

Rozprawa doktorska p.t.:

**ModeRNA:
program komputerowy do przewidywania struktury
trzeciorzędowej RNA z zastosowaniem podejścia
modelowania homologicznego**

Magdalena Rother

Pracownia Bioinformatyki
Instytut Biologii Molekularnej i Biotechnologii
Uniwersytet im. Adama Mickiewicza
Poznań

Promotor: prof. dr hab. Janusz M. Bujnicki

Pracownia Bioinformatyki
Instytut Biologii Molekularnej i Biotechnologii
Uniwersytet im. Adama Mickiewicza
Poznań

Pracownia Bioinformatyki i Inżynierii Białka
Międzynarodowy Instytut Biologii Molekularnej i Komórkowej
Warszawa

Poznań 2012

Podziękowania

Dziękuję mojemu promotorowi **prof. dr hab. Januszowi M. Bujnickiemu** za niezwykle ciekawy okres studiów doktoranckich pod jego opieką.

Dziękuję Dziekanowi Wydziału Biologii **prof. dr hab. Bogdanowi Jackowiakowi** oraz Dyrektorce Instytutu Biologii Molekularnej i Biotechnologii **prof. dr hab. Zofii Szweykowskiej-Kulińskiej** za pomoc i wsparcie. Dziękuję **Radzie Wydziału Biologii UAM** za przeprowadzenie mojego przewodu doktorskiego oraz **dr Karolinie Stefaniak** za cenne rady i wskazówki.

Dziękuję **prof. dr hab. Arturowi Jarmołowskiemu** i **prof. dr hab. Ryszardowi W. Adamiakowi** za czas poświęcony na recenzowanie niniejszej pracy.

Dziękuję **prof. dr hab. Jordi Villa i Freixa** i **dr Raulowi E. Alcantara** oraz **mgr Ignasio Buchowi** za okazaną gościnność.

Dziękuję współautorom projektu ModeRNA. **Mgr Tomaszowi Putonowi** za testowanie programu oraz przygotowanie funkcji `find_clashes`. **Dr Kristianowi Rother** za przygotowanie części testowej programu i funkcji `fix_backbone`. **Mgr Kai Milanowskiej** za przygotowanie interfejsu Django serwera ModeRNA. **Lic. Pawłowi Skibie** za pracę nad wstępną wersją modułu do modelowania izosterycznych par zasad. **Lic. Jarosławowi Jeleniewiczowi** dziękuję za przygotowanie bazy struktur szablonów. **Prof. dr hab. Januszowi M. Bujnickiemu** za wyznaczenie głównych cech programu ModeRNA.

Dziękuję mgr **Kai Milanowskiej** oraz **mgr Tomaszowi Putonowi** za opinie i uwagi merytoryczne oraz redakcyjne na temat niniejszej pracy.

Dziękuję adiunktom, doktorantom, magistrantom i licencjuszom Pracowni Bioinformatyki, z którymi miałam zaszczyt współpracować. W szczególności **dr Annie Czerwoniec**, **dr Kristianowi Rotherowi**, **mgr Joannie Kasprzak**, **mgr Kai Milanowskiej**, **mgr Annie Philips**, **mgr Tomaszowi Putonowi**, **lic. Pawłowi Skibie**, **mgr Katarzynie Mikołajczak**, **mgr Marcinowi Skorupskiemu**, **mgr Ewie Bielskiej**, **mgr Annie Łukasik**, **mgr Zuzannie Balcer**, **mgr Xavierowi Lucasowi** i **lic. Mateuszowi Korycińskiemu** za wspaniałą atmosferę w pracy oraz ciekawe dyskusje, zarówno naukowe jak i prywatne.

Dziękuję koleżankom i kolegom z Instytutu Biologii Molekularnej i Biotechnologii, Wydziału Biologii UAM oraz Pracowni Inżynierii Białka za wymiany poglądów na temat projektu ModeRNA, pracy naukowej i życia, które pomagały mi rozwijać się, jako człowiekowi i naukowcowi w okresie studiów doktoranckich. W szczególności dziękuję **dr Agnieszce Obarskiej-Kosińskiej, dr Elizie Głowskiej, dr Ewie Wywiął, dr Michałowi Bonieckiemu, dr Janowi Kosińskiemu, dr Wojciechowi Potrzebowskiemu, mgr Natalii Osten-Sacken, mgr Halinie Pietrykowskiej, mgr Juliuszowi Stasiewiczowi i mgr Irinie Tuszyńskiej.**

Dziękuję moim rodzicom, **Bożenie i Bogusławowi Musielak** za wychowanie, a w szczególności za to, że nauczyli mnie wytrwale dążyć do celu. Dziękuję również siostrze **Annie Musielak** za wiarę w moje możliwości i okazane wsparcie.

Dziękuję mojemu mężowi **Kristianowi Rotherowi** za to, że uczestniczył ze mną nie tylko w sukcesach, ale również wspierał mnie w trudnych chwilach. Dziękuję również naszemu synowi **Wojtkowi Rotherowi**, który cierpliwie dzielił mamę z jej pracą doktorską.

Spis treści

1. Wstęp.....	1
1.1. Budowa i funkcja RNA	3
1.2. RNA w nauce i medycynie	14
1.3. Sposoby ustalania struktury przestrzennej RNA	18
1.4. Opis problemu	26
2. Cel pracy	27
3. Materiały	28
3.1. Pliki ze współrzędnymi położenia atomów struktur RNA	28
3.2. Przyrównania sekwencji tRNA	29
4. Metody	31
4.1. Modelowanie homologiczne.....	31
4.2. Narzędzia programistyczne	33
4.3. Miary wykorzystane do ewaluacji modeli.....	38
4.4. Programy zewnętrzne	41
5. Wyniki.....	43
5.1. Modelu homologicznego z programem ModeRNA	43
5.2. Operacje zaimplementowane w programie ModeRNA.....	46
5.3. Użycie programu ModeRNA.....	72
5.4. Architektura programu ModeRNA.....	83
5.5. Modele zbudowane przy użyciu programu ModeRNA.....	90
5.6. Program MARS do ewaluacji jakości modeli RNA.....	96

6.	Dyskusja	99
6.1.	Porównanie programu ModeRNA z innymi metodami.....	99
6.2.	Ograniczenia i mocne strony programu ModeRNA	107
6.3.	Wybór szablonu i przygotowanie przyrównania	111
6.4.	Biblioteka fragmentów	114
6.5.	Program MARS	119
6.6.	Perspektywy rozwoju programu ModeRNA	121
6.7.	Wnioski.....	123
7.	Załącznik	125
7.1.	Kod programu ModeRNA (CD).....	125
7.2.	Wykaz identyfikatorów wykorzystanych plików PDB	125
7.3.	Przyrównanie 99 sekwencji tRNA	126
	Bibliografia.....	132
	Spis rycin.....	142
	Spis table	146

Finansowanie

Rozwój oprogramowania ModeRNA był finansowany przez:

- Wydział Biologii Uniwersytetu im. Adama Mickiewicza w Poznaniu (grant PBWB-03/2009 dla M.R.).
- Polskie Ministerstwo Nauki i Szkolnictwa Wyższego (grant HISZPANIA/152/2006 dla J.M.B.).
- Szósty Program Ramowy Komisji Europejskiej (Europejska Nagroda dla Młodego Naukowca w ramach sieci EURASNET – Europejska Sieć Splicingu Alternatywnego, LSHG-CT-2005-518238 dla J.M.B.).

Rozwój serwera ModeRNA był finansowany przez:

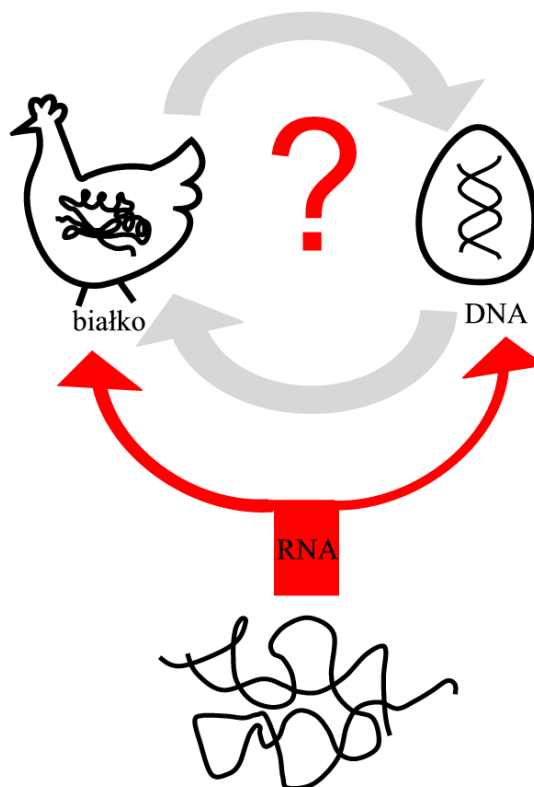
- Fundację na rzecz Nauki Polskiej (grant TEAM/2009-4/2 dla J.M.B.).

Publikacje wynikające z pracy

- **Rother, M.**, Rother, K., Puton, T. and Bujnicki, J.M. (2011) ModeRNA: a tool for comparative modeling of RNA 3D structure, *Nucleic Acids Res*, 39, 4007-4022.
- **Musielak, M.**, Rother, K., Puton, T. and Bujnicki, J.M. (2010) ModeRNA builds RNA 3D Models from Template Structures, *ERCIM News*, 82, 19-20.
- **Rother, M.**, Milanowska, K., Puton, T., Jeleniewicz, J., Rother, K. and Bujnicki, J.M. (2011) ModeRNA server: an online tool for modeling RNA 3D structures, *Bioinformatics*, 27 (17), 2441-2.
- **Rother, M.**, Rother, K., Puton, T. and Bujnicki, J.M. (2011) RNA tertiary structure prediction with ModeRNA, *Brief Bioinform*, 12, 601-613.

1. Wstęp

Kwas rybonukleinowy (RNA) występuje we wszystkich organizmach. RNA przez długi czas był postrzegany jako pośrednik pomiędzy przechowującym informację genetyczną kwasem deoksyrybonukleinowym (DNA) a aktywnymi biologicznie białkami. Pogląd ten zmienił się wraz z odkryciem katalitycznych właściwości RNA przez S. Altmana (Guerrier-Takada, i wsp., 1983) i T. Cecha (Kruger, i wsp., 1982), za co obaj naukowcy otrzymali nagrodę Nobla w dziedzinie chemii w 1989 roku. RNA jest obecnie uważany za jedyną cząsteczkę, która może zarówno przechowywać informację genetyczną oraz ma potencjał katalityczny (Westhof and Auffinger, 2000) (DNAzymy, cząsteczki DNA o właściwościach katalitycznych, pochodzą z selekcji *in vitro* i nie zostały zaobserwowane w naturze (Mastroiannopoulos, i wsp., 2010)). Twierdzenie to pozwoliło rozwiązać spór o początki życia na Ziemi: co było pierwsze – DNA czy białko, kura czy jajko (Rycina 1). Białka są doskonałymi katalizatorami, jednak nie posiadają pamięci biologicznej i potrzebują DNA. DNA przechowuje informację genetyczną, jednak nie jest w stanie katalizować reakcji i aby się powielić potrzebuje białek. RNA robi obie te rzeczy, choć wiele z jego funkcji zostało w toku ewolucji zastąpionych przez DNA i białka.



Rycina 1. Dylemat pochodzenia życia na Ziemi (co było pierwsze: DNA czy białko?) i jego rozwiązanie – teoria świata RNA.

RNA to nie tylko „skamienielina” pozostała po dawnym świecie RNA. Niższe organizmy, takie jak *Ceanorhabditis elegans* mają porównywalną do człowieka ilość genów kodujących białko, jednak genom ludzki jest 30 razy większy. Sugeruje to, że część genomu niekodująca białka ma decydujące znaczenie dla olbrzymiej złożoności wyższych eukariotów. P. P. Pandolfi i wsp. (Salmena, i wsp., 2011) zaproponowali niedawno hipotezę o współzawodniczących endogennych RNA („*competing endogenous RNA hypothesis*”, „*ceRNA hypothesis*”), według której kodujące i niekodujące RNA komunikują się wzajemnie poprzez miejsca wiązania mikroRNA. Cząsteczki RNA, konkurując ze sobą o zasoby miRNA, tworzą sieć regulatorową na skalę całego transkryptomu, co może częściowo wyjaśniać korelację między wielkością genomu i złożonością organizmu. O tym, że niekodujące RNA to nie tylko pozostałość po świecie RNA świadczą też ważne funkcjonalnie ustrukturyzowane cząsteczki pełniące role katalityczne i regulatorowe.

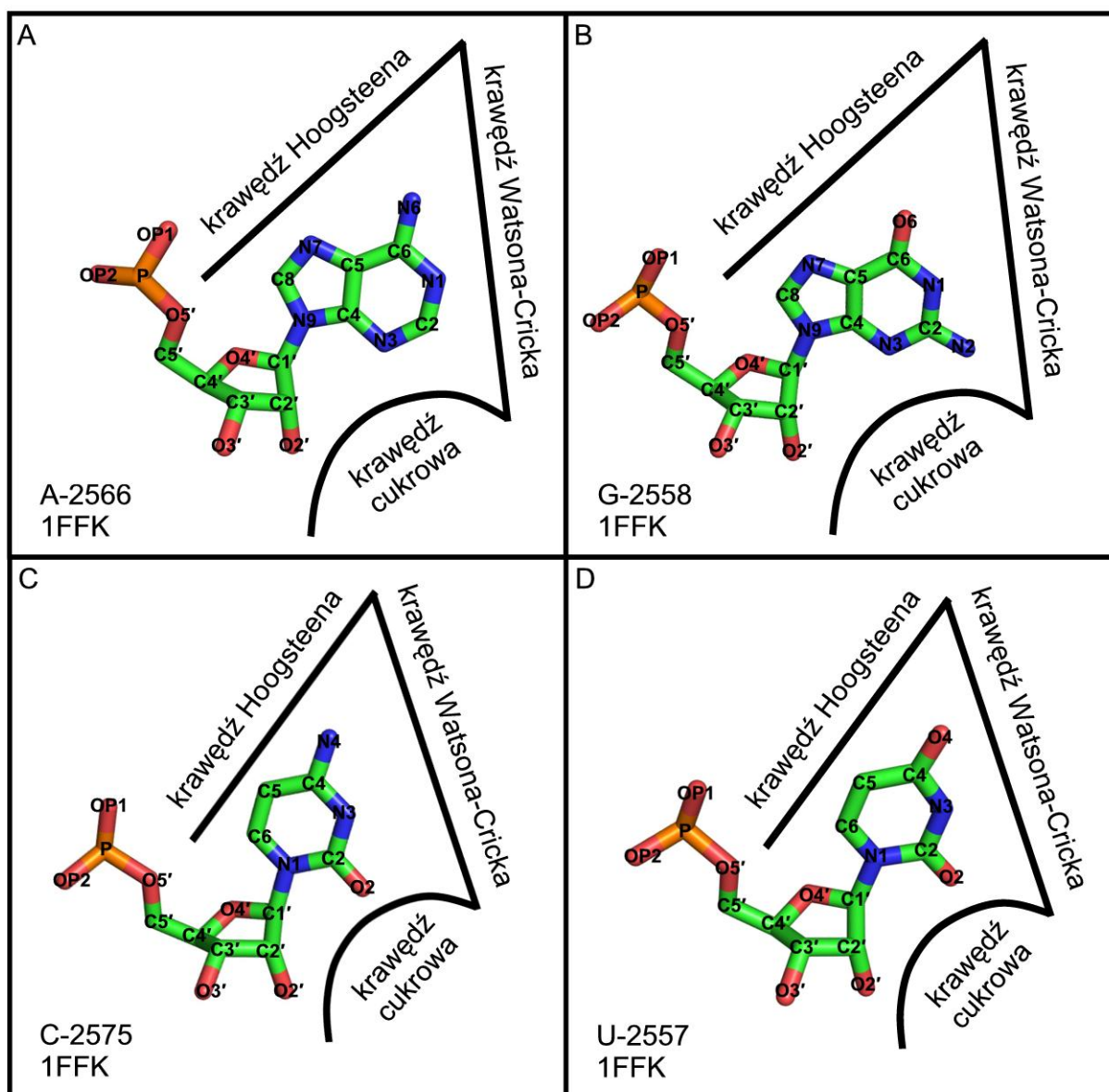
Przeprowadzanie reakcji katalitycznych wymaga precyzyjnego ułożenia atomów w przestrzeni, z tego powodu cząsteczki RNA posiadają dokładnie zdefiniowaną strukturę przestrzenną (Westhof and Auffinger, 2000) (rozdział 1.1). Znajomość struktury i jej związek z funkcją pozwalają wnioskować o mechanizmie działania rybozymów i regulatorowych RNA (Hoogstraten and Sumita, 2007). Zastosowanie cząsteczek RNA ma duży potencjał w medycynie i innych naukach (np. (Erdmann, i wsp., 2006), (Wheeler, i wsp., 2009)). Są prowadzone badania nad ich wykorzystaniem w terapii chorób zwyrodnieniowych, wirusowych, bakteryjnych oraz nowotworowych (rozdział 1.2). Praktyczne użycie ustrukturalizowanych cząsteczek RNA często wymaga znajomości ich struktury, jako że ta jest ściśle powiązana z funkcją. Liczba znanych struktur RNA zwiększa się, jednak doświadczalne wyznaczanie struktury pozostaje procedurą czasochłonną i kosztowną, uzależnioną od specjalistycznego sprzętu. Alternatywą są powstające metody *in silico* (rozdział 1.3). Praca niniejsza przedstawia program ModeRNA, który pozwala na komputerowe przewidywanie struktury trzeciorzędowej RNA, opierając się na strukturze spokrewnionej cząsteczki szablonu i przyrównaniu sekwencji cząsteczki przewidywanej i szablonu.

1.1. Budowa i funkcja RNA

RNA to liniowy polimer zbudowany z reszt rybonukleotydowych połączonych wiązaniami fosfodiesterowymi. Rybonukleotydy składają się z pięciowęglowego cukru rybozy, jednej z czterech zasad azotowych (adeniny, guaniny, cytozyny lub uracylu) oraz kwasu fosforowego. Pod względem swojej budowy chemicznej RNA przypomina cząsteczkę DNA. W odróżnieniu od DNA, monomery występujące w RNA tworzą skomplikowaną sieć oddziaływań, powodując, że RNA ma nie tylko strukturę pierwszorzędową (sekwencja), ale również może tworzyć strukturę drugo-, trzecio- i czwartorzędową. Pod względem budowy przestrzennej RNA jest więc bardziej zbliżone do białek. Ponadto reszty rybonukleotydowe w RNA, podobnie jak reszty aminokwasowe w białkach, podlegają potranskrypcyjnym (odpowiednio potranslacyjnym) modyfikacjom, które są ważne dla struktury i funkcji cząsteczki. Analogia do białek dotyczy również funkcji. RNA oprócz swojej roli w biosyntezie białek jest katalizatorem reakcji chemicznych oraz reguluje ekspresję genów.

1.1.1. Struktura pierwszorzędowa

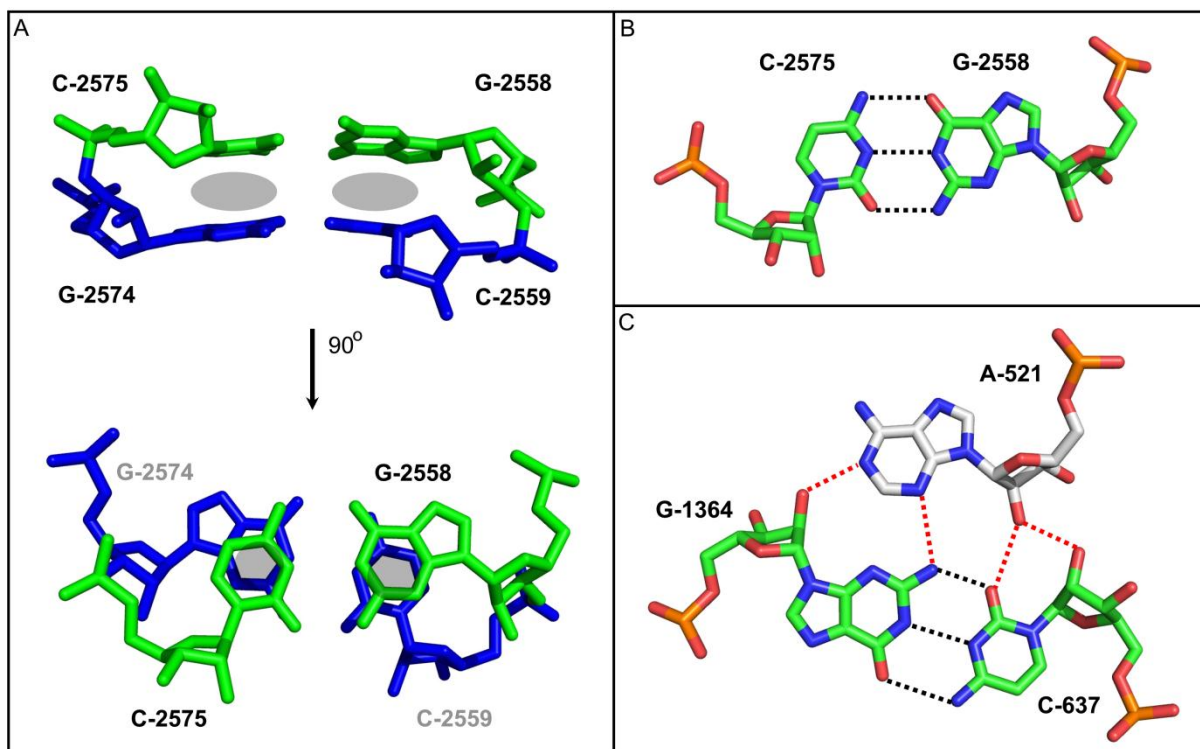
Kwas rybonukleinowy jest ujemnie naładowanym polimerem złożonym z czterech typów monomerów połączonych wiązaniem fosfodiesterowym. Pojedynczy monomer jest zbudowany z ufosforylowanego pierścienia rybozy (identycznego we wszystkich rybonukleotydach), do którego przyłączona jest poprzez wiązanie N-glikozydowe jedna z czterech zasad azotowych. W RNA występują zasady purynowe, czyli adenina i guanina, oraz pirymidynowe, czyli cytozyna i uracyl (Rycina 2). Pierwszy poziom organizacji cząsteczki RNA to sekwencja zasad przyłączonych do szkieletu cukrowo-fosforanowego. Pod względem budowy chemicznej RNA przypomina DNA, różni się tylko w dwóch aspektach: w DNA zamiast uracylu występuje tymina, która posiada dodatkową grupę metylową przy węglu C5. Druga zmiana to obecność grupy 2'-hydroksylowej w pierścieniu cukrowym. Te dwie niewielkie różnice chemiczne powodują duże różnice w strukturze i funkcji pomiędzy tymi molekułami. W odróżnieniu od DNA, a podobnie do białek, łańcuch RNA może związać się w skomplikowaną strukturę przestrzenną (Westhof and Auffinger, 2000).



Rycina 2. Reszty nukleotydów budujące RNA: A) fosforan 5'-adenozyny; B) fosforan 5'-guanozyny; C) fosforan 5'-cytydyny; D) fosforan 5'-urydyny. Zaznaczone krawędzie wskazują potencjalne miejsca oddziaływań z innymi resztami. W lewym dolnym rogu znajduje się informacja o pochodzeniu każdej z reszt (identyfikator reszty i struktury).

1.1.2. Oddziaływania pomiędzy zasadami

W tworzenie struktury wyższego rzędu (drugo- i trzeciorzędowej) zaangażowane są oddziaływania warstwowe oraz wiązania wodorowe (Rycina 3). Oddziaływania warstwowe pomiędzy zasadami kwasów nukleinowych stabilizują strukturę podwójnej helisy. Mogą również występować pomiędzy całymi helisami, gdy te oddzielone są po jednej stronie regionem niesparowanym (koosiowe oddziaływanie warstwowe między helisami). Powstają one w wyniku działania sił dyspersyjnych Londona, oddziaływania między ładunkami cząstkowymi pierścieni zasad, oraz sił hydrofobowych w rozpuszczalniku (Holbrook, 2008).



Rycina 3. Oddziaływania pomiędzy zasadami w RNA: A) oddziaływanie warstwowe; B) parowanie kanoniczne typu Watsona-Cricka; C) parowanie niekanoniczne (czerwone linie) w motywie A-minor.

W tworzenie struktury wyższego rzędu zaangażowane są również wiązania wodorowe pomiędzy zasadami. Są one tworzone przez donory protonu i akceptory protonu występujące w zasadach azotowych oraz rybozie. Na podstawie lokalizacji potencjalnych donorów i akceptorów w każdej zasadzie, wyznaczone zostały trzy krawędzie, przez które może zostać utworzone oddziaływanie z inną zasadą (Leontis and Westhof, 1998). W stabilizacji podwójnej helisy uczestniczą pary zasad AU oraz GC oddziałujące krawędziami Watsona-Cricka. Na poziomie atomowym pary te zaobserwowane zostały w strukturach krystalicznych w 1976 roku (Rosenberg, i wsp., 1976; Seeman, i wsp., 1976). W tworzeniu podwójnej helisy uczestniczy także tzw. chwiejna para (*wobble pair*) pomiędzy guaniną i uracylem, jest to oddziaływanie krawędziami Watsona-Cricka. Typowa dla RNA jest helisa A (Westhof and Auffinger, 2000), choć pod pewnymi warunkami RNA może także tworzyć inne typy helisy (np. helisę Z w roztworze 6M NaClO₄) (Adamiak, i wsp., 2009; Popena, i wsp., 2004).

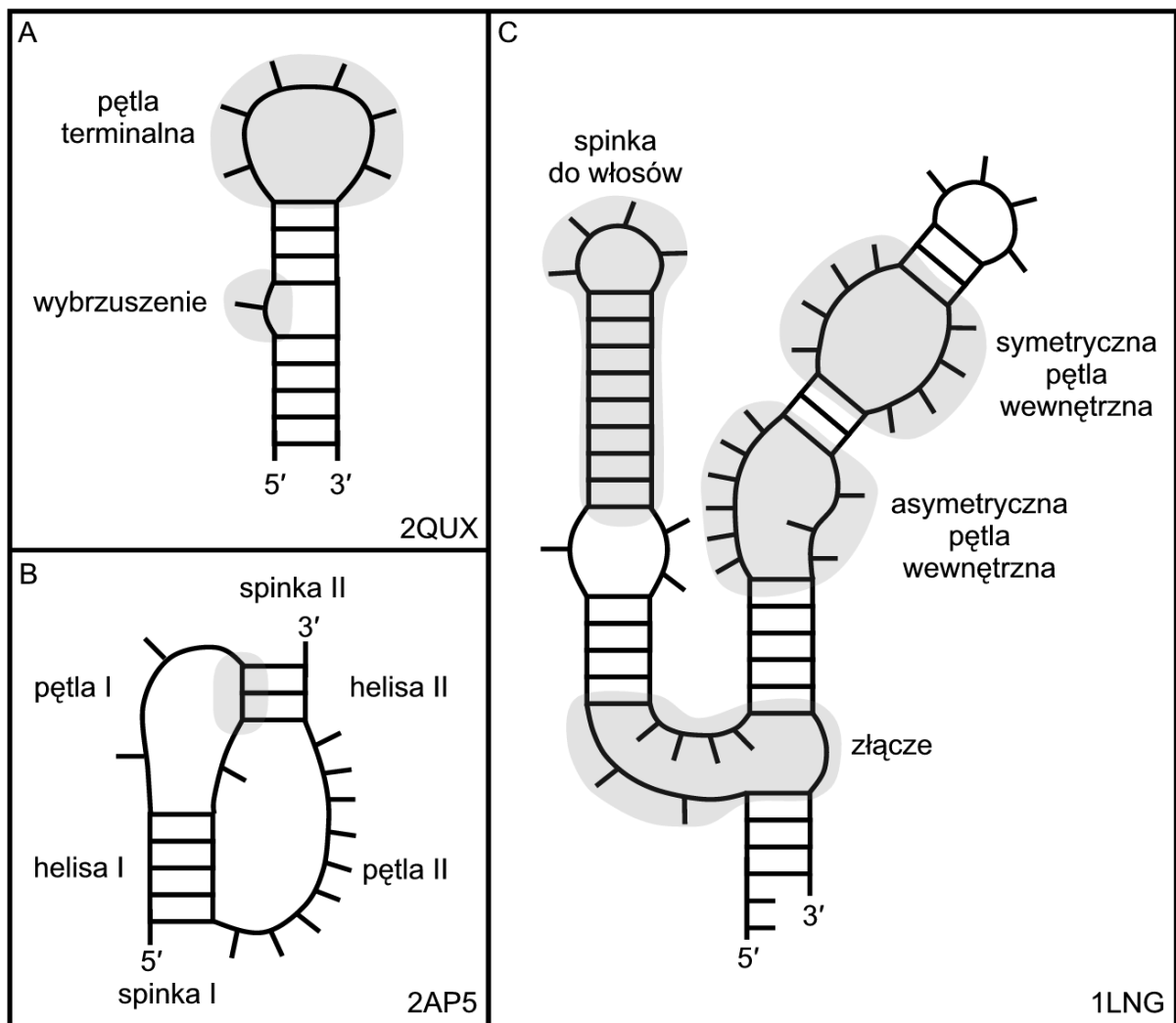
W strukturze RNA oprócz oddziaływań Watsona-Cricka występują również oddziaływania innego typu, w które zaangażowane są inne krawędzie. Tworzą one motywy pośredniczące w dalekozasięgowych oddziaływaniach RNA-RNA, oraz miejsca wiązania dla białek i ligandów (Leontis, i wsp., 2002). Trzy potencjalne miejsca tworzenia wiązań

wodorowych każdej zasady to krawędź Watsona-Cricka, krawędź Hoogsteena i krawędź cukrowa (Rycina 2). Oddziałujące ze sobą zasady mogą się też różnić orientacją wiązania glikozydowego w stosunku do oddziaływania wodorowego. Wyróżnić można konformację *cis* (wiązania glikozydowe są zorientowane po tej samej stronie wiązania wodorowego) i *trans* (wiązania glikozydowe znajdują się po przeciwnej stronie wiązania wodorowego). Rodzaje krawędzi oraz położenie wiązania glikozydowego stały się podstawą klasyfikacji par zasad, która opisuje dwanaście możliwych typów oddziaływań (Leontis, i wsp., 2002). Wszystkie z nich zaobserwowane zostały w strukturach rozwiązanych doświadczalnie, przykład oddziaływania innego typu niż Watsona-Cricka przedstawia Rycina 3 C.

Oddziaływania wodorowe mogą być również tworzone pomiędzy zasadami i cukrowo-fosforanowym szkieletem cząsteczki. Zaobserwowane zostały oddziaływania prostopadłe pomiędzy zasadami oraz oddziaływania rybonukleotydów, w które zaangażowany jest wyłącznie pierścień rybozy z jednej lub obu reszt (Leontis, i wsp., 2002). Jako akceptory wiązań wodorowych wykorzystywane mogą być atomy OP1, OP2, O2', O3', O4' lub O5' szkieletu cukrowo-fosforanowego cząsteczki, a jako donory np. ugrupowanie 2'OH (Ulyanov and James, 2010).

1.1.3. Struktura drugorzędowa

Opisane oddziaływania odpowiadają za strukturę drugorzędową i trzeciorzędową cząsteczki RNA. Strukturę drugorzędową można zdefiniować jako regiony podwójnej helisy utworzone przez pary zasad typu Watson-Crick, połączone przez regiony niesparowane. Element łączący końce helisy - koniec 3' jednej nici z końcem 5' drugiej nici, to pętla terminalna spinki do włosów. Dwa regiony jednoniciowe, łączące dwa segmenty podwójnej helisy, tworzą pętlę wewnętrzną, a w przypadku gdy region niesparowany występuje tylko po jednej stronie określa się go jako wybrzuszenie. Połączone trzy lub więcej segmenty dwuniciowe tworzą złącze (Holbrook, 2008). Przykłady elementów drugorzędowych, zaczerpnięte z rozwiązanych doświadczalnie struktur zdeponowanych w bazie PDB (*Protein Data Bank*), przedstawione zostały na Rycina 4.

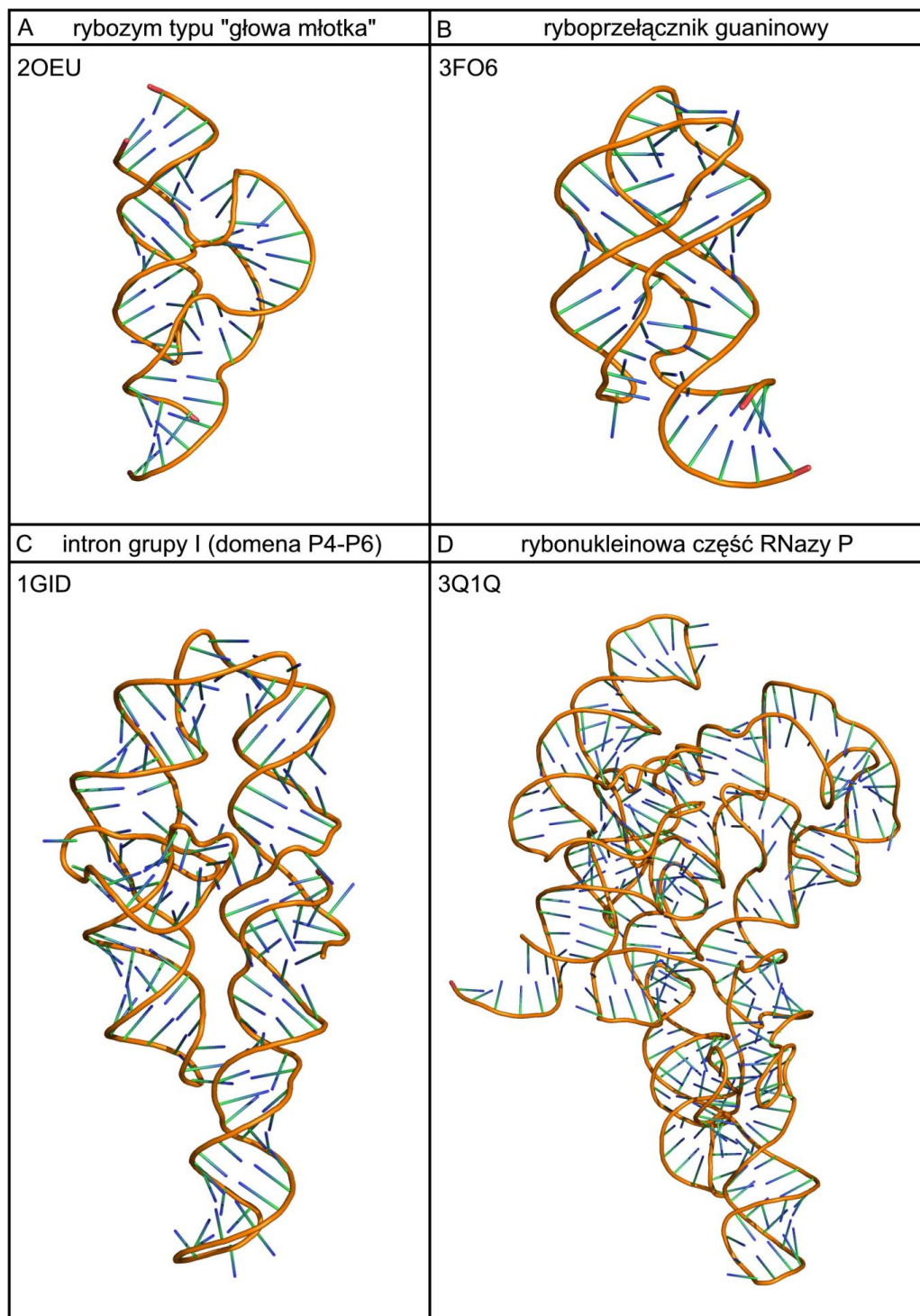


Rycina 4. Podstawowe typy motywów drugorzędowych na przykładzie struktur RNA: A) operatora translacji 2QUX (Chao, i wsp., 2008); B) pseudowężła z mRNA roślinnego luteowirusa 2AP5 (Cornish, i wsp., 2006); C) SRP19 archeowca *Methanococcus jannaschii* (Hainzl, i wsp., 2002) (struktura drugorzędowa na podstawie (Laing and Schlick, 2010)).

1.1.4. Struktura trzeciorzędowa

Kolejny poziom organizacji RNA to struktura trzeciorzędowa, która jest centralnym elementem zainteresowania tej pracy. Jest to układ przestrzenny tworzony przez oddziałujące ze sobą elementy drugorzędowe. Pierwsza poznana na poziomie atomowym struktura – drożdżowe tRNA^{Phe}, została rozwiązana w 1974 roku (Kim, i wsp., 1974) (Robertus, i wsp., 1974), a pierwszą strukturę katalitycznego RNA, rybozemu typu „głowy młotka”, opublikowano w 1994 roku (Pley, i wsp., 1994). Dzięki znajomości struktury możliwe było zaproponowanie mechanizmu reakcji katalitycznej (Scott, i wsp., 1995). Funkcja cząsteczki jest bowiem powiązana z przyjmowaną przez nią strukturą. Rosnąca liczba znanych cząsteczek RNA zdeponowanych w bazie PDB, pozwala na odkrywanie takich zależności (Hoogstraten and Sumita, 2007). Struktura przestrzenna rybozemu typu „głowy młotka”

(Rycina 5 A) oraz trzech innych cząsteczek RNA posiadających funkcję katalityczną lub regulatorową zostały przedstawione poniżej (Rycina 5). Ryboprzełącznik guaninowy przedstawiony w części B ryciny posiada więcej niż jedną aktywną funkcjonalnie konformację, podobnie jak inne ryboprzełączniki (Montange and Batey, 2008).



Rycina 5. Przykłady funkcjonalnych RNA posiadających stabilną strukturę trzeciorzędową: A) rybozom typu „głowa młotka” (kod PDB: 2OEU) (Martick, i wsp., 2008); B) ryboprzełącznik guaninowy (kod PDB: 3FO6) (Gilbert, i wsp., 2009); C) domena P4-P6 intronu grupy I (kod PDB: 1GID) (Cate, i wsp., 1996); D) rybonukleaza P (kod PDB: 3Q1Q) (Reiter, i wsp., 2010).

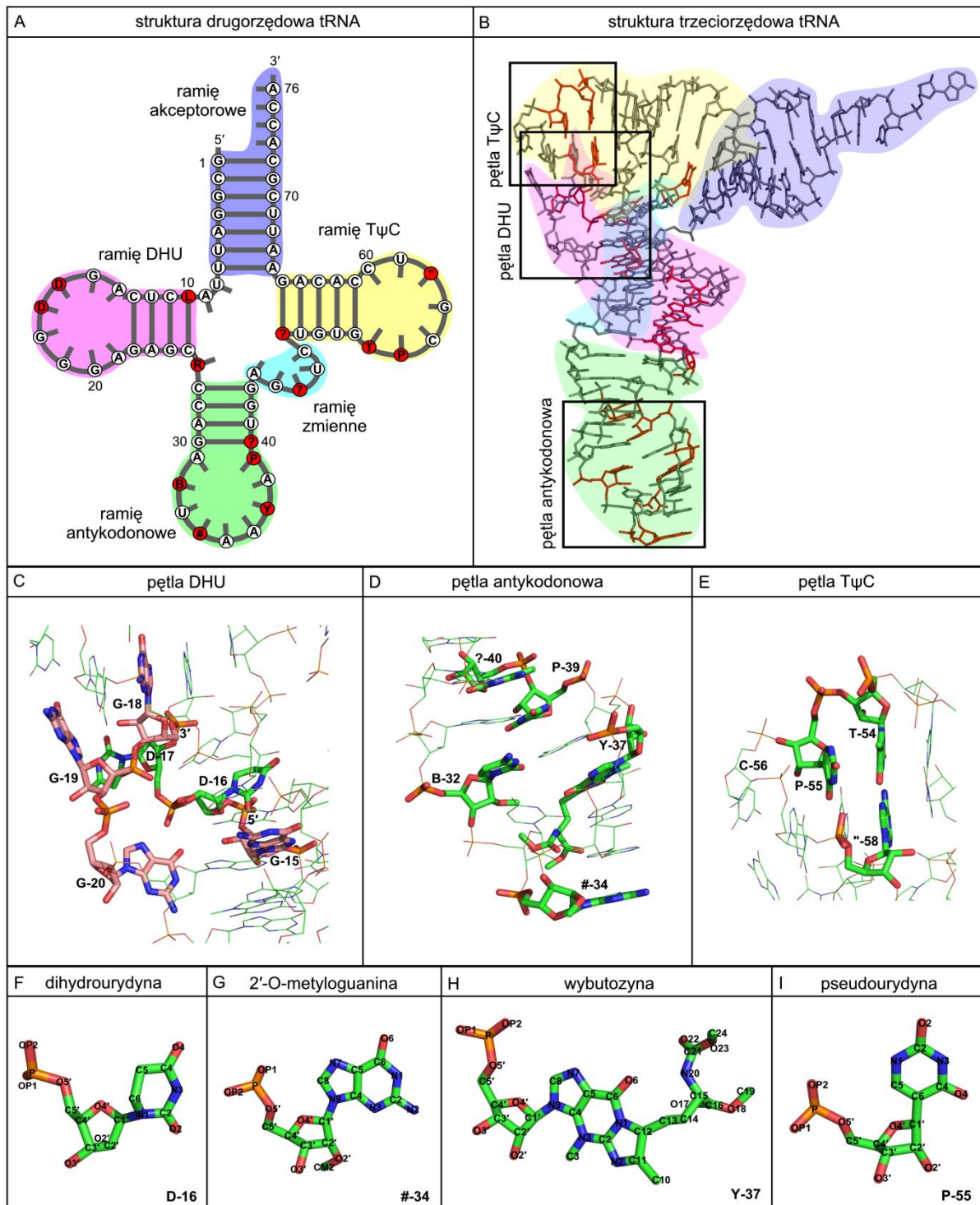
1.1.5. Proces zwijania się RNA

Stabilność i zwijanie struktury trzeciorzędowej zależą od obecności jedno i dwudodatnich jonów (przede wszystkim K^+ i Mg^{2+}). Kationy pozwalają na zwijanie cząsteczki redukując odpychanie pomiędzy negatywnie naładowanymi fosforanami oraz stabilizując oddziaływania trzeciorzędowe, uczestniczą także w reakcjach przeprowadzanych przez niektóre rybozomy (Donghi and Schnabl, 2011). Innym czynnikiem podczas zwijania się struktury jest prędkość transkrypcji. Jest ona modulowana prędkością elongacji oraz zależnymi od sekwencji przystąpięciami polimerazy RNA. Proces zwijania się rozpoczyna się natychmiast, gdy ulegająca transkrypcji nić osiągnie długość pozwalającą na parowanie zasad. Regiony ulegające transkrypcji wcześniej mogą się zatem zwijać faworyzując lokalnie stabilną strukturę i współzawodnicząc z bardziej stabilną strukturą globalną, która powstałaby z dłuższego transkryptu. Przerwy w transkrypcji pozwalają na utworzenie mniej stabilnej struktury tymczasowej, która pozwoli później na zwinięcie się cząsteczki w funkcjonalną strukturę natywną (Cruz and Westhof, 2009).

1.1.6. Potranskrypcyjne modyfikacje rybonukleotydów

Proces zwijania się, stabilność struktury oraz funkcja RNA może zależeć również od obecności modyfikowanych reszt nukleotydów w cząsteczce. Są one powszechne w wielu rodzajach RNA (tRNA, rRNA, mRNA, snRNA i innych) (Grosjean, 2009). Znanych jest ponad sto typów modyfikacji. Jedną z najlepiej zbadanych pod tym kątem cząsteczek jest tRNA, którego modyfikacje na niektórych pozycjach są silnie konserwowane (Motorin and Helm, 2010) (Rycina 6). Przykładowo dihydrourydyny obecne w pętli D destabilizują konformację C3'-endo rybozy modyfikowanego nukleotydu oraz nukleotydów sąsiadujących z nim na końcu 5', dzięki czemu pętla ta jest giętka i może oddziaływać z pętlą T (Dalluge, i wsp., 1996). Powstające w ten sposób oddziaływania trzeciorzędowe stabilizują modyfikacje występujące w pętli T niezbędne dla uzyskania typowej struktury trzeciorzędowej w kształcie litery L. Silnie konserwowane modyfikacje w pętli antykodonowej ograniczają jej dynamikę i zwiększają dokładność wiązania kodonu (Agris, 2008). Konserwowane modyfikacje pętli D, T i antykodonowej zostały przedstawione na przykładzie struktury drożdżowego tRNA^{Phe} (Rycina 6). O ważnej roli modyfikacji może też świadczyć przykład ludzkiego mitochondrialnego tRNA^{Lys}: Niemodyfikowana cząsteczka uzyskuje strukturę drugorzędową spinki. Dopiero metylacja adeniny w pozycji dziewiątej (m^1A), powstająca na etapie

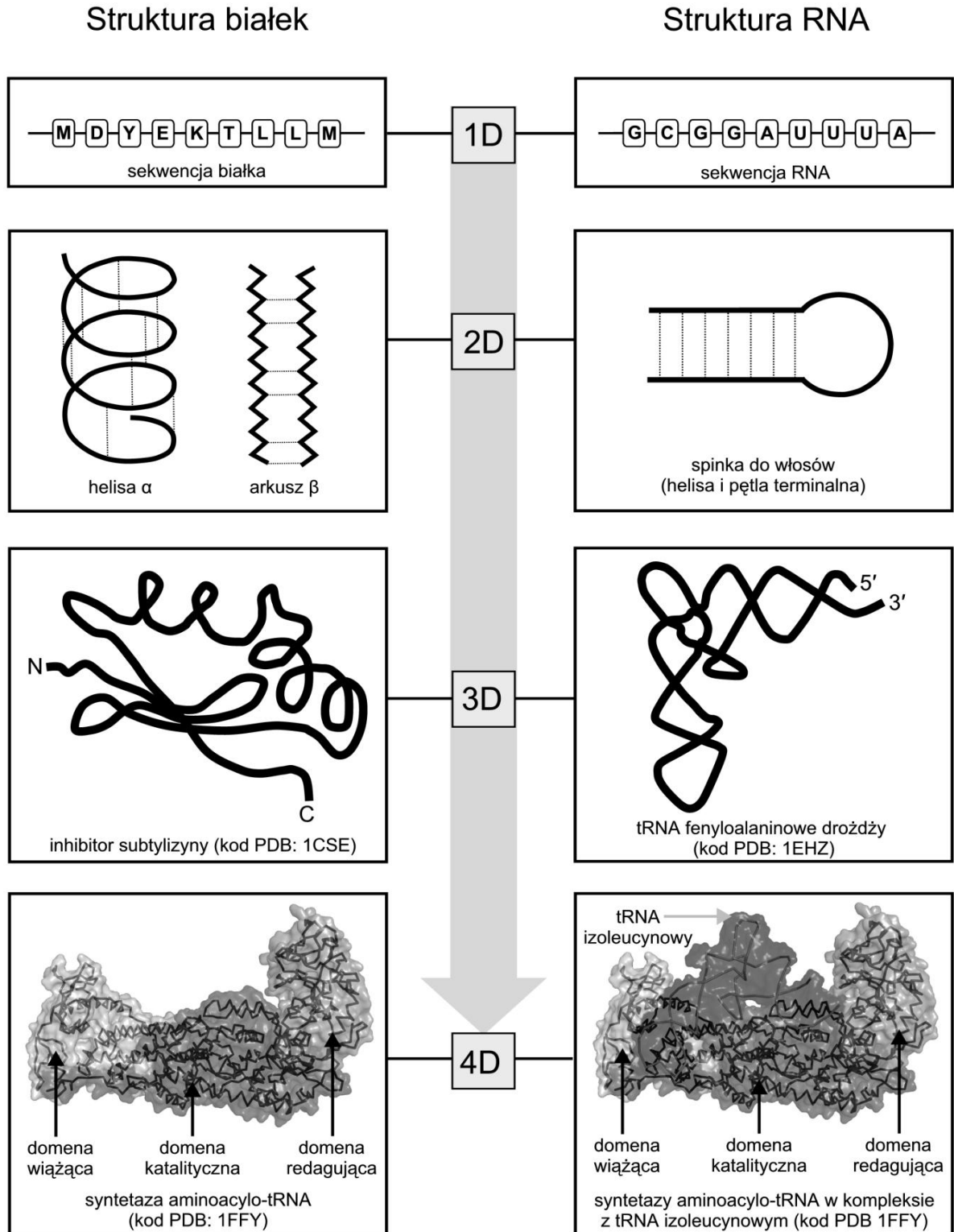
wczesnego dojrzewania, powoduje przesunięcie równowagi w stronę typowej struktury liścia koniczyny (Motorin and Helm, 2010).



Rycina 6. Modyfikowane nukleotydy występujące w tRNA na przykładzie struktury 1EHZ: A) rozmieszczenie modyfikowanych nukleotydów w strukturze drugorzędowej; B) rozmieszczenie modyfikowanych nukleotydów w strukturze trzeciorzędowej; C-E) konformacja fragmentów struktury zawierających najbardziej konserwowane modyfikacje; F-I) struktury pojedynczych modyfikowanych nukleotydów. Na podstawie (Rother, i wsp., 2011).

1.1.7. Podobieństwa RNA i białka

RNA zwija się spontanicznie w natywną strukturę przestrzenną dzięki informacji zakodowanej w sekwencji złożonej z czterech podstawowych rodzajów elementów (rybonukleotydów) rozszerzonych o potranskrypcyjne modyfikacje. W strukturze wyróżnić można hierarchię – kilka rzędów złożoności. Pod tymi względami RNA bardzo przypomina białka, które również są zbudowane z ograniczonej liczby elementów (aminokwasów), mogących ulegać potranslacyjnej modyfikacji. Podobnie jak RNA większość cząsteczek białka zwija się spontanicznie, a ich struktura ma charakter hierarchiczny (Rycina 7). Podstawową różnicą między cząsteczkami RNA i białka jest całkowicie odmienna budowa ich elementów składowych (aminokwasów i rybonukleotydów). Ponadto struktura drugorzędowa białka utrzymywana jest przez wiązania wodorowe między elementami łańcucha głównego, podczas gdy w RNA wiązania wodorowe tworzą się przez oddziałujące ze sobą zasady („łańcuchy boczne”). Proces zwijania się cząsteczki jest wywołany przez inne zjawiska: W przypadku RNA jest to neutralizacja negatywnych ładunków grup fosforanowych, podczas gdy w białkach grupy boczne hydrofobowych reszt aminokwasowych zagrzebywane są wewnątrz struktury i chronione przed rozpuszczalnikiem. Skutkuje to innym kształtem obu cząsteczek. Białka mają architekturę sferyczną (poza niektórymi białkami przeźbłonowymi i włóknami np. kolagenem i miozyną), a RNA, którego hydrofobowe pierścienie zasad chronione są przed rozpuszczalnikiem we wnętrzu podwójnej helisy, mają kształt bardziej spłaszczony. Tworzą „spodki” o grubości odpowiadającej średnicy pojedynczej helisy lub grubszej dla dużych cząsteczek (np. 16S rRNA ma trzykrotną grubość helisy) (Holbrook, 2008). W obu przypadkach można jednak powiedzieć, że procesem zwijania się rządzą te same prawa fizyko-chemiczne, a struktura cząsteczki zapewnia szkielet dla jej biologicznej funkcji.



Rycina 7. Hierarchiczna budowa RNA i białek, na podstawie (Rother, i wsp., 2011).

1.1.8. Funkcje RNA

Podobnie jak białka cząsteczki RNA mogą mieć aktywność katalityczną. Przyspieszają reakcję nawet 10^{11} razy, jak w przypadku intronów grupy I (Wilson and Lilley, 2009). Są wyposażone w różne strategie katalityczne, oparte na transferze protonów, lub wykorzystaniu jonów metali. Samodzielnie lub też wspólnie z białkami (rybonukleoproteiny, RNP) przeprowadzają reakcje hydrolizy, transestryfikacji czy syntezy wiązania peptydowego (Wilson and Lilley, 2009) (Hoogstraten and Sumita, 2007). RNA ma również funkcje regulatorowe sprawowane np. przez mikroRNA czy ryboprzełączniki. Przykłady funkcjonalnych, ustrukturalizowanych RNA przedstawia Tabela 1. Naturalnie występujące rybozomy i cząsteczki regulatorowe mogą zostać zmodyfikowane przez inżynierię genetyczną i zostać wykorzystane do badań biotechnologicznych i nanotechnologicznych czy w terapii chorób różnego pochodzenia (wirusowych, bakteryjnych, zwyrodnieniowych i nowotworowych) (rozdział 1.2).

Tabela 1. Przykłady cząsteczek RNA pełniących funkcje katalityczne i/lub regulatorowe.

Cząsteczka	Opis
Rybozym typu „głowa młotka”	Został znaleziony w sensownej nici satelitarnego RNA wirusa mozaiki tytoniowej. Kowalencyjnie zamknięty kołowy RNA satelitarny jest substratem dla polimerazy gospodarza. Podczas replikacji powstaje komplementarna forma liniowa, która zawiera wiele kopii kołowego RNA. Funkcją rybozomu typu „głowa młotka” jest rozcięcie poszczególnych kopii, oraz prawdopodobnie ich dalsza ligacja w formę kołową (Scott, i wsp., 2009).
Intron grupy I	Cząsteczki z tej grupy występują w wielu organizmach, m. in. w glonach, porostach, grzybach oraz u niektórych bakterii. Nie są niezbędne do funkcjonowania komórki. Są samolubnymi elementami genetycznymi, których strategią przetrwania jest „wklejanie się” do różnych genów, jednocześnie nie będąc szkodliwym dzięki umiejętności samowycinania się poprzez przeprowadzanie dwustopniowej reakcji transestryfikacji z użyciem guanozyny jako kofaktora (Vicens and Cech, 2006).
Ryboprzełączniki	Elementy regulatorowe występujące najczęściej w regionach niepodlegających translacji na końcu 5' (5'-UTR) bakteryjnego mRNA. Reagują bezpośrednio z metabolitami komórkowymi, co wpływa na ekspresję regionu kodującego poprzez zmianę w strukturze drugorzędowej ryboprzełącznika (Garst, i wsp., 2011).
RNaza P	Kompleks rybonukleoproteinowy katalizujący dojrzewanie końca 5' prekursorowego tRNA we wszystkich organizmach. Składa się z P RNA oraz przynajmniej jednej podjednostki białkowej, niezbędnej

	do aktywności cząsteczki <i>in vivo</i> . Podczas gdy RNA jest odpowiedzialny za katalizę reakcji odcięcia sekwencji na końcu 5' pre-tRNA, białko stabilizuje strukturę rybonukleazy oraz zwiększa jej powinowactwo do substratu i jonów (Smith, i wsp., 2007).
Telomeraza	Kompleks rybonukleoproteinowy katalizujący dodawanie powtórzonych sekwencji telomerowych na końcu 3' chromosomu odgrywając przez to decydującą rolę w utrzymywaniu stabilności genomu. Komponent RNA telomerazy nie tylko zapewnia szablon do syntezy telomerowego DNA, ale jest niezbędny dla jej funkcji katalitycznej, akumulacji, lokalizacji i składania całego kompleksu (Zhang, i wsp.).
Rybosom	Rybonukleoproteina odpowiedzialna za tłumaczenie informacji genetycznej zakodowanej w mRNA do białka. Składa się z dwóch podjednostek – dużej i małej. Każda z nich zawiera rRNA oraz kilkadziesiąt białek. Miejsce aktywne katalizuje tworzenie wiązania peptydowego. Tworzone jest ono całkowicie przez RNA dużej podjednostki, a najbliższe aminokwasy znajdują się w promieniu 18 Å. Mała podjednostka odpowiedzialna jest za poprawne odczytywanie kodu. Sprawdzana jest tu zgodność między antykodonem tRNA i kodonem mRNA (Simonovic and Steitz, 2009).
Spliceosom	Jest jednym z największych i najbardziej skomplikowanych kompleksów w komórce. Odpowiada za wycinanie intronów z pre-mRNA. U człowieka jest zbudowany ze 150 podjednostek białkowych oraz pięciu cząsteczek snRNA. Podjednostka U6 zawiera sekwencję podobną do sekwencji miejsca aktywnego intronu grupy II. Obecność domeny „ <i>H-like</i> ” w białkach spliceosomu może świadczyć o tym, że one również są zaangażowane w tworzenie miejsca aktywnego (Valadkhan, 2010).

1.2. RNA w nauce i medycynie

Właściwości RNA są wykorzystywane praktycznie w nanotechnologii, biologii i medycynie. Ze względu na hierarchiczne i modułowe właściwości struktury RNA jest on doskonałym obiektem do badań nad samoorganizowaniem się materii i tworzenia nanoukładów o ściśle kontrolowanej geometrii. RNA o właściwościach katalitycznych są wykorzystywane również do tworzenia bibliotek pozwalających na odkrywanie funkcji nowych genów. Katalityczne RNA mają wiele zastosowań w medycynie, podobnie jak antysensowne oligonukleotydy oraz RNA interferencyjne (RNAi) (np. (Purzycka and Adamiak, 2008)). Pozwalają na wyciszenie lub naprawę nietypowej ekspresji endogennych genów, która jest przyczyną wielu chorób zwyrodnieniowych (np. dystrofii mięśniowej). Zastosowane mogą tu być rybozomy tnące wybrane sekwencje (np. rybozym typu „głowy młotka”) lub pozwalające wymienić fragment sekwencji mRNA (np. intron grupy I). W terapii przeciwnowotworowej RNA dają potencjalną możliwość zastosowania przeciwko

różnego rodzaju rakom i przede wszystkim zmniejszenia toksyczności leczenia, a więc zminimalizowania efektów ubocznych, które w klasycznych terapiach są często bardzo szkodliwe. Katalityczne RNA umożliwiają również walkę z infekcjami wirusowymi. Przykładowo prowadzone są próby kliniczne w zastosowaniu rybozymu typu „głowa młotka” przeciwko wirusowi HIV-1. Ryboprzełączniki, które podobnie jak rybozomy mają skomplikowaną strukturę przestrzenną, są atrakcyjnym celem w terapiach przeciw szczepom bakterii opornym na antybiotyki, regulują one, bowiem wiele kluczowych procesów życiowych organizmów bakteryjnych. Poniżej omówione są konkretne przykłady praktycznego zastosowania ustrukturalizowanych RNA.

1.2.1. Ryboprzełączniki w leczeniu chorób bakteryjnych

Bakterie zamieszkują różne nisze i są narażone na ciągłe zmiany środowiska. Jako organizmy o wysokich zdolnościach adaptacyjnych rozwinęły liczne mechanizmy sensorowe w celu ekspresji odpowiednich genów w odpowiedzi na dane środowisko, co pozwala na oszczędzanie energii. Wśród różnorodnych mechanizmów regulujących ekspresję genów bakterie wykorzystują RNA zarówno, jako elementy *trans*, które oddziałują z docelowym mRNA przez parowanie zasad, jak i elementy *cis*, m. in. ryboprzełączniki występujące w cząsteczce mRNA, która podlega regulacji. W ten sposób mogą być regulowane m. in. geny odpowiadające za wirulencję. Jest to szczególnie ważne dla wielu bakteryjnych patogenów, które stają się wirulentne w odpowiedzi na sygnał gospodarza, aby uniknąć jego systemu odporności (Geissmann, i wsp., 2006). Przykładowo ryboprzełącznik guaninowy jest kluczowy dla zjadliwości gronkowca złocistego (*Staphylococcus aureus*) w komórkach ssaków (Mulhbacher, i wsp., 2010).

Ryboprzełączniki występują głównie w niepodlegających translacji regionach bakteryjnego mRNA. Są zbudowane z dwóch domen: aptameru i platformy ekspresyjnej. Aptamer rozpoznaje specyficzny ligand, natomiast platforma ekspresyjna kontroluje ekspresję genu poprzez przyjęcie jednej z dwóch wzajemnie się wykluczających konformacji zależnych od oddziaływania z ligandem. Ryboprzełączniki stanowią nowe rozwiązanie dla terapii przeciw szczepom bakterii opornych na antybiotyki. Mogą zostać wykorzystane jako cele w terapii, gdzie antagonistyczny ligand uniemożliwia ekspresję genu związanego z ryboprzełącznikiem. Obecnie prowadzone są badania nad wykorzystaniem w rozwoju leków przeciwbakteryjnych, między innymi: ryboprzełącznika tiaminopirofosforanowego (TPP) (negatywnie reguluje ekspresję białek zaangażowanych w biosyntezę i transport tiaminy),

ryboprzełącznika lizynowego (bierze udział w regulacji syntezy i transportu lizyny), ryboprzełącznika FMN (odpowiedzialny za ekspresję genu *LacZ*) i ryboprzełącznika guaninowego (reguluje syntezę i transport puryn) (Mulhbacher, i wsp., 2010).

1.2.2. Rybozimy wykorzystywane w medycynie

Istnieje również wiele opublikowanych prac o terapeutycznym wykorzystaniu RNA katalitycznych. Szczegółowo scharakteryzowane zostały: rybozym typu „głowa młotka”, rybozym spinki do włosów, intron grupy I, część rybonukleinowa rybonukleazy P i rybozym wirusa zapalenia wątroby typu D (rybozym HDV). Grupa tych pięciu, występujących naturalnie rybozymów, jest najczęściej używana jako podstawa do projektowania rybozymów modyfikowanych o działaniu terapeutycznym. Najpopularniejszy z nich to rybozym typu „głowa młotka”, który jest najmniejszym i najlepiej poznanym RNA katalitycznym. Jego nazwa jest związana z charakterystycznym kształtem struktury drugorzędowej. Został odkryty w infekcyjnym satelitarnym RNA roślinnym. Aktywność wewnątrzcząsteczkowego cięcia, którą posiada ten rybozym, jest konieczna do replikacji cząsteczki wirusoidu. Rybozimy typu „głowy młotka” są modyfikowane genetycznie tak, że mogą ciąć dowolnie wybrany RNA poprzez międzycząsteczkowy atak *trans* w obecności jonów Mg^{2+} . Taka modyfikacja została osiągnięta przez zmianę ramienia odpowiedzialnego za rozpoznawanie substratu. Cięcie substratu następuje zaraz po sekwencji NUX (gdzie N oznacza dowolną zasadę, a X to A, C lub U) w docelowej cząsteczce (Kalota, i wsp., 2006).

1.2.3. Rybozimy w terapii antywirusowej

Rybozym typu „głowy młotka” został wykorzystany między innymi w terapii przeciwko wirusowi HIV-1. Wirus ten rozpoznaje glikoproteinę CD4 jako główny receptor oraz używa glikoprotein CXCR4 lub CCR5 jako receptora pomocniczego przy infekowaniu komórki. Glikoproteiny CD4 oraz CXCR4 są niezbędne dla komórki, jednak obecność glikoproteiny CCR5 nie jest konieczna. Prowadzone są badania nad wyciszeniem genu glikoproteiny CCR5 przy użyciu kilku rybozymów typu „głowa młotka” RZ1-7, których celem jest siedem niepowtarzalnych miejsc w ludzkim mRNA genu glikoproteiny CCR5. Podczas dwumiesięcznego doświadczenia użycie rybozymu spowodowało prawie całkowite zahamowanie replikacji odmiany R5 wirusa HIV-1. Inny rodzaj rybozymu typu „głowy młotka” (OZ1) był też testowany w celu zastąpienia terapii HAART (wysoko aktywna terapia przeciwretrowirusowa), która jest często toksyczna i może prowadzić do niekorzystnych oddziaływań z innymi lekami oraz powstawania opornych szczepów wirusa HIV. Druga faza

testów terapii genowej, bazującej na rybozymie OZ1, przeprowadzona na grupie 74 pacjentów, zakończyła się pozytywnie sugerując, że również ten rybozym może zostać użyty w terapii przeciwko wirusowi HIV-1 (Mulhbacher, i wsp., 2010).

1.2.4. Rybozimy w terapii antynowotworowej

W zwalczaniu nowotworów katalityczne RNA są atrakcyjnymi kandydatami na leki, potencjalnie rozwiązując problemem wysokiej toksyczności klasycznych terapii. Cząsteczki RNA (rybozimy, a także antysensowny RNA) są skierowane przeciwko specyficznym genom, które podlegają nieprawidłowej ekspresji w chorych komórkach i w związku z tym powodując mniejsze i mniej poważne efekty uboczne (Kalota, i wsp., 2006). Przykłady badań obejmują rybozym typu „głowy młotka” skierowany przeciwko HER-2. HER-2 jest członkiem rodziny receptorów epidermalnego czynnika wzrostu (EGF), kodującym białko p-185, którego nadekspresja towarzyszy 20-30% nowotworów złośliwych pochodzenia nabłonkowego, które są często związane z rakiem piersi. Projektowane są również rybozimy typu „głowa młotka”, których celem jest mRNA surwiwiny. Białko to jest czynnikiem przeciwdziałającym apoptozie i podlega nadekspresji we wszystkich typach raka (jest używane jako biopskaźnik w diagnostyce raka). Przykład innego rybozimu, mającego potencjalne zastosowanie w terapii antynowotworowej to intron grupy I (Mulhbacher, i wsp., 2010).

1.2.5. Rybozimy w terapii chorób zwyrodnieniowych

Trwają również badania nad zastosowaniem rybozymów w leczeniu chorób zwyrodnieniowych, np. dystrofii mięśniowej typu 1 (DM1). Jest to choroba nerwowo-mięśniowa, charakteryzująca się dużą liczbę powtórzeń CTG usytuowanych w regionie 3' UTR genu DMPK. Transkrypt zmutowanego genu DMPK z regionem 3' UTR zawierającym zwiłokrotnioną liczbę powtórzeń CUG akumuluje się i tworzy skupiska RNA w jądrze komórek DM1. Jeden z zaproponowanych mechanizmów zwalczania tej choroby to usuwanie skupisk nagromadzonego mRNA poprzez cięcie go przez rybozym typu „głowy młotka”. Inne podejście to wykorzystanie intronu grupy I w celu modyfikowania liczby powtórzeń CUG w ludzkim transkrypcie DMPK. W dotychczasowych próbach osiągnięte zostało zastąpienie dwunastu powtórzeń pięcioma, zarówno *in vitro* jak i w komórkach ssaków. Inne choroby zwyrodnieniowe, w których terapii pomoc mogą RNA o właściwościach katalitycznych, to między innymi miażdżycy, nadciśnienie, choroba Alzheimera oraz choroba Parkinsona (Mastroiannopoulos, i wsp., 2010).

1.2.6. Przeszkody w praktycznym zastosowaniu rybozymów w medycynie

Problemem przy stosowaniu rybozymów jako terapeutyków jest racjonalny wybór miejsc docelowych w mRNA, stabilność rybozymu wewnątrz komórki i dostarczenie terapeutyku do odpowiednich komórek (Kalota, i wsp., 2006). W zwiększaniu stabilności pomocne są modyfikacje zasad reszt nukleotydowych rybozymu. Jednak te z kolei mogą negatywnie wpływać na jego aktywność katalityczną (Sioud, 2006). Pomimo tych trudności, jak wskazują opisane powyżej przykłady, funkcjonalne, ustrukturyzowane RNA mają duży potencjał zastosowania w terapiach różnych chorób.

1.2.7. Rybozimy w identyfikacji funkcji genów

Ustrukturyzowane, funkcjonalne RNA są stosowane nie tylko w medycynie, ale także w badaniach innych dziedzin naukowych. Przykładowo katalityczne RNA wykorzystuje się do identyfikacji funkcji genów poprzez wprowadzenie biblioteki aktywnych rybozymów do komórki, z późniejszymi badaniami przesiewowymi na zmiany fenotypowe. Technologia rybozymów uzupełnia inne narzędzie, bazujące na bibliotekach małych interferencyjnych RNA (siRNA). Ze względu na wysoką specyficzność rybozymów i wysoką aktywność siRNA, obie technologie uzupełniają się (Matsumoto, i wsp., 2006).

1.2.8. Wykorzystanie ustrukturalizowanych RNA w nanotechnologii

RNA jest także stosowany w nanotechnologii. Właściwości strukturalne tej cząsteczki pozwalają na wykorzystanie jej w badaniach nad samoorganizacją materii. Fragmenty RNA o takiej samej geometrii z „lepkimi końcami” mogą być wykorzystane jako części do tworzenia okresowych nanomateriałów o predefiniowanej geometrii. Elementy takie mogą służyć do organizowania z dużą precyzją, we względnie zdefiniowanych pozycjach, różnych komponentów molekularnych, oraz do budowy nanoczipów, nanoobwodów i nanokryształów (Chworos, i wsp., 2004).

1.3. Sposoby ustalania struktury przestrzennej RNA

W badaniach nad funkcjonalnymi RNA istotną rolę odgrywa znajomość ich struktury przestrzennej. To ona stanowi szkielet dla ich biologicznej funkcji. Istnieje wiele technik doświadczalnych, które pozwalają na uzyskanie informacji o strukturze lub jej różnych aspektach. Wyznaczenie dokładnego położenia atomów w cząsteczce, a zatem jej dokładnej struktury trzeciorzędowej, jest jednak drogie i czasochłonne (rozdział 1.3.1). Odczytywanie

sekwencji genów i całych genomów jest doświadczeniem o wiele tańszym i szybszym. Sprawia to, że liczba znanych sekwencji jest dużo większa od liczby znanych struktur, a różnica ta ciągle się powiększa. We wrześniu 2010 roku liczba znanych struktur RNA dostępnych w bazie PDB wynosiła 1848, a liczba znanych sekwencji dostępnych w bazie Rfam (baza zawierająca przyrównania sekwencji oraz modele kowariancji rodzin niekodujących RNA) równa była 1101833 (Rother, i wsp., 2011). Sytuacja ta przypomina analogiczny problem ze świata białek. Rozumienie kodu 1D-3D oraz prawidłowości rządzących procesem ewolucji cząsteczek daje podstawy dla teoretycznego przewidywania struktury trójwymiarowej. Zidentyfikowanych zostało kilka strategii, które zostały z sukcesem zastosowane dla komputerowego przewidywania struktury białka (Rother, i wsp., 2011). Niniejsza praca opisuje program, który pozwala na przewidywanie struktury przestrzennej RNA oraz przedstawia obecny stan wiedzy bioinformatycznej w dziedzinie modelowania RNA (rozdział 1.3.2).

1.3.1. Metody doświadczalne ustalania struktury RNA

Istnieje wiele metod doświadczalnych, które dostarczają informacji o różnych aspektach struktury RNA (Tabela 2). Różnią się one między sobą rodzajem uzyskiwanej informacji oraz nakładami finansowymi, czasowymi i sprzętowymi, których wymaga doświadczenie. Dostępność łańcucha głównego dla rozpuszczalnika, lokalizacja oddziaływań warstwowych i wodorowych, lokalna dynamika cząsteczki oraz oddziaływania dalekozasięgowe mogą być badane przez różnego rodzaju sondy chemiczne i enzymatyczne (Weeks, 2010). Opracowany został standard współdzielenia danych pochodzących z tych doświadczeń, obejmujący ujednoliczoną nomenklaturę, wspólny format reprezentacji danych (SNRNASM) oraz procedurę udostępniania danych (Rocca-Serra, i wsp., 2011). Sondy chemiczne pozwoliły przykładowo na ustalenie oddziaływań podczas dimeryzacji wirusa HIV-2 (Purzycka, i wsp., 2011). Na badanie pewnych aspektów struktury RNA, takich jak obecność modyfikacji, czy oddziaływań RNA-białko, pozwala też spektroskopia mas (Thomas and Akoulitchev, 2006). Bardziej ogólnych informacji, na poziomie kształtu całej cząsteczki, dostarcza kriomikroskopia elektronowa. Ta technika pozwala na wizualizację w niskiej rozdzielczości dużych kompleksów takich jak rybosom, czy spliceosom (Budkevich, i wsp., 2011), (Golas, i wsp., 2010).

Tabela 2. Metody doświadczalne pozwalające na uzyskiwanie informacji o strukturze RNA. Na podstawie (Felden, 2007).

Metoda	Opis
Krystalografia rentgenowska	Rejestracja obrazów dyfrakcyjnych promieni rentgenowskich oddziałujących z chmurami elektronów atomów w kryształach. Pozwala na uzyskanie struktury trójwymiarowej na poziomie atomowym. Wymaga uzyskania kryształów wysokiej jakości.
Rozpraszanie rentgenowskie pod małym kątem (SAXS)	Rejestracja obrazów dyfrakcyjnych promieni rentgenowskich rozproszonych pod małym kątem. Ta metoda pozwala uzyskać dane o niskiej rozdzielczości, takie jak kształt i rozmiar cząsteczki. Nie jest wymagana obecność kryształu.
Spektroskopia magnetycznego rezonansu jądrowego (NMR)	Zapis promieniowania elektromagnetycznego emitowanego po wzbudzeniu spinów jądrowych w polu magnetycznym. Umożliwia poznanie struktury atomowej małych i średnich cząsteczek w rozpuszczalniku.
Kriomikroskopia elektronowa	Pozwala na wizualizację dużych kompleksów oraz ich dynamiki w niskiej rozdzielczości (10 – 13 Å).
Spektroskopia mas (MS)	Polega na jonizacji badanej substancji, a następnie badaniu odchylenia strumienia jonów w polu elektrycznym. Służy do analizy sekwencji i struktury oraz identyfikacji modyfikowanych nukleotydów.
Sondowanie enzymatyczne	Badanie oddziaływań drugo i trzeciorzędowych przy użyciu enzymów np. nukleazy S1, rybonukleazy V1, rybonukleazy T1.
Sondowanie chemiczne	Badanie oddziaływań drugo i trzeciorzędowych przy użyciu substancji chemicznych np. imidazolu, ketoksalu.
Mapowanie interferencji analogów nukleotydów (NAIM)	Mapowanie zakłóceń wprowadzonych do struktury poprzez umieszczenie w RNA w losowych miejscach analogów nukleotydów. Umożliwia identyfikację nukleotydów istotnych dla funkcji cząsteczki.
Tłumienie zakłóceń analogów nukleotydów (NAIS)	Identyfikacja analogów zasad, które zakłócają funkcję danej populacji, jednocześnie nie obniżając aktywności populacji z przygotowanego w tle doświadczenia NAIM. Umożliwia szukanie partnerów oddziaływań trzeciorzędowych.
Miejscowo specyficzna mutageneza	Wprowadzanie syntetycznych modyfikacji w znanej pozycji. Pozwala na określenie znaczenia pojedynczych grup dla funkcji i struktury cząsteczki.
Miejscowo specyficzne fotozszywanie	Monitorowanie oddziaływań RNA-RNA i RNA-białko poprzez wprowadzanie do cząsteczki fotoreaktywnych analogów zasad.
Spektroskopia fluorescencyjna (FRET)	Opiera się na efekcie sprzężenia dwóch fluoroforów zdolnych do absorpcji fali elektromagnetycznej o określonej długości (np. fluoresceiny i tetrametylorodaminy). Pozwala na wykrywanie

	grup będących w dystansie 10 – 100 Å.
Denaturacja termiczna	Umożliwia określenie stabilności struktury drugo i trzeciorzędowej. Daje informacje o hierarchii i dynamice zwijania się cząsteczki.

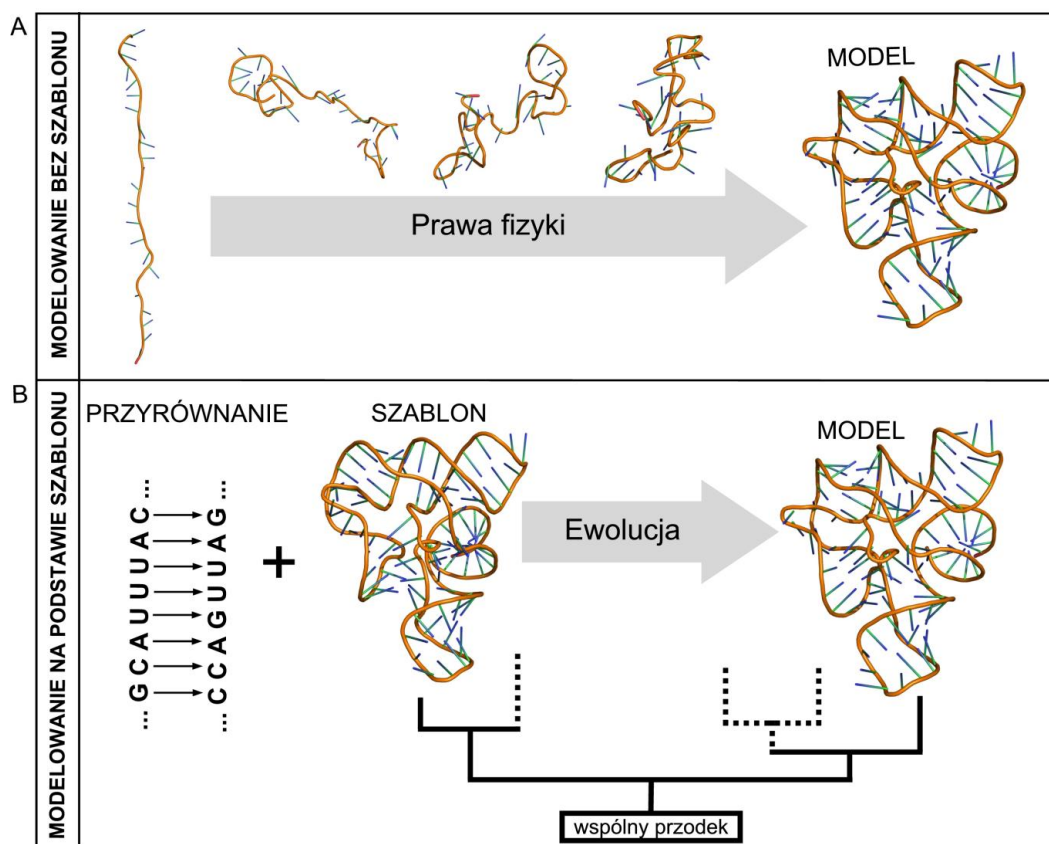
Najdokładniejszymi metodami poznania struktury trzeciorzędowej RNA są krystalografia rentgenowska i NMR (Ke and Doudna, 2004; Thomas and Akoulitchev, 2006; Zidek, i wsp., 2001). Są to metody ważne dla niniejszej pracy, dostarczają bowiem wysokiej jakości szablonów strukturalnych. Obie pozwalają wyznaczyć położenie atomów w cząsteczce z dokładnością nawet poniżej 1 Å. Uzyskane w ten sposób dane są deponowane w bazie PDB (Berman, i wsp., 2000), stworzonej pierwotnie dla struktur białkowych, oraz NDB (*Nucleic Acid Database*) (Berman, i wsp., 2002) dedykowanej kwasom nukleinowym. Pierwszą strukturą rozwiązaną na tym poziomie dokładności był drożdżowy tRNA fenyloalaninowy, opublikowany w 1974 roku (Kim, i wsp., 1974; Robertus, i wsp., 1974). NMR pozwala na rozwiązywanie struktur o małych i średnich rozmiarach w rozpuszczalniku. Krystalografia rentgenowska może być stosowana zarówno dla małych jak i dużych struktur. V. Ramakrishnan wraz ze współpracownikami wykorzystał między innymi tę technikę do badań rybosomu, za które otrzymał nagrodę Nobla w dziedzinie chemii w 2009 roku. Jednym z ograniczeń krystalografii rentgenowskiej, poza dużymi nakładami finansowymi i czasowymi, jest problem z uzyskaniem odpowiedniej jakości kryształów. Stabilność cząsteczki, ujemny ładunek i wydłużony kształt struktury sprawiają, że krystalizacja RNA jest o wiele trudniejsze niż białek (Ke and Doudna, 2004; Thomas and Akoulitchev, 2006).

Ograniczenia metod doświadczalnych oraz duża liczba znanych sekwencji spowodowały, że rozwijane są metody teoretycznego przewidywania struktury przestrzennej RNA.

1.3.2. Metody teoretyczne przewidywania struktury RNA

Duże koszty i nakłady czasu potrzebne na doświadczalne rozwiązanie struktury przestrzennej spowodowały rozwój metod teoretycznych umożliwiających jej poznanie drogą przewidywania komputerowego. Podejście to odniosło duży sukces dla białek - uzyskiwane są modele o jakości pozwalającej na praktyczne zastosowanie (Rother, i wsp., 2011). Obecnie opracowywane są podobne procedury dla RNA. Ogólnie, strategie przewidywania struktury przestrzennej, zarówno białek jak i RNA, podzielić można na dwie grupy. Pierwsza opiera się na prawach fizyki kierujących zwijaniem cząsteczki, druga opiera się na prawidłowościach

rzządzających procesem ewolucji oraz wiedzy zgromadzonej uprzednio w bazach danych (Rycina 8).



Rycina 8. Dwa główne podejścia do modelowania komputerowego struktury trzeciorzędowej RNA: A) modelowanie bez użycia szablonu; B) modelowanie z wykorzystaniem struktury szablonu. Na podstawie (Rother, i wsp., 2011).

Podejście korzystające z praw fizyki, nazywane również podejściem *ab initio*, opiera się na termodynamicznej hipotezie Anfinsena, według której natywna struktura cząsteczki odpowiada globalnemu minimum energii swobodnej układu zawierającego tę cząsteczkę (Anfinsen, 1973). Anfinsen zdefiniował ją dla białek, jednak ze względu na powszechność praw fizyki jest ona prawdziwa również dla innych cząsteczek. Metody z tej kategorii modelują proces zwijania się cząsteczki poprzez symulowanie zmian konformacji podczas szukania przez cząsteczkę stanu minimalnej energii swobodnej (Rother, i wsp., 2011). Poważnymi problemami tego podejścia są ogromna liczba stopni swobody cząsteczki ($N_{\text{atomów}} * 3-5$ (Rother, i wsp., 2012)) oraz złożoność funkcji obliczającej energię układu. Ponadto, podczas zwijania się, cząsteczka napotyka na liczne pułapki energetyczne (lokalne minima energetyczne), które mogą uniemożliwić osiągnięcie stanu natywnego. W konsekwencji podejście *ab initio* jest bardzo kosztowne obliczeniowo (wymaga dużej mocy obliczeniowej) i jest stosowane do przewidywania struktury małych cząsteczek.

Do programów stosujących to podejście wprowadza się także uproszczenia pozwalające zmniejszyć złożoność obliczeniową, np. zmniejsza się ilość stopni swobody poprzez używanie pseudoatomów zamiast reprezentacji pełnoatomowej cząsteczki. Przykładowo programy YUP (Chen, i wsp., 2006) i NAST (Jonikas, i wsp., 2009) używają jednego pseudoatomu do reprezentowania całej reszty nukleotydowej.

Druga grupa metod opiera się na obserwacji ewolucji. Cząsteczki pochodzące od wspólnego przodka (cząsteczki homologiczne) zachowują zazwyczaj ten sam zwój (ułożenie elementów drugorzędowych w przestrzeni) pomimo nagromadzenia mutacji w sekwencji. Co więcej, rozbieżności w strukturze przestrzennej spokrewnionych cząsteczek pojawiają się o wiele wolniej niż różnice w sekwencji. Prawidłowości te zostały zdefiniowane dla białek, jednak są również prawdziwe dla RNA (Rother, i wsp., 2011; Rother, i wsp., 2012). Rozwinięto metody, które pozwalają na przyrównanie sekwencji cząsteczki o nieznannej strukturze (celu) do znanej spokrewnionej struktury (szablonu), wymodelowanie ogólnego zwoju cząsteczki celu opierając się na szablonie i wprowadzenie do struktury modelu zmian wynikających z substytucji, insercji i delecji. Proces transformacji współrzędnych atomowych szablonu w model cząsteczki celu nazywany jest modelowaniem homologicznym lub też modelowaniem porównawczym. W przypadku modelowania porównawczego cząsteczka celu i szablonu są strukturalnie podobne, natomiast nie muszą być ze sobą spokrewnione. Do tej grupy programów należy program RNABuilder (Flores, i wsp., 2010). Modelowanie homologiczne umożliwiają również program RNA123 (SantaLucia, 2002) i platforma PARADISE (Jossinet, i wsp., 2010). Istnieje także grupa programów dająca możliwość manipulacji strukturą poprzez interfejs graficzny, np. program RNA2D3D (Martinez, i wsp., 2008) czy program ERNA-3D (Mueller, 1998) (Tabela 3).

Istnieją również metody hybrydowe, które łączą ze sobą wykorzystanie praw fizyki z wiedzą o strukturach rozwiązanych doświadczalnie. Takie podejście jest określane mianem *de novo*. Wiedza o rozwiązanych strukturach może służyć do opracowania funkcji oceniającej model, gdzie zamiast stosować kryterium wyłącznie parametrów fizycznych oceniane jest podobieństwo modelu do rzeczywistych cząsteczek występujących w naturze. Znane struktury mogą również służyć jako źródło fragmentów do budowy modelu, jak w przypadku programu FARNA (Das and Baker, 2007) i MC-Fold/MC-Sym (Parisien and Major, 2008). Podejście *de novo* jest również używane przez programy 3D-RNAPredict+CYANA, DMD/iFoldRNA (Ding, i wsp., 2008; Sharma, i wsp., 2008) oraz PARADISE (Jossinet, i wsp., 2010) i RNA123 (SantaLucia, 2002) (Tabela 3).

Tabela 3. Charakterystyka programów do modelowania struktury przestrzennej RNA.

Program	Opis programu
3D-RNAPredict + CYANA	Wysokoprzepustowa metoda do przewidywania struktury w niskiej rozdzielczości. 3D-RNAPredict wykorzystuje strukturę drugorzędową, motywy trzeciorzędowe, dane doświadczalne i inne dostępne dane do generowania danych wejściowych dla programu CYANA (algorytm dynamiki kątów torsyjnych), który przewiduje strukturę 3D cząsteczki (Popena, i wsp., 2006).
DMD/iFoldRNA	Platforma sieciowa do zwijania się RNA. Wykorzystuje dynamikę molekularną do przeprowadzenia dyskretnej symulacji RNA używając uproszczonej reprezentacji cząsteczki (jeden nukleotyd reprezentowany jest przez dwa punkty) (Ding, i wsp., 2008; Sharma, i wsp., 2008).
ERNA-3D	Komercyjne oprogramowanie pozwalające na interaktywne modelowanie RNA i białek na podstawie sekwencji i struktury drugorzędowej. Posiada interfejs graficzny, który umożliwia manipulowanie cząsteczką na poziomie drugo i trzeciorzędowym w reprezentacji pełnoatomowej (Mueller, 1998).
FARNA	Program wykorzystujący podejście <i>de novo</i> do modelowania, pozwala uzyskać model przestrzeny cząsteczki RNA o niewielkich rozmiarach (do około 40 nukleotydów) z sekwencji. Szuka struktury o najniższej energii przeprowadzając symulację Monte Carlo, rozpoczynając od struktury liniowej o idealnych wartościach kątów i wiązań. Podczas każdego kroku symulacji losowo wybrany fragment o długości trzech nukleotydów zastępowany jest przez fragment z biblioteki (Das and Baker, 2007).
MC-Fold/MC-Sym	Procedura MC-Fold/MC-Sym pobiera jako dane wejściowe sekwencję. W pierwszym kroku przewiduje strukturę drugorzędową rozszerzoną o oddziaływania niekanoniczne (część MC-Fold) i na jej podstawie w drugim kroku modeluje strukturę trzeciorzędową (część MC-Sym) korzystając z podejścia <i>de novo</i> . Do konstrukcji modelu używa cyklicznych fragmentów nukleotydowych uzyskanych ze struktur rozwiązanych doświadczalnie (Parisien and Major, 2008).
MMB (RNABuilder)	Oprogramowanie wykorzystujące podejście modelowania homologicznego. Pozwala na strukturalne przyrównanie sekwencji celu do trójwymiarowej struktury szablonu, przy zachowaniu kontroli użytkownika nad całym procesem. Umożliwia zastosowanie kilku szablonów do budowy jednego modelu, oraz modeluje regiony bez szablonu. Proces modelowania odbywa się na różnych poziomach. W celu uniknięcia pułapek energetycznych wszystkie atomy łańcucha głównego traktowane są jako niezależne, zasady zaś jako bryły sztywne (Flores, i wsp., 2010).
NAST	Program umożliwia przeprowadzanie symulacji dynamiki molekularnej kwasów nukleinowych na podstawie przewidywanej

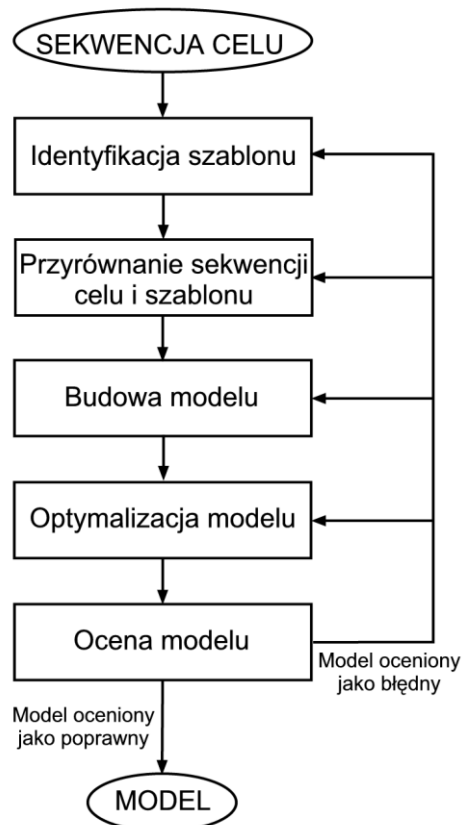
	struktury drugorzędowej i oddziaływań trzeciorzędowych. Używa potencjału empirycznego oraz uproszczonej reprezentacji struktury, w której jedna zasada reprezentowana jest przez jeden pseudoautom. Pozwala na zastosowanie danych doświadczalnych (np. z doświadczenia SAXS) do filtrowania uzyskanych struktur (Jonikas, i wsp., 2009).
PARADISE (Assemble)	Oprogramowanie pozwalające na analizowanie, manipulowanie i tworzenie modeli złożonych struktur trójwymiarowych RNA. Modelowanie może być wykonane zarówno przy użyciu podejścia <i>de novo</i> , jak i modelowania homologicznego. Uwzględnione mogą zostać również mapy gęstości elektronowej. Posiada interfejs graficzny, który umożliwia m.in. redagowanie struktury drugorzędowej i przeszukiwanie biblioteki adnotowanych struktur trzeciorzędowych (Jossinet, i wsp., 2010).
RNA123	Komercyjna platforma integrująca narzędzia do modelowania struktury przestrzennej zarówno z wykorzystaniem podejścia <i>de novo</i> jak i modelowania homologicznego. Umożliwia analizę struktury, jej wizualizację i tworzenie przyrównania sekwencji. Wykrywa oddziaływania wodorowe pomiędzy zasadami i klasyfikuje je. Posiada wbudowane pole siłowe zoptymalizowane dla RNA (SantaLucia, 2002).
RNA2D3D	Procedura pozwalająca uzyskać automatycznie wstępne, przybliżone modele struktury trójwymiarowej RNA. Wbudowany interfejs graficzny umożliwia ich interaktywne redagowanie, m.in. szybkie przeglądanie alternatywnych konformacji 3D, dodawanie i usuwanie par zasad, oraz manipulację oddziaływaniami warstwowymi. Ostatecznie uzyskane modele mogą zostać poddane optymalizacji energii (Martinez, i wsp., 2008).
YUP	Pakiet języka Python z zaimplementowanymi różnymi rodzajami reprezentacji cząsteczki. Dla RNA dostępna jest uproszczona reprezentacja, gdzie jeden nukleotyd to jeden pseudoautom (Chen, i wsp., 2006).

Modelowanie bazujące na szablonie, które stanowi podstawę teoretyczną niniejszej pracy ma długą tradycję w przewidywaniu struktury białek. To dla białek opracowane zostały dwa główne sposoby transformacji szablonu w model. Jeden z nich to wykorzystanie kątów i odległości między atomami z przyrównanych do siebie regionów celu i szablonu jako więzy w czasie budowy modelu. Został on zaimplementowany w programie MODELLER (Sali and Blundell, 1993) do przewidywania struktury białka, a także przeniesiony do rzeczywistości RNA i użyty w programie RNABuilder (Flores, i wsp., 2010). Drugi sposób to bezpośrednie kopiowanie współrzędnych szablonu, zarówno atomów łańcucha głównego, jak i grup bocznych, w przyrównanych do siebie regionach. Miejsca insercji w sekwencji celu i delekcji

w szablonie są następnie uzupełniane fragmentami z bazy danych znanych z innych struktur pętli. Taki schemat zaimplementowany został w programie SWISS-MODEL (Schwede, i wsp., 2003) przewidującym strukturę przestrzenną białka.

1.4. Opis problemu

Dotychczas nie powstał program do przewidywania struktury RNA, który wykorzystuje modelowanie homologiczne i przeprowadza transformację struktury szablonu w model poprzez bezpośrednie kopiowanie współrzędnych, w sposób zaimplementowany w programie SWISS-MODEL (Schwede, i wsp., 2003) do przewidywania struktury białek. Brakuje również oprogramowania wspierającego kroki niezbędne podczas budowy modelu homologicznego (Rycina 9), np. ocenę modelu i redagowanie jego niesatysfakcjonujących fragmentów czy szukanie i przygotowanie szablonu. W szczególności zaś nie został do tej pory opracowany program, który umożliwi modelowanie modyfikowanych nukleotydów oraz użycie różnych scenariuszy podczas modelowania homologicznego (np. przygotowanie dużego zbioru modeli, modelowanie z wykorzystaniem wielu szablonów, składanie modelu z predefiniowanych fragmentów).



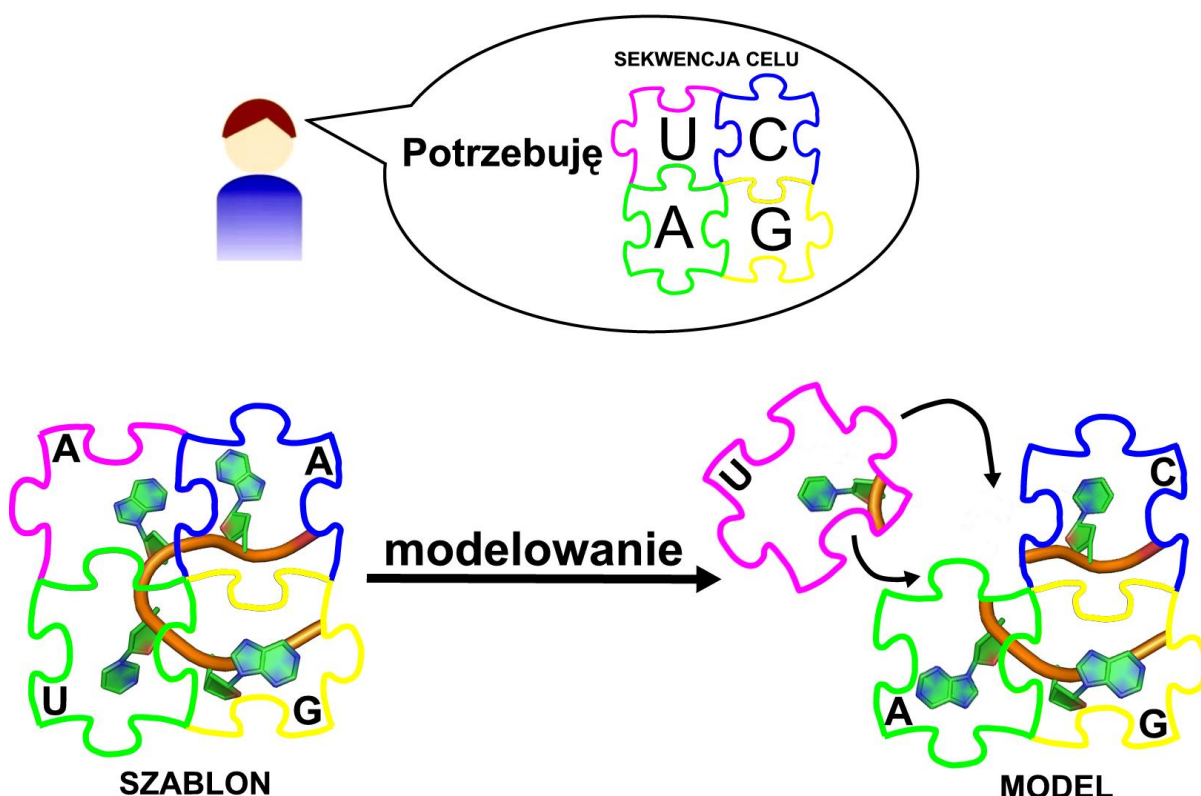
Rycina 9. Kroki niezbędne do zbudowania modelu homologicznego. Schemat przygotowany na podstawie: (Kosinski, i wsp., 2008).

Cel pracy

Celem pracy jest **implementacja programu do modelowania struktury przestrzennej RNA** w oparciu o podejście modelowania homologicznego oraz metodologię opracowaną wcześniej dla białek (Rycina 10).

Program ma umożliwić homologiczne oraz porównawcze budowanie modeli z wykorzystaniem różnych strategii (modelowanie automatyczne, modelowanie w oparciu o kilka szablonów, redagowanie wybranych części struktury). Ma dać możliwość modelowania modyfikowanych nukleotydów oraz wspierać wszystkie kroki niezbędne do budowy modelu homologicznego.

Ponadto celem pracy jest również **przetestowanie uzyskanego programu** na zbiorze tRNA posiadających strukturę rozwiązana doświadczalnie i zdeponowaną w bazie PDB.



Rycina 10. Głównym celem pracy jest program, który pozwoli na budowanie trójwymiarowych modeli RNA na podstawie struktury szablonu.

2. Materiały

Model homologiczny powstaje na podstawie szablonu strukturalnego oraz przyrównania sekwencji, zawierającego informację, które reszty nukleotydów odpowiadają sobie pomiędzy przewidywaną cząsteczką i szablonem:

$$\text{szablon} + \text{przyrównanie} = \text{model}$$

Do realizacji niniejszej pracy wykorzystane zostały pliki z danymi o rozwiązanej doświadczalnie strukturze przestrzennej RNA (rozdział 2.1) oraz danymi o powiązaniach ewolucyjnych między cząsteczkami tRNA na poziomie sekwencji (rozdział 2.2).

2.1. Pliki ze współrzędnymi położenia atomów struktur RNA

W celu implementacji oraz testowania programu do przewidywania struktury przestrzennej RNA używane były pliki ze współrzędnymi położenia atomów w strukturach RNA. Przede wszystkim były to pliki w formacie PDB pobrane z bazy *Protein Data Bank* (PDB) (Berman, i wsp., 2000) oraz *Nucleic Acid Database* (NDB) (Berman, i wsp., 2002) (rozdziały 2.1.1, 0). Użyte zostały także pliki w formacie MOL pochodzące z bazy MODOMICS (Czerwonec, i wsp., 2009) (rozdział 2.1.3). Identyfikatory wszystkich użytych plików PDB są dostępne w Załączniku (rozdział 6.2).

2.1.1. Zbiór struktur wykorzystany do ewaluacji modelowania

Testowanie możliwości programu wykonywanego w ramach niniejszej pracy zostało przeprowadzone na zbiorze 99 struktur tRNA pobranych w postaci plików PDB z baz *Protein Data Bank* (PDB) (Berman, i wsp., 2000) i *Nucleic Acid Database* (NDB) (Berman, i wsp., 2002). Rodzina tRNA została wybrana ze względu na występowanie w niej licznych i różnorodnych modyfikacji (zarówno prostych metylacji jak i bardziej złożonych grup funkcyjnych, np. wybutozyny). Ponadto, pomimo silnej konserwacji struktury o kształcie litery L, należące do tej rodziny cząsteczki wykazują dużą zmienność lokalną. W celu zapewnienia różnorodności zbioru wybrane zostały struktury pochodzące z różnych organizmów (np. *Escherichia coli*, *Saccharomyces cerevisiae*), kodujące różne aminokwasy (np. treoninę, asparaginę, fenyloalaninę) i reprezentujące różne stany biologiczne (np. oddziałujące z rybosomem, oddziałujące z syntetazą aminoacylo-tRNA, w stanie wolnym). Identyfikatory wszystkich 99 struktur dostępne są w Załączniku (rozdział 6.2.).

2.1.2. Zbiór struktur wykorzystany do utworzenia biblioteki fragmentów

Dla umożliwienia modelowania rejonów bez szablonu utworzona została biblioteka fragmentów. Do jej wygenerowania posłużył zbiór struktur rnaDB2005 oryginalnie użyty do wyliczenia możliwych konformacji łańcucha głównego RNA (Murray and Carr, 2008), a przez to zawierający szeroką gamę konformacji. Został on utworzony ze struktur o wysokiej rozdzielczości (do 1.8 Å) pobranych z bazy NDB. Zbiór został również oczyszczony ze struktur powtarzających się. Jest on dostępny na:

<http://kinemage.biochem.duke.edu/downloads/datasets/rnaDB2005/>.

Wykaz identyfikatorów PDB wymieniony jest w Załączniku (rozdział 6.2).

Skrypt do tworzenia biblioteki fragmentów na podstawie zbioru jest elementem oprogramowania opisanego w niniejszej pracy (`moderna/LIRdb.py`) i może zostać wykorzystany do wyliczenia biblioteki fragmentów z dowolnego zbioru struktur.

2.1.3. Zbiór struktur pojedynczych modyfikowanych nukleotydów

W celu modelowania modyfikowanych nukleotydów stworzona została biblioteka grup funkcyjnych biorących udział w powstawaniu poszczególnych modyfikacji. Pojedyncze modyfikowane nukleotydy zostały pobrane z bazy MODOMICS (Czerwoniec, i wsp., 2009). Następnie z każdego pliku wyekstrahowane zostały pojedyncze grupy atomów miejsc modyfikowanych i zapisane jako pliki PDB. Po usunięciu plików powtarzających się zostały one włączone do oprogramowania omawianego w niniejszej pracy. Ponieważ dane uzyskane z bazy MODOMICS zapisane były w formacie MOL, zmienione zostały nazwy wszystkich atomów tak, aby ze skrótów jednoliterowych (C, O, P, S, itd.) uzyskać nazwy typowe dla formatu PDB (C1, C2, OP1 itd.). Zbiór uzyskanych fragmentów dostępny jest w źródle programu implementowanego w ramach niniejszej pracy (`moderna/data/modification_fragments`).

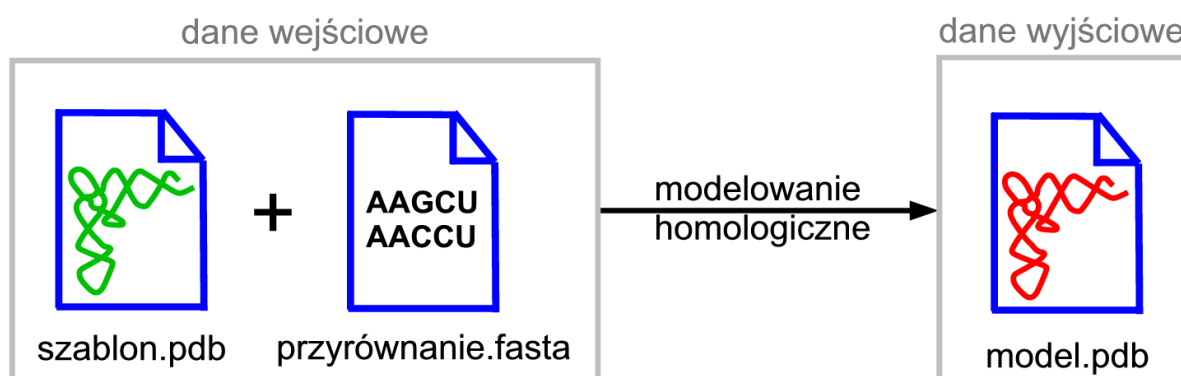
2.2. Przyrównania sekwencji tRNA

Przyrównania sekwencji dla 99 cząsteczek tRNA (rozdział 2.1.1) potrzebne do testowania programu rozwijanego w ramach niniejszej pracy uzyskane zostały z bazy Rfam (Gardner, i wsp., 2009). Pobrane zostało pełne przyrównanie wielu sekwencji dla całej rodziny tRNA (1101833 sekwencji), a następnie niezbędne sekwencje zostały wyekstrahowane. W tym celu sekwencje odczytane ze struktur tRNA, z których usunięto modyfikacje

(przyrównanie uzyskane z bazy Rfam nie uwzględniało modyfikowanych nukleotydów) wykorzystane zostały do przeszukiwania pliku pobranego z bazy Rfam. W ostatecznie uzyskanym przyrównaniu 99 sekwencji tRNA modyfikacje w postaci jednoliterowych skrótów zostały przywrócone w odpowiednich pozycjach. Całe przyrównanie w formacie FASTA dostępne jest w Załączniku (rozdział 6.3).

3. Metody

Program rozwijany w ramach niniejszej pracy pobiera, jako dane wejściowe, przygotowane przez użytkownika pliki PDB oraz FASTA i na ich podstawie generuje plik PDB ze współzrędnymi atomów przewidywanej struktury przestrzennej cząsteczki RNA (Rycina 11). Proces modelowania opiera się na metodzie modelowania homologicznego/porównawczego (rozdział 3.1). Implementacja wykonana została w języku programowania Python, wykorzystane zostały biblioteki dostępne dla tego języka (rozdział 3.2). Program został przetestowany na zbiorze tRNA (rozdział 2.1.1). Dla porównania podobieństwa struktur modeli ze strukturami natywnymi zastosowane zostały cztery miary (rozdział 3.3). Podczas implementacji i testowania programu używane były również programy zewnętrzne (rozdział 3.4).



Rycina 11. Język programowania Python i podejście modelowania homologicznego zostały wykorzystane do napisania programu, który buduje modele przestrzenne RNA na podstawie szablonu i przyrównania.

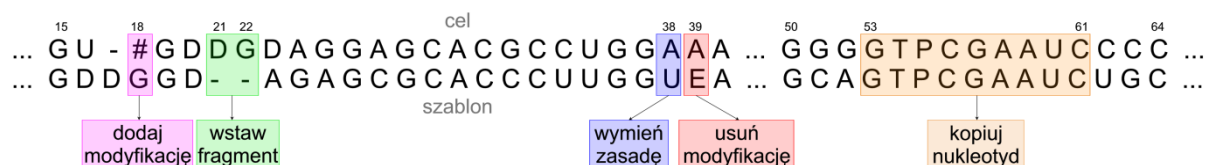
3.1. Modelowanie homologiczne

Analiza porównawcza spokrewnionych (pochodzących od wspólnego przodka) cząsteczek RNA (Dror, i wsp., 2005) wykazała, że struktura drugo- i trzeciorzędowa jest zwykle silniej konserwowana niż sekwencja. Ponadto regiony ważne dla stabilności i funkcji są zwykle konserwowane na wszystkich poziomach. Przyjmuje się, że w homologicznych rodzinach RNA, zwój pozostaje niezmienny, co umożliwia modelowanie oparte na szablonie. Do rozpoznania takiej strukturalnej konserwacji mogą zostać wykorzystane sekwencja oraz struktura drugorzędowa.

Podobna zależność została wcześniej odkryta i wykorzystana dla białek (Chothia and Lesk, 1986). Po doświadczalnym rozwiązaniu niewielkiej ilości struktur białkowych stało się bowiem jasne, że ewolucyjnie spokrewnione cząsteczki zwykle zachowują ułożenie i wzajemne położenie elementów struktury drugorzędowej, pomimo nagromadzenia mutacji w sekwencji. Odkryta została również prawidłowość, że dywergencja struktury zachodzi znacznie wolniej niż dywergencja sekwencji, choć cechy te są ze sobą połączone (Chothia and Lesk, 1986).

Analizy te zainspirowały rozwój metod do przyrównywania sekwencji cząsteczki, dla której szukana jest struktura (**celu**) do struktury innej spokrewnionej cząsteczki posiadającej strukturę rozwiązana doświadczalnie (**szablonu**), modelowania ogólnego zwoju celu opierając się na szablonie i przewidywania jak struktura szablonu zmieni się pod wpływem substytucji, insercji i delecji (Cohen-Gonsaud, i wsp., 2004; Krieger, i wsp., 2003). Dla białek proces identyfikacji spokrewnionego strukturalnie szablonu nazwany został „rozpoznawaniem zwoju”, natomiast przekształcenie współrzędnych atomów struktury szablonu w model celu jest zwykle określane jako „modelowanie homologiczne” lub „modelowanie porównawcze”. W drugim przypadku pod uwagę brane są szablony, które nie są homologiczne do cząsteczki celu, a jedynie strukturalnie podobne.

Oprogramowanie rozwijane w ramach niniejszej pracy buduje model cząsteczki RNA celu, opierając się na dostarczonej przez użytkownika strukturze szablonu i przyrównaniu sekwencji celu i szablonu. Program podczas etapu budowy modelu (Rycina 9) iteruje przez kolejne pozycje w przyrównaniu i dla każdej z nich przyporządkowuje jedną lub więcej operacji (Rycina 12). Następnie wszystkie operacje są wykonywane. Identyczne współrzędne atomowe nukleotydów są bezpośrednio kopiowane z szablonu (Rycina 12 - kolor żółty), w przypadku substytucji zasada jest wymieniana (Rycina 12 - kolor niebieski), następnie modelowane są potranskrypcyjne modyfikacje (Rycina 12 - kolor czerwony i różowy). Regiony bez szablonu modelowane są przez wybieranie z biblioteki fragmentów elementu strukturalnego pasującego pomiędzy reszty okalające ten region (reszty kotwiczące), bazując głównie na geometrii reszt kotwiczących (Rycina 12 - kolor zielony). Podobnie działa program do modelowania homologicznego białek SWISS-MODEL (Schwede, i wsp., 2003), który był inspiracją dla niniejszej metody. Istnieją również programy modelujące w podobny sposób pętle. Baza LIP (*Loops In Proteins*) i serwer SuperLooper (Michalsky, i wsp., 2003) pomogły w opracowaniu funkcji oceniającej dopasowanie fragmentów do reszty modelu.



Rycina 12. Przystawienie sekwencji celu (górną) i szablonu (dolną) z zaznaczonymi głównymi typami operacji wykonywanymi podczas modelowania homologicznego.

Podejście modelowania homologicznego ma dwa główne ograniczenia. Po pierwsze modelowanie struktury celu rozpoczyna się od innej znanej struktury cząsteczki podobnej, używanej jako szablon. Oznacza to, że jeśli taka cząsteczka nie istnieje lub nie może być zidentyfikowana modelowanie nie może zostać przeprowadzone. Po drugie każdy element sekwencji cząsteczki celu musi zostać przystawiony do strukturalnie odpowiadającego elementu w sekwencji/strukturze szablonu. Również program rozwijany w ramach niniejszej pracy wymaga od użytkownika dostarczenia struktury szablonu oraz przystawienia sekwencji celu i szablonu.

3.2. Narzędzia programistyczne

Program rozwijany w ramach niniejszej pracy został zaimplementowany w języku Python (rozdział 3.2.1). Wykorzystano jego biblioteki powstałe dla rozwiązywania typowych problemów programistycznych (pozwalające np. na czytania i zapisywanie plików czy komunikację z systemem operacyjnym) (rozdział 3.2.2). Użyte zostały również biblioteki dedykowane problemom biologicznym. Pakiet PDB z biblioteki Biopython dostarczył obiektów do reprezentowania struktur przestrzennych cząsteczek RNA oraz metody do ich przestrzennego nakładania. Pakiet BaseRecognizer (rozdział 3.2.4) pozwolił na precyzyjne rozpoznawanie poszczególnych modyfikowanych nukleotydów, opierając się na ich budowie chemicznej.

3.2.1. Język programowania Python

Język programowania Python został stworzony w latach pięćdziesiątych przez Guida van Rossum. Jego nazwa pochodzi od nazwy brytyjskiego serialu „*Latający cyrk Monty Pythona*”. Projekt rozwijany jest pod licencją otwartego oprogramowania, a implementacja oraz dokumentacja dostępne są na stronie organizacji *Python Software Foundation* (<http://www.python.org/>).

Python należy do grupy języków interpretowanych, co oznacza, że jego kod jest na bieżąco tłumaczony i wykonywany przez interpreter, dzięki czemu jest on niezależny od platformy, choć nie tak wydajny jak kod języków kompilowanych (np. C++). Typy w języku Python przypisywane są dynamicznie, zarządzanie pamięcią również odbywa się automatycznie poprzez zbieranie nieużytków (*garbage collection*). Silnie wspierane jest programowanie obiektowe, choć styl programowania nie jest wymuszany i możliwe jest również programowanie strukturalne i funkcjonalne. Dużym udogodnieniem języka Python jest dostępność trybu interaktywnego. Język Python jest językiem wysokiego poziomu, co oznacza, że pod względem składni i semantyki przypomina język ludzki.

Kod w języku Python sterowany jest przez wcięcia (indentacje, standardowo cztery znaki spacji), przez co programy pisane w tym języku są przejrzyste i mają łatwą w czytaniu składnię. Umożliwia to wgląd w kod zewnętrznym użytkownikom. Implementacja programu wykonywanego w ramach niniejszej pracy opiera się na paradygmacie programowania obiektowego. Elementy struktury (atomy, reszty, łańcuchy, struktura) są osobnymi obiektami powiązаныmi ze sobą hierarchicznie. Również sekwencja, przyrównanie dwóch sekwencji, poszczególne pozycje przyrównania oraz inne ważne w modelowaniu homologicznym elementy stanowią osobne obiekty z przypisanymi właściwościami i funkcjami. Zaletą języka Python jest dostęp do wielu standardowych bibliotek oraz bibliotek umożliwiających pracę z danymi biologicznymi (Cock, i wsp., 2009).

Język Python poprzez wspieranie obiektowości i hierarchicznej architektury ułatwił również użycie niektórych technik rozwoju oprogramowania. Stosowane były między innymi karty CRC (*class-responsibility-collaboration cards*), opowieści użytkownika (*user stories*) i testy jednostkowe (*unit tests*) (Rother, i wsp.). Ponadto, wszystkie moduły i funkcje były dokumentowane w postaci dostępnych dla użytkownika krótkich opisów (*docstrings*).

3.2.2. Biblioteki programistyczne

Bogaty zasób bibliotek dostępnych dla języka Python oferuje gotowe rozwiązania typowych problemów pojawiających się podczas rozwijania oprogramowania, np. przygotowanie wersji instalacyjnej programu jest możliwe dzięki bibliotece `distutils`, a wersji wykonywalnej programu dzięki bibliotece `py2exe`. Operacje matematyczne, wykonywane między innymi podczas ewaluacji zbioru modeli opracowanego w ramach niniejszej pracy oraz podczas dopasowywania fragmentów bez szablonu do modelu, były

możliwe dzięki funkcjom biblioteki `math`, a wykorzystanie biblioteki `numpy` pozwoliło na przyspieszenie niektórych obliczeń. Biblioteki `os` i `sys` służą do komunikacji z systemem i były używane np. w celu przeszukiwania zawartości katalogu z szablonami podczas automatycznej analizy zbioru 99 tRNA. Do poszukiwania wzoru w tekście stosowana była biblioteka `re`, np. podczas konwersji plików PDB do nowej bądź starej notacji.

Dane tabularyczne, potrzebne przykładowo podczas tworzenia modyfikowanych nukleotydów (dane na temat każdej modyfikacji – fragmenty z grupami funkcyjnymi, które muszą zostać dodane oraz wykaz atomów użytych do nałożenia przestrzennego, zgromadzone zostały w tabeli), obsługiwane są przy pomocy biblioteki `csv`. Na zaimplementowanie interfejsu dostępnego przez wiersz poleceń pozwoliła biblioteka `optparser`. Poszukiwanie błędów w kodzie zostało ułatwione przez bibliotekę `pdb`, która pozwala przenieść się w dowolne miejsce wykonywanego kodu i na bieżąco sprawdzać rezultat działania poszczególnych linii kodu. Podczas implementacji zostały również użyte biblioteki `random`, `tempfile` i `warnings`.

Wykorzystane biblioteki zostały przedstawione poniżej (Tabela 4). Dwie biblioteki biologiczne, szczególnie ważne dla implementacji programu rozwijanego w ramach niniejszej pracy – `Bio.PDB` i `BaseRecognizer`, opisane zostały bardziej szczegółowo w kolejnych rozdziałach.

Tabela 4. Biblioteki języka Python wykorzystane podczas implementacji programu ModeRNA.

Biblioteka	Opis
<code>csv</code>	Służy do czytania i zapisu danych tabularycznych w formacie CSV (<i>Comma Separated Values</i>). Pozwala na korzystanie z danych arkusza kalkulacyjnego.
<code>distutils</code>	(<i>Distributing Python Core Modules</i>) pozwala na przygotowanie wersji instalacyjnej programu.
<code>math</code>	Zawiera liczne funkcje, operacje i stałe matematyczne. Znajdują się tu np. funkcje trygonometryczne (<code>math.sin</code> , <code>math.cos</code>), funkcja przekształcająca wartości kątów ze stopni na radiany i odwrotnie (<code>math.degrees</code>), funkcja pierwiastkująca (<code>math.sqrt</code>), stała PI (<code>math.pi</code>).
<code>numpy</code>	Umożliwia zaawansowane operacje matematyczne na macierzach. Podstawowy obiekt tej biblioteki to <code>numpy.array</code> (macierz), a przykładowe operacje to dodawanie macierzy (<code>numpy.add</code>), odwracanie macierzy (<code>numpy.cross</code>), transpozycja macierzy

	(<code>numpy.transpose</code>), wypełnianie macierzy zerami (<code>numpy.zeros</code>).
<code>optparse</code>	Wczytuje z wiersza poleceń opcje i argumenty użyte podczas uruchamiania programu.
<code>os</code>	Daje możliwość korzystania z funkcji zależnych od systemu operacyjnego. Przykładowo, pozwala na manipulację ścieżkami (<code>os.path</code>), oraz czytanie danych z katalogu (<code>os.listdir</code>).
<code>pdb</code>	Podczas wykonywania programu przenosi użytkownika z miejsca, w którym się znajduje do interfejsu języka Python, co pozwala na lokalizowanie i usuwanie błędów z kodu.
<code>py2exe</code>	Konwertuje programy języka Python do wersji wykonywalnej (<code>.exe</code>), dzięki czemu mogą one zostać użyte na platformie Windows bez zainstalowanego języka Python.
<code>random</code>	Generuje liczby pseudolosowe (<code>random.random</code>).
<code>re</code>	Służy do wykonywania operacji na wyrażeniach regularnych np. poszukiwania wzorca w tekście (<code>re.search</code>).
<code>sys</code>	Pozwala na dostęp do obiektów i funkcji używanych przez interpreter języka Python.
<code>tempfile</code>	Daje możliwość tworzenia plików tymczasowych.
<code>Warnings</code>	Pozwala na zarządzanie ostrzeżeniami generowanymi przez program.

3.2.3. Pakiet PDB z biblioteki Biopython (Bio.PDB)

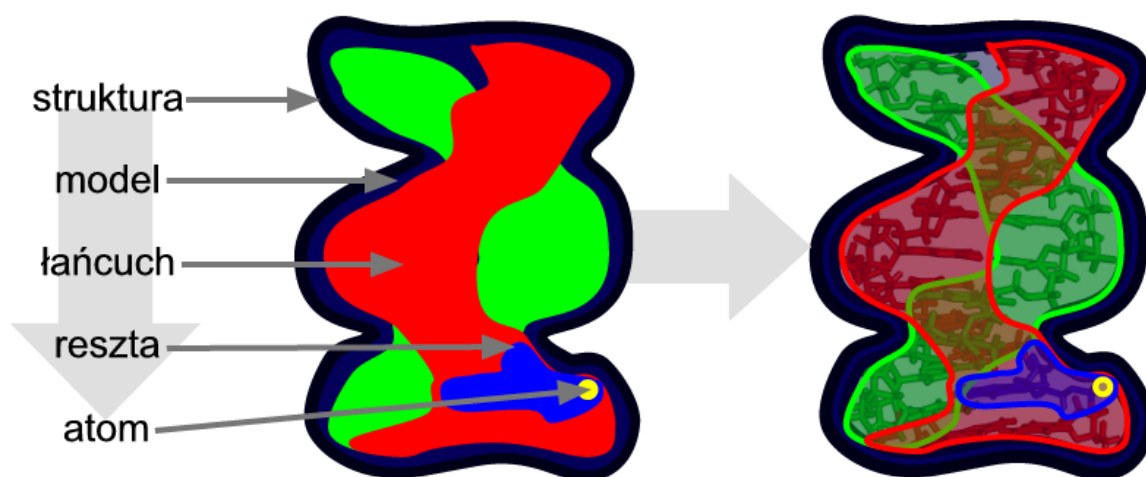
Pakiet `Bio.PDB` (Hamelryck and Manderick, 2003) był podstawą do implementacji modułów związanych z przetwarzaniem danych o strukturze przestrzennej cząsteczki. W szczególności wykorzystano go do:

- Czytania i zapisywania plików PDB.
- Reprezentacji struktury przestrzennej.
- Nakładania przestrzennego struktur.
- Wykrywania zderzeń przestrzennych.

Pliki PDB są wczytywane do programu rozwijanego w ramach niniejszej pracy dzięki parserowi formatu PDB dostępnemu w pakiecie `Bio.PDB` - `Bio.PDB.PDBParser`.

Przekształca on informację zawartą w pliku na obiekty, które są używane w ciągu pracy programu. Parser obsługuje pliki z różnymi rodzajami cząsteczek (m. in. białka, DNA, RNA, ligandy). Trójwymiarowe modele, które są efektem działania programu zapisywane są do pliku PDB również dzięki pakietowi `Bio.PDB - Bio.PDB.PDBIO`.

Struktura cząsteczki jest w pakiecie `Bio.PDB` reprezentowana przez kilka obiektów o charakterze hierarchicznym (Rycina 13). W programie rozwijanym w ramach niniejszej pracy bezpośrednio wykorzystano obiekt `Bio.PDB.Atom`, który używany jest bez żadnych modyfikacji, oraz obiekt `Bio.PDB.Residue`, który stał się podstawą dla obiektu reprezentującego specyficzną resztę nukleotydową RNA. W programie zachowana została hierarchiczna struktura obiektów reprezentujących strukturę cząsteczki RNA. Obiekty wyższego rzędu zorientowane zostały jednak na proces modelowania porównawczego cząsteczki RNA.



Rycina 13. Hierarchiczna struktura obiektów reprezentujących cząsteczkę w bibliotece Biopython.PDB.

Pakiet `Bio.PDB` udostępnia funkcje do nałożenia przestrzennego dwóch cząsteczek i obliczenia pomiędzy nimi RMSD (miary średniego dystansu pomiędzy atomami), dostępne w klasie `Bio.PDB.Superimposer`. W pakiecie `Bio.PDB` używany jest do tego celu algorytm opisany w 1989 roku przez Goluba i Van Loana (Golub and Van Loan, 1989). Nałożenie przestrzenne i obliczanie wartości RMSD wykorzystywane jest w programie wykonywanym w ramach niniejszej pracy do modelowania modyfikowanych reszt nukleotydowych (grupy funkcyjne nakładane są na niemodyfikowany nukleotyd), elementów cząsteczki celu bez szablonu (dopasowywanie fragmentów do modelu) oraz do ewaluacji modeli zbioru testowego.

Wykrywanie zderzeń przestrzennych w cząsteczce możliwe jest dzięki klasie `Bio.PDB.NeighborSearch`. Pozwala ona na szukanie atomów (lub innych obiektów) znajdujących się w określonej odległości od siebie lub od danego punktu w przestrzeni. Do poszukiwania sąsiadujących obiektów pakiet `Bio.PDB` wykorzystuje strukturę danych drzewa k -wymiarowego (*KD tree*) (Bentley, 1975). W programie rozwijanym w ramach niniejszej pracy zderzenia steryczne są liczone podczas oceny kandydatów do wstawienia w regiony bez szablonu. Obecność zderzeń sterycznych to również jedna z cech pozwalających ocenić wstępną jakość modelu.

3.2.4. Pakiet `BaseRecognizer`

Pakiet `BaseRecognizer` (Rother, i wsp., 2010) pozwala na rozpoznawanie cząsteczek modyfikowanych nukleotydów, a także innych małych cząsteczek. Posiada on zbiór zdefiniowanych topologii w postaci krótkich linearnych napisów w formacie ASCII. Notacja taka przypomina opis związków chemicznych SMILES (*Simplified Molecular-input Line-Entry Specification*) (Weininger, 1988), jednakże uwzględnia topologie nieokreślone i wykluczone. Cząsteczki są rozpoznawane poprzez szukanie izomorficznych grafów (Ullman, 1976) i przypisywanie danego związku chemicznego do odpowiedniego wzoru lokalnej topologii.

Pakiet został wykorzystany do odczytywania sekwencji ze struktury PDB. Ta metoda ustalania sekwencji pozwala na uniknięcie błędów wynikających z nieprawidłowości w plikach PDB oraz niejednoznacznej nomenklatury modyfikowanych nukleotydów. Nazwy rozpoznanych na podstawie geometrii reszt są przez program, rozwijany w ramach niniejszej pracy, udostępniane jako pełna nazwa chemiczna lub skrót. W sekwencji pojawia się skrót jednoliterowy. Wszystkie rodzaje użytej nomenklatury zaczerpnięte zostały z bazy MODOMICS (Czerwoniec, i wsp., 2009).

3.3. Miary wykorzystane do ewaluacji modeli

W celu wiarygodnego porównania modeli uzyskanych przy pomocy programu rozwijanego w ramach niniejszej pracy ze strukturami rozwiązanymi doświadczalnie użyte zostały cztery miary podobieństwa między strukturami. Najbardziej typową miarą jest średnia odległość pomiędzy atomami struktur (RMSD, *Root Mean Square Deviation*). Druga z użytych miar to globalny test dystansów (GDT-TS, *Global Distance Test – Total Score*) (rozdział 3.3.2), została ona opracowana dla białek w celu zmniejszenia wpływu

lokalnej zmienności na wynik porównania. Pozostałe dwie miary, wskaźnik zniekształcenia (DI, *Deformation Index*) (rozdział 3.3.3) i profil zniekształcenia (DP, *Deformation Profile*) (rozdział 3.3.4), zostały opracowane dla porównywania cząsteczek RNA i biorą pod uwagę oddziaływanie występujące w tych cząsteczkach.

3.3.1. RMSD

Najczęściej stosowaną miarą podobieństwa dwóch struktur jest średnia odległość pomiędzy ich odpowiadającymi sobie atomami (RMSD, *Root Mean Square Deviation*). Jest ona liczona dla optymalnie nałożonych przestrzennie struktur według wzoru:

$$RMSD(X, Y) = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}},$$

gdzie X i Y to porównywane struktury, $(x_i - y_i)$ to dystans pomiędzy kolejnymi, odpowiadającymi sobie atomami w tych strukturach, a n to liczba atomów. Obliczeń można dokonywać używając wszystkich atomów (RMSD pełnoatomowe) lub tylko na wybranych atomach, np. P i C4', co umożliwia uzyskanie RMSD dwóch różnych struktur (nie tylko modelu i struktury referencyjnej). Wadą tej miary jest to, że lokalna różnica w strukturze może spowodować wysoką wartość RMSD, sugerując, że porównywalne struktury bardzo się od siebie różnią.

W niniejszej pracy miara RMSD była wykorzystywana do ewaluacji zbioru modeli tRNA. Wartość RMSD została obliczona dla każdej pary model – struktura natywna (RMSD pełnoatomowe oraz atomów P i C4') oraz dla par szablon – struktura natywna (RMSD atomów P i C4').

3.3.2. GDT-TS

Miarą popularnie używaną do ewaluacji modeli białkowych jest globalny test dystansów (GDT-TS, *Global Distance Test – Total Score*) (Cozzetto, i wsp., 2007). Obliczenia są dokonywane dla optymalnie nałożonych struktur. Jest to suma par atomów znajdujących się w odległości 1,0 Å, 2,0 Å, 4,0 Å i 8,0 Å, podzielona przez czterokrotną ilość atomów. Wynik ma wartość od 0 do 1, a wyższy wynik oznacza większe podobieństwo struktur. Dla białek do obliczeń używane są pary atomów Ca.

W niniejszej pracy miara GDT-TS była wykorzystywana do ewaluacji zbioru modeli tRNA. Do obliczeń użyto par atomów P oraz pary atomów C4'.

3.3.3. DI

Wskaźnik zniekształcenia (DI, *Deformation Index*) (Parisien, i wsp., 2009) jest miarą dedykowaną ewaluacji modeli cząsteczek RNA. Porównuje on podobieństwo sieci oddziaływań zasad i jest definiowany, jako RMSD dwóch optymalnie nałożonych struktur, podzielone przez dokładność sieci oddziaływań (INF, *the Interaction Network Fidelity*):

$$DI(M, R) = \frac{RMSD(M, R)}{INF(M, R)},$$

gdzie M to model, a R to struktura referencyjna. Dokładność sieci oddziaływań (INF) uwzględnia oddziaływania wodorowe oraz warstwowe pomiędzy zasadami. Jest ona równa współczynnikowi korelacji Matthews'a (MCC, *Matthews correlation coefficient*):

$$INF(M, R) = MCC(M, R) = \sqrt{PPV * STY},$$

gdzie PPV i STY to odpowiednio specyficzność i czułość, przedstawione wzorami:

$$PPV(\text{specyficzność}) = \frac{|TP|}{|TP| + |FP|},$$

$$STY(\text{czułość}) = \frac{|TP|}{|TP| + |FN|}.$$

TP to oddziaływania przewidziane poprawnie (występujące w obu porównywanych strukturach – modelu M i strukturze referencyjnej R). Oddziaływania przewidziane błędnie (występujące tylko w modelu M) to FP , a oddziaływania, które nie zostały przewidziane (występujące tylko w strukturze referencyjnej R) to FN . Do adnotowania sieci oddziaływań struktury wykorzystane mogą zostać programy MC-Annotate (Gendron, i wsp., 2001; Lemieux and Major, 2002), FR3D (Sarver, i wsp., 2008) lub RNAView (Yang, i wsp., 2003).

W niniejszej pracy miara DI była wykorzystywana do ewaluacji zbioru modeli tRNA. Podczas obliczeń pod uwagę brane były oddziaływania warstwowe, oddziaływania pomiędzy

zasadami typu Watson-Crick oraz innych typów. Do adnotacji sieci oddziaływań w modelach i strukturach doświadczalnych wykorzystano program RNAView (rozdział 3.4.1).

3.3.4. DP

Profil zniekształcenia (DP, *Deformation Profile*) (Parisien, i wsp., 2009) podobnie jak miara DI jest dedykowany ocenie dokładności modeli RNA. Jest to macierz dystansów reprezentująca średnie dystanse pomiędzy modelem i strukturą referencyjną. Otrzymuje się ją poprzez obliczenie wszystkich jednonukleotydowych nałożeń struktury modelu i struktury referencyjnej, a następnie wyliczeniu dla każdego nałożenia średniego dystansu pomiędzy każdą zasadą w strukturze referencyjnej i odpowiadającą jej zasadą w modelu. Macierz DP pokazuje różnice pomiędzy strukturami na poziomie pojedynczych reszt nukleotydowych i pozwala szybko zidentyfikować źródło rozbieżności pomiędzy modelem i strukturą natywną.

W niniejszej pracy macierz DP została obliczona dla każdej pary model – struktura natywna w zbiorze tRNA. Wyliczona została również średnia wartość z macierzy DP.

3.4. Programy zewnętrzne

Podczas testowania oprogramowania rozwijanego w ramach niniejszej pracy wykorzystane zostały również programy zewnętrzne. Program RNAView (rozdział 3.4.1) posłużył do liczenia niektórych miar podobieństwa podczas ewaluacji modeli, a program PyMOL (rozdział 3.4.2) pozwolił na wizualizację modeli.

3.4.1. RNAView

Program RNAView (Yang, i wsp., 2003) służy do identyfikacji i klasyfikacji oddziaływań pomiędzy zasadami w strukturze RNA. Rozpoznaje on zarówno pary zasad typu Watson-Crick jak i oddziaływania innego typu (poprzez inne krawędzie). Program RNAView był wykorzystywany podczas ewaluacji zbioru modeli tRNA. Pozwolił na obliczenie wartości DI i DP dla poszczególnych par model – struktura rozwiązana doświadczalnie.

3.4.2. PyMOL

Program PyMOL (DeLano, 2002) służy do wizualizacji struktur przestrzennych cząsteczek. W ramach niniejszej pracy był wykorzystywany do indywidualnej („ręcznej”) oceny modeli. Przykładowo, podczas optymalizacji funkcji dopasowującej fragmenty

do modelu mały zbiór par struktur bogatych w insercje/delecje (wybrany ze zbioru 99 tRNA) oceniany był wizualnie na podstawie wiedzy o budowie RNA. Pomogły w tym takie cechy programu PyMOL jak możliwość różnego reprezentowania struktury, swobodna zmiany skali, pomiar dystansów (cechy te są częste w wielu programach do wizualizacji). Bardzo pomocna była obecność interfejsu skryptowego, który ułatwił i przyspieszył pracę. Program PyMOL wykorzystany został również do generowania rysunków zamieszczonych w niniejszej pracy.

4. Wyniki

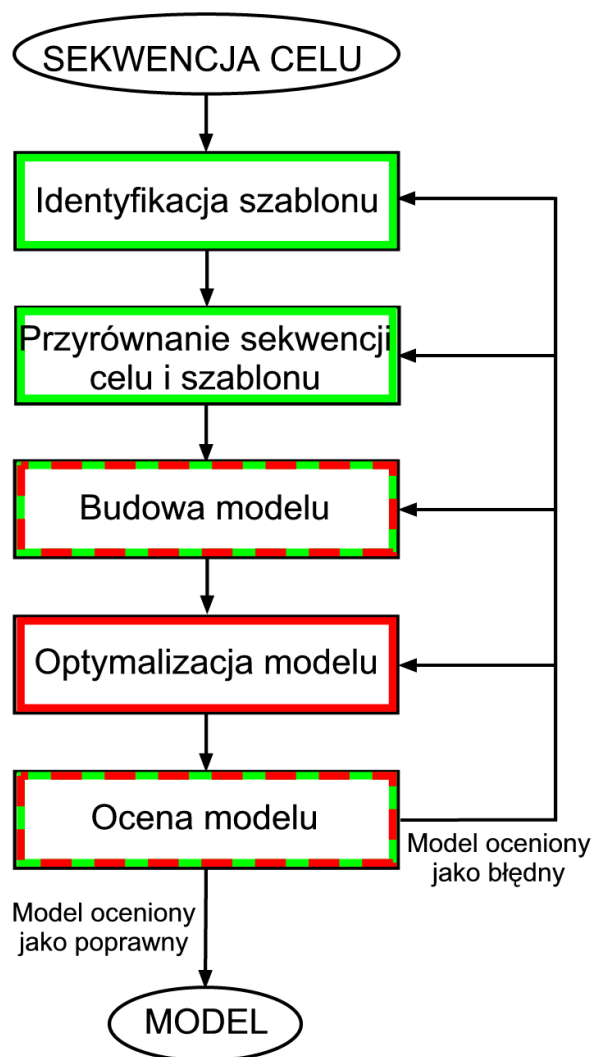
Głównym wynikiem pracy jest program ModeRNA do przewidywania i redagowania struktury przestrzennej RNA. Program ModeRNA (Musielak, i wsp., 2010; Rother, i wsp., 2011) modeluje strukturę trzeciorzędową na podstawie szablonu strukturalnego i przyrównania pomiędzy sekwencją celu i szablonu (rozdział 4.1). Modelowanie oraz analiza struktury są wykonywane dzięki zaimplementowanym w nim operacjom (rozdział 4.2). Program został zaprojektowany dla trzech rodzajów użytkowników i można go używać przez interfejs na stronie internetowej, wiersz poleceń lub interfejs programistyczny (rozdział 4.3). Jego implementacja to ponad czternaście tysięcy linii kodu zgromadzonych w czterdziestu dziewięciu modułach, których poprawność działania zabezpiecza zbiór automatycznych testów (rozdział 4.4).

Możliwości programu ModeRNA zostały sprawdzone poprzez przygotowanie szeregu modeli dla zbioru cząsteczek tRNA posiadających strukturę rozwiązana doświadczalnie, zdeponowaną w bazi PDB (rozdział 4.5). Do oceny jakości modeli użyty został program MARS, służący do porównywania modeli teoretycznych i struktur natywnych (rozdział 4.6).

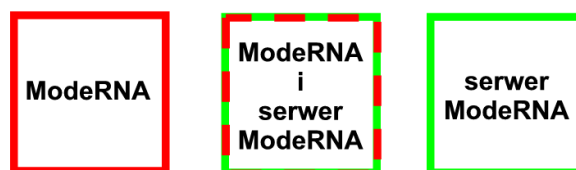
Podsumowując, w wyniku niniejszej pracy powstał program ModeRNA do modelowania homologicznego RNA, modele trójwymiarowe tRNA oraz program pozwalający na porównanie tych modeli ze strukturami rozwiązanymi doświadczalnie.

4.1. Modelu homologicznego z programem ModeRNA

Uzyskanie modelu homologicznego, niezależnie od wykorzystywanego oprogramowania, odbywa się przez wykonanie pięciu kroków (identyfikacji szablonu strukturalnego; przygotowania przyrównania pomiędzy sekwencją celu i szablonu; zbudowania modelu; optymalizacji modelu; oceny modelu). Jego podstawową zasadą, a zarazem warunkiem użycia programu ModeRNA, jest obecność szablonu strukturalnego oraz przyrównania sekwencji celu do sekwencji szablonu. Na ich podstawie generowany jest model przestrzenny, który po procesie optymalizacji zostaje poddany ocenie (Rycina 14).



Możliwości programu:



Rycina 14. Kroki niezbędne do zbudowania modelu homologicznego z wyróżnionymi operacjami wykonywanymi przez program ModeRNA oraz interfejs internetowy programu.

Pierwsze kroki w budowie modelu to **wybór szablonu** oraz **przygotowanie przyrównania** między sekwencjami celu i szablonu. Ich wykonanie wspiera serwer ModeRNA (nie są one zaś dostępne w lokalnej wersji programu). Wyszukiwanie szablonu ograniczone jest do struktur z bazy PDB o rozdzielczości lepszej niż 2,5 Å. Przyrównanie wykonywane jest przez programy i procedury zewnętrzne udostępniane przez serwer ModeRNA (rozdział 4.3.3). Ze względu na obecność w sekwencji RNA tylko czterech podstawowych znaków jest to proces trudny, zwłaszcza dla sekwencji o niskim

podobieństwie i często wymaga wprowadzania poprawek przez użytkownika na podstawie np. znanej struktury drugorzędowej obu cząsteczek. Sukces modelowania bardzo silnie zależy od jakości szablonu i przyrównania.

Główną funkcją programu ModeRNA jest **budowa modelu**. Użytkownik ma do wyboru wiele opcji, które pozwalają na indywidualizację modelowania i dostosowanie go do potrzeb użytkownika. Korzystając z serwera internetowego model może zostać wykonany automatycznie bez konieczności instalacji programu. Alternatywnie, program może zostać pobrany ze strony internetowej i zainstalowany, co pozwoli na modelowanie z zachowaniem kontroli nad poszczególnymi operacjami, np. wybraniem najlepszych kandydatów na fragmenty dla sekwencji celu, która nie ma szablonu.

Optymalizacja i ocena struktury nie są konieczne do uzyskania modelu. Niemniej optymalizacja często pozwala na znaczącą poprawę jakości modelu. W wersji lokalnej programu ModeRNA opcja ta jest dostępna poprzez skrypt uruchamiający zewnętrzny program MMTK (Hinsen, 2000) na cząsteczce modelu. Dostępne są również funkcje optymalizujące ciągłość łańcucha głównego. Jednoznaczna weryfikacja poprawności modelu możliwa jest na ogół dopiero po jego rozwiązaniu doświadczalnym. Program ModeRNA, w wersji lokalnej oraz poprzez serwer, zapewnia jednak możliwość oszacowania jego jakości. Umożliwia sprawdzenie poprawności geometrycznej cząsteczki. Nie jest to kryterium pewne – struktury zdeponowane w bazie PDB również posiadają nietypowe wartości kątów i wiązań, jednak umożliwia to zidentyfikowanie regionów wymodelowanych zupełnie błędnie (Davis, i wsp., 2007). Ponadto, dostępne są operacje pozwalające na analizę modelu np. sprawdzenie parowania zasad, co również może pomóc ocenić uzyskany model.

Podsumowując, oprogramowanie ModeRNA w różnym stopniu pozwala na wykonanie każdego z pięciu etapów prowadzących do uzyskania modelu homologicznego. Użytkownik może korzystać z serwera internetowego ModeRNA lub zainstalować program na swoim komputerze, co pozwoli na użycie wiersza poleceń lub interfejsu programistycznego. Oba sposoby użycia umożliwiają automatyczną budowę modelu oraz ocenę jego jakości. Zaletą serwera jest to, że pozwala na znalezienie szablonu i wykonanie przyrównania. Zainstalowana wersja programu daje możliwość optymalizacji modelu oraz dostęp do poszczególnych operacji zaimplementowanych w programie, co pozwala na kontrolowaną budowę modelu oraz zaawansowane redagowanie struktury.

4.2. Operacje zaimplementowane w programie ModeRNA

Operacje wykonywane przez program ModeRNA podzielić można na trzy grupy. W celu uniknięcia problemów w trakcie modelowania zaimplementowane zostały operacje, które pozwalają sprawdzić poprawność danych wejściowych (rozdział 4.2.1). Największa grupa operacji dostępnych w programie ModeRNA jest związana z modelowaniem struktury przestrzennej cząsteczki (rozdział 4.2.2). Struktura (zarówno szablon, jak i model), może być analizowana, co ułatwia wybór szablonu i pozwala na wstępną ocenę jakości modelu (rozdział 4.2.3). Wszystkie operacje zostały podsumowane w rozdziale 4.2.4. Cały proces modelowania, podczas którego dla każdej pozycji w przyrównaniu zostaje przeprowadzona przynajmniej jedna operacja opisana w rozdziale 4.2.2, może zostać wykonany automatycznie (rozdział 4.2.5). Użytkownik ma również możliwość redagowania wybranych elementów struktury bez uruchamiania całego procesu modelowania. Niektóre operacje, takie jak manipulacja fragmentami drugorzędowymi, nie są uwzględnione w automatycznym procesie modelowania, lecz dostępne jedynie przez interfejs programistyczny.

Podsumowując, program ModeRNA udostępnia trzy grupy operacji związane z: przygotowaniem danych wejściowych, modelowaniem oraz analizą struktury. Użytkownik może wywołać je pojedynczo, używając funkcji z interfejsu programistycznego, lub wykonać jedną komendą proces modelowania, w którym podstawowe operacje pozwalające na zbudowanie modelu zostaną wywołane automatycznie.

4.2.1. Przygotowanie danych wejściowych

Program ModeRNA obsługuje dwa rodzaje plików wejściowych: pliki ze strukturą przestrzenną cząsteczek RNA w formacie PDB i pliki z przyrównaniem w formacie FASTA. Pliki PDB przechowują dane strukturalne i mogą zawierać koordynaty szablonu do modelowania lub struktury, która ma zostać poddana analizie lub redagowaniu. W procedurze automatycznej model jest budowany w oparciu o pojedynczy szablon (jeden łańcuch), aby wykorzystać kilka szablonów należy użyć interfejsu programistycznego programu. W formacie FASTA dostarczone są dane o przyrównaniu dwóch sekwencji. Jeśli użytkownik chce jedynie analizować lub redagować strukturę nie są one konieczne. Za poprawność merytoryczną obu plików odpowiada użytkownik. Poprawność techniczna może zostać sprawdzona przez program ModeRNA. W przypadku danych strukturalnych kontrolowana jest obecność cech, które mogą stanowić problem w procesie modelowania

(rozdział 4.2.1.1). Dla przyrównania analizowana jest zgodność sekwencji między przyrównaniem i szablonem (rozdział 4.2.1.2). **Podsumowując**, program ModeRNA sprawdza poprawność techniczną plików z danymi wejściowymi w formacie PDB i FASTA.

4.2.1.1. Oczyszczanie struktury szablonu

Struktura szablonu wczytywana jest do programu ModeRNA z pliku PDB. Użytkownik musi podać jego nazwę oraz identyfikator łańcucha, który ma zostać wykorzystany do modelowania. Program ModeRNA sprawdza czy dana struktura nie zawiera cech, które utrudnią lub uniemożliwią modelowanie. Przygotowuje szczegółowy raport o zidentyfikowanych trudnościach i wymienia identyfikatory problematycznych reszt. Część problemów może zostać usunięta z pliku PDB podczas operacji oczyszczania struktury. Wszystkie cechy wskazane jako utrudniające modelowanie zostały ustalone przez testowanie programu na plikach z bazy PDB i są to:

- Cząsteczki ligandów, jonów lub wody.
- Nieciągłość łańcucha RNA.
- Łańcuch białkowy.
- Niekompletna lub brakująca zasada.
- Stara nomenklatura nazw atomów rybozy i grupy fosforanowej.
- Brakujące reszty fosforanowe.
- Reprezentacja uproszczona (reszta nukleotydu reprezentowana jest przez pojedynczy atom fosforu).
- Obecność alternatywnych współrzędnych dla atomu.

Obecność wody, jonów lub ligandów w łańcuchu może spowodować, że sekwencja szablonu nie będzie zgodna z sekwencją w przyrównaniu. Na końcu sekwencji szablonu odczytanej przez program ModeRNA pojawią się kropki oddzielone podkreśleniem dolnym. Kropka (.) oznacza nierozpoznaną resztę, natomiast podkreślenie dolne (⏟) wskazuje, że nie jest ona połączona z innymi resztami cząsteczki RNA. Przykładowo, sekwencja łańcucha RNA, w którym występuje pojedynczy jon będzie wyglądała następująco:

AAAAA_.

A w przypadku, jeśli pojawi się w nim kilka (w poniższym przykładzie dziesięć) cząsteczek jonów:

AAAAA_._._._._._._._._._.

Seryjne występowanie wody i jonów jest częste w strukturach PDB i można je obserwować np. w tRNA fenyloalaninowym drożdży – kod PDB: 1EHZ (60 cząsteczek wody oraz sześć jonów magnezu i trzy jony manganu).

Nieciągłość łańcucha RNA może być spowodowana brakującymi atomami łańcucha głównego, zbyt dużą odległością między atomami (niepoprawna długość wiązania/wiązań), lub brakiem całej reszty/grupy reszt. Oznaczać może również, że dwie reszty nie są połączone np. stanowią końce 3' i 5' helisy, której obie nici mają ten sam identyfikator łańcucha. W programie ModeRNA obecność przerwy jest reprezentowana w sekwencji przez znak podkreślenia dolnego (), np.

AA_

Cecha ta nie jest naprawiana podczas operacji oczyszczania szablonu, ponieważ wiązałoby się to ze zbyt dużą ingerencją w koordynaty struktury rozwiązywanej doświadczalnie. Ciągłość łańcucha jest naprawiana w modelu. Użytkownik może też próbować poprawić ciągłość łańcucha w szablonie na własną odpowiedzialność, używając dedykowanej do tego celu operacji (funkcja `fix_backbone()`). Przyczyny nieciągłości łańcucha nie są jednoznaczne i w niektórych przypadkach jej usunięcie nie jest możliwe. Przykładowo, w strukturze większej podjednostki rybosomu *Haloarcula marismortui* (kod PDB: 1FFK) występuje dziewięć regionów nieciągłości łańcucha o identyfikatorze „0”. Wszystkie wynikają z nieobecności niektórych reszt nukleotydów. Brakuje fragmentów o długości od jednej reszty aż do stu (między resztami o identyfikatorach 2136 i 2237). Nieciągłości łańcucha wynikające z takiej przyczyny nie mogą być naprawione przez optymalizację długości wiązań, wymagają użycia opcji do modelowania i wstawienia brakujących reszt (rozdział 4.2.2.4), co w przypadku fragmentu o długości stu reszt może prowadzić do uzyskania cząsteczki o strukturze dalekiej od natywnej.

W przypadku wczytania do programu ModeRNA **łańcucha białkowego** jego sekwencja nie zostanie rozpoznana i będzie przedstawiona jako kropki (.) oddzielone podkreśleniami dolnymi (_):

· _ · _ · _ · _ · _ · _ · _ · _ · _ · _ ·

Program ModeRNA nie modeluje białek i reszty aminokwasów nie są rozpoznawane. Podczas procesu oczyszczania takiego szablonu wszystkie reszty zostaną usunięte, a wynikiem będzie pusty łańcuch. Użytkownik powinien użyć łańcucha zawierającego cząsteczkę kwasu nukleinowego do przeprowadzenia modelowania.

Niekompletna lub brakująca zasada nie zostanie zidentyfikowana przez program ModeRNA i w sekwencji pojawi się na miejscu takiej reszty znak kropki (.), tym razem wewnątrz sekwencji, np.

AAA.AA

Naprawienie takiej reszty nie jest możliwe podczas operacji oczyszczania struktury. Jednak nadal może ona być wykorzystywana w procesie modelowania i jej zasada może zostać zastąpiona przez dowolną inną, pod warunkiem, że reszta ma kompletny pierścień rybozy oraz posiada atom N1 (w przypadku pirymidyny) lub atom N9 (w przypadku puryny).

Stara nomenklatura nazw atomów rybozy i grupy fosforanowej (spotykana w przed ujednoliceniem nazewnictwa w 2008 roku), czyli obecność znaku gwiazdki (*) zamiast znaku prim (') oraz niewłaściwy szyk liter (np. O1P zamiast OP1), uniemożliwia poprawne rozpoznawanie wiązania fosfodiesterowego pomiędzy resztami nukleotydów. Skutkuje to pojawieniem się podkreśleń dolnych (_) w sekwencji, wskazujących na nieciągłości łańcucha:

A _ A _ A _ A _ A

Niestandardowe nazwy podczas operacji oczyszczania struktury są zastępowane nową nomenklaturą.

Brak grupy fosforanowej na końcu 5' nie ma swoich konsekwencji w sekwencji odczytanej przez program ModeRNA, w innych miejscach spowoduje wstawienie znaku podkreślenia dolnego (_). Ponadto spowoduje to problemy gdy dana reszta ma zostać użyta jako reszta kotwicząca podczas wstawiania fragmentu (rozdział 4.2.2.4). Atom fosforu jest

bowiem niezbędny przy nakładaniu fragmentu tak, aby pasował do struktury modelu. Brak grupy fosforanowej dotyczy najczęściej reszty nukleotydowej na końcu 5' łańcucha (np. 16S RNA *E. coli*, kod PDB: 1YOU). Program ModeRNA podczas operacji oczyszczania struktury dobudowuje brakującą grupę fosforanową.

W przypadku, kiedy **reszty mają reprezentację uproszczoną**, np. jednej reszcie odpowiada pojedynczy atom fosforu lub atom fosforu i węgiel C4', a pozostałe atomy nie są obecne, modelowanie nie może zostać przeprowadzone. Program ModeRNA wymaga reprezentacji pełnoatomowej szablonu. Reszty zawierające jedynie pojedynczy atom fosforu zostaną usunięte podczas operacji oczyszczania szablonu.

Dla **atomów posiadających alternatywne położenia** używane są współrzędne wymienione jako pierwsze w pliku PDB. Większa liczba współrzędnych wynika z różnego położenia części populacji danego atomu podczas analizy kryształu, co jest spowodowane nieuporządkowaniem danego fragmentu bądź błędem podczas krystalizacji. Podczas sprawdzania struktury na obecność cech mogących przeszkodzić w modelowaniu generowany jest komunikat o istnieniu takich atomów. Jeśli użytkownik chce zachować inne współrzędne niż te wymienione jako pierwsze musi redagować plik ze strukturą i zamienić odpowiednie linie.

Podsumowując, program ModeRNA nie tylko sprawdza występowanie problematycznych cech w strukturze, ale może również usunąć lub poprawić część z nich:

- Usuwa cząsteczki wody, jonów i ligandów z łańcucha.
- Usuwa reszty aminokwasowe z łańcucha.
- Zmienia nomenklaturę nazw atomów rybozy i grupy fosforanowej (C4* → C4', O1P → OP1).
- Dodaje brakujące grupy fosforanowe.
- Usuwa reszty nukleotydów o reprezentacji uproszczonej do pojedynczych atomów fosforu.

Użytkownik może zatem sprawdzić i wyeliminować problematyczne cechy. Operacja ta może być wykonana niezależnie od procesu modelowania. Cechy takie mogą bowiem nie tylko utrudnić budowę modelu przez program ModeRNA, ale mogą również stanowić

przeszkodę we wczytaniu pliku do innych programów bioinformatycznych. **Podsumowując**, po oczyszczeniu szablonu użytkownik dostaje strukturę, która nie zawiera wody, jonów, ligandów, aminokwasów i reszt o uproszczonej reprezentacji, ma nową nomenklaturę nazw atomów, a wszystkie reszty zawierają kompletną grupę fosforanową.

Funkcje umożliwiające wykonanie operacji sprawdzenia i oczyszczenia struktury szablonu to: `examine_structure()` i `clean_structure()`.

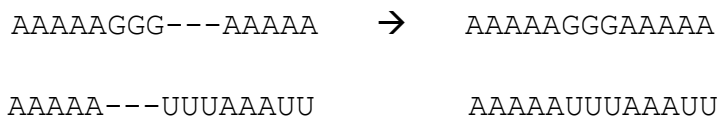
4.2.1.2. Weryfikacja poprawności technicznej przyrównania

Przyrównanie wczytywane jest do programu ModeRNA z pliku FASTA o podanej przez użytkownika lokalizacji. Plik powinien zawierać dwie sekwencje - dodatkowe sekwencje zostaną zignorowane. Jako pierwsza musi zostać podana sekwencja celu. Jest to sekwencja, którą będzie miała cząsteczka uzyskana w wyniku modelowania. Druga sekwencja należy do szablonu i musi być zgodna z rzeczywistą sekwencją struktury szablonu. Insercje i delecje reprezentowane są przez znak myślnika (-). Długość sekwencji celu musi być równa sekwencji szablonu (licząc znaki myślnika). Niepoprawne pod względem technicznym przyrównanie uniemożliwia przeprowadzenie modelowania. Program ModeRNA oferuje możliwość sprawdzenia czy oba pliki wejściowe (przyrównanie i szablon) są zgodne. Poprawność merytoryczna przyrównania zależy całkowicie od użytkownika. Błędne przyrównanie z dużym prawdopodobieństwem spowoduje wygenerowanie błędnego modelu.

Przyczyną niezgodności pomiędzy przyrównaniem i szablonem może być brakująca lub uszkodzona zasada. Nawet jeśli reszta bez zasady jest oznaczona jako fosforan-5'-adenozyny (A) dla programu ModeRNA będzie to reszta niezidentyfikowana (.), ponieważ rozpoznaje on reszty na podstawie analizy geometrycznej, a nie przez informację tekstową zawartą w pliku. Błędy mogą również wynikać z cech takich jak obecność cząsteczek wody lub jonów, które mają ten sam identyfikator łańcucha, co cząsteczka RNA. Przypadki takie mogą zostać rozwiązane poprzez oczyszczenie pliku PDB (rozdział 4.2.1.1).

Program ModeRNA automatycznie wykrywa, kiedy w strukturze szablonu występuje modyfikowana reszta nukleotydu, a w sekwencji w przyrównaniu podany jest jej niemodyfikowany odpowiednik. Rozpoznawana jest również obecność przerw w łańcuchu, zarówno w strukturze jak i podanych w przyrównaniu jako znaki podkreślenia dolnego (_).

Dodatkowo program ModeRNA minimalizuje liczbę niesparowanych reszt w przypadku, gdy jest to możliwe:



Podsumowując, operacja sprawdzania poprawności przyrównania weryfikuje zgodność sekwencji szablonu, znajdującej się w pliku z przyrównaniem, oraz sekwencji odczytanej ze struktury szablonu. W jej wyniku użytkownik otrzymuje informacje czy sekwencje są zgodne. Operacja ta odnosi się jedynie do poprawności technicznej, za merytoryczną ocenę przyrównania odpowiedzialny jest użytkownik.

Funkcja umożliwiająca sprawdzenie przyrównania to `match_template_with_alignment()`.

4.2.2. Modelowanie struktury

Przyrównanie sekwencji celu i szablonu jest instrukcją dla procesu modelowania. Do każdej pozycji w przyrównaniu zostaje przypisana operacja kopiowania reszty (rozdział 4.2.2.1), wymiany zasady (rozdział 4.2.2.2) lub dodania albo usunięcia modyfikacji (rozdział 4.2.2.3). W przypadku, gdy część sekwencji celu nie ma swojego odpowiednika w szablonie, do modelu wstawiany jest fragment strukturalny pochodzący z biblioteki programu ModeRNA. W odwrotnej sytuacji, kiedy szablon posiada region nieobecny w sekwencji celu, nadwymiarowa grupa reszt musi zostać usunięta (rozdział 4.2.2.4) (Rother, i wsp., 2011). Kiedy redagowania wymaga fragment struktury drugorzędowej dodatkowe informacje, oprócz danych z przyrównania, muszą zostać podane przez użytkownika (rozdział 4.2.2.5).

Podsumowując, program ModeRNA buduje model stosując operację kopiowanie reszt nukleotydów, wymiany zasady, dodawania lub usuwania modyfikacji oraz wstawiania fragmentów z wewnętrznej biblioteki. Użytkownik może również przy pomocy programu ModeRNA redagować strukturę drugorzędową modelu.

4.2.2.1. Modelowanie nukleotydów identycznych pomiędzy celem i szablonem

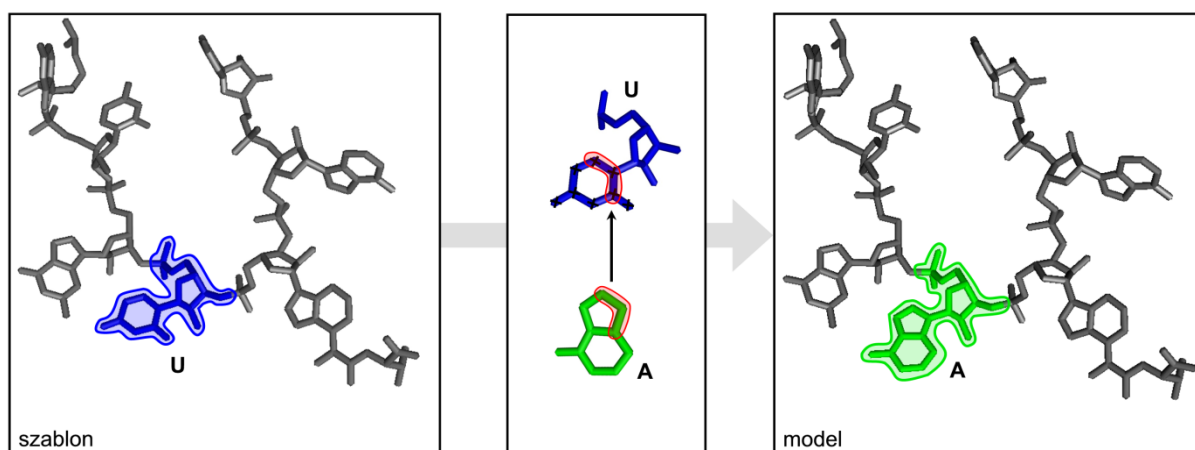
Dla reszt identycznych pomiędzy sekwencją celu i szablonu współrzędne atomów są kopiowane z szablonu do modelu. Od tej operacji rozpoczyna się proces modelowania.

Do tak utworzonego rusztowania dodawane są następnie reszty z wymienioną zasadą, modyfikacje oraz fragmenty z biblioteki. Użytkownik może kontrolować tę operację i kopiować pojedyncze reszty nukleotydów.

Funkcje związane z kopiowaniem reszt identycznych między celem i szablonem to: `copy_identical_residues()`, `copy_single_residue()`, `copy_some_residues()`.

4.2.2.2. Modelowanie substytucji

W przypadku, kiedy reszcie celu w szablonie odpowiada reszta z inną zasadą, do modelu kopiowane są atomy rybozy i grupy fosforanowej, zasada zaś jest wymieniana. Pogram ModeRNA zawiera reprezentatywne współrzędne położenia atomów czterech standardowych zasad występujących w RNA. Nowa zasada, zgodna z sekwencją celu, jest nakładana na starą tak, aby uzyskać konformację pasującą do konformacji rybozy. Operacja wykonywana jest przy pomocy modułu `Superimposer` (rozdział 3.2.3) z pakietu `Bio.PDB`, a do nałożenia wykorzystywane są trzy atomy zasady najbliższe wiązaniu N-glikozydowemu. W przypadku puryny są to atomy C8-N9-C4, a dla pirymidyny C6-N1-C2. Zasada szablonu zostaje następnie usunięta (Rycina 15). **Podsumowując**, operacja modelowania substytucji powoduje, że w modelu znajdują się atomy łańcucha głównego skopiowane z szablonu oraz zasada zgodna z sekwencją celu, pasująca swoją konformacją do konformacji rybozy.



Rycina 15. Operacja zastąpienia uracylu przez adeninę.

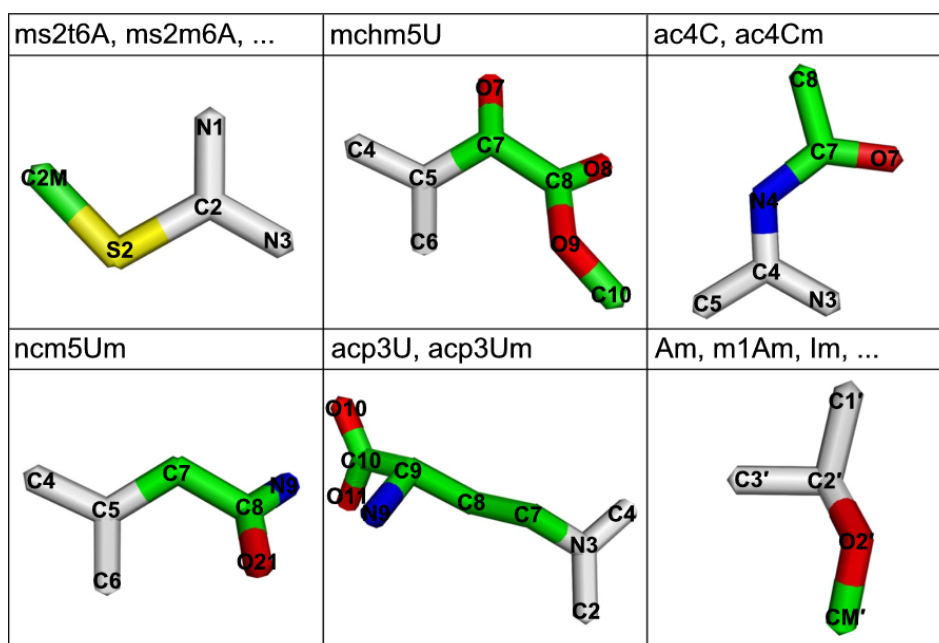
Funkcje odpowiedzialne za modelowanie substytucji to `exchange_mismatches()`, `exchange_single_base()`, `exchange_some_bases()`.

4.2.2.3. Modelowanie modyfikowanych nukleotydów

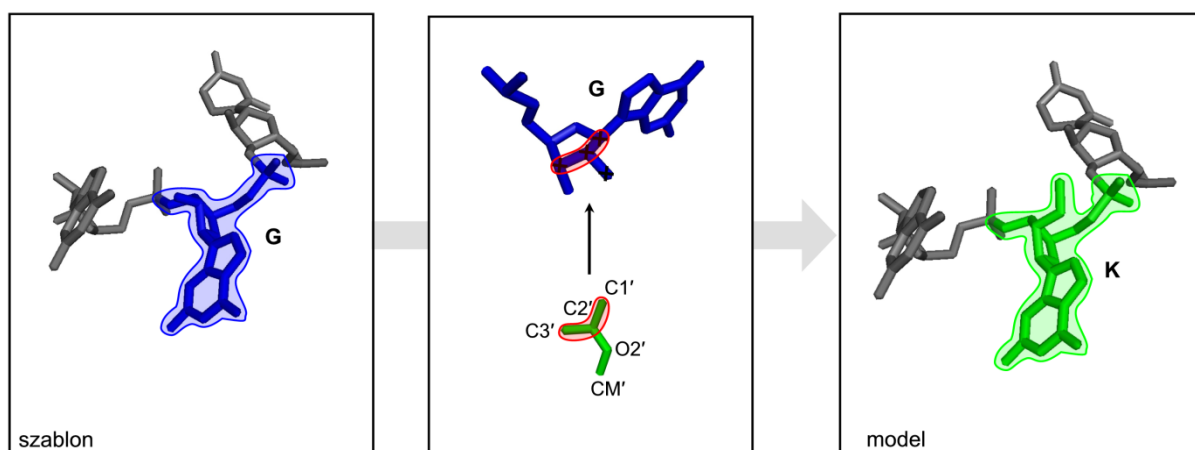
Program ModeRNA buduje modele, w których pojawiają się reszty modyfikowanych nukleotydów nieobecne w szablonie, usuwa także modyfikacje, które występują w szablonie, ale nie w sekwencji celu. Modelowanych jest 115 modyfikacji.

Podczas operacji dodawania modyfikacji do modelu wstawiona zostaje reszta nukleotydu, od którego pochodzi dana modyfikacja (zależnie od tego, czy taka reszta występuje w szablonie, czy też jest obecna inna reszta nukleotydu, dokonywana jest operacja kopiowania – rozdział 4.2.2.1 lub wymiany zasady – rozdział 4.2.2.2), następnie dodawane są do niej odpowiednie grupy funkcyjne (Rycina 16), a zbędne atomy są usuwane (Rycina 17).

Program ModeRNA ma wbudowaną bibliotekę fragmentów grup funkcyjnych. Biblioteka zawiera 70 fragmentów. Każdy fragment przechowywany jest w osobnym pliku PDB. Oprócz atomów grupy funkcyjnej fragment zawiera trzy atomy, których odpowiedniki obecne są w oryginalnej reszcie i które zostają wykorzystane do nałożenia grupy funkcyjnej na niemodyfikowaną resztę w modelu. Dzięki temu, że kilka grup funkcyjnych może zostać dodanych do jednego nukleotydu, pozwalają one na modelowanie 115 modyfikacji. Przykładowo, aby utworzyć 1,2'-metyloadenozynę należy dodać do adenozyiny grupę metylową (CH_3 -) w pozycji 1 pierścienia adeniny oraz w pozycji 2' pierścienia rybozy. Fragmenty zostały przygotowane w oparciu o bazę MODOMICS (Czerwoniec, i wsp., 2009).



Rycina 16. Przykłady fragmentów używanych przy dodawaniu modyfikacji. Na biało zaznaczono atomy wykorzystywane do nałożenia fragmentu na niemodyfikowany nukleotyd.



Rycina 17. Operacja dodania grupy metylowej do rybozy reszty guanozyny i zamiany guanozyny na 2'-O-metyloguanozynę.

Utworzenie konkretnej modyfikacji jest możliwe dzięki zbiorowi instrukcji zgromadzonych w pliku tekstowym `modifications`. Pojedyncza modyfikacja posiada przynajmniej jedną instrukcję, a gdy do utworzenia modyfikacji konieczne jest dodanie więcej niż jednej grupy funkcyjnej, wtedy liczba instrukcji odpowiada liczbie dodawanych grup. Poniżej przedstawiona została instrukcja używana podczas tworzenia 2'-O-metyloguanozyny:

```
Gm | G | O2' | C1',C2',C3' | C1',C2',C3' | C1pC2pC3p_O2p_m.pdb
```

Instrukcja zawiera:

- Kilkuliterowy skrót nazwy modyfikacji.

Gm

- Nazwę oryginalnego nukleotydu, od którego pochodzi modyfikacja.

G

- Listę atomów do usunięcia z oryginalnego nukleotydu (oprócz trzech atomów użytych do nałożenia fragmentu).

[O2']

- Listę trzech atomów z fragmentu użytych do przestrzennego nałożenia grupy funkcyjnej na niemodyfikowany nukleotyd (Rycina 16 - zaznaczone na białło).

[C1', C2', C3']

- Listę trzech atomów z oryginalnego nukleotydu użytych do przestrzennego nałożenia, jak wyżej.

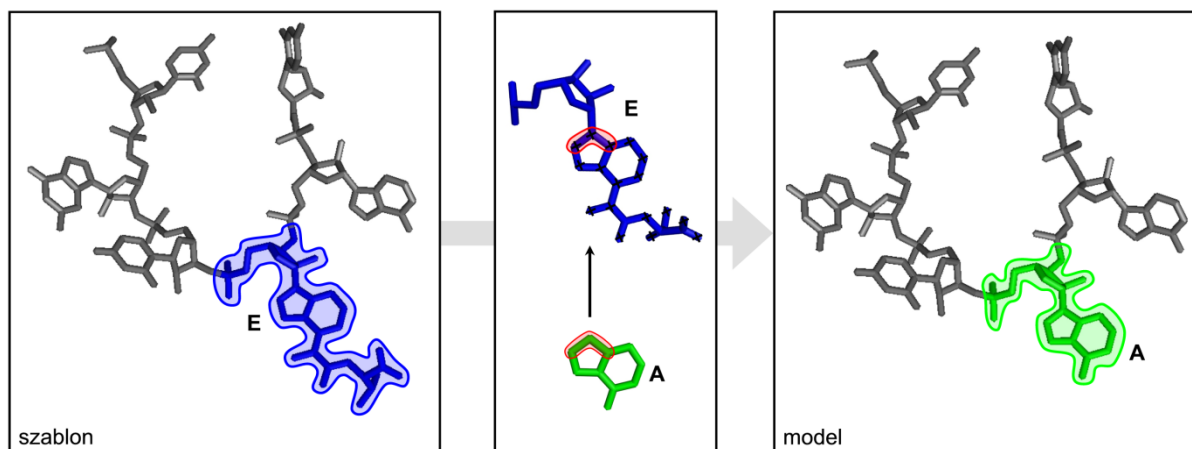
[C1', C2', C3']

- Nazwę pliku PDB zawierającego odpowiednią grupę atomów.

C1pC2pC3p_O2p_m.pdb

W przypadku, gdy zwykła reszta nukleotydu ma zostać zastąpiona przez pseudourydynę lub jej pochodną, niezbędna jest izomeryzacja pierścienia uracylu. Zasada ta zostaje wczytana i nałożona na niemodyfikowaną według atomów C4-C5-C6 (zasada z szablonu) i C2-N1-C6 (wczytana zasada). Następnie wszystkie atomy zasady niemodyfikowanej zostają usunięte.

Podczas operacji usunięcia modyfikacji reszta nukleotydu posiadająca dodatkową grupę funkcyjną zostaje zastąpiona przez resztę niemodyfikowaną. Program ModeRNA wczytuje standardową zasadę i nakłada ją na modyfikowaną używając trzech atomów najbliższych wiązaniu N-glikozydowemu, a zasada reszty modyfikowanej oraz zbędne atomy rybozy zostają usunięte (Rycina 18).



Rycina 18. Operacja usunięcia modyfikacji i zastąpienia N6-metylo-N6-trenylokarbomyloadenozyny przez adenozyne.

Nomenklatura modyfikowanych nukleotydów zgromadzona jest w pliku `modification_names_table`. Oprócz pełnej nazwy np. N6-methyl-N6-threonylcarbamoyladenosine (N6-metylo-N6-trenylokarbomyloadenozyna) nukleotyd może być reprezentowany przez cztery rodzaje skrótów:

- „Tradycyjny”, najczęściej używany skrót kilkuliterowy (np. m6t6A).

- Skrót jednoliterowy używany do reprezentowania reszty w sekwencji (np. E) (McCloskey and Rozenski, 2005).
- Skrót numeryczny (np. 15A) (Dunin-Horkawicz, i wsp., 2006).
- Skrót typowy dla formatu PDB (np. 66A).

Dla każdej reszty w strukturze użytkownik może odczytać każdą z tych nazw. Taka strategia wynika z braku idealnego systemu nazewnictwa modyfikowanych nukleotydów. Pełna nazwa, najbardziej jednoznaczna, jest długa (może mieć nawet 50 znaków np. 2-methylthio-N6-hydroxynorvalyl carbamoyladenosine (2-metyltio-N6-hydroksynorwalyl karbamoyloadenozyny)) i ze względów praktycznych byłaby trudna do wykorzystania w sekwencji. Podobnie skrót kilkuliterowy, choć jest łatwy do skojarzenia z pełną nazwą modyfikacji, byłby trudny do umieszczenia w sekwencji, a ponadto nomenklatura ta nie jest jednolita i zdarza się, że jedna modyfikacja ma kilka skrótów, np. dla 5-metylocytydiny (5-methylcytidine) spotykane są w literaturze skróty 5mC, m5C, m⁵C, mC⁵ i mC (Rother, i wsp., 2011). Skrót jednoliterowy może zostać umieszczony w sekwencji, jednak ze względu na ograniczoną liczbę znaków alfanumerycznych nie wszystkie ze 115 modyfikacji mogą być modyfikowane w ten sposób. Skrót z bazy PDB, podobnie jak skrót kilkuliterowy, jest trudny do użycia w sekwencji, a ponadto nie jest jednoznaczny, np. w pliku PDB o identyfikatorze 1F7U 1-metyloadenozyna 958 z łańcucha B oznaczona jest jako 1MA, podczas gdy chemicznie identyczna reszta 58 z pliku PDB o identyfikatorze 1OB2 (łańcuch B) nosi nazwę MAD (Rother, i wsp., 2011). Nazwy i skróty używane dla modyfikowanych reszt nukleotydów w programie ModeRNA zostały zaczerpnięte z bazy MODOMICS (Czerwonec, i wsp., 2009).

Program ModeRNA umożliwia modelowanie modyfikowanych reszt nukleotydów. Dodanie i usunięcie modyfikacji może się odbyć w sposób automatyczny na podstawie przyrównania lub poprzez wskazanie przez użytkownika konkretnej reszty. Dla operacji dodawania musi zostać również podany kilkuliterowy skrót nazwy modyfikacji. W przypadku usuwania modyfikacji użytkownik ma możliwość jednoczesnego zastąpienia wszystkich modyfikowanych reszt występujących w strukturze modelu lub szablonu przez reszty standardowe.

Funkcje związane z dodawaniem modyfikacji to `add_all_modifications()`, `add_modification()`. Funkcje związane z usuwaniem modyfikacji to:

```
remove_all_modificatons(),    remove_mismatching_modifications(),
remove_modification().
```

4.2.2.4. Modelowanie insercji i delecji

W sekwencji celu mogą pojawić się trzy rodzaje insercji (przerw w sekwencji szablonu): na końcu 5' (sekwencja AAAAA w poniższym przykładzie), we wnętrzu sekwencji (GGGGG w poniższym przykładzie) lub na końcu 3' (sekwencja CCCCC w poniższym przykładzie):

```
>cel
AAAAAUUUUUUGGGGGUUUUUCCCCC
>szablon
-----UUUUUU-----UUUUUU-----
```

Dla każdej insercji do modelu zostaje wprowadzony fragment o długości odpowiadającej przerwie w sekwencji szablonu. Insercja może oznaczać pojawienie się w modelu nieobecnego w szablonie wybruszenia, pętli (lub wydłużenie istniejącej pętli), albo całego elementu struktury drugorzędowej.

Budowa fragmentu

Fragment zbudowany jest z reszt nukleotydów, których liczba odpowiada długości insercji oraz z reszt kotwiczących, które są punktem odniesienia pomiędzy fragmentem i modelem (w modelu są to reszty okalające insercję, na które zostanie nałożony fragment (pogrubione **U** w powyższym przykładzie), we fragmencie są to dodatkowe reszty, które po procesie nałożenia przestrzennego zostaną usunięte). Fragment wstawiany wewnątrz struktury ma dwie reszty kotwiczące, natomiast fragment dodawany na końcu 5' lub 3' ma tylko jedną taką resztę, odpowiednio na końcu 3' lub 5'.

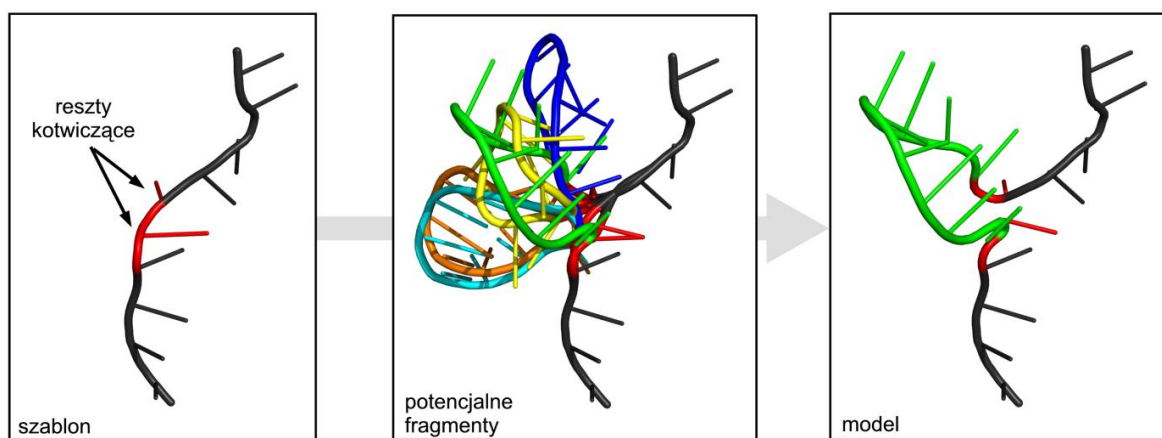
Fragmenty używane podczas modelowania insercji i delecji we wnętrzu struktury zostały uzyskane ze zbioru rnaDB2005 – 172 struktur pochodzących z różnych rodzin RNA (rozdział 2.1.2). Źródłem fragmentów wstawianych na końcu 5' lub 3' jest pojedyncza nić helisy o odpowiedniej długości. Biblioteka fragmentów zawiera 131316 ciągłych fragmentów o długości od dwóch do dziesiętnastu reszt nukleotydów. Na podstawie tej biblioteki modelowane mogą być insercje i delecje o maksymalnej długości siedemnastu reszt nukleotydów (długość insercji/delecji + dwie reszty kotwiczące). Rozmiar ten został wybrany ze względu na wielkość biblioteki – 20MB, przy większej długości fragmentów rozmiar

wzrasta, co może być niepożądane ze strony użytkownika instalującego oprogramowanie na dysku. W przypadku konieczności modelowania insercji/delecji o większej długości użytkownik ma możliwość pobrania ze strony programu ModeRNA pliku z rozszerzoną biblioteką z fragmentami o długości do 100 reszt nukleotydów (111MB). Wszystkie fragmenty z biblioteki (zarówno standardowej jak i rozszerzonej) charakteryzowane są przez:

- Długość (ilość reszt bez reszt kotwiczących, od zera do siedemnastu).
- Identyfikator struktury i łańcucha, z których pochodzi fragment.
- Identyfikatory reszt kotwiczących na końcu 5' i 3'.
- Sekwencję (zawiera reszty kotwiczące).
- Strukturę drugorzędową.
- Sześć dystansów międzyatomowych.

Wyszukiwanie fragmentu

Program ModeRNA dla każdej insercji/delecji identyfikuje fragment o odpowiedniej długości, dla którego dopasowanie reszt kotwiczących pomiędzy modelem i fragmentem jest jak najlepsze, a liczba zderzeń sterycznych ze strukturą modelu minimalna (Rycina 19). Poszukiwanie odpowiedniego fragmentu obejmuje dwa etapy. W pierwszym sprawdzane jest dopasowanie reszt kotwiczących wszystkich fragmentów o odpowiedniej długości z biblioteki. Drugi etap polega na wstawieniu do modelu 50 najlepszych kandydatów i ich ewaluacji.



Rycina 19. Operacja wstawienia fragmentu strukturalnego o długości 9 reszt nukleotydowych do modelu.

zostać obliczone. Różnice odpowiadających dystansów z modelu i fragmentu zostają podniesione do kwadratu i zsumowane:

$$ocena = (u_m - u_{fr})^2 + (w_m - w_{fr})^2 + (v_m - v_{fr})^2 + (x_m - x_{fr})^2 + (y_m - y_{fr})^2 + (z_m - z_{fr})^2,$$

gdzie indeks m oznacza model, indeks fr fragment, a u ; w ; v ; x ; y i z to kolejne dystanse (np. u oznacza dystans $O5'_{r5}-O5'_{r3}$).

Jest to przybliżone oszacowanie wartości RMSD nałożenia danego fragmentu. Porównanie pozwala wyłonić 50 najlepiej dopasowanych fragmentów, które podlegają dalszej ewaluacji.

Drugi etap wyszukiwania fragmentu

Najlepiej dopasowane fragmenty zostają wstawione do modelu przez nałożenie atomów $C1'_{r5}$, $C4'_{r5}$, $C3'_{r5}$, $O3'_{r5}$ i $N1_{r5}/N9_{r5}$ reszty kotwiczącej na końcu 5' oraz atomów $C1'_{r3}$, $C4'_{r3}$, $C3'_{r3}$, $O5'_{r3}$, $N1_{r3}/N9_{r3}$ i $C5'_{r3}$ reszty kotwiczącej na końcu 3' fragmentu na odpowiadające atomy w resztach kotwiczących modelu. Sekwencja fragmentu zostaje zmieniona tak, aby odpowiadała przyrównaniu. Fragment jest oceniany na podstawie RMSD uzyskanego po nałożeniu reszt kotwiczących, niepodobieństwa oryginalnej sekwencji fragmentu i sekwencji z przyrównania oraz liczby zderzeń sterycznych z resztą modelu:

$$ocena = a * RMSD + b * sekwencja + c * zderzenia$$

Największą wagę przy ocenie fragmentu ma nałożenie reszt kotwiczących ($a = 10.0$), następnie liczba zderzeń sterycznych pomiędzy fragmentem i modelem ($c = 2.0$). W przypadku, gdy dla dwóch fragmentów wartości te są bardzo zbliżone decydujące znaczenie ma niepodobieństwo sekwencji ($b = 1.0$). Jest ono liczone przez zastosowanie następującej matrycy substytucji dla każdej pozycji w przyrównaniu:

- 0.0 – takie same nukleotydy (np. A + A; m6t6A + m6t6A).
- 0.1 – nukleotyd i jego modyfikacja (np. A + m6t6A; U + D).
- 0.5 – tranzycja (np. A + G; C + U).
- 1.0 – transwersja (np. A + U; C + G).

Liczone jest niepodobieństwo zamiast podobieństwa sekwencji tak, aby wszystkie wyniki cząstkowe równania, jak i ostateczna ocena, były tym lepsze im niższą wartość osiągną.

Modelowanie delecji w sekwencji celu

Wstawienie fragmentu strukturalnego ma również miejsce w przypadku, gdy sekwencja szablону jest dłuższa od sekwencji celu, np.:

```
>cel
AAAAAU-----UAAAAA
>szablon
AAAAAUGGGGUAAAAA
```

Przykład takiej sytuacji to usunięcie pętli zmiennej tRNA, lub zastąpienie jej przez krótszą pętlę. Niepotrzebny element struktury zostaje usunięty, a pozostające po nim dwa wolne końce muszą ulec połączeniu. W tym celu wyszukiwany jest i wstawiany fragment składający się z dwóch połączonych ze sobą reszt kotwiczących.

Podsumowanie

Program ModeRNA pozwala na modelowanie sekwencji celu niemającej odpowiednika w szablónie oraz na usuwanie zbytecznych fragmentów szablónu z zachowaniem ciągłości struktury modelu poprzez wprowadzanie do modelu fragmentów strukturalnych wybranych na podstawie kryteriów geometrycznych z wewnętrznej biblioteki programu. Podejście to jest podobne do zastosowanego w programie SWISS-MODEL (Schwede, i wsp., 2003).

Funkcje związane z operacją wstawiania do modelu fragmentów to: `apply_all_indels()`, `apply_indel()`, `apply_missing_ends()`, `create_fragment()`, `find_fragment()`, `insert_fragment()`.

4.2.2.5. *Modelowanie struktury drugorzędowej*

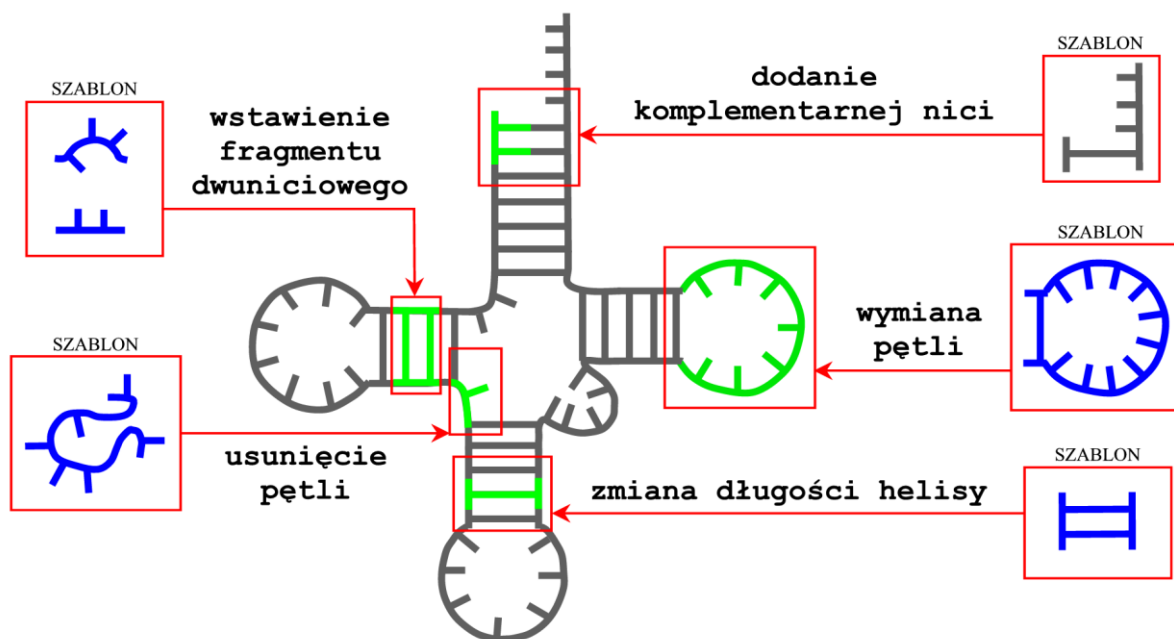
Insercje i delecje w sekwencji celu modelowane są z wykorzystaniem biblioteki fragmentów (rozdział 4.2.2.4) Takie podejście nie jest jednak wystarczające dla modelowania struktury drugorzędowej. Zagadnienie to ilustruje poniższe przyrównanie szablonu posiadającego strukturę spinki do włosów i sekwencji celu:

```
>cel
AAUUUUUAAGGGGGUUAAAAUU
>szablon
AA-----AAGGGGGUU-----UU
```

Szablon jest zbudowany z czterech par zasad typu Watson-Crick (A-U) oraz pętli terminalnej o długości pięciu reszt nukleotydów. Utworzony na jego podstawie model ma mieć helisę dłuższą o pięć par U-A. Oddzielne wstawianie brakujących fragmentów spowoduje, że nie będą one tworzyły helisy. Do modelowania takiego przypadku program ModeRNA oferuje zbiór funkcji operujących na strukturze drugorzędowej.

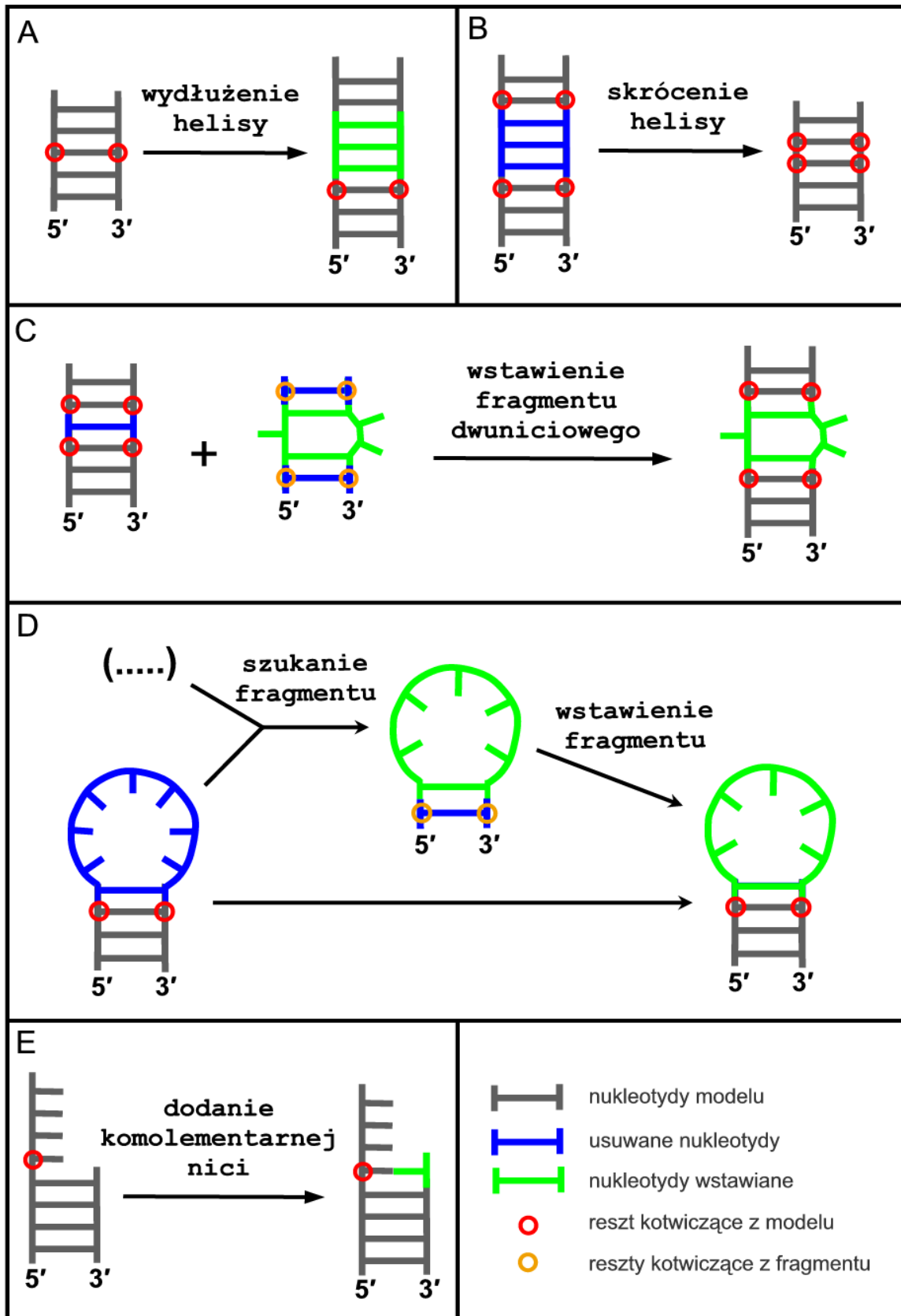
Przypadki, w których w programie ModeRNA może zostać zastosowane bezpośrednie redagowanie struktury drugorzędowej to (Rycina 21):

- Przedłużanie helisy (jak w powyższym przykładzie).
- Skracanie helisy.
- Wstawianie fragmentu dwuniciowego.
- Wstawianie fragmentu jednoniciowego o zadanej strukturze drugorzędowej.
- Dodanie komplementarnej nici do fragmentu jednoniciowego.



Rycina 21. Przypadki, w których niezbędne jest redagowanie całych fragmentów struktury drugorzędowej na przykładzie tRNA.

Zmiana długości helisy odbywa się na podstawie zawartej w programie ModeRNA struktury helisy A typowej dla RNA. W przypadku operacji przedłużania helisy użytkownik musi wskazać dwie reszty kotwiczące, które pozwolą na nałożenie helisy. Część modelu połączona z resztami kotwiczącymi zostanie nałożona na końcu nowo wstawionego odcinka helisy (Rycina 22, A). Skracanie helisy wymaga podania czterech reszt, z pomiędzy których usunięta zostanie helisa, tak, że będą one połączone w uzyskanym modelu (Rycina 22, B). Wstawienie elementu dwuniciowego może się odbyć tylko, jeśli użytkownik ma strukturę takiego fragmentu (plik PDB). Program ModeRNA nie zawiera biblioteki fragmentów dwuniciowych. Operacja wstawienia takiego fragmentu wymaga podania czterech reszt kotwiczących z fragmentu i czterech odpowiadających im reszt kotwiczących z modelu (Rycina 22, C). Wyszukanie i wstawienie fragmentu jednoniciowego o zadanej strukturze drugorzędowej przypomina wstawienie zwykłego fragmentu jednoniciowego. Użytkownik musi jednak dodatkowo zdefiniować strukturę drugorzędową w formacie VIENNA (Rycina 22, D). Dodawanie komplementarnej nici jest wykonywane nukleotyd po nukleotydzie (Rycina 22, E). Użytkownik musi wskazać, do której zasady ma zostać dobudowana para. Dodanie nukleotydów do nici, która nie ma regularnej konformacji helisy A może spowodować, że nie utworzą one ciągłej struktury.



Rycina 22. Operacje na strukturze drugorzędowej dostępne w programie ModeRNA: A-B) zmiana długości helisy; C) wstawienie fragmentu dwuniciowego, posiadającego cztery reszty kotwiczące; D) wyszukanie i wstawienie fragmentu o zadanej strukturze drugorzędowej i dwóch resztach kotwiczących; E) dodanie nukleotydu tak, by utworzył helisę z zadany nukleotydem z komplementarnej nici.

Podsumowując, Program ModeRNA modeluje oddziaływania typu Watson-Crick pomiędzy zasadami modelu. Użytkownik może redagować strukturę drugorzędową poprzez pięć operacji: skrócenie helisy, wydłużenie helisy, dodanie fragmentu dwuniciowego, dodanie fragmentu jednoniciowego o zadanej strukturze drugorzędowej oraz dodanie komplementarnej nici.

Funkcje związane z modelowaniem struktury drugorzędowej to: `add_pair_to_base()`, `create_fragment()`, `extend_helix()`, `insert_fragment()`, `insert_two_strand_fragment()`, `shrink_helix()`.

4.2.3. Analiza struktury

Analiza struktury może zostać wykorzystana podczas identyfikacji szablonu do modelowania lub oceny modelu. Program ModeRNA odczytuje sekwencję struktury bezpośrednio ze współrzędnych atomowych, co pozwala na przygotowanie danych wejściowych do modelowania (rozdział 4.2.3.1). Sprawdza miejsca występowania modyfikowanych nukleotydów i strukturę drugorzędową cząsteczki (rozdział 4.2.3.2), co ułatwia wybór szablonu. Analizuje również geometrię struktury, co daje możliwość wstępnej oceny jakości modelu oraz szablonu (rozdział 4.2.3.3).

Podsumowując, program ModeRNA umożliwia analizę struktury modelu lub szablonu, a także dowolnie wybranej przez użytkownika struktury RNA w formacie PDB.

4.2.3.1. Sprawdzanie rzeczywistej sekwencji cząsteczki

W pliku w formacie PDB sekwencja cząsteczki jest zapisana w wierszach oznaczonych jako SEQRES. Nie zawsze jest ona zgodna ze strukturą, której współrzędne przechowywane są w pliku. Przykładowo w strukturze o identyfikatorze 1H3E brakuje dwóch zasad (w reszcie 16 i reszcie 47D), choć w sekwencji są odnotowane jako C i U. Ponadto dla modyfikowanych nukleotydów używane są niejednoznaczne skróty. Program ModeRNA odczytuje sekwencję struktury na podstawie analizy geometrii poszczególnych reszt nukleotydów (rozdział 3.2.4) (Rother, i wsp., 2010). Dzięki temu użytkownik ma pewność co do zgodności sekwencji i struktury.

W sekwencji może występować 71 rodzajów znaków (Tabela 5). Litery A, C, G i U oznaczają reszty ze standardowymi zasadami (adeniną, cytozyną, guaniną i uracylem), a w przypadku, gdy są pisane małą literą reszta zawiera deoksyrybozę.

Ze 115 modyfikowanych nukleotydów rozpoznawanych przez program ModeRNA, 57 najczęściej występujących w strukturach RNA posiada jednoliterowe skróty (Juhling, i wsp., 2009). Pozostałe 58 oznaczanych jest jako x. Reszta, której program nie jest w stanie rozpoznać jest reprezentowana w sekwencji jako kropka (.), a jeśli zasada została rozpoznana, a problemem jest rozpoznanie modyfikacji, w sekwencji pojawi się znak ; (nieznana modyfikacja guaniny), < (cytozyny), H (adeniny) lub N (uracylu). Znak podkreślenia dolnego () oznacza nieciągłość łańcucha głównego.

Tabela 5. Jednoliterowe skróty mogące występować w sekwencji w programie ModeRNA.

Kategoria	Jednoliterowy skróty w sekwencji
Standardowe reszty nukleotydów RNA	A, C, G, U
Reszty nukleotydów DNA	a, c, g, t
Reszty nukleotydów modyfikowanych	!, ", #, \$, %, &, ', (,), *, +, ,, /, 1, 2, 3, 4, 5, 6, 7, 8, 9, :, =, >, ?, @, B, D, E, F, I, J, K, L, M, O, P, Q, R, S, T, V, W, X, Y, Z, [, \,], ^, ` , {, , }, ~
Nieznane modyfikacje	; (guaniny), < (cytozyny), H (adeniny), N (uracylu)
Reszta modyfikowanego nukleotydu bez skrótu jednoliterowego	x
Nieznana reszta	.
Nieciągłość łańcucha	–

Program ModeRNA umożliwia odczytanie sekwencji struktury na podstawie geometrii jej reszt. Operacja ta może służyć do sprawdzenia czy sekwencja szablonu zawarta w przyrównaniu jest poprawna (taka sama jak rzeczywista sekwencja struktury) lub czy sekwencja uzyskanego modelu jest zgodna z celem. Sekwencje odczytywać można z dowolnych struktur RNA w formacie PDB, np. przy analizie dużego zbioru struktur. Przykładowo była ona wykorzystywana przy tworzeniu bazy fragmentów dostępnej w programie (rozdział 4.2.2.4) oraz przyrównania 99 sekwencji tRNA, wykorzystanych do merytorycznego testowania programu (rozdział 4.5).

Funkcja związana z odczytywaniem sekwencji to `get_sequence()`. Funkcja pozwalająca na wyszukanie modyfikacji to `find_modifications()`.

4.2.3.2. *Sprawdzenie struktury drugorzędowej*

RNA ma budowę modułową i składa się z elementów drugorzędowych i motywów trzeciorzędowych. Znajomość struktury drugorzędowej cząsteczki celu lub jej wiarygodne przewidywanie ułatwia przyrównanie sekwencji celu i szablonu (Rother, i wsp., 2011). Struktura drugorzędowa szablonu jest jednoznaczna i wynika z jego budowy przestrzennej, a jej odczytanie nie jest zadaniem trywialnym. Program ModeRNA identyfikuje wiązania wodorowe poprzez wyszukiwanie par donor protonu – akceptor, leżących w zasadach dwóch różnych reszt w odległości mniejszej niż 3.4 Å i na tej podstawie odczytuje strukturę drugorzędową. Przykładowo w parze G-C wiązania wodorowe tworzone są między grupą NH₂ (donor) przy węglu czwartym cytozyny i O (akceptor) przy węglu szóstym guaniny; N (akceptor) w pozycji trzeciej cytozyny i NH (donor) w pozycji pierwszej guaniny oraz O (akceptor) przy węglu drugim cytozyny i grupą NH₂ (donor) przy węglu drugim guaniny. Wynikiem tej operacji jest sekwencja drugorzędowa w notacji kropek i nawiasów (format VIENNA). Przykładowo struktura spinki do włosów o helisie zbudowanej z ośmiu par zasad nukleotydów i pętli terminalnej o długości pięciu reszt nukleotydowych będzie w formacie VIENNA przedstawiona jako:

(((((((((.))))))))

Podczas odczytywania struktury drugorzędowej nie są uwzględniane pseudowęzły (struktury zawierające dwie helisy, gdzie nieć jednej z helis znajduje się pomiędzy niciami drugiej helisy) – tylko jedna helisa jest odnotowana w strukturze drugorzędowej.

Funkcje związane z odczytywaniem struktury drugorzędowej to `get_base_pairs()`, `get_secstruc()` i `write_secstruc()`.

W implementacji tej operacji brali udział lic. Paweł Skiba i dr Kristian Rother.

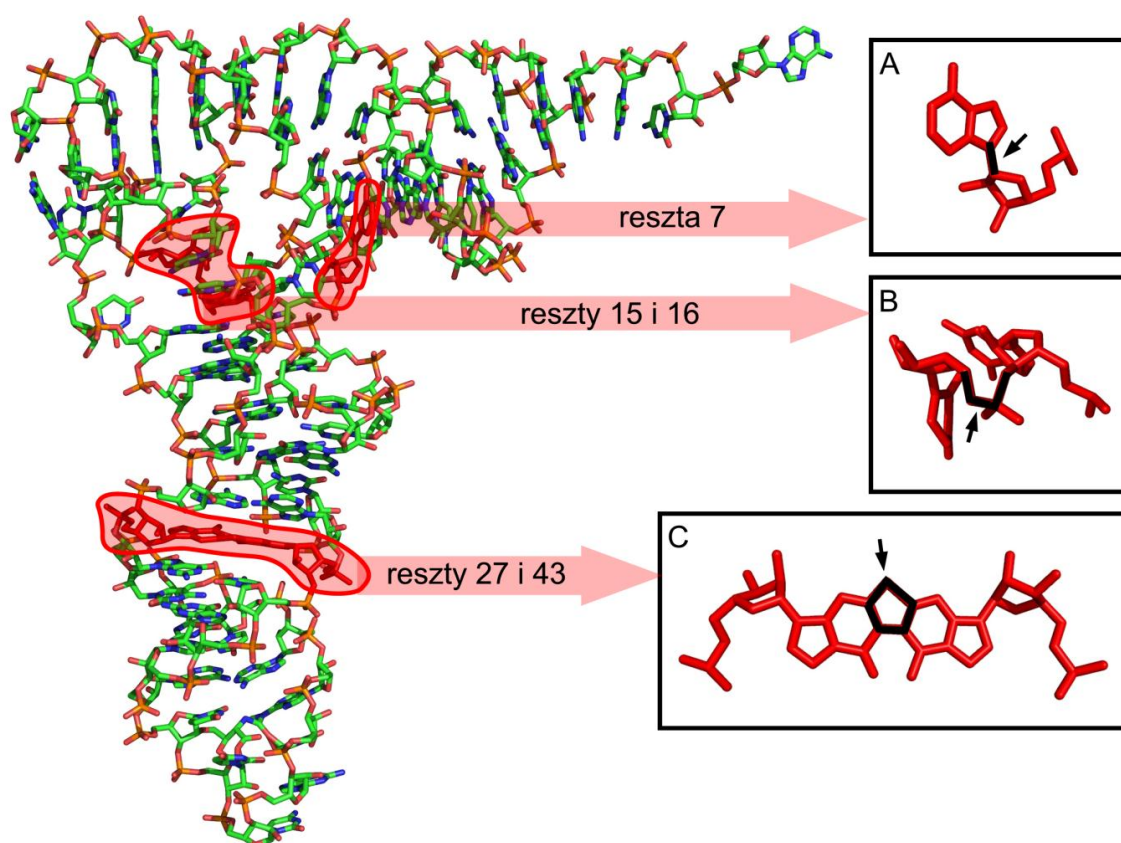
4.2.3.3. *Sprawdzenie geometrii struktury*

Analiza geometrii struktury pozwala na wstępne sprawdzenie poprawności modelu. Zaburzenia geometryczne niekoniecznie oznaczają błędny model. Struktury rozwiązane doświadczalnie również zawierają nietypowe wartości (Murray and Carr, 2008) i niestandardowa geometria w modelu może być spowodowane kopiowaniem błędów obecnych w szablonie. Wysoka liczba błędów może jednak świadczyć o niepoprawnym przyrównaniu lub złym doborze szablonu.

Badanie geometrii struktury może zostać zastosowane, oprócz modelu, również do szablonu i każdej innej struktury RNA w formacie PDB. Program ModeRNA pozwala na identyfikację:

- Nietypowych długości wiązań.
- Nietypowych wartości kątów płaskich.
- Nietypowych wartości kątów torsyjnych.
- Zderzeń sterycznych (nakładania się promieni van der Waalsa dwóch atomów).

Przykładową analizę geometrii struktury modelu przedstawia Rycina 23.



Rycina 23. Reszty nukleotydów wskazane jako niepoprawne w modelu tRNA^{Thr} zbudowanym na szablonie 1EHZ: A) nietypowa długość wiązania N-glikozydowego - 1.38 Å w reszcie 7; B) nietypowa wartość kąta torsyjnego pomiędzy atomami O3' (reszty 15) - P (reszty 15) - O5' (reszty 16) - C5' (reszty 16); C) Zderzenie steryczne między atomami zasad reszt 27 i 43. Przez wartości nietypowe rozumiane są takie, które odbiegają od najczęściej występujących w strukturach w bazie PDB.

Funkcje związane z analizą geometrii struktury to `analyze_geometry()` i `find_clashes()`.

W implementacji tej operacji brał udział mgr. Tomasz Puton.

4.2.4. Podsumowanie operacji programu ModeRNA

Program ModeRNA udostępnia użytkownikom trzy grupy operacji:

- Operacje przygotowujące dane wejściowe.
 - Oczyszczenie struktury szablonu.
 - Sprawdzenie poprawności technicznej przyrównania.
- Operacje związane z modelowaniem.
 - Kopiowanie reszt nukleotydów.
 - Wymiana zasady.
 - Dodawanie i usuwanie grup funkcyjnych modyfikowanych reszt nukleotydów.
 - Wstawianie fragmentów.
 - Redagowanie elementów drugorzędowych.
- Operacje pozwalające na analizę struktury.
 - Odczytywanie sekwencji struktury na podstawie geometrii tworzących ją reszt.
 - Odczytywanie struktury drugorzędowej.
 - Sprawdzanie poprawności geometrycznej struktury.

Program ModeRNA oferuje użytkownikowi funkcje do modelowania, sprawdzenia szablonu i przyrównania, oraz analizy struktury. Trójwymiarowy model cząsteczki celu może zostać uzyskany przez modelowanie pojedynczych reszt, grup reszt, albo też poprzez modelowanie automatyczne. Zaimplementowane w programie ModeRNA operacje mogą zostać również wykorzystane do redagowania modelu lub dowolnej struktury PDB, np. do usunięcia modyfikacji ze struktury, która ma zostać poddana symulacji dynamiki molekularnej, przez program, który nie obsługuje modyfikowanych nukleotydów.

Funkcje odpowiedzialne za użycie poszczególnych operacji podane są w rozdziałach im poświęconych.

4.2.5. Automatyczna budowa modelu

Cały proces modelowania może zostać wykonany automatycznie przy użyciu jednej komendy - program ModeRNA dokona wtedy analizy przyrównania i automatycznie wywoła odpowiednie operacje dla poszczególnych pozycji. Operacje mogą też zostać wywołane przez użytkownika pojedynczo dla poszczególnych reszt nukleotydów. Użytkownik może więc samodzielnie analizować przyrównanie i dla każdej pozycji wybierać preferowaną operację. Inną możliwością to wykonanie pojedynczej operacji lub grupy operacji dla całego przyrównania, np. skopiowanie wszystkich identycznych reszt między szablonem i modelem.

Modelowanie pojedynczych pozycji przez użytkownika pozwala zachować kontrolę nad procesem modelowania, ale jest czasochłonne. Ponadto wymaga użycia interfejsu programistycznego, który dla użytkowników niemających styczności z programowaniem może nie być intuicyjny. Dla prostych przypadków modelowania, kiedy identyczność pomiędzy sekwencją celu i szablonu jest duża, a wszystkie reszty celu posiadają swój odpowiednik w szablonie, łatwiej przeprowadzić modelowanie automatyczne, które może zostać wykonane przez linię poleceń lub serwer programu ModeRNA. Serwer zapewnia szczególnie uproszczoną obsługę programu bez konieczności jego instalacji.

Podczas modelowania automatycznego, po podaniu danych wejściowych w postaci pliku ze strukturą szablonu w formacie PDB oraz przyrównania w formacie FASTA, wykonywane są następujące operacje:

- Oczyszczenie struktury szablonu.
- Weryfikacja zgodności sekwencji szablonu z przyrównaniem z rzeczywistą sekwencją struktury szablonu.
- Kopiowanie reszt identycznych między celem i szablonem.
- Kopiowanie atomów łańcucha głównego i dodanie odpowiedniej zasady dla reszt, które mają swój odpowiednik w szablonie.
- Dodawanie grup funkcyjnych do reszt nukleotydów, które są modyfikowane w sekwencji celu, a modyfikacja nie występuje w strukturze szablonu
- Zastępowanie modyfikowanych zasad standardowymi dla pozycji w przyrównaniu, gdzie modyfikowana jest tylko reszta szablonu.

- Wstawianie jednonuciowych fragmentów strukturalnych dla sekwencji celu, które nie posiadają szablonu.
- Sprawdzenie i naprawa ciągłości łańcucha głównego.
- Sprawdzenie zgodności sekwencji uzyskanego modelu z sekwencją celu podaną przez użytkownika w przyrównaniu.

Podczas modelowania automatycznego nie są brane pod uwagę elementy struktury drugorzędowej. Wszystkie insercje/delecje są traktowane jako niezależne.

Użytkownik może również automatycznie utworzyć wstępny szkielet modelu, zawierający reszty identyczne, substytucje i reszty modyfikowane, a następnie indywidualnie modelować insercje/delecje. Pozwala to nie tylko dodawać elementy drugorzędowe, ale również daje więcej możliwości przy wstawianiu fragmentów jednonuciowych. Użytkownik może poddać wstępnej ocenie większą ilość fragmentów, oraz zapisać struktury kandydatów razem z modelem i przy pomocy przeglądarki graficznej samodzielnie wybrać najlepszy fragment. W przypadku insercji/delecji, dla których nie został zidentyfikowany satysfakcjonujący fragment, insercja/delecja może zostać rozszerzona, tak, że dwie nowe reszty kotwiczące zostaną wzięte pod uwagę.

Na zbiorowe zastosowanie operacji oprócz modelowania insercji/delecji pozwala funkcja `apply_alignment()`. Automatyczne modelowanie jest możliwe dzięki funkcji `create_model()`.

4.3. Użycie programu ModeRNA

Program ModeRNA zaprojektowany został dla trzech rodzajów użytkowników, którzy korzystają z wiersza poleceń (rozdział 4.3.1), interfejsu programistycznego (rozdział 4.3.2) lub interfejsu internetowego (rozdział 4.3.3) (Rycina 24). Do używania wiersza poleceń i interfejsu programistycznego niezbędna jest instalacja programu na komputerze użytkownika. Jego źródło dostępne jest na stronie:

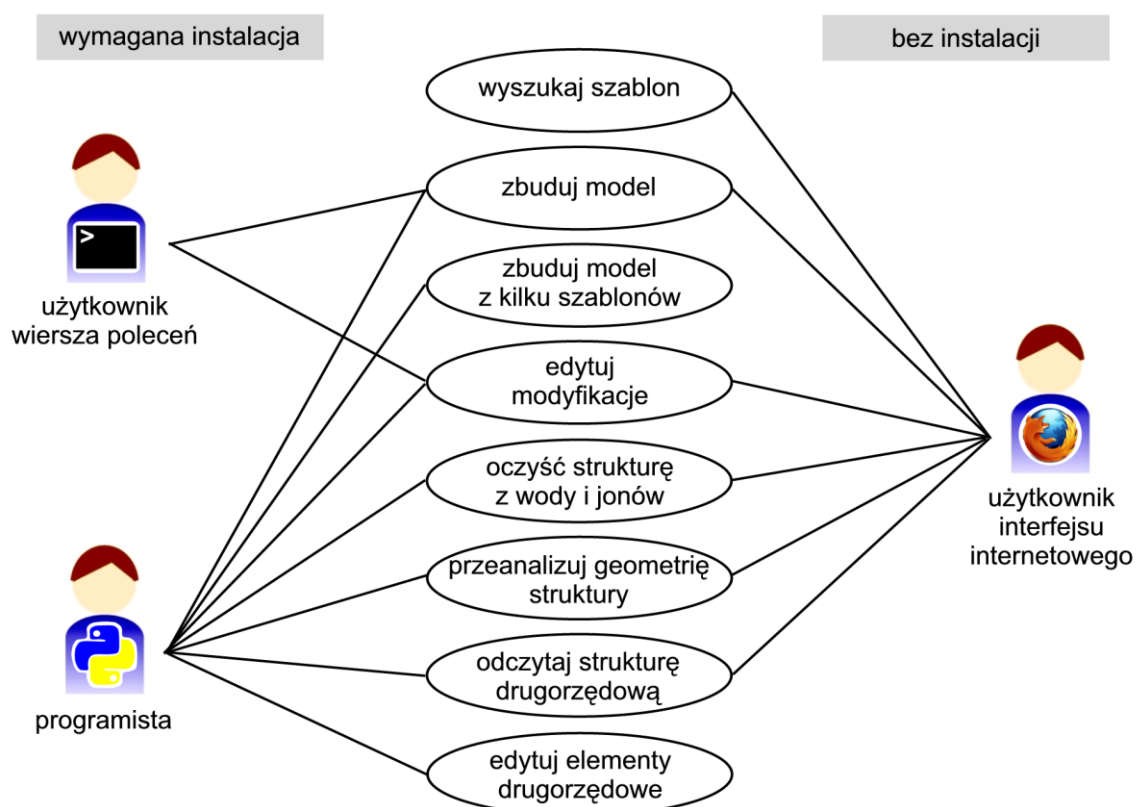
<http://iimcb.genesilico.pl/moderna/download>

Poprawne zainstalowanie programu ModeRNA wymaga wcześniejszej instalacji języka Python (rozdział 3.2.1) i biblioteki Biopython (rozdział 3.2.2). Użycie interfejsu

internetowego nie wymaga instalacji. Użytkownik wysyła na serwer swoje dane wejściowe i może pobrać dane wynikowe. Serwer dostępny jest na stronie:

<http://iimbc.genesilico.pl/modernaserver>

Wszystkie trzy sposoby użycia programu ModeRNA dają użytkownikowi różny zakres kompetencji. Wiersz poleceń pozwoli na modelowanie automatyczne dużego zbioru struktur. Interfejs programistyczny posiada funkcje do zaawansowanego redagowania modelu. Serwer umożliwi intuicyjne użycie części funkcji.



Rycina 24. Różne tryby korzystania z program ModeRNA na przykładzie najważniejszych operacji.

Podsumowując, program ModeRNA ma trzy tryby użycia. W zależności od złożoności przypadku modelowania użytkownik może wybrać, który będzie dla niego najszybszy i najwygodniejszy.

4.3.1. Użycie programu przez wiersz poleceń

Użycie programu ModeRNA przez wiersz poleceń pozwala na automatyczne zbudowanie modelu oraz wykonanie na nim dodatkowych operacji bez znajomości języka programowania Python. Tryb ten umożliwia operowanie na dużych zbiorach danych, np. został on użyty

do utworzenia 9702 modeli testowych (rozdział 4.5). Wymaga instalacji na komputerze użytkownika języka programowania Python, biblioteki Biopython oraz samego programu ModeRNA dostępnego na stronie:

<http://iimcb.genesilico.pl/moderna/download>

Wykorzystanie tego trybu polega w praktyce na uruchomieniu wiersza poleceń i wpisaniu komendy z wybranymi przez użytkownika opcjami i argumentami (jak poniżej). Dostępne opcje i argumenty przedstawione są poniżej (Tabela 6).

```
python moderna.py <opcja> <argument> <opcja> <argument>
```

Tabela 6. Opcje dostępne w programie ModeRNA przez wiersz poleceń.

Opcja	Argument	Nazwa	Znaczenie
-h	-	<i>help</i> (pomoc)	Wyświetla dostępne opcje.
-a	Plik w formacie FASTA.	<i>alignment</i> (przyrównanie)	Wczytuje przyrównanie z danego pliku FASTA.
-t	Plik w formacie PDB.	<i>template</i> (szablon)	Wczytuje szablon z danego pliku PDB.
-e	-	<i>examine</i> (sprawdź)	Sprawdza szablon na obecność cech mogących zakłócić modelowanie.
-l	-	<i>clean</i> (wyczyść)	Oczyszcza strukturę (usuwa wodę, jony, ligandy, itp.).
-c	Identyfikator łańcucha.	<i>chain</i> (łańcuch)	Wskazuje, który łańcuch ma zostać wczytany z pliku danego z opcją -t lub -s (domyślnie A).
-o	Nazwa pliku.	<i>output</i> (plik wyjściowy)	Nazwa pliku wyjściowego, w którym zostanie zapisany model (domyślnie ModeRNA.out.pdb)
-s	Plik w formacie PDB	<i>structure</i> (struktura)	Wczytuje strukturę z danego pliku.
-m	Kilkuliterowy skrót modyfikacji (np. m2G).	<i>modification</i> (modyfikacja)	Dodaje daną modyfikację do struktury.
-p	Identyfikator reszty.	<i>position</i> (pozycja)	Wskazuje, w której pozycji ma zostać dodana modyfikacja.

Przykładowo komenda, która pozwoli na zbudowanie modelu na podstawie łańcucha B struktury z pliku 1QF6.pdb i przyrównania zapianego w pliku ali.fasta oraz zapisanie go w pliku model.pdb będzie wyglądała następująco:

```
python moderna.py -t 1QF6.pdb -c B -a ali.fasta -o model.pdb
```

Natomiast na zamianę reszty 45 (guanozyny) ze struktury zapisanej w pliku 1EHZ.pdb (łańcuch A) na 2'-O-metyloguanozynę (Gm) pozwoli komenda:

```
python moderna.py -s 1EHZ.pdb -c A -m Gm -p 45
```

W celu wykonania operacji na większym zbiorze danych np. zbudowania modeli na szablonie tRNA^{Phe} z pliku 1EHZ.pdb oraz stu różnych przyrównaniach zgromadzonych w katalogu 1EHZ_ali należy 100 razy wywołać komendę budującą model. Proces ten można zautomatyzować przez napisanie skryptu w dowolnym języku. W języku Python skrypt taki, będzie wyglądał następująco:

```
import os
pliki = os.listdir('1EHZ_ali')
for p in pliki:
    os.system('python moderna.py -t 1EHZ.pdb -c A
              -a 1EHZ_ali/%s' %p)
```

Skrypt zapisany w pliku o dowolnej nazwie np. zbuduj_modele.py można wykonać poprzez wpisanie w wierszu poleceń:

```
python zbuduj_modele.py
```

Podsumowując, jednym z trybów użycia programu ModeRNA to praca w wierszu poleceń. Główną zaletą tego trybu jest możliwość operowania na dużych zbiorach danych.

4.3.2. Użycie programu przez interfejs języka Python

Przez interfejs programistyczny programu ModeRNA użytkownik może kontrolować proces modelowania w każdym szczególe. Oprócz automatycznego budowania modelu, ma możliwość modelowania poszczególnych pozycji w przyrównaniu (rozdział 4.2.4) lub redagowania wybranych fragmentów struktury. Użycie tego trybu wymaga instalacji języka programowania Python, biblioteki Biopython oraz samego programu ModeRNA dostępnego na stronie:

<http://iimcb.genesilico.pl/moderna/download>

Interfejs programistyczny składa się z 45 funkcji (Tabela 7), których można używać interaktywnie, wpisując je z odpowiednimi argumentami do konsoli języka Python. Użytkownik ma wówczas ciągły dostęp do budowanego modelu i w każdym momencie może go poddać ocenie geometrycznej lub zapisać i obejrzeć w przeglądarce graficznej, a następnie zdecydować czy chce zakończyć proces modelowania lub dalej redagować strukturę. Funkcje można również zapisywać w skrypcie języka Python. Stosując to rozwiązanie, wprowadzenie drobnej zmiany do procesu modelowania wymaga ponownego uruchomienia skryptu i wykonania całego procesu jeszcze raz. Skrypty pozwalają jednak zachować raz zastosowaną kombinację funkcji do ponownego użycia.

Tabela 7. Funkcje programu ModeRNA dostępne przez interfejs skryptowy. Na białym tle przedstawione są funkcje związane z modelowaniem. Niebieskie wyróżnienie oznacza funkcje pozwalające na analizę struktury. Zielone tło wskazuje funkcje związane z przygotowaniem danych wejściowych.

Nazwa funkcji	Opis
add_all_modifications	Dodaje do struktury modelu wszystkie modyfikacje obecne w sekwencji celu, a niewystępujące w szablonie.
add_modification	Zmienia wskazaną resztę w określony modyfikowany nukleotyd.
add_pair_to_base	Dodaje parę typu Watson-Crick do pojedynczej reszty nukleotydu, tak żeby utworzyć strukturę dwuniciową.
analize_geometry	Sprawdza, czy długości wiązań oraz wartości kątów płaskich i torsyjnych są zbliżone do wartości referencyjnych uzyskanych z analizy bazy PDB.
apply_alignment	Wprowadza zmiany wynikające z przyrównania. W zbudowanym w ten sposób modelu znajdują się identyczne reszty skopiowane z szablonu, reszty, które uległy mutacji, oraz te, które wymagały wprowadzenia lub usunięcia modyfikacji.
apply_all_indels	Wprowadza do dostarczonego szkieletu modelu wszystkie fragmenty bez szablonu, których obecność wynika z przyrównania.
apply_indel	Szuka fragmentów pasującego pomiędzy wskazane reszty modelu i wstawia najlepiej ocenionego kandydata zmieniając jego sekwencję na zadaną.
apply_missing_ends	Wstawia do modelu obecne w przyrównaniu fragmenty bez szablonu na końcu 5' i 3'.
change_sequence	Zmienia sekwencję struktury na zadaną.

clean_structure	Usuwa ze struktury wodę, jony, aminokwasy, oraz niezidentyfikowane reszty. Zmienia nazwy atomów rybozy z zawierających '*' na zawierające '''. Dodaje brakujące grupy fosforanowe. Sprawdza czy łańcuch jest ciągły.
copy_identical_residues	Kopiuje wszystkie reszty nukleotydów, których zasady są identyczne pomiędzy celem i szablonem do modelu na podstawie przyrównania.
copy_single_residue	Kopiuje pojedynczą resztę nukleotydu do modelu.
copy_some_residues	Kopiuje grupę reszt nukleotydów do modelu.
create_fragment	Tworzy fragment strukturalny o zadanej sekwencji, pasujący dzięki nałożeniu przestrzennemu między wskazane reszty kotwiczące.
create_model	Buduje model na podstawie danego szablonu i przyrównania.
delete_residue	Usuwa ze struktury resztę o zadanym identyfikatorze.
examine_structure	Sprawdza strukturę na obecność cech mogących zakłócić proces modelowania (woda, jony, niezidentyfikowane reszty, stara nomenklatura atomów rybozy, brakujące reszty fosforanowe, nieciągłość łańcucha).
exchange_mismatches	Zmienia w modelu wszystkie standardowe zasady z szablonu na zgodne z przyrównaniem.
exchange_single_base	Wymienia zasadę wskazanej reszty nukleotydu na zadaną.
exchange_some_bases	Wymienia zasady wskazanych reszt nukleotydów na zadane.
extend_helix	Wydłuża helisę od wskazanych nukleotydów na jej końcu 5' i 3' (lub w jej środku) poprzez dodawanie par typu Watson-Crick tworzących helisę typu A o zadanej sekwencji.
find_clashes	Szuka zderzeń sterycznych pomiędzy atomami różnych reszt w strukturze. Autorem tej funkcji jest mgr Tomasz Puton.
find_fragment	Wyszukuje fragmenty pasujące pomiędzy dwie zadane reszty kotwiczące.
find_modifications	Przygotowuje spis wszystkich modyfikowanych reszt nukleotydów obecnych w strukturze.
fix_backbone	Naprawia łańcuch główny cząsteczki tak, aby zachowana została ciągłość pomiędzy resztami nukleotydów. W pierwszej próbie naprawy łańcucha poruszane są tylko fosfor i oddziałujące z nim atomy tlenu. W przypadku, gdy to podejście okaże się

	nieskuteczne używany jest algorytm FCCD. Autorem tej funkcji jest dr Kristian Rother.
get_base_pairs	Zwraca informację o parowaniu zasad w cząsteczce.
get_secstruc	Odczytuje z trójwymiarowych koordynat cząsteczki jej strukturę drugorzędową (nie uwzględnia pseudowęzłów). Używa formatu VIENNA.
get_sequence	Odczytuje z trójwymiarowych koordynat cząsteczki sekwencję.
insert_fragment	Wstawia dany fragment do modelu.
insert_two_strand_fragment	Wstawia dany fragment dwuniciowy pomiędzy dwie pary wskazanych reszt. Wymaga podania 4 reszt kotwiczących z modelu i 4 z fragmentu.
load_alignment	Ładuje do programu przyrównanie z pliku FASTA.
load_model	Ładuje do programu uprzednio zbudowany model lub inną cząsteczkę, którą użytkownik chce poddać modyfikacji. Czyta pliki w formacie PDB.
load_template	Ładuje do programu strukturę szablonu odczytaną z pliku PDB.
match_alignment_with_model	Sprawdza czy sekwencja odczytana z koordynat modelu jest identyczna z pierwszą sekwencją w przyrównaniu.
match_template_with_alignment	Sprawdza, czy sekwencja odczytana z koordynat szablonu jest identyczna z drugą sekwencją w przyrównaniu.
remove_all_modifications	Usuwa wszystkie modyfikacje z danej struktury. Reszty nukleotydów modyfikowanych zostają zastąpione standardowymi, od których pochodzą.
remove_mismatching_modifications	Usuwa wszystkie modyfikowane reszty obecne w szablonie i nieobecne w sekwencji celu zastępując je odpowiadającymi im resztami standardowymi.
remove_modification	Usuwa modyfikację z reszty nukleotydu o wskazanym identyfikatorze zastępując ją standardową, od której pochodzi modyfikacja.
renumber_chain	Zmienia numerację reszt w łańcuchu na ciągłą, rozpoczynając od identyfikatora podanego przez użytkownika i pierwszej reszty w łańcuchu.
rotate_chi	Obraca zasadę danej reszty nukleotydu wokół wiązania N-glikozydowego o zadaną wartość.
shrink_helix	Wycina część helisy pomiędzy czterema wskazanymi resztami i łączy pozostałe dwa fragmenty struktury.
write_fragment_candidates	Zapisuje struktury kandydatów na fragmenty do wstawienia w miejscu bez szablonu (wraz z całym

	modelem) do osobnych plików PDB. Pliki ponumerowane są według dopasowania fragmentów.
write_log_file	Tworzy plik tekstowy z opisem wszystkich operacji wykonanych przez program ModeRNA do momentu zapisania pliku. Informacje te są następnie usuwane z pamięci programu.
write_model	Zapisuje model do pliku PDB.
write_secstruc	Zapisuje strukturę drugorzędową odczytaną z trójwymiarowych współrzędnych położenia atomów cząsteczki do pliku w formacie VIENNA.

Automatyczna budowa modelu poprzez interfejs programistyczny wymaga użycia następujących funkcji:

```
t = load_template('nazwa_pliku.pdb', 'identyfikator_łańcucha')
a = load_alignment('nazwa_pliku.fasta')
m = create_model(t, a)
write_model(m, 'nazwa_pliku_wyjściowego.pdb')
```

Przykład bardziej zaawansowanego modelowania, które można wykonać jedynie przez interfejs programistyczny to użycie dwóch szablonów. Taka procedura została wykorzystana m. in. przy budowie modelu tRNA^{Thr} *E. coli* oddziałującego z syntetazą aminoacylową tRNA (Rother, i wsp., 2011). Oprócz głównego szablonu, struktury 1EHZ, użyty został dodatkowy szablon, struktura 1C0A, do modelowania pętli antykodonowej w odpowiedniej konformacji. Modelowanie to zostało wykonane przez zastosowanie następujących funkcji:

```
t = load_template('1EHZ.pdb', 'A')
a = load_alignment('target_1EHZ.fasta')
m = create_model(t, a)
fragment = create_fragment('fr.pdb', m['30'], m['40'], ch='B'
    seq='AUUCGUEAU')
insert_fragment(m, fragment)
write_model('model.pdb')
```

Przypadek ten został szczegółowo opisany w publikacji „*RNA tertiary structure prediction with ModeRNA*” (Rother, i wsp., 2011). Inne przykłady modelowania możliwe do wykonania

jedynie poprzez interfejs programistyczny to redagowanie struktury drugorzędowej oraz osobne modelowanie poszczególnych pozycji w przyrównaniu.

Podsumowując, interfejs programistyczny programu ModeRNA to 45 funkcji, które pozwalają na indywidualne modelowanie poszczególnych pozycji w przyrównaniu lub modelowanie automatyczne. Jego główną zaletą jest możliwość wykonania zaawansowanych operacji, takich jak modelowanie całych elementów struktury drugorzędowej, czy użycie dwóch szablonów.

4.3.3. Serwer ModeRNA

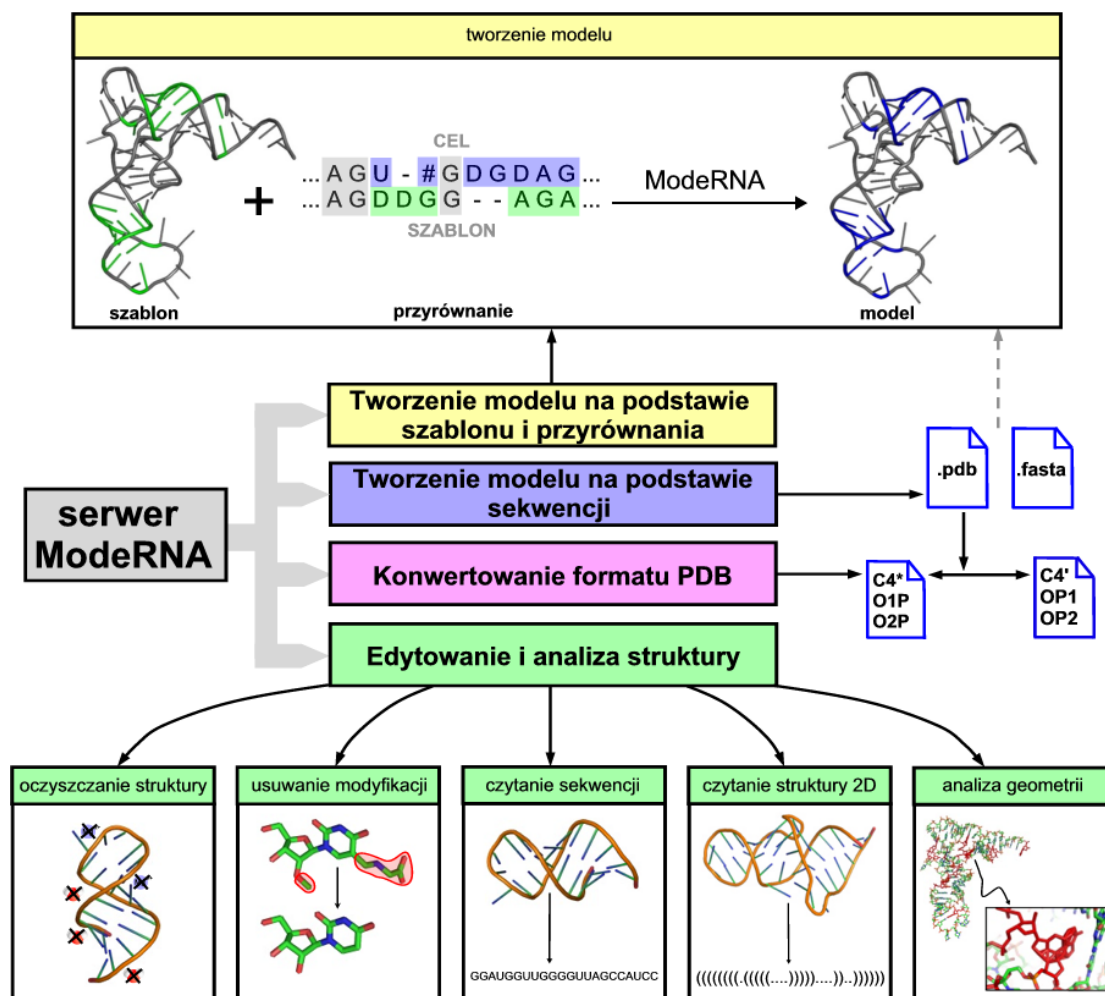
Interfejs internetowy programu ModeRNA pozwala na automatyczne budowanie modelu oraz redagowanie i analizę modelu bez konieczności instalacji programu. Użytkownik wysyła swoje polecenia oraz pliki wejściowe na serwer dostępny pod adresem:

`http://iimcb.genesilico.pl/modernaserver`

Efektom działania serwera jest wyświetlenie strony internetowej, z której można pobrać model oraz wyniki innych analiz. Dodatkowo, w odróżnieniu od wiersza poleceń i interfejsu programistycznego, serwer daje możliwość wyszukiwania szablonu do modelowania, przygotowania przyrównania, oraz konwertowania formatu PDB ze starej notacji na nową i odwrotnie.

Czterem głównym zadaniom, które można realizować przez serwer (Rycina 25) odpowiadają zakładki na stronie serwera:

- Budowa modelu na podstawie szablonu i przyrównania – zakładka: *Build model*.
- Budowa modelu na podstawie sekwencji – zakładki: *Find template*, *Align sequences*.
- Konwertowanie formatu PDB do nowej i starej notacji – zakładka: *Convert format*.
- Analiza struktury przestrzennej cząsteczki RNA – zakładka: *Analyse structure*.



Rycina 25. Operacje dostępne przez serwer ModeRNA (interfejs internetowy programu).

Proces modelowania, który można wywołać przez zakładkę *Build model* przebiega automatycznie (jak w przypadku użycia wiersza poleceń) i użytkownik nie może ingerować w poszczególne operacje. Ma natomiast możliwość oczyszczenia struktury szablону oraz uzyskania podstawowych danych geometrycznych o zbudowanym modelu.

Budowa modelu na podstawie sekwencji celu wymaga znalezienia szablónu oraz przyrównania sekwencji szablónu i sekwencji celu. Zakładka *Find template* pozwala na zidentyfikowanie rodziny, do której należy zadana sekwencja na podstawie sekwencji zgromadzonych w bazie Rfam (Gardner, i wsp., 2009). Następnie rodzina ta sprawdzana jest pod kątem obecności struktur rozwiązanych doświadczalnie. W przypadku dostępności struktury/struktur o rozdzielczości lepszej niż 2,5 Å wykonywane jest przyrównanie pomiędzy sekwencjami celu i szablónu z użyciem modelu kowariancji danej rodziny pobranego z bazy Rfam. W efekcie użytkownik zakładki *Find template* otrzymuje informację

o rodzinie/rodzinach, do których może należeć cząsteczka celu oraz przyrównanie dla dostępnych w wysokiej rozdzielczości struktur. Zarówno struktura szablonu jak i przyrównanie mogą zostać pobrane ze strony serwera. W sytuacji, gdy użytkownik zna strukturę szablonu, a brakującym elementem jest przyrównanie, możliwe jest skorzystanie z zakładki *Align sequences*. Umożliwia ona uzyskanie przyrównania dwóch sekwencji z wykorzystaniem programu R-coffee (Wilm, i wsp., 2008). Dane uzyskane z wykorzystaniem zakładek *Find template* i *Align sequences* mogą stanowić bezpośrednio pliki wejściowe do modelowania automatycznego w zakładce *Build model*.

Uzyskany poprzez serwer model ma format PDB i może zostać użyty jako plik wejściowy do innych programów. W celu ułatwienia dalszej pracy z plikiem zakładka *Convert format* konwertuje plik PDB według wskazań użytkownika. Zmienione mogą zostać:

- Nazwy atomów rybozy (* lub ').
- Nomenklatura nazw atomów grupy fosforanowej (np. OP1 lub O1P).
- Pozycja nazwy reszty.
- Flaga wiersza ze współrzędnymi (ATOM lub HETATM).

Model można również analizować i redagować z wykorzystaniem samego serwera. Zakładka *Analyze structure* pozwala na następujące operacje:

- Oczyszczenie struktury.
- Wyszukanie wszystkich modyfikacji w strukturze.
- Usunięcie wszystkich modyfikacji ze struktury (zastąpienie ich niemodyfikowanymi odpowiednikami).
- Zmianę sekwencji struktury.
- Zmianę numeracji reszt w strukturze.
- Odczytanie sekwencji ze struktury.
- Odczytanie struktury drugorzędowej.
- Analizę geometrii.

Powyższe operacje są również dostępne przez interfejs programistyczny. Zostały one szczegółowo opisane w rozdziale 4.2.

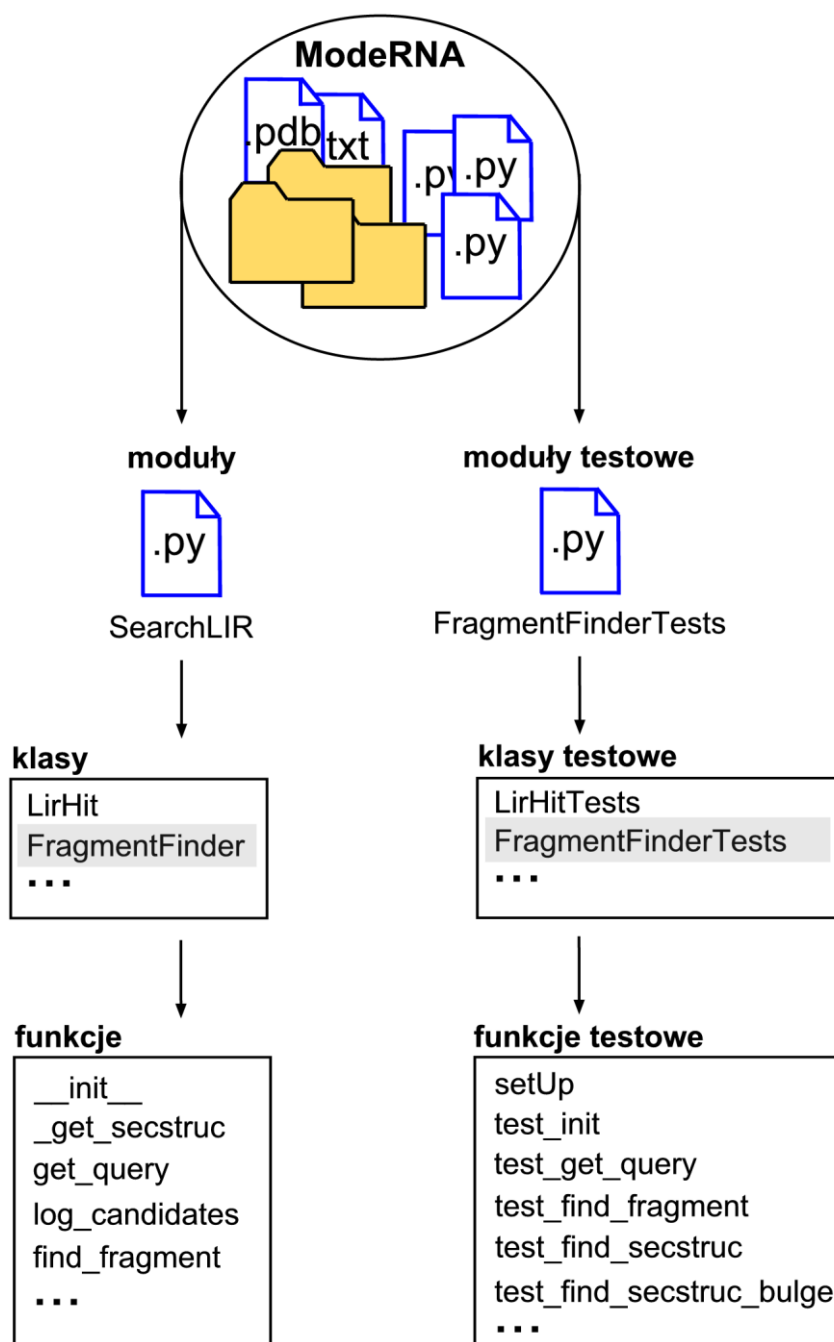
Podsumowując, serwer jest interfejsem programu ModeRNA, który nie wymaga instalacji. Umożliwia automatyczne budowanie modelu oraz operacje analizy i redagowania struktur trzeciorzędowych RNA. Dodatkowo pozwala na wyszukiwanie szablonu do modelowania oraz przyrównanie dwóch sekwencji.

Interfejs internetowy z serwerem Django został zaimplementowany przez mgr Kaję Milanowską. Mgr Magdalena Rother jest autorką funkcji programu ModeRNA podłączanych do interfejsu oraz zaplanowała zewnętrzną architekturę serwera.

4.4. Architektura programu ModeRNA

Program ModeRNA zaimplementowany został zgodnie z paradygmatem programowania obiektowego. Program działa przez instancje klas, które wykonują określone zadania, mogą się między sobą komunikować i zmieniać swój stan. Wykorzystywane są również takie cechy programowania obiektowego jak dziedziczenie i polimorfizm.

Fizycznie program pobierany jest na dysk użytkownika jako skompresowany plik `moderna_source_wersja.programu.tar.gz`, o rozmiarze 13 MB. Katalog otrzymany po rozpakowaniu pliku zawiera bibliotekę `moderna`, testy technicznej poprawności oraz pliki tekstowe z licencją programu (*GNU General Public License*), ogólnymi informacjami na temat programu i jego instalacji oraz opisem charakterystycznych cech pobranej wersji. Zarówno część funkcjonalna jak i testowa składa się z kilku pakietów (katalogów), w których zawarte są moduły i dane tekstowe. W modułach (skrypty języka Python posiadające rozszerzenie `.py`) zaimplementowane są obiekty (klasy) programu ModeRNA. Moduły funkcjonalne i testowe wzajemnie ze sobą korespondują. Zawierają one funkcje, które pozwalają obiektom na wykonywanie różnych czynności. Architekturę fizyczna programu przedstawia Rycina 26.



Rycina 26. Architektura programu ModeRNA. Biblioteka zawiera oprócz pakietów (katalogi) i modułów (pliki .py) pliki z danymi w formacie tekstowym i PDB. Po lewej stronie został przedstawiony przykładowy moduł (SearchLIR) i zawarte w nim klasy i funkcje klas. Po prawej stronie znajduje się odpowiadający mu moduł testowy.

Część funkcjonalna programu to pięć pakietów, w którym zgromadzone są 62 moduły. Liczba klas to 117. Zawierają one 761 funkcji (Tabela 8). Jeden moduł odpowiada przeważnie jednej klasie. Najwięcej klas (27) zgromadzonych zostało w module `Errors.py`, który przechowuje wszystkie rodzaje błędów właściwych dla programu ModeRNA. Przykładowo, pozwala wyświetlić inny rodzaj błędu, kiedy użytkownik poda błędny identyfikator łańcucha, a inny kiedy poda błędny identyfikator reszty. Funkcje są najliczniej

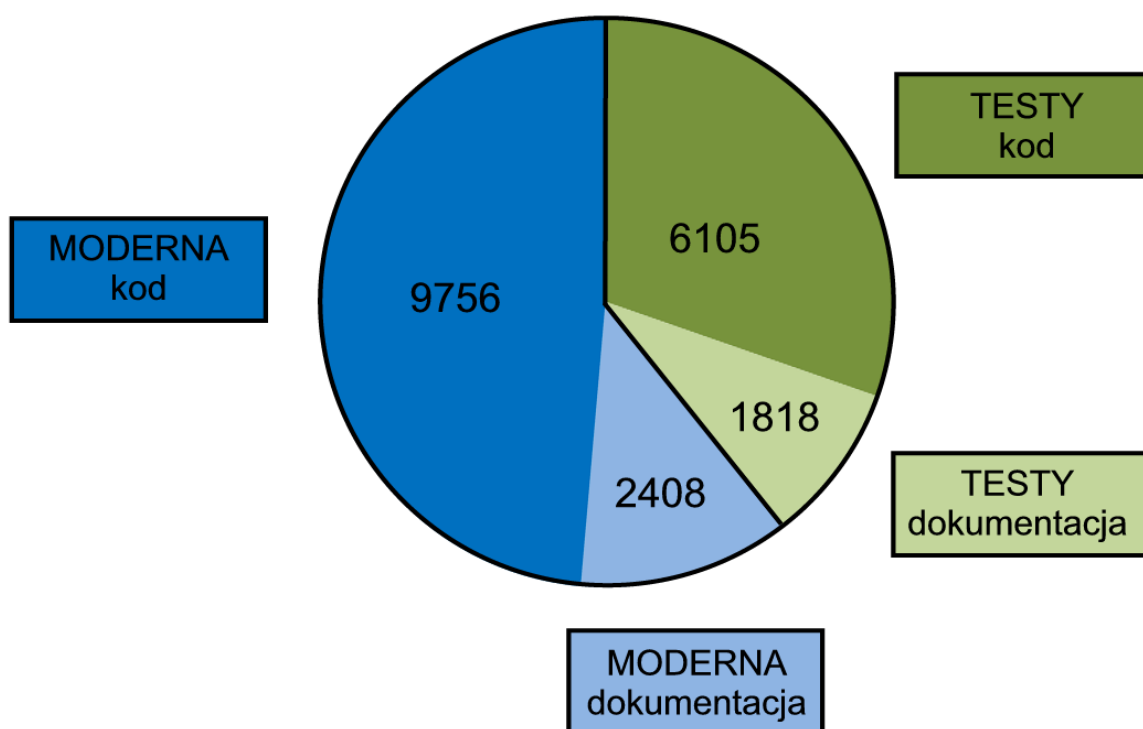
reprezentowane w module `ModernaFragment.py`. Jest to związane z dużą złożonością operacji wstawienia fragmentu. Łącznie część funkcjonalna programu ModeRNA to 14235 linii kodu, dokumentacji i komentarzy.

Tabela 8. Moduły programu ModeRNA (pliki `.py`) z liczbą importowanych bibliotek, liczbą klas i funkcji, liczbą wszystkich linii kodu (jako kod rozumiane są również linie importujące biblioteki i definiujące klasy i funkcje) i dokumentacji, a także całkowitą liczbą linii (zawiera również puste linie oddzielające poszczególne funkcje). Wyróżnione zostały najwyższe wartości dla głównego pakietu programu. Moduły testowe zostały przedstawione zbiorczo. Na dole tabeli znajduje się podsumowanie całego programu z uwzględnieniem testów.

Moduł (nazwa pliku)	Importowane biblioteki	Klasy	Funkcje	Linie kodu	Dokumentacja	Całkowita liczba linii
Pakiet moderna						
<code>AlignmentMatcher.py</code>	5	2	15	123	30	175
<code>CheckPdb.py</code>	10	1	22	339	66	491
<code>Constants.py</code>	4	0	0	71	38	144
<code>Errors.py</code>	0	27	0	46	12	92
<code>FragmentInsertion.py</code>	0	1	2	21	12	36
<code>Helix.py</code>	8	3	12	133	30	180
<code>IsostericModelling.py</code>	8	1	5	67	22	109
<code>Isostericity.py</code>	9	1	9	168	39	256
<code>IsostericityMatrices.py</code>	2	1	4	55	17	82
<code>LIR.py</code>	8	2	14	137	74	243
<code>LIRdb.py</code>	7	2	13	198	41	272
<code>LogFile.py</code>	2	1	9	59	29	114
<code>ModernaAlignment.py</code>	5	2	14	191	61	292
<code>ModernaAlphabet.py</code>	2	2	13	121	20	168
<code>ModernaFragment.py</code>	6	6	48	266	81	406
<code>ModernaResidue.py</code>	15	1	17	323	54	431
<code>ModernaSequence.py</code>	4	1	15	135	26	182
<code>ModernaStructure.py</code>	8	1	22	235	68	347
<code>ModernaSuperimposer.py</code>	2	1	4	59	17	87
<code>PdbConverter.py</code>	10	4	19	169	55	272
<code>RNAAlignment.py</code>	4	3	42	371	68	504
<code>RNAChain.py</code>	11	1	23	207	48	294
<code>RNAModel.py</code>	14	1	30	281	83	441
<code>RNAResidue.py</code>	12	1	33	334	67	458
<code>Renumerator.py</code>	2	1	9	113	24	154
<code>SearchLIR.py</code>	21	5	43	441	97	634
<code>SecstrucFragment.py</code>	6	5	30	190	56	294
<code>StructureLibrary.py</code>	3	2	7	87	25	126
<code>Template.py</code>	1	1	2	32	14	50
<code>__init__.py</code>	1	0	0	18	3	28
<code>commands.py</code>	24	0	46	785	118	1161

Moduł (nazwa pliku)	Importowane biblioteki	Klasy	Funkcje	Linie kodu	Dokumentacja	Całkowita liczba linii
decorators.py	3	0	4	43	17	68
moderna_cli.py	4	0	2	71	17	115
setup.py	4	0	1	67	12	90
validators.py	11	0	16	105	24	153
Pakiet moderna.analyze						
BasePairCalculator.py	3	1	11	134	23	184
BaseRecognizer.py	6	2	9	154	27	211
ClashRecognizer.py	8	1	11	137	51	213
Constants.py	1	0	0	118	34	180
GeometryAnalyzer.py	3	1	9	71	10	96
GeometryParameters.py	0	1	3	168	21	199
GeometryStatistics.py	4	6	29	271	54	379
HBondCalculator.py	7	3	17	124	35	184
MolGraphParser.py	2	3	16	197	38	269
MolParameters.py	0	0	0	216	49	282
MolTopologies.py	1	2	7	460	50	549
MolecularGraph.py	5	2	25	526	142	753
PuckerCalculator.py	2	2	9	74	18	109
RNASuites.py	0	0	0	55	12	70
StackingCalculator.py	5	3	11	161	43	232
__init__.py	0	0	0	0	0	0
topology_graph.py	1	1	4	48	16	71
Pakiet moderna.builder						
BackboneBuilder.py	8	1	10	141	22	182
Constants.py	0	0	0	7	8	15
CoordBuilder.py	3	0	2	44	18	74
FCCDLoopCloser.py	3	2	9	103	33	167
PhosphateBuilder.py	12	2	17	235	49	322
__init__.py	0	0	0	0	0	0
Pakiet moderna.examples						
__init__.py	0	0	0	0	0	0
usage_examples.py	0	0	0	11	138	206
Pakiet moderna.minimize						
__init__.py	0	0	0	0	0	0
refine_model_mmtk.py	16	4	17	240	52	339
Podsumowanie moderna	353	117	761	9756	2408	14235
Wszystkie moduły testowe	433	81	702	6105	1818	9200
Podsumowanie	786	198	1463	15861	4226	23435

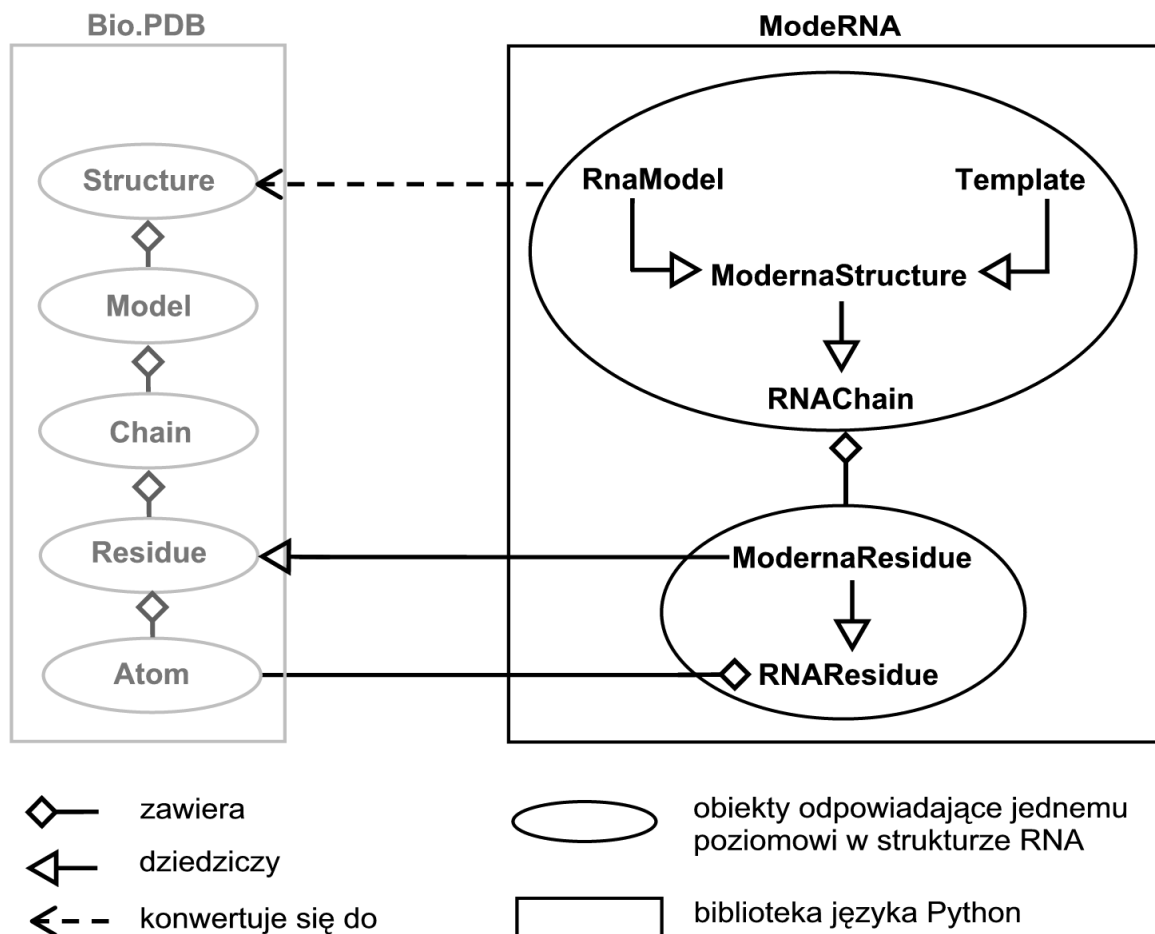
Dużą część programu stanowią moduły testowe (Rycina 27). Zabezpieczają one poprawność techniczną programu. Są to testy jednostkowe oraz akceptacyjne, sprawdzające czy poszczególne elementy (moduły, funkcje, operacje) programu działają poprawnie. W większości korespondują one z funkcjami i modułami części funkcjonalnej. 117 modułom części funkcjonalnej odpowiada 81 modułów testowych, a 761 funkcjom z modułów funkcjonalnych 702 funkcje testowe. **Część testowa programu została zaimplementowana przez dr Kristiana Rothera.**



Rycina 27. Liczba linii kodu i dokumentacji programu ModeRNA oraz jego części testowej. Przez dokumentację rozumiane są opisy poszczególnych funkcji i klas (*docstrings*) oraz komentarze wewnątrz kodu (linie rozpoczynające się znakiem '#').

Najważniejsze obiekty programu służą przechowywaniu informacji o strukturze przestrzennej cząsteczki RNA. Model reprezentowania tej informacji jest wzorowany na bibliotece `Bio.PDB` (Rycina 28) (rozdział 3.2.3). Osobne obiekty służą do przechowywania danych o atomie, reszcie nukleotydu i całej strukturze. Występuje hierarchia: struktura zawiera reszty, reszty zawierają atomy. Do operowania atomami program ModeRNA używa bezpośrednio obiektów `Bio.PDB.Atom`. Reszcie nukleotydu odpowiadają dwa obiekty. Obiekt `RNAResidue` dziedziczy po obiekcie `Bio.PDB.Residue`, lecz został dodatkowo wyposażony w cechy ułatwiające pracę z RNA, np. przechowywanie informacji o nazwie w różnych nomenklaturach.

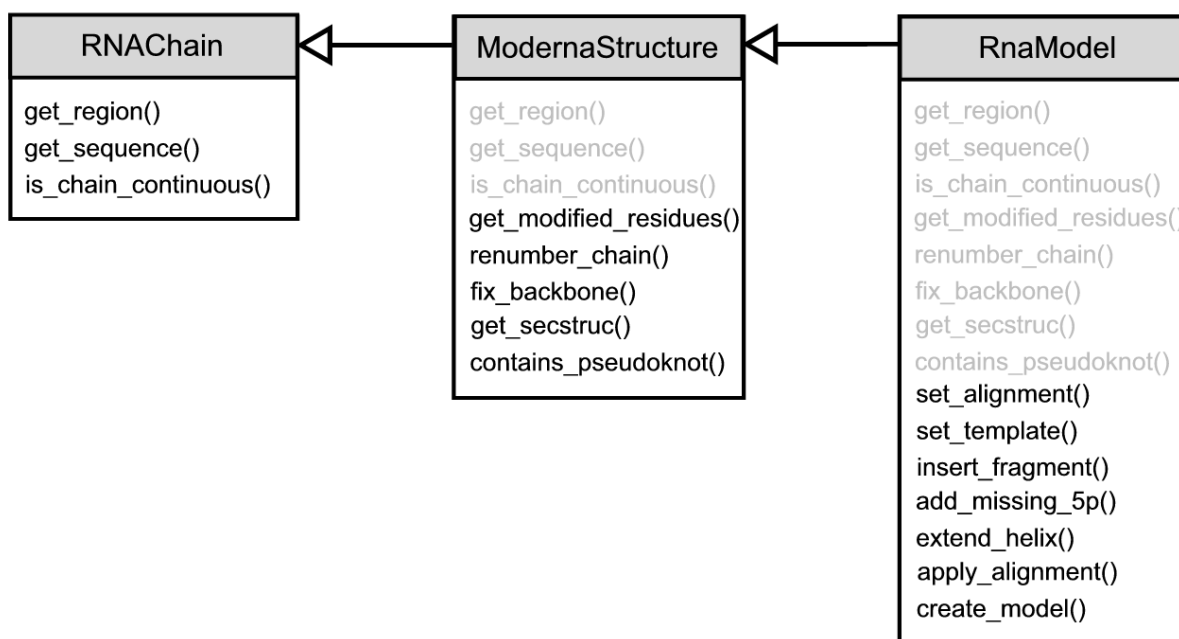
Obiekt `ModernaResidue` posiada te same możliwości, co `RNAResidue` (dziedziczy po obiekcie `RNAResidue`) i dodatkowo ma funkcje umożliwiające modelowanie, np. dodawanie lub usuwanie modyfikacji. W praktyce podczas modelowania użytkownik ma do czynienia zawsze z obiektami `ModernaResidue`. Podział na `RNAResidue` i `ModernaResidue` został wprowadzony, by nie tworzyć zbyt dużych obiektów. Średnia długość klasy w programie `ModeRNA` to 300 linii, dzięki czemu kod jest przejrzysty i czytelny. Obiekty dedykowane całej strukturze zostały podzielone według zadań, do których są przeznaczone.



Rycina 28. Hierarchiczna struktura obiektów reprezentujących model w programie `ModeRNA`. Po lewej stronie przedstawione zostały zależności pomiędzy obiektami w pakiecie `Bio.PDB`, po prawej zależności między głównymi obiektami biblioteki `ModeRNA`. Schemat wykorzystuje elementy języka UML.

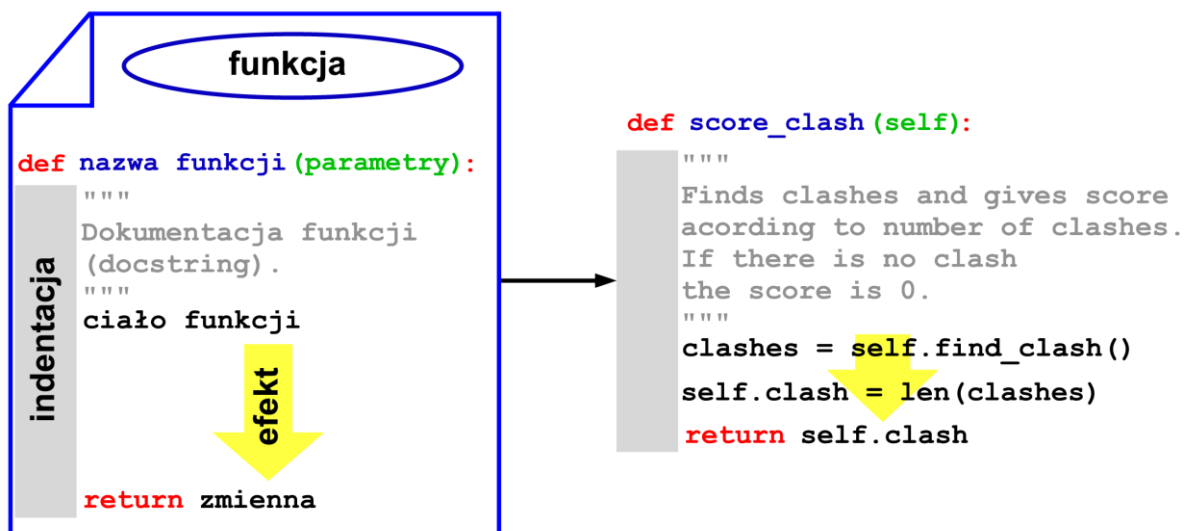
Podstawowym obiektem struktury RNA jest `RNACHain` (Rycina 29). Zawiera on funkcje związane z analizą struktury. Obiekt `ModernaStructure` rozszerza go o więcej funkcji do analizy oraz funkcje pozwalające na redagowanie struktury. Obiekt `RNAModel` to obiekt poświęcony modelowaniu. Obiekt `Template` odpowiada za szablon.

Dostosowuje on numerację jego reszt do przyrównania tak, aby ułatwić następnie proces modelowania. Wszystkie obiekty związane ze strukturą przeznaczone są do przechowywania jednego łańcucha. Pod tym względem program ModeRNA ma uproszczoną reprezentację w stosunku do biblioteki `Bio.PDB`, która została zaprojektowana z myślą o danych doświadczalnych. Obiekt struktury w `Bio.PDB` może zawierać nie tylko kilka łańcuchów, ale również kilka modeli (np. z doświadczenia NMR).



Rycina 29. Obiekty reprezentujące strukturę w programie ModeRNA na różnym poziomie złożoności i charakterystyczne dla nich funkcje. Wyróżniona nazwa w każdej ramce oznacza rodzaj obiektu. Funkcje wypisane na szaro nie są zaimplementowane w danym obiekcie, ale dziedziczone. Strzałki oznaczają zależność dziedziczenia - grot skierowany jest w stronę obiektu, który jest dziedziczony. Schemat wykorzystuje elementy języka UML.

Funkcje dostępne przez obiekty programu ModeRNA umożliwiają przede wszystkim działania na strukturze trzeciorzędowej RNA. Wszystkie funkcje zaimplementowane zostały z myślą o zachowaniu przejrzystości kodu. Nazwa funkcji opisuje czynność, którą ona wykonuje. Jedna funkcja odpowiada jednemu zadaniu. Funkcje są krótkie i zawierają dokumentację (*docstring*), która wyjaśnia działanie danej funkcji (Rycina 30).



Rycina 30. Przykładowa funkcja programu ModeRNA. Po lewej stronie przedstawiona jest ogólna budowa funkcji. Po prawej funkcja `score_clash` należąca do obiektu `LirHit` z modułu `SearchLIR`.

Podsumowując, program ModeRNA to ponad 23 tysiące linii kodu. Jego implementacja reprezentuje paradygmat modelowania obiektowego. Najważniejsze obiekty są związane ze strukturą przestrzenną RNA i mają układ hierarchiczny, podobnie jak w bibliotece `Bio.PDB`. Zarówno obiekty jak i ich funkcje posiadają dokumentację. Autorka dążyła do tego, aby kod był przejrzysty dla innych programistów. Poprawność techniczna programu jest zabezpieczona zbiorem automatycznych testów.

4.5. Modele zbudowane przy użyciu programu ModeRNA

W celu merytorycznego sprawdzenia programu ModeRNA zbudowano modele dla 99 cząsteczek tRNA, które posiadają strukturę rozwiązana doświadczalnie (rozdział 2.1.1). Sekwencja jednej cząsteczki wykorzystywana była jako cel, a pozostałe 98 struktur jako szablony, co pozwoliło na uzyskanie 9702 modeli. Dodatkowo powstał zbiór kontrolny zawierający 99 modeli, dla których sekwencja celu modelowana była na szablonie odpowiadającym ich (własnej) strukturze doświadczalnej. Rodzina tRNA została wybrana do testu ze względu na dużą dostępność struktur rozwiązanych doświadczalnie oraz dużą zmienność lokalną, np. pętli zmiennej i pętli D, ujawniającą się pomimo silnej konserwacji globalnej struktury tej rodziny. Ponadto cząsteczki tRNA zawierają liczne modyfikacje, co pozwoliło na przetestowanie operacji modelowania modyfikowanych reszt nukleotydów. Przyrównania dwóch sekwencji – celu i szablonu, dla wszystkich par tRNA zostały uzyskane z bazy Rfam (Gardner, i wsp., 2009) (rozdział 2.2). Do modelowania sekwencji

nieposiadającej odpowiednika w szablonie (insercji/delecji) użyta została biblioteka fragmentów, z której usunięte zostały wszystkie fragmenty pochodzące z tRNA.

Modele budowane były automatycznie z wykorzystaniem interfejsu wiersza poleceń. Proces modelowania zajął 48 godzin na komputerze wyposażonym w procesor 2GHz Intel Xeon. 9675 modeli uzyskało kompletną strukturę i sekwencję zgodną z sekwencją celu. Pozostałych 126 modeli było niekompletnych i nie podlegało dalszej ocenie. We wszystkich przypadkach niekompletnych modeli modelowanie dotyczyło struktury tRNA^{Tyr}(GUA) w kompleksie z syntetazą tyrozylo-tRNA (struktura 1H3E) jako celu lub jako szablonu. W strukturze tej reszty 16 i 47D, oznaczone są jako niemodyfikowana cytydyna i urydyna, nie mają jednak współrzędnych atomowych dla zasad. Przez to, miejsce to nie może być modelowane automatycznie. Kompletny model mógłby zostać uzyskany przez użycie interfejsu programistycznego i wprowadzenie „ręcznych” poprawek do przyrównania. Takie podejście pozwoliłoby również na poprawę jakości części modeli. Test przeprowadzony na grupie struktur tRNA miał jednak na celu sprawdzenie działania programu ModeRNA, a nie wiedzy eksperckiej na temat tRNA i w związku z tym używany był wyłącznie interfejs linii poleceń i modelowanie automatyczne.

Jakość modeli została oceniona przez porównanie ich ze strukturami rozwiązanymi doświadczalnie. Zastosowane zostały następujące kryteria:

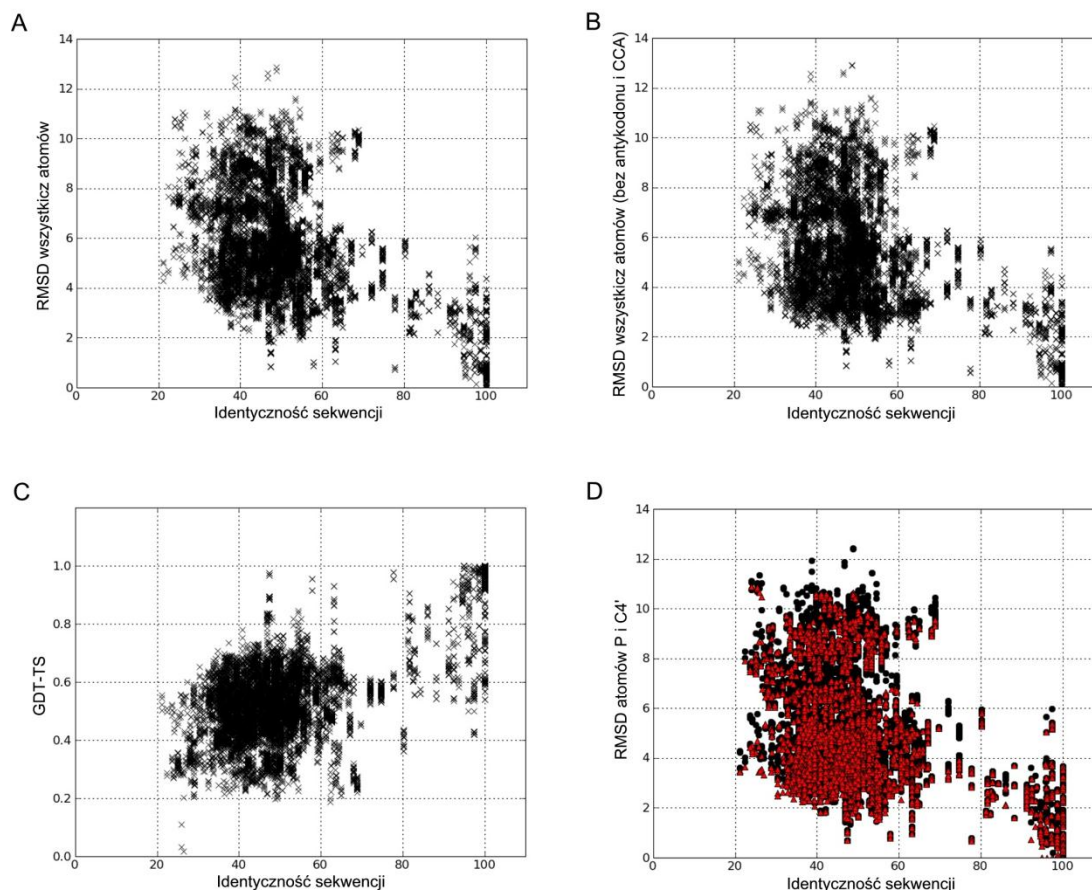
- RMSD pomiędzy wszystkimi atomami modelu i struktury rozwiązanej doświadczalnie.
- RMSD pomiędzy atomami P i C4'.
- RMSD pomiędzy wszystkimi atomami oprócz regionów antykodonu i ramienia akceptorowego.
- RMSD pomiędzy atomami P i C4' z wyłączeniem regionów antykodonu i ramienia akceptorowego.
- GDT-TS - globalny test dystansów (*Global Distance Test – Total Score*) (Cozzetto, i wsp., 2007).
- DI - wskaźnik zniekształcenia (*Deformation Index*) (Parisien, i wsp., 2009).

- Średnia wartość DP - profilu zniekształcenia (*Deformation Profile*) (Parisien, i wsp., 2009).

Średnie wartości uzyskane dla 9675 modeli, według powyższych kryteriów przedstawia Tabela 9. Średnie RMSD wszystkich atomów modeli względem struktur rozwiązanych doświadczalnie wynosi 5,6 Å (Rycina 31). Jeśli brać pod uwagę tylko atomy P i C4' to średnie RMSD wynosi 5,2 Å. Podobny stopień różnorodności występuje w strukturach natywnych – średnie RMSD dla wszystkich par szablon-szablon wynosi 4,9 Å. Rycina 31 pokazuje, że najczęściej model jest w przybliżeniu tak podobny do struktury natywnej jak szablon użyty do modelowania, co jest cechą typową dla modelowania homologicznego. Warto jednak podkreślić, że dla 2135 przypadków modele miały RMSD równe lub niższe niż struktura szablonu.

Tabela 9. Średnie wyniki jakości 9675 modeli tRNA zbudowanych automatycznie przez program ModeRNA w odniesieniu do struktur rozwiązanych doświadczalnie.

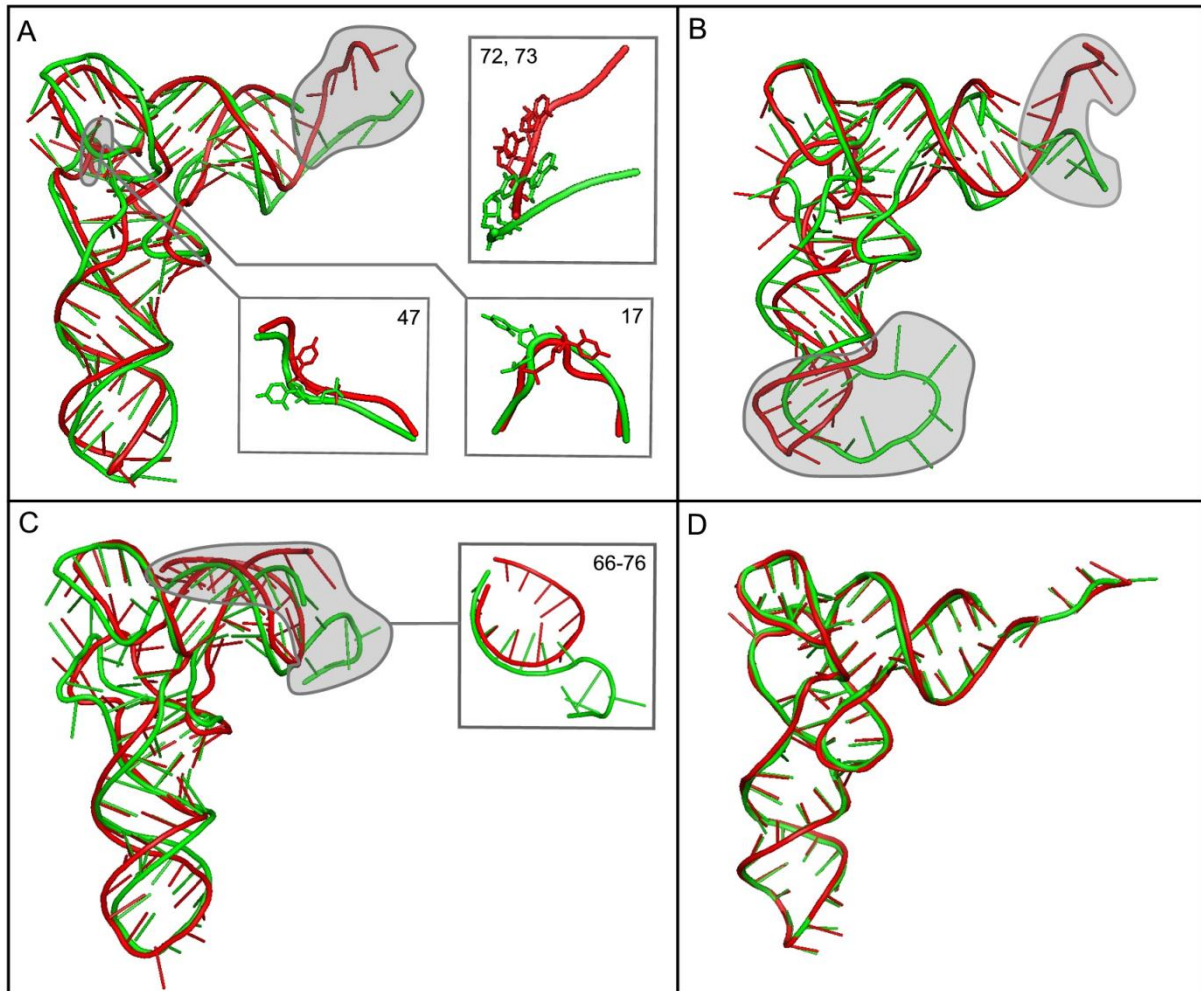
Kryterium	Wynik
Identyczność sekwencji pomiędzy celem i szablonem	51%
Pełnoatomowe RMSD [Å]	5,60
RMSD łańcucha głównego (atomy P i C4') [Å]	5,23
Pełnoatomowe RMSD bez antykodonu i końca 3' [Å]	4,85
RMSD łańcucha głównego bez antykodonu i końca 3' [Å]	4,48
Ocena GDT-TS	0,53
DI	0,62
Średnie DP	13,82



Rycina 31. Ewaluacja modeli tRNA, relacja pomiędzy idencjonalnością sekwencji celu i szablonu, a: A) RMSD wszystkich atomów w odniesieniu do struktur rozwiązanych doświadczalnie; B) RMSD wszystkich atomów wykluczając reszty antykodonu oraz regionu CCA na końcu 3' w odniesieniu do struktur rozwiązanych doświadczalnie C) GDT-TS; D) RMSD atomów P i C4' modeli w odniesieniu do struktur rozwiązanych doświadczalnie (na czarno), oraz struktur celu rozwiązanych doświadczalnie w odniesieniu do struktur szablonów (na czerwono). Na podstawie (Rother, i wsp., 2011).

Ramię akceptorowe oraz pętla antykodonu są regionami o nieszytywnej strukturze i łatwo zmieniają konformację podczas tworzenia kompleksów z białkami (Yang, i wsp., 2006). Można to zaobserwować na przykładzie modelu związanego do rybosomu tRNA^{Phe} *E. coli* (kod PDB: 2J00, łańcuch W) zbudowanego na szablonie 2HGP (łańcuch D), gdzie rybosom jest w stanie poinicjacyjnym (Rycina 32). RMSD modelu wynosi 3,6 Å, pomimo, że sekwencja obu tRNA jest idencjonalna. Wysokie RMSD jest spowodowane przede wszystkim przez reszty 17 (RMSD = 8,5 Å) i 47 (RMSD = 6,7 Å), których zasady mają różną orientację w modelu i w strukturze natywnej, oraz reszty 72 i 73 poprzedzające koniec 3' CCA (Rycina 32 A). Podobna obserwacja dotyczy modelu tRNA^{Thr} *E. coli* zbudowanego na szablonie tRNA^{Phe}, pochodzącym również z *E. coli* (kod PDB: 1B23, łańcuch R) (Rycina 32 B). Struktura natywna tRNA^{Thr} (kod PDB: 1QF6, łańcuch B) oddziałuje z syntetazą tRNA, podczas gdy szablon jest w kontakcie z czynnikiem

elongacji translacji (EF-Tu), powodując wzajemne przesunięcie ramienia akceptorowego i pętli antykodonu o kilka Å. To ilustruje, że dokładność modelowania porównawczego tRNA (oraz innych rodzin RNA) jest ograniczona odpowiednim doбором szablonu we właściwym stanie fizjologicznym.



Rycina 32. Przykłady modeli tRNA zbudowanych przy użyciu programu ModeRNA. Model pokazany jest na czerwono i nałożony na strukturę rozwiązana doświadczalnie (w kolorze zielonym). A) model tRNA^{Phe} *E. coli* (kod PDB struktury natywnej: 2J00, łańcuch W) zbudowany na podstawie szablonu tRNA^{Phe} *E. coli* (kod PDB: 2HGP, łańcuch D). Pomimo 100% identyczności sekwencji celu i szablonu, RMSD modelu i struktury natywnej wynosi 3,61 Å. Reszty, które najsilniej przyczyniają się do wysokiego RMSD zostały zaznaczone na szaro i pokazane w reprezentacji pełnoatomowej; B) model tRNA^{Thr} *E. coli* (kod PDB struktury natywnej: 1QF6, łańcuch B) zbudowany na szablonie tRNA^{Cys} *E. coli* (kod PDB: 1B23, łańcuch R). Struktura natywna oddziałuje z syntetazą tRNA^{Thr}, natomiast szablon z czynnikiem elongacji translacji EF-Tu, co powoduje inną konformację ramienia akceptorowego i pętli antykodonowej – oba regiony zostały zaznaczone na szaro; C) model tRNA^{Glu} (kod PDB struktury natywnej: 2DXI, łańcuch C) zbudowany na podstawie szablonu 2DET. Sekwencje celu i szablonu mają wysokie podobieństwo (72%), jednak RMSD modelu w odniesieniu do struktury rozwiązanej doświadczalnie wynosi 8,05 Å. Główną przyczyną wysokiego RMSD jest brak sześciu nukleotydów na końcu 3' szablonu, które w efekcie zostały wymodelowane w innej konformacji niż w strukturze natywnej; D) model tRNA^{Met} *E. coli* (kod PDB struktury natywnej: 2HGI, łańcuch C) zbudowany na podstawie szablonu tRNA^{Phe} *E. coli* (kod PDB: 2B64, łańcuch V). Model w odniesieniu do struktury rozwiązanej doświadczalnie ma niskie RMSD (1,38 Å), pomimo niezbyt wysokiego podobieństwa sekwencji celu i szablonu (47%). Na podstawie (Rother, i wsp., 2011).

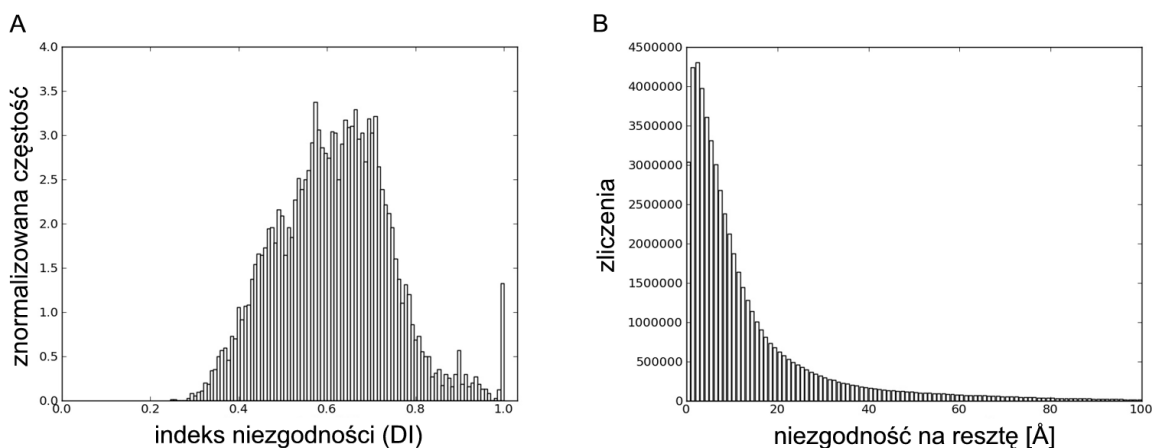
W celu otrzymania wyników bez uwzględniania najbardziej zmiennych regionów, podczas obliczeń pominiętych zostało pięć reszt z antykodononu: 33-37 lub 38-42 dla struktur 2V0G (łańcuch F), 2BTE (łańcuch B i E), 2V0G (łańcuch B), 2BYT (łańcuch E) i 2BUT (łańcuch B). Dla końca 3' usunięto reszty CCA. Po wycięciu tych regionów średnie RMSD pomiędzy modelami i strukturami natywnymi spadło do 4,9 Å (Rycina 31).

Analiza wyników wskazuje, zgodnie z oczekiwaniami, że wysokie podobieństwo sekwencji koreluje z niskim RMSD (Rycina 31). Niektóre modele tRNA mają wysoką jakość pomimo niskiego podobieństwa sekwencji celu i użytego szablonu. Przykładowo model struktury 2HGI (łańcuch C) na bazie szablonu 2B64 (łańcuch V) (Rycina 32 D). Istnieje również grupa modeli, która wykazuje wysokie RMSD w stosunku do struktur natywnych pomimo wysokiego podobieństwa sekwencji celu i szablonu. Jest to spowodowane głównie przez różne konformacje tej samej cząsteczki znajdującej się w odmiennych kompleksach białko-RNA. Podczas analizy wykryto jednak przypadki takie jak model struktury 2DXI (łańcuch C) – Rycina 32 C, zbudowany na szablonie 2DET (łańcuch C) (podobieństwo sekwencji 72%), gdzie RMSD modelu względem struktury natywnej jest bardzo wysokie i wynosi 8.05 Å. Przyczyną jest sześć reszt na końcu 3', które były modelowane *de novo* ze względu na brak odpowiednika w szablonie. Program ModeRNA dobudował w tym miejscu fragment helikalny, podczas gdy w rzeczywistości ma on inną konformację.

Pod względem kryterium GDT-TS (rozdział 3.3.2), średnia ocena modeli wyniosła 0.50. Szczegółowe wyniki przedstawia Rycina 31.

W ocenie uwzględnione zostały również kryteria DI (rozdział 3.3.3) i średnie DP (rozdział 3.3.4), miary dedykowane modelom kwasów rybonukleinowych, wprowadzone przez Parisien i wsp. (Parisien, i wsp., 2009). Dla zbioru modeli tRNA średnie DI wyniosło 0,62, a średnie DP 13,82 (Rycina 33). Większość modeli uzyskała wartość DP w zakresie 0,5-0,8 (Rycina 33 A). Niska średnia jest spowodowana głównie przez zmianę międzycząsteczkowych oddziaływań w różnych stanach funkcjonalnych tRNA (np. wychylenie zasad w kierunku białkowego partnera w tworzonym kompleksie). Uzyskane wartości średnie i maksymalne DP są wyższe niż opisane w artykule Parisien i wsp. (Parisien, i wsp., 2009). Wynika to z faktu, iż DP zależy od rozmiaru badanej cząsteczki, a tRNA jest większy niż badane przez Parisien i wsp. przykłady. Ponadto konformacja i oddziaływania w antykodonie i ramieniu akceptorowym ulegają dużym zmianom (zależnie od stanu funkcjonalnego i oddziaływania z innymi cząsteczkami). W strukturach dopasowanych

na podstawie reszt z tych regionów, pozostałe reszty mogą ulegać przesunięciu i prowadzić do globalnego RMSD o wartości nawet 50 Å. Większość reszt generuje dopasowanie z wartością RMSD około 5,0 Å.



Rycina 33. Ewaluacja modeli tRNA: A) wartości DI; B) wartości DP. Na podstawie (Rother, i wsp., 2011).

Podsumowując, merytoryczna poprawność działania programu ModeRNA została przetestowana na zbiorze 99 tRNA, dla których zbudowano 9801 modele. Jakość modeli została oceniona przez porównanie ich do struktur natywnych przy użyciu kilku niezależnych kryteriów. Pod względem kryterium RMSD modele pokazywały zmienność typową dla obecnej w samym zbiorze 99 tRNA.

Modele tRNA zostały również opisane w publikacji „*ModeRNA: A tool for comparative modeling of RNA 3D structure*” (Rother, i wsp., 2011).

4.6. Program MARS do ewaluacji jakości modeli RNA

Testy merytoryczne programu ModeRNA spowodowały powstanie dużej liczby modeli. Ich jakość była oceniana przez porównywanie ze strukturami natywnymi według kilku kryteriów. W celu automatyzacji procesu ewaluacji powstał program MARS (*Model Assessment of RNA Structures*). Obecna wersja wraz z dokumentacją jest dostępna pod licencją GPL na żądanie od autorki.

Program MARS pozwala na porównanie podobieństwa dwóch struktur z wykorzystaniem różnych miar, a także na porównywanie całych zbiorów, tak że wszystkie struktury jednego zbioru (katalogu) zostaną wykorzystane jako struktury referencyjne do oceny każdej struktury z drugiego zbioru. Kryteria użyte do porównania to RMSD, GDT-TS (Cozzetto, i wsp., 2007), DI i DP (Parisien, i wsp., 2009). Użytkownik ma też możliwość indywidualizacji analizy:

- Porównania tylko wybranego fragmentu struktur (np. miejsca aktywnego).
- Wykluczenie części struktury z analizy (np. pętli antykodonu w tRNA).
- Wyboru rodzaju atomów, które będą wykorzystane do obliczeń (np. tylko atomy P i C4').
- Wyboru rodzaju nałożenia przestrzennego struktur poddawanych analizie (np. już istniejące nałożenie, nałożenie z wykorzystaniem wszystkich atomów, nałożenie według atomów P i C4', nałożenie według wybranego fragmentu, lub inne nałożenie zdefiniowane przez użytkownika).

Ponadto program kontroluje, czy model ma identyczny skład atomów jak struktura referencyjna i generuje raport o ewentualnych różnicach. Inną właściwością jest liczenie RMSD dla pojedynczych par reszt.

Program MARS ma interfejs programistyczny, który może być używany przez wiersz poleceń języka Python lub skrypty przygotowane przez użytkownika. W tym celu wymagana jest instalacja programu RNAView, ModeRNA oraz samej biblioteki MARS. Przykładowo, jeśli użytkownik jest zainteresowany porównaniem dwóch struktur o identycznym składzie atomowym według kryterium GDT-TS może użyć komendy:

```
calc_gdtts(ref=reference_structure, model=second_structure)
```

Bardziej skomplikowany przykład to wyliczenie wartości RMSD dla dwóch struktur, które różnią się składem atomowym z zachowaniem bieżącego nałożenia przestrzennego. Operację taką można wykonać komendą:

```
calc_rmsd(ref=str1, model=str2, do_superposition=False,
          strict=False, atoms = ['P', "C4'"])
```

W czasie obliczeń pod uwagę zostaną wzięte jedynie atomy P i C4', jednak użytkownik może tu zdefiniować dowolne atomy występujące w obu strukturach. Dla porównania dwóch zbiorów struktur należy użyć komendy:

```
compare_sets(ref_set=set1, model_set=set2)
```

Pozwoli ona na obliczenie RMSD pełnoatomowego, RMSD atomów P i C4', GDT-TS, DI oraz DP dla wszystkich kombinacji par struktur z obu zbiorów, a użytkownik otrzyma raport ze wszystkimi wartościami.

Pod względem implementacji program MARS, podobnie jak program ModeRNA, reprezentuje paradygmat programowania obiektowego. Jest jednak dużo mniejszy i zawiera tylko 12 modułów oraz odpowiadające im moduły testowe. Każda z kryteriów oceny podobieństwa reprezentowane jest przez osobny obiekt, wyposażony w funkcje do liczenia cząstkowej i całkowitej wartości oceny. Osobne obiekty poświęcone są zarządzaniu resztami i atomami porównywanych struktur. Odpowiadają one za łączenie odpowiednich reszt i atomów w pary, które następnie ulegną porównaniu. Istnieje również obiekt dedykowany nakładaniu przestrzennemu struktur. Do reprezentacji struktury w przestrzeni i wczytywania plików PDB biblioteka korzysta z obiektów programu ModeRNA.

Podsumowując, program MARS pozwala na automatyczne porównywanie struktur. Oprócz kryteriów oceny podobieństwa, zaimplementowanych zgodnie z opublikowanymi algorytmami RMSD, GDT-TS (Cozzetto, i wsp., 2007), DI i DP (Parisien, i wsp., 2009), program pozwala na identyfikowanie różnic w składzie atomowym struktur oraz na analizowanie wybranych fragmentów struktur. Do dyspozycji użytkownika jest interfejs programistyczny, a implementacja może zostać udostępniona na żądanie.

Funkcja do liczenia punktacji GDT-TS wykorzystuje moduł napisany przez mgr Tomasza Putona. Funkcje do liczenia DI i DP zostały przygotowane przez dr Kristiana Rothera.

5. Dyskusja

Program ModeRNA służy do modelowania struktury przestrzennej RNA z wykorzystaniem podejścia modelowania homologicznego. Dlatego też w dyskusji omawiane są poniższe problemy:

- Porównanie programów przewidujących strukturę RNA oraz przykłady opublikowanych modeli uzyskanych za pomocą różnych metod (rozdział 5.1).
- Kluczowe cechy programu wykorzystującego podejście modelowania porównawczego w kontekście ograniczeń i możliwości programu ModeRNA (rozdział 5.2).
- Wybór szablonu i przygotowanie przyrównania, dwa najpoważniejsze wyzwania modelowania porównawczego (rozdział 5.3).
- Modelowanie regionów bez szablonu i możliwości wykorzystania zewnętrznych baz fragmentów (rozdział 5.4).
- Ewaluacja i porównywanie struktur programem MARS (rozdział 5.5).
- Potencjalne możliwości rozbudowania programu ModeRNA (rozdział 5.6).

5.1. Porównanie programu ModeRNA z innymi metodami

Brak doświadczalnie rozwiązanych struktur dla większości biologicznych sekwencji RNA spowodował rozwój metod obliczeniowych przewidujących strukturę przestrzenną cząsteczki. Wcześniej wzrost ilości informacji zgromadzonej w biologicznych bazach danych dla białek wywołał podobny rozwój narzędzi bioinformatyki strukturalnej dla białek. Powstały liczne programy, opierające się na różnych podstawach (np. podejście *de novo*: ROSETTA (Simons i wsp., 1997), TASSER (Zhang and Skolnick, 2004a), CABS (Kolinski and Bujnicki, 2005); podejście homologiczne: MODELLER (Sali and Blundell, 1993), SWISS-MODEL (Schwede, i wsp., 2003)). Są one testowane co dwa lata podczas eksperymentu CASP (*Critical Assessment of Techniques for Protein Structure Prediction*) (Moult, 2005). Obecnie dyscyplina przewidywania struktury przestrzennej RNA rozwija się również dynamicznie, choć eksperyment oceniający jakość metod bioinformatycznych do przewidywania struktury RNA miał pierwszą edycję dopiero w roku 2011 i to na znacznie mniejszą skalę niż eksperyment CASP (ujawnione zostały sekwencje tylko trzech cząsteczek) (Cruz, i wsp.,

2012). Dlatego też przedstawione poniżej porównanie programów do modelowania RNA opiera się na danych dostępnych w publikacjach.

5.1.1. Porównanie modeli tRNA

Jednym z często podejmowanych zagadnień w dziedzinie modelowania RNA jest przewidywanie struktury tRNA. Jest to rodzina, dla której istnieje wiele modeli rozwiązanych oświadczalnie, co umożliwia weryfikację wyników uzyskanych komputerowo. Ponadto rodzina ta wykazuje silną konserwację globalnego zwoju i jednocześnie dużą zmienność lokalną. Zaprezentowany w niniejszej pracy program ModeRNA był testowany na zbiorze 99 tRNA, dla których zbudowane zostały 9702 modele, ocenione następnie pod względem podobieństwa do struktury natywnej (rozdział 4.5).

Modele tRNA zostały również zbudowane przez siedem innych metod. Kilka modeli drożdżowego tRNA^{Phe} zostało wygenerowanych, pogrupowanych i ocenionych przy użyciu programu NAST (Jonikas, i wsp., 2009). Używa on do przewidywania struktury cząsteczki RNA, która jest reprezentowana przy pomocy pseudoatomów odpowiadających grupom atomów rzeczywistych, oraz potencjału opracowanego na podstawie znanych struktur. Otrzymane w wyniku tej analizy trzy klastry struktur wykazały w stosunku do struktury natywnej RMSD 8,0 Å, 13,6 Å oraz 15,6 Å, a pod względem kryterium GDT-TS odpowiednio 0,20, 0,08 oraz 0,07. Flores i Altman również zbudowali model tRNA^{Phe} używając ograniczonej liczby znanych kontaktów trzeciorzędowych, oddziaływań warstwowych oraz danych z doświadczenia NMR (Flores and Altman, 2010). Efektem obliczeń była ścieżka atomów fosforu o RMSD 9,6 Å w stosunku do struktury natywnej. Lavender i wsp. wyliczyli model tRNA^{Asp}, dla którego RMSD uwzględniające wyłącznie atomy fosforu wyniosło 6,2 Å (Lavender, i wsp., 2010). Ding i Dokholyan otrzymali model tRNA^{Phe} o RMSD 7,2 Å, używając oprogramowania DMD oraz więzów struktury drugorzędowej (Ding, i wsp., 2008). Ci sami autorzy, używając wyłącznie dynamiki molekularnej, otrzymali model tRNA o RMSD 4,0 Å (Gherghe, i wsp., 2009). Cao i Chen opublikowali pełnoatomowy model tRNA o RMSD wszystkich atomów 4,2 Å w porównaniu do struktury natywnej (Cao and Chen, 2011). Został on zbudowany używając kombinacji potencjału bazującego na wiedzy uzyskanej z analizy znanych struktur oraz symulacji dynamiki molekularnej. Na stronie programu MC-Fold/MC-Sym (Parisien and Major, 2008) znajduje się skrypt do modelowania tRNA^{Phe} drożdżowego (struktura natywna 1EVV), który generuje model o RMSD 4,0 Å w stosunku do struktury natywnej.

Modele tRNA zbudowane programem ModeRNA osiągnęły średnią wartość podobieństwa do struktury natywnej liczoną według kryterium RMSD 5,6 Å. Dla poszczególnych modeli wartość ta wahała się od 0,15 Å, do 12,9 Å i była silnie skorelowana z podobieństwem sekwencji celu i szablonu. Średnie GDT-TS wyniosło 0,53. Wartości RMSD oraz innych kryteriów, porównujących podobieństwo modeli wygenerowanych przez program ModeRNA ze strukturami natywnymi, opisane zostały szczegółowo w rozdziale 4.5. Ich omówienie znajduje się również w publikacji „*ModeRNA: A tool for comparative modeling of RNA 3D structure*” (Rother, i wsp., 2011). Porównując modele tRNA otrzymane przy pomocy programu ModeRNA z wygenerowanymi przez inne narzędzia zauważyć należy, że większość z wymienionych w poprzednim paragrafie metod zaczyna modelowanie od sekwencji celu i symuluje proces zwijania się cząsteczki. Program ModeRNA używa struktury szablonu, co ułatwia budowę modelu. Jednak modelowanie porównawcze również posiada własne wyzwania, przede wszystkim identyfikację szablonu oraz przygotowanie przyrównania sekwencji celu i szablonu (rozdział 5.3). Jak widać z zaprezentowanych wyników, metody, które symulują zwijanie z użyciem więzów pochodzących z doświadczeń są w stanie zbudować wierny model tRNA, a modele zbudowane przez program ModeRNA są co najmniej porównywalne. Należy również zauważyć, że użycie dynamiki molekularnej, ze względu na czas obliczeń, nie pozwoliłoby na zbudowanie zbioru 9700 modeli.

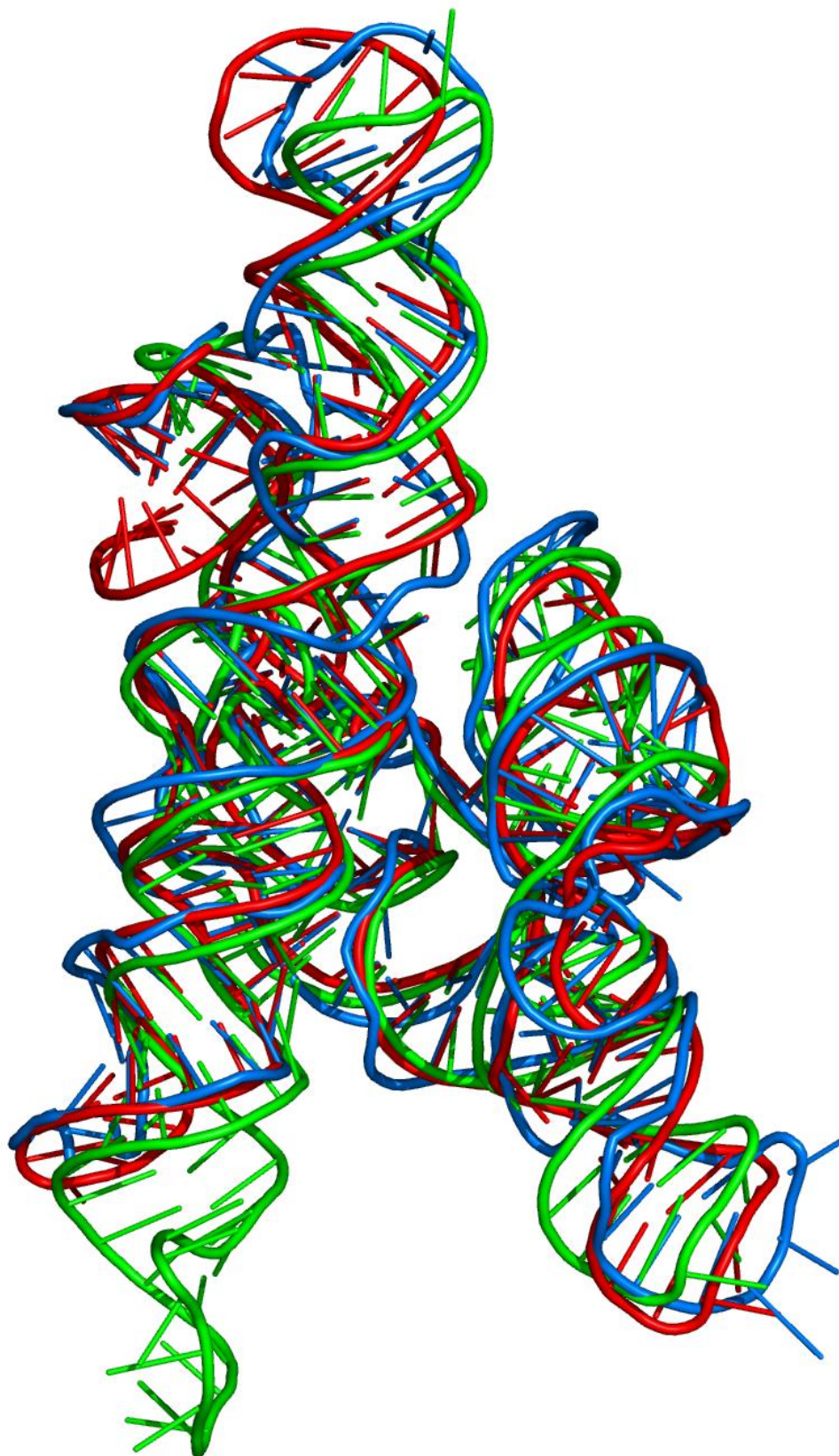
Omówienie różnych modeli tRNA znajduje się również w publikacji „*RNA tertiary structure prediction with ModeRNA*” (Rother, i wsp., 2011).

5.1.2. Porównanie programu ModeRNA z programem RNABuilder

Biorąc pod uwagę grupę programów wykorzystujących podejście modelowania porównawczego, modele generowane przez program ModeRNA zostały szczegółowo porównane z modelami zbudowanym przy użyciu programu RNABuilder (Flores, i wsp., 2010). Jest to metoda do modelowania porównawczego struktury RNA, w której cząsteczka RNA jest modelowana równolegle na różnym poziomie dokładności, od niskiej rozdzielczości do skali atomowej (uwzględniając atomy wodoru). Używa wewnętrznej dynamiki układów współrzędnych, aby wypełnić pobrane z szablonu więzy na wszystkich poziomach. Pole siłowe działające w niskiej rozdzielczości składa się z sił i momentów obrotu, które umożliwiają zbliżenie zasad i ich parowanie w regionach zdefiniowanych przez użytkownika. Użytkownik może również wyznaczyć regiony sztywne i giętkie

w cząsteczce. Programy ModeRNA i RNABuilder bazują na innych zasadach, a stosując analogię do modelowania białek, można je odpowiednio porównać do programów SWISS-MODEL (Schwede, i wsp., 2003) i MODELLER (Sali and Blundell, 1993).

W publikacji grupy Altmana (Flores, i wsp., 2010) zaprezentowany został model intronu grupy I *Azoarcus*. W artykule omówiono również przyrównanie wykorzystane podczas modelowania (przyrównanie sekwencji *Azoarcus* - celu i intronu grupy I *Twort* – szablonu (1Y0Q, łańcuch A)). Model tej cząsteczki zbudowany przez programu ModeRNA w trybie automatyczny (jeden szablon + przyrównanie dwóch sekwencji, celu i szablonu → model) osiągnął całkowite RMSD 6,9 Å w porównaniu do struktury natywnej. Następnie interfejs skryptowy został użyty do modelowania oddziaływania trzeciorzędowego pętli L9/P5 i L2/P8 na podstawie intronu *Tetrahymena*, wykorzystanego jako dodatkowy szablon według opisu zaawansowanego protokołu modelowania użytego do modelowania przez grupę Altmana. Dzięki drugiemu szablónowi i modelowaniu poprzez interfejs programistyczny powstał model o całkowitym RMSD 4,3 Å, a więc o 2,6 Å lepszy od modelu wygenerowanego automatycznie. RMSD struktury rdzenia drugiego modelu (zbudowanego z użyciem dwóch szablonów) wyniosło 2,0 Å. Wyniki te są nieco lepsze, niż dla modelu zbudowanego przez program RNABuilder – 4,4 Å (całkowite RMSD) i 2,7 Å (RMSD rdzenia) w odniesieniu do struktury natywnej. Modele zbudowane przy pomocy obydwu programów oraz strukturę natywną przedstawia Rycina 34, a szczegółowe porównanie wartości RMSD podsumowuje Tabela 10. Jak obrazuje powyższy przykład, użycie wielu szablonów wraz z zaawansowanym interfejsem skryptowym może poprawić jakość modelu w porównaniu do modelowania automatycznego na pojedynczym szablonie. Opis tego przypadku modelowania zawarty został również w artykule „*ModeRNA: A tool for comparative modeling of RNA 3D structure*” (Rother, i wsp., 2011). **Modelowanie intronu grupy I *Azoarcus* wykonał dr Kristian Rother.**



Rycina 34. Model intronu grupy I *Azoarcus* zbudowany przy użyciu programu ModeRNA (kolor czerwony) i RNABuilder (kolor niebieski), nałożone na strukturę rozwiązana doświadczalnie – kod PDB: 1U6B (kolor zielony). Na podstawie (Rother, i wsp., 2011).

Tabela 10. Wartości RMSD dla modeli intronu grupy I *Azoarcus* w odniesieniu do struktury rozwiązanej doświadczalnie (kod PDB: 1U6B). Modele zostały zbudowane przy użyciu programów ModeRNA i RNABuilder (Flores, i wsp., 2010). Na podstawie (Rother, i wsp., 2011).

Region	Model ModeRNA zbudowany automatycznie	Model ModeRNA zbudowany na dwóch szablonach	Model RNABuilder zbudowany na dwóch szablonach
	RMSD (Å) do struktury rozwiązanej doświadczalnie		
Cała struktura	6,87	4,28	4,41
Rdzeń (P1,P3, P4, P6, P7, P8, P8a)	4,16	1,99	2,72
Miejsce aktywne	2,61	2,30	2,53
Regiony peryferyjne (P2, P5, P5a, P6a, P9, P9.0)	8,32	5,25	5,47

5.1.3. Porównanie cech różnych metod do modelowania RNA

Jeden z programów, które również umożliwia modelowanie homologiczne to S2S/Assemble (dostępny przez platformę PARADISE) (Jossinet and Westhof, 2005). Jest to graficzny system, który łączy różne narzędzia i serwisy sieciowe we wszechstronne środowisko do redagowania sekwencji i struktury RNA. Daje ono możliwość bezpośredniego przypisywania par zasad oraz oddziaływań warstwowych, przyrównywania wielu sekwencji, oraz zapewnia automatyczną procedurę generowania modeli 3D na podstawie uprzedniej adnotacji lub też *de novo*. Wadą tego systemu jest to, że wszystkie oddziaływania muszą zostać przypisane przez użytkownika, co utrudnia przygotowanie wysokoprzepustowej analizy, takiej jak analiza modeli tRNA opisana w Wynikach (rozdział 4.5). Użytkownik ma kontrolę nad pojedynczymi oddziaływaniami i może wymusić oddziaływanie trzeciorzędowe z poziomu sekwencji, w związku z czym cały proces zależy silnie od jego doświadczenia. Przez to jakość uzyskanego modelu wiąże się z umiejętnościami użytkownika. Program PARADISE, podobnie jak inne programy zapewniające graficzne środowisko do interaktywnego modelowania eksperckiego opisane poniżej, może zostać użyty do manipulowania oraz łączenia modeli zbudowanych z użyciem programu ModeRNA, np. do budowy skomplikowanej struktury, która wymaga przestrzennego składania pojedynczo wymodelowanych elementów.

Program RNA2D3D (Martinez, i wsp., 2008), jak sugeruje nazwa, generuje przybliżone modele 3D rozpoczynając od sekwencji i struktury drugorzędowej. Procedura programu nie uwzględnia symulacji procesu zwijania się i z tego powodu nie jest w stanie automatycznie wygenerować satysfakcjonującego modelu 3D jedynie na podstawie informacji o parowaniu zasad. Konieczna jest manipulacja modelem poprzez interfejs graficzny, który daje możliwość operowania oddziaływaniami pomiędzy pojedynczymi zasadami oraz parami zasad. Przykładowo, zbudowanie modelu tRNA o typowym kształcie litery L, wymaga wykonania przynajmniej dwóch ważnych operacji poprzez interfejs graficznym (operacji wprowadzenia oddziaływania warstwowego między dwiema helisami oraz obrotu pary oddziałującej warstwowo z zachowaniem względnego położenia pozostałych elementów modelu).

ERNA-3D (Zwieb and Muller, 1997) to następny program, w którym modelowanie opiera się na pracy użytkownika w interfejsie graficznym. Interfejs pozwala na operowanie elementami strukturalnymi na różnym poziomie abstrakcji, podczas gdy rotacje ugrupowań chemicznych wokół pojedynczego wiązania wzdłuż łańcucha głównego cząsteczki są symulowane w czasie rzeczywistym.

Program 3D-RNAPredict + CYANA (Popenda, i wsp., 2006) wykorzystuje do budowy modelu RNA dane o strukturze drugorzędowej, motywach trzeciorzędowych, oraz opcjonalnie dane doświadczalne i pochodzące z dynamiki molekularnej. Metoda ta jest wysokoprzepustowa i może być wykorzystywana w trybie automatycznym. Program została zaprojektowana do szybkiego generowania modeli w niskiej rozdzielczości. Przewidywane mogą być struktury dużych cząsteczek – testy programu obejmowały strukturę 23S rRNA *Haloarcula marismortui* (kod PDB: 1jj2).

Na automatyczne uzyskanie modelu pozwalają także programy MC-Fold/MC-Sym (Parisien and Major, 2008) oraz FARNA (Das and Baker, 2007), wykorzystujące podejście *de novo*. Budują one modele z fragmentów uzyskanych ze struktur rozwiązanych doświadczalnie. W pewnym sensie modelują one zatem strukturę na bazie wielu szablonów. Jest to jednak inne wykorzystanie kombinacji różnych szablonów niż w modelowaniu homologicznym i z tego powodu metody te nie są tu szerzej omawiane.

Zestawienie najważniejszych cech wymienionych programów do modelowania struktury trzeciorzędowej RNA przedstawia Rycina 11.

Tabela 11. Porównanie programów do modelowania struktury przestrzennej RNA. Czarne kropki wskazują na obecność danej cechy, zaś puste okręgi jej brak. Gwiazdka oznacza, że kod źródłowy programu jest dostępny po wysłaniu zapytania do autorów. Na podstawie (Rother, i wsp., 2011).

Cechy \ Program	ModeRNA (niniejsza praca)	RNABuilder (Flores, i wsp., 2010)	PARADISE/Assemble (Jossinet and Westhof, 2005)	RNA2D3D (Martinez, i wsp., 2008)	ERNA-3D (Zwieb and Muller, 1997)	MC-Fold/MC-Sym (Parisien and Major, 2008)	FARNA/FARFAR (Das, i wsp., 2010)	NAST. (Jonikas, i wsp., 2009)	DMD/foldRNA (Sharma, i wsp., 2008)	YUP (Tan, i wsp., 2006)	3D-RNApredict (Popenda, i wsp., 2006)	RNA123 (Santalucia, 2002)	PyMOL (DeLano, 2002)	AmiraMol (Biegeleisen, 2006)
De novo	○	○	●	○	○	●	●	○	●	○	●	●	○	○
Przewidywanie z 2D	○	●	●	●	●	●	○	●	○	●	●	●	○	○
Porównawcze	●	●	●	●	●	○	○	○	○	○	○	●	○	○
Użycie przyrównania	●	●	●	○	○	○	○	○	○	○	○	●	○	●
Więzy 2D	●	●	●	●	●	●	●	●	○	●	●	●	○	○
Więzy 3D	○	●	●	●	●	●	○	●	○	●	●	●	○	○
Modyfikacje	●	○	○	○	○	○	○	○	○	○	○	○	○	○
Biblioteka fragmentów	●	○	●	●	○	●	●	●	○	○	●	●	○	○
Pełnoatomowe	●	●	●	●	●	●	●	●	○	○	●	●	●	●
Redagowanie	●	●	●	●	●	○	○	○	○	●	○	●	●	●
MD/MC/RE	○	●	○	○	○	○	●	●	●	●	●	●	○	○
Minimalizacja energii	○	○	●	●	○	●	●	○	●	●	○	●	○	●
Analiza geometrii	●	○	●	○	○	●	●	○	●	●	○	●	○	○
Interfejs graficzny	○	○	●	●	●	○	○	○	○	○	○	●	●	●
Interfejs skryptowy	●	●	○	○	○	●	●	●	○	●	○	○	●	●
Interfejs www	●	○	○	○	○	●	○	○	●	○	○	○	○	○
Darmowy	●	●	●	●	○	●	●	●	●	●	●	○	●	○
Otwarte źródło	●	○	○*	○	○	○*	○	●	○	○	○	○	●	○
Platforma	Win, Lin, Mac	Win, Mac	Lin, Mac	Lin	Win, SGI	Lin, Mac	Win, Lin	Win, Lin, Mac	wszy stkie	Lin, Mac	?	Win	wszy stkie	Win, Lin, Mac

Należy również nadmienić, że programy przeznaczone do wizualizacji cząsteczek mogą być również wykorzystywane do modelowania struktury RNA. W takich przypadkach cały proces modelowania wykonywany jest przez użytkownika poprzez środowisko graficzne. Niektóre z tych programów posiadają także interfejs programistyczny, który pomaga w dostosowaniu go do specyficznych zadań modelowania. Dla porównania z możliwościami typowych programów do modelowania zestawienie cech (Rycina 11) uwzględnia programy PyMOL (DeLano, 2002) oraz AmiraMol (Biegeleisen, 2006).

5.2. Ograniczenia i mocne strony programu ModeRNA

W publikacji programu RNABuilder Altman i wsp. zdefiniowali następujące cztery kryteria dla programu do modelowania porównawczego (Flores, i wsp., 2010): 1) pozwala użytkownikowi określić odpowiadające sobie reszty w sekwencji celu i strukturze szablonu; 2) pozwala na użycie więcej niż jednego szablonu; 3) jest w stanie modelować regiony bez szablonu (insercje/delecje); oraz 4) jest dostępny dla wszystkich naukowców, także tych, którzy nie są ekspertami w naukach komputerowych. Poniżej przedstawiona została dyskusja programu ModeRNA w odniesieniu do powyższych kryteriów.

1) Wyznaczenie odpowiadających sobie reszt pomiędzy sekwencją celu i strukturą szablonu.

Korespondencja pomiędzy celem i szablonem jest określona poprzez przyrównanie sekwencji dostarczone przez użytkownika (to typowa cecha wszystkich programów do modelowania porównawczego, włączając programy ModeRNA (Rother, i wsp., 2011), PARADISE (Jossinet and Westhof, 2005), RNABuilder (Flores, i wsp., 2010), a także w pewnym stopniu AmiraMol (Biegeleisen, 2006)). Tworzenie pary typu Watson-Crick dla danej reszty nukleotydowej jest kluczową funkcją programów do modelowania, która jest dostępna w większości programów, w tym w programie ModeRNA. Mocną stroną programu ModeRNA jest możliwość modelowania potranskrypcyjnych modyfikacji (co może zostać zaznaczone już na poziomie przyrównania).

2) Użycie kilku szablonów.

Użycie wielu szablonów może zostać osiągnięte przez program ModeRNA (Rother, i wsp., 2011), RNABuilder (Flores, i wsp., 2010) i PARADISE (Jossinet and Westhof, 2005) (Jossinet and Westhof, 2005). Taki sposób modelowania może poprawić jakość modelu przygotowanego metodą porównawczą, jak w przypadku modelu intronu grupy I *Azoarcus*

(opisanym w rozdziale 5.1). Możliwość bezpośredniej zmiany długości segmentu helisy i dodanie fragmentu o zadanej strukturze pozwala na modelowanie złożonych zmian w strukturze. W szerszym sensie programy MC-Fold/MC-Sym (Parisien and Major, 2008), FARNA (Das and Baker, 2007) i inne metody bazujące na składaniu fragmentów zawsze używają wielu szablonów, jednak nie jest to modelowanie homologiczne/porównawcze. Ograniczeniem programu ModeRNA jest brak procedury minimalizacji energii, która poprawiłaby nietypowe wartości geometryczne - potencjalną konsekwencję modelowania na wielu szablonach. ModeRNA pozwala natomiast na naprawienie nieciągłości łańcucha, gdy wstawiany do struktury fragment nie pasuje dokładnie do reszty modelu.

3) Modelowanie regionów bez szablonu (insercji/delecji).

Program ModeRNA (Rother, i wsp., 2011) może modelować regiony bez szablonu poprzez wstawianie fragmentów, wykorzystując podejście podobne do metody RLoM (Schudoma, i wsp., 2010). Wybór biblioteki fragmentów jest dyskutowany w rozdziale 5.4. Dla krótkich insercji/delecji (indels) (do 15 nukleotydów) element strukturalny potencjalnie podobny do natywnego może zostać wstawiony z biblioteki fragmentów, może też zostać wygenerowany cały zbiór kandydatów, pośród których użytkownik, po zwizualizowaniu wyników, może wybrać odpowiedni fragment pasujący do reszty modelu. Ta sytuacja jest podobna w programach: PARADISE (Jossinet and Westhof, 2005) i RNABuilder (Flores, i wsp., 2010), które również mogą konstruować krótkie fragmenty bez szablonu. Program ModeRNA pozwala ponadto wstawiać większe fragmenty zdefiniowane przez użytkownika, a także ograniczyć przeszukiwanie biblioteki fragmentów do posiadających określoną strukturę drugorzędową. Żadna z metod modelowania porównawczego nie jest zdolna do związania dużych fragmentów, które nie mają korespondującego szablonu. Jest to domena programów wykorzystujących podejście *ab initio* lub *de novo* np. FARNA/FARFAR (Das and Baker, 2007; Das, i wsp., 2010), DMD/iFoldRNA (Ding, i wsp., 2008; Sharma, i wsp., 2008), the MC-Fold/MC-Sym (Parisien and Major, 2008), lub NAST (Jonikas, i wsp., 2009). Istnieje oczywista potrzeba rozwinięcia oprogramowania do modelowania, które będzie w stanie związać duże fragmenty RNA w kontekście wymodelowanego homologicznie rdzenia.

4) Intuicyjny sposób użycia.

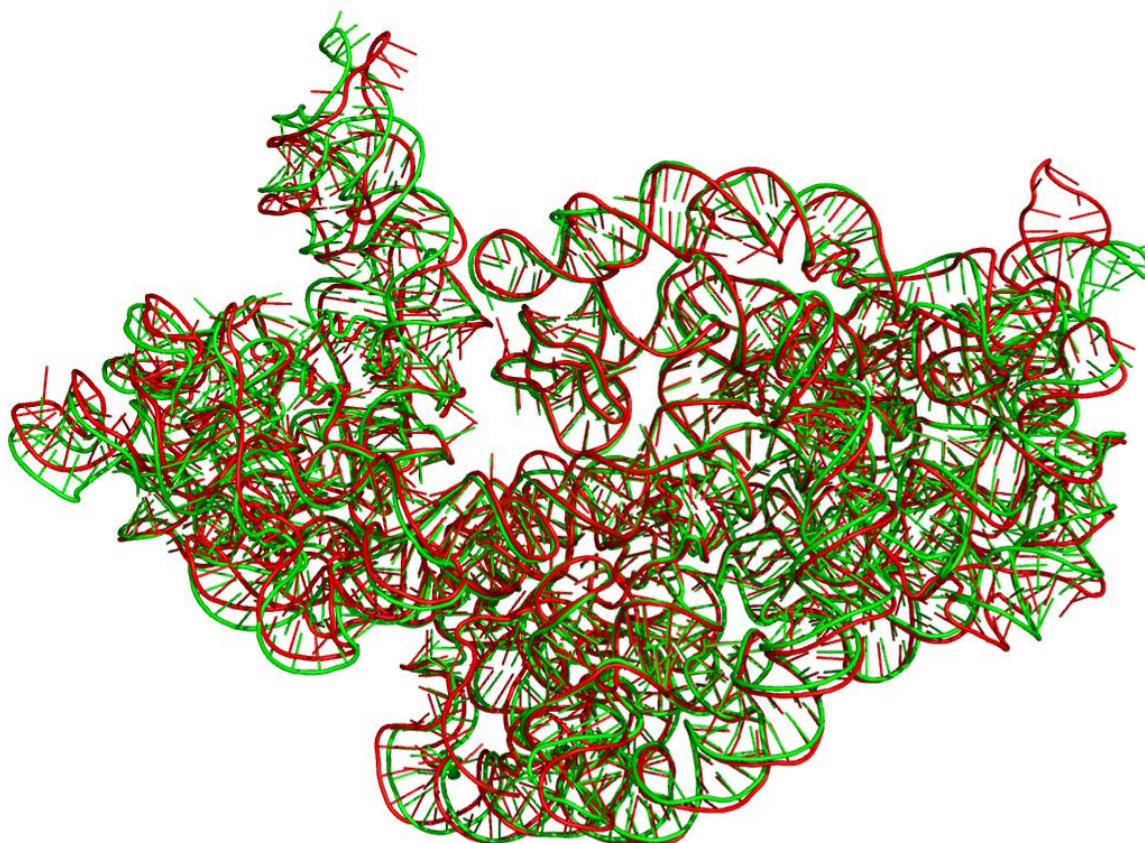
Możliwości istniejących programów do modelowania RNA różnią się w ilości pracy potrzebnej ze strony użytkownika do uzyskania modelu. Istnieje cała gama rozwiązań,

od interfejsów sieciowych wymagających jedynie wklejenia sekwencji celu, do złożonych interfejsów graficznych posiadających wiele zaawansowanych opcji. Dla obecnej wersji programu ModeRNA zaimplementowany został serwer internetowy (Rother, i wsp., 2011). Uzupełnia on samodzielną aplikację ModeRNA (Rother, i wsp., 2011) i ma na celu ułatwienie użycia programu użytkownikom nieposiadającym umiejętności programistycznych. Serwer nie zapewnia jednak wszystkich operacji dostępnych po zainstalowaniu programu na dysku (np. modelowania z użyciem kilku szablonów). Dla przypadku modelowania, gdzie model może być zbudowany tylko na podstawie informacji zawartej w szablonie i przyrównaniu, zwłaszcza, kiedy szablon i cel uważane są za bardzo podobne, serwer jest wystarczającym rozwiązaniem. Program ModeRNA zainstalowany na dysku posiada interfejs skryptowy, który oprócz prostych przypadków umożliwia też modelowanie eksperckie i redagowanie struktury i sekwencji. W celu ułatwienia modelowania eksperckiego na stronie programu ModeRNA zgromadzony został zbiór przykładów modelowania oraz gotowe skrypty dopasowane do konkretnych sytuacji. Dzięki interfejsowi programistycznemu program ModeRNA może być również łączony z innymi programami do modelowania, przetwarzania danych i wizualizacji struktury.

Program ModeRNA (Rother, i wsp., 2011) spełnia wszystkie kryteria wyznaczone przez Altmana i wsp. jako niezbędne dla programu do modelowania porównawczego struktury RNA (Flores, i wsp., 2010). Niewątpliwie jego dużym ograniczeniem jest wymóg doświadczalnie rozwiązanej struktury szablonu niezbędnej do przeprowadzenia modelowania. Struktura taka musi istnieć, zostać prawidłowo zidentyfikowana oraz przyrównana do sekwencji celu. Cecha ta jest wspólna dla wszystkich programów wykorzystujących podejście modelowania homologicznego/porównawczego i została szerzej omówiona w rozdziale 5.3.

Problemu tego unikają programy używające podejścia *de novo* lub *ab initio*, takie jak MC-Fold/MC-Sym (Parisien and Major, 2008), FARNA (Das and Baker, 2007), NAST (Jonikas, i wsp., 2009), DMD/iFoldRNA (Ding, i wsp., 2008; Sharma, i wsp., 2008), YUP (Chen, i wsp., 2006), 3D-RNAPredict (Popena, i wsp., 2006), czy RNA123 (SantaLucia, 2002). Symulacja procesu zwijania się cząsteczki jest jednak bardzo kosztowna obliczeniowo i wymaga dużo czasu już dla niewielkich struktur. Przykładowo, program MC-Fold/MC-Sym modeluje strukturę drożdżowego tRNA^{Phe} (struktura natywna 1EVV ma długość 76 nukleotydów) w ciągu 24 godzin. Zaletą programu ModeRNA jest to, że nie wymaga dużych zasobów obliczeniowych. Przykładowo, modelowanie cząsteczki tRNA zajmuje

od dwóch do dwudziestu sekund na komputerze z procesorem 2,4GHz. Dokładny czas obliczeń zależy przede wszystkim od ilości i rozmiaru insercji/delecji w szablonie. Dzięki temu możliwe jest budowanie modeli nawet dla bardzo dużych cząsteczek. Poniżej (Rycina 35) przedstawiony został model 16S rRNA z podjednostki 30S rybosomu *T. thermophilus* wymodelowany programem ModeRNA.



Rycina 35. Model 16S rRNA z podjednostki 30S rybosomu *T. thermophilus* (na czerwono) nałożony na strukturę rozwiązana doświadczalnie – kod PDB: 1J5E (na zielono). Na podstawie (Rother, i wsp., 2011).

5.3. Wybór szablonu i przygotowanie przyrównania

Jednym z podstawowych ograniczeń program ModeRNA (Rother, i wsp., 2011) (tak jak wszystkich programów do modelowania porównawczego/homologicznego) jest nieodzowność struktury szablonu. Baza PDB (Berman, i wsp., 2000) zawiera przedstawicieli wielu ważnych rodzin, jednak znalezienie odpowiedniego szablonu dla konkretnego celu jest często trudnym zadaniem. Capriotti i wsp. opisali „strefę cienia” sekwencji RNA, w której modelowanie homologiczne jest trudne. Do sfery tej należą sekwencje o podobieństwie na poziomie 0,42-0,71 (Capriotti and Marti-Renom, 2010). Zbiór tRNA omawiany w niniejszej pracy (rozdział 4.5) ma średnią identyczność sekwencji na poziomie 51%. Ponadto, nawet jeśli struktura szablonu jest dostępna, kolejnym warunkiem udanego modelowania jest poprawne, istotne biologicznie, przyrównanie sekwencji celu i szablonu. W dziedzinie porównawczego przewidywania struktury białek te trzy zagadnienia: identyfikacja szablonu (określane dla białek jako rozpoznawanie zwoju), przygotowanie przyrównania i modelowanie są traktowane jako oddzielne problemy. O podobnym podziale mówić można w przypadku RNA, tym bardziej, że również dla tej cząsteczki istnieją oddzielne programy do tworzenia przyrównania sekwencja-sekwencja i sekwencja-struktura. Program ModeRNA skupia się na samym procesie przekształcania szablonu w szukaną strukturę.

W celu ułatwienia identyfikacji szablonu, przyrównywania sekwencji i identyfikacji elementów struktury drugorzędowej, strona www programu ModeRNA zapewnia sekwencje i struktury drugorzędowe (w formacie VIENNA) reprezentatywnego zbioru struktur z bazy PDB, które mogą służyć jako potencjalne szablony. Nie rozwiązuje to problemu poszukiwania szablonu, jednak może stanowić narzędzie, które ułatwi to zadanie użytkownikowi. Ponadto na serwerze ModeRNA udostępnione zostały operacje dedykowane przygotowaniu danych wejściowych do modelowania przy pomocy programu ModeRNA (oczyszczenie struktury szablonu, formatowanie pliku PDB), jak i programów zewnętrznych (ParAlign, Infernal, R-coffee). Szukanie szablonu do modelowania oraz przygotowanie przyrównania są głównymi wartościami dodanymi serwera w stosunku do samodzielnego programu ModeRNA instalowanego na dysku (rozdział 4.3.3) (Rother, i wsp., 2011). Alternatywą dla narzędzi udostępnianych przez serwer ModeRNA może być algorytm BLAST (Sayers, i wsp., 2009). Jednak podczas wyszukiwania bierze on pod uwagę tylko

dane o sekwencji, nie uwzględniając struktury drugorzędowej, co może powodować uzyskiwanie wyników gorszej jakości, niż te dostarczane przez serwer ModeRNA.

Serwer ModeRNA został wykorzystany między innymi do wyszukania szablonu dla przypadku modelowania tRNA^{Thr} *E. coli*, opisanego w „*RNA tertiary structure prediction with ModeRNA*” (Rother, i wsp., 2011), którego celem było zbudowanie modelu w konformacji związanej do aminoacylo-tRNA syntetazy, zaczynając od sekwencji cząsteczki celu. Wybrane zostały dwa alternatywne szablony – 1EHZ (tRNA^{Phe} *S. cerevisiae*) oraz 1C0A (tRNA^{Asp} *E. coli*). Sekwencja celu miała taką samą długość jak szablon 1EHZ i pojedynczą delecję o długości jednego nukleotydu w stosunku do szablonu 1C0A. Identyczność sekwencji szablony 1EHZ i 1C0A do sekwencji celu wynosiła odpowiednio 0,59 i 0,46. Sprawdzona została również poprawność geometryczna obu struktur (Tabela 12). Szablon 1EHZ wykazywał większe podobieństwo do cząsteczki celu na poziomie sekwencji oraz zawierał mniej problemów sterycznych, jednak szablon 1C0A znajdował się w pożądanym stanie biologicznym. Modele zbudowane w oparciu o obydwa szablony przedstawia Rycina 36. Lepsze oceny podczas porównania ze strukturą natywną według większości miar otrzymał model zbudowany na szablonie 1C0A (Tabela 12).

Tabela 12. Ocena jakości modeli tRNA^{Thr} *E. coli* zbudowanych na dwóch różnych szablony. Pogrubiona czcionka oznacza lepszą ocenę.

	Kryterium	Model bazujący na szablonie 1EHZ	Model bazujący na szablonie 1C0A
Model	Międzyatomowe zderzenia steryczne	1	2
	Nietypowe długości wiązań	2	7
	Nietypowe wartości kątów płaskich	0	3
	Nietypowe wartości kątów torsyjnych	2	3
Model w odniesieniu do struktury doświadczalnej	RMSD wszystkich atomów	5,07	3,38
	RMSD łańcucha głównego (atomy P, C4')	4,37	2,60
	Ocean TM	0,56	0,67
	Ocena GDT-TS	0,55	0,66
	DI	0,82	0,79
	Średnie DP	12,70	9,85



Rycina 36. Modele tRNA^{Thr} *E. coli* zbudowane na podstawie szablonu tRNA^{Phe} *Saccharomyces cerevisiae* (kod PDB: 1EHZ; kolor czerwony) i szablonu tRNA^{Asp} (kod PDB: 1C0A; kolor niebieski) nałożone na strukturę rozwiązana doświadczalnie – kod PDB: 1QF6. Na podstawie (Rother, i wsp., 2011).

Jak to obrazuje przypadek *E. coli* tRNA^{Thr} identyczność sekwencji nie powinna być jedynym kryterium przy wyborze szablonu. Podczas modelowania porównawczego użytkownik jest odpowiedzialny za określenie pożądanego stanu funkcjonalnego dla struktury celu. Makrocząsteczki takie jak białka i RNA wykazują bowiem lokalne lub/i globalne zmiany konformacyjne zależne od środowiska, a w szczególności od oddziaływań z innymi cząsteczkami.

Przyrównania użyte podczas modelowaniu tRNA^{Thr}, tak jak szablony, również zostały uzyskane przez serwer ModeRNA, dzięki zaimplementowanej na nim nakładce na metodę Infernal. Dla przygotowania 9701 modeli tRNA, podczas testowania programu ModeRNA (rozdział 4.5), użyte zostało przyrównanie sekwencji z bazy Rfam (Gardner, i wsp., 2009). Jedyną „manualną” zmianą wprowadzoną w tym przyrównaniu było dodanie modyfikacji

w odpowiednich pozycjach (na podstawie struktur PDB), przyrównania w bazie Rfam nie zawierają bowiem modyfikowanych nukleotydów. Gotowe przyrównania istnieją dla wielu rodzin RNA (baza Rfam przechowuje dane na temat 1973 rodzin). Do poprawy istniejącego przyrównania, gdy znane są dane o strukturze drugorzędowej, wykorzystać można pakiet R-Coffee (Wilm, i wsp., 2008). Ponadto istnieje wiele narzędzi do tworzenia automatycznego przyrównania (w przypadku kiedy szablon jest już znany). Niektóre z nich, jak Muscle lub ClustalW (Thompson, i wsp., 1994), posługują się wyłącznie sekwencją, inne, jak ConSan (przyrównujący dwie sekwencje) (Dowell and Eddy, 2006) lub LocARNA (Otto, i wsp., 2008), FoldalignM (Torarinsson, i wsp., 2007) i Stemloc (Holmes, 2005) (przyrównujące wiele sekwencji), korzystają również z informacji o strukturze drugorzędowej. Programy do przyrównania sekwencji RNA są szczegółowo dyskutowane w publikacji programu R-coffee (Wilm, i wsp., 2008).

W celu zademonstrowania wpływu jakości przyrównania na modelowanie porównawcze metody Infernal i ClustalW zostały wykorzystane do przyrównania intronu grupy I *Azoarcus* i *Twort*. Modele stworzone na tych danych porównane zostały z modelem uzyskanym na podstawie przyrównania eksperckiego (opisanego w publikacji Altmana i wsp. (Flores, i wsp., 2010)). Przyrównanie automatyczne spowodowało pogorszenie jakości modelu. Wartość RMSD zwiększyła się z 4,28 Å (model zbudowany z użyciem programu ModeRNA) i 4,41 Å (model zbudowany z użyciem programu RNABuilder) do 23,9 Å i 26,4 Å (ModeRNA-Infernal, ModeRNA-ClustalW) oraz 57,1 Å i 44,7 Å (RNABuilder-Infernal, RNABuilder-ClustalW). Na obniżenie jakości modeli wpływ miało użycie tylko jednego szablonu, jednak kluczowa była zła jakość przyrównania. Program ModeRNA, podobnie jak inne programy do modelowania porównawczego, wymaga ostrożnego przygotowania przyrównania sekwencji celu i szablonu. Jest to, obok identyfikacji odpowiedniego szablonu strukturalnego, drugie wyzwanie modelowania porównawczego.

5.4. Biblioteka fragmentów

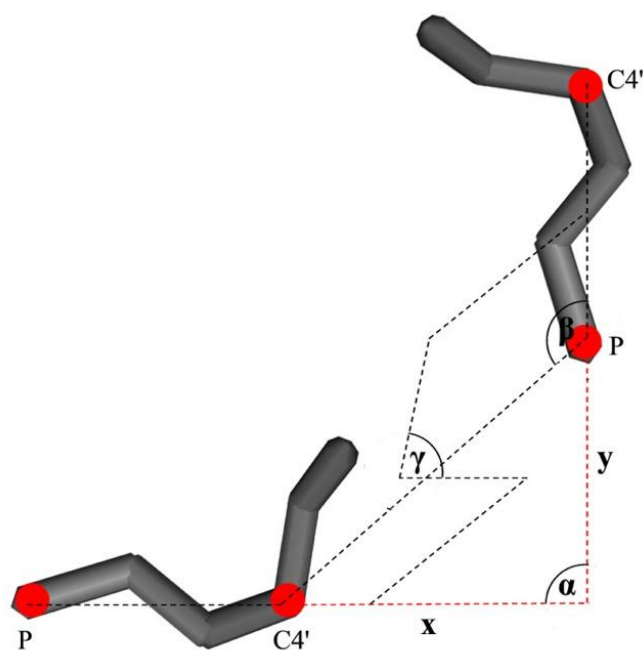
Program ModeRNA przeprowadza modelowanie na podstawie dostarczonego przez użytkownika szablonu i przyrównania. Miejsca w sekwencji celu, które nie mają odpowiednika w strukturze szablonu uzupełniane są fragmentami z biblioteki dołączonej do źródła programu. Głównym celem biblioteki fragmentów jest zapewnienie fragmentów łączących dwie reszty, a nie szukanie konkretnych motywów. Do wygenerowania biblioteki posłużył zbiór struktur rnaDB2005 oryginalnie użyty do wyliczenia możliwych konformacji

łańcucha głównego RNA (Murray and Carr, 2008). Zawiera on szeroką gamę konformacji, i jak pokazały testy na rodzinie tRNA, może zostać użyty do wygenerowania fragmentów o wystarczającej zmienności konformacyjnej pozwalającej na modelowanie insercji i delecji w kontekście modelowania porównawczego (Rother, i wsp., 2011). Fragmenty pochodzą z różnych rodzin, i żadna rodzina nie jest faworyzowana pod względem liczby struktur. Już sama struktura rRNA (również zawarta w zbiorze rnaDB2005) jest, rozważana jako wystarczające źródło fragmentów do budowy modelu (Das and Baker, 2007).

Podobne podejście do modelowania fragmentów bez szablonu używane jest w programie SWISS MODEL do modelowania porównawczego białek (Schwede, i wsp., 2003). Metoda oceny dopasowania fragmentów z biblioteki do reszty modelu była inspirowana bazą danych LIP (Michalsky, i wsp., 2003). Baza ta dedykowana jest modelowaniu homologicznemu pętli w białkach. Przechowuje fragmenty białek o długości od jednego do piętnastu aminokwasów. Procedura wyszukiwania w bazie fragmentu pasującego pomiędzy dwa niepołączone aminokwasy szablonu podzielona jest na dwa etapy. W pierwszej, bardzo szybkiej, fazie szacowane jest RMSD fragmentów o odpowiedniej długości, a także identyczność szukanej sekwencji z sekwencją fragmentu. W drugim, zajmującym więcej czasu, etapie wyselekcjonowana pula fragmentów jest dopasowywana do modelu, liczone jest dokładne RMSD i odrzucane są fragmenty, które tworzące poważne zderzenia steryczne z modelem.

Biblioteka fragmentów ModeRNA zawiera fragmenty RNA o maksymalnej długości 17 nukleotydów. Takie ograniczenie wiąże się z rozmiarem całego pakietu ModeRNA. W przypadku, gdy konieczne jest wstawienie dłuższego fragmentu ze strony programu ModeRNA pobrać można bibliotekę zawierającą fragmenty o długości do 100 nukleotydów (111MB). Procedura dopasowania fragmentów w programie ModeRNA również podzielona jest na dwa etapy. Podczas optymalizacji całego procesu szczególnie dużo czasu poświęcono na optymalizację funkcji szacującej RMSD. Porównane zostały różne wartości geometryczne (np. dystanse i kąty na płaszczyźnie wyznaczonej przez dwa wektory będące przedłużeniami odcinków pomiędzy atomami P i C4' w obydwu resztach kotwiczących – Rycina 37). Najdokładniejsze dopasowanie fragmentów uzyskane zostało, gdy do szacowania RMSD używanych było sześć dystansów pomiędzy odpowiadającymi sobie atomami z reszt kotwiczących (O5'; C5'; C4'; C3'; C1'; N1 lub N9). Wszystkim dystansom przypisana jest równa waga (rozdział 4.2.2.4). Analiza ta prowadzona była przez wielokrotne budowanie

modeli dla zbioru 99 tRNA. Na czas tej analizy z biblioteki wyłączone zostały fragmenty pochodzące z tRNA.

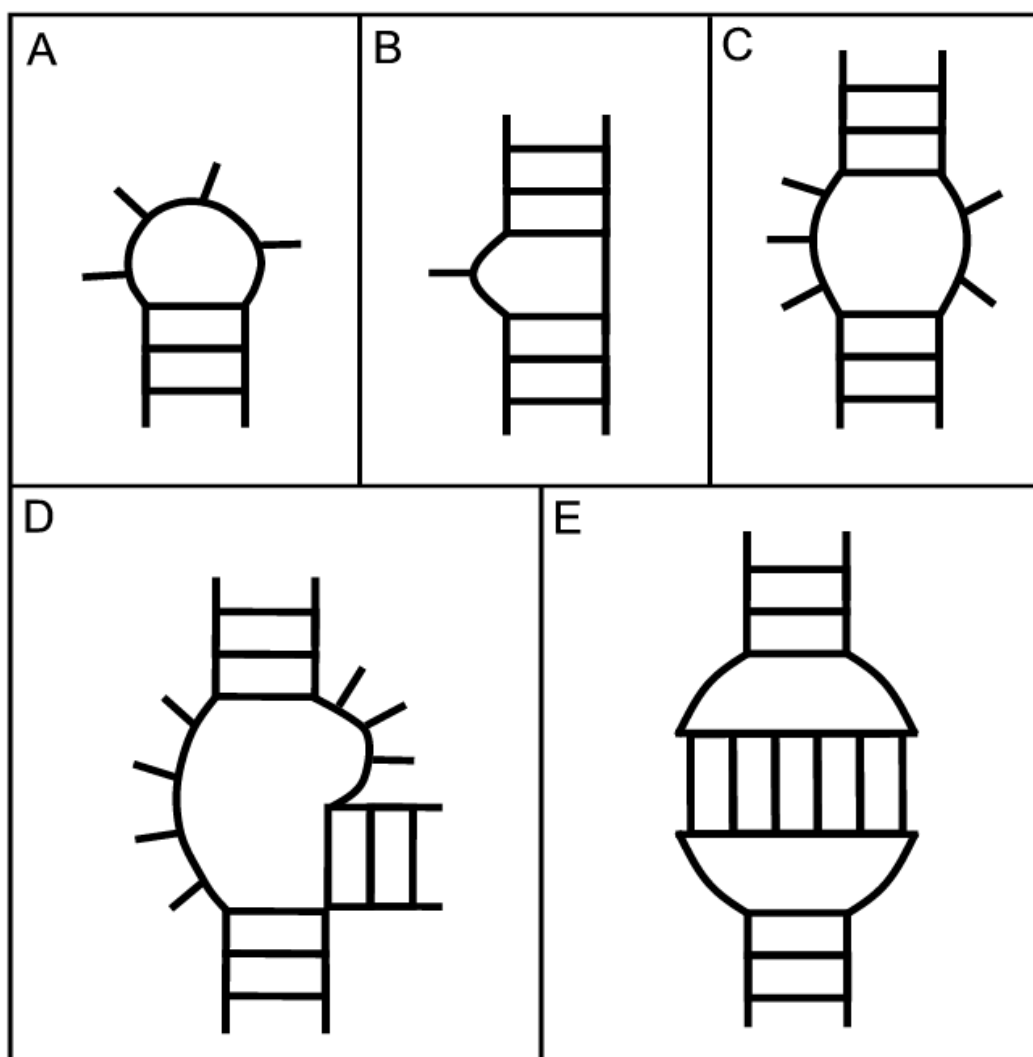


Rycina 37. Przykłady wartości rozważanych podczas implementacji programu ModeRNA jako kryteria wstępnego dopasowania fragmentu. Ostateczne kryteria używane w bieżącej wersji programu ModeRNA omówione zostały w rozdziale 4.2.2.4 i przedstawia je Rycina 20.

Dalsze wsparcie dla zastosowanej metody modelowania elementów sekwencji celu bez szablonu zostało przedstawione przez Schudomę i wsp. (Schudoma, i wsp., 2010; Schudoma, i wsp., 2010), którzy rozwinęli narzędzie do wyszukiwania fragmentów działające na bardzo podobnej zasadzie jak metoda zaimplementowana w ModeRNA. Przeprowadzona przez nich analiza wykazała, że podobieństwo strukturalne jest utrzymywane nawet w dużych pętlach i że jest skorelowane z podobieństwem sekwencyjnym. W badaniach przeprowadzonych przez Schudomę i wsp. uwzględnione zostały jedynie fragmenty jednoniciowe. W ModeRNA podejście modelowania regionów bez szablonu poprzez wstawianie fragmentów z biblioteki było z sukcesem stosowane do wszystkich regionów strukturalnych cząsteczki RNA. W czasie optymalizacji procedury wstawiania fragmentów porównane zostały: biblioteka zawierająca jedynie fragmenty pochodzące z regionów jednoniciowych oraz biblioteka utworzona ze wszystkich regionów cząsteczek RNA (również helis). Biblioteka zawierająca fragmenty pochodzące zarówno z regionów jedno- jak i dwuniciowych pozwoliła na uzyskanie dokładniejszego dopasowania fragmentów do modelu.

Struktura drugorzędowa została uwzględniona jako dodatkowe kryterium, które może zostać opcjonalnie wykorzystane podczas przeszukiwania biblioteki fragmentów programu ModeRNA. Biblioteka nie jest jednak przeznaczona do wyszukiwania motywów trzeciorzędowych, czy elementów dwuniciowych. W celu uwzględnienia takich fragmentów program ModeRNA wspiera bezpośrednie użycie elementów pobranych z innych szablonów. Kilka baz danych specjalizujących się w motywach strukturalnych RNA ułatwia szukanie odpowiedniego fragmentu. Przykładowo RNAJunction zapewnia 12000 motywów złącza i całujących się pętli, uzyskanych z 1176 struktur RNA z bazy PDB (Bindewald, i wsp., 2008). Baza SCOR (*Structural Classification of RNA*) zawiera klasyfikację strukturalną i funkcjonalną 579 plików PDB. Wspiera ona wyszukiwanie motywów (np. zwrotu i pętli GNRA). Pozwala też na przegląd motywów znajdujących się we wskazanej strukturze PDB (Klosterman, i wsp., 2002). Baza WebFR3D (Sarver, i wsp., 2008) umożliwia szukanie motywów trzeciorzędowych poprzez definiowanie matrycy oddziaływań pomiędzy poszczególnymi nukleotydami.

Zaawansowanym narzędziem do poszukiwania fragmentów o zadanej sekwencji i strukturze drugorzędowej (określanej przez użytkownika w notacji VIENNA) jest baza FRABASE opracowana na podstawie 1990 plików PDB zawierających struktury RNA (Popena, i wsp., 2008), (Popena, i wsp., 2010). Przykłady struktur drugorzędowych, które mogą być wyszukiwane poprzez interfejs internetowy tej bazy przedstawia Rycina 38. Ostatnio podobna baza – RNA CoSSMos, została zaproponowana przez Vanegasa i wsp. Do szukania różnego rodzaju motywów pętli (symetrycznych i asymetrycznych pętli wewnętrznych, pętli terminalnych i wybrzuszeń) (Vanegas, i wsp., 2012).



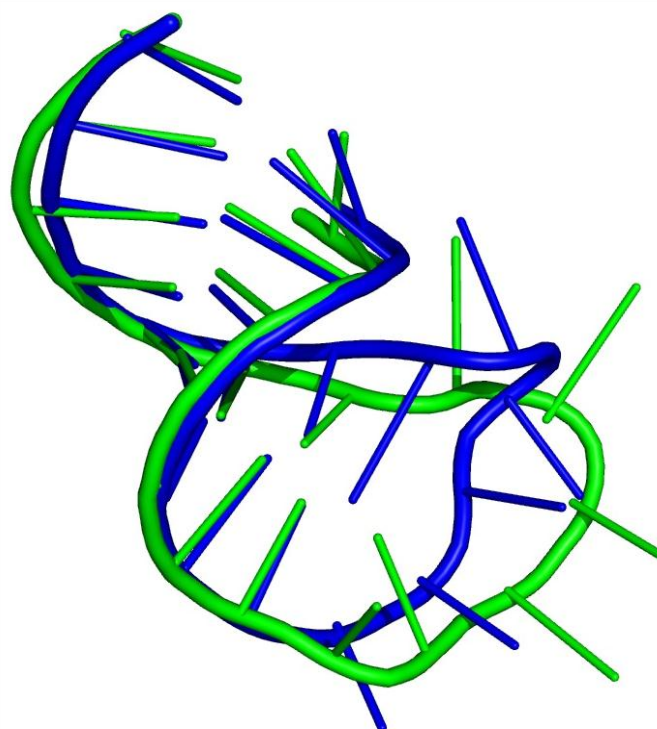
Rycina 38. Przykłady motywów dostępnych w bazie FRABASE: A) spinka do włosów; B) wybrzuszenie; C) asymetryczna pętla wewnętrzna; D) złącze; E) całujące się pętle. Rycina przygotowana na podstawie ilustracji dostępnych na stronie bazy FRABASE (Popenda, i wsp., 2008).

Fragmenty wyszukane przez niezależne bazy danych mogą zostać z pomocą programu ModeRNA wycięte i wstawione do struktury modelu. Mogą być to zarówno fragmenty jednokłasiowe i dwukłasiowe. Należy zauważyć, że wprowadzenie nawet prostego motywu może wpłynąć na całą strukturę przestrzenną (np. (Popenda, i wsp., 2008)). W programie ModeRNA użytkownik może też łączyć ze sobą dowolną ilość fragmentów. Wybiega to poza modelowanie prostych pętli i pozwala składać model RNA z motywów uzyskanych z wymienionych wcześniej bibliotek. Zautomatyzowane składanie fragmentów drugorzędowych pozwoliłoby na modelowanie *de novo* podobne do prezentowanego przez program RNA2D3D (Martinez, i wsp., 2008), FARNA (Das and Baker, 2007), lub MC-Fold/MC-Sym (Parisien and Major, 2008), jednak ten typ modelowania nie jest obecnie dostępny w programie ModeRNA.

5.5. Program MARS

Zbiór 9702 modeli tRNA był wielokrotnie budowany i oceniany – porównywany ze strukturami natywnymi. Implementacja wielu cech w programie ModeRNA, takich jak wyszukiwanie fragmentów z biblioteki wbudowanej w program czy dodawanie motywów z drugiego szablonu, wymagała bowiem wielokrotnego testowania. Również porównywanie programu ModeRNA z innymi narzędziami wymagało przestrzennego nakładania modeli (jak np. intronu grupy I) i liczenia wartości RMSD, GDT-TS i innych kryteriów podobieństwa. Istnieje kilka narzędzi, którymi można się posłużyć w celu takiej analizy. Przykładowo, biblioteka Biopython.PDB (Hamelryck and Manderick, 2003) zawiera moduł służący do wykonywania nałożenia przestrzennego struktur, pozwalający uzyskać RMSD takiego nałożenia. Również program PyMOL (DeLano, 2002) (rozdział 3.4.2) umożliwia taką operację, a R3D Align to serwer internetowy pozwalający na nałożenie struktur. Programy RNAView (Yang, i wsp., 2003) (rozdział 3.4.1) i FR3D+Matlab (Sarver, i wsp., 2008) mogą zostać użyte do wyliczenia danych potrzebnych do uzyskania DI i DP.

Nie istnieje jednak program, który umożliwiłby nałożenie przestrzenne struktur i porównanie ich według kilku kryteriów jednocześnie, automatyczne wykonanie analizy dla większej ilości struktur oraz zindywidualizowanie takiej analizy. W celu ułatwienia i zautomatyzowania porównywania wielu modeli ze strukturą/strukturami natywnymi lub porównywania modeli między sobą powstał program MARS (*Model Assessment of RNA Structures*) (rozdział 4.6). Umożliwia on przestrzenne nałożenie struktur oraz wyliczenie RMSD, GDT-TS, DI i DP. Program pozwala na zindywidualizowanie analizy - kontrolę nad tym, które części struktury i które atomy są używane do nałożenia i porównywania struktur. Użytkownik ma wgląd w sposób przeprowadzania analizy, dzięki czemu analiza jest obiektywna. Przykładowo wartość RMSD może bardzo się zmieniać w zależności od rodzaju atomów użytych do obliczeń. Rycina 39 przedstawia ramię antykodonowe modelu tRNA^{Thr}, wykazuje ono w stosunku do struktury natywnej (1QF6) RMSD 4,2 Å jeśli brane są pod uwagę wszystkie atomy, 2,8 Å jeśli nie uwzględnia się zasad, 2,6 Å dla atomów fosforu i 2,4 Å dla atomów P i C4'. Duża różnica wynikająca ze sposobu dokonywania obliczeń powoduje, że należy być ostrożnym przy porównywaniu wartości RMSD (i innych miar) pochodzących z różnych źródeł.



Rycina 39. Ramię antykodonowe modelu tRNA^{Thr} *E. coli* zbudowanego na podstawie szablonu tRNA^{Asp} (kod PDB: 1C0A; kolor niebieski) nałożone na strukturę rozwiązana doświadczalnie – kod PDB: 1QF6 (kolor zielony). Na podstawie (Rother, i wsp., 2011). Rycina 36 przedstawia cały model tej cząsteczki.

Dla modeli tRNA opisanych w rozdziale 5.1 dane o RMSD pochodziły z różnych publikacji, liczone były na podstawie różnych atomów i dają możliwość jedynie przybliżonego porównania dokładności modeli. W przypadku modelu intronu *Azoarcus* (rozdział 5.1), dzięki komunikacji z grupą profesora Altmana, uzyskany został model tej cząsteczki wykonany z użyciem programu RNABuilder. Model został oceniony przez program MARS, podobnie jak model zbudowany programem ModeRNA, co dało możliwość obiektywnego porównania wyników (Tabela 10).

Innym problemem przy porównywaniu wartości RMSD pochodzących z różnych źródeł jest wykluczanie z obliczeń reszt, które nie zostały poprawnie nałożone pomiędzy modelem i strukturą oryginalną (np. w ten sposób działa procedura obliczająca RMSD w programie PyMOL). To również wpływa znacząco na uzyskaną wartość RMSD. W programie MARS domyślnie pod uwagę brane są wszystkie atomy. Dopiero wskazanie przez użytkownika reszt, które nie mają być brane pod uwagę podczas obliczeń pozwala na wykluczenie części struktury z analizy. Biblioteka MARS jest uniwersalna i może być zastosowana do porównywania modeli przestrzennych RNA wygenerowanych dowolną metodą lub też dowolnych natywnych cząsteczek RNA zapisanych w formacie PDB.

5.6. Perspektywy rozwoju programu ModeRNA

Program ModeRNA zapewnia użytkownikom wykonywanie operacji związanych z modelowaniem, analizą i redagowaniem struktury trzeciorzędowej. Model może być budowany automatycznie w oparciu o dostarczony szablon i przyrównanie oraz redagowany z wykorzystaniem dodatkowych szablonów lub motywów strukturalnych zidentyfikowanych w zewnętrznych bazach danych. Zaimplementowane w programie cechy pozwalają, jak udowodniły opisane w niniejszej pracy testy, na skuteczne przygotowanie modeli o jakości zbliżonej do różnorodności występującej pośród struktur natywnych (rozdział 4.5). Niemniej nadal istnieją cechy, których dodanie do programu mogłoby ułatwić użytkownikowi proces modelowania i prawdopodobnie wpłynęłoby pozytywnie na jakość modeli.

Istotną cechą struktury trzeciorzędowej RNA, jest występowanie w niej izosterycznych par zasad. Są to strukturalnie podobne do siebie pary zasad, które mogą się wzajemnie zastępować nie powodując zakłóceń w strukturze trzeciorzędowej (Stombaugh, i wsp., 2009). Utworzone dla takich par matryce izosteryczności wraz z adekwatnym zbiorem struktur (Lescoute, i wsp., 2005) pozwalają zidentyfikować dla danej pary, oddziałujących ze sobą zasad w szablonie, parę, która w modelu nie zmieni konformacji przestrzennej danego motywu. W programie ModeRNA modelowanie par izosterycznych nie jest obecnie dostępne. W przypadku, gdy w szablonie para oddziałujących ze sobą zasad ma inną sekwencję niż w modelu to zasady wymieniane są niezależnie, co może spowodować utratę ważnego dla struktury lub funkcji oddziaływania. Ciekawym przedsięwzięciem byłoby więc zaimplementowanie metody, która na podstawie matryc izosteryczności automatycznie identyfikuje pary izosteryczne pomiędzy celem i szablonem. Wybiera ze zbioru struktur izosterycznych par zasad odpowiednią parę i wymienia ją. Następnie zaś sprawdza jak ta zmiana w sposobie modelowania wpływa na jakość modeli.

Pierwsze próby wykonania tego zadania podjął **lic. Paweł Skiba**, który podczas swojego licencjatu zaimplementował nieautomatyczną metodę wymiany izosterycznych par zasad. Okazało się jednak, że wymiana całej pary nie zawsze może być dokonana bezpośrednio, a dopasowanie pary do istniejącego łańcucha cukrowo-fosforanowego wymaga dalszej optymalizacji.

Jednym z rozwiązań, które mogłoby pomóc w dopasowywaniu fragmentów wstawianych do modelu uzyskanego na podstawie szablonu jest zaimplementowanie w programie ModeRNA procedury do minimalizacji energii. Problem ten dotyczy nie tylko

wymieniania par zasad, ale także opcji zaimplementowanych już w programie. Przykładowo, po wstawieniu do cząsteczki modelu fragmentu z zewnętrznej biblioteki, lub fragmentu wyszukanego dla regionu bez szablonu, końce cukrowo-fosforanowego łańcucha głównego modelu i fragmentu pozostają czasami niepołączone. Wprowadzenie w programie ModeRNA procedury do minimalizacji energii lub integracja z programem pozwalającym na wykonanie tego zadania (np. SimRNA (Rother, i wsp., 2012)) prawdopodobnie poprawiłoby jakość niektórych modeli, zwłaszcza gdy dla sekwencji celu nie ma bliskiego homologa i istnieje wiele regionów, które muszą zostać wymodelowane bez szablonu. W obecnej chwili program ModeRNA ma zaimplementowany algorytm FCCD do usuwania nieciągłości łańcucha (*Full Cyclic Coordinate Descent*, pozwala na utworzenie segmentu łączącego dwie sztywne części łańcucha głównego, został zaimplementowany dla białek (Boomsma and Hamelryck, 2005)). Cząsteczka modelu może również zostać poddana symulacji w programie MMTK (Hinsen, 2000). Na stronie programu ModeRNA znajduje się skrypt, który pozwala przeprowadzić taką symulację po zainstalowaniu programu MMTK.

Innym zagadnieniem z pogranicza modelowania porównawczego i modelowania bez szablonu jest przewidywanie struktury dłuższych pętli. Niewątpliwie użyteczne byłoby zaimplementowanie i zintegrowanie z programem ModeRNA metody, która byłaby w stanie modelować *ab initio* fragment struktury dopasowany do pozostałej, nieruchomej części modelu, pochodzącej z szablonu. Problem ten jednak wykracza poza modelowanie porównawcze. W obecnej implementacji programu ModeRNA dłuższe insercje/delecje mogą być modelowane na podstawie dodatkowego szablonu lub też poprzez wyszukiwanie fragmentu z poszerzonej biblioteki pobranej ze strony programu (zawiera fragmenty o długości do 100 nukleotydów). W przyszłości mogłaby powstać metoda (jako część programu ModeRNA lub osobna aplikacja) łącząca ze sobą funkcjonalność programu ModeRNA i SimRNA (Rother, i wsp., 2012).

Ważnym problemem zidentyfikowanym podczas testowania programu ModeRNA jest również możliwość szybkiego i precyzyjnego porównywania modeli ze strukturą natywną lub między sobą. W tym celu zaimplementowany został program MARS. Był on jednak traktowany jedynie jako narzędzie pomocnicze podczas rozwoju oprogramowania ModeRNA. Jest dostępny jedynie na żądanie użytkownika (poprzez pocztę internetową) i nie posiada udogodnień w postaci interfejsu. W przypadku dalszego rozwoju, a więc również testowania programu ModeRNA, prawdopodobnie wygodnym rozwiązaniem byłoby zintegrowanie programu MARS z programem ModeRNA lub też z jego serwerem.

Architektura programu ModeRNA została tak zaplanowana, że możliwe jest wprowadzanie kolejnych funkcjonalności. Działanie kodu jest zabezpieczone zbiorem 702 testów, które po dodaniu nowego elementu pozwolą sprawdzić, czy żadna z dotychczas istniejących funkcji nie przestała działać poprawnie. Dodawanie kolejnych modułów powinna też ułatwić obszerna dokumentacja, którą posiadają już istniejące komponenty kodu. Otwartą kwestią jest strategia dalszej rozbudowy programu. Możliwe jest rozwijanie programu ModeRNA jako jednej bardzo wszechstronnej aplikacji do pracy z modelem, lub też zbioru mniejszych programów dedykowanych poszczególnym zadaniom (podobnie jak w zajmującym się przewidywaniem struktury drugorzędowej RNA pakiecie Vienna (Lorenz, i wsp.)). Oba podejścia mają swoje wady i zalety. Ważną kwestią jest również zakres cech, które dostępne będą w samodzielnej aplikacji ModeRNA oraz w wersji udostępnianej przez serwer.

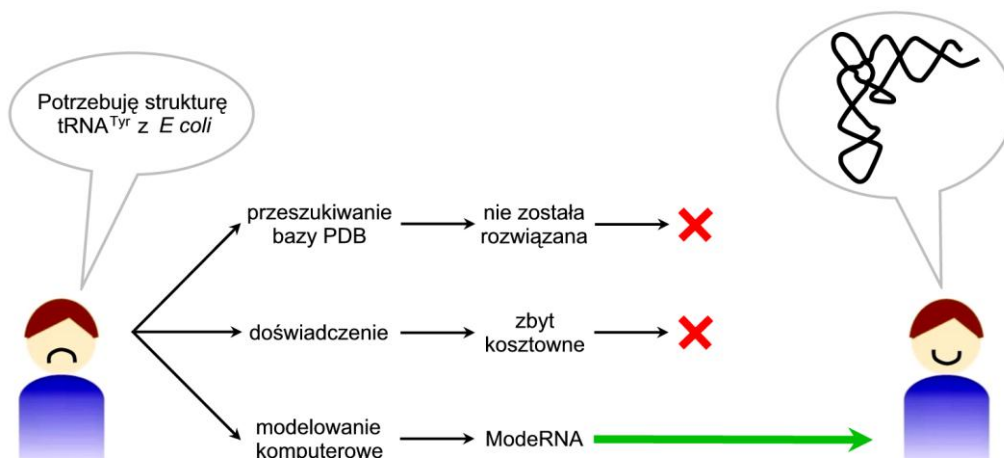
5.7. Wnioski

ModeRNA to program do modelowania struktury przestrzennej RNA. Jest on dostępny pod licencją wolnego i otwartego oprogramowania (GNU GPL) na stronie <http://iimcb.genesilico.pl/moderna>. Serwer programu ModeRNA udostępnia najważniejsze opcje programu bez konieczności instalowania go na dysku twardym i jest dostępny na stronie <http://iimcb.genesilico.pl/modernaserver>.

Głównymi zaletami programu ModeRNA, które odróżniają go od innych narzędzi do modelowania jest elastyczny interfejsu skryptowy, który umożliwia budowanie modeli RNA z wykorzystaniem wielu strategii (włączając szybkie modelowanie automatyczne bazujące na szablonie i przyrównaniu bez dodatkowych informacji) oraz możliwość modelowania modyfikowanych nukleotydów. Nie ma ograniczeń w wielkości struktur, które mogą być modelowane. Automatyczny tryb modelowania jest przeznaczony do modelowania struktur wysoce podobnych do szablonu. Dla bardziej wymagających przypadków wspierane są zaawansowane operacje takie jak: składanie struktury z fragmentów uzyskanych z zewnętrznych baz danych oraz modelowanie struktury drugorzędowej. Ponadto obiektowa architektura programu, zestaw testów i obszerna dokumentacja kodu powodują, że program może być dalej rozszerzany o nowe funkcjonalności, w zależności od zgłaszanych potrzeb użytkowników.

Program został zainspirowany oprogramowaniem przewidującym strukturę białek. Ta dziedzina ma długą tradycję. W 1994 roku został przeprowadzony pierwszy eksperyment oceniający dokładność programów przewidujących strukturę białka (CASP) (Moult, i wsp., 1995). Od tej pory jest on przeprowadzany co dwa lata stymulując ciągły rozwój nowych i udoskonalanie już istniejących metod (ostatni odbył się w 2010 roku – CASP9 (Kryshtafovych, i wsp.) i projektowany jest kolejny w roku 2012). Modele powstałe w wyniku modelowania porównawczego są wykorzystywane w praktyce, między innymi do przewidywania funkcji białek i do projektowania nowych leków (Jacobson and Sali, 2004).

Biorąc pod uwagę sukces komputerowego przewidywania białek oraz podobieństwa białka i RNA, dziedzina modelowania RNA może okazać się równie ważną. Tym bardziej, że dla ustrukturalizowanych RNA znanych jest wiele zastosowań praktycznych (Erdmann, i wsp., 2006), a w roku 2011 odbył się pierwszy eksperyment oceniający jakość metod do modelowania RNA (Cruz, i wsp., 2012). Autorka niniejszej pracy uważa, że program ModeRNA okaże się pomocny dla naukowców badających strukturę i funkcję RNA oraz będzie stymulował rozwój innych narzędzi (Rycina 40).



Rycina 40. Możliwości otrzymania struktury trójwymiarowej RNA.

6. Załącznik

6.1. Kod programu ModeRNA (CD)

Na płycie CD załączony został cały kod programu ModeRNA w wersji 1.7.1.

6.2. Wykaz identyfikatorów wykorzystanych plików PDB

W trakcie realizacji niniejszej pracy pobrano wiele danych z bazy PDB (Berman, i wsp., 2000) i NDB (Berman, i wsp., 2002). Posłużyły one między innymi do opracowania biblioteki fragmentów dla modelowania regionów celu bez szablonu, ewaluacji programu oraz porównania go z innymi programami do modelowania RNA (rozdział 2). Tabela 13 zawiera wykaz identyfikatorów wszystkich użytych struktur.

Tabela 13. Pliki wykorzystane podczas pracy nad projektem ModeRNA. Niektóre pliki zostały użyte do kilku zastosowań.

Zastosowanie	Identyfikator struktury
Pliki wykorzystane do utworzenia biblioteki fragmentów	157D, 1A34, 1A9N, 1APG, 1AV6, 1B7F, 1BMV, 1BR3, 1C0A, 1C9S, 1CSL, 1CVJ, 1CX0, 1D4R, 1DDL, 1DFU, 1DI2, 1DK1, 1DQF, 1DQH, 1DUH, 1DUQ, 1E7K, 1EC6, 1EFO, 1EHZ, 1ET4, 1F27, 1F7U, 1F8V, 1FEU, 1FFY, 1FIX, 1FXL, 1G1X, 1G2E, 1G2J, 1G4Q, 1GAX, 1GTF, 1GTN, 1H4S, 1HMH, 1HQ1, 1HR2, 1HYS, 1I5L, 1I6U, 1I9X, 1IL2, 1J1U, 1J8G, 1JBR, 1JJM, 1JZV, 1K8W, 1KD5, 1KNZ, 1KQ2, 1KXK, 1L2X, 1L3Z, 1L9A, 1LC4, 1LNG, 1LNT, 1M5O, 1M8V, 1M8X, 1M8Y, 1MHK, 1MMS, 1MSW, 1MSY, 1MWL, 1N35, 1N78, 1NB7, 1NLC, 1NTB, 1NUJ, 1NYI, 1OOA, 1OSU, 1P79, 1PGL, 1PJO, 1PVO, 1Q2R, 1Q9A, 1QC0, 1QF6, 1QLN, 1QTQ, 1R3E, 1R9F, 1RC7, 1RLG, 1RNA, 1RPU, 1S03, 1S72, 1S76, 1SA9, 1SAQ, 1SDR, 1SDS, 1SER, 1SI3, 1SJ3, 1T0E, 1TFW, 1U0B, 1U8D, 1U9S, 1URN, 1VFG, 1WMQ, 1WNE, 1WSU, 1XJR, 1XMQ, 1XOK, 1Y26, 1Y27, 1YFG, 1ZDJ, 205D, 255D, 259D, 280D, 283D, 2A8V, 2BBV, 2FMT, 333D, 353D, 354D, 397D, 398D, 402D, 404D, 405D, 409D, 413D, 420D, 422D, 433D, 434D, 438D, 439D, 466D, 472D, 479D, 5MSF, 6MSF, 7MSF, ARB0, ARB0, ARB0, ARB0, DRB0, DRB0, DRB0, DRB0, DRB0, DRB0, DRB0, DRD0, URB0, URB0, URB0, URC0

Szablony tRNA	1B23, 1C0A, 1EFW, 1EFW, 1EHZ, 1EIIY, 1EUQ, 1EUY, 1EXD, 1F7U, 1F7V, 1FFY, 1G59, 1G59, 1GTS, 1H3E, 1H4S, 1IL2, 1IL2, 1J1U, 1J2B, 1J2B, 1N77, 1N77, 1N78, 1N78, 1OB2, 1PNS, 1PNS, 1QF6, 1QRS, 1QRT, 1QRU, 1QTQ, 1QU2, 1QU3, 1WZ2, 1WZ2, 1YL4, 1YL4, 1ZJW, 2AKE, 2AZX, 2AZX, 2B64, 2B64, 2B9M, 2B9M, 2B9O, 2B9O, 2BTE, 2BTE, 2BYT, 2BYT, 2CSX, 2CSX, 2CT8, 2CT8, 2CV0, 2CV1, 2CV1, 2CV2, 2CV2, 2D6F, 2D6F, 2DER, 2DER, 2DET, 2DEU, 2DEU, 2DR2, 2DU3, 2DU4, 2DU5, 2DU6, 2DXI, 2DXI, 2HGI, 2HGI, 2HGP, 2HGP, 2HGP, 2HGR, 2HGR, 2IY5, 2J00, 2J02, 2J02, 2OW8, 2OW8, 2QNH, 2QNH, 2TRA, 2V0G, 2V0G, 2V46, 2V48, 3TRA, 4TNA
Pliki wykorzystane w dyskusji (rozdział 5)	1GID, 1J5E, 1M5O, 1Y0Q, 1Y26, 1Y27, 2AVY, 1B23, 1C0A, 1EFW, 1EHZ, 1EXD, 1H3E, 1H4S, 1I6U, 1L3D, 1QF6, 1RAW, 1RNA, 2AP0, 2BTE, 2DU3, 2OW8, 3FU2, 3JYV, 5BRU, 1U6B

6.3. Przyrównanie 99 sekwencji tRNA

Podczas ewaluacji programu ModeRNA zbudowane zostały 9702 modele tRNA. Budowa każdego modelu wymagała struktury szablonu i przyrównania sekwencji celu i szablonu (rozdział 2). Poniżej przedstawione jest przyrównanie 99 sekwencji tRNA uzyskane na podstawie bazy Rfam (Gardner, i wsp., 2009), z którego ekstrahowane były wszystkie 9702 przyrównania dwóch sekwencji. W nagłówku każdej sekwencji zawarty został kod PDB odpowiadającej jej struktury.

```

> 2j00_W.pdb
GCCCC-GAUAGCUCAGU--CGGU--AGA-GCAGG-GGAU-UGAAAAUCCCCGU-----G-UC-CUUGGU UC-G--AU--U-C-CG--A-G-UC-CGGGCA-CCA
> 1c0a_B.pdb
GGAGC-GG4AGUUCAGD--CGGD-DAGA-AUACC-UGCC-UQUCACGCAGGGG-----7-UC-GCGGGT PC-G--AG--U-C-CC--G-P-CC-GUUCGG-CCA
> 1i12_D.pdb
-CCGU-GAUAGUUPAAD---GGD-CAGA-AUGGG-CGCP-UGUCKCGUGCCAG-----A--U-?GGGGT PC-A--AU--U-C-CC--C-G-UC-GCGGA-----
> 4tna_A.pdb
GCGGA-UUUALCUCAGD--DGGG--AGA-GCRCC-AGAB-U#AAYAP?UGGAG-----7-UC-?UGUGT PC-G--"U--C-C-AC--A-G-AA-UUCGCA-CCA
> 2b64_W.pdb
GCGGA-UUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAYAPCUGGAG-----G-UC-CUGUGU PC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 1n78_D.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 1qrt_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGCA-----U-UC-CGAGGU UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 1ehz_A.pdb
GCGGA-UUUALCUCAGD--DGGG--AGA-GCRCC-AGAB-U#AAYAP?UGGAG-----7-UC-?UGUGT PC-G--"U--C-C-AC--A-G-AA-UUCGCA-CCA
> 1y14_B.pdb
GCGGA-UUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAGAUCUGGAG-----G-UC-CUGUGU UC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 1y14_C.pdb
GCGGA-UUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAGAUCUGGAG-----G-UC-CUGUGU UC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 2hgp_C.pdb
GCCCC-GAUAGCUCAGU--CGGU--AGA-GCAGG-GGAU-UGAAAAUCCCCGU-----G-UC-CUUGGU UC-G--AU--U-C-CG--A-G-UC-CGGGCA-CCA
> 2azx_C.pdb
GACCU-CGUGGCGCAAU---GGU--AGC-GCGUC-UGAC-UCCAGAUCAGAAG-----G-UU-GCGUGU UC-G--AA--U-C-AC--G-U-CG-GGGUCA-----
> 1pns_V.pdb
GCGGA-UUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAGAUCUGGAG-----G-UC-CUGUGU UC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 2cv1_D.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 1g59_D.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2qnh_z.pdb
CGCGG-GGUGGAGCAGCC-UGGU--AGC-UCGUC-GGGC-UCAUAACCCGAAG-----A-UC-GUCGGU UC-A--AA--U-C-CG--G-C-CC-CCGCAA-CCA
> 2cv0_C.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 1i12_C.pdb
UCCGU-GAUAGUUPAAD---GGD-CAGA-AUGGG-CGCP-UGUCKCGUGCCAG-----A--U-?GGGGT PC-A--AU--U-C-CC--C-G-UC-GCGGAG-CCA
> 1ob2_B.pdb
GCGGA-UUUALCUCAGD--DGGG--AGA-GCRCC-AGAB-U#AAYAP?UGGAG-----7-UC-?UGUGU PC-G--"U--C-C-AC--A-G-AA-UUCGCA-CCA
> 1n77_C.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA

```

```

> 2b9m_v.pdb
GCGGA-UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAGAUCUGGAG-----G-UC-CUGUGU      UC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 1gts_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGCA-----U-UC-CGAGGU      UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 2b64_v.pdb
GCGGA-UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAGAUCUGGAG-----G-UC-CUGUGU      UC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 1n78_C.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU      UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2det_C.pdb
GUCCC-CUUCGUCUAGA---GGCCCAGG-ACACC-GCCC-UUUCACGGCCGUA-----A--C-AGGGGU      UC-G--AA--U-C-CC--C-U-AG-GGG-----
> 1wz2_D.pdb
GCGGG-GGUUGCCGAGCC-UGGUCAAAG-GCGGG-GGAC-UCAAGAUCUCCCGU-AGGGGUU--C-CGGGGU      UC-G--AA--U-C-CC--C-G-CC-CCCGCA-CCA
> 1b23_R.pdb
GGCGC-GU4AACAAGC---GGD--DAU-GUAGC-GGAP-UGCA*APCCGUCU-----A--G-UCCGGT      PC-G--AC--U-C-CG--G-A-AC-GCGCCU-CCA
> 2v48_W.pdb
CGCGG-GGUGGAGCAGCC-UGGU--AGC-UCGUC-GGGC-UCAUAACCCGAAG-----G-UC-GUCGGT      UC-A--AA--U-C-CG--G-C-CC-CCGCAA-CCA
> 2ake_B.pdb
GACCU-CGUGGCGCAAU---GGU--AGC-GCGUC-UGAC-UCCAGAUCAGAAG-----G-UU-GCGUGU      UC-G--AA--U-C-AC--G-U-CG-GGGUCA----
> 1zjw_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGCA-----U-UC-CGAGGU      UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCa
> 1n77_D.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU      UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2cv2_C.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU      UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2dr2_B.pdb
GACCU-CGUGGCGCAAU---GGU--AGC-GCGUC-UGAC-UCCAGAUCAGAAG-----G-UU-GCGUGU      UC-G--AA--U-C-AC--G-U-CG-GGGUCA-CCA
> 2der_D.pdb
--CCC-CUUCGUCUAGA---GGCCCAGG-ACACC-GCCC-UUUCACGGCCGUA-----A--C-AGGGGU      UC-G--AA--U-C-CC--C-U-AG-GGGACG----
> 2azx_D.pdb
GACCU-CGUGGCGCAAU---GGU--AGC-GCGUC-UGAC-UCCAGAUCAGAAG-----G-UU-GCGUGU      UC-G--AA--U-C-AC--G-U-CG-GGGUCA----
> 2dxi_C.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU      UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2ct8_D.pdb
GGCGG-CGUAGCUCAGC--UGGU-CAGA-GCGGG-GAUC-UCAUAAGUCCAG-----G-UC-GGAGGU      UC-G--AG--U-C-CU--C-C-CG-CCGCCA----
> 1qf6_B.pdb
GCCGA-UAUAGCUCAGD--DGGD--AGA-GCAGC-GCAU-UCGUEAUGCAGG-----7-UC-GUAGGT      PC-G--AC--U-C-CU--A-U-UA-UCGGCA-CCA
> 2cv2_D.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU      UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2ow8_z.pdb
GCCCC-GA4AGCUCAGD--CGGD--AGA-GCAGG-GGAP-UGAA*APCCCCGU-----7-UC-CUUGGT      PC-G--AU--U-C-CG--A-G-UC-CGGGCA-CCA

```

```

> 1qtq_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGCA-----U-UC-CGAGGU UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 1efw_C.pdb
GGAGC-GG4AGUUCAGD--CGGD-DAGA-AUACC-UGCC-UQUC/CGCAGGGG-----7-UC-GCGGGT PC-G--AG--U-C-CC--G-P-CC-GUUC-----
> 1qrs_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGCA-----U-UC-CGAGGU UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 1qru_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGCA-----U-UC-CGAGGU UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 2b9o_W.pdb
GCGGA_UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAYAPCUGGAG-----_-----G-UC-CUGUGU PC-G--AU--C-C-AC_-A-G-AA-UUCGCA_CCA
> 2b9o_V.pdb
GCGGA_UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAGAUCUGGAG-----G-UC-CUGUGU UC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 1j1u_B.pdb
CCGGC-GGUAGUUCAGCC-UGGU--AGA-ACGGC-GGAC-UGUAGAUCGCAU-----G-UC-GCUGGU UC-A--AA--U-C-CG--G-C-CC-GCCGGA-----
> 2j02_V.pdb
CGCGG-GGUGGAGCAGCC-UGGU--AGC-UCGUC-GGGC-UCAUAACCCGAAG-----G-UC-GUCGGT UC-A--AA--U-C-CG--G-C-CC-CCGCAA-CCA
> 1pns_W.pdb
GCGGA_UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAGAUCUGGAG-----G-UC-CUGUGU UC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 2b9m_W.pdb
GCGGA_UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAYAPCUGGAG-----_-----G-UC-CUGUGU PC-G--AU--C-C-AC_-A-G-AA-UUCGCA_CCA
> 2j02_W.pdb
GCCCC-GAUAGCUCAGU--CGGU--AGA-GCAGG-GGAU-UGAAAAUCCCCGU-----G-UC-CUUGGU UC-G--AU--U-C-CG--A-G-UC-CGGGCA-CCA
> 2dxi_D.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2hgp_B.pdb
GCCCC-GAUAGCUCAGU--CGGU--AGA-GCAGG-GGAU-UGAAAAUCCCCGU-----G-UC-CUUGGU UC-G--AU--U-C-CG--A-G-UC-CGGGCA-CCA
> 1g59_B.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 2hgp_D.pdb
GCCCC-GAUAGCUCAGU--CGGU--AGA-GCAGG-GGAU-UGAAAAUCCCCGU-----G-UC-CUUGGU UC-G--AU--U-C-CG--A-G-UC-CGGGCA-CCA
> 2ct8_C.pdb
GGCGG-CGUAGCUCAGC--UGGU-CAGA-GCGGG-GAUC-UCAUAAGUCCAG-----G-UC-GGAGGU UC-G--AG--U-C-CU--C-C-CG-CCGCCA-----
> 2cv1_C.pdb
GGCCC-CAUCGUCUAGC---GGU-UAGG-ACGCG-GCCC-UCUCAAGGCCGAA-----A--C-GGGGGU UC-G--AU--U-C-CC--C-C-UG-GGGUCA-CCA
> 1f7v_B.pdb
PUCCU-CGUKLCCCAAD---GGD-CACG-GCRPC-UGGC-UICGAACCAGAAG-----A-DU-?CAGGT PC-A--"G--U-C-CU--G-G-CG-GGGGAG-----
> 1wz2_C.pdb
GCGGG-GGUUGCCGAGCC-UGGUCAAAG-GCGGG-GGAC-UCAAGAUCUUCCCGU-AGGGGUU--C-CGGGGU UC-G--AA--U-C-CC--C-G-CC-CCCGCA-CCA
> 2v46_W.pdb
CGCGG-GGUGGAGCAGCC-UGGU--AGC-UCGUC-GGGC-UCAUAACCCGAAG-----G-UC-GUCGGT UC-A--AA--U-C-CG--G-C-CC-CCGCAA-CCA

```

```

> 1f7u_B.pdb
PUCCU-CGUKLCCCAAD---GGD-CACG-GCRPC-UGGC-UICGAACCAGAAG-----A-DU-?CAGGT      PC-A--"G--U-C-CU--G-G-CG-GGGAAG-CCA
> 3tra_A.pdb
UCCGU-GAUAGUUPAAD---GGD-CAGA-AUGGG-CGCP-UGUCKCGUGCCAG-----A--U-?GGGGT      PC-A--AU--U-C-CC--C-G-UC-GCGGAG----
> 2tra_A.pdb
UCCGU-GAUAGUUPAAD---GGD-CAGA-AUGGG-CGCP-UGUCKCGUGCCAG-----A--U-?GGGGT      PC-A--AU--U-C-CC--C-G-UC-GCGGAG-C--
> 1ffy_T.pdb
GGGCU-UGUAGCUCAGG--UGGU-UAGA-GCGCA-CCCC-UGAUAAGGGUGAG-----G-UC-GGUGGU      UC-A--AG--U-C-CA--C-U-CA-GGCCCA-C--
> 2deu_C.pdb
GUCCC-CUUCGUCUAGA---GGCCCAGG-ACACC-GCCC-UUUCACGGCGGUA-----A--C-AGGGGU      UC-G--AA--U-C-CC--C-U-AG-GGGACG-C--
> 1qu3_T.pdb
GGGCU-UGUAGCUCAGG--UGGU-UAGA-GCGCA-CCCC-UGAUAAGGGUGAG-----G-UC-GGUGGU      UC-A--AG--U-C-CA--C-U-CA-GGCCCA-C--
> 2deu_D.pdb
GUCCC-CUUCGUCUAGA---GGCCCAGG-ACACC-GCCC-UUUCACGGCGGUA-----A--C-AGGGGU      UC-G--AA--U-C-CC--C-U-AG-GGGACG-C--
> 1qu2_T.pdb
GGGCU-UGUAGCUCAGG--UGGU-UAGA-GCGCA-CCCC-UGAUAAGGGUGAG-----G-UC-GGUGGU      UC-A--AG--U-C-CA--C-U-CA-GGCCCA-C--
> 2der_C.pdb
GUCCC-CUUCGUCUAGA---GGCCCAGG-ACACC-GCCC-UUUCACGGCGGUA-----A--C-AGGGGU      UC-G--AA--U-C-CC--C-U-AG-GGGACG-C--
> 1eiy_C.pdb
GCCGA-GGUAGCUCAGU--UGGU--AGA-GCAUG-CGAC-UGAAAAUCGCAGU-----G-UC-CGCGGU      UC-G--AU--U-C-CG--C-G-CC-UCGGCA-CCA
> 2iy5_T.pdb
GCCGA-GGUAGCUCAGU--UGGU--AGA-GCAUG-CGAC-UGAAAAUCGCAGU-----G-UC-CGCGGU      UC-G--AU--U-C-CG--C-G-CC-UCGGCA-CCA
> 2hgr_D.pdb
GCCGA-GGUAGCUCAGU--UGGU--AGA-GCAUG-CGAC-UGAAAAUCGCAGU-----G-UC-GGCGGU      UC-G--AU--U-C-CG--C-U-CC-UCGGCA-CCA
> 2hgi_D.pdb
GCCGA-GGUAGCUCAGU--UGGU--AGA-GCAUG-CGAC-UGAAAAUCGCAGU-----G-UC-GGCGGU      UC-G--AU--U-C-CG--C-U-CC-UCGGCA-CCA
> 2hgi_C.pdb
CGCGG-GGUGGAGCAGC--CGGU--AGC-UCGUC-GGGC-UCAUAACCCGAAG-----G-UC-GUCGGU      UC-A--AA--U-C-CG--G-C-CC-CCGCAA-CCA
> 2hgr_C.pdb
CGCGG-GGUGGAGCAGC--CGGU--AGC-UCGUC-GGGC-UCAUAACCCGAAG-----G-UC-GUCGGU      UC-A--AA--U-C-CG--G-C-CC-CCGCAA-CCA
> 2du3_D.pdb
--GCC-AGGGUGGCAGA---GGGGCUUU-GCGGC-GGAC-UGCAGAUCCGCUU-----UA--C-CCCAGU      UC-G--AA--U-C-CG--G-G-CC-CUG-GC----
> 2du4_C.pdb
--GCC-AGGGUGGCAGA---GGGGCUUU-GCGGC-GGAC-UGCAGAUCCGCUU-----UA--C-CCCAGU      UC-G--AA--U-C-CG--G-G-CC-CUG-GC----
> 1h4s_T.pdb
-GGAG-UAGCGCAGCCC---GGUA-GCG-CACCU-CGUU-CGGGACGAGGGGG-----G--C-GCUGGT      PC-A--GA--U-C-CA--G-U--C-UCC-----
> 2byt_E.pdb
GCCGG-GGUGGCGGA-AU-GGGU--AGACGCGCA-UGAC_AUCAUGUGCGCAA-----GCGU-GCGGGU      UC-A--AG--U-C-CC--G-C-CC-CCGGCA-CCA
> 2byt_B.pdb
GCCGG-GGUGGCGGA-AU-GGGU--AGACGCGCA-UGAC_AUCAUGUGCGCAA-----GCGU-GCGGGU      UC-A--AG--U-C-CC--G-C-CC-CCGGCA-CCA

```

```

> 2bte_B.pdb
GCCGG-GGUGGCCGGA-AU-GGGU--AGACGCGCA-UGAC_AUCAUGUGCGCAA-----GCGU-GCGGGU UC-A--AG--U-C-CC--G-C-CC-CCGGCA-CCA
> 2v0g_B.pdb
GCCGG-GGUGGCCGAA--_GGUA--GAC-GCGCA-UGAC_AUCAUGUGCGCAA-----G-CGUGCGGGU UC-A--AG--U-C-CC--G-C-CC-CCGGCA-CCA
> 2v0g_F.pdb
GCCGG-GGUGGCCGAA--_GGUA--GAC-GCGCA-UGAC_AUCAUGUGCGCAA-----G-CGUGCGGGU UC-A--AG--U-C-CC--G-C-CC-CCGGCA-CCA
> 1euq_B.pdb
G-GGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGC-----AG-CGAGGU UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 2bte_E.pdb
GCCGG-GGUGGCCGGA-AU-GGGU--AGACGCGCA-UGAC_AUCAUGUGCGCAA-----GCGU-GCGGGU UC-A--AG--U-C-CC--G-C-CC-CCGGCA-CCA
> 1euy_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGGCA-----A--G-CGAGGU UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 2ow8_0.pdb
GCGGA-UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAAAUCUGGAG-----G-UC-CUGUGU PC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 1exd_B.pdb
-GGGG-UAUCGCCAAGC---GGU--AAG-GCACC-GGAU-UCUGAUUCCGG-A-----G-GU-CGAGGU UC-G--AA--U-C-CU--C-G-UA-CCCCAG-CCA
> 1j2b_C.pdb
GGGCC-CGUGGUCUAGUU-GGUC--AUG-ACGCC-GCCC-UUACGAGGCGGAG-----G-UC-CGGGGU UC-A--AG--U-C-CC--C-G-CG-GGCCCA-CCA
> 1j2b_D.pdb
GGGCC-CGUGGUCUAGUU_-----G-ACGCC-GCCC-UUACGAGGCGGAG-----G-UC-CGGGGU UC-A--AG--U-C-CC--C-G-CG-GGCCCA-CCA
> 2csx_C.pdb
GGCGG-CGUAGCUCAGC--UGGU-CAGA-GCGGG-GAUC-UCAUAAGUCCAG-----G-UC-GGAGGU UC-G--AG--U-C-CU--C-C-CG-CCGCCA-C--
> 2csx_D.pdb
GGCGG-CGUAGCUCAGC--UGGU-CAGA-GCGGG-GAUC-UCAUAAGUCCAG-----G-UC-GGAGGU UC-G--AG--U-C-CU--C-C-CG-CCGCCA-C--
> 1efw_D.pdb
GGAGC-GG4AGUUCAGD--CGGD-DAGA-AUACC-UGCC-UQUC/CGCAGGGG-----7-UC-GCGGG018UPC-G--AG--U-C-CC--G-P-CC-GUUC-----
> 2qnh_2.pdb
GCGGA-UUUAGCUCAGU--UGGG--AGA-GCGCC-AGAC-UGAAAAUCUGGAG-----G-UC-CUGUGU PC-G--AU--C-C-AC--A-G-AA-UUCGCA-CCA
> 2d6f_F.pdb
AGUCC-CGUGGGGU--AG-UGGU--AAU-CCUGCUGGGC-UUUGGACCCG-GC-----G-AC-AGCGGU UC-G--AC--U-C-CG--C-U-CG-GGACUA-CC-
> 2d6f_E.pdb
AGUCC-CGUGGGGU--AG-UGGU--AAU-CCUGCUGGGC-UUUGGACCCG-GC-----G-AC-AGCGGU UC-G--AC--U-C-CG--C-U-CG-GGACUA-CC-
> 2du5_D.pdb
--GCC-AGGGUGGCAGA---GGGGCUUU-GCGGC-GGAC-UUCAGAUCCGCUUU-----A--C-CCCGGU UC-G--AA--U-C-CG--G-G-CC-CUG-GC----
> 2du6_D.pdb
--GCC-AGGGUGGCAGA---GGGGCUUU-GCGGC-GGAC-UUCAGAUCCGCUUU-----A--C-CCCGGU UC-G--AA--U-C-CG--G-G-CC-CUG-GC----
> 1h3e_B.pdb
GGGCA-GGUUCCCGAG.---GGCCAAAG-GGGAC-GGUC-UGPAAAACCGUUGGCGU-A.GCCUU--C-GCUGGT PC-G--"A--U-C-CA--G-C-C--CUGCCC-A--

```

Bibliografia

- Adamiak, D.A., Milecki, J., Adamiak, R. and Rypniewski, W. (2009) The hydration and unusual hydrogen bonding in the crystal structure of an RNA duplex containing alternating CG base pairs, *New Journal of Chemistry*, 35, 903-909.
- Agris, P.F. (2008) Bringing order to translation: the contributions of transfer RNA anticodon-domain modifications, *EMBO Rep*, 9, 629-635.
- Anfinsen, C.B. (1973) Principles that govern the folding of protein chains, *Science*, 181, 223-230.
- Bentley, J.L. (1975) Multidimensional binary search trees used for associative searching, *Communications of the ACM*, 18, 509-517.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000) The Protein Data Bank, *Nucleic Acids Res*, 28, 235-242.
- Berman, H.M., Westbrook, J., Feng, Z., Iype, L., Schneider, B. and Zardecki, C. (2002) The Nucleic Acid Database, *Acta Crystallogr D Biol Crystallogr*, 58, 889-898.
- Biegeleisen, K. (2006) The probable structure of the protamine-DNA complex, *J Theor Biol*, 241, 533-540.
- Bindewald, E., Hayes, R., Yingling, Y.G., Kasprzak, W. and Shapiro, B.A. (2008) RNAJunction: a database of RNA junctions and kissing loops for three-dimensional structural analysis and nanodesign, *Nucleic Acids Res*, 36, D392-397.
- Boomsma, W. and Hamelryck, T. (2005) Full cyclic coordinate descent: solving the protein loop closure problem in Calpha space, *BMC Bioinformatics*, 6, 159.
- Budkevich, T., Giesebrecht, J., Altman, R.B., Munro, J.B., Mielke, T., Nierhaus, K.H., Blanchard, S.C. and Spahn, C.M. (2011) Structure and dynamics of the mammalian ribosomal pretranslocation complex, *Mol Cell*, 44, 214-224.
- Cao, S. and Chen, S.J. (2011) Physics-based de novo prediction of RNA 3D structures, *J Phys Chem B*, 115, 4216-4226.
- Capriotti, E. and Marti-Renom, M.A. (2010) Quantifying the relationship between sequence and three-dimensional structure conservation in RNA, *BMC Bioinformatics*, 11, 322.
- Cate, J.H., Gooding, A.R., Podell, E., Zhou, K., Golden, B.L., Kundrot, C.E., Cech, T.R. and Doudna, J.A. (1996) Crystal structure of a group I ribozyme domain: principles of RNA packing, *Science*, 273, 1678-1685.
- Chao, J.A., Patskovsky, Y., Almo, S.C. and Singer, R.H. (2008) Structural basis for the coevolution of a viral RNA-protein complex, *Nat Struct Mol Biol*, 15, 103-105.

- Chen, L.J., Qian, H.J., Lu, Z.Y., Li, Z.S. and Sun, C.C. (2006) An automatic coarse-graining and fine-graining simulation method: application on polyethylene, *J Phys Chem B*, 110, 24093-24100.
- Chothia, C. and Lesk, A.M. (1986) The relation between the divergence of sequence and structure in proteins, *EMBO J*, 5, 823-826.
- Chworos, A., Severcan, I., Koyfman, A.Y., Weinkam, P., Oroudjev, E., Hansma, H.G. and Jaeger, L. (2004) Building programmable jigsaw puzzles with RNA, *Science*, 306, 2068-2072.
- Cock, P.J., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. and de Hoon, M.J. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics, *Bioinformatics*, 25, 1422-1423.
- Cohen-Gonsaud, M., Catherinot, V., Labesse, G. and Douguet, D. (2004) From molecular modeling to drug design. In Bujnicki, J.M. (ed), *Practical Bioinformatics*. Springer-Verlag, Berlin, 35-71.
- Cornish, P.V., Stammler, S.N. and Giedroc, D.P. (2006) The global structures of a wild-type and poorly functional plant luteoviral mRNA pseudoknot are essentially identical, *RNA*, 12, 1959-1969.
- Cozzetto, D., Giorgetti, A., Raimondo, D. and Tramontano, A. (2007) The Evaluation of Protein Structure Prediction Results, *Mol Biotechnol*.
- Cruz, J.A., Blanchet, M.F., Boniecki, M., Bujnicki, J.M., Chen, S.J., Cao, S., Das, R., Ding, F., Dokholyan, N.V., Flores, S.C., Huang, L., Lavender, C.A., Lisi, V., Major, F., Mikolajczak, K., Patel, D.J., Philips, A., Puton, T., Santalucia, J., Sijenyi, F., Hermann, T., Rother, K., Rother, M., Serganov, A., Skorupski, M., Soltysinski, T., Sripakdeevong, P., Tuszyńska, I., Weeks, K.M., Waldsich, C., Wildauer, M., Leontis, N.B. and Westhof, E. (2012) RNA-Puzzles: A CASP-like evaluation of RNA three-dimensional structure prediction, *RNA*.
- Cruz, J.A. and Westhof, E. (2009) The dynamic landscapes of RNA architecture, *Cell*, 136, 604-609.
- Czerwoniec, A., Dunin-Horkawicz, S., Purta, E., Kaminska, K.H., Kasprzak, J.M., Bujnicki, J.M., Grosjean, H. and Rother, K. (2009) MODOMICS: a database of RNA modification pathways. 2008 update, *Nucleic Acids Res*, 37, D118-121.
- Dalluge, J.J., Hashizume, T., Sopchik, A.E., McCloskey, J.A. and Davis, D.R. (1996) Conformational flexibility in RNA: the role of dihydrouridine, *Nucleic Acids Res*, 24, 1073-1079.
- Das, R. and Baker, D. (2007) Automated de novo prediction of native-like RNA tertiary structures, *Proc Natl Acad Sci U S A*, 104, 14664-14669.
- Das, R., Karanicolas, J. and Baker, D. (2010) Atomic accuracy in predicting and designing noncanonical RNA structure, *Nat Methods*, 7, 291-294.
- Davis, I.W., Leaver-Fay, A., Chen, V.B., Block, J.N., Kapral, G.J., Wang, X., Murray, L.W., Arendall, W.B., 3rd, Snoeyink, J., Richardson, J.S. and Richardson, D.C. (2007) MolProbity: all-atom contacts and structure validation for proteins and nucleic acids, *Nucleic Acids Res*, 35, W375-383.

- DeLano, W.L. (2002) The PyMOL Molecular Graphics System. DeLano Scientific, San Carlos, CA, USA, <http://www.pymol.org>.
- Ding, F., Sharma, S., Chalasani, P., Demidov, V.V., Broude, N.E. and Dokholyan, N.V. (2008) Ab initio RNA folding by discrete molecular dynamics: from structure prediction to folding mechanisms, *RNA*, 14, 1164-1173.
- Donghi, D. and Schnabl, J. (2011) Multiple roles of metal ions in large ribozymes, *Met Ions Life Sci*, 9, 197-234.
- Dowell, R.D. and Eddy, S.R. (2006) Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints, *BMC Bioinformatics*, 7, 400.
- Dror, O., Nussinov, R. and Wolfson, H. (2005) ARTS: alignment of RNA tertiary structures, *Bioinformatics*, 21 Suppl 2, ii47-ii53.
- Dunin-Horkawicz, S., Czerwoniec, A., Gajda, M.J., Feder, M., Grosjean, H. and Bujnicki, J.M. (2006) MODOMICS: a database of RNA modification pathways, *Nucleic Acids Res*, 34, D145-149.
- Erdmann, V.A.C., Brosius, J. and Barciszewski, J. (2006) *RNA Towards Medicine*. Springer, Berlin Heidelberg.
- Felden, B. (2007) RNA structure: experimental analysis, *Curr Opin Microbiol*, 10, 286-291.
- Flores, S.C. and Altman, R.B. (2010) Turning limited experimental information into 3D models of RNA, *RNA*, 16, 1769-1778.
- Flores, S.C., Wan, Y., Russell, R. and Altman, R.B. (2010) Predicting RNA structure by multiple template homology modeling, *Pac Symp Biocomput*, 216-227.
- Gardner, P.P., Daub, J., Tate, J.G., Nawrocki, E.P., Kolbe, D.L., Lindgreen, S., Wilkinson, A.C., Finn, R.D., Griffiths-Jones, S., Eddy, S.R. and Bateman, A. (2009) Rfam: updates to the RNA families database, *Nucleic Acids Res*, 37, D136-140.
- Garst, A.D., Edwards, A.L. and Batey, R.T. (2011) Riboswitches: structures and mechanisms, *Cold Spring Harb Perspect Biol*, 3, 1-13.
- Geissmann, T., M., P., Huntzinger, E., Fechter, P., Ehresmann, C. and Romby, P. (2006) Regulatory RNAs as Mediators of Virulence Gene Expression in Bacteria. In Erdmann, V.A.C., Brosius, J. and Barciszewski, J. (eds), *RNA Towards Medicine*. Springer, Berlin Heidelberg, 9-43.
- Gendron, P., Lemieux, S. and Major, F. (2001) Quantitative analysis of nucleic acid three-dimensional structures, *J Mol Biol*, 308, 919-936.
- Gherghe, C.M., Leonard, C.W., Ding, F., Dokholyan, N.V. and Weeks, K.M. (2009) Native-like RNA tertiary structures using a sequence-encoded cleavage agent and refinement by discrete molecular dynamics, *J Am Chem Soc*, 131, 2541-2546.
- Gilbert, S.D., Reyes, F.E., Edwards, A.L. and Batey, R.T. (2009) Adaptive ligand binding by the purine riboswitch in the recognition of guanine and adenine analogs, *Structure*, 17, 857-868.

- Golas, M.M., Sander, B., Bessonov, S., Grote, M., Wolf, E., Kastner, B., Stark, H. and Luhrmann, R. (2010) 3D cryo-EM structure of an active step I spliceosome and localization of its catalytic core, *Mol Cell*, 40, 927-938.
- Golub, G.H. and Van Loan, C.F. (1989) *Matrix Computations*. Johns Hopkins, Baltimore.
- Grosjean, H. (2009) *Fine-Tuning of RNA Functions by Modification and Editing*. Springer, Berlin Heidelberg.
- Guerrier-Takada, C., Gardiner, K., Marsh, T., Pace, N. and Altman, S. (1983) The RNA moiety of ribonuclease P is the catalytic subunit of the enzyme, *Cell*, 35, 849-857.
- Hainzl, T., Huang, S. and Sauer-Eriksson, A.E. (2002) Structure of the SRP19 RNA complex and implications for signal recognition particle assembly, *Nature*, 417, 767-771.
- Hamelryck, T. and Manderick, B. (2003) PDB file parser and structure class implemented in Python, *Bioinformatics*, 19, 2308-2310.
- Hinsen, K. (2000) The molecular modeling toolkit: a new approach to molecular simulations, *J Comp Chem*, 21, 79-85.
- Holbrook, S.R. (2008) Structural principles from large RNAs, *Annu Rev Biophys*, 37, 445-464.
- Holmes, I. (2005) Accelerated probabilistic inference of RNA structure evolution, *BMC Bioinformatics*, 6, 73.
- Hoogstraten, C.G. and Sumita, M. (2007) Structure-function relationships in RNA and RNP enzymes: recent advances, *Biopolymers*, 87, 317-328.
- Jacobson, M. and Sali, A. (2004) Comparative Protein Structure Modeling and Its Applications to Drug Discovery. In Overington, J. (ed), *Annual Reports in Medicinal Chemistry*. London.
- Jonikas, M.A., Radmer, R.J., Laederach, A., Das, R., Pearlman, S., Herschlag, D. and Altman, R.B. (2009) Coarse-grained modeling of large RNA molecules with knowledge-based potentials and structural filters, *RNA*, 15, 189-199.
- Jossinet, F., Ludwig, T.E. and Westhof, E. (2010) Assemble: an interactive graphical tool to analyze and build RNA architectures at the 2D and 3D levels, *Bioinformatics*.
- Jossinet, F. and Westhof, E. (2005) Sequence to Structure (S2S): display, manipulate and interconnect RNA data from sequence to structure, *Bioinformatics*, 21, 3320-3321.
- Juhling, F., Morl, M., Hartmann, R.K., Sprinzl, M., Stadler, P.F. and Putz, J. (2009) tRNAdb 2009: compilation of tRNA sequences and tRNA genes, *Nucleic Acids Res*, 37, D159-162.
- Kalota, A., R., D.V. and Gewirtz, A.M. (2006) Progress in the Development of Nucleic Acid Therapeutics. In Erdmann, V.A.C., Brosius, J. and Barciszewski, J. (eds), *RNA Towards Medicine*. Springer, Berlin Heidelberg.
- Ke, A. and Doudna, J.A. (2004) Crystallization of RNA and RNA-protein complexes, *Methods*, 34, 408-414.

- Kim, S.H., Suddath, F.L., Quigley, G.J., McPherson, A., Sussman, J.L., Wang, A.H., Seeman, N.C. and Rich, A. (1974) Three-dimensional tertiary structure of yeast phenylalanine transfer RNA, *Science*, 185, 435-440.
- Klosterman, P.S., Tamura, M., Holbrook, S.R. and Brenner, S.E. (2002) SCOR: a Structural Classification of RNA database, *Nucleic Acids Res*, 30, 392-394.
- Kosinski, J., Tkaczuk, K.L., Kasprzak, J.M. and Bujnicki, J.M. (2008) Template based prediction of three-dimensional protein structures: Fold recognition and comparative modeling *Prediction of Protein Structures, Functions and Interactions*.
- Krieger, E., Nabuurs, S.B. and Vriend, G. (2003) Homology modeling, *Methods Biochem Anal*, 44, 509-523.
- Kruger, K., Grabowski, P.J., Zaug, A.J., Sands, J., Gottschling, D.E. and Cech, T.R. (1982) Self-splicing RNA: autoexcision and autocyclization of the ribosomal RNA intervening sequence of Tetrahymena, *Cell*, 31, 147-157.
- Kryshtafovych, A., Moulton, J., Bartual, S.G., Bazan, J.F., Berman, H., Casteel, D.E., Christodoulou, E., Everett, J.K., Hausmann, J., Heidebrecht, T., Hills, T., Hui, R., Hunt, J.F., Seetharaman, J., Joachimiak, A., Kennedy, M.A., Kim, C., Lingel, A., Michalska, K., Montelione, G.T., Otero, J.M., Perrakis, A., Pizarro, J.C., van Raaij, M.J., Ramelot, T.A., Rousseau, F., Tong, L., Wernimont, A.K., Young, J. and Schwede, T. Target highlights in CASP9: Experimental target structures for the critical assessment of techniques for protein structure prediction, *Proteins*, 79 Suppl 10, 6-20.
- Laing, C. and Schlick, T. (2010) Computational approaches to 3D modeling of RNA, *J Phys Condens Matter*, 22, 283101.
- Lavender, C.A., Ding, F., Dokholyan, N.V. and Weeks, K.M. (2010) Robust and generic RNA modeling using inferred constraints: a structure for the hepatitis C virus IRES pseudoknot domain, *Biochemistry*, 49, 4931-4933.
- Lemieux, S. and Major, F. (2002) RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire, *Nucleic Acids Res*, 30, 4250-4263.
- Leontis, N.B., Stombaugh, J. and Westhof, E. (2002) The non-Watson-Crick base pairs and their associated isostericity matrices, *Nucleic Acids Res*, 30, 3497-3531.
- Leontis, N.B. and Westhof, E. (1998) Conserved geometrical base-pairing patterns in RNA, *Q Rev Biophys*, 31, 399-455.
- Lescoute, A., Leontis, N.B., Massire, C. and Westhof, E. (2005) Recurrent structural RNA motifs, Isostericity Matrices and sequence alignments, *Nucleic Acids Res*, 33, 2395-2409.
- Lorenz, R., Bernhart, S.H., Hoener Zu Siederdisen, C., Tafer, H., Flamm, C., Stadler, P.F. and Hofacker, I.L. ViennaRNA Package 2.0, *Algorithms Mol Biol*, 6, 26.
- Martick, M., Lee, T.S., York, D.M. and Scott, W.G. (2008) Solvent structure and hammerhead ribozyme catalysis, *Chem Biol*, 15, 332-342.

- Martinez, H.M., Maizel, J.V., Jr. and Shapiro, B.A. (2008) RNA2D3D: a program for generating, viewing, and comparing 3-dimensional models of RNA, *J Biomol Struct Dyn*, 25, 669-683.
- Mastroiannopoulos, N.P., Uney, J.B. and Phylactou, L.A. (2010) The application of ribozymes and DNAzymes in muscle and brain, *Molecules*, 15, 5460-5472.
- Matsumoto, S., Akashi, H. and Taira, K. (2006) Screening and Determination of Gene Function Using Randomized Ribozyme and siRNA Libraries. In Erdmann, V.A.C., Brosius, J. and Barciszewski, J. (eds), *RNA Towards Medicine*. Springer, Berlin Heidelberg.
- McCloskey, J.A. and Rozenski, J. (2005) The Small Subunit rRNA Modification Database, *Nucleic Acids Res*, 33, D135-138.
- Michalsky, E., Goede, A. and Preissner, R. (2003) Loops In Proteins (LIP)-a comprehensive loop database for homology modelling, *Protein Eng*, 16, 979-985.
- Montange, R.K. and Batey, R.T. (2008) Riboswitches: emerging themes in RNA structure and function, *Annu Rev Biophys*, 37, 117-133.
- Motorin, Y. and Helm, M. (2010) tRNA stabilization by modified nucleotides, *Biochemistry*, 49, 4934-4944.
- Moult, J. (2005) A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction, *Curr Opin Struct Biol*, 15, 285-289.
- Moult, J., Pedersen, J.T., Judson, R. and Fidelis, K. (1995) A large-scale experiment to assess protein structure prediction methods, *Proteins*, 23, ii-v.
- Mueller, F. (1998) ERNA-3D, <http://www.rna123.com>.
- Mulhbachter, J., Brouillette, E., Allard, M., Fortier, L.C., Malouin, F. and Lafontaine, D.A. (2010) Novel riboswitch ligand analogs as selective inhibitors of guanine-related metabolic pathways, *PLoS Pathog*, 6.
- Mulhbachter, J., St-Pierre, P. and Lafontaine, D.A. (2010) Therapeutic applications of ribozymes and riboswitches, *Curr Opin Pharmacol*, 10, 551-556.
- Murray, J.M. and Carr, A.M. (2008) Smc5/6: a link between DNA repair and unidirectional replication?, *Nat Rev Mol Cell Biol*, 9, 177-182.
- Musielak, M., Rother, K., Puton, T. and Bujnicki, J.M. (2010) ModeRNA builds RNA 3D Models from Template Structures, *ERCIM News*, 82, 19-20.
- Otto, W., Will, S. and Backofen, R. (2008) Structure local multiple alignment of RNA. *GCB'2008*. Germany, 178-188.
- Parisien, M., Cruz, J.A., Westhof, E. and Major, F. (2009) New metrics for comparing and assessing discrepancies between RNA 3D structures and models, *RNA*, 15, 1875-1885.

- Parisien, M. and Major, F. (2008) The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data, *Nature*, 452, 51-55.
- Pley, H.W., Flaherty, K.M. and McKay, D.B. (1994) Three-dimensional structure of a hammerhead ribozyme, *Nature*, 372, 68-74.
- Popenda, L., Adamiak, R.W. and Gdaniec, Z. (2008) Bulged adenosine influence on the RNA duplex conformation in solution, *Biochemistry*, 47, 5059-5067.
- Popenda, M., Bielecki, L. and Adamiak, R.W. (2006) High-throughput method for the prediction of low-resolution, three-dimensional RNA structures, *Nucleic Acids Symp Ser (Oxf)*, 67-68.
- Popenda, M., Blazewicz, M., Szachniuk, M. and Adamiak, R.W. (2008) RNA FRABASE version 1.0: an engine with a database to search for the three-dimensional fragments within RNA structures, *Nucleic Acids Res*, 36, D386-391.
- Popenda, M., Milecki, J. and Adamiak, R.W. (2004) High salt solution structure of a left-handed RNA double helix, *Nucleic Acids Res*, 32, 4044-4054.
- Popenda, M., Szachniuk, M., Blazewicz, M., Wasik, S., Burke, E.K., Blazewicz, J. and Adamiak, R.W. (2010) RNA FRABASE 2.0: an advanced web-accessible database with the capacity to search the three-dimensional fragments within RNA structures, *BMC Bioinformatics*, 11, 231.
- Purzycka, K.J. and Adamiak, R.W. (2008) The HIV-2 TAR RNA domain as a potential source of viral-encoded miRNA. A reconnaissance study, *Nucleic Acids Symp Ser (Oxf)*, 511-512.
- Purzycka, K.J., Pachulska-Wieczorek, K. and Adamiak, R.W. (2011) The in vitro loose dimer structure and rearrangements of the HIV-2 leader RNA, *Nucleic Acids Res*, 39, 7234-7248.
- Reiter, N.J., Osterman, A., Torres-Larios, A., Swinger, K.K., Pan, T. and Mondragon, A. (2010) Structure of a bacterial ribonuclease P holoenzyme in complex with tRNA, *Nature*, 468, 784-789.
- Robertus, J.D., Ladner, J.E., Finch, J.T., Rhodes, D., Brown, R.S., Clark, B.F. and Klug, A. (1974) Structure of yeast phenylalanine tRNA at 3 Å resolution, *Nature*, 250, 546-551.
- Rocca-Serra, P., Bellaousov, S., Birmingham, A., Chen, C., Cordero, P., Das, R., Davis-Neulander, L., Duncan, C.D., Halvorsen, M., Knight, R., Leontis, N.B., Mathews, D.H., Ritz, J., Stombaugh, J., Weeks, K.M., Zirbel, C.L. and Laederach, A. (2011) Sharing and archiving nucleic acid structure mapping data, *RNA*, 17, 1204-1212.
- Rosenberg, J.M., Seeman, N.C., Day, R.O. and Rich, A. (1976) RNA double-helical fragments at atomic resolution. II. The crystal structure of sodium guanylyl-3',5'-cytidine nonahydrate, *J Mol Biol*, 104, 145-167.
- Rother, K., Hoffmann, S., Bulik, S., Hoppe, A., Gasteiger, J. and Holzhutter, H.G. (2010) IGERS: inferring Gibbs energy changes of biochemical reactions from reaction similarities, *Biophys J*, 98, 2478-2486.

- Rother, K., Potrzebowski, W., Puton, T., Rother, M., Wywiał, E. and Bujnicki, J.M. (2011) A toolbox for developing bioinformatics software, *Brief Bioinform*, 1-14.
- Rother, K., Rother, M., Boniecki, M., Puton, T. and Bujnicki, J.M. (2011) RNA and protein 3D structure modeling: similarities and differences, *J Mol Model*, doi: 10.1007/s00894-010-0951-x.
- Rother, K., Rother, M., Boniecki, M., Puton, T., Tomala, K., Lukasz, P. and Bujnicki, J.M. (2012) Templated-based and template free modeling of RNA 3D structure: Inspirations from protein structure modeling. In Leontis, N.B. and Westhof, E. (eds), *RNA Structure Prediction and Modeling*. Springer, Berlin Heidelberg.
- Rother, M., Milanowska, K., Puton, T., Jeleniewicz, J., Rother, K. and Bujnicki, J.M. (2011) ModeRNA server: an online tool for modeling RNA 3D structures, *Bioinformatics*, 1, 2441-2442.
- Rother, M., Rother, K., Puton, T. and Bujnicki, J.M. (2011) ModeRNA: a tool for comparative modeling of RNA 3D structure, *Nucleic Acids Res*, 39, 4007-4022.
- Rother, M., Rother, K., Puton, T. and Bujnicki, J.M. (2011) RNA tertiary structure prediction with ModeRNA, *Brief Bioinform*, 12, 601-613.
- Sali, A. and Blundell, T.L. (1993) Comparative protein modelling by satisfaction of spatial restraints, *J Mol Biol*, 234, 779-815.
- Salmena, L., Poliseno, L., Tay, Y., Kats, L. and Pandolfi, P.P. (2011) A ceRNA hypothesis: the Rosetta Stone of a hidden RNA language?, *Cell*, 146, 353-358.
- SantaLucia, J., Jr. (2002) RNA123.
- Sarver, M., Zirbel, C.L., Stombaugh, J., Mokdad, A. and Leontis, N.B. (2008) FR3D: finding local and composite recurrent structural motifs in RNA 3D structures, *J Math Biol*, 56, 215-252.
- Sayers, E.W., Barrett, T., Benson, D.A., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., DiCuccio, M., Edgar, R., Federhen, S., Feolo, M., Geer, L.Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D.J., Madden, T.L., Maglott, D.R., Miller, V., Mizrachi, I., Ostell, J., Pruitt, K.D., Schuler, G.D., Sequeira, E., Sherry, S.T., Shumway, M., Sirotkin, K., Souvorov, A., Starchenko, G., Tatusova, T.A., Wagner, L., Yaschenko, E. and Ye, J. (2009) Database resources of the National Center for Biotechnology Information, *Nucleic Acids Res*, 37, D5-15.
- Schudoma, C., May, P., Nikiforova, V. and Walther, D. (2010) Sequence-structure relationships in RNA loops: establishing the basis for loop homology modeling, *Nucleic Acids Res*, 38, 970-980.
- Schudoma, C., May, P. and Walther, D. (2010) Modeling RNA loops using sequence homology and geometric constraints, *Bioinformatics*, 26, 1671-1672.
- Schwede, T., Kopp, J., Guex, N. and Peitsch, M.C. (2003) SWISS-MODEL: An automated protein homology-modeling server, *Nucleic Acids Res*, 31, 3381-3385.

- Scott, W.G., Finch, J.T. and Klug, A. (1995) The crystal structure of an all-RNA hammerhead ribozyme: a proposed mechanism for RNA catalytic cleavage, *Cell*, 81, 991-1002.
- Scott, W.G., Martick, M. and Chi, Y.I. (2009) Structure and function of regulatory RNA elements: ribozymes that regulate gene expression, *Biochim Biophys Acta*, 1789, 634-641.
- Seeman, N.C., Rosenberg, J.M., Suddath, F.L., Kim, J.J. and Rich, A. (1976) RNA double-helical fragments at atomic resolution. I. The crystal and molecular structure of sodium adenylyl-3',5'-uridine hexahydrate, *J Mol Biol*, 104, 109-144.
- Sharma, S., Ding, F. and Dokholyan, N.V. (2008) iFoldRNA: three-dimensional RNA structure prediction and folding, *Bioinformatics*, 24, 1951-1952.
- Simonovic, M. and Steitz, T.A. (2009) A structural view on the mechanism of the ribosome-catalyzed peptide bond formation, *Biochim Biophys Acta*, 1789, 612-623.
- Sioud, M. (2006) Ribozymes and siRNAs: From Structure to Preclinical Applications. In Erdmann, V.A.C., Brosius, J. and Barciszewski, J. (eds), *RNA Towards Medicine*. Springer, Berlin Heidelberg.
- Smith, J.K., Hsieh, J. and Fierke, C.A. (2007) Importance of RNA-protein interactions in bacterial ribonuclease P structure and catalysis, *Biopolymers*, 87, 329-338.
- Stombaugh, J., Zirbel, C.L., Westhof, E. and Leontis, N.B. (2009) Frequency and isostericity of RNA base pairs, *Nucleic Acids Res*, 37, 2294-2312.
- Tan, R.K.Z., Petrov, A.S. and Harvey, S.C. (2006) YUP: A molecular simulation program for coarse-grained and multiscaled models, *J Chem. Theory Comput.*, 2, 529-540.
- Thomas, B. and Akoulitchev, A.V. (2006) Mass spectrometry of RNA, *Trends Biochem Sci*, 31, 173-181.
- Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic.Acids.Res.*, 22, 4673-4680.
- Torarinsson, E., Havgaard, J.H. and Gorodkin, J. (2007) Multiple structural alignment and clustering of RNA sequences, *Bioinformatics*, 23, 926-932.
- Ullman, J.D. (1976) An Algorithm for Subgraph Isomorphism, *Journal of the ACM*, 23, 31-42.
- Ulyanov, N.B. and James, T.L. (2010) RNA structural motifs that entail hydrogen bonds involving sugar-phosphate backbone atoms of RNA, *New J Chem*, 34, 910-917.
- Valadkhan, S. (2010) Role of the snRNAs in spliceosomal active site, *RNA Biol*, 7, 345-353.
- Vanegas, P.L., Hudson, G.A., Davis, A.R., Kelly, S.C., Kirkpatrick, C.C. and Znosko, B.M. (2012) RNA CoSSMos: Characterization of Secondary Structure Motifs--a searchable database of secondary structure motifs in RNA three-dimensional structures, *Nucleic Acids Res*, 40, D439-444.

- Vicens, Q. and Cech, T.R. (2006) Atomic level architecture of group I introns revealed, *Trends Biochem Sci*, 31, 41-51.
- Weeks, K.M. (2010) Advances in RNA structure analysis by chemical probing, *Curr Opin Struct Biol*, 20, 295-304.
- Weininger, D. (1988) SMILES, a chemical language and information system, *J Chem Inf Comput Sci*, 28, 31-36.
- Westhof, E. and Auffinger, P. (2000) RNA Tertiary Structure. In Meyers, R.A. (ed), *Encyclopedia of Analytical Chemistry*. John Wiley & Sons Ltd, Chichester, 5222–5232.
- Wheeler, T.M., Sobczak, K., Lueck, J.D., Osborne, R.J., Lin, X., Dirksen, R.T. and Thornton, C.A. (2009) Reversal of RNA dominance by displacement of protein sequestered on triplet repeat RNA, *Science*, 325, 336-339.
- Wilm, A., Higgins, D.G. and Notredame, C. (2008) R-Coffee: a method for multiple alignment of non-coding RNA, *Nucleic Acids Res*, 36, e52.
- Wilson, T.J. and Lilley, D.M. (2009) Biochemistry. The evolution of ribozyme chemistry, *Science*, 323, 1436-1438.
- Yang, H., Jossinet, F., Leontis, N., Chen, L., Westbrook, J., Berman, H. and Westhof, E. (2003) Tools for the automatic identification and classification of RNA base pairs, *Nucleic Acids Res*, 31, 3450-3460.
- Yang, X.L., Otero, F.J., Ewalt, K.L., Liu, J., Swairjo, M.A., Kohrer, C., Rajbhandary, U.L., Skene, R.J., McRee, D.E. and Schimmel, P. (2006) Two conformations of a crystalline human tRNA synthetase-tRNA complex: implications for protein synthesis, *EMBO J*, 25, 2919–2929.
- Zhang, Q., Kim, N.K. and Feigon, J. Architecture of human telomerase RNA, *Proc Natl Acad Sci U S A*, 108, 20325-20332.
- Zidek, L., Stefl, R. and Sklenar, V. (2001) NMR methodology for the study of nucleic acids, *Curr Opin Struct Biol*, 11, 275-281.
- Zwieb, C. and Muller, F. (1997) Three-dimensional comparative modeling of RNA, *Nucleic Acids Symp Ser*, 69-71.

Spis rycin

Rycina 1. Dylemat pochodzenia życia na Ziemi (co było pierwsze: DNA czy białko?) i jego rozwiązanie – teoria świata RNA.....	1
Rycina 2. Reszty nukleotydów budujące RNA: A) fosforan 5'-adenozyny; B) fosforan 5'-guanozyny; C) fosforan 5'-cytydyny; D) fosforan 5'-urydyny. Zaznaczone krawędzie wskazują potencjalne miejsca oddziaływań z innymi resztami. W lewym dolnym rogu znajduje się informacja o pochodzeniu każdej z reszt (identyfikator reszty i struktury).	4
Rycina 3. Oddziaływania pomiędzy zasadami w RNA: A) oddziaływanie warstwowe; B) parowanie kanoniczne typu Watsona-Cricka; C) parowanie niekanoniczne (czerwone linie) w motywie A-minor.	5
Rycina 4. Podstawowe typy motywów drugorzędowych na przykładzie struktur RNA: A) operatora translacji 2QUX (Chao, i wsp., 2008); B) pseudowęzła z mRNA roślinnego luteowirusa 2AP5 (Cornish, i wsp., 2006); C) SRP19 archeowca <i>Methanococcus jannaschii</i> (Hainzl, i wsp., 2002) (struktura drugorzędowa na podstawie (Laing and Schlick, 2010)).....	7
Rycina 5. Przykłady funkcjonalnych RNA posiadających stabilną strukturę trzeciorzędową: A) rybozym typu „głowa młotka” (kod PDB: 2OEU) (Martick, i wsp., 2008); B) ryboprzełącznik guaninowy (kod PDB: 3FO6) (Gilbert, i wsp., 2009); C) domena P4-P6 intronu grupy I (kod PDB: 1GID) (Cate, i wsp., 1996); D) rybonukleaza P (kod PDB: 3Q1Q) (Reiter, i wsp., 2010).....	8
Rycina 6. Modyfikowane nukleotydy występujące w tRNA na przykładzie struktury 1EHZ: A) rozmieszczenie modyfikowanych nukleotydów w strukturze drugorzędowej; B) rozmieszczenie modyfikowanych nukleotydów w strukturze trzeciorzędowej; C-E) konformacja fragmentów struktury zawierających najbardziej konserwowane modyfikacje; F-I) struktury pojedynczych modyfikowanych nukleotydów. Na podstawie (Rother, i wsp., 2011).	10
Rycina 7. Hierarchiczna budowa RNA i białek, na podstawie (Rother, i wsp., 2011).	12
Rycina 8. Dwa główne podejścia do modelowania komputerowego struktury trzeciorzędowej RNA: A) modelowanie bez użycia szablonu; B) modelowanie z wykorzystaniem struktury szablonu. Na podstawie (Rother, i wsp., 2011).....	22
Rycina 9. Kroki niezbędne do zbudowania modelu homologicznego. Schemat przygotowany na podstawie: (Kosinski, i wsp., 2008).	26
Rycina 10. Głównym celem pracy jest program, który pozwoli na budowanie trójwymiarowych modeli RNA na podstawie struktury szablonu.	27
Rycina 11. Język programowania Python i podejście modelowania homologicznego zostały wykorzystane do napisania programu, który buduje modele przestrzenne RNA na podstawie szablonu i przyrównania.	31

Rycina 12. Przyrównanie sekwencji celu (górna) i szablonu (dolna) z zaznaczonymi głównymi typami operacji wykonywanymi podczas modelowania homologicznego.	33
Rycina 13. Hierarchiczna struktura obiektów reprezentujących cząsteczkę w bibliotece Biopython.PDB.	37
Rycina 14. Kroki niezbędne do zbudowania modelu homologicznego z wyróżnionymi operacjami wykonywanymi przez program ModeRNA oraz interfejs internetowy programu.	44
Rycina 15. Operacja zastąpienia uracylu przez adeninę.	53
Rycina 16. Przykłady fragmentów używanych przy dodawaniu modyfikacji. Na biało zaznaczono atomy wykorzystywane do nałożenia fragmentu na niemodyfikowany nukleotyd.	54
Rycina 17. Operacja dodania grupy metylowej do rybozy reszty guanozyny i zamiany guanozyny na 2'-O-metyloguanozynę.	55
Rycina 18. Operacja usunięcia modyfikacji i zastąpienia N6-metylo-N6-trenylokarbomyloadenozynej przez adenozynej.	56
Rycina 19. Operacja wstawienia fragmentu strukturalnego o długości 9 reszt nukleotydowych do modelu.	59
Rycina 20. Odległości pomiędzy atomami używane jako wstępne kryterium dopasowania przy wyszukiwaniu fragmentów do wstawienia między dwie wskazane reszty (reszty kotwiczące).	60
Rycina 21. Przypadki, w których niezbędne jest redagowanie całych fragmentów struktury drugorzędowej na przykładzie tRNA.	64
Rycina 22. Operacje na strukturze drugorzędowej dostępne w programie ModeRNA: A-B) zmiana długości helisy; C) wstawienie fragmentu dwuniciowego, posiadającego cztery reszty kotwiczące; D) wyszukanie i wstawienie fragmentu o zadanej strukturze drugorzędowej i dwóch resztach kotwiczących; E) dodanie nukleotydu tak, by utworzył helisę z zadany nukleotydem z komplementarnej nici.	65
Rycina 23. Reszty nukleotydów wskazane jako niepoprawne w modelu tRNA ^{Thr} zbudowanym na szablonie 1EHZ: A) nietypowa długość wiązania N-glikozydowego - 1.38 Å w reszcie 7; B) nietypowa wartość kąta torsyjnego pomiędzy atomami O3' (reszty 15) - P (reszty 15) - O5' (reszty 16) - C5' (reszty 16); C) Zderzenie steryczne między atomami zasad reszt 27 i 43. Przez wartości nietypowe rozumiane są takie, które odbiegają od najczęściej występujących w strukturach w bazie PDB.	69
Rycina 24. Różne tryby korzystania z program ModeRNA na przykładzie najważniejszych operacji.	73
Rycina 25. Operacje dostępne przez serwer ModeRNA (interfejs internetowy programu).	81
Rycina 26. Architektura programu ModeRNA. Biblioteka zawiera oprócz pakietów (katalogi) i modułów (pliki .py) pliki z danymi w formacie tekstowym i PDB. Po lewej stronie został przedstawiony przykładowy moduł (SearchLIR) i zawarte w nim klasy i funkcje klas. Po prawej stronie znajduje się odpowiadający mu moduł testowy.	84

- Rycina 27.** Liczba linii kodu i dokumentacji programu ModeRNA oraz jego części testowej. Przez dokumentację rozumiane są opisy poszczególnych funkcji i klas (*docstrings*) oraz komentarze wewnątrz kodu (linie rozpoczynające się znakiem '#')..... 87
- Rycina 28.** Hierarchiczna struktura obiektów reprezentujących model w programie ModeRNA. Po lewej stronie przedstawione zostały zależności pomiędzy obiektami w pakiecie Bio.PDB, po prawej zależności między głównymi obiektami biblioteki ModeRNA. Schemat wykorzystuje elementy języka UML..... 88
- Rycina 29.** Obiekty reprezentujące strukturę w programie ModeRNA na różnym poziomie złożoności i charakterystyczne dla nich funkcje. Wyróżniona nazwa w każdej ramce oznacza rodzaj obiektu. Funkcje wypisane na szaro nie są zaimplementowane w danym obiekcie, ale dziedziczone. Strzałki oznaczają zależność dziedziczenia - grot skierowany jest w stronę obiektu, który jest dziedziczony. Schemat wykorzystuje elementy języka UML..... 89
- Rycina 30.** Przykładowa funkcja programu ModeRNA. Po lewej stronie przedstawiona jest ogólna budowa funkcji. Po prawej funkcja `score_clash` należąca do obiektu `LirHit` z modułu `SearchLIR`..... 90
- Rycina 31.** Ewaluacja modeli tRNA, relacja pomiędzy identycznością sekwencji celu i szablonu, a: A) RMSD wszystkich atomów w odniesieniu do struktur rozwiązanych doświadczalnie; B) RMSD wszystkich atomów wykluczając reszty antykodonu oraz regionu CCA na końcu 3' w odniesieniu do struktur rozwiązanych doświadczalnie C) GDT-TS; D) RMSD atomów P i C4' modeli w odniesieniu do struktur rozwiązanych doświadczalnie (na czarno), oraz struktur celu rozwiązanych doświadczalnie w odniesieniu do struktur szablonów (na czerwono). Na podstawie (Rother, i wsp., 2011). 93
- Rycina 32.** Przykłady modeli tRNA zbudowanych przy użyciu programu ModeRNA. Model pokazany jest na czerwono i nałożony na strukturę rozwiązaną doświadczalnie (w kolorze zielonym). A) model tRNA^{Phe} *E. coli* (kod PDB struktury natywnej: 2J00, łańcuch W) zbudowany na podstawie szablonu tRNA^{Phe} *E. coli* (kod PDB: 2HGP, łańcuch D). Pomimo 100% identyczności sekwencji celu i szablonu, RMSD modelu i struktury natywnej wynosi 3,61 Å. Reszty, które najsilniej przyczyniają się do wysokiego RMSD zostały zaznaczone na szaro i pokazane w reprezentacji pełnoatomowej; B) model tRNA^{Thr} *E. coli* (kod PDB struktury natywnej: 1QF6, łańcuch B) zbudowany na szablonie tRNA^{Cys} *E. coli* (kod PDB: 1B23, łańcuch R). Struktura natywna oddziałuje z syntetazą tRNA^{Thr}, natomiast szablon z czynnikiem elongacji translacji EF-Tu, co powoduje inną konformację ramienia akceptorowego i pętli antykodonowej – oba regiony zostały zaznaczone na szaro; C) model tRNA^{Glu} (kod PDB struktury natywnej: 2DXI, łańcuch C) zbudowany na podstawie szablonu 2DET. Sekwencje celu i szablonu mają wysokie podobieństwo (72%), jednak RMSD modelu w odniesieniu do struktury rozwiązywanej doświadczalnie wynosi 8,05 Å. Główną przyczyną wysokiego RMSD jest brak sześciu nukleotydów na końcu 3' szablonu, które w efekcie zostały wymodelowane w innej konformacji niż w strukturze natywnej; D) model tRNA^{fMet} *E. coli* (kod PDB struktury natywnej: 2HGI, łańcuch C) zbudowany na podstawie szablonu tRNA^{Phe} *E. coli* (kod PDB: 2B64, łańcuch V). Model w odniesieniu do struktury rozwiązywanej doświadczalnie ma niskie RMSD (1,38 Å), pomimo niezbyt wysokiego podobieństwa sekwencji celu i szablonu (47%). Na podstawie (Rother, i wsp., 2011). 94
- Rycina 33.** Ewaluacja modeli tRNA: A) wartości DI; B) wartości DP. Na podstawie (Rother, i wsp., 2011)..... 96

Rycina 34. Model intronu grupy I <i>Azoarcus</i> zbudowany przy użyciu programu ModeRNA (kolor czerwony) i RNABuilder (kolor niebieski), nałożone na strukturę rozwiązana doświadczalnie – kod PDB: 1U6B (kolor zielony). Na podstawie (Rother, i wsp., 2011).....	103
Rycina 35. Model 16S rRNA z podjednostki 30S rybosomu <i>T. thermophilus</i> (na czerwono) nałożony na strukturę rozwiązana doświadczalnie – kod PDB: 1J5E (na zielono). Na podstawie (Rother, i wsp., 2011).	110
Rycina 36. Modele tRNA ^{Thr} <i>E. coli</i> zbudowane na podstawie szablonu tRNA ^{Phe} <i>Saccharomyces cerevisiae</i> (kod PDB: 1EHZ; kolor czerwony) i szablonu tRNA ^{Asp} (kod PDB: 1C0A; kolor niebieski) nałożone na strukturę rozwiązana doświadczalnie – kod PDB: 1QF6. Na podstawie (Rother, i wsp., 2011).....	113
Rycina 37. Przykłady wartości rozważanych podczas implementacji programu ModeRNA jako kryteria wstępnego dopasowania fragmentu. Ostateczne kryteria używane w bieżącej wersji programu ModeRNA omówione zostały w rozdziale 5.2.2.4 i przedstawia je Rycina 20.	116
Rycina 38. Przykłady motywów dostępnych w bazie FRABASE: A) spinka do włosów; B) wyrzuszenie; C) asymetryczna pętla wewnętrzna; D) złącze; E) całujące się pętle. Rycina przygotowana na podstawie ilustracji dostępnych na stronie bazy FRABASE (Popenda, i wsp., 2008).....	118
Rycina 39. Ramię antykodonowe modelu tRNA ^{Thr} <i>E. coli</i> zbudowanego na podstawie szablonu tRNA ^{Asp} (kod PDB: 1C0A; kolor niebieski) nałożone na strukturę rozwiązana doświadczalnie – kod PDB: 1QF6 (kolor zielony). Na podstawie (Rother, i wsp., 2011). Rycina 36 przedstawia cały model tej cząsteczki.	120
Rycina 40. Możliwości otrzymania struktury trójwymiarowej RNA.	124

Spis tabel

Tabela 1. Przykłady cząsteczek RNA pełniących funkcje katalityczne i/lub regulatorowe.	13
Tabela 2. Metody doświadczalne pozwalające na uzyskiwanie informacji o strukturze RNA. Na podstawie (Felden, 2007).	20
Tabela 3. Charakterystyka programów do modelowania struktury przestrzennej RNA.	24
Tabela 4. Biblioteki języka Python wykorzystane podczas implementacji programu ModeRNA.	35
Tabela 5. Jednoliterowe skróty mogące występować w sekwencji w programie ModeRNA.	67
Tabela 6. Opcje dostępne w programie ModeRNA przez wiersz poleceń.	74
Tabela 7. Funkcje programu ModeRNA dostępne przez interfejs skryptowy. Na białym tle przedstawione są funkcje związane z modelowaniem. Niebieskie wyróżnienie oznacza funkcje pozwalające na analizę struktury. Zielone tło wskazuje funkcje związane z przygotowaniem danych wejściowych.	76
Tabela 8. Moduły programu ModeRNA (pliki <i>.py</i>) z liczbą importowanych bibliotek, liczbą klas i funkcji, liczbą wszystkich linii kodu (jako kod rozumiane są również linie importujące biblioteki i definiujące klasy i funkcje) i dokumentacji, a także całkowitą liczbą linii (zawiera również puste linie oddzielające poszczególne funkcje). Wyróżnione zostały najwyższe wartości dla głównego pakietu programu. Moduły testowe zostały przedstawione zbiorczo. Na dole tabeli znajduje się podsumowanie całego programu z uwzględnieniem testów.	85
Tabela 9. Średnie wyniki jakości 9675 modeli tRNA zbudowanych automatycznie przez program ModeRNA w odniesieniu do struktur rozwiązanych doświadczalnie.	92
Tabela 10. Wartości RMSD dla modeli intronu grupy I <i>Azoarcus</i> w odniesieniu do struktury rozwiązanej doświadczalnie (kod PDB: 1U6B). Modele zostały zbudowane przy użyciu programów ModeRNA i RNABuilder (Flores, i wsp., 2010). Na podstawie (Rother, i wsp., 2011).	104
Tabela 11. Porównanie programów do modelowania struktury przestrzennej RNA. Czarne kropki wskazują na obecność danej cechy, zaś puste okręgi jej brak. Gwiazdka oznacza, że kod źródłowy programu jest dostępny po wysłaniu zapytania do autorów. Na podstawie (Rother, i wsp., 2011).	106
Tabela 12. Ocena jakości modeli tRNA ^{Thr} <i>E. coli</i> zbudowanych na dwóch różnych szablonach. Pogrubiona czcionka oznacza lepszą ocenę.	112
Tabela 13. Pliki wykorzystane podczas pracy nad projektem ModeRNA. Niektóre pliki zostały użyte do kilku zastosowań.	125