo (okolní prostředí) zkoumané struktury. Po nasazení rozpouštědla jsou do systému případně dodány ionty pro nastavení celkové neutrality.

Při výpočtu pohybových rovnic molekul a atomů se využívá různých integračních a numerických metod. Určení pozice atomů a jejich počátečního pohybu je provedeno v inicializační fázi. Příslušná metodika bude podrobněji popsána v další části této sekce.

Poté se provede zpracování souborů a termostatů s následným spuštěním běhu simulace po zvolenou dobu. Závěrem je provedena analýza výsledků, která je společně se získanou dynamikou předána uživateli [23].



Obrázek 3.1: Kroky výpočtu molekulární dynamiky (převzato z [23]).

3.2 Silová pole (force field)

Silová pole zachycují parametry, kterými můžeme definovat potencionální energii atomů a molekul v rámci simulačního modelu. Hodnoty parametrů jsou získány experimenty v laboratořích současně s aplikací kvantové mechaniky. Dnes rozlišujeme podle typu parametrů tři hlavní skupiny silových polí [34]:

- **atomové (all-atom)** – definujeme síly jednotlivých atomů.
- **molekulové (united-atom)** – určujeme síly působící v rámci skupin atomů a molekul (např. sloučenina uhlíku s vodíkem je reprezentována jako celek s větším poloměrem).
- **hrubořnné (coarse-grained)** – atomy a molekuly shlukujeme ve větší celky a ty bereme jako jednotku disponující určitou silou. Tento přístup se používá především v delších simulacích makromolekul (především proteinů).

Výpočet silových polí

Jak bylo zmíněno výše, v rámci struktur působí *vazebné* a *nevazebné* síly. Z těchto sil jsou i definována silová pole, tedy celkovou energii můžeme definovat jako [23]:

$$E_{celkem} = E_{vazebna} + E_{nevazebna} \quad (3.1)$$

přičemž energie vazebných a nevazebných sil závisí na kovalentních a nekovalentních vazebných interakcích, které mohou být vyjádřeny následovně [23]:

$$E_{vazebna} = E_{vazby} + E_{uhlova} + E_{vzpeti} \quad (3.2)$$

$$E_{nevazebna} = E_{elektrostaticke} + E_{vanderWalls} \quad (3.3)$$

V současnosti jsou vazby a vazebné energie obvykle modelovány pomocí harmonických oscilátorů, které zachycují jejich kmitání a zabraňují rozbití chemických vazeb.

V oblasti nekovalentních vazeb působí více meziatomárních interakcí. Obecně se modelují elektrostatické a van der Waalsovy síly. Pro tento účel se využívá Lennard-Jonesův potenciál a Coulombův zákon [26].

3.3 Pohyb částic

Molekulární dynamiky pro zachycení pohybu částic v čase využívají druhého Newtonova pohybového zákona. Energie částice (silové pole F_i) může být vypočítána pomocí vztahu

$$F_i = m_i a_i \quad (3.4)$$

kde m_i vyjadřuje její hmotnost a a_i udává její zrychlení. Zrychlení částice v systému je určeno podle sil na ni působících a je vyjádřeno jako poměr změny rychlosti částice v daném čase nebo jako změna potenciální energie v závislosti na pozici v systému.

$$a_i = \frac{\Delta v_i}{\Delta t_i} = -\frac{1}{m_i} \frac{\Delta E}{\Delta r_i} \quad (3.5)$$

Při vyjádření energie částice pomocí gradientu potenciální energie systému (V)

$$F_i = -\nabla_i V \quad (3.6)$$

můžeme vyjádřit vztah mezi změnou potenciální energie systému a pozice částice jako funkci závislou na čase (rovnice 3.7).

$$-\frac{\Delta V}{\Delta r_i} = m_i \frac{\Delta^2 r_i}{\Delta t^2} \quad (3.7)$$

Inicializační nastavení hybnosti částic je nastaveno z náhodného rozložení a později upraveno tak, aby celková hybnost systému byla nulová (rovnice 3.8).

$$P = \sum_{i=1}^N m_i v_i = 0 \quad (3.8)$$

Hodnota hybnosti v_i je získána obvykle z Gaussova nebo Maxwell-Boltzmannova rozložení s ohledem na teplotu systému [26, 27].

3.3.1 Integrační metody

Jelikož neexistuje analytické řešení pohybových rovnic, musí být přikročeno k jejich numerickému řešení. Pro integraci pohybových rovnic se využívá mnoho rozdílných algoritmů, které dělíme na dvě skupiny: *metody konečného rozdílu* (*finite difference method*) a *metody prediktor-korektor* (*predictor-corrector method*).

Metody konečného rozdílu provádí integraci do malých úseků o velikosti kroku Δt . Síla, která působí na částici, je určena součtem sil, kterými na ni působí částice v okolí. Pomocí síly je vypočítáno její zrychlení, přes které je postupnou integrací zjištěna pozice a rychlost. Příkladem metod konečného rozdílu mohou být: *Verletův*, *Beemanův*, *Rychlostní Verletův* a *Přeskokový* (*Leap-frog*) algoritmus.

Princip metod prediktor-korektor se skládá ze tří kroků. Prvním z nich je odhadnutí nových hodnot rychlosti, souřadnic a zrychlení pomocí Taylorova polynomu. V druhém kroku je obdobně jako u metod konečného rozdílu získána energie částice a z ní odvozeny hodnoty atributů částice. Hodnoty (společně s odhadnutými) jsou použity ke korekci souřadnic, rychlosti a zrychlení.

Díličí záznamy o pozici, rychlosti a zrychlení částic jsou společně s časovou značkou průběžně ukládány do *souboru trajektorie* (*trajectory file*), který tvoří jeden z hlavních výstupů simulace [56, 43].

Verletův algoritmus

Verletův algoritmus patří k nejčastěji používaným algoritmům v simulacích molekulárních dynamik. Principy, které tento algoritmus využívá, byly publikovány již v roce 1907 Carlem Stormem při studiu elektrických částic v magnetickém poli. V rámci molekulárních dynamik však byly poprvé použity v šedesátých letech minulého století Loupem Verletem. Mezi přednostmi algoritmu patří jeho jednoduchost a numerická stabilita [17].

Algoritmus je tvořen kombinací dvojice Taylorových polynomů (viz rovnice 3.9 a rovnice 3.10), které aproximují rychlosti (v_i), pozice (r_i) a zrychlení (a_i). Rovnice 3.9 zapisuje polynom v čase od t do $t + \Delta t$ a rovnice 3.10 od $t - \Delta t$ do t .

$$r_i(t + \Delta t) = r_i(t) + v(t)\Delta t + \frac{1}{2}a_i(t)\Delta t^2 + \frac{1}{6}\frac{\Delta^3 r_i(t)}{\Delta t^3}(t)\Delta t^3 + O(\Delta t^4) \quad (3.9)$$

$$r_i(t - \Delta t) = r_i(t) - v(t)\Delta t + \frac{1}{2}a_i(t)\Delta t^2 - \frac{1}{6}\frac{\Delta^3 r_i(t)}{\Delta t^3}(t)\Delta t^3 + O(\Delta t^4) \quad (3.10)$$

$O(\Delta t^4)$ vyjadřuje lokální chybu pozice Verletova integrátoru. Sečtením rovnic 3.9 a 3.10 získáme

$$r_i(t + \Delta t) + r_i(t - \Delta t) = 2r_i(t) + a_i(t)\Delta t^2 + O(\Delta t^4) \quad (3.11)$$

a substitucí $a_i = \frac{F_i}{m_i}$ (z rovnice 3.4) získáme tvar

$$r_i(t + \Delta t) = 2r_i(t) - r_i(t - \Delta t) + \frac{F_i}{m_i}(t)\Delta t^2 + O(\Delta t^4) \quad (3.12)$$

Podle vztahů vyplývajících z rovnic 3.5 a 3.6 můžeme zrychlení odhadnout pomocí funkce vyjadřující pozici v čase $r_i(t)$

$$a(t) \approx -\frac{1}{m_i}\nabla_i V(r_i(t)) \quad (3.13)$$

a rychlost v čase t jako

$$v(t) \approx \frac{r_i(t + \Delta t) - r_i(t - \Delta t)}{2\Delta t} \quad (3.14)$$

Metoda obsahuje dva hlavní kroky a je závislá na hodnotách $r_i(t + \Delta t)$ a $r_i(t - \Delta t)$, které musí být pro čas $t = 0$ dodatečně spočítány (např. prostřednictvím Eulerovy metody) [23, 25]. Metoda neumožňuje explicitní vyjádření rychlosti, které je závislé na výpočtu pozice částice v dalším kroku.

Přeskokový algoritmus

Přeskokový algoritmus (označován někdy jako Verletův přeskokový algoritmus) vznikl za účelem odstranění nevýhod výše zmíněného algoritmu. Algoritmus umožňuje explicitní vyjádření rychlosti a redukuje chybu Verletova integrátoru. Mezi chyby, které si algoritmus přenesl od svého předchůdce, patří problém *samostatného startu* (v čase $t = 0$), jenž musí být stále řešen Eulerovou metodou. Algoritmus přeskakuje kroky výpočtu tak, že rychlost $v(t)$ je známa až v čase $(t + \frac{\Delta t}{2})$ [23]. Z tohoto důvodu pracuje metoda v *nesynchronním* režimu, kdy výpočet pozice a rychlosti je proveden v jiném čase, takže není možné určit kinetickou energii částice v daném čase [43].

Podle principů použitých ve Verletově algoritmu můžeme získat rychlost středů intervalů od $(t - \Delta)$ do t (rovnice 3.15) a od t do $(t + \Delta)$ (rovnice 3.16) jako [23]:

$$v(t - \frac{\Delta t}{2}) = \frac{r_i(t + \Delta t) - r_i(t)}{\Delta t} \quad (3.15)$$

$$v(t + \frac{\Delta t}{2}) = \frac{r_i(t + \Delta t) - r_i(t)}{\Delta t} \quad (3.16)$$

a zrychlení lze definovat jako

$$a(t) = \frac{v_i(t + \frac{\Delta t}{2}) - v_i(t - \frac{\Delta t}{2})}{\Delta t} = \frac{F_i(t)}{m_i} \quad (3.17)$$

Tedy rychlost v čase $(t + \frac{\Delta t}{2})$ může být vyjádřena jako

$$v_i(t + \frac{\Delta t}{2}) = v_i(t - \frac{\Delta t}{2}) + \frac{\Delta t F_i(t)}{m_i} \quad (3.18)$$

Rychlostní Verletův algoritmus

Pro odstranění nevýhod, které neošetřil ani přeskokový algoritmus, vznikl *rychlostní Verletův algoritmus*. Tento algoritmus stanovuje rychlost a pozici částic ve stejném časovém okamžiku, takže dochází k běhu v synchronním režimu. Metoda stanovuje rychlost a pozici částice v dalším kroku na základě hodnot aktuálního. Hlavní předností algoritmu je absence ztráty přesnosti. Pozice a rychlost částice může být vyjádřena jako [23, 43, 8]:

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t + \frac{1}{2}a_i(t)\Delta t^2 \quad (3.19)$$

$$v_i(t + \Delta t) = v_i(t) + \frac{a_i(t) + a_i(t + \Delta t)}{2} \quad (3.20)$$

Beemanův algoritmus

Jako alternativa k Verletově algoritmu byl v roce 1973 představen Beemanův algoritmus. Metoda poskytuje integraci diferenciální rovnice druhého řádu, čímž dosahuje vyšší přesnosti. Vyšší přesnost metody je vykoupena složitějším tvarem rovnice, která způsobuje vyšší výpočetní náročnost [23, 43]. Rovnice posledního zástupce metod konečného rozdílu jsou zachyceny níže [8]:

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t + \frac{2}{3}a_i(t)\Delta t^2 - \frac{1}{6}a_i(t - \Delta t)\Delta t^2 \quad (3.21)$$

$$v_i(t + \Delta t) = v_i(t) + \frac{7}{6}a_i(t)\Delta t + \frac{1}{6}a_i(t - \Delta t)\Delta t \quad (3.22)$$

Metoda prediktor-korektor

Tato metoda byla poprvé představena v roce 1971 Williamem Gearem. Její princip spočívá v postupné úpravě (korekci) atributů hodnot v závislosti na změně systému. Jejím základem jsou tři kroky.

V prvním kroku dochází k odhadu nových souřadnic, rychlostí a zrychlení částic pomocí Taylorova polynomu. Poté je vypočítána energie částic, ze které je vypočítáno zrychlení. Dále jako u metod konečného rozdílu dochází k derivacím zrychlení a postupnému získání hodnot rychlosti a pozice částice. Pro tyto výpočty a pro odhady získané v prvním kroku se zjistí jejich rozdíl.

$$\Delta a_i(t + \Delta t) = a_i^c(t + \Delta t) - a_i(t + \Delta t) \quad (3.23)$$

Získaný rozdíl je použit v posledním (korekčním) kroku následovně:

$$r_i^c(t + \Delta t) = r_i(t + \Delta t) + c_0\Delta a_i(t + \Delta t) \quad (3.24)$$

$$v_i^c(t + \Delta t) = v_i(t + \Delta t) + c_1\Delta a_i(t + \Delta t) \quad (3.25)$$

$$a_i^c(t + \Delta t) = \frac{a_i(t + \Delta t)}{2} + c_2\Delta a_i(t + \Delta t) \quad (3.26)$$

Hodnoty koeficientů c_0, c_1, c_2 byly navrženy Williamem Gearem a jsou navrženy v závislosti na stupni Taylorova polynomu [8].

3.4 Výběr souborů a termostatů

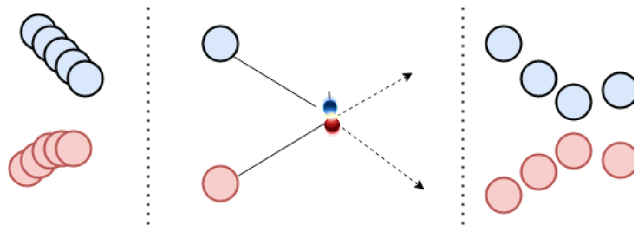
Soubor (*ensemble*) je velká skupina mikroskopicky popsaných stavů systému s určitými vlastnostmi. V simulacích molekulárních dynamik se využívá ke kvantifikaci dějů v celém systému za přítomnosti různých termodynamických podmínek. Soubory tvoří parametry pro molekulární podjednotky obvykle na atomové úrovni. Obecně rozlišujeme tři typy souborů [42]:

- **Mikroskopický soubor:** popisuje plně izolovaný systém s neproměnlivým počtem částic a s konstantní energií. V systému dochází k výměně kinetické a potenciální energie mezi částicemi systému, ale ne mezi jeho okolím. Každý prvek je charakterizován pozicí ($r_i(t)$), rychlostí ($v_i(t)$) a potenciálovou energií ($U_i(r)$) [41, 59].
- **Kanonický soubor:** (označován také jako molekulární dynamika konstantní teploty) označuje soubor, který provádí tepelnou výměnu s okolím o konstantní teplotě. Okolní prostředí se nazývá termostat a bude vysvětlen v další části [41, 59].
- **Izotermický-izobarický soubor:** popisuje soubory s konstantní teplotou a tlakem. Tento soubor je nejčastěji používán při simulaci chemických dějů, protože jsou často prováděny v laboratoři za konstantního tlaku [59].

Termostaty upravují pohybové vlastnosti částic na základě teploty systému. Termostaty umožňují statisticky předpovídat chování částic v jiných teplotních podmínkách. Příkladem algoritmu, kde se termostaty využívají, je *metoda simulovaného žhání*, která předchází ustálení dynamik v lokálním minimu pomocí postupně klesající energie, kdy vyšší teplota umožňuje změnu stavu systému. Termostaty pracují na dvou principech: deterministickém a stochastickém [28].

3.5 Běh simulace

Před spuštěním samotného běhu výpočtu je potřeba nastavit velikost diskrétního kroku simulace. Při nastavení malého kroku dochází k prodloužení výpočtu a produkci většího objemu dat, která nemají přidanou informační hodnotu. Naopak při volbě příliš velkého kroku hrozí vznik kolizí částic i v případě, kdy by díky působení vzájemných sil ke kolizi nedošlo. Jako optimální krok v oblasti molekulárních dynamik se nastavuje hodnota odpovídající $\frac{1}{10}$ doby trvání nejkratšího vibračního pohybu částice v systému (viz obrázek 3.3). Příkladem může být vazba uhlíku s vodíkem, která kmitá v periodě $10fs$ a tedy hodnota kroku simulace je $1fs$ [8].



Obrázek 3.2: Ilustrace možných situací při nastavení: malého kroku (vlevo), příliš velkého kroku (uprostřed) a optimálního kroku (vpravo).

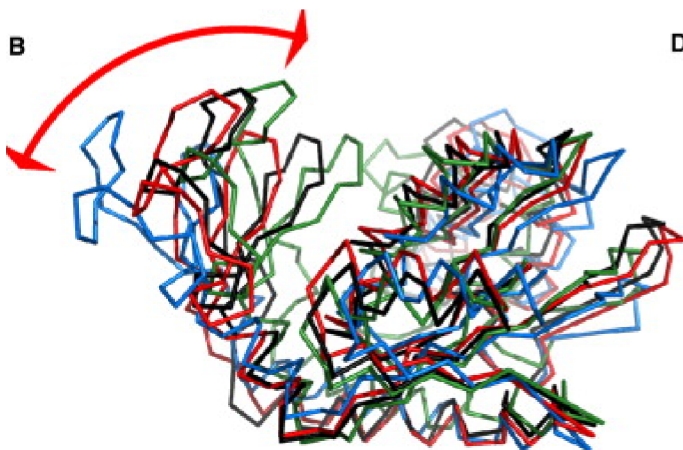
3.6 Aproximativní metody

Ve většině přístupů, které se používají k výpočtu řešení proměnlivosti proteinů, se využívá simulací molekulárních dynamik. Navzdory nárůstu výkonu počítačů a pokroku v technikách algoritmizace a paralelizace jsou výpočty molekulárních dynamik stále výpočetně i časově nákladné. Takže se často například nedaří překonat vysokoenergetické bariéry, které simulaci zabraňují přejít do jiných zajímavých konformací [47].

Za účelem zmírnění problému s výběrem vzorků bylo vyvinuto několik pokročilých simulačních metod založených na molekulárních dynamikách včetně molekulárních dynamik s výměnou replik (replica-exchange molecular dynamics – REMD) [50], konformační záplavy (conformation flooding) [22, 32] a cílené molekulární dynamiky (targeted molecular dynamics – TMD) [50, 54], které byly s úspěchem aplikovány na řadu problémů v oblasti výzkumu proteinů. Bohužel ani tyto metody nejsou často použitelné pro efektivní vzorkování konformačních přechodů v rozumném čase [47].

Proto vznikají i úplně odlišné aproximativní metody, které cílí na rychlejší prozkoumání větší části konformačního prostoru proteinu. Jedním ze zástupců je metoda CONCOORD [19] a z ní odvozená metoda tCONCOORD [47]. Tyto metody fungují na čistě geometrickém principu a jsou tudíž velmi rychlé. Prvním krokem jejich výpočtu je převedení proteinu na sadu vrcholů (atomů) a hran (důležitých vazeb) s daným kritériem (např. maximální délka vazby). Poté jsou atomy umístěny náhodně do prostoru a program je následně iterativně transformuje tak, aby bylo splněno co nejvíce kritérií pro vazby. Díky náhodnému prvotnímu umístění atomů je tak možné najít i úplně odlišné konformace, které mají vysokou energetickou bariéru a tudíž by byly postupnou simulací z prvotního stavu velmi těžko dosažitelné. V každém kroku vznikají desítky nových možných konformací, z nichž lze dále počítat další tvary proteinu [47].

Původní metoda CONCOORD pracuje pouze na principu generování nových konformací v okolí dané struktury. Metoda tCONCOORD tento postup rozšiřuje o predikování konformačních přechodů mezi jednotlivými stavy a umožňuje tak predikovat trajektorie v proteinových strukturách.



Obrázek 3.3: Grafické znázornění práce metody tCONCOORD: struktura je proložena geometrickou reprezentací s její následnou transformací (převzato z [47]).

tCONCOORD tvoří alternativu k výpočtu molekulárních dynamik. Molekulární dynamiky investují výpočetní výkon do přesného určení koncové pozice atomu v rámci simulačního kroku. Oproti tomu metoda tCONCOORD tento výpočetní výkon přesouvá do pro-

cesu ověření správnosti náhodně vygenerovaných snímků. Pomocí programu tCONCOORD je možné dosáhnout generování velice různorodých konformací daného proteinu, kterých by při běhu simulace bylo dosaženo až po dlouhém výpočtu. Získáme tak různorodé konformace v poměrně krátkém čase.

Kapitola 4

Nástroj CAVER a CaverDock

4.1 CAVER – Nástroj pro analýzu transportních tunelů

Jak bylo zmíněno v sekci o experimentálních metodách 2.4.1, v dnešní době je možné přesně identifikovat 3D strukturu proteinů. Tyto metody nám umožňují zachytit pozice jednotlivých atomů. Pro identifikaci tunelů je však potřeba přistoupit k podrobnější analýze pomocí specializovaných nástrojů. Jedním z těchto nástrojů je i CAVER, který bude popsán v následující části této kapitoly. Informace o nástroji CAVER byly získány z článku [14].

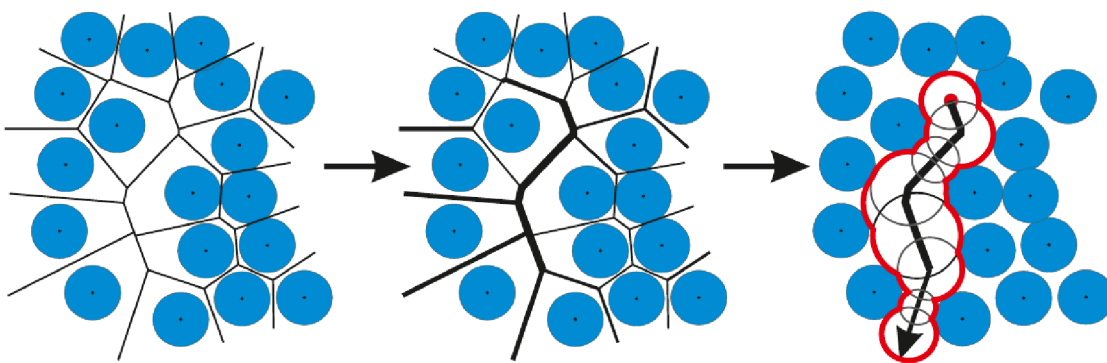
Nástroj CAVER je bioinformatický software, jenž slouží k analýze transportních tunelů v dynamických chemických strukturách. Nástroj hledá tunel z uživatelem definovaného počátečního bodu až do vnějšího prostředí rozpouštědla. Software byl vyvinut týmem specialistů v Loschmidtových laboratořích v Brně. Program je implementován v jazyce Java a v současnosti je k dispozici již třetí verze tohoto nástroje. Existující alternativu tvoří nástroje MOLE, HOLE a MolAxis.

4.1.1 Identifikace tunelů

Konstrukce Voroného diagramu

Nástroj CAVER 3.0 reprezentuje strukturu pomocí tzv. *obyčejného Voroného diagramu*, který pracuje s koulemi stejného poloměru. Podobný přístup byl použit i v nástrojích MOLE 1.2 a MolAxis 1.4, kdy koule v diagramu reprezentují atomy zkoumané struktury. Nevýhodou tohoto přístupu je vznik relativně velké chyby ve velikosti poloměru identifikované tunelu z důvodu odlišných velikostí atomů různých chemických prvků, které obyčejný Voroného diagram nedokáže implicitně reprezentovat. Chyba dosahuje velikosti až $r_1 - r_2$, kde r_1 představuje poloměr největšího atomu a r_2 poloměr nejmenšího atomu.

CAVER 3.0 pro zachování rozlišení velikostí jednotlivých atomů aproximuje všechny atomy ve struktuře, které jsou větší než nejmenší z nich, náhodným samplováním prostoru uživatelem daným počtem koulí s poloměrem odpovídajícím van der Waalsovému poloměru nejmenšího atomu ve struktuře. Nejmenší atom je pak reprezentován jednou koulí o poloměru, který odpovídá velikosti jeho van der Waalsovému poloměru. Výsledná reprezentace tak nikdy nepřesahuje skutečný van der Waalsův povrch struktury. To znamená, že průměr identifikovaných tunelů může být nadhodnocený, ale nikdy nemůže dojít k podhodnocení.



Obrázek 4.1: Konstrukce Voroného diagramu. Atomy reprezentují stavební body (site points – body, podle kterých se rozkládá prostor) a hrany diagramu reprezentují oblast s největší vzdáleností od atomů. Hrany tvoří osy proteinového tunelu a sousedící hrany se dotýkají ve Voroného vrcholu (převzato z [3]).

Identifikace tunelů

Počátek pro výpočet tunelu je určen jako střed z uživatelem zadaných reziduí, atomů nebo jako bod zadaný pomocí souřadnic na jehož základě je vybrán takový Voroného vrchol, který leží maximálně ve vzdálenosti d_{max} (maximální vzdálenost, obrázek 4.2) od počátečního bodu a jeho vzdálenost od struktury reprezentující protein je alespoň r_{min} (požadovaný poloměr, obrázek 4.2). Pokud takový vrchol neexistuje, je vybrán vrchol ve vzdálenosti do d_{max} mající maximální vzdálenost od reprezentace struktury. Pokud nebude nalezen odpovídající vrchol, je hodnota d_{max} navýšena na hodnotu 3.0 Å a procedura se zopakuje. Pokud ani v tomto případě není vrchol nalezen, je vybrán vrchol s nejmenší vzdáleností od počátečního bodu.

Tunely se skládají z Voroného vrcholů a Voroného hran. Při hledání tunelu se rozlišují dva typy vrcholů: *vrcholy rozpouštědla* a *vrcholy povrchu*. *Vrchol rozpouštědla* je definován jako vrchol dostupný z vnějšího prostředí struktury (rozpouštědla) pomocí sondy o daném poloměru r_s (poloměr povrchové sondy, obrázek 4.2). Do každého vrcholu rozpouštědla je umístěna koule o maximálním poloměru nemající kolizi s reprezentací struktury. Množina *vrcholů povrchu* je určena jako body ležící v sousedství s koulemi z vrcholů rozpouštědla tak, že je k nim možné přistoupit z koule se středem putujícím ze středu koule ve vrcholu rozpouštědla (se středem v bodě A a poloměrem r) po hraně s poloměrem r_B zvoleným tak, že nikdy není v kolizi s reprezentací struktury a vždy leží uvnitř koule se středem v bodě A a poloměrem $r + d_s$, kde d_s prvotně nastaveno na hodnotu 4 Å. Dále je definovaná množina *hraničních povrchových vrcholů* jako body z množiny povrchových vrcholů spojené hranou s vrcholem, který neleží v množině vrcholů. Podobně tak množina *hraničních vrcholů rozpouštědla* je množina bodů, které náleží množině vrcholů rozpouštědla a jsou hranou spojeny s vrcholem, který do této množiny nenáleží.

Hledání tunelů se uskutečňuje ve dvou krocích. V prvním kroku jsou pomocí Dijkstrova algoritmu nalezeny tunely propojující startovací vrchol s vrcholy hraničního povrchového bodu. V druhém kroku je každý takový tunel prodloužen až k vrcholu rozpouštědla. Poté jsou na nalezený tunel v intervalových krocích položeny body, které slouží jako středy koulí s poloměrem nastaveným tak, aby byl maximální a nebyl v kolizi se strukturou. Na závěr jsou z konců tunelů odstraněny koule, jejichž poloměr přesahuje definovaný největší poloměr r_s .

Při identifikaci může dojít k identifikaci redundantních tunelů, které se jen mírně liší od ostatních. Proto je po nalezení tunelů spuštěno jejich čištění způsobem, kterým se nalezne nejlépe hodnocený a všechny tunely, jejichž vzdálenost od nejlépe ohodnoceného je menší než uživatelem definovaná, jsou odstraněny. Výpočet vzdálenosti jednotlivých tunelů bude vysvětlen v části 4.1.2.

Hodnotící funkce

Prostor je reprezentován pomocí stavebních bodů (site points – středy koulí se stejným poloměrem reprezentující atomy) a Voroného hran. Tunel ve struktuře je založen na sekvenci Voroného hran, kde dvě sousedící hrany sdílí jeden Voroného vrchol. Na tyto hrany jsou umístěny středy koulí, jejichž poloměr je stanoven tak, aby byl maximální a současně nekolidoval s reprezentací struktury (s koulemi se středy ve stavebních bodech diagramu). K ohodnocení pravděpodobnosti tunelu o délce L , zda slouží k transportu látek, byla vytvořena *hodnotící funkce*, která je zachycena v rovnici 4.1. Tato varianta počítá s konstantním poloměrem (r) koulí na Voroného hranách.

$$c(r, L) = \frac{L}{r^n} \quad (4.1)$$

n je uživatelem zadané reálné číslo v rozmezí 0 až 100, které upřednostňuje délku tunelu před jeho šířkou a naopak (pokud n je nulové, tak je upřednostněna délka).

K ohodnocení tunelu s proměnlivou šířkou byla vytvořena další hodnotící funkce

$$\int_0^L r(l)^{-n}, dl \quad (4.2)$$

kde $r(l)$ je maximální poloměr koule nekolidující se strukturou a mající střed ve vzdálenosti l po ose tunelu od počátečního vrcholu. Uživatel pro hodnotící funkci tunelu s proměnlivou šířkou dodatečně definuje maximální hodnotu funkce $r(l)$ (r_{max}), která je použita při přesažení hodnoty r_{max} . Minimální hodnota poloměru koule (r_{min}) je stanovena na 0.1 Å.

Před začátkem výpočtu hodnotící funkce jsou odstraněny ty tunely, které nesplňují uživatelem zadané podmínky minimální velikosti hrdla tunelu (r_B).

Na závěr je hodnota hodnotící funkce přetransformována do intervalu v rozmezí 0 až 1 označující *propustnost* daného tunelu pomocí vztahu e^{-cost} , kde $cost$ je hodnota hodnotící funkce a e je Eulerovo číslo. Vyšší hodnota propustnosti označuje tunely s větším potenciálem pro transport látek.

4.1.2 Shlukování tunelů

Nástroj CAVER 3.0 umožňuje analyzovat data získána z molekulárních dynamik tak, že podrobí analýze každý získaný *snímek* (*snapshot*). Výsledkem je soubor tunelů pro každý snímek, které se následně kvůli statistice a snazší analýze slučují podle podobnosti. Pro určení odpovídajících si tunelů napříč jednotlivými snímky a pro odstranění redundantních tunelů se využívá výpočtu *vzdálenosti tunelů*.

Pro výpočet vzdálenosti tunelů je využit princip, kdy jsou jednotlivé tunely proloženy sekvencí N bodů a vzdálenost dvojice tunelů se vypočítá pomocí vzdálenosti pořadí si odpovídajících bodů porovnávaných tunelů. Tento princip má pouze lineární časovou složitost, a proto urychluje výpočet. Na začátku je určen bod S jako střed mezi pozicemi počátečních bodů ve všech snímcích. Poté je vytvořena množina bodů P postupnou iterací přes nalezené

tunely ve vzestupném pořadí podle jejich cen. Pro každý zpracovávaný tunel je nalezen *extrémní bod* X , který odpovídá bodu tunelu nejméně vzdálenému od S . Ze vzdálenosti bodů S a X je stanovena *přímá délka*. Dále je zvolen bod γ z P tak, že úhel $XS\gamma$ je minimální. Pokud je množina P prázdná nebo velikost úhlu překročila uživatelem definovaný limit, bod X je přidán do množiny P a jeho váha se nastaví na jedna. Jinak je X odstraněno a bod γ se přesune po spojnici s S tak, že jejich nová vzdálenost je $|SX| + \omega_\gamma * |S\gamma|$, kde ω_γ reprezentuje váhu bodu γ , která je následně inkrementována o jedna. Na konci zpracování množina P sdružuje průměrné délky tunelů mající konec blízko u sebe.

Poté je opětovně zpracován soubor tunelů, kdy se vytvoří podmnožina Q z P tak, že pro extrémní bod X aktuálního tunelu a jakýkoli bod γ z Q platí, že velikost úhlu $XS\gamma$ je nižší než uživatelem zadaný parametr. Pro získané tunely je vypočítána hodnota d_{end} , která odhaduje průměrnou velikost přímé délky v blízkosti aktuálně zpracovaného tunelu, pomocí vztahu 4.3.

$$\frac{\sum_{Z \in Q} w_Z |ZS|}{\sum_{Z \in Q} w_Z} \quad (4.3)$$

Poté je osa každého tunelu převedena do množiny N bodů. Pro každý tunel je jeho vzdálenost d_{end} rozdělena do $N+2$ intervalů $I_{start}, I_1, I_2, \dots, I_N$ a I_{end} , kde velikost intervalů I_{start} a I_{end} určuje uživatel a nejsou použity k porovnání podobnosti tunelů. Pro i od 1 do N je vypočteno těžiště G_i bodů, jejichž vzdálenost od bodu S spadá do intervalu I_i . Následně vzniká pro každý tunel sekvence těžišť G_1, G_2, \dots, G_N a vzdálenost tunelů A a B je dána jako

$$\frac{1}{N} + \sum_{i=1}^N Nw\left(\frac{i-1}{N-1}\right) |G_{Ai}G_{Bi}| \quad (4.4)$$

, kde $|G_{Ai}G_{Bi}|$ vyjadřuje Euklidovskou vzdálenost odpovídajících si bodů z tunelů A a B . w představuje lineární funkci, která určuje váhu koncového a počátečního bodu tunelu. Koeficienty funkce w mohou být uživatelem nastaveny prostřednictvím parametru q ($q > 1$ je priorita přiřazena koncové sekci).

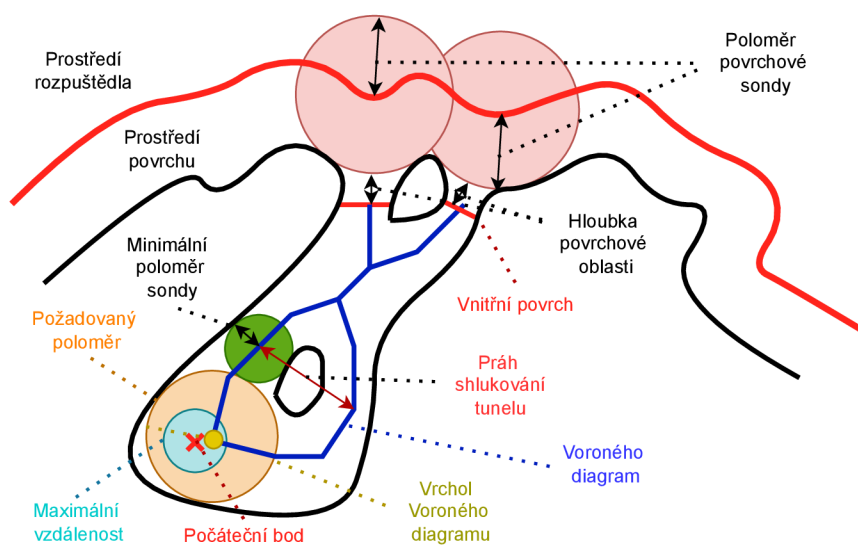
Pro shlukování tunelů je využito hierarchické shlukování schopné velmi přesného nalezení hranic jednotlivých shluků. Pro konstrukci stromu byl v poslední verzi software zvolen algoritmus m-Adic [40], který dosahuje lepší časové složitosti. Jeho principem je srovnávání vzdáleností tunelů jako řetězce a následně sestavený strom udává nejdelší prefix jednotlivých vzdáleností. Jednotlivé úrovně stromů odpovídají řádům hodnoty vzdáleností tunelů. Pokud tunely sdílejí bod ve stromě, znamená to, že daný řád vzdáleností je shodný. Tímto způsobem je vytvořen strom s úrovní větvení až deset [40]. Výsledné shluky tunelů jsou poté vytvořeny říznutím stromu na uživatelem definované úrovni.

4.1.3 Parametry nástroje

S ohledem na specifičnost různých proteinů, malých molekul i tunelů není možné jednoznačně definovat univerzální parametry výpočtů. Proto CAVER poskytuje nepřeborné množství parametrů, kterými je možné ladit kterýkoliv krok výpočtu. Vybrané klíčové parametry jsou znázorněny na obrázku 4.2 a blíže popsány níže:

- **Minimální poloměr sondy** (minimum probe radius) – určuje minimální poloměr, který musí mít nejužší místo tunelu, aby byl identifikován.

- **Hloubka povrchové oblasti** (shell depth) – určuje maximální hloubku povrchové oblasti, tj. části vstupní struktury, která se nachází pod objemovou oblastí rozpouštědla. V povrchové oblasti není povoleno větvení tunelu.
- **Poloměr povrchové sondy** (shell radius) – určuje poloměr povrchové sondy, jež bude použita k určení, které části Voroného diagramu představují vnější prostředí.
- **Práh shlukování tunelů** (clustering threshold) – určuje úroveň detailu, na které bude stromová hierarchie shluků tunelů rozdělena, a ovlivňuje velikost výsledných shluků. Čím vyšší je práh shlukování, tím větší jsou výsledné shluky.
- **Maximální vzdálenost** (maximal distance) – udává maximální vzdálenost počátečního bodu výpočtu (tj. vrcholu Voroného diagramu) od počátečního výchozího bodu. Pokud není nalezen žádný vrchol v uživatelsky zadané vzdálenosti, použije se výchozí hodnota `max_distance`. Pokud ani pak není nalezen žádný vrchol, je jako výchozí bod výpočtu vybrán vrchol, který je nejbližší počátečnímu výchozímu bodu.
- **Požadovaný poloměr** (desired radius) – specifikuje požadovaný poloměr volného prostoru kolem počátečního bodu (vrcholu Voroného diagramu vybraného pomocí parametru maximální vzdálenosti), ve kterém se nesmí nacházet atomy proteinové struktury. Pokud takový vrchol neexistuje, vybere se ze všech vrcholů, které se nacházejí v zadané vzdálenosti od počátečního výchozího bodu, vrchol s maximální vzdáleností od koulí představujících vstupní strukturu.



Obrázek 4.2: Grafické znázornění jednotlivých parametrů nástroje CAVER.

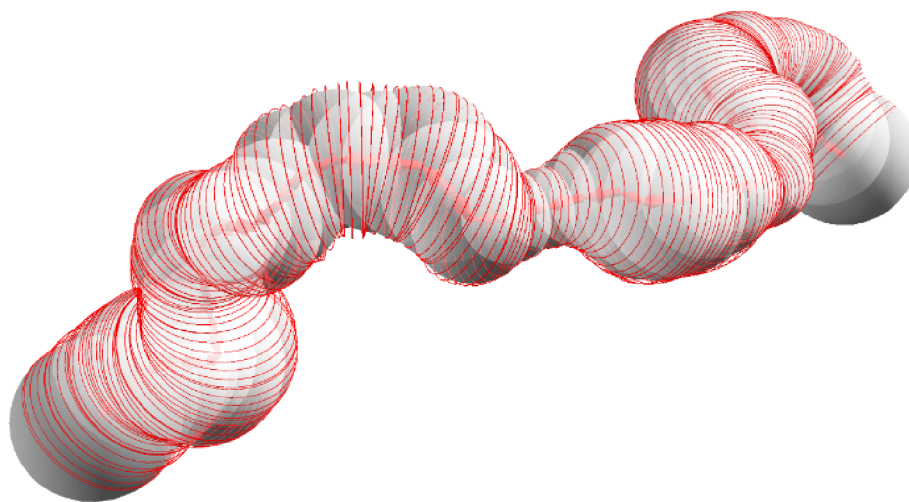
4.2 CaverDock – Nástroj pro analýzu transportu ligandů

Tunely a kanály, které se nacházejí ve struktuře proteinu, slouží jako přístupové (transportní) cesty pro malé molekuly, jež se nazývají *ligandy*. V případě tunelů se jedná o pří-

stupové cesty do zanořených vazebných a katalytických míst proteinů. Využívání kanálů pozorujeme především u membránových proteinů, které jsou zodpovědné za výměnu látek mezi prostředími (např. vnější a vnitřní prostředí buňky).

Při analýze tunelů a kanálů chceme zjistit, jestli tyto cesty mohou sloužit k transportu námi vybraných ligandů. Za tímto účelem vznikl nástroj CaverDock, který byl vyvinut v Loschmidtových laboratořích a byl publikován v článku [55]. Nástroj je založen na prostorově omezeném modifikovaném algoritmu nástroje AutoDock Vina [52] určeného pro molekulové dokování, kterým iterativně ohodnocuje interakční energii ligandu s proteinem v každém úseku tunelu. Jeho výstupem je pak energetický profil, který může být využit pro identifikaci geometrických a energetických bariér průchodu.

Program přijímá tunely popsané pomocí posloupnosti koulí se středem na ose tunelu (sfér) v datovém formátu PDB a byl optimalizován pro výstup programu CAVER (viz sekce 4.1). Posloupnost koulí je poté diskretizována na posloupnost disků (průřezů) o maximální vzdálenosti zadané uživatelem [55] (obrázek 4.3).

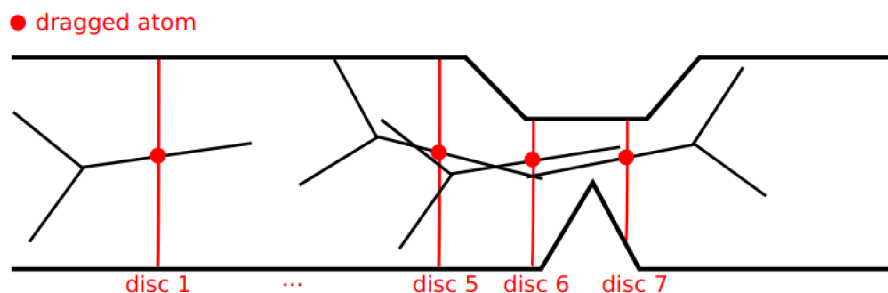


Obrázek 4.3: Diskretizace tunelu do průřezů o maximální vzdálenosti zadané uživatelem (převzato z [21]).

Jednotlivé disky slouží pro umístění ligandu a ohodnocení vazebné energie a interakcí s proteinem. Proces analýzy průchodu ligandu se skládá ze tří kroků: (i) vybraný atom ligandu se umístí na disk tak, aby vyhovoval polohovým omezením; (ii) CaverDock minimalizuje konformaci ligandu a vyhodnotí jeho vazebnou volnou energii pomocí skórovací funkce z programu AutoDock Vina [52]; (iii) trajektorie ligandu se vytvoří agregací dokovaných pozícií ligandu v každém po sobě následujícím disku [55].

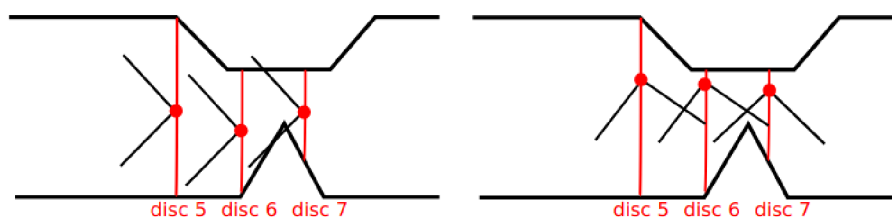
Trajektorie vzorkuje tunel podle uživatelem zadaného rozlišení. Jelikož je v každém kroku (disku) nalezena konformace ligandu, která odpovídá jeho nejnížší dosažitelné volné vazebné energii v tomto místě, nalezená trajektorie může být nespojitá (nekontinuální) (obrázek 4.4). Tato nespojitá posloupnost poloh slouží pro odhad dolní hranice (nejnížšího) energetického profilu transportu ligandu tunelem [55].

Protože nespojitý pohyb se může vyhnout úzkým místům ligandu prostřednictvím biologicky nerealistických změn orientace nebo konformace, jako je například okamžité zrcadlové otočení, výsledná hodnota energetického profilu může být vyšší. Možným řešením je využití výpočet spojité trajektorie (horní hranice energetického profilu, obrázek 4.5), která vynucuje maximální povolené změny polohy a konformace. V každém kroku je ligand dokován



Obrázek 4.4: Příklad nespojité trajektorie ligandu. Dojde k překlopení ligandu o 180° mezi disky 6 a 7. Polohy odpovídají nejnižším volným vazebným energiím v daných polohách, ale nezaručují spojitost přechodů mezi jednotlivými disky (převzato z [21]).

v blízkosti své předchozí polohy, což mu umožňuje provést pouze malé změny v jeho konformaci. Počet možných spojitých trajektorií roste exponenciálně s počtem disků, protože každý přechod na nový disk může vést ke změnám polohy, orientace a konformace ligandu. Proto se k nalezení spojitě trajektorie (nazývané horní odhad energetického profilu) využívá heuristická metoda, která ohodnocuje aktuální volnou vazebnou energii umístěného ligandu vůči jeho dolnímu odhadu energetického profilu. Pokud je vazebná volná energie výrazně vyšší, než její spodní odhad, využije se backtrackingu, kdy se nejprve pozmění konformace ligandu a následně se aplikuje přechod ligandu do předchozích disků. Backtracking se ukončí v případě dosažení počátečního disku nebo při sblížení dopředných a zpětných přechodů [55].



Obrázek 4.5: Příklad spojitě trajektorie ligandu (v prvním případě průchod tunelem není umožněn z důvodu špatné konformace) (převzato z [21]).

Zatímco trajektorie s dolní hranicí je schopna kompletně vzorkovat trajektorii ligandu, trajektorie horní vazby je zcela plynulá. Informace z dolního odhadu energetického profilu jsou pro účely porovnání dostatečné, ale mohou přehlédnout malá úzká místa způsobená rychlými změnami orientace ligandu. Náhlé změny orientace však mohou napodobovat flexibilitu proteinu a snížit nepřírozené energetické bariéry způsobené rigiditou receptoru během vazby nebo rozvázání. Na druhé straně může horní odhad energetického profilu generovat nerealistické konformace ve velmi úzkých částech tunelu, které se charakterizují ostrými vrcholy v energetickém profilu vazebné energie. Energetický profil dolního odhadu reprezentuje nejlepší možný scénář transportu ligandu, zatímco ne vždy optimální horní odhad může vykazovat extrémní energetické profily [55].

Kapitola 5

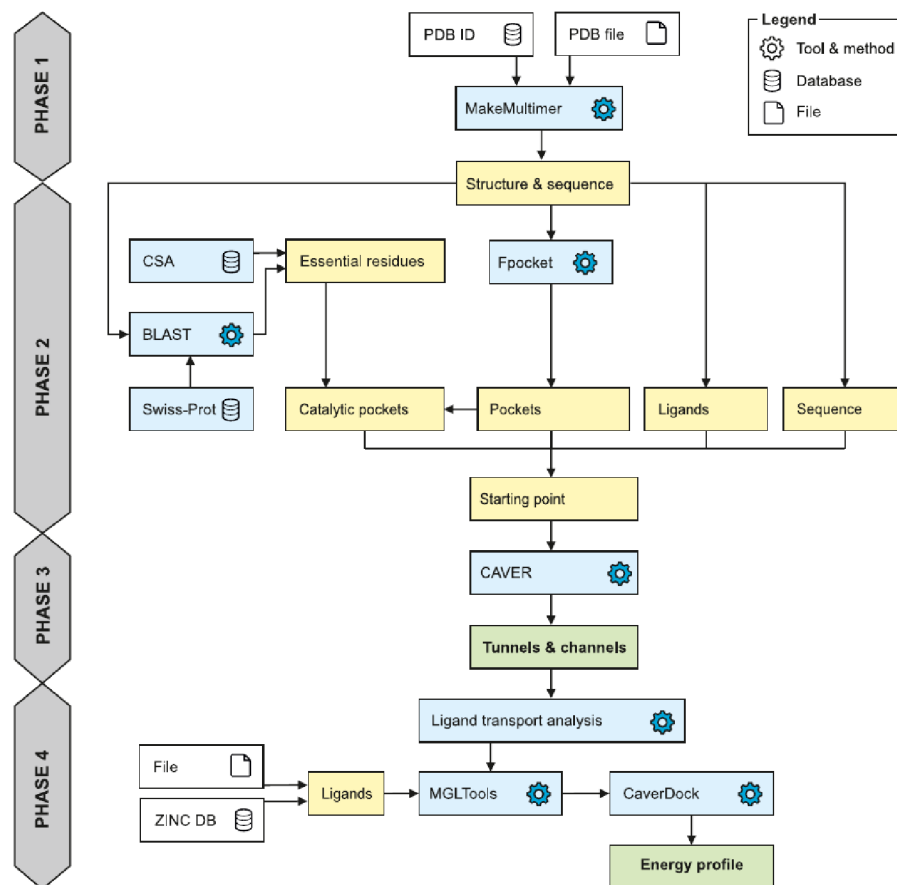
Caver Web 1.0

Caver Web 1.0 [49] je webový server určený na identifikaci proteinových tunelů a kanálů, pro které umožňuje spustit analýzy transportu ligandů. Program se vyznačuje intuitivním a uživatelsky přívětivým rozhraním s minimem požadovaných vstupů od uživatele. Server je vhodný i pro výzkumníky bez pokročilých bioinformatických nebo technických znalostí. Jeho současná verze je ve vědecké komunitě dobře zavedena a velmi využívána (35 000 dokončených výpočtů během dvou let provozu). Nástroj byl vyvinut Loschmidtovými laboratoři a je dostupný na adrese: <https://loschmidt.chemi.muni.cz/caverweb/>. Nejvýznamnějším omezením současné verze je možnost analyzovat pouze statickou strukturu, což často poskytuje neúplný biologický obraz.

První verze aplikace se skládá z přímočarého workflow, které uživatele provede celým procesem bez nutnosti jeho větších zásahů. Pracovní protokol se skládá ze čtyř fází, ve kterých uživatel postupně definuje (i) strukturu, se kterou se bude pracovat; (ii) počáteční bod, ze kterého budou hledány proteinové tunely; (iii) výpočet proteinových tunelů a (iv) analýzu proteinových tunelů s možností studovat transport ligandu skrze tyto tunely. Workflow je zachyceno na obrázku 5.1. Popis jednotlivých kroků byl čerpán ze článku [49].

V první fázi je uživatel vyzván k výběru struktury, se kterou bude program pracovat. Uživatel může vložit svoji proteinovou strukturu ve formátu PDB nebo mmCIF (obrázek 5.2b). Jelikož jsou struktury obvykle uloženy ve formě asymetrických jednotek, které nemusí odrážet jejich přirozeně se vyskytující kvartérní formy (biologické jednotky), provedené analýzy těchto struktur mohou vést k biologicky nesprávným výsledkům. Za účelem odstranění těchto rizik je spuštěn software MakeMultimer (<http://watcut.uwaterloo.ca/tools/makemultimer/>), který nalezne odpovídající biologické jednotky. Jejich seznam a popis je poskytnut uživateli, který může zvolit nejvhodnější biologickou jednotku nebo ji odmítnout a pokračovat s původní strukturou (obrázek 5.2a) [49].

Nejdůležitějším krokem při detekci tunelu je výběr správného výchozího bodu. Pro detekci tunelů je využit software CAVER (viz sekce 4.1), kde výchozí bod definuje společný bod pro všechny detekované tunely. Špatně umístěný bod může významně ovlivnit relevanci detekovaných tunelů. Pro usnadnění výběru počátečního bodu se využívá automatických protokolů, které zahrnují výběr bodů pomocí nejběžnějšího scénáře – selekce bodu uvnitř katalytické kapsy enzymů. U enzymů se uživatelé často zajímají o přístupové cesty vedoucí k aktivním nebo vazebným místům, kde by mohlo dojít k navázání ligandu. Nejlepší výchozí bod pro tuto analýzu je tedy obvykle umístěn uvnitř kapsy obsahující katalytická rezidua (katalytická kapsa). Proto Caver Web poskytuje plně automatický režim schopný detekce katalytických kapes na základě geometrické identifikace kapes a informací o esenciálních (katalytických) reziduí z databází. Kapsy jsou detekovány pomocí programu Fpocket 2 [33],



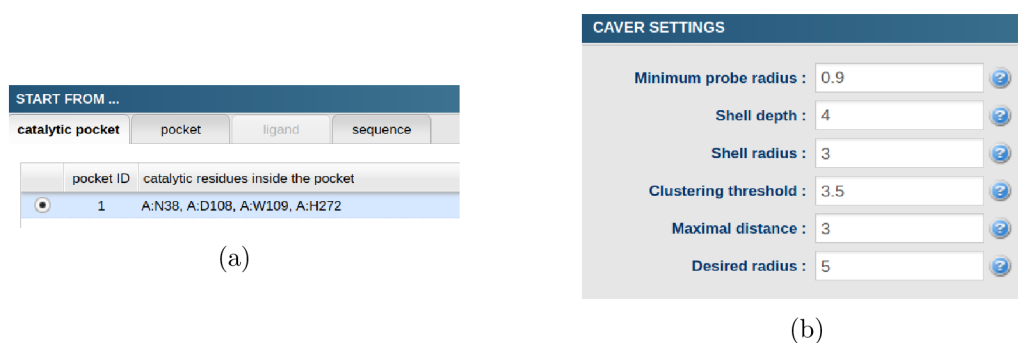
Obrázek 5.1: Workflow původní verze aplikace. Skládá se ze čtyř fází: (i) zadání vstupu uživatelem; (ii) výběr počátečního bodu pro výpočet proteinových tunelů; (iii) identifikace tunelů a (iv) zobrazení výsledků s možností analýzy transportu ligandů (převzato z [49]).

který je založen na vyhledávání prázdných prostor (sfér) v reprezentaci proteinových struktur pomocí Voroného diagramu a následném shlukování těchto prostor do větších celků. Výhodou tohoto nástroje je, že mimo jiné vypočítává skóre vaznosti (druggability), které reprezentuje pravděpodobnost, že se malé molekuly (zvané ligandy) mohou navázat do identifikované kapsy. Katalytická rezidua jsou získána z manuálně anotovaných částí databází Mechanism and Catalytic Site Atlas [46] a SwissProt [15]. Záznamy v Mechanism and Catalytic Site Atlas jsou mapovány pomocí přístupových kódů PDB. Databáze SwissProt je prohledávána pomocí nástroje BLAST s požadavkem na 30% shodu v sekvenci s délkou odpovídající 90 až 110 % délky původní sekvence. Po identifikaci katalytických reziduí jsou kapsy přiřazeny k těmto zbytkům. Kapsy, které obsahují alespoň jedno katalytické reziduum, jsou označeny jako potenciálně katalytické a seřazeny dle jejich skóre vaznosti. Pokud nejsou katalytické kapsy nalezeny, jsou poskytnuty další dvě možnosti výběru počátečního bodu pro výpočet tunelů. V prvním z nich jsou uvedeny všechny zjištěné kapsy v pořadí podle jejich predikovaného skóre vaznosti (druggability). Druhý způsob umožňuje definovat startovací bod v rámci středu jakéhokoli ligandu přítomného ve struktuře. Tím je možné jednoduše startovat z experimentálně potvrzené pozice navázaného ligandu. Tento režim však vyžaduje, aby byl protein kokrytalizován (co-crystallized) nebo obsahoval ligandy, které se nacházejí v okolí funkčního místa proteinu. Jako poslední možnost



Obrázek 5.2: Definice vstupu uživatelem: (a) vložení identifikátoru PDB do databáze proteinových struktur (Protein Database Bank); (b) nahrání souboru ve formátu PDB nebo mmCIF [49]

výběru polohy výchozího bodu nástroj poskytuje střed uživatelem vybraných aminokyselin v sekvenci proteinu. Identifikované souřadnice výchozího bodu lze vždy upravit ruční optimalizací souřadnic [49].



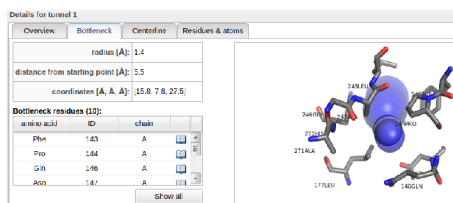
Obrázek 5.3: Výběr počátečního bodu a nastavení parametrů pro nástroj CAVER: (a) výchozí bod pro identifikaci tunelů může být vybrán z polohy: katalytické kapsy, proteinové kapsy, ligandu (možnost je deaktivována při absenci ligandů ve struktuře) a středu uživatelem vybraná sekvence proteinu; (b) nastavení parametrů nástroje CAVER.

Detekci tunelů provádí program CAVER [14], který pomocí Dijkstrova algoritmu identifikuje ve Voroného reprezentaci proteinových struktur tunely s daným minimálním poloměrem. Podrobněji je tento nástroj popsán v sekci 4.1. Uživatelé mohou upravit několik vybraných důležitých (viz sekce 4.1.3) konfiguračních parametrů: (i) výběr částí struktury (residues considered for tunnel calculation), jež bude CAVER brát v úvahu při analýze, aby bylo možné vyloučit ligandy, ionty a molekuly vody; (ii) minimální velikost (poloměr) hrdla (minimum probe radius); (iii) maximální vzdálenost odpovídajících si vrcholů tunelu z počátečního bodu k povrchu (shell depth); (iv) poloměr pláště (shell radius); (v) práh shlukování (clustering threshold); (vi) maximální vzdálenost (maximal distance), a (vii) požadovaný počáteční poloměr (desired radius) [49].

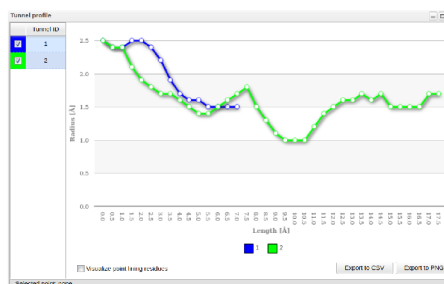
Posledním (nepovinným) krokem pracovního postupu je analýza transportu ligandů skrze detekované tunely. Pro analýzu transportu se využívá software CaverDock (viz sekce 4.2) [55]. Požadovaným vstupem pro kalkulace transportu ligandu jsou dvě informace. První část vstupu tvoří ligand, což je malá molekula, kterou lze nahrát v podobě souboru, textového popisu nebo ji nakreslit pomocí přítomného nástroje. Lze zadat i více ligandů současně (obrázek 5.4d). Za druhou část vstupu je považován výběr alespoň jednoho identifikovaného tunelu, který se použije jako cesta pro transport zvolených ligandů. Caver Web přidává ke

id	bottleneck radius [Å]	length [Å]	curvature	throughput
1	1.4	6.9	1.0	0.78
2	1.0	17.0	1.5	0.47

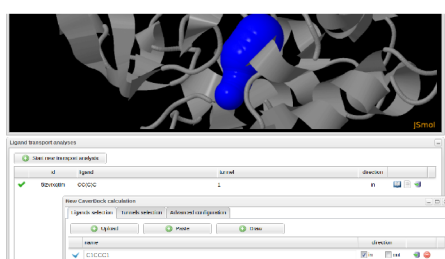
(a)



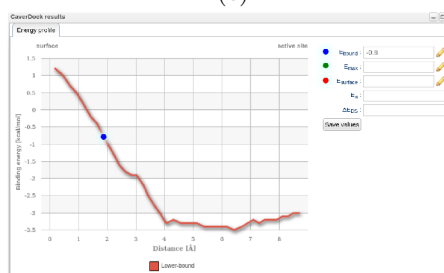
(b)



(c)



(d)



(e)

Obrázek 5.4: Zobrazení výsledků v nástroji Caver Web: (a) tabulka identifikovaných tunelů; (b) podrobnosti tunelu (informace o hrdle tunelu); (c) graf proměnlivosti šířky tunelu od počátečního místa až na povrch proteinu; (d) grafické zobrazení tunelu ve struktuře, seznam vypočítaných úloh softwarem CaverDock a zadání nové úlohy pro CaverDock; (e) energetický profil dokování ligandu (obrázky získány z aplikace Caver Web [49]).

každému atomu Gastaigerovy náboje a typy atomů kompatibilní s AutoDock Vina [38, 52] pomocí skriptů z balíčku MGLTools [39]. Poté se využije metoda Discretizer [21] k rozřezání tunelu na průřezy se zadanými rozestupy. Následně se spustí CaverDock, který provede iterativní dokování ligandu do každé části tunelu pomocí prostorově omezeného dokovacího algoritmu AutoDock Vina [49].

Po ukončení identifikace proteinových tunelů jsou uživateli zobrazeny výsledky v prohlížeči výsledků. Okno s výsledky obsahuje 3 části. První obsahuje tabulku s identifikovanými tunely (obrázek 5.4a), ve které jsou zobrazeny veškeré statistiky a informace o tunelu jako: délka, velikost hrdla, členitost a propustnost tunelu. Po otevření podrobného náhledu lze zjistit, z kterých aminokyselin je tunel zformován, které aminokyseliny vytváří jeho hrdlo (obrázek 5.4b) a jak se mění velikost hrdla daného tunelu (obrázek 5.4c). V druhé části lze daný tunel graficky zobrazit. V poslední sekci je zobrazen seznam vypočítaných dokovacích úloh (analýz transportu ligandu, obrázek 5.4d). Každý záznam disponuje daty získanými při analýze. Jsou to data pro animaci dokovacího (transportního) procesu ligandu do/z počátečního (uživatelé zvoleného) místa identifikovaných tunelů a data energetického profilu dokování ligandu (obrázek 5.4e).

Tyto kroky a prohlížeč výsledků jsou implementovány i v nové verzi aplikace. Pracovní protokol jednoduché analýzy statické struktury poskytuje rychlou analýzu, která prezentuje

základní vlastnosti proteinu v krátkém čase, a proto je stále relevantní i pro novou verzi aplikace s integrací molekulárních dynamik. Grafická implementace jednotlivých kroků bude znázorněna v dalších sekcích tohoto textu.

Kapitola 6

Návrh nové verze nástroje Caver Web

Cílem této práce je navrhnout Caver Web 2.0 – novou verzi webové aplikace, která rozšiřuje původní aplikaci Caver Web 1.0 o nové vlastnosti a zachovává charakteristiky původní verze – poskytnutí relevantních výsledků bez nutnosti hlubší znalosti integrovaných nástrojů. Vše je zprostředkováno prostřednictvím jednoduchého a interaktivního uživatelského rozhraní.

Motivací pro vývoj nové verze aplikace byla možnost rozšíření aplikace o výpočet molekulárních dynamik (viz kapitola 3) a metod pro získání různých konformací proteinů (viz sekce 3.6). Tyto přístupy generují velké množství dat, které je velmi obtížné manuálně zpracovat, a proto musí projít automatizovaným předzpracováním. Předzpracovaná data je nutné zobrazit v uživatelském rozhraní tak, aby byla přehledná pro širokou skupinu uživatelů (především z vědeckého prostředí).

Kapitola bude rozdělena do čtyř logických celků. V první části bude uveden výčet a porovnání existujících řešení. V druhé části budou popsány nové pracovní protokoly, které je možné v aplikaci provést. Ve třetím celku bude uvedeno rozšíření výpočetního jádra používaného v Loschmidtových laboratořích a v poslední části kapitoly se zaměřím na návrh nového uživatelského rozhraní.

6.1 Existující řešení

Nástroj Caver Web 2.0 tvoří novou generaci již existujícího nástroje. Nová verze aplikace přináší rozšíření o molekulární dynamiky. Integrace molekulárních dynamik v rámci webové aplikace vytváří zcela novou kategorii nástrojů a v současné době není známa aplikace se stejnými možnostmi.

V rámci tvorby předchozí verze byl proveden průzkum již existujících řešení v oblasti analýzy proteinových struktur a proteinových tunelů. Srovnání existujících řešení obsahuje tabulka 6.1, která byla převzata z článku [49] a byla rozšířena o novou porovnávanou kategorii.

6.2 Návrh nového workflow

Při návrhu nového pracovního postupu bylo potřeba definovat, které nové kroky bude nutné zpřístupnit. Během provozování první verze Caver Webu byly sbírány podněty a připomínky

Software	Analýza tunelů a kanálů			Analýza transportu ligandu			Integrace molekulárních dynamik	Reference
	Vstup	Podporováno	Výběr startovacího bodu	Podporováno	Zdroj ligandu	Výstup	Podporováno	
Caver Web 2.0	PDB ID ^a , PDB/CIF soubor ^b	Ano	katalytická kapsa, kapsa, ligandy, výběr sekvence, souřadnice katalytická kapsa,	Ano	ZINC15, soubor uživatele, kreslení	tunely/kanály, trajektorie proteinu, trajektorie ligandu, energetický profil	Ano	tato práce
Caver Web	PDB ID ^b , PDB/CIF soubor ^b	Ano	kapsa, ligandy, výběr sekvence, souřadnice	Ano	ZINC15, soubor uživatele, kreslení	tunely/kanály, trajektorie ligandu, energetický profil	Ne	[49]
MolAxis	PDB ID, PDB/CIF soubor	Ano	největší dutina, souřadnice	Ne	^d	tunely/kanály	Ne	[62]
MoleOnline	PDB ID ^c , PDB/CIF soubor ^c	Ano	katalytická rezidua, rezidua, kapsa, souřadnice, výběr sekvence	Ne	^d	tunely/kanály	Ne	[45]
BetaCavityWeb	PDB ID, PDB soubor	Ano	není vyžadováno	Ne	^d	tunely/kanály	Ne	[29]
PoreWalker	PDB soubor	Ano	není vyžadováno katalytická rezidua,	Ne	^d	kanály	Ne	[44]
ChExVis	PDB ID, PDB soubor	Ano	rezidua, heteroatomové záznamy	Ne	^d	tunely/kanály	Ne	[37]
MoMA-LigPath ^a	PDB soubor	Ne	^d	Ano	součástí PDB souboru	trajektorie ligandu	Ne	[16]

Tabulka 6.1: Porovnání existujících řešení v doméně analýzy proteinových struktur a transportu ligandu (převzato z [49] a upraveno).

^a Webový server SLITHER pro analýzu transportu ligandu nebyl v době psaní textu přístupný.

^b Biologické jednotky detekovány pomocí skriptu MakeMultimer.

^c Biologické jednotky načteny z databáze PDBe [24].

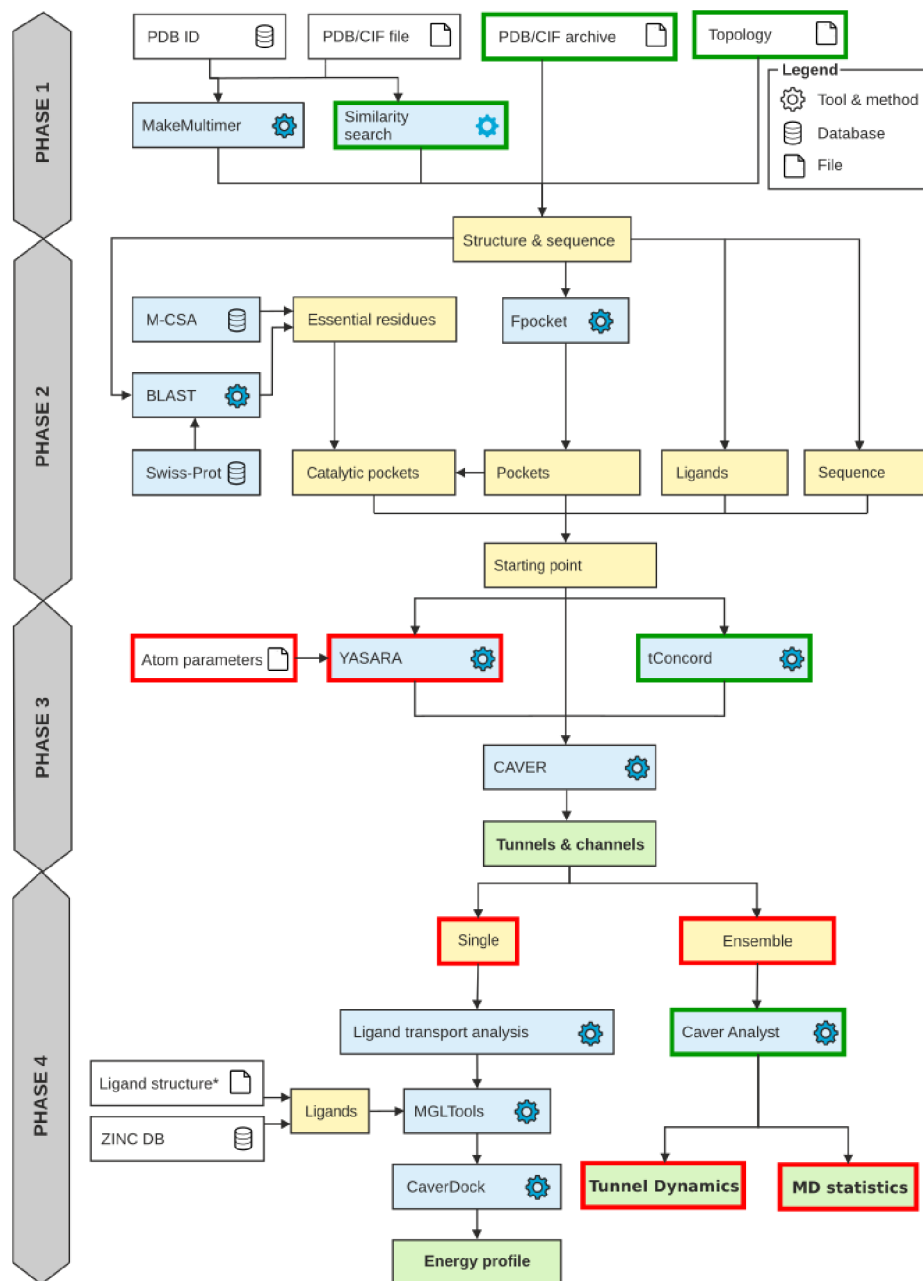
^d Není možné provést.

z řad členů vědecké komunity. Po rozšíření kapacit výpočetního clusteru byly tyto připomínky zpracovány a posloužily k definici nových vlastností aplikace. Z analýz, které provedly odborníci z Loschmidtových laboratoří, byl určen hlavní cíl nové verze aplikace, a to zpracování molekulárních dynamik. Dalším pozorovaným rysem využívání aplikace bylo porovnávání identifikovaných tunelů v rámci velmi podobných struktur, kdy byla pro každou strukturu spuštěna samostatná úloha, byly separátně určeny počáteční body identifikace tunelů a nalezené tunely byly jednotlivě analyzovány a porovnávány.

Z výše popsanych poznatků bylo jasné, že rozšířením projde většina částí aplikace, a to vstupy, výpočetní jádro a prezentace výsledků. Po provedení počátečních úvah bylo ve spolupráci s doménovými experty vytvořeno nové workflow aplikace (obrázek 6.1), které reflektuje všechny předpokládané pracovní postupy v aplikaci.

V oblasti vstupů aplikace (phase 1 v obrázku 6.1) došlo k nárůstu variant. Jelikož nová verze analyzuje molekulární dynamiky, a proto je plánováno umožnit nahrávání již vypočítaných trajektorií. Současně se předpokládá s podporou porovnávání více struktur v jedné úloze, kdy uživatel nahraje více souborů v archivu, nebo jeden po druhém. V případě, že uživatel disponuje pouze jednou strukturou a aplikace mu nabídne k porovnání nalezené podobné struktury (phase 1 – similarity search).

Poslední upravovanou částí je prezentace výsledků. Při výpočtu nových konformací proteinu se generuje velké množství snímků (snapshotů), které je vhodné popisovat souhrnnými statistikami. Za tímto účelem byla vytvořena zcela nová větev workflow (phase 4 – ensem-



Obrázek 6.1: Návrh nového workflow aplikace. Červeně jsou zvýrazněny nově přidané prvky. Zeleně jsou znázorněny prvky, které byly navrženy jako cíle pro novou verzi aplikace, ale jejich implementace nebyla součástí této práce.

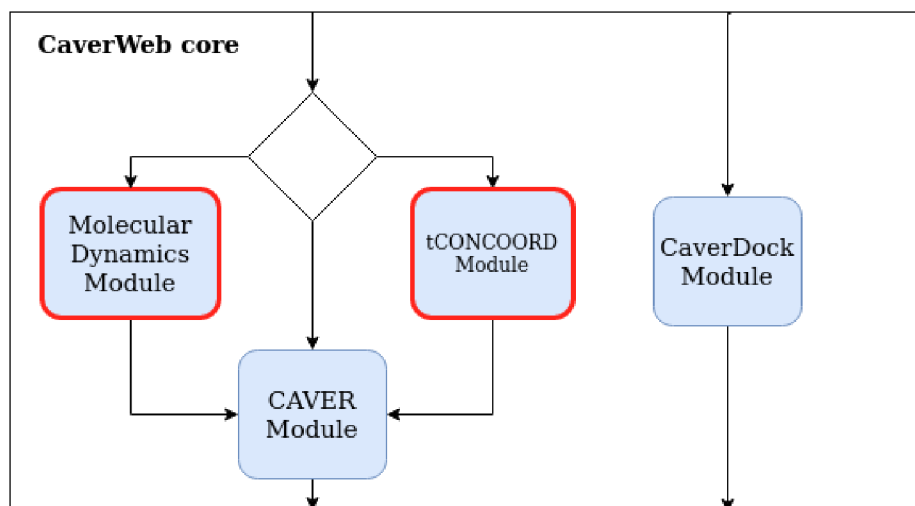
ble), která nabídne přehled statistik pro celou simulaci a umožní výběr nejvhodnějších snímků pro podrobnější analýzu.

S ohledem na náročnost implementace všech nových vlastností zmíněných výše bylo pro implementační část práce vybráno pouze to nejdůležitější, čímž je automatický výpočet molekulárních dynamik. Grafický přehled všech vlastností nové verze je zvýrazněn tučným ohraničením na obrázku workflow 6.1, přičemž jsou barevně odlišeny rozšíření přímo implementované v této práci (červeně) a rozšíření plánovaná k implementaci v budoucnu (zeleně).

Při návrhu nového pracovního postupu byl kladen důraz na striktní rozdělení jednotlivých částí (fází). V novém návrhu je stále podporována původní funkčnost aplikace, která umožňuje rychlou analýzu statické struktury se ziskem zběžného přehledu o vlastnostech proteinu.

6.3 Návrh rozšíření výpočetního jádra

V případě nástroje Caver Web existující jádro obsahuje implementaci pro většinu výpočetních modulů. Jsou to implementace modulů, které zapouzdřují práci s nástroji: CAVER a CaverDock. Navržené rozšíření obsahuje moduly pro výpočet proteinových konformací pomocí molekulárních dynamik a pomocí metody tCONCOORD (obrázek 6.2).



Obrázek 6.2: Grafické znázornění zapojení nově implementovaných modulů ve výpočetním jádru aplikace Caver Web. Červeně jsou zvýrazněny nově přidané prvky.

Tyto moduly budou podmíněně zapojeny (jen v případě výpočtu proteinových dynamik) do výpočetního schématu před identifikací tunelů, pro něž budou jejich výstupy představovat analyzované struktury. Rozšíření jádra o nové komponenty vyžaduje díky modulární infrastruktuře provedení následujících kroků: (i) implementaci vybraných modulů; (ii) rozšíření třídy zastřešující výsledky úlohy o možnost uchování většího počtu snímků; (iii) instalaci vybraných nástrojů do výpočetního clusteru; (iv) vytvoření spouštěcích skriptů pro dané nástroje.

Při hledání vhodného nástroje pro výpočty molekulární dynamiky se nejprve uvažovalo o využití nástroje Amber [13], který je brán jako state-of-the-art software. Bohužel se však

jedná o komerční software a jeho vývojáři bohužel nebyla udělena licence k jeho používání v rámci webové služby.

Jako alternativa k nástroji Amber [13] byl zvolen nástroj Yasara [31], jehož autoři nám poskytli licenci. Kvůli zachování vysoké kvality simulací byla při volbě jiného nástroje provedena rešerše literatury a nalezeny články potvrzující jeho výkon ve srovnání s konkurencí [30].

6.4 Návrh nového uživatelského rozhraní

Aplikace Caver Web je tvořena jednoduchým uživatelským rozhraním, které je logicky rozděleno do čtyř sekcí. Každá sekce vede uživatele k jedné konkrétní akci, která je nezbytná pro správné spuštění odpovídajících nástrojů. V první sekci uživatel vkládá proteinovou strukturu, kterou chce analyzovat. V druhé sekci se uživatel zaměřuje na určení počátečního bodu pro hledání tunelů. Třetí sekce nabízí přehled z probíhajících kalkulací a poslední čtvrtá sekce poskytuje přehled výsledků, jako jsou identifikované tunely a analýzy transportu ligandů.

Při návrhu nového uživatelského rozhraní byl kladen důraz na zachování původního stylu aplikace, které se vyznačuje svou přehledností a přímočarostí. Současně však po uživateli vyžadujeme či mu dovolujeme provést mnohem více akcí jak ve vstupní fázi, tak i v přehledu výsledků. Z tohoto důvodu rozdělujeme jednotlivé fáze do dílčích podkroků tak, aby uživatel nebyl přehlcen požadavky na jeho akce. Jednotlivé komponenty a rozložení uživatelského rozhraní byly navrženy a pravidelně konzultovány se zkušenými odborníky z Loschmidových laboratoří. Výsledný návrh rozložení je prezentován v této části kapitoly.

6.4.1 Zadávání vstupních dat

V sekci zadávání vstupních dat je nově vyžadováno několik nových akcí. V této části během interakce s uživatelem je cílem zjistit, který pracovní protokol chce uživatel provést a která vstupní data má k dispozici.

Input section

What do you wish to analyze?

Tunnels

Channels

(a)

Input

Single structure

Multiple structures

(b)

Obrázek 6.3: Návrh dotazování ve vstupní části: (a) cíl analýzy; (b) specifikace vstupních dat.

Celý krok zadávání vstupních dat je rozdělen do čtyř podkroků, které se uživateli zobrazují jeden po druhém, vždy po zodpovězení požadovaných otázek nebo po dodání po-

třebných dat. Jako první je položen dotaz, o který typ analýz má uživatel zájem (obrázek 6.3a). Může si vybrat mezi analýzou proteinových tunelů (například při zkoumání enzymů) nebo proteinových kanálů (například při analýze membránových proteinů). Po zodpovězení otázky je zobrazen druhý podkrok, který slouží k zadání vstupních dat. Uživatel specifikuje, jaký typ a množství dat má k dispozici (obrázek 6.3b). K výběru jsou dvě možnosti – jedna struktura (single structure) a více struktur (multiple structures).

Zatímco při zadání libovolné možnosti v podkroku výběru cíle analýzy je vždy zobrazena stejná komponenta pro výběr vstupních dat, po výběru typu vstupních dat je uživateli zobrazena jiná komponenta v závislosti na jeho volbě. Při zvolení jedné struktury je zpřístupněna komponenta (obrázek 6.4a), ve které lze zadat zkoumanou strukturu pomocí jejího identifikátoru v databázi Protein Data Bank (PDB) [9], nebo ji nahrát ve podporovaných formátech PDB a mmCIF. Po zadání struktury dojde ihned k automatickému vyhledání biologických jednotek a zobrazení jejich výčtu se základními informacemi a možností zobrazení podrobnějšího náhledu.

Režim více struktur pak podporuje několik způsobů jejich nahrání. Prvním z nich je poskytnutí struktur v jednotlivých souborech či výčtem jejich přístupových kódů do PDB databáze (obrázek 6.4b). Dále je možné struktury nahrát v archivu (obrázek 6.4c) nebo je vložit jako trajektorii v některém z nejčastěji používaných formátů molekulární dynamiky. Nahrání trajektorie je zpracováváno v odděleném případě, protože aplikace s ní bude pracovat v jiném režimu. Po splnění těchto kroků je umožněno přejít přímo k vyplnění informací o úloze.

Na rozdíl od nahrání více souborů, kdy autor jednoznačně určuje, co s daty chce dělat, při volbě jedné struktury tato informace stále není jasná. Autor totiž může chtít provést akce jako: (i) zpracování pouze jedné statické struktury (tedy práce v původním režimu aplikace); (ii) nalezení podobné struktury a porovnat jejich vlastnosti (uživatel nezná tyto proteiny); (iii) vypočítání trajektorie pomocí molekulární dynamiky nebo (iv) vytvoření alternativních konformací proteinu pomocí nástroje tCONCOORD.

Volbu mezi možnými scénáři uživatel provede prostřednictvím přepínače. Potvrzením volby jednoho souboru pokračuje nadále pouze vyplněním informací o úloze a naplňuje scénář realizovaný v předchozí verzi aplikace (obrázek 6.5). Komponenta pro vyplnění dat o úloze (kontakt na uživatele a jméno úlohy) je zobrazena vždy po vyplnění potřebných dat daného scénáře.

Input

Select PDB file

Source: PDB code
 Upload file

PDB ID:

	Molecule	Chains	Quaternary structure	
Biological units:	<input checked="" type="radio"/> MoleculeId	<input type="radio"/> ChainId	Structure type	<input type="button" value="Download"/>

Use original structure

(a)

Input

Upload multiple file
 Upload archive
 Upload trajectory

Select PDB file

Source: PDB code
 Upload file

PDB ID:

(b)

Upload multiple file
 Upload archive
 Upload trajectory

Upload file:

(c)

Obrázek 6.4: Vložení vstupních souborů do aplikace pomocí: (a) databázového identifikátoru proteinu; (b) komponenty pro nahrání více souborů; (c) archivu.

Output

The image shows a web interface for selecting an output scenario. At the top, there are four buttons: 'Single file' (highlighted in blue), 'PDB database', 'MD', and 'tCONCOORD'. Below these buttons is a rounded rectangular box titled 'User information'. Inside this box, there are two input fields: 'Job name:' and 'Email (optional):'. Below the input fields is a button labeled 'Go to the job settings'.

Obrázek 6.5: Volba scénáře pro zpracování statické struktury (běh původním režimu aplikace).

Zvolením prohledávání proteinové databáze (obrázek 6.6a) je poskytnut výčet nalezených struktur z databáze Protein Data Bank pomocí nástroje BLAST s požadavkem na 100% shodu v sekvenci s délkou odpovídající 50 až 100 % délky původní sekvence. Tabulka nalezených struktur obsahuje záznamy s identifikátorem proteinů, podobností, krátkým popisem a možností stažení souboru. Pomocí checkboxu si uživatel může následně vybrat, které proteiny budou zahrnuty do výpočtu.

Při volbě výpočtu molekulárních dynamik (obrázek 6.6b) se uživateli zobrazí komponenta, která mu nabízí možnost modifikovat důležité parametry. Zároveň však budou všechny hodnoty ve výchozím stavu nastaveny tak, aby byly co nejuniverzálnější a bylo tak nutné je měnit pouze ve specifických případech. Konkrétně se jedná se o parametry nastavující teplotu, délku simulace, hodnotu pH, velikost kroku simulace (rychlost) a volbu silového pole. Stejně jako u dalších komponent je i zde využito rozdělení parametrů na základní a pokročilý mód, aby nedocházelo ke zbytečnému matení nezkušených uživatelů. V základním režimu tak budou viditelné pouze parametry pro nastavení teploty a délky simulace.

Poslední možností je volba výpočtu trajektorie proteinu pomocí nástroje tCONCOORD. Tento nástroj požaduje pouze zadání vstupní struktury, se kterou poté interně pracuje. Volba výpočtu pomocí nástroje tCONCOORD, proto nabídne pouze vyplnění informací o uživateli a názvu úlohy stejně jako v případě volby jednoho souboru.

Output

<input type="checkbox"/>	PDB ID	Match [%]	Length match [%]	Description	
<input type="checkbox"/>	1cv2	100	98		Download
<input type="checkbox"/>	2wut	99	86		Download

(a)

Output

Molecular dynamics settings

Temperature:

Duration:

pH:

Speed:

Force fields:

[Reset settings](#)

(b)

Obrázek 6.6: Grafický návrh nových komponent poskytující: (a) výběr podobných struktur k vzájemnému porovnání; (b) nastavení parametrů pro běh simulace molekulárních dynamik.

6.4.2 Výběr počátečního bodu pro výpočet tunelů

Stejně jako v předchozí verzi je na výběr počátečního bodu pro výpočet proteinových tunelů kladen značný důraz, protože špatně zvolený počáteční bod může vést k jejich špatné identifikaci. Stránka výběru počátečního bodu zachovává rozložení a logiku komponent z předchozí verze. Skládá se ze tří komponent (obrázek 6.7). První komponenta slouží k usnadnění výběru počátečního místa a nabízí stejné možnosti jako původní verze – katalytické kapsy, kapsy, ligandy a úseky sekvence. Více informací k jednotlivým z nich stejně lze nalézt v kapitole 5.

Všechny vybrané varianty se interaktivně zobrazují ve vizualizační komponentě, kde si je uživatel může podrobněji prohlédnout. Nejdůležitějším objektem bude koule označující počáteční bod pro výpočet tunelu.

Při výběru libovolného objektu (kapsy, ligandu nebo úseků sekvence) se uživateli zobrazí červený bod ve vizualizační komponentě, který označuje výsledný počáteční bod, který bude zvolen pro výpočet tunelů. Jeho souřadnice se zobrazí v třetí komponentě, ve které je uživateli umožněna jejich manuální editace.

Start from

	Catalytic pockets	Pockets	Ligands	Sequence
	ID	Relevance score	Volume	Drugability
<input checked="" type="radio"/>	1	100	98	0.86
<input type="radio"/>	2	99	86	0.08

Visualization

Coordinates of starting point

x: y: z:

Obrázek 6.7: Návrh výběru počátečního bodu pro výpočet proteinových tunelů. Skládá se ze tří částí: (i) přehled objektů, pomocí kterých lze vybrat počáteční bod; (ii) vizualizace struktury i se zobrazenými prvky; (iii) možnost manuální úpravy souřadnic počátečního bodu.

6.4.3 Nastavení nástroje CAVER a interakce s uživatelem během výpočtu

Poslední stránkou, kterou uživatel před odesláním úlohy vidí, je stránka pro nastavení parametrů nástroje CAVER (obrázek 6.8a). Obsahuje celkem dvě sekce. V první si uživatel volí, které molekuly nacházející se ve struktuře chce do výpočtů tunelů zařadit. Tato volba ovlivňuje, co bude ze vstupního souboru odstraněno a nebude tak představovat umělou překážku při hledání tunelů. Základní volba počítá s aminokyselinami, ale ponechává na uživateli, jestli do výpočtů chce začlenit i vodu (HOH).

Druhá sekce obsahuje nastavení základních parametrů nástroje CAVER (více popsány v sekci 4.1.3). Parametry jsou opět rozděleny na základní a pokročilý mód. U ostatních nastavení bylo pozorováno, že je uživatelé mění velmi málo a pro jejich správné nastavení je nutné hlubší pochopení funkčnosti nástroje CAVER. Po uvážení všech parametrů uživatel odesílá úlohu ke zpracování na server.

Během výpočtu úlohy, který může nově trvat i desítky hodin, jsou uživateli zobrazeny logovací informace z probíhající úlohy (obrázek 6.8b). Po doběhnutí všech výpočtů jsou ke stažení nabídnuta všechna data, jež byla získána během kalkulací a data informující

o průběhu výpočtů molekulárních dynamik či proteinových tunelů. Po stisknutí tlačítka pro zobrazení výsledků je uživatel přemístěn na stránku výsledků.

The image shows two parts of a web interface. Part (a) is a settings panel titled "Residues considered for tunnel calculation" with two checkboxes: "standard amino acids" (checked) and "HOH" (unchecked). Below it is a "Caver settings" section with input fields for: "Minimum probe radius" (0.9), "Shell depth" (4), "Shell radius" (3), "Clustering threshold" (3.5), "Maximal distance" (3), and "Desired radius" (5). A "Reset settings" button is at the bottom right. Part (b) shows "Job info" with "Job name: Experiment 1", "Job ID: scaassd1", and "Status: Running molecular dynamics simulations". Below is an "Analysis" section with a "Show analysis" button. The "Download" section contains buttons for "Raw data", "Caver log", "Caver configuration", "MD configuration", and "MD log".

(a)

(b)

Obrázek 6.8: Návrh komponent pro nastavení nástroje Caver (a) a pro interaktivní informování uživatele o stavu běžících výpočtů (b).

6.4.4 Prezentace výsledků

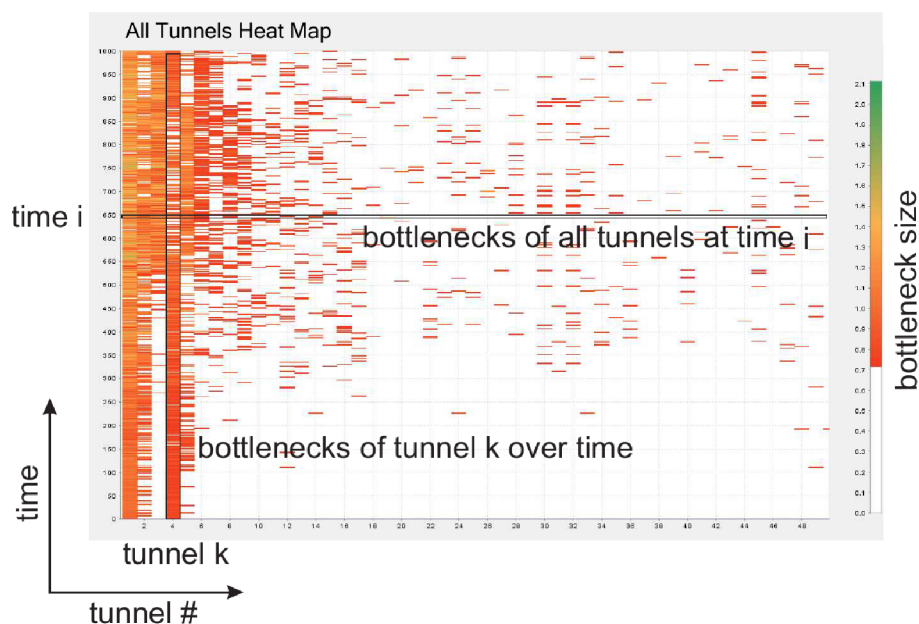
Zavedením výpočtu molekulárních dynamik významně narůstá množství výsledných dat, proto je nutné zavést nové techniky jejich prezentování. Pracovní postup proteinových inženýrů v této oblasti spočívá ve výběru úzké skupiny kandidátních tunelů pro pozdější podrobné analýzy. Je tedy nutné z celé simulace a stovek identifikovaných tunelů vybrat takové tunely, které jsou v čase stabilní (nejčastěji otevřené) a mají co největší propustnost. Vyfiltrované tunely jsou dále zkoumány v jejich dalších parametrech, jako je zkroucení a proměnlivost šířky v celé délce od aktivního místa až po povrch proteinu. Tunely jsou tak vyříděny do jednotek kandidátů, u kterých se zkoumají možnosti zlepšení jejich vlastností a navrhují se případné mutace v sekvenci proteinu, nebo se volí vhodnější ligand [12].

Velké množství generovaných snímků proteinové trajektorie (tzv. snapshotů) činí tuto manuální analýzu proměnlivosti proteinových tunelů v reálném čase velice obtížnou až přímo nemožnou.

Průzkum možností vizualizace dat z molekulární trajektorie

Za účelem umožnění analýz trajektorie byl proveden průzkum existujícího řešení zabývajícího se analýzou molekulárních dynamik. K podrobné analýze byl vybrán nástroj MoleCollar [12]. Autoři nástroje působí v brněnské vědecké skupině, která se zabývá vizualizací biologických dat. MoleCollar je desktopová aplikace, která analyzuje trajektorie proteinových struktur. Jednou z nabízených funkcí je identifikace a analýza proteinových tunelů. Tyto analýzy přesně odpovídají požadavkům našeho řešení, a proto bylo rozhodnuto o jejich implementaci v aplikaci Caver Web 2.0.

Pro implementaci byly zvoleny dvě metody prezentace výsledných dat: (i) All tunnels Heat Map (tepelná mapa všech tunelů) a (ii) Single tunnel Heat Map (tepelná mapa jednoho tunelu v rámci celé simulace).

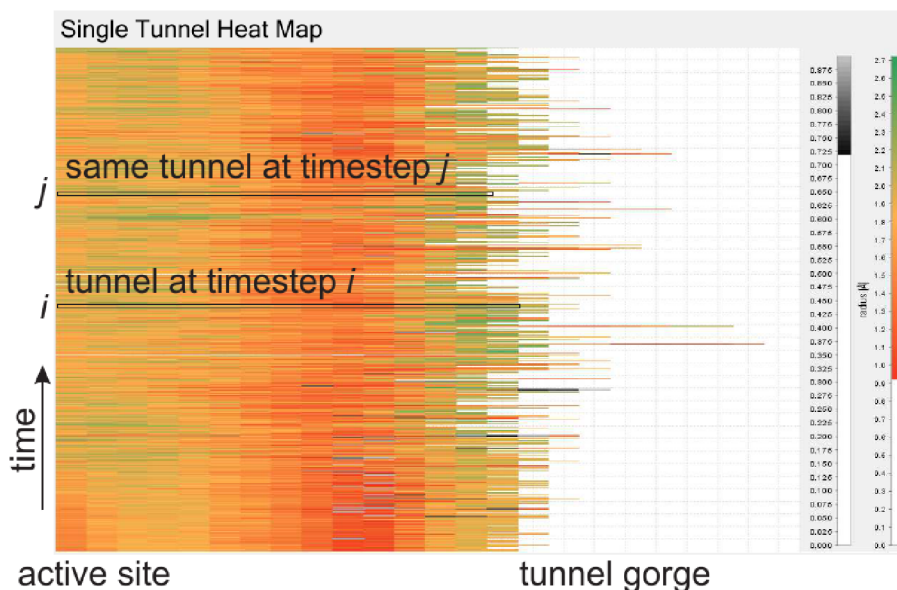


Obrázek 6.9: Tepelná mapa všech tunelů zobrazuje variabilitu velikostí hrdel jednotlivých tunelů napříč celou dobou simulace. Bílá barva znamená, že tunel nebyl v daném snímku se zadanou hodnotou minimálního poloměru hrdla klasifikován (převzato z [12]).

Tepelná mapa všech tunelů (obrázek 6.9) slouží k zobrazení všech tunelů, které byly během běhu simulace identifikovány. Tunely jsou aproximovány velikostí jejich hrdla. To znamená, že jeden tunel v konkrétním čase je reprezentován jedním bodem (barevným obdélníkem). Barevná výplň bodů je škálována podle šířky tunelu, přičemž bílá barva označuje, že tunel v daném čase a ve vybraném snímku nesplnil podmínky identifikace (hodnota velikosti jeho hrdla byla menší než uživatelem definovaný parametr) [12]. Získaná tepelná mapa usnadňuje výběr kandidátních tunelů pro podrobnější analýzu, které jsou v průběhu času nejčastěji otevřeny (jsou stabilní). Tyto tunely jsou vhodné k následné analýze transportu ligandů do aktivních míst proteinů. K analýze je vhodné vybrat snímek tunelu v čase, ve

kterém disponuje nejmenší velikostí hrdla, protože velikost hrdla významně ovlivní propustnost celého tunelu.

Po výběru skupiny kandidátů z tepelné mapy všech tunelů lze jednotlivé tunely zobrazit v tepelné mapě jednoho tunelu (obrázek 6.10). Tato datová reprezentace na rozdíl od předchozí neaproximuje tunely jednou hodnotou, ale zobrazuje proměnlivost jednoho tunelu v celé jeho délce od aktivního místa po povrch proteinu [12]. Tento přehled nám umožňuje identifikovat úseky tunelu, které mohou být modifikovány. Například při záměru zvýšit výtěžek z enzymu se snažíme rozšířit užší místa tunelu a při snaze tento protein blokovat v jeho činnosti hledáme způsoby jak toto místo zablokovat nebo ještě více zúžit.



Obrázek 6.10: Tepelná mapa jednoho tunelu zobrazuje proměnlivost šířky tunelu v celé jeho délce (od počátečního místa až po dosažení povrchu) napříč celou dobou simulace. Bílá barva znamená, že tunel nebyl v daném snímku se zadanou hodnotou minimálního poloměru hrdla klasifikován (převzato z [12]).

Návrh stránky zobrazující výsledky úlohy

Po definování nových komponent pro zobrazení dat z molekulárních dynamik byla navržena stránka pro prezentaci výsledků úlohy. Stránka se skládá ze dvou záložek, které od sebe oddělují data popisující souhrnně celý běh simulace a data analyzující jednu statickou strukturu nebo snímek ze simulace.

První záložka obsahuje celkové statistiky ze všech snímků úlohy. Tato záložka je zobrazena na obrázku 6.11a. Skládá se ze čtyř komponent, které byly za sebe poskládány tak, aby vedly uživatele v očekávaných krocích, tedy: (i) analýza celého průběhu a identifikace skupin tunelů jako kandidátů pro podrobnější analýzu; (ii) zkoumání a vyřazení nevhodných kandidátů; (iii) analýza vybraných kandidátů v záložce jedné struktury.

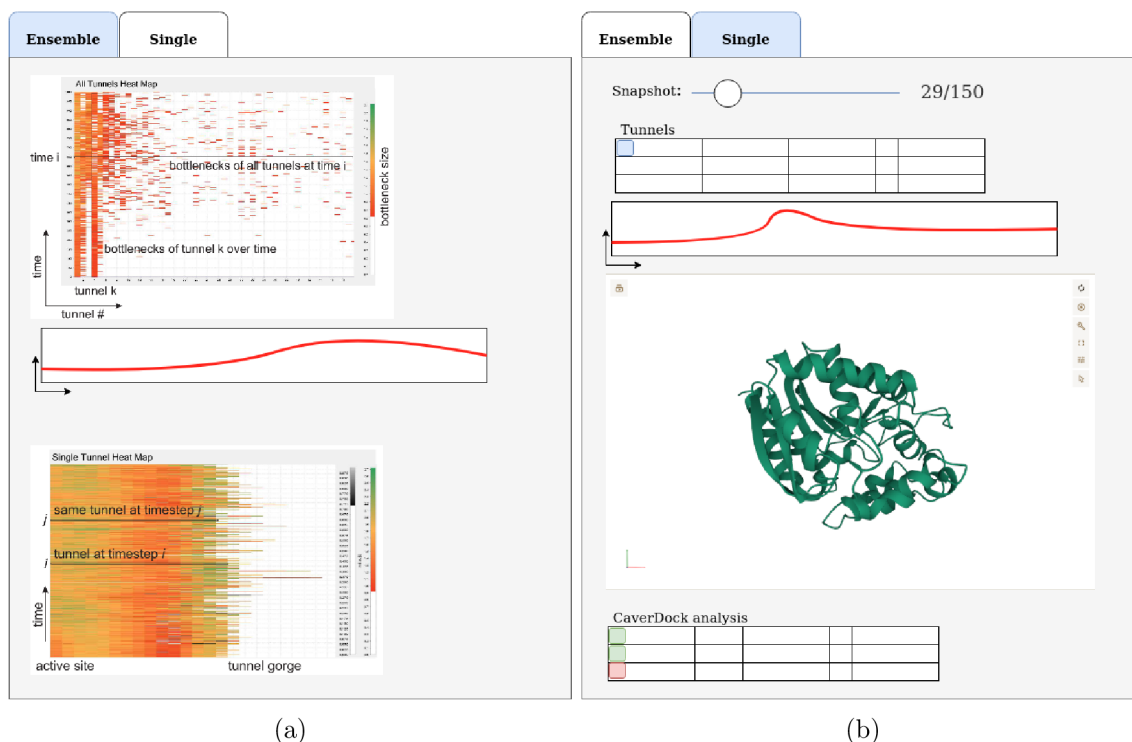
K celkovému přehledu statistik je využito komponenty tepelné mapy všech tunelů, jež zobrazí místa, ve kterých dochází k hledanému chování tunelů nejčastěji. Tento přehled je doplněn o komponentu statistik celé simulace. Tvoří ji tabulka s příloženým grafem. Jednotlivé řádky tabulky obsahují záznamy o shlucích tunelů (clusterech). Za shluk je v aplikaci

označován výskyt jednoho tunelu v různých časových snímcích trajektorie. Shluky jsou vytvořeny pomocí nástroje CAVER. Záznam tabulky obsahuje statistický popis shluku jako je: (i) počet snímků, ve kterých byl zkoumán; (ii) počet snímků, ve kterých byl identifikován; (iii) průměrná velikost a průměrná hodnota maximálních a minimálních velikostí hrdla tunelu; (iv) směrodatné odchylky hodnot týkajících se hrdla tunelu; (v) propustnost tunelu.

Vybraní kandidáti jsou dále vytříděni pomocí komponenty pro tepelnou mapu jednoho tunelu. Z těchto dat si uživatel určí snímky, které chce podrobněji zkoumat, a přesune se do druhé záložky. Záložka statistik není zpřístupněna, pokud aplikace neobsahuje výpočet molekulární trajektorie (byla spuštěna v režimu analýzy statické struktury, nebo porovnání struktury).

Záložka analýzy statické struktury je zobrazena vždy. Oproti původní verzi obsahuje jednu novou komponentu, která se podmíněně zobrazuje pouze v případě analýzy molekulární trajektorie. Jedná se o komponentu pro výběr snímku a nachází se hned na začátku záložky na obrázku 6.11b.

Další komponentou je tabulka s přehledem identifikovaných tunelů v konkrétním snímku. Uživatel může zobrazit podrobné informace o tunelu a prohlédnout si proměnu šířky tunelu od počátečního místa až k povrchu proteinu. K zobrazení proměnlivosti šířky je použit graf pod tabulkou. Vybrané tunely jsou interaktivně zobrazeny ve vizualizační komponentě. Poslední komponentou je výčet provedených kalkulací transportu ligandu pomocí nástroje CaverDock s možností animace transportu ve vizualizační komponentě nebo stažení v podobě PyMol sezení.



Obrázek 6.11: Návrh rozložení stránky zobrazující výsledky úlohy obsahuje dvě záložky: (a) statistika všech snímků simulace a (b) analýza jednoho vybraného snímku nebo statické struktury.

Kapitola 7

Implementace nástroje Caver Web

Aplikace Caver Web je velice komplexní a zapouzdřuje velké množství nástrojů, čemuž odpovídá i jeho struktura. Skládá se z celkem ze tří hlavních komponent – frontendu uživatelského rozhraní, backendu uživatelského rozhraní a výpočetního jádra. Frontend a backend uživatelského rozhraní jsou zodpovědné za interakci s uživatelem, sbírání jejich vstupů a zobrazování výsledků. Výpočetní jádro pak zajišťuje uskutečnění samotných výpočtů a zpracování jejich výstupních dat. Všechny komponenty budou blíže popsány v následujících kapitolách. Jelikož aplikace vychází již z existující verze, přebírá z ní určité části implementace.

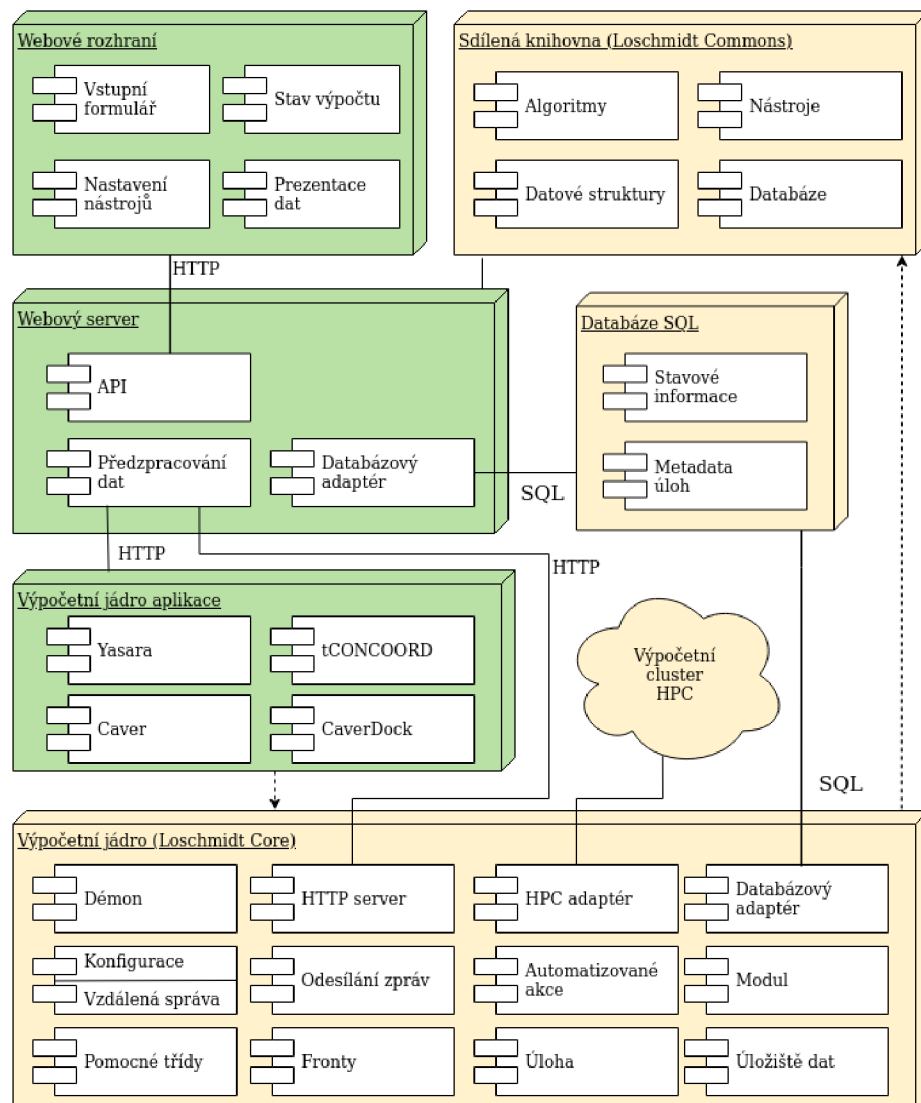
7.1 Architektura aplikace

Architektura aplikace byla navržena a sestavena podle pravidel pro tvorbu aplikací založených na platformě Loschmidt Core (obrázek 7.1). Výpočetní jádro tvoří samostatnou jednotku, která je plně oddělena od webové aplikace s grafickým rozhraním. Nové úlohy a informace o jejich stavu se předávají s využitím databáze MySQL a definičních XML souborů, které obsahují uživatelské vstupy a konfigurační parametry. Výsledky jsou pak z jádra poskytovány uživatelskému rozhraní standardním HTTP protokolem. Tato vlastnost umožňuje libovolně měnit dílčí části systému bez nutnosti dalších potřebných úprav. Současně je možné navýšit počet komponent, které tyto soubory produkují, čímž nabízí prostor k efektivní škálovatelnosti systému [57].

Jedinou věcí, kterou musí komponenty interagující v systému respektovat, je formát dat, jež slouží k popisu biologických dat. Na obrázku 7.2 je zobrazen datový model, který je sdílený mezi komponentami a slouží k uchování zpracovaných informací o tunelech a jejich parametrech. Proteinové tunely (třída `Tunnel`) jsou tvořeny sekvencí koulí (sfér), které jsou reprezentovány třídou `Centerline`. Tunel, který se v čase mění, je držen ve shluku (třída `TunnelCluster`), aby bylo možné propojit tentýž tunel v odlišných snímcích (třída `Snapshot`). Každý tunel je lemován sekvencí aminokyselin (třída `Residue`) a jeho nejužší část reprezentuje třída `Bottleneck`. Aminokyselina je složena z atomů, které jsou reprezentovány stejnojmennou třídou `Atom`.

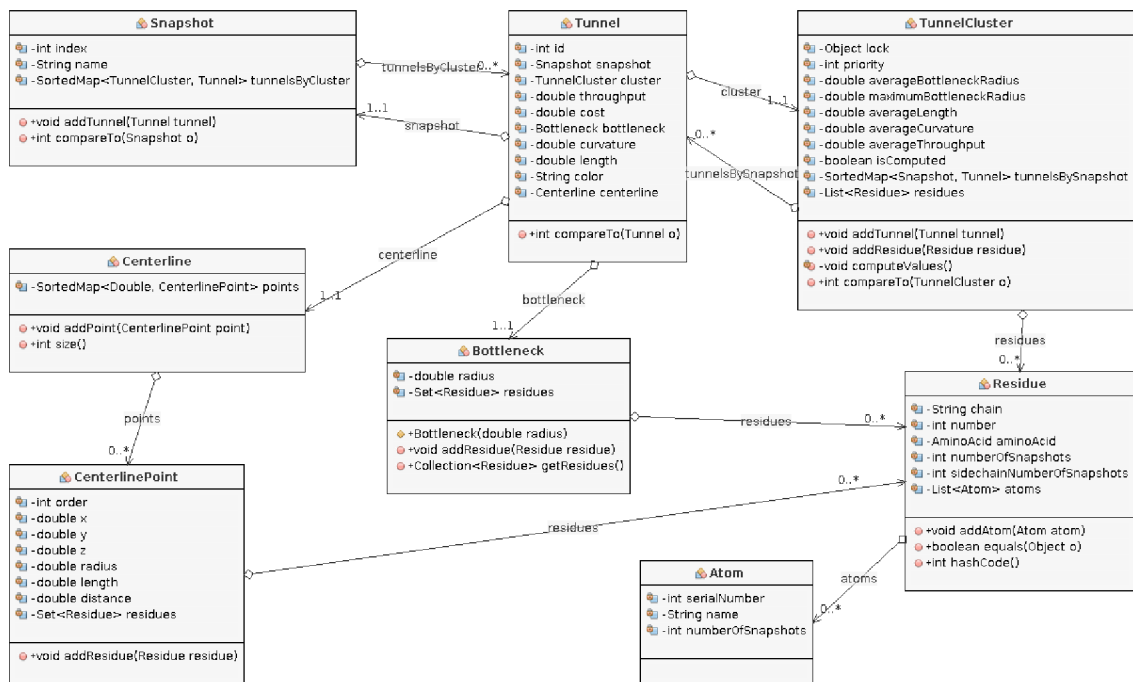
7.2 Výpočetní jádro

Výpočetní jádro je platforma vyvinutá v Loschmidtových laboratořích [58], která zajišťuje automatizované spouštění výpočetních úloh na základě specifikace. Jeho implementaci lze nalézt na přiloženém cd v archivu `core.zip`. Je implementovaná v jazyce Java a slouží jako



Obrázek 7.1: Architektura nástroje, zeleně jsou vyznačeny rozšířené nebo nově implementované části [57].

robustní základ pro bioinformatické webové aplikace, kterým poskytuje širokou plejádu základních funkcí a umožňuje tak jejich programátorům se soustředit výhradně na implementaci bioinformatických analýz. Mezi tyto funkce pak patří například: (i) spravování zadaných úloh; (ii) přiřazení potřebné diskové a výpočetní kapacity; (iii) odesílání e-mailové zprávy o změně stavu výpočtu; (iv) obsluha databáze; (v) spouštění, řazení a kontrola závislostí modulů; (vi) zajištění sdílení výsledků mezi moduly. Hotové výpočetní jádro pak běží jako samostatná služba a uživatelské rozhraní v něm spouští jednotlivé úlohy dle zadání uživatele. Úlohy jsou zadané pomocí souboru v datovém formátu `xml` (příklad ve výpisu 7.1), ve kterém jsou uvedeny základní informace o úloze, krocích výpočtu (modulech) s odpovídajícími parametry a vstupních souborech. Jádro tento definiční soubor zpracuje. Poté na jeho základě provede všechny výpočetní kroky a výsledky zpřístupní zpět uživatelskému rozhraní [58].



Obrázek 7.2: Datová reprezentace výsledných biologických dat v jádře aplikace [57].

```

<job id="example">
  <info>
    <title>1cv2 - molecular dynamics</title>
    <email/>
    <date>2022/04/12 12:30:34</date>
    <isPrivate>false</isPrivate>
  </info>
  <global>
    <structure>
      <source>PDB</source>
    </structure>
  </global>
  <modules>
    <module name="yasara"/>
    <module name="caver">
      <startingPoint origin="COORDS">
        <x>17</x>
        <y>10</y>
        <z>25</z>
      </startingPoint>
      <caver>
        <param key="desired_radius">5</param>
        ...
      </caver>
    </module>
  </modules>
</job>

```

```

    </module>
  </modules>
  <inputFiles>
    <file key="pdb">http://caverweb.cz/api/source-file/jobXYZ</file>
  </inputFiles>
</job>

```

Výpis 7.1: Příklad definice úlohy pro výpočetní jádro.

Implementace výpočetního jádra probíhala ve formě vytvoření chybějících modulů, které rozšířily již stávající implementaci analyzující statické struktury.

Definiční XML soubor úlohy obsahuje informace o vstupní proteinové struktuře, počátečním bodu pro výpočet proteinových tunelů a modulech, které mají být použity.

Jádro bylo potřeba rozšířit o výpočet dynamiky proteinových struktur. Do výpočetního clusteru byly přidány patřičné nástroje. Jedná se o programy Yasara [31] a tCONCOORD [47]. V obou případech je předpokládáným výstupem sekvence snímků (PDB souborů). Soubory jsou vyprodukovány a poskytnuty k výpočtu proteinových tunelů modulu nástroje CAVER. Program CAVER je již svou implementací na výpočet proteinových tunelů v rámci více snímků připraven. Modulu je tedy nutné předat pouze odkazy na příslušné soubory, s kterými má pracovat. Toho je docíleno pomocí sdíleného objektu výsledků, který slouží pro přenos informací mezi moduly.

Program Yasara pracuje na základě tří druhů vstupů: (i) prostřednictvím grafického uživatelského rozhraní; (ii) pomocí příkazové řádky; (iii) spuštěním předpřipravených skriptů. S ohledem na běhové prostředí výpočetního clusteru byl pro účely Caver Webu vybrán režim spouštění pomocí Yasara skriptů (skript se nachází na příloženém CD v souboru `scripts/cluster/yasara.sh`). Pro spouštění různých úloh byly autory připraveny skripty, které parametrizují výpočty pomocí specifického skriptovacího jazyka. Jedním z takto připravených skriptů je i skript pro výpočet molekulárních dynamik – `md_run.mcr` [2], který je přiložen na CD v souboru `scripts/cluster/md_run-parameters.mcr`. Tento skript byl převzat a byly mu umožněny pozměňovat pouze vybrané parametry. Modul na základě vstupního XML souboru, který popisuje úlohu a tedy i parametry pro spuštění nástroje Yasara, sestaví skript pro běh molekulárních dynamik. Sestavený skript je odeslán ke kalkulaci na výpočetní cluster.

Jelikož program Yasara neumožňuje přímo produkovat jednotlivé snímky ve formátu PDB, jsou vypočítané snímky převedeny pomocí dalšího skriptu – `md_convert.mcr` [1] do formátu PDB (příložen na CD v souboru `scripts/cluster/md_convert-parameters.mcr`). Simulace jsou během výpočtu ukládány ve speciálním datovém formátu nástroje Yasara (`sim`), který ukládá oproti ostatním formátům i rychlost a směr částic, takže umožňuje navázat na již ukončený výpočet simulace. Tento datový formát nabízí možnost na budoucí šetření výpočtů. Snímky ve formátu `sim` budou uloženy do serverové cache a budou dále rozšiřovány jen v případě požadavku na výpočet delší simulace.

Po ukončení kalkulace a konverze molekulárních dynamik jsou snímky poskytnuty modulu CAVER, který v nich identifikuje tunely a vytvoří výstupní soubor. Soubor je opět ve formátu XML a obsahuje statistická data všech tunelů a clusterů.

Experiment délky běhu simulace

Pro možnost poskytnutí odhadovaného času výpočtu byly během integrace experimentálně naměřeny délky výpočtů pro různorodé proteiny. Výsledky jsou zobrazeny v tabulce 7.1. Požadavkem byl výpočet 5 nanosekund trajektorie vždy při shodných podmínkách. Každá

úloha požadovala přidělit 16 jader a 16 GB operační paměti. Při odeslání úloh do fronty bylo pozorováno, že tyto požadavky jsou ve frontě často předbíhány méně náročnými úlohami, protože požadují méně zdrojů. Při běhu úloh bylo zjištěno, že nástroj Yasara používá k výpočtu přibližně 8 jader a využívá do 5 GB operační paměti, takže požadavky na výpočetní zdroje mohou být sníženy.

Během běhu simulace bylo vyprodukováno 50 snímků (interval pro ukládání snímků byl 0.1ns), které vyprodukovaly od 26 MB (pro 4OEE) až 436 MB (pro 2AKA). Jak můžeme vidět v tabulce 7.1, délka výpočtu částečně koreluje s velikostí struktury. V některých případech však dochází k odchylkám, kdy u výpočtu struktur 4E46 a 4LRJ je čas značně odlišný i přes podobný počet atomů.

Z výsledků měření můžeme konstatovat, že velikost dat nebude z pohledu zpracování serverem komplikovaná. Důležitým zdrojem je však čas. Čas výpočtu se mění a není podle počtu atomů možné predikovat čas potřebný pro jeho doběhnutí, ale je potřeba brát v povědomí i oligomerní stav proteinu. Podle naměřených dat lze pozorovat, že heterodimery potřebují více času k simulaci.

Pozorování během experimentu vedlo k závěru, že za provozu budou sbírány statistiky délky výpočtů, ze kterých se aproximací bude predikovat předpokládaný čas výpočtu.

PDB kód	Oligomerní stav	Počet aminokyselin	Počet atomů	Délka výpočtu [h]
4OEE	monomer	155	1 068	1:58
2OMB	homodimer	434	6 387	16:09
4E46	monomer	296	2 491	5:51
4LRJ	heterodimer	296	2 258	7:33
5K3A	homodimer	590	4 707	3:50
1CFJ	monomer	535	4 244	9:17
5HQ4	monomer	669	5 344	10:42
5KSK	homodimer	1 496	11 095	38:42
2AKA	heterodimer	1 075	8 625	60:22

Tabulka 7.1: Délka výpočtu simulace molekulárních dynamik v délce 5 ns na výpočetním clusteru (převzato a upraveno z [57]).

7.3 Webová aplikace

Uživatelské rozhraní

Uživatelské rozhraní je nově vytvořeno v JavaScriptu (resp. jazyce TypeScript) s využitím knihovny React. Dále se využívá React Router pro zobrazování obsahu na základě specifikované URL adresy. Implementace React aplikace se nachází na přiloženém CD v archivu `frontend.zip`.

Framework React funguje na principu reakce na změnu stavu aplikace. Interně si udržuje reprezentaci DOM ¹ a při jeho změně vypočítá a překreslí jen modifikované komponenty. Komponenty představují jednotlivé prvky stránky, jež disponují rozhraním, skrze které jsou napojeny na globální stav aplikace. Ke správě globálního stavu se využívá knihovna Redux. Stav aplikace tvoří jeden objekt a jeho modifikace se provádí pomocí akcí, které jsou vždy

¹Document Object Model – objektově orientovaná reprezentace HTML [60]

cílené jen na určitou část (atributy) stavu. Při změně stavu je vyvolána akce, která určí, jaké komponenty je nutné překreslit s novými daty.

Aplikace tedy disponuje svým stavem, který zastřešuje data aplikace, a komponentami, které tato data zobrazují. K propojení těchto dvou funkčních jednotek se využívá tzv. kontejnerů. V kontejneru si vývojář určí, k jakým částem stavu bude daná komponenta napojena a které akce vůči globálnímu stavu může provádět. Specifikované rozhraní se nazývá *properties*.

Rozložení komponent v jednotlivých stránkách koresponduje s návrhem, který byl popsán v rámci kapitoly 6. K implementaci jednotlivých komponent bylo využito knihovny `React-bootstrap` (<https://react-bootstrap.github.io/>), která disponuje již předpřipravenými elementy. Využívá se především komponenta pro zobrazení tabulky, vyskakovacích poznámek a dialogového okna modal. Společně s integrací knihovny `React-bootstrap` byly začleněny i předdefinované kaskádové styly, které slouží jako základní grafický nástroj. Nadstavbu základních stylů knihovny `bootstrap` tvoří kaskádové styly Loschmidtových laboratoří, které již byly definovány při tvorbě předchozích aplikací a sjednocují vzhled všech nástrojů této organizace.

Jednotný sdílený vzhled a rozložení mají i základní komponenty a stránky Loschmidtových laboratoří. Základní rozložení a vzhled komponent je zobrazen na obrázku 7.3. Stránka je tvořena záhlavím, patičkou a bočním panelem, ve kterém se nachází loga a informace o autorech. Poslední částí je hlavní panel, který je určen k individuální implementaci vývojářem nástroje. Pro sjednocení i této části se využívá sdílených kontejnerových komponent (boxů), které zabalují informace do zaoblených obdélníků s úvodním titulkem.

Stránka sbírající vstupní data od uživatele je tvořena hlavně formuláři a tlačítky, které se zobrazují na základě předchozích voleb. K efektu reaktivního zobrazování jednotlivých komponent se využívá již dříve zmíněného stavu. Akce (například kliknutí na tlačítko) pozmění příznak v globálním stavu aplikace a způsobí překreslení stránky. Stav tak slouží k dvěma účelům – definici vzhledu stránky a držení kontextu probíhající úlohy.

Více interaktivní je stránka výběru počátečního bodu pro výpočet tunelu. V této stránce uživatelovy volby ovlivňují vizualizační komponentu, která vykresluje proteinovou strukturu a nalezené katalytické kapsy. K vizualizaci biologických dat se využívá nové knihovny `Mol*` [48], která byla vytvořena brněnskou vědeckou skupinou, jež se zabývá vizualizací biologických dat.

Následující stránka pro nastavení nástroje CAVER je tvořena opět prostřednictvím formuláře. Po kliknutí na tlačítko pro odeslání úlohy jsou potřebná data zpracována backendem do formy XML souboru, který je následně odeslán do výpočetního jádra.

Po odeslání úlohy je uživatel přesunut na stránku prohlížeče výsledků. Zde je zobrazena informace o stavu probíhajícího výpočtu. Informace o stavu výpočtu je sdílena prostřednictvím záznamu v databázi, ke které má přístup jádro i backend, a slouží tak k jejich komunikaci. Tlačítko pro zobrazení výsledků je neaktivní až do chvíle dobehnutí výpočtů. Poté je umožněno pomocí prokliku přejít do stránky výsledků.

Stránka výsledků je tvořena dvěma záložkami, které slouží pro zobrazení statistik celé simulace a analýzy jednotlivých snímků. V případě analýzy statické struktury je záložka statistik deaktivována. V opačném případě je uživateli zobrazena a obsahuje tepelnou mapu všech tunelů, statistickou tabulku tunelových clusterů a tepelnou mapu jednoho tunelu. K vykreslení grafů se používá knihovna `recharts`. Záložka analýzy jednoho snímku nebo statické struktury obsahuje komponenty a grafy pro zobrazení a analýzy identifikovaných tunelů. Hlavními komponentami jsou tabulka tunelů, vizualizační komponenta a úlohy nástroje CaverDock.

The screenshot displays the CAVER web interface. At the top left is the CAVER logo and version (v2.0), and at the top right is the logo of the Loschmidt Laboratories. Below the header is a navigation bar with links: Submit new job, Help, Example, Acknowledgements, Full page layout (toggle), Job ID (input field), and Find job. The main content area is titled "Input selection" and contains two sections: "What do you wish to analyze?" with buttons for "Tunnels" and "Channels", and "Input" with buttons for "Single structure" and "Multiple structures". Below these is a "SELECT PDB FILE" section with radio buttons for "Enter PDB code" (selected) and "Upload PDB file", a text input for "PDB ID" (containing "ex: 1cv2"), and a "Download file" button. The right sidebar contains several sections: "REFERENCES" with a list of citations and PubMed/Open Access logos; "USER STATISTICS" showing visitor and job counts; "CONTACT" with Loschmidt Laboratories details; "OTHER TOOLS" listing various bioinformatics tools; and "ACKNOWLEDGEMENT" with logos for elixir and C4Sys. At the bottom, there are keywords, a disclaimer, and links for terms of use and privacy.

Obrázek 7.3: Základní rozložení prvků webových aplikací Loschmidtových laboratoří. Základními prvky uživatelského rozhraní jsou záhlaví, patička a boční panel. V hlavním panelu se využívají kontejnerové komponenty za účelem sdružení sémanticky podobných prvků do jednoho celku (modrý box).

Aplikační backend

Aplikační backend je implementován v jazyce Java a využívá framework Spring Boot (<https://spring.io/projects/spring-boot>). Jeho implementaci lze nalézt na přiloženém CD v archivu `backend.zip`. Tento moderní framework byl vybrán z důvodů jeho vlastností, které umožňují rychlý vývoj aplikací s lehkou konfigurací. Hlavní předností je poskytnutí robustního zpracování HTTP požadavků a správa koncových bodů REST API. Dalšími užitečnými vlastnostmi jsou: automatická konfigurace, poskytnutí anotací, snadné řízení závislostí a vestavěný Servlet kontejner [53].

Při spuštění Spring Boot aplikace je vyhledán vstupní bod aplikace a je spuštěna automatická konfigurace. Poté se provede procedura skenování komponent, které jsou v projektu zahrnuty. Skenování nalezne všechny deklarace balíčků a komponent (beanů), které inicializuje podle konfigurace [53].

Při tvorbě datových tříd se využívá programovacího modelu založeného na technologii Spring, který usnadňuje užívání technologií pro přístup k databázím, map-reduce frameworkům a cloudovým datovým službám [10].

V projektu se využívá práce s repositáři, které nabízí tvorbu dynamických databázových dotazů z názvů metod repositáře. Byla použita knihovna Lombok, která prostřednictvím anotací generuje metody typu getter a setter. Další využívanou částí frameworku je Spring Data REST, pomocí kterého se vytváří koncové body aplikace [10].

Tato část systému je zodpovědná za předzpracování dat obdržených z výpočetního jádra a zpřístupnění dat ze souborového systému či databáze. Aplikační frontend se dotazuje na jednotlivé koncové body API pomocí HTTP požadavků. Koncové body byly navrženy přesně pro potřeby aplikace. Na straně této komponenty dochází i k identifikaci struktur, které pomáhají s výběrem počátečního bodu pro výpočet proteinových tunelů. Koncové body představují kontroléry, které reagují na jednotlivé požadavky a volají běžící služby k obsluze požadovaných akcí, jako jsou například práce se souborovým systémem a komunikace s databází.

Hlavní úlohou backendu je zpracovávat požadavky od uživatele a spravovat (případně vytvářet) potřebné soubory, jako jsou identifikované biologické jednotky, konfigurační soubory pro jádro a vygenerované výstupní soubory programu CAVER. Tyto úlohy se provádí pro každou úlohu zvlášť, takže je nutné jednotlivé soubory rozlišovat. V rámci aplikace byla navržena služba (`StorageService`), která zaštiťuje práci se souborovým systémem. Při zahájení nové úlohy je vygenerován unikátní identifikátor, který slouží jako přístupový klíč do datového prostoru úlohy. Díky tomu je například soubor, který slouží pro identifikaci kapes, spuštění molekulární dynamiky a nalezení tunelů, vždy uložen v souboru pod jménem `structure.pdb`.

Každá úloha je v databázi identifikována jedním záznamem, který obsahuje informace o uživateli, stavu úlohy a adrese konfiguračního souboru určeného pro jádro. Inicializační skript databáze se nachází na příloženém disku v souboru `scripts/database/LL_caverweb_structure.sql`. Po vytvoření konfiguračního souboru je v databázi změněn status úlohy, který ji označí za připravenou k vykonání. Jádro, které databázi pravidelně prohlíží, stáhne konfigurační soubor a spustí specifikované výpočty. Dále se k úloze váží metadata, která jsou uložena v samostatné tabulce. Metadata slouží k doplnění úlohy o další potřebné informace, jako jsou konfigurace analýz transportu ligandu, protože během jedné úlohy můžeme spustit více těchto analýz s rozličnými ligandy a v různých tunelech.

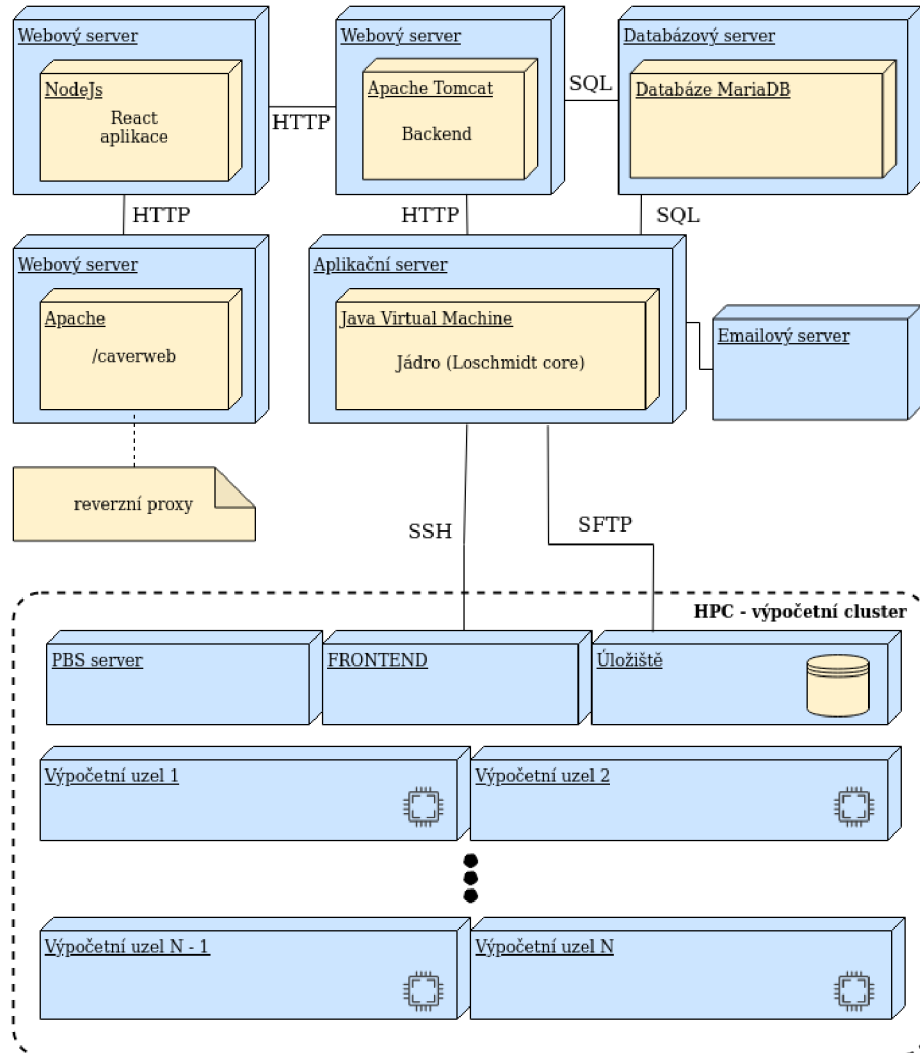
Výsledky jsou nejprve staženy z výpočetního clusteru a následně umístěny na specifická místa v datovém prostoru úlohy. Jelikož jsou data uložena v souborech ve formátu `xml`, je nutné je zpracovat a přeformátovat do datové reprezentace, v které budou poskytnuty webovému rozhraní. Za tímto účelem v aplikaci figurují loadery, které jsou zodpovědné za převedení z obecného datového formátu na interní datovou reprezentaci aplikace. Dotazováním se konkrétní koncové body API se data filtrují a ještě více transformují. Problémem a největší výzvou, která je v současné době neustále řešena, je dosažení minimální zátěže klienta, protože analýzy molekulárních dynamik produkují velké množství dat.

7.4 Nasazení

Celá aplikace je cíleně navržena pro snadné nasazení bez ohledu na geografickou či síťovou blízkost jednotlivých komponent. Proto se k veškeré komunikaci se využívá výhradně standardních síťových protokolů jako je HTTP, SSH či MySQL. Pro snažší nasazení a pře-

nositelnost jsou všechny komponenty (s výjimkou výpočetního clusteru) připraveny a distribuovány jako obrazy pro systém Docker (<https://www.docker.com/>).

V rámci plánovaného nasazení v produkčním prostředí Loschmidtových laboratoří (obrázek 7.4) pak bude aplikaci předřazeno navíc ještě reverzní proxy zajišťující kompletní flexibilitu ve skutečném umístění běhového prostředí.



Obrázek 7.4: Znázornění nasazení nástroje Caver Web 2.0 [57].

Kapitola 8

Závěr

Studium proteinových struktur je v posledních letech čím dál více populární. Důvodem nárůstu zájmu je možnost jejich průmyslového využití. Hlavní skupinou zájmu jsou enzymy, které se od svých protějšků liší schopností katalyzovat chemické reakce, čímž se stávají zajímavou alternativou k tradičním chemickým katalyzátorům. Nevýhodou ale zůstává jejich citlivost na podmínky v okolním prostředí a jejich selektivita vůči substrátům, kdy katalyzují někdy jen jednu konkrétní molekulu. Studium enzymů umožňuje jejich vlastnosti zlepšovat a ovládat jejich funkčnost v požadovaném směru [36].

Enzymy často formují svá aktivní místa hluboko ve svém vnitru, přičemž jsou z vnějšího prostředí přístupné prostřednictvím proteinových tunelů. Proteiny a proteinové tunely byly popsány v kapitole 2. Tato přístupová cesta (a především její hrdlo) se stává hlavním bodem zájmu výzkumu. Zkoumání těchto úzkých míst, skrývajících se uvnitř proteinů, je v laboratorních podmínkách velmi zdlouhavé a nákladné, a proto vznikly specializované programy, které umožňují analyzovat již nasnímané proteinové struktury. Příkladem nástrojů, které se zaměřují na průzkum proteinových tunelů, jsou CAVER a CaverDock. Nástroje byly popsány v kapitole 2.

S rozvojem počítačových systémů narostly i možnosti studií, které mohou zachycovat i dynamické děje v proteinových komplexech. Problematika simulací molekulárních dynamik včetně existujících alternativních přístupů byla uvedena v kapitole 3.

V současné době existuje velké množství programů, které nabízí různorodé druhy analýz. Jejich používání je doprovázeno nutností jejich prvotní studie, jež stojí drahocenný čas vědeckých pracovníků. Z tohoto důvodu vznikají nové komplexnější aplikace, které sdružují jednotlivé nástroje do pracovních protokolů v uživatelsky přívětivé formě. Jednou takovou aplikací je i aplikace Caver Web, jejíž nová verze byla navržena v kapitole 6. Nová verze aplikace přináší v dnešní době unikátní možnost výpočtů molekulárních dynamik v rámci webové aplikace a stane se tak první, která tuto možnost poskytne. Důležitou součástí práce bylo zachování jednoduchosti uživatelského rozhraní v kombinaci se značným nárůstem prezentovaných informací. Vývoj uživatelského rozhraní byl pravidelně konzultován s doménovými experty z Loschmidtových laboratoří.

Vývoj aplikace znamenal i rozšíření celého systému, který nyní zaštiťuje kalkulaci molekulárních dynamik na výpočetním clusteru. Návrh rozšíření a implementace nových modulů jádra systému byly popsány v kapitolách 6 a 7. Uživatelské rozhraní bylo implementováno (včetně nově navržených prvků) za využití technologie React.

Výstupem vývoje je první prototyp aplikace, který umožňuje spustit plně automatickou simulaci molekulární dynamiky a provést základní zobrazení a analýzu nalezených tunelů ve webovém rozhraní. Aplikace je v aktuální podobě bohužel nestabilní, neboť se nestihlo

plně dokončit zefektivnění zpracovávání a vizualizace velkého množství dat, které generuje simulace molekulární dynamiky. Práce na vývoji i nadále pokračuje a v blízké budoucnosti začne i implementace dalších rozšíření navržených v rámci této práce.

Práce byla na konci roku 2021 úspěšně prezentována formou posteru na konferenci pod záštitou iniciativy ELIXIR CZ [4], která propojuje vědní skupiny zabývající se strukturální biologii. V květnu roku 2022 byla práce ústně prezentována na konferenci Excel@FIT [5], kde byla oceněna odbornou komisí za multioborový přesah a kvalitní textové zpracování. Dále získala i cenu Jiřího Kunovského, která byla udělena na základě hlasování veřejnosti.

Literatura

- [1] *Pre-made script in YASARA written by Elmar Krieger for molecular dynamics simulation* [http://www.yasara.org/md_convert.mcr]. Accessed: 2022-03-21.
- [2] *Pre-made script in YASARA written by Elmar Krieger for molecular dynamics simulation – Convert between Sim, XTC, MDCrd and PDB simulation trajectories* [http://www.yasara.org/md_run.mcr]. Accessed: 2022-03-21.
- [3] *Caver Web: Identification of Tunnels and Channels in Proteins and Analysis of Ligand transport*. Jan Stourac, Ondrej Vavra, Piia Kokkonen, Jiri Filipovic, Gaspar Pinto, Jan Brezovsky, Jiri Damborsky, David Bednar, 2019. Conference name.
- [4] *Identification of the protein tunnels using molecular dynamics*. ELIXIR CZ Annual Conference 2021, 2021.
- [5] *Identifikace proteinových tunelů s využitím molekulárních dynamik*. Vysoké Učení Technické v Brně, Fakulta informačních technologií, 2022, sv. 25.
- [6] ALBERTS, B. *Základy buněčné biologie : Úvod do molekulární biologie buňky*. Ústí nad Labem: Espero Publishing, 1998. ISBN 80-902906-0-4.
- [7] BANK, P. data. *Methods for Determining Atomic Structures*. [Navštíveno online: <https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/methods-for-determining-structure>].
- [8] BENEŠ, M. *Molekulová dynamika*. [Navštíveno online: <https://slideplayer.cz/slide/2514717/>].
- [9] BERMAN, H., HENRICK, K. a NAKAMURA, H. Announcing the worldwide protein data bank. *Nature Structural & Molecular Biology*. Nature Publishing Group. 2003, sv. 10, č. 12, s. 980–980.
- [10] BOOT, S. *Spring Boot*. [Navštíveno online: <https://spring.io/projects/spring-boot>].
- [11] BREZOVSKY, J., CHOVANCOVA, E., GORA, A., PAVELKA, A., BIEDERMANNNOVA, L. et al. Software tools for identification, visualization and analysis of protein tunnels and channels. *Biotechnology advances*. Elsevier. 2013, sv. 31, č. 1, s. 38–49.
- [12] BYŠKA, J., JURČÍK, A., GRÖLLER, M. E., VIOLA, I. a KOZLIKOVA, B. MoleCollar and tunnel heat map visualizations for conveying spatio-temporo-chemical properties across and along protein voids. In: Wiley Online Library. *Computer Graphics Forum*. 2015, sv. 34, č. 3, s. 1–10.

- [13] CASE, D. A., CHEATHAM III, T. E., DARDEN, T., GOHLKE, H., LUO, R. et al. The Amber biomolecular simulation programs. *Journal of computational chemistry*. Wiley Online Library. 2005, sv. 26, č. 16, s. 1668–1688.
- [14] CHOVANCOVA, E., PAVELKA, A., BENES, P., STRNAD, O., BREZOVSKY, J. et al. CAVER 3.0: a tool for the analysis of transport pathways in dynamic protein structures. *PLoS Comput Biol*. Public Library of Science. 2012, sv. 8, č. 10, s. e1002708.
- [15] CONSORTIUM, T. U. UniProt: the universal protein knowledgebase. *Nucleic acids research*. 2018, sv. 46, D1, s. 2699–2699.
- [16] CORTÉS, J., LE, D. T., IEHL, R. a SIMÉON, T. Simulating ligand-induced conformational changes in proteins using a mechanical disassembly method. *Physical Chemistry Chemical Physics*. Royal Society of Chemistry. 2010, sv. 12, č. 29, s. 8268–8276.
- [17] CUENDET, M. A. a GUNSTEREN, W. F. van. On the calculation of velocity-dependent properties in molecular dynamics simulations using the leapfrog integration algorithm. *The Journal of chemical physics*. American Institute of Physics. 2007, sv. 127, č. 18, s. 184102.
- [18] CUSABIO. *Biological Activity of Proteins*. [Převzato z: <https://www.cusabio.com/c-20943.html>].
- [19] DE GROOT, B., VAN AALTEN, D., SCHEEK, R., AMADEI, A., VRIEND, G. et al. Prediction of protein conformational freedom from distance constraints. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 1997, sv. 29, č. 2, s. 240–251.
- [20] DELANO, W. L. et al. Pymol: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr*. Citeseer. 2002, sv. 40, č. 1, s. 82–92.
- [21] FILIPOVIČ, J., VÁVRA, O., PLHÁK, J., BEDNÁŘ, D., MARQUES, S. M. et al. CaverDock: a novel method for the fast analysis of ligand transport. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. IEEE. 2019, sv. 17, č. 5, s. 1625–1638.
- [22] GRUBMÜLLER, H. Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Physical Review E*. APS. 1995, sv. 52, č. 3, s. 2893.
- [23] GUPTA, S. a VARADWAJ, P. K. A BRIEF OVERVIEW ON MOLECULAR DYNAMICS SIMULATION OF BIOMOLECULAR SYSTEM: PROCEDURE, ALGORITHMS AND APPLICATIONS. *INTERNATIONAL JOURNAL OF PHARMACEUTICAL SCIENCES AND RESEARCH*. INT JOURNAL PHARMACEUTICAL SCIENCES & RESEARCH 1117, SECTOR 12 A, PANCHKULA 2018, sv. 9, č. 4, s. 1333–1350.
- [24] GUTMANAS, A., ALHROUB, Y., BATTLE, G. M., BERRISFORD, J. M., BOCHET, E. et al. PDBe: protein data bank in Europe. *Nucleic acids research*. Oxford University Press. 2014, sv. 42, D1, s. D285–D291.

- [25] HAILE, J. *Molecular Dynamics Simulation : Elementary Methods*. New York: John Wiley and Sons, 1992. ISBN 0-471-81966-2.
- [26] HÅKAN, H. W. a ÅGREN, H. Quantum Mechanics/Classical Mechanics Modeling of Biological Systems. In: *Multiscale Modeling and Simulation in Science*. Springer, 2009, s. 291–294.
- [27] HENTSCHKE, R. Integrating the Equations of Motion. In: *Classical Mechanics*. Springer, 2017, s. 123–153.
- [28] JOHNSON, A., JOHNSON, T. a KHAN, A. Thermostats in Molecular Dynamics Simulations. *University of Massachusetts Amherst*. 2012, sv. 1, s. 29.
- [29] KIM, J.-K., CHO, Y., LEE, M., LASKOWSKI, R. A., RYU, S. E. et al. BetaCavityWeb: a webserver for molecular voids and channels. *Nucleic acids research*. Oxford University Press. 2015, sv. 43, W1, s. W413–W418.
- [30] KRIEGER, E. a VRIEND, G. New ways to boost molecular dynamics simulations. *Journal of computational chemistry*. Wiley Online Library. 2015, sv. 36, č. 13, s. 996–1007.
- [31] LAND, H. a HUMBLE, M. S. YASARA: a tool to obtain structural guidance in biocatalytic investigations. In: *Protein Engineering*. Springer, 2018, s. 43–67.
- [32] LANGE, O. F., SCHÄFER, L. V. a GRUBMÜLLER, H. Flooding in GROMACS: accelerated barrier crossings in molecular dynamics. *Journal of computational chemistry*. Wiley Online Library. 2006, sv. 27, č. 14, s. 1693–1702.
- [33] LE GUILLOUX, V., SCHMIDTKE, P. a TUFFERY, P. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*. BioMed Central. 2009, sv. 10, č. 1, s. 1–11.
- [34] LECKBAND, D. a ISRAELACHVILI, J. Intermolecular forces in biology. *Quarterly reviews of biophysics*. Cambridge University Press. 2001, sv. 34, č. 2, s. 105.
- [35] MAREK, J., VÉVODOVÁ, J., SMATANOVÁ, I. K., NAGATA, Y., SVENSSON, L. A. et al. Crystal structure of the haloalkane dehalogenase from *Sphingomonas paucimobilis* UT26. *Biochemistry*. ACS Publications. 2000, sv. 39, č. 46, s. 14082–14086.
- [36] MARQUES, S. M., DUNAJOVA, Z., PROKOP, Z., CHALOUPOKVA, R., BREZOVSKY, J. et al. Catalytic cycle of haloalkane dehalogenases toward unnatural substrates explored by computational modeling. *Journal of chemical information and modeling*. ACS Publications. 2017, sv. 57, č. 8, s. 1970–1989.
- [37] MASOOD, T. B., SANDHYA, S., CHANDRA, N. a NATARAJAN, V. CHEXVIS: a tool for molecular channel extraction and visualization. *BMC bioinformatics*. BioMed Central. 2015, sv. 16, č. 1, s. 1–19.
- [38] MORRIS, G. M., HUEY, R., LINDSTROM, W., SANNER, M. F., BELEW, R. K. et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*. Wiley Online Library. 2009, sv. 30, č. 16, s. 2785–2791.

- [39] MORRIS, G. M., HUEY, R., LINDSTROM, W., SANNER, M. F., BELEW, R. K. et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*. Wiley Online Library. 2009, sv. 30, č. 16, s. 2785–2791.
- [40] MURTAGH, F. a CONTRERAS, P. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. Wiley Online Library. 2012, sv. 2, č. 1, s. 86–97.
- [41] NGUYEN, T. D., PHILLIPS, C. L., ANDERSON, J. A. a GLOTZER, S. C. Rigid body constraints realized in massively-parallel molecular dynamics on graphics processing units. *Computer Physics Communications*. Elsevier. 2011, sv. 182, č. 11, s. 2307–2313.
- [42] OKAMOTO, Y. Generalized-ensemble algorithms: enhanced sampling techniques for Monte Carlo and molecular dynamics simulations. *Journal of Molecular Graphics and Modelling*. Elsevier. 2004, sv. 22, č. 5, s. 425–439.
- [43] OLOMOUCI, U. P. v. *Molekulová dynamika*. [Navštíveno online: <https://fch.upol.cz/skripta/momo/MD.pdf>].
- [44] PELLEGRINI CALACE, M., MAIWALD, T. a THORNTON, J. M. PoreWalker: a novel tool for the identification and characterization of channels in transmembrane proteins from their three-dimensional structure. *PLoS computational biology*. Public Library of Science San Francisco, USA. 2009, sv. 5, č. 7, s. e1000440.
- [45] PRAVDA, L., SEHNAL, D., TOUŠEK, D., NAVRÁTILOVÁ, V., BAZGIER, V. et al. MOLEonline: a web-based tool for analyzing channels, tunnels and pores (2018 update). *Nucleic acids research*. Oxford University Press. 2018, sv. 46, W1, s. W368–W373.
- [46] RIBEIRO, A. J. M., HOLLIDAY, G. L., FURNHAM, N., TYZACK, J. D., FERRIS, K. et al. Mechanism and Catalytic Site Atlas (M-CSA): a database of enzyme reaction mechanisms and active sites. *Nucleic acids research*. Oxford University Press. 2018, sv. 46, D1, s. D618–D623.
- [47] SEELIGER, D., HAAS, J. a GROOT, B. L. de. Geometry-based sampling of conformational transitions in proteins. *Structure*. Elsevier. 2007, sv. 15, č. 11, s. 1482–1492.
- [48] SEHNAL, D., ROSE, A. S., KOCA, J., BURLEY, S. K. a VELANKAR, S. Mol*: Towards a Common Library and Tools for Web Molecular Graphics. In: *MolVa@ EuroVis*. 2018, s. 29–33.
- [49] STOURAC, J., VAVRA, O., KOKKONEN, P., FILIPOVIC, J., PINTO, G. et al. Caver Web 1.0: identification of tunnels and channels in proteins and analysis of ligand transport. *Nucleic acids research*. Oxford University Press. 2019, sv. 47, W1, s. W414–W422.
- [50] SUGITA, Y. a OKAMOTO, Y. Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*. Elsevier. 1999, sv. 314, 1-2, s. 141–151.

- [51] SVENDSEN, A. *Understanding Enzymes: Function, Design, Engineering, and Analysis*. CRC Press, 2016. Chapter 12: Marques S. M., Brezovsky J., Damborsky J.. *Role of tunnels, channels and gates in enzymatic catalysis*.
- [52] TROTT, O. a OLSON, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*. Wiley Online Library. 2010, sv. 31, č. 2, s. 455–461.
- [53] TUTORIALSPPOINT. *Spring Boot – Introduction*. [Navštíveno online: https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm].
- [54] VAN DER VAART, A. a KARPLUS, M. Simulation of conformational transitions by the restricted perturbation-targeted molecular dynamics method. *The Journal of chemical physics*. American Institute of Physics. 2005, sv. 122, č. 11, s. 114903.
- [55] VAVRA, O., FILIPOVIC, J., PLHAK, J., BEDNAR, D., MARQUES, S. M. et al. CaverDock: a molecular docking-based tool to analyse ligand transport through protein tunnels and channels. *Bioinformatics*. Oxford University Press. 2019, sv. 35, č. 23, s. 4986–4993.
- [56] VERLET, L. Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical review*. APS. 1967, sv. 159, č. 1, s. 98.
- [57] ŠTOURAC, J. *Softwarový nástroj pro automatický návrh mutací a chytrých knihoven pro proteinové inženýrství [online]*. 2017 [cit. 2020-12-29]. Diplomová práce. Masarykova univerzita, Fakulta informatiky, Brno. Vedoucí práce BREZOVSKÝ, J. Dostupné z: <https://is.muni.cz/th/hskmf/>.
- [58] ŠTOURAC, J. *Systém na integraci bioinformatických nástrojů pro analýzu tunelů v proteinech [online]*. 2017 [cit. 2022-04-11]. Bakalářská práce. Masarykova univerzita, Fakulta informatiky, Brno. Vedoucí práce BREZOVSKÝ, J.
- [59] WERESZCZYNSKI, J. a MCCAMMON, J. A. Statistical mechanics and molecular dynamics in evaluating thermodynamic properties of biomolecular recognition. *Quarterly reviews of biophysics*. NIH Public Access. 2012, sv. 45, č. 1, s. 1.
- [60] WIKIPEDIA. *Document Object Model — Wikipedia, The Free Encyclopedia* [https://en.wikipedia.org/wiki/Document_Object_Model]. 2020. [Online; navštíveno 5. 5. 2022].
- [61] WIKIPEDIA. *Protein structure — Wikipedia, The Free Encyclopedia* [<http://en.wikipedia.org/w/index.php?title=Protein%20structure&oldid=995899200>]. 2020. [Online; navštíveno 29. 12. 2020].
- [62] YAFFE, E., FISHELOVITCH, D., WOLFSON, H. J., HALPERIN, D. a NUSSINOV, R. MolAxis: a server for identification of channels in macromolecules. *Nucleic acids research*. Oxford University Press. 2008, sv. 36, suppl_2, s. W210–W215.