Летний всемирный научный марафон
Территория инновационных идей, технологий и творческих решений

ФГБОУ ВО «Уфимский государственный нефтяной технический университет»

Ступень обучения: Бакалавриат
Направление: Технические науки
Тематика: Биоинформатика

Выпускная квалификационная работа

Определение генетических отклонений методами генной статистики

Работу выполнил:
Кудряшова Кира Константиновна
Студент 4 курса
ФГБОУ ВО «Уфимский государственный нефтяной технический университет»

Научный руководитель:
Дружинская Елена Владимировна
Старший преподаватель каф. ВТИК
ФГБОУ ВО «Уфимский государственный нефтяной технический университет»

Уфа, 2020

РЕФЕРАТ

Бакалаврская работа 80 л., 43 рис., 2 табл., 11 источников, 3 прил.

АЛГОРИТМЫ, ГЕНЕТИЧЕСКИЕ ОТКЛОНЕНИЯ, СТРОКИ, СИСТЕМА ПОИСКА, ВЫРАВНИВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ, ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

В работе рассматриваются и сравниваются алгоритмы генной статистики локального и глобального выравнивания с целью выявления эффективного по времени алгоритма поиска гомологичных участков биологических последовательностей.

Целью работы является определение незакономерных подпоследовательностей в длинных генетических цепочках.

Для выполнения поставленной цели были сформированы следующие задачи:

− изучить строение генома;

− рассмотреть алгоритмы селекции;

− провести сравнительный анализ изученных алгоритмов;

− реализовать алгоритм, эффективный по времени;

− провести тестирование и проверку корректности алгоритма;

− создать систему поиска, использующую разработанный алгоритм.

В работе приведен обзор предметной области, содержащий описание строения генома и генных мутаций, разбор и сравнительный анализ методов локального и глобального выравнивания последовательностей; проектирование и моделирование выбранного алгоритма: описание инструментов проектирования, общее описание алгоритма с блок-схемой, блок-схемы подпрограмм сплайсинга и транскрипции, а также модель алгоритма Смита-Ватермана, созданная средствами MS Excel; реализация: описание инструментальных средств и системы поиска.

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ДНК – дезоксирибонуклеиновая кислота

РНК – рибонуклеиновая кислота

ИТ – информационные технологии

BLOSUM – BLOck SUbstitution Matrix (матрица замен)

BLAST – Basic Local Alignment Search Tool (эвристический метод)

PAM – Percent Accepted Mutation (матрица замен)

ОС – операционная система

ПК – персональный компьютер

ЭВМ – электронно-вычислительная машина

ООП – объектно-ориентированное программирование

БД – база данных

ВВЕДЕНИЕ

В настоящее время развитие компьютеров и информационных технологий играет важную роль во многих современных науках. Математические алгоритмы всех областей научной деятельности адаптируются для решения специфических задач, производится их программная реализация.

Внедрение ИТ в биологические науки повлекло за собой возникновение новой междисциплинарной области науки – биоинформатики. Она сочетает такие направления, как статистика, математика и компьютерная наука.

Биоинформатика работает с моделями биологических систем и на сегодняшний день внесла значительный вклад в сравнительную геномику, моделирование молекул биологических веществ и предсказание их мутаций.

В основе биоинформатики лежит исследование биологических цепочек. Одним из важнейших направлений является изучение свойств макромолекул ДНК, РНК и белков, определяемых некоторым расположением нуклеотидов и аминокислот друг относительно друга. Для того, чтобы преобразовать информацию, закодированную в нуклеотидных и аминокислотных последовательностях в символьные модели, существует множество методов. Наиболее часто используемым является метод секвенирования.

Секвенирование – процесс перевода последовательности нуклеотидов ДНК в символьную последовательность.

В настоящее время секверирование генома конкретного человека занимает всего несколько дней. Это повлекло возникновение проблемы экспоненциального роста объема массива экспериментальных данных: последовательностей ДНК, РНК и белков. Данная проблема обосновывает необходимость создания нового ПО для хранения, управления и анализа биологических данных.

Анализ генетических цепочек является важной задачей в биоинформатике. Процесс сравнения строк с целью выявления гомологий называется выравниванием.

Для нахождения оптимального выравнивания между двумя последовательностями широко применяются алгоритмы Смита-Ватермана и

Нидлмана-Вунша. Выравнивание позволяет выявить схожесть двух последовательностей, что является важным при изучении эволюции различных видов.

На сегодняшний день большинство реализаций этих алгоритмов находят лишь значение веса выравнивания и не отображают его картину, важную при расчёте средней гомологии. Подобная реализация осуществляется с целью уменьшения объема вычислений.

Ввиду высокой вычислительной сложности алгоритмов, при поиске гомологичных последовательностей в базах данных нуклеотидных и аминокислотных цепочек часто используются быстрые эвристические методы FASTA и BLAST. Однако, они не гарантируют нахождения оптимального выравнивания и могут упустить отдельные участки последовательностей.

Квадратичная сложность алгоритмов, а также экспоненциальный рост объема баз данных первичных биологических последовательностей увеличивает актуальность проблемы анализа длинных генетических цепочек и оптимизации алгоритма.

Существующие на сегодняшний день оптимизированные реализации алгоритмов близки по быстродействию к эвристическим методам, так как используют различные способы распараллеливания на различных архитектурах (FPGA, Cell/BE и другие). Однако, при больших вычислительных возможностях данные реализации являются дорогостоящими.

Цель работы состоит в создании системы поиска незакономерных подпоследовательностей генетических цепочек. Система должна основываться на алгоритме, наиболее пригодном для выравнивания последовательности в произвольное положение другой.

Структура и объем работы. Работа состоит из введения, трёх разделов, заключения, списка литературы, включающий 11 наименованийи трёх приложений.

1 Обзор предметной области

1.1 Структура генома

Геном – совокупность наследственного материала, содержащегося в клетке и несущего биологическую информацию, необходимую для развития и функционирования всего организма. В структуре генома всех эукариотов, в том числе и человека, имеется две основные особенности: наличие большого количества не кодирующей ДНК и сжатое хранение информации в кодирующих областях генома. Функции и назначение «избыточной» ДНК на сегодняшний день остаются во многом неизвестными.

Вся наследственная информация каждого организма кодируется с помощью ДНК. Дезоксирибонуклеиновая кислота (ДНК) – одна из трёх макромолекул, обеспечивающая хранение, передачу и реализацию генетической программы развития и функционирования живых организмов. ДНК хранит биологическую информацию в виде генетического кода и содержит информацию о структуре различных видов РНК и белков. Структурно она представляет из себя две полинуклеотидные цепи, состоящие из углевода – дезоксирибоза, остатка фосфорной кислоты и одного из четырёх азотистых оснований: A (аденин), C (цитозин), G (гуанин) и T (тимин), соединяемых по принципу комплементарности (аденин-тимин и гуанин-цитозин). Генетическая информация, записанная в ДНК посредством рибонуклеиновой кислоты (РНК), реализуется в виде белков.

Общая длина генома человека составляет около 3 миллиардов пар оснований [1], из которых различия имеются лишь в 0,1 %. Регуляторные элементы, и в частности области начала репликации хромосом, и 36 % копирующих себя генов, не несущие функционального смысла, делают геном кусочно похожим. Такие повторы расположены в среднем через каждые 40 тысяч пар оснований и встречаются в геноме человека около ста тысяч раз. Длина единиц повтора варьируется от 1-2 до 2000 пар оснований, число копий – от десятков до сотен тысяч на геном. Повторяющиеся

участки могут быть сгруппированы в кластеры, либо распределены по молекуле ДНК, находиться в определенном районе хромосомы, либо в разных.

Ген – участок ДНК, который транскрибируется в РНК и, возможно, в дальнейшем в белок. Часть генов, не преобразующихся в белки, отвечают лишь за работу других генов. Кодирование белков осуществляется всего 2 % генов. Центральная догма молекулярной биологии постулирует, что передача генетической информации осуществляется от нуклеиновых кислот к белкам. Стоит отметить, что данный процесс является необратимым. Процесс реализации белков из ДНК осуществляется посредством РНК. Основным отличием РНК от ДНК является замена нуклеотида Т (тимина) на U (урацил). Изучение механизмов биосинтеза белка выявило целый спектр различных функций РНК, помимо переноса информации от генов к белкам.

У млекопитающих, в отличие от бактерий, гены имеют очень сложную структуру, поэтому, чтобы производить преобразования, недостаточно выделить подстроку и работать с ней. Говоря о биосинтезе белка, ген является строкой, имеющей подстроки – экзоны и интроны (рисунок 1.1).

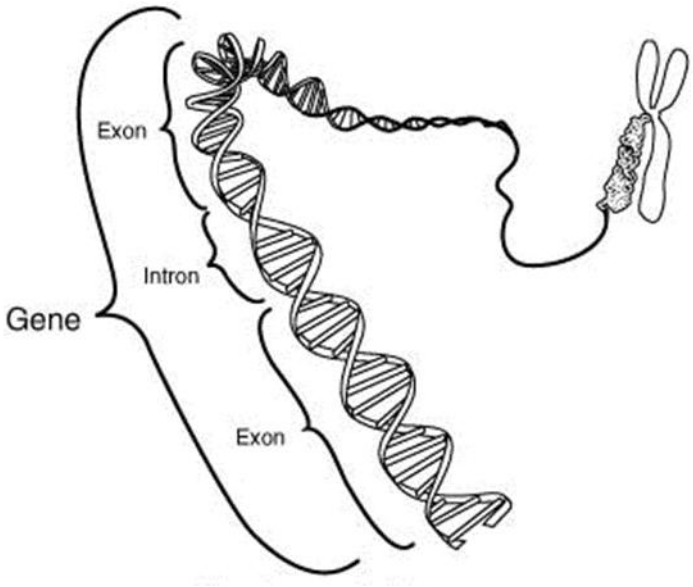


Рисунок 1.1 – Экзоны и интроны

Интроны – участки гена, не кодирующие никакую генетическую информацию. Для построения белков из общей последовательности удаляются интроны, затем выделяются и соединяются экзоны. Процесс удаления интронов из гена, в процессе его преобразования из ДНК в РНК и далее в белок называется сплайсинг [2]. Чаще всего интроны начинаются с «GT» и заканчиваются «AG». Отметим, что сплайсинг осуществляется после транскрипции. Транскрипция – процесс синтеза РНК из ДНК с использованием матрицы, то есть перенос генетической информации с ДНК на РНК. Процесс синтеза белка из РНК называется трансляцией. Полипептидная цепь белка формирует уникальную пространственную структуру, которая кодируется последовательностью аминокислот [2, 9]. У большинства организмов протеины строятся на основе 20 аминокислот.

Организация белка разделяется на три основных уровня:

− первичная структура;

− вторичная структура;

− третичная (трёхмерная) структура.

Первичная структура представляет из себя последовательность букв. Вторая структура состоит из альфа-спиралей и бетта-листов (рисунок 1.2).

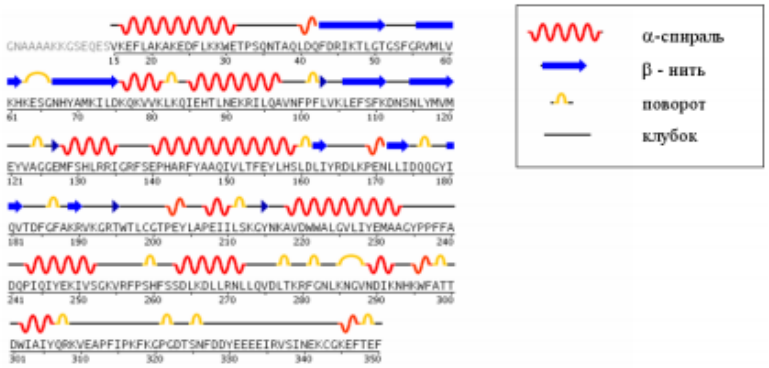


Рисунок 1.2 – Первичная и вторичная структуры организации белков

Трёхмерная структура включает в себя трёхмерную модель белка (рисунок 1.3). Четвертичная структура представляет собой комплекс белков.

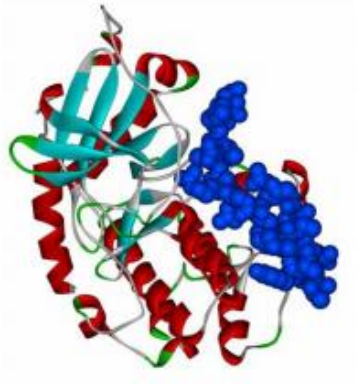


Рисунок 1.3 – Третичная и четвертичная структура организации белка

Структура белка содержит в себе активные сайты – участки взаимодействий с другими молекулами (рисунок 1.4). Именно активные сайты задают функции белка.

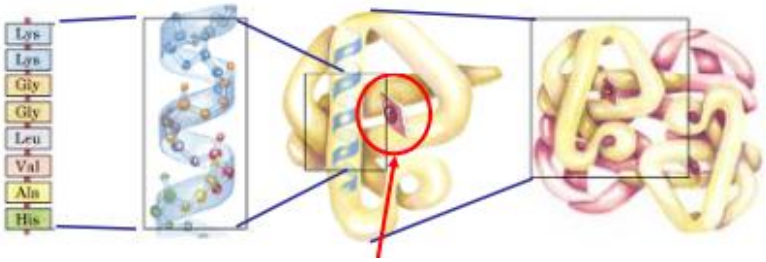


Рисунок 1.4 – Активные сайты в структуре белка

С конца 20 века задачи в таких разделах молекулярной генетики [2, 11], как геномика, протеомика и транскриптомика, изучающих геном и гены, белки и РНК соответственно, стали решаться с помощью информационных технологий.

Биоинформатика – направление науки, изучающая биологические процессы с точки зрения информатики. Для проведения анализа первичных биологических последовательностей с помощью информационных технологий, необходимо преобразовать цепочку ДНК в символьную последовательность. Для этого выделяется ДНК из клеток и с помощью специальных химических реакций и приборов расшифровывается анализируемый ген. Процесс определения первичной

нуклеотидной последовательности ДНК называется секвенированием. Появление быстрых методов секвенирования привело к экспоненциальному росту объема баз данных биологических последовательностей (рисунок 1.5).

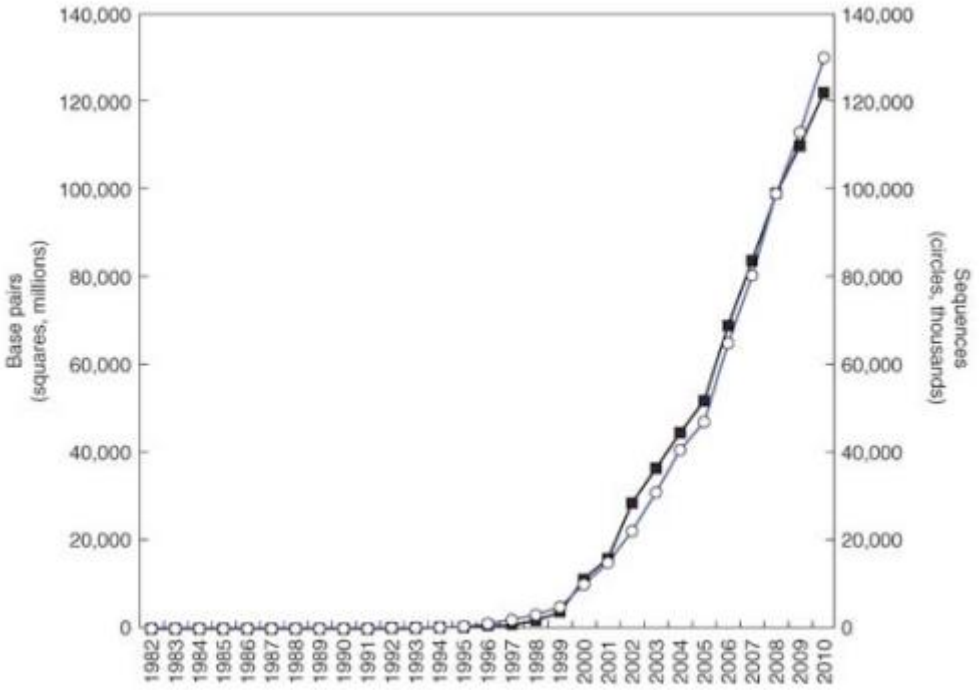


Рисунок 1.5 – Экспоненциальный рост объема баз данных первичных биологических цепочек

В настоящее время открыто около 100 тысяч различных структур белка. Перед биоинформатиками встала задача сравнительного анализа белков с целью поиска гомологичности и определения мутационных процессов, возникших в ходе эволюции организма [1].

1.2 Генные мутации

Генная мутация – устойчивое изменение последовательности ДНК, образующей ген. Они могут затрагивать как один структурный элемент ДНК, так и крупный сегмент хромосомы, содержащей ряд генов [8].

Выделяется два вида генных мутаций:

− генеративные мутации наследуются от родителя и сохраняются в каждой клетке организма на всем протяжении жизни человека. Такие мутации возникают в яйцеклетке или сперматозоидах родителя и передаются как наследственные мутации;

− соматические мутации возникают на каком-то этапе жизни человека и затрагивают лишь некоторые клетки, а не каждую клетку организма. Такие изменения могут быть вызваны воздействием факторов окружающей среды, например ультрафиолетовым солнечным излучением, или ошибкой копирования ДНК при делении клетки. Подобные приобретенные мутации не передаются потомству.

Генетиками выделяется три элементарных мутационных процесса (рисунок 1.6):

1) замена (substitution) нуклеотидов или аминокислот;
2) разрыв (gap);
3) вставка и удаление (delection).

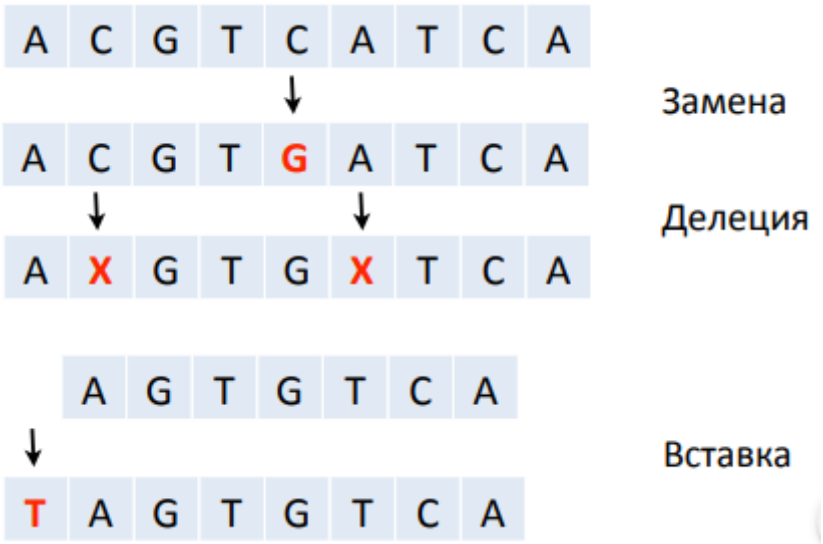


Рисунок 1.6 – Элементарные мутационные процессы

Выявление у человека генетического заболевания, связанного с изменением структуры ДНК в гене, требует проведения ряда тестов и исследований в молекулярно-генетической лаборатории с целью анализа ДНК определённого гена.

Один ген может кодироваться 10000 и более нуклеотидами, поэтому данная процедура может продлиться от двух недель до нескольких месяцев. Однако наличие в базе данных исследований генетического материала человека или его родственников определённой давности может значительно сократить временные затраты, так как анализ сведется к сравнению двух гомологичных последовательностей.

1.3 Алгоритмы выравнивания

1.3.1 Выравнивание последовательностей.

В процессе эволюции с биологическими макромолекулами живых организмов происходит множество мутационных изменений, что приводит к перестройке отдельных отрезков последовательности или к образованию отдельных точечных мутаций. Соответственно, если биологические макромолекулы имеют общие последовательности мономеров, то, как правило, обнаруживаются гомологии в структурах и в биологических функциях. В данном случае принято считать, что если два белка или две нуклеотидные последовательности имеют большое сходство как структурно, так и функционально, то они имеют общего предшественника.

С помощью сравнения предположительно гомологичных последовательностей можно определить степень их изменения в ходе эволюции организма. Однако, если два белка имеют подобную трёхмерную структуру, но их последовательности отличны, в этом случае белки не являются гомологами [10].

Биоинформатика в большей степени принимает во внимание именно гомологичные белки с общим происхождением, 3D-структурой и более или менее сходной аминокислотной последовательностью. Процесс определения гомологичности отдельных участков биологических последовательностей называется выравниванием. Данный процесс может производиться как для двух последовательностей (парное), так и для нескольких (множественное). Стоит отметить, что множественное выравнивание осуществляется на основе результатов

парного. Процесс подразумевает сохранение исходного порядка остатков в последовательностях.

Выравнивание последовательностей является основным инструментом биоинформатики, так как именно с помощью него устанавливаются структурные, функциональные и эволюционные сходства и различия последовательностей [5].

1.3.2 Парное выравнивание.

Парное выравнивание – это сравнение двух биологических цепочек: ДНК, РНК или белков, с целью оценки гомологии данных последовательностей [2, 4]. С его помощью определяют структурную и функциональную взаимосвязь двух белков или генов; выявляют общие стабильные (домены) и изменяющиеся в процессе эволюции (мотивы) участки белков; используют для анализа и аннотации генома. Таким образом, парное выравнивание является фундаментальной операцией биоинформатики.

Выравнивание аминокислотных последовательностей является более информативным, чем последовательности ДНК. Во-первых, последовательности аминокислот строятся на основе двадцатибуквенного алфавита, в отличие от нуклеотидных цепочек, в построении которых задействовано всего четыре символа. Во-вторых, изменение в третьей позиции нуклеотидного кодона часто не изменяет аминокислоту. Поэтому необходимо в первую очередь произвести перевод ДНК в белок, и затем выполнить выравнивание.

Как описывалось ранее, для того, чтобы привести последовательность ДНК к аминокислотному виду, сначала производится удаление белого шума – не кодирующих участков (интронов) и транскрипция её в РНК. Над полученной последовательностью производится трансляция – процесс получения белка из РНК. Для этого последовательность разбивается на кодоны – 3 нуклеотида, каждый из которых соответствует какой-либо аминокислоте (рисунок 1.7). При этом определено три стоп-кодона: UAA, UAG, UGA.

Из таблицы трансляции видно, что вариаций кодонов $4^3=64$, а кодирующих аминокислот 20, что свидетельствует об избыточности кода. Так как изменение третьей буквы кодона не приводит к изменению аминокислоты, считается, что

мутации, не приводящие к изменению белка, происходят каждую третью букву кодона.

ВТОРОЙ НУКЛЕОТИД

ПЕРВЫЙ НУКЛЕОТИД	U		C		A		G		ТРЕТИЙ НУКЛЕОТИД
U	UUU	Phe F	UCU	Ser S	UAU	Tyr	UGU	Cys C	U
	UUC	Phe	UCC		UAC	Tyr O	UGC	Cys	C
	UUA	Leu	UCA		UAA	Term	UGA	Term	A
	UUG	Leu	UCG		UAG	Term	UGG	Trp W	G
C	CUU	Leu	CCU	Pro P	CAU	His H	CGU	Arg	U
	CUC	Leu L	CCC		CAC	His	CGC		C
	CUA	Leu	CCA		CAA	Gln Q	CGA		A
	CUG	Leu	CCG		CAG	Gln	CGG		G
A	AUU	Ile	ACU	Thr T	AAU	Asn N	AGU	Ser	U
	AUC	Ile I	ACC		AAC	Asn	AGC	Ser	C
	AUA	Ile	ACA		AAA	Lys K	AGA	Arg R	A
	AUG	Met M	ACG		AAG	Lys	AGG	Arg	G
G	GUU	Val	GCU	Ala A	GAU	Asp D	GGU	Gly G	U
	GUC	Val V	GCC		GAC	Asp	GGC		C
	GUA	Val	GCA		GAA	Glu E	GGA		A
	GUG	Val	GCG		GAG	Glu	GGG		G

Рисунок 1.7 – Таблица трансляции РНК в белок

После подготовки сравниваемых последовательностей, приведения их к аминокислотному виду, производится парное выравнивание.

В общем виде процесс можно описать следующим образом:

− выбрать способ генерации оценки сходства;

− определить штрафы для вставки, удаления, пробелов и замен при выравнивании;

− выбрать алгоритм выравнивания;

− определиться с требуемыми выходными данными (выводить только степень сходства или необходимо также отображение картины выравнивания.

Для генерации оценки сходства последовательностей вводится понятие расстояния.

Мерой расстояния является функция, определяющая разность между двумя последовательностями. Чем больше получается значение, тем меньше сходства у

сравниваемых последовательностей. Существует два способа определения расстояния [6]:

1) расстояние Хэмминга выявляет количество негомологичных позиций;

2) расстояние Левенштейна (редакционное расстояние) определяет минимальное количество точечных преобразований (замен, вставок и удалений), необходимых для получение из одной последовательности другую.

Стоит отметить, что не все преобразования имеют одинаковый вес. Поэтому при расчёте расстояния каждому виду присваивают различный вес. Для подсчёта сходства нуклеиновых кислот в общем случае добавляют +1 балл при совпадении символа и -1 балл за несовпадение.

Рассматривая аминокислотные последовательности, необходимо учесть тот факт, что замена одной аминокислоты родственной может не изменить функциональность белка, а замена разнородной – прекратить функционирование. Поэтому для поиска оптимального выравнивания в белках сформированы матрицы замещения [3, 5]: PAM и BLOSUM (рисунки 1.8, 1.9).

Значения в матрицах умножены на 10 во избежание дробей. Матрицы замещения симметричны относительно диагонали.

При выборе алгоритма выравнивания необходимо определиться каким образом будет производиться выравнивание: по всей длине последовательности или сосредоточенно на отдельных отрезках.

В настоящее время выделяют два основных метода выравнивания последовательностей: глобальное и локальное (рисунок 1.10).

Локальное выравнивание [2, 11] фокусируется на отдельных областях сходства в некоторых частях последовательностей. Данный метод является наиболее эффективным для выравнивания преимущественно отличающихся последовательностей разной длины, которые предположительно могут содержать одинаковые участки.

Глобальное выравнивание [2, 11] ищет сходства по всей длине последовательности. Применяется метод к схожим строкам приблизительно одной длины.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Ala (A)	**6**	−7	−4	−3	−6	−4	−2	−2	−7	−5	−6	−7	−5	−8	−2	0	−1	−13	−8	−2
Arg (R)	−7	**8**	−6	−10	−8	−2	−9	−9	−2	−5	−8	0	−4	−9	−4	−3	−6	−2	−10	−8
Asn (N)	−4	−6	**8**	2	−11	−3	−2	−3	0	−5	−7	−1	−9	−9	−6	0	−2	−8	−4	−8
Asp (D)	−3	−10	2	**8**	−14	−2	2	−3	−4	−7	−12	−4	−11	−15	−8	−4	−5	−15	−11	−8
Cys (C)	−6	−8	−11	−14	**10**	−14	−14	−9	−7	−6	−15	−14	−13	−13	−8	−3	−8	−15	−4	−6
Gln (Q)	−4	−2	−3	−2	−14	**8**	1	−7	1	−8	−5	−3	−4	−13	−3	−5	−5	−13	−12	−7
Glu (E)	−2	−9	−2	2	−14	1	**8**	−4	−5	−5	−9	−4	−7	−14	−5	−4	−6	−17	−8	−6
Gly (G)	−2	−9	−3	−3	−9	−7	−4	**6**	−9	−11	−10	−7	−8	−9	−6	−2	−6	−15	−14	−5
His (H)	−7	−2	0	−4	−7	1	−5	−9	**9**	−9	−6	−6	−10	−6	−4	−6	−7	−7	−3	−6
Ile (I)	−5	−5	−5	−7	−6	−8	−5	−11	−9	**8**	−1	−6	−1	−2	−8	−7	−2	−14	−6	2
Leu (L)	−6	−8	−7	−12	−15	−5	−9	−10	−6	−1	**7**	−8	1	−3	−7	−8	−7	−6	−7	−2
Lys (K)	−7	0	−1	−4	−14	−3	−4	−7	−6	−6	−8	**7**	−2	−14	−6	−4	−3	−12	−9	−9
Met (M)	−5	−4	−9	−11	−13	−4	−7	−8	−10	−1	1	−2	**11**	−4	−8	−5	−4	−13	−11	−1
Phe (F)	−8	−9	−9	−15	−13	−13	−14	−9	−6	−2	−3	−14	−4	**9**	−10	−6	−9	−4	2	−8
Pro (P)	−2	−4	−6	−8	−8	−3	−5	−6	−4	−8	−7	−6	−8	−10	**8**	−2	−4	−14	−13	−6
Ser (S)	0	−3	0	−4	−3	−5	−4	−2	−6	−7	−8	−4	−5	−6	−2	**6**	0	−5	−7	−6
Thr (T)	−1	−6	−2	−5	−8	−5	−6	−6	−7	−2	−7	−3	−4	−9	−4	0	**7**	−13	−6	−3
Trp (W)	−13	−2	−8	−15	−15	−13	−17	−15	−7	−14	−6	−12	−13	−4	−14	−5	−13	**13**	−5	−15
Tyr (Y)	−8	−10	−4	−11	−4	−12	−8	−14	−3	−6	−7	−9	−11	2	−13	−7	−6	−5	**10**	−7
Val (V)	−2	−8	−8	−8	−6	−7	−6	−5	−6	2	−2	−9	−1	−8	−6	−6	−3	−15	−7	**7**

Рисунок 1.8 – Матрица замещения PAM30

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	−1	5																		
Asn	−2	0	6																	
Asp	−2	−2	1	6																
Cys	0	−3	−3	−3	9															
Gln	−1	1	0	0	−3	5														
Glu	−1	0	0	2	−4	2	5													
Gly	0	−2	0	−1	−3	−2	−2	6												
His	−2	0	1	−1	−3	0	0	−2	8											
Ile	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
Leu	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
Lys	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
Met	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
Phe	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
Pro	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
Ser	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
Thr	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
Trp	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Tyr	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

Рисунок 1.9 – Матрица замещения BLOSUM62

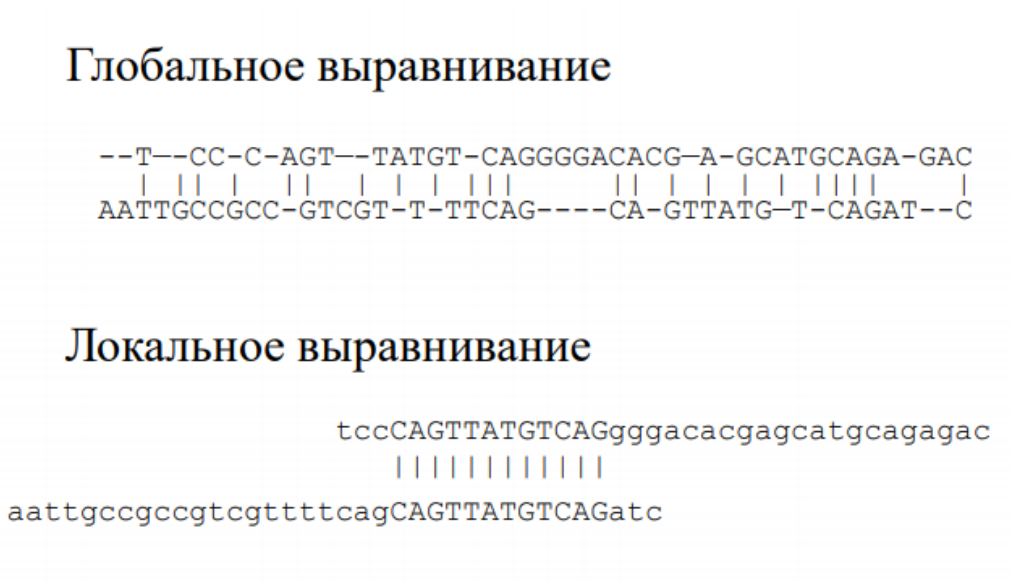


Рисунок 1.10 – Пример глобального и локального выравнивания последовательностей

Рассмотреть все возможные выравнивания, с целью поиска оптимального из них, физически невозможно, даже для коротких строк. В связи с этим, одним из основных математических методов, применяемых для нахождения парного выравнивания, является динамическое программирование, то есть решение сложных задач путем разбиения их на более простые подзадачи и дальнейшему объединению решений в общее решение поставленной задачи.

Динамическим алгоритмом нахождения глобального выравнивания является алгоритм Нидлмана-Вунша, а для локального – Смита-Ватермана [2, 4].

1.3.3 Алгоритм локального выравнивания Смита – Ватермана.

Алгоритм Смита-Ватермана решает задачу поиска гомологичных участков в двух последовательностях, применяя матрицу сходства $M_{i,j}$, где i и j изменяются от нуля до длины сравниваемых строк. Это позволяет учитывать различные замены, вставки и удаления отдельно взятых аминокислот.

Входными параметрами метода являются две последовательности S_1 и S_2 длины n и m соответственно и матрица замещения, определяющая штрафы разного рода несоответствия для каждой комбинации элементов выравнивания.

На момент начала построения значения элементов верхней строки и левого столбца матрицы переходов M считаются равными нулю. В процессе построения значение каждого из остальных элементов $M_{i,j}$ i-й строки j-го столбца матрицы переходов вычислялось по формуле (1.1), где g, w принимают различные значения в зависимости от того, совпадают ли соответствующие элементу $M_{i,j}$ аминокислотные остатки последовательностей.

$$M_{i,j} = \max\{M_{i-1,j} + g; M_{i,j-1} + g; M_{i-1,j-1} + w; 0\}, i, j > 1, \quad (1.1)$$

Возвращается наибольший из аргументов.

$$M_{i,j} = M_{i-1,j} + g, \quad (1.2)$$

$$M_{i,j} = M_{i,j-1} + g, \quad (1.3)$$

$$M_{i,j} = M_{i-1,j-1} + w. \quad (1.4)$$

Для каждого $M_{i,j} > 0$ направление перехода определяется следующим образом: если (1.2), то переход в элемент $M_{i,j}$ осуществляется из элемента $M_{i-1,j}$; если (1.3) – из элемента $M_{i,j-1} + g$; если (1.4) и $M_{i-1,j-1} > 0$ – из элемента $M_{i-1,j-1}$ (рисунок 1.11).

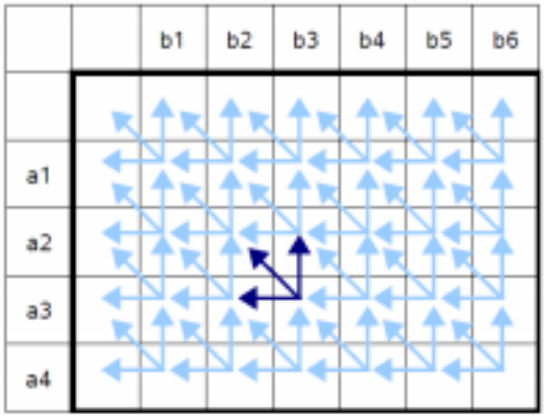


Рисунок 1.11 – Матрица выравнивания с указанием ячеек, от которых зависит вычисление текущей ячейки

Максимальный элемент матрицы будет равен максимальной степени сходства.

Для получения значения веса для каждого элемента матрицы динамического программирования добавлена возможность нулевого веса, в случае, если все остальные варианты выбора дают отрицательные значения.

Для получения картины выравнивания необходимо осуществить обратный обход матрицы, начиная с элемента с наибольшим количеством очков в матрице подсчёта и заканчивая в ячейке матрицы, имеющей оценку 0.

Алгоритм Смита- Ватермана является наиболее точным, но работает медленней других алгоритмов выравнивания.

Сложность алгоритма Смита-Ватермана: N^2.

1.3.4 Алгоритм глобального выравнивания Нидлмана-Вунша.

Алгоритм глобального выравнивания Нидлемана-Вунша заключается в заполнении матрицы M. На начальном этапе верхний левый элемент матрицы считается равным нулю, для вычисления значений прочих элементов верхней строки $M_{1,j}$ и левого столбца $M_{i,1}$ используются, соответственно, формулы (1.5) и (1.6), где g — число, прибавляемое к весу выравнивания за каждый из разрывов.

$$M_{i,j} = M_{i,j-1} + g, \tag{1.5}$$

$$M_{i,j} = M_{i-1,j} + g, \tag{1.6}$$

$$M_{i,j} = M_{i-1,j-1} + w. \tag{1.7}$$

Значение каждого из остальных элементов $M_{i,j}$ i-й строки j-го столбца матрицы переходов вычисляются по формуле (1.8), где g, w – штрафы за разрывы в соответствии с матрицей замен.

$$M_{i,j} = \max\{M_{i-1,j} + g; M_{i,j-1} + g; M_{i-1,j-1} + w\}, i, j > 1, \tag{1.8}$$

Если (1.6), то переход в элемент $M_{i,j}$ осуществляется из элемента $M_{i-1,j}$; если (1.5) – из элемента $M_{i,j-1} + g$; если (1.7) – из элемента $M_{i-1,j-1}$.

Результатом выравнивания будет являться число, полученное в правом нижнем углу матрицы. Для построения картины выравнивания необходимо совершить обратный обход матрицы, начиная с ячейки правого нижнего угла, по ячейкам, имеющих максимальный вес (рисунок 1.12). Данный алгоритм имеет квадратичные затраты памяти и времени.

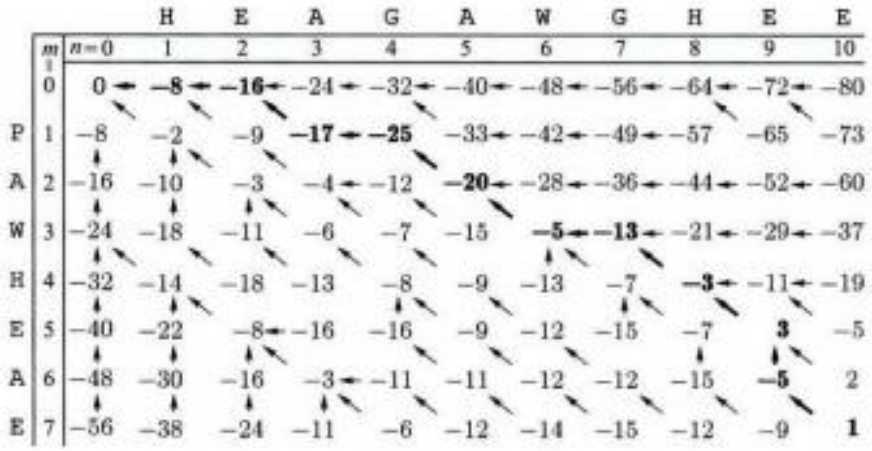


Рисунок 1.12 – Глобальное выравнивание методом Нидлмана-Вунша

1.3.5 Алгоритм Миллера-Маерса.

Алгоритм Миллера-Маерса производит поиск оптимального выравнивания путем разбиения одной из последовательностей на две равные части. Далее для каждой точки x – линии раздела, находятся веса оптимальных выравниваний из начала в x и из конца в x.

Вес оптимального выравнивания равен его максимальному значению, проходящего через точку x.

Таким образом, за время $T=(C*n^2)$ найдена точка, через которую проходит оптимальное выравнивание. Найденная точка x разбивает матрицу выравнивания на четыре квадрата, из которых два крайних заведомо не содержат оптимального выравнивания (рисунок 1.13).

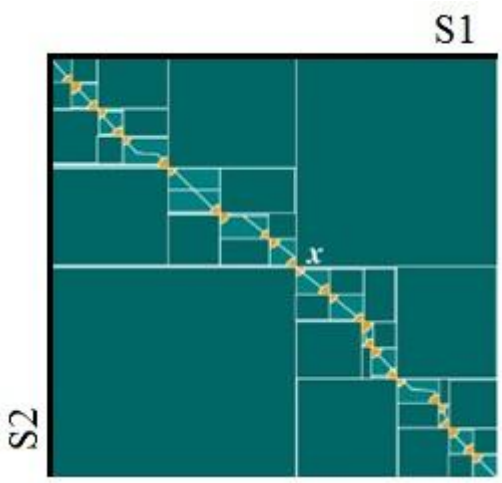


Рисунок 1.13 – Графическая интерпретация алгоритма Миллера - Маерса

Таким образом, применением того же метода для двух центральных квадратов и запоминанием x’ и x” найти оптимальное выравнивание. При этом, каждое последующее действие будет занимать вдвое меньше времени ($T=(C*n^2)/2$).

Важно отметить, что при проходе матрицы обратные переходы не запоминаются, что исключает возможность построения картины выравнивания.

Общее затрачиваемое время на поиск оптимального выравнивания $T=(C*n^2)$, затраты памяти – $O(n+m)$.

1.4 Сравнительный анализ алгоритмов выравнивания

Для сравнения алгоритмов парного выравнивания Смита-Ватермана и Нидлмана-Вунша были выбраны следующие критерии:

− тип выравнивания;

− временная характеристика;

− ресурсная характеристика (память);

− задача алгоритма;

− значение счёта в процессе заполнения матрицы переходов;

− построение картины выравнивания (обратный обход);

− пригодность для работы с большими данными;

− применимость;

− пригодность для работы с сильно различающимися последовательностями.

В отличие от алгоритма Нидлмана-Вунша, который осуществляет выравнивание последовательностей по всей длине, алгоритм Смита-Ватермана сравнивает отрезки всех возможных длин и оптимизирует меру сходства по всем отрезкам и всем выравниваниям этих отрезков. Очевидно, что поэтому длина локального выравнивания меньше.

Главным отличием двух типов выравнивания можно назвать отбрасывание концевых участков глобального выравнивания в локальном. Затраты времени и памяти у обоих алгоритмов квадратичны из-за большого объема входных данных.

Основные сходства и различия между алгоритмами Нидлмана – Вунша и Смита-Ватермана представлены в таблице 1.1.

Таблица 1.1 – Сравнительный анализ алгоритмов Нидлмана-Вунша и Смита-Ватермана

Критерий оценивания	Алгоритм	
	Нидлмана - Вунша	Смита - Ватермана
Тип выравнивания	Глобальное	Локальное
Расходуемая память	O(n*m)	O(n*m)
Время работы	O(n*m)	O(n*m)
Задача	Поиск наиболее весомого пути между вершинами (0,0) и (n,m) графа	Поиск наиболее весомого пути среди всех путей между вершинами
Счёт	Может быть отрицательным	Отрицательная оценка устанавливается в 0
Построение картины выравнивания	Начинается с ячейки в правом нижнем углу, завершается в левой верхней ячейке	Начинается с небольшим количеством очков, завершается при достижении 0
Пригодность для работы с большими данными	Нет	Да

Критерий оценивания	Алгоритм	
	Нидлмана - Вунша	Смита - Ватермана
Применение	Выравнивание полных последовательностей	Выравнивание сегмента последовательности в произвольное положение другой
Пригодность для работы с сильно различающимися последовательностями	Да	Нет

Завершение выравнивания в алгоритме Смита-Ватермана может произойти в любой ячейке матрицы, имеющей нулевое значение, в алгоритме Нидлмана-Вунша – для завершения требуется пройти максимально весомый путь от нижней правой до верхней левой ячейки.

Сравнительный анализ выявил, что наиболее эффективным способом для сравнения сильно различающихся строк разной длины и поиска генетических отклонений в некотором отрезке генома является алгоритм для локального выравнивания Смита-Ватермана. Во-первых, он пригоден для работы с большими данными. Во-вторых, с помощью данного алгоритма можно произвести выравнивание одной последовательности не по всей длине другой, а в произвольном положении.

1.5 Обзор программных реализаций алгоритма Смита-Ватермана

Алгоритм Смита-Ватермана для поиска оптимального выравнивания требует значительных вычислительных ресурсов и памяти. На сегодняшний день существуют следующие реализации данного алгоритма [4, 10]:

− эвристические, значительно сокращающие время поиска посредством уменьшения точности (FASTA, BLAST);

− основанные на многоядерных технологиях ускорения параллельных вычислений (CUDASW++);

− другие реализации (SSEARCH, PSI-Search).

В основном, данные реализации алгоритма Смита-Ватермана базируются на недорогом оборудовании.

Описание некоторых реализаций приведены в таблице 1.2.

Таблица 1.2 – Обзор программных реализаций алгоритма Смита-Ватермана

Пакет ПО	Описание пакета ПО
FASTA	Пакет FASTA получает на вход нуклеотидную или аминокислотную последовательность и совершает поиск соответствующей базы данных последовательностей с помощью локального выравнивания с целью поиска гомологий. Программа следует эвристическим методам, что обеспечивает высокую скорость выполнения. Алгоритм основан на поиске наиболее значимой диагонали матрицы выравнивания. Процесс поиска осуществляется в четыре этапа. На первом этапе производится хеширование с целью определения точных совпадений последовательностей длиной более *к* элементов. При этом запоминается номер диагонали, которой принадлежит совпадение. Штрафы за пропуске выставляются в соответствии с матрицей замещений BLOSUM50. В результате хеширования отбирается 10 диагоналей с лучшим весом. На втором этапе производится построение локального выравнивания для каждого фрагмента. Лучшие результаты именуются как initl. На третьем этапе для нескольких, выбранных на предыдущем этапе, формируется оптимальное выравнивание. Возвращаемый пользователю результат именуется как initn. На заключительном этапе строится выравнивание по алгоритму Смита-Ватермана, в ограниченном окне из восьми диагоналей, которые отсчитываются от initl. Выравнивание именуется как Opt. Этапы повторяются для каждой последовательности базы данных, после чего формируется статистика значимых выравниваний.
BLAST	Программа BLAST предназначена для поиска в базах данных Swiss-prot и PDB нуклеотидных или белковых последовательностей, имеющих гомологии с входной последовательностью. BLAST следует эвристическим методам и является менее точным, чем FASTA. Программа применима для построения парного и множественного выравнивания. Для запуска работы программы необходимо ввести изучаемую последовательность, наименование базы данных, размер участка. Данная информация отправляется на сервер NBCI. Программа

Пакет ПО	Описание пакета ПО
	создает таблицу всех «слов» (для белков это триплеты – участки последовательности, состоящие из трёх аминокислот каждый, для нуклеиновых кислот – из 11 нуклеотидов) и гомологичных слов. Далее в базе данных выполняется поиск. При обнаружении соответствий размеры «слов» увеличиваются (триплеты до 4 и более, нуклеотиды – 12 и более), при этом сначала без учета пробелов, затем с их использованием.
BLAST	После того, как размеры «слов» увеличены до максимального размера, программа определяет выравнивание с наибольшим количеством совпадений для каждой пары («изучаемая последовательность – последовательности из базы данных») записывается в структуру SeqAlign (представление таблицы приводится в традиционном виде, в виде таблицы и графически). На данном этапе программа BLAST для каждой пары определяется значимость сходства. Для определения показателя сходства программа использует матрицу замещения BLOSUM62. После определения совпадений BLAST использует модифицированный алгоритм Смита-Ватермана. Для каждой пары производится расширение в обе стороны и определяется влияние на показатели сходства (если они уменьшаются, то расширение продолжать не стоит). В результате работы алгоритма выбирается набор локального выравнивания с учетом штрафов за пропуски. Для выявления оптимального выравнивания производится статистическая оценка значимости. Наиболее значимые выравнивания и будут представлены пользователю.
CUDASW++	CUDASW предназначена для поиска наиболее гомологичной последовательности к входной. Программа находит оптимальное выравнивание для каждой пары «входная последовательность – последовательность из базы данных». В программе реализовано два подхода к решению этой задачи. Во-первых, параллелизм на уровне задач, то есть для решения задачи выполняется один поток. Данный подход затрачивает значительный объем памяти, но обладает лучшей производительностью. Во-вторых, параллелизм внутри задач, то есть для решения задачи выполняется блок потоков, параллельно выполняющих расчёты. Затраты памяти этого подхода меньше, что позволяет работать с длинными последовательностями. Для выполнения алгоритма Смита-Ватермана требуется память для хранения промежуточных результатов. Для заполнения матрицы требуются значения матрицы замещения. Для параллельного вычисления матрицы выравнивания разбиваются на равные блоки 8x8.

Пакет ПО	Описание пакета ПО
SSEARCH	SSEARCH осуществляет поиск и выравнивание последовательностей на основе алгоритма Смита-Ватермана. Данная реализация работает медленнее эвристического метода FASTA, но является более чувствительной. Результаты выводятся на основе статистического анализа отобранных выравниваний. Применим только к белковым последовательностям.
PSI-Search	Программа PSI-Search комбинирует алгоритм поиска гомологий Смита-Ватермана и PSI-BLAST. В основе заложен поиск отдаленно связанных последовательностей белка, а также предотвращение ошибок чрезмерного расширения гомологичных участков. Программа неприменима к нуклеотидных последовательностей.

Таким образом, геном человека представляет собой последовательность кодирующих и некодирующих участков ДНК длиной порядка 3 миллиардов пар оснований. В процессе эволюции биологические макромолекулы перестраиваются, из-за происходящих над ними мутационных изменений. Для выявления незакономерных участков в биоинформатике применяются алгоритм глобального выравнивания Нидлмана-Вунша и локального выравнивания Смита-Ватермана. Сравнительный анализ данных алгоритмов показал, что наиболее эффективным алгоритмом для поиска оптимального выравнивания преимущественно различающихся последовательностей разной длины является алгоритм Смита-Ватермана.

Программная реализация данного алгоритма стала составляющей разработанной системы поиска оптимального выравнивания.

2 Проектирование и моделирование

2.1 Методы проектирования

Проектирование можно определить, как процесс составления описания, необходимого для дальнейшей реализации алгоритма. Оно опирается на первичные теоритические знания о работе алгоритма и формируется путем его детализации, дополнения, оптимизации и расчётов.

Проектирование является важным шагом при решении сложных задач. Существуют различные методы проектирования, классифицируемые по разным признакам. Наиболее распространенными подходами в проектировании алгоритмов являются объектно-ориентированный и структурный. К основным принципам структурного подхода относятся нисходящее, восходящее и модульное программирование.

Нисходящее проектирование («сверху вниз») – это метод, основанный на последовательном разбиении сложной задачи на ряд отдельных вспомогательных подзадач, каждая из которых решается более элементарным способом.

Восходящее проектирование («снизу вверх») – метод, с помощью которого осуществляется построение программы из имеющихся элементов, начиная с примитивов языка программирования.

Модульное программирование – это метод, предполагающий разбиение программы на модули, каждый из которых автономно выполняет одну или несколько функций алгоритма.

Для каждого подхода проектирования существует свой способ представления алгоритмов. Для ООП применяются UML-диаграммы (диаграммы классов), для структурного программирования – словесно-описательный или графический способ. Представление алгоритма словесно-описательным способом осуществляется на естественном языке, однако описанные таким образом сложные алгоритмы становятся ненаглядными и громоздкими, поэтому данный способ используется лишь на начальных стадиях проектирования. К сложным алгоритмам применим способ

графического представления, то есть в виде последовательности связанных между собой блоков и стрелок, соединяющих их. В блок-схемах каждому типу соответствует определённая геометрическая фигура, что позволяет компактно и наглядно изобразить алгоритм.

2.2 Описание использованных инструментов проектирования

На сегодняшний день существует множество сервисов, позволяющих без лишних затрат и навыков строить разного рода графики, диаграммы, блок-схемы и прочее.

Для построения блок-схем в рамках данной работы был выбран сервис Draw.io, предназначенный для построения диаграмм и схем. С его помощью можно создавать:

− UML-модели;

− графики;

− блок-схемы;

− формы;

− диаграммы и другое.

Сервис является бесплатным приложением с возможностью интеграции с Google Drive, DropBox, OneDrive и GitHub. Существует веб-версия, версия для установки на ПК, поддерживающая различные ОС (Windows, MacOS и Linux), а также мобильное приложение для смартфонов и планшетов, работающее как на iOS, так и на Android. Форматы для экспорта созданного изображения – PDF, GPG, SVG, XML и JPG. Сервис поддерживает 27 языков.

Сервис Draw.io обладает простым и удобным интерфейсом (рисунок 2.1). После перехода к графической части работы можно создавать схему самостоятельно или воспользоваться одним из предложенных шаблонов. Для удобства имеется верхняя панель управления с кнопками отмены действий, масштаба, прокрутки и некоторые другие. Поле построения имеет разметку, что позволяет выравнивать блоки друг относительно друга.

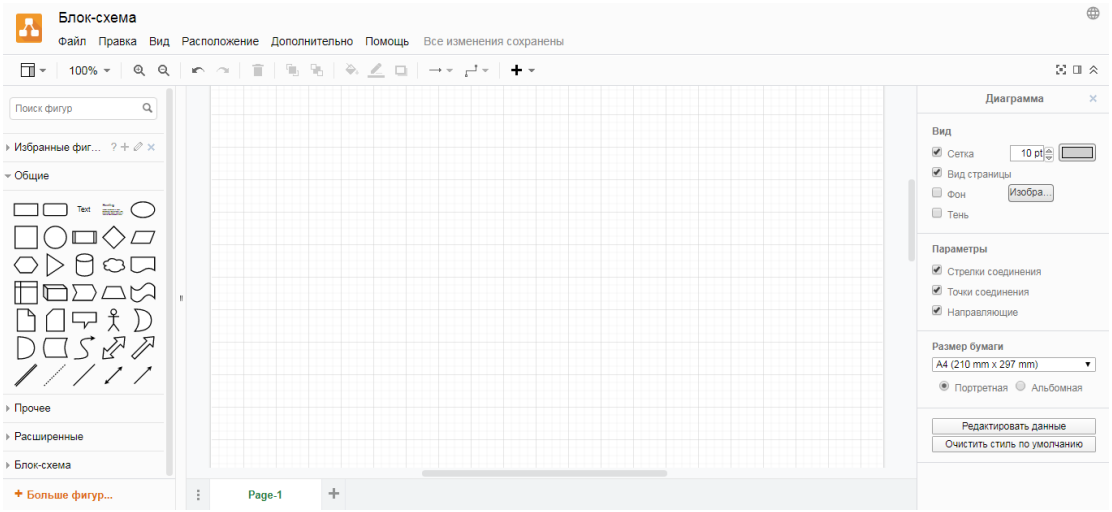


Рисунок 2.1 – Интерфейс веб-сервиса Draw.io

При построении схемы сервис автоматически выравнивает блоки по ширине и высоте, что значительно облегчает работу. Также есть возможность оформления деталей схемы по предпочтениям автора (например, изменение вида и размера кончика стрелки).

Таким образом, данный сервис имеет удобный, доступный для понимания и управления интерфейс и достаточно обширный набор возможностей. Сервис Draw.io значительно сокращает трудоёмкость и время построение схем и диаграмм, чем если это делать вручную в неприспособленных для этого редакторах.

2.3 Описание алгоритма программы

В настоящей работе для поиска незакономерных подпоследовательностей биологических цепочек реализуется алгоритм и на его основе строится система поиска.

Хранение исследуемой последовательности ДНК осуществляется в текстовом формате. Известные белковые последовательности также организованы в виде файла *.txt. Данный способ хранения выбран с целью экономии процессорного времени, так как считывание и обработка данных из БД занимает гораздо больше времени, чем работа с текстовым файлом.

Стоит отметить, что длины последовательностей заранее неизвестны.

Исследуемая последовательность ДНК должна быть подготовлена к дальнейшему анализу. Для этого в алгоритме необходимо осуществить:

1) перевод входной последовательности ДНК в РНК (транскрипция);

2) отчистку входной последовательности ДНК от некодирующих участков (интронов);

3) представление полученной в ходе транскрипции последовательности РНК в аминокислотный вид.

Помимо вышеперечисленных операций алгоритм должен выполнять следующие действия:

1) считывание биологических последовательностей неизвестной длины из файлов формата *.txt;

2) анализ последовательностей по алгоритму Смита-Ватермана;

3) вывод степени сходства и картины выравнивания анализируемых последовательностей.

С целью сокращения трудоёмкости программирования, алгоритм разбит на подзадачи, каждую из которых выполняет отдельная функция, последовательность выполнения которых отображены в блок-схеме алгоритма (рисунок 2.2).

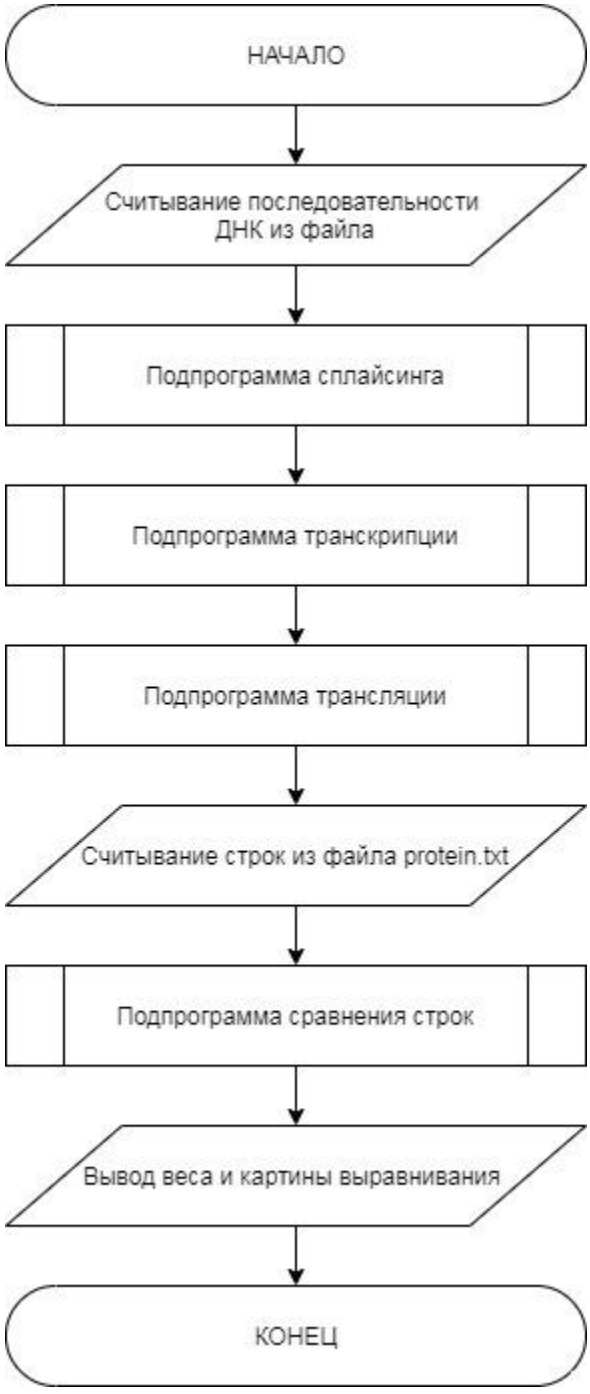


Рисунок 2.2 – Блок-схема реализуемого алгоритма

2.3.1 Описание подпрограммы транскрипции.

На первом этапе подготовки последовательности ДНК к дальнейшему анализу, необходимо произвести транскрипцию. Данный процесс подразумевает преобразование последовательности ДНК в РНК. Основное отличие между данными макромолекулами заключается в замене одного нуклеотида: тимина на урацил.

Соответственно, задача перевода ДНК в РНК сводится к замене в строке символа T символом U (рисунок 2.3).

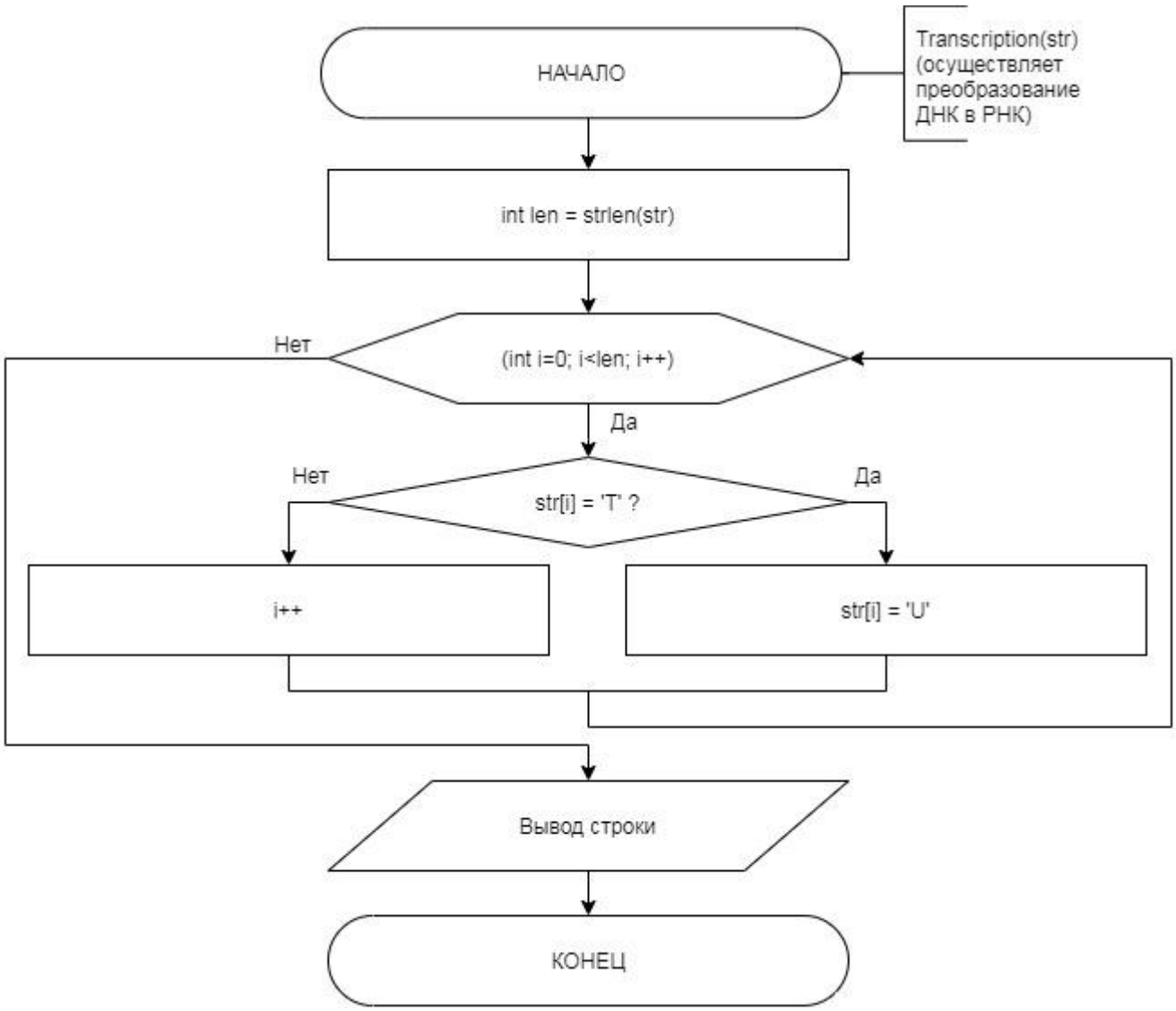


Рисунок 2.3 – Блок-схема подпрограммы «Транскрипция»

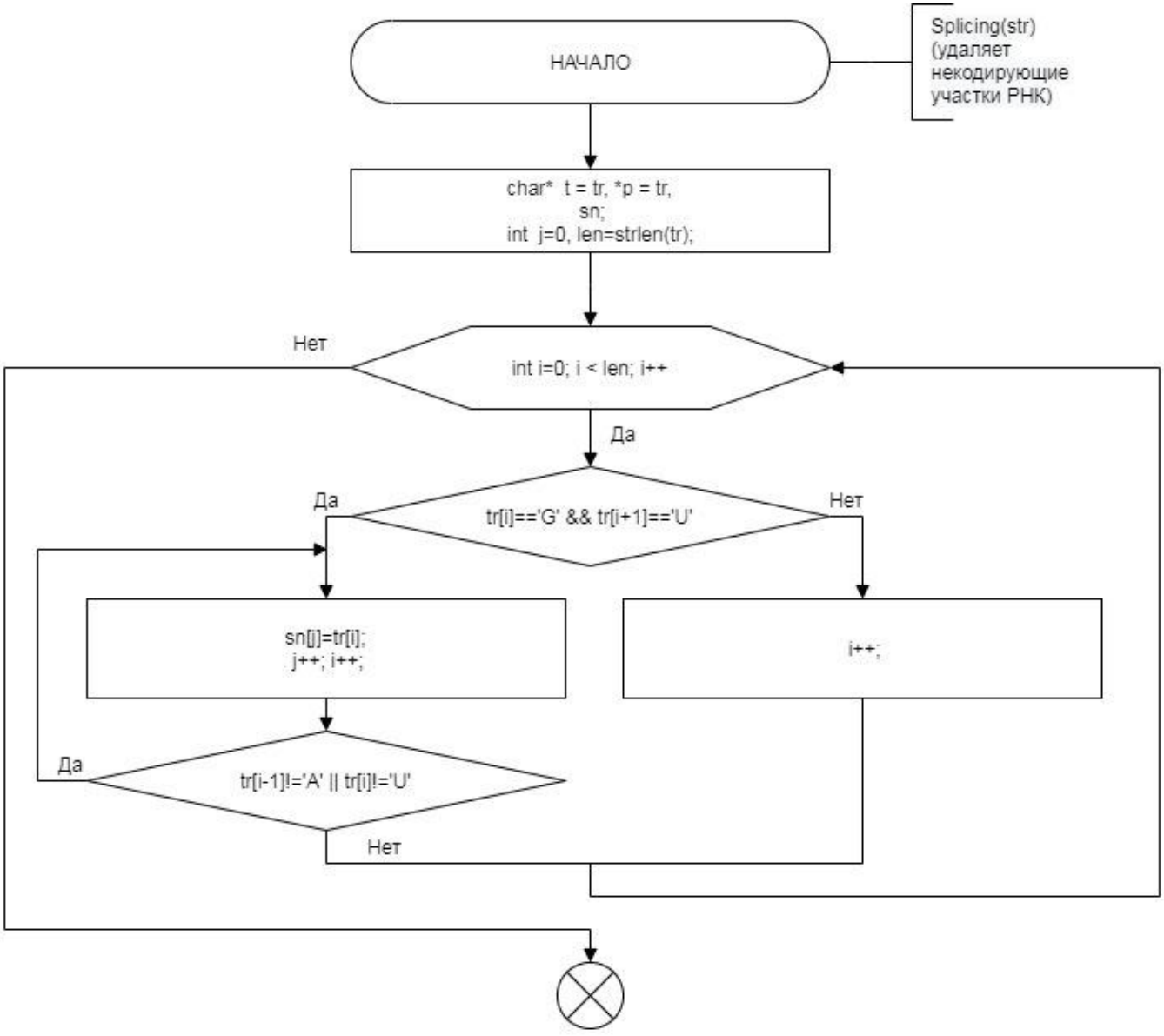
2.3.2 Описание подпрограммы сплайсинга.

На втором этапе подготовки последовательности ДНК к дальнейшему анализу, необходимо произвести удаление некодирующих участков (интронов). Данный процесс называется сплайсингом.

Известно, что интроны начинаются с «GT» и заканчиваются «AT». Однако сплайсинг осуществляется после трансляции. Длины интронов и их расположение в

строке заранее неизвестны. Однако, сплайсинг осуществляется после перевода в РНК, поэтому началом интрона будет являться «GU», а концом – «AU».

Для того, чтобы удалить записать некодирующую последовательность необходимо осуществить считывание подстроки в отдельную строку, а затем удалить её из исследуемой последовательности (рисунок 2.4).



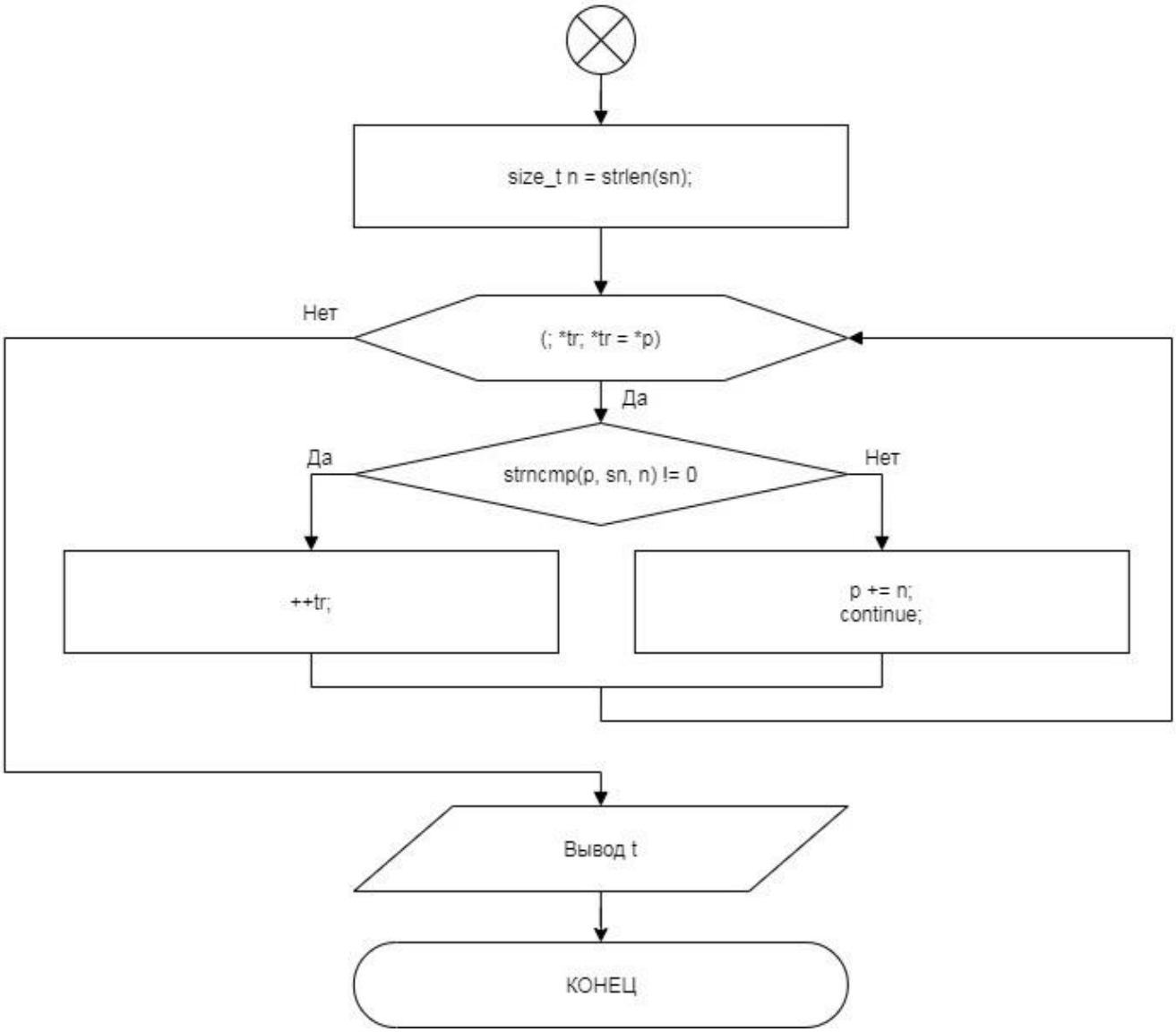


Рисунок 2.4 – Блок-схема подпрограммы «Сплайсинг»

2.3.3 Описание подпрограммы транскрипции.

На последнем этапе подготовки последовательности ДНК к анализу необходимо представить её в виде последовательности аминокислот, то есть осуществить процесс трансляции.

Одна аминокислота кодируется последовательностью трёх нуклеотидов, именуемой кодоном. Так как последовательность РНК кодируется при помощи четырёх нуклеотидов, вариаций кодонов 4^3=64. Все белки, обеспечивающие правильное функционирование организма, кодируются при помощи 20 аминокислот.

Исходя из вышесказанного, код избыточен. Однако, изменение каждого третьего элемента кодона, чаще всего, не изменяет аминокислоту.

Для реализации транскрипции необходимо воспользоваться таблицей перевода кодонов в аминокислоты (рисунок 1.7).

Подпрограмма организуется при помощи условного оператора if. Алгоритм работает следующим образом:

− объявляется строка для записи однобуквенных обозначений аминокислот;

− пока не достигнут конец строки, осуществляется последовательная проверка условий: если последовательность считанных символов равна UUU или UUC, тогда в новой строке записывается F, иначе – переход к следующему условию.

− по завершении проверки условий, подпрограмма возвращает и выводит полученную аминокислотную последовательность.

2.3.4 Описание подпрограммы выравнивания.

Прежде чем приступить к анализу последовательностей, необходимо считать с файла белки. Для этого создается динамическая структура.

Необходимо также определить штрафы для вставки, удаления, пробелов и замен при выравнивании. Стоит отметить, что не все преобразования имеют одинаковый вес. Поэтому при расчёте расстояния каждому виду присваивают различный вес. Для подсчёта сходства нуклеиновых кислот в общем случае добавляют +1 балл при совпадении символа и -1 балл за несовпадение.

Рассматривая аминокислотные последовательности, необходимо учесть тот факт, что замена одной аминокислоты родственной может не изменить функциональность белка, а замена разнородной – прекратить функционирование. Поэтому в подпрограмме для поиска оптимального выравнивания в белках используется матрица замещения BLOSUM62 (рисунок 2.5).

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	−1	5																		
Asn	−2	0	6																	
Asp	−2	−2	1	6																
Cys	0	−3	−3	−3	9															
Gln	−1	1	0	0	−3	5														
Glu	−1	0	0	2	−4	2	5													
Gly	0	−2	0	−1	−3	−2	−2	6												
His	−2	0	1	−1	−3	0	0	−2	8											
Ile	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
Leu	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
Lys	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
Met	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
Phe	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
Pro	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
Ser	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
Thr	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
Trp	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Tyr	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

Рисунок 2.5 – Матрица замещения BLOSUM62

Для лучшего понимания принципа работы алгоритма локального выравнивания Смита-Ватермана проводится его моделирование посредством MS Excel. Для моделирования берутся две строки разной длины в аминокислотном виде (RVA и QTVAK).

В первую очередь, для моделирования в соответствии с матрицей замещения BLOSUM62 строится таблица «Веса замен» (рисунок 2.6).

	Веса замен (BLOSUM62)			
		R (Arg)	V (Val)	A (Ala)
Q (Gln)		1	-2	-1
T (Thr)		-1	0	0
V (Val)		-3	4	0
A (Ala)		-1	0	4
K (Lys)		2	-2	-1

Рисунок 2.6 – Таблица «Веса замен», построенная средствами MS Excel

В соответствии с алгоритмом Смита-Ватермана строится матрица переходов (рисунок 2.7). Значения элементов верхней строки и левого столбца матрицы переходов считаются равными нулю. В процессе построения значение каждого из

остальных элементов $M_{i,j}$ i-й строки j-го столбца матрицы переходов вычисляется по формуле (2.1), значение g принимается в соответствии с таблицей «Веса замещения».

$$M_{i,j} = \max\{M_{i-1,j} + g; M_{i,j-1} + g; M_{i-1,j-1} + g; 0\}, i, j > 1, \quad (2.1)$$

Возвращается наибольший из аргументов.

	Матрица переходов			
	i строки	R	V	A
j столбцы	0	0	0	0
Q	0	1	0	0
T	0	0	1	1
V	0	0	5	5
A	0	0	5	9
K	0	2	3	8

Рисунок 2.7 – Матрица переходов, построенная средствами MS Excel

Для построения картины выравнивания, необходимо совершить обратный обход матрицы. На данном этапе моделирования строится таблица «Направления переходов» (рисунок 2.8).

Для каждого $M_{i,j} > 0$ направление перехода определяется следующим образом: если (2.2), то переход в элемент $M_{i,j}$ осуществляется из элемента $M_{i-1,j}$; если (2.3) – из элемента $M_{i,j-1} + g$; если (2.4) и $M_{i-1,j-1} > 0$ – из элемента $M_{i-1,j-1}$.

$$M_{i,j} = M_{i-1,j} + g, \quad (2.2)$$

$$M_{i,j} = M_{i,j-1} + g, \quad (2.3)$$

$$M_{i,j} = M_{i-1,j-1} + g. \quad (2.4)$$

	Направления переходов			
		R	V	A
Q		↑		
T			↖	←
V			↑	←
A			↑	↑
K		↑	↑	↑

Рисунок 2.8 – Таблица «Направления переходов», построенная средствами MS Excel

Таким образом, оптимальным локальным выравниванием последовательностей RVA и QTVAK является подстрока VA (рисунок 2.9).

Результат выравнивания				
	R	V	A	
Q	T	V	A	K
Максимальная степень сходства : 9				

Рисунок 2.9 – Таблица «Результаты выравнивания», построенная средствами MS Excel

Проектирование и моделирование сложных алгоритмов, значительно снижает временные затраты на реализацию и её трудоёмкость. Разбиение изученного алгоритма позволяет детально изучить структуру и возможности системы поиска незакономерных участков, частью которой он является.

3 Реализация

3.1 Описание инструментальных средств

3.1.1 Среда разработки Visual Studio 2013.

Microsoft Visual Studio – продукты компании Microsoft (интегрированная среда разработки ПО и ряд других инструментальных средств). С их помощью можно разрабатывать:

− консольные приложения;

− приложения с графическим интерфейсом (в том числе и WindowsForm;

− веб-сайты;

− веб-приложения;

− веб-службы.

Visual Studio 2013 среда программирования многофункциональных приложений: для Office, Windows Server, SharePoint, облачных систем и Интернета и многих других (рисунок 3.1).

Рисунок 3.1 – Среда разработки Visual Studio 2013

Для удобства работы среда разработки Visual Studio 2013:

− позволяет перемещать фрагменты кода при помощи сочетания клавиш;

− предоставляет разработчику возможность писать форматированный код;

− подсказывает и дополняет код;

− отображает предупреждения и сообщения об ошибках;

− предоставляет возможность перемещаться между классами, функциями и символами;

− показывает разработчику место в коде, где была описана или вызвана функция и другое.

Среда разработки имеет все необходимые инструменты для программирования на различных языках: C#, C++, HTML, JavaScript, .NET. В Visual Studio 2013 доступна гибкая среда для интеграции с инструментами Eclipse и Xcode, это позволяет создавать приложения на разных платформах. Среда содержит средства проверки качества и работоспособности созданных проектов, что обеспечивает своевременное обнаружение допущенных в коде ошибок и недочётов.

Помимо всех выше перечисленных свойств и возможностей Visual Studio 2013 позволяет пользователю настроить оформление по предпочтению: темное или светлое, а также имеет удобный интерфейс (рисунок 3.2).

Таким образом, среда разработки Visual Studio 2013 позволяет эффективно управлять полным жизненным циклом программного продукта: с этапа проектирования и реализации до эксплуатации и дальнейшей поддержки.

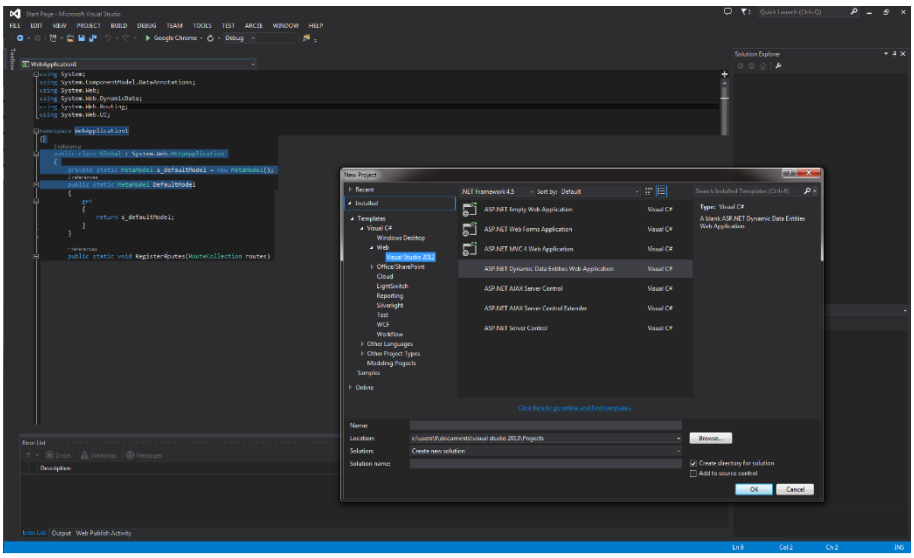


Рисунок 3.2 – Окно создания проекта Visual Studio 2013

3.2 Описание системы поиска

В данном разделе будет представлено описание системы, реализованной в рамках выполнения ВКР. Данная система предназначается для поиска гомологии между двумя генетическими последовательностями.

Для того, чтобы осуществить выравнивание с помощью разработанной системы необходимо сохранить исследуемую входную последовательность, это может быть, как ДНК, так и аминокислотная последовательность, в текстовый файл. Перед тем, как начать работу, система потребует выбрать файл, в котором хранится исследуемая строка. Стоит отметить, что длина входной последовательности на момент её обработки является неизвестной. Последовательность ДНК строится на основе четырёхбуквенного алфавита (A, C, T, G), Белковая последовательность – на основе двадцати буквенного алфавита представляющего собой однобуквенные наименования аминокислот входящих в её состав (рисунок 3.3).

Символ	Обозначаемые остатки	Мнемоническое правило
A	A	**A**denine
C	C	**C**ytosine
G	G	**G**uanine
T	T	**T**hymine
U	U	**U**racil
R	A или G	pu**R**ine
Y	C, T или U	p**Y**rimidines
K	G, T или U	**K**етон
M	A или C	Содержит а**М**иногруппу
S	C или G	**S**trong interaction (три водородные связи)
W	A, T или U	**W**eak interaction (две водородные связи)
B	Любой, кроме A	после A
D	Любой, кроме C	после C
H	Любой, кроме G	после G
V	Любой, кроме T и U	после U
N	Любой	**N**ucleic acid
X	Любой	Неизвестная величина

Рисунок 3.3 – Аминокислоты и соответствующие им однобуквенные наименования

При запуске приложения, открывается главное окно системы (рисунок 3.4). В зависимости от представления входной последовательности, пользователем осуществляется нажатие кнопки «Поиск по базе данных» или «Выравнивание белков» и переход к дальнейшим действиям: перевод последовательности ДНК в РНК, удаление некодирующих участков и приведение к аминокислотному виду; осуществление выравнивания белковых последовательностей. Работа системы может быть завершена нажатием кнопки «Завершить».

Рисунок 3.4 – Главное окно системы поиска

В главном окне расположена кнопка «О программе». После её нажатия системой открывается окно с основной информацией о системе (рисунок 3.5), а также о поддерживаемых форматах входных данных (рисунок 3.6) и о параметрах, выводимых в качестве результата работы системы (рисунок 3.7).

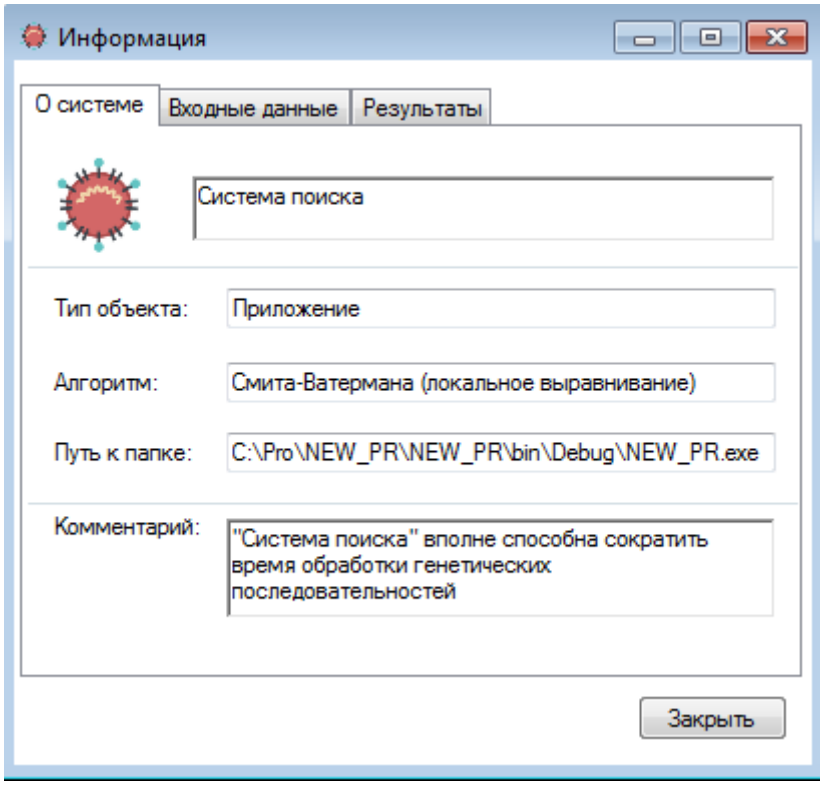


Рисунок 3.5 – Окно с информацией о системе

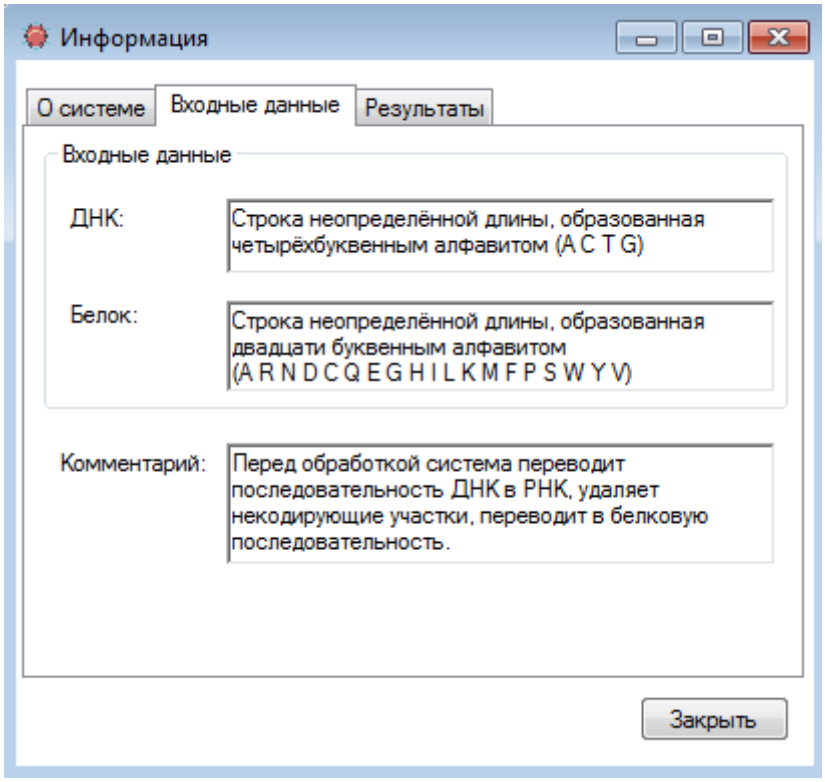


Рисунок 3.6 – Окно с информацией о входных данных

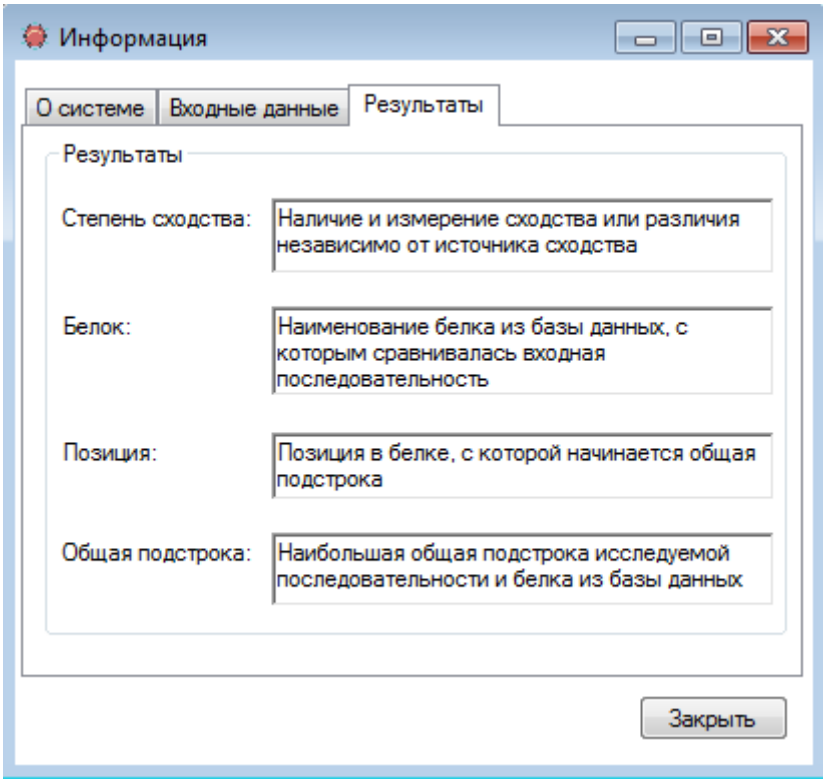


Рисунок 3.7 – Окно с информацией о параметрах, выводимых в качестве результата работы системы

3.2.1 Поиск по базе данных.

Нажатие кнопки «Поиск по базе данных» на главном окне системы поиска осуществляет переход к подготовке входной последовательности ДНК, осуществляемой в три основных этапа: транскрипция, сплайсинг, трансляция (рисунок 3.8). Все преобразования выполняются непосредственно системой, пользователю лишь необходимо выбрать файл с исследуемой последовательностью (рисунок 3.9).

На первом этапе подготовки ДНК необходимо представить в виде последовательности РНК. Для этого в системе реализована проверка входной строки на наличие в ней нуклеотида T (тимина) и замена всех его вхождений на U (урацил).

После нажатия кнопки «Транскрибировать», система выводит преобразованную последовательность на экран (рисунок 3.10) и передаёт её на дальнейшую обработку.

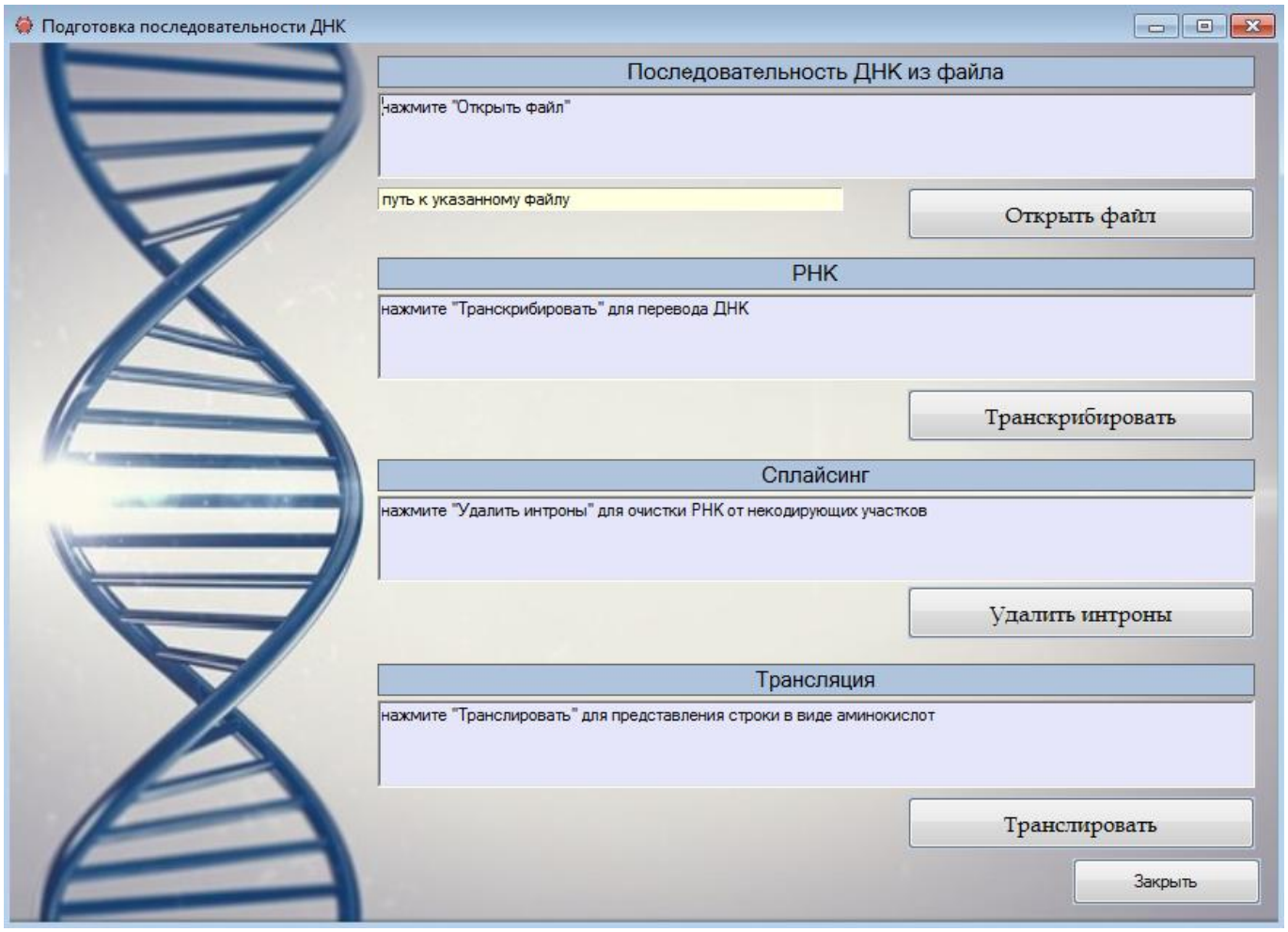


Рисунок 3.8 – Окно «Подготовка последовательности ДНК»

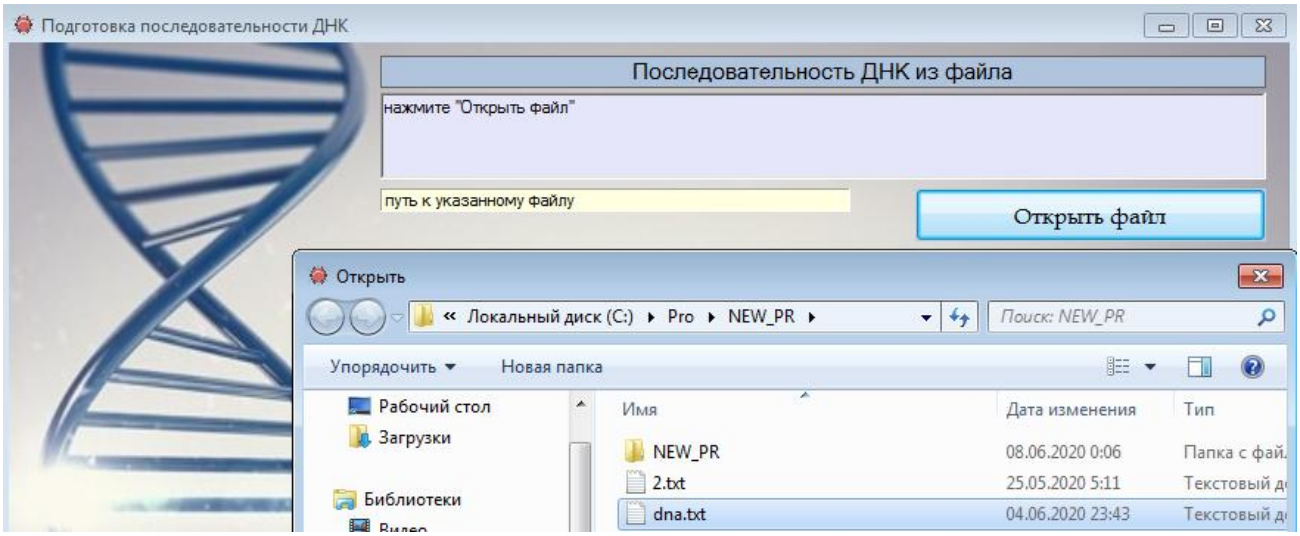


Рисунок 3.9 – Выбор файла с исследуемой ДНК

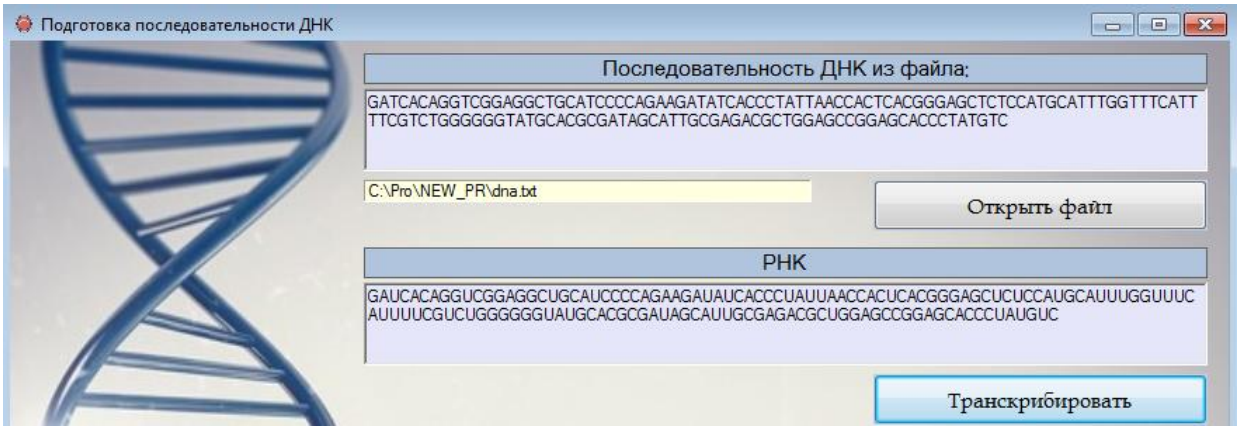


Рисунок 3.10 – Вывод преобразованной последовательности ДНК

Вторым этапом подготовки является удаление интронов, некодирующих участков, из полученной последовательности РНК. Известно, что данные участки начинаются с «GU» и заканчиваются «AU». Длина интронов неизвестна.

Для того, чтобы удалить интроны, в системе реализован просмотр последовательности РНК и удаление всех некодирующих подстрок, если таковые имеются. Таким образом, осуществляется перезапись строки в новую строку с последовательным исключением подстрок-интронов.

Результат сплайсинга выводится на экран (рисунок 3.11) и последовательность передаётся на заключительный этап подготовки.

Рисунок 3.11 – Вывод результата сплайсинга

Этап трансляции полученной строки представляет из себя замену трёх символов строки (кодона) на соответствующую аминокислоту. Так как длина и количество удаляемых на предыдущем этапе подстрок была неизвестна, переданная строка также имеет неопределённую длину. Перед тем, как перейти к процессу преобразования последовательности в аминокислотный вид, система определяет её длину.

Транскрипция реализована в виде считывания строки по три символа, и в соответствии с их комбинацией, в новую строку записывается однобуквенное обозначение аминокислоты.

Таким образом, системой формируется последовательность аминокислот длиной в 3 раза меньше исходной. Результат выводится на экран и система выводит сообщение об успешном завершении подготовки ДНК к выравниванию (рисунок 3.12).

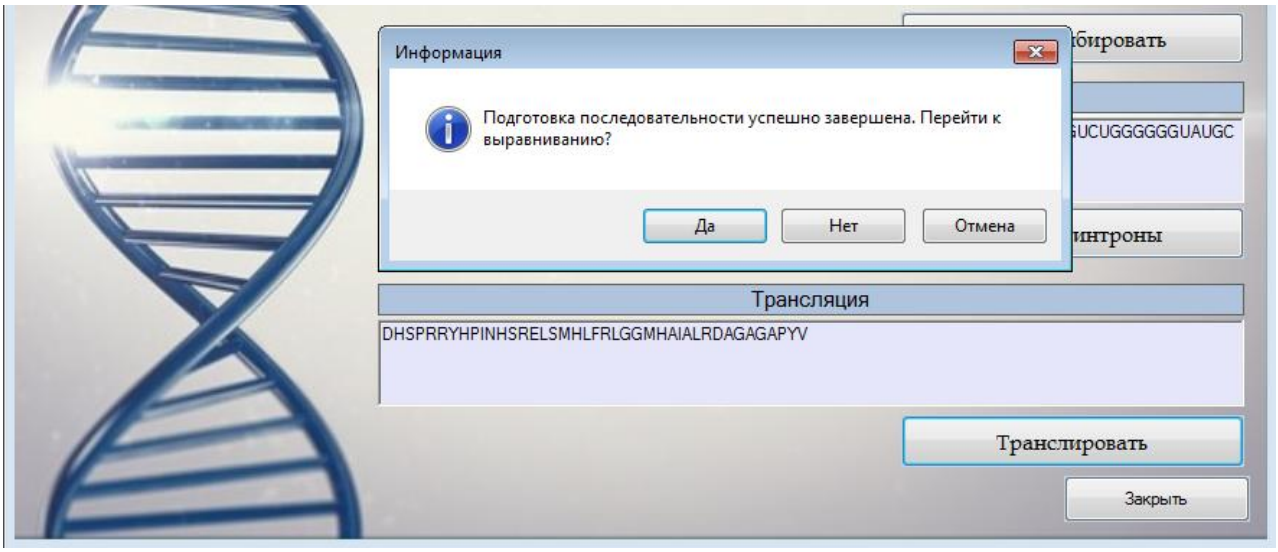


Рисунок 3.12 – Вывод результата транскрипции и сообщения о завершении подготовки ДНК

После того, как пользователь дал согласие на переход к выравниванию, система открывает окно «Выравнивание последовательностей» (рисунок 3.13). Кнопка «База

белков» позволяет просмотреть наименования и последовательности белков, взятых из PDB и сохранённых в текстовом файле (рисунок 3.14).

Рисунок 3.13 – Окно «Выравнивание последовательностей»

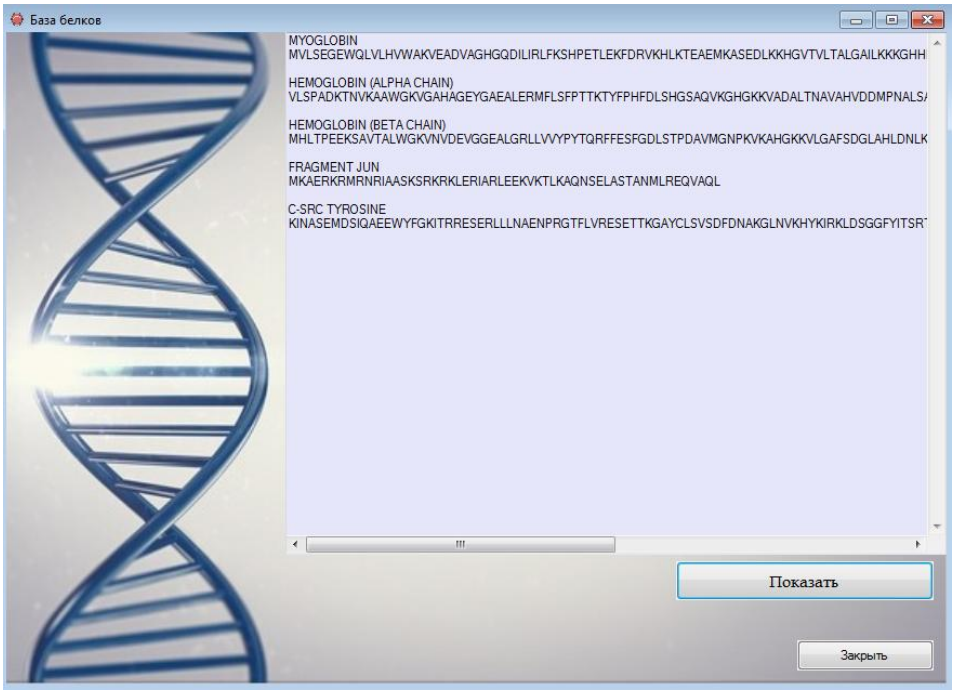


Рисунок 3.14 – Окно «База белков»

Выполнение процесса выравнивания в системе реализовано в несколько этапов:

− построение матрицы весов для сравниваемых последовательностей;

− построение матрицы переходов для сравниваемых последовательностей;

− определение степени сходства;

− поиск максимальной общей подстроки и её стартовой позиции.

3.2.1.1 Построение матрицы весов.

Матрица весов для сравниваемых последовательностей реализована в виде двумерного массива типа int. Количество строк массива равно длине первой последовательности, количество столбцов – длине второй.

Заполняется матрица весов в соответствии с матрице замен BLOSUM62, которая объявлена в качестве глобальной переменной. Замена и совпадение каждой аминокислоты имеет свой штраф. В матрице замен аминокислоты располагаются в алфавитном порядке.

Для того, чтобы заполнить матрицу весов создана функция, которая, получив на вход строку args, создаёт соответствующий ей одномерный массив типа int: indexArgs. Размерность массива равна длине переданной строки, а элементы принимают значения от 0 до 19: если текущий элемент строки args[i] равен обозначению первой по алфавиту аминокислоты 'A', в массив indexArgs[j] записывается 0, если args[i] равен 'R' – indexArgs[j] = 1 и так далее.

Таким образом, элемент массива indexArgs соответствует индексу аминокислоты в матрице замен.

Заполнение матрицы весов $W_{i,j}$ осуществляется по формуле (3.1).

$$W_{i,j} = mtx[indexT[i]][indexP[j]], \quad (3.1)$$

где mtx[][] – матрица замен BLOSUM62, indexT[i], indexP[j] – массивы индексов аминокислот первой и второй строки соответственно.

Для каждой пары последовательностей в текстовый файл WM.txt, расположенный в папке с проектом, записывается матрица весов (рисунок 3.15).

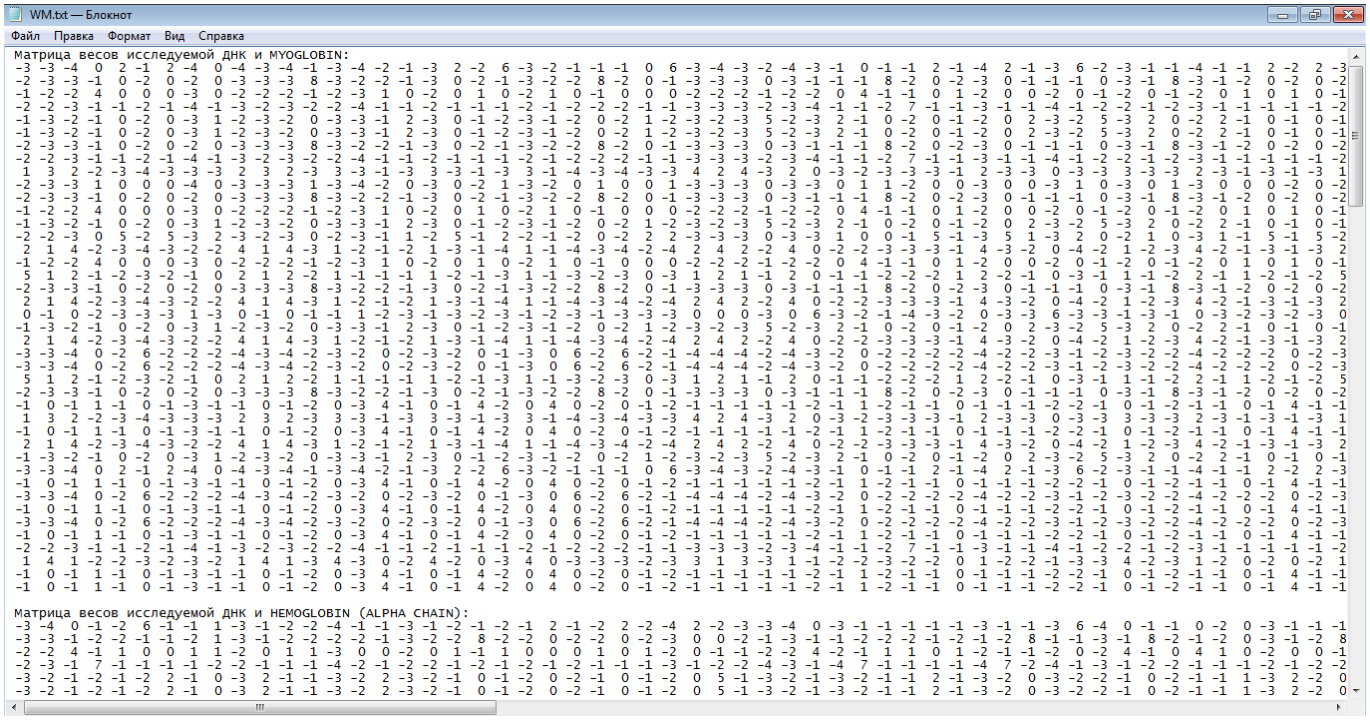


Рисунок 3.15 – Файл WM.txt с матрицами весов

3.2.1.2 Построение матрицы переходов.

Матрица переходов представляет из себя двумерный массив типа int с размерностью, на единицу большей, чем длины сравниваемых строк. Первая строка и первый столбец матрицы переходов $T_{i,j}$ заполняются нулями (3.2) и (3.3).

$$T_{i,0} = 0, \tag{3.2}$$

$$T_{0,j} = 0. \tag{3.3}$$

Заполнение остальной части матрицы осуществляется по формуле (3.4), советующей алгоритму локального выравнивания Смита-Ватермана.

$$T_{i,j} = \max\{T_{i-1,j} + W_{i-1,j-1}; T_{i,j-1} + W_{i-1,j-1}; T_{i-1,j-1} + W_{i-1,j-1}; 0\}, \tag{3.4}$$

Матрицы переходов для каждой пары сравниваемых последовательностей записывается в текстовый файл TM.txt, расположенный в папке с проектом (рисунок 3.16).

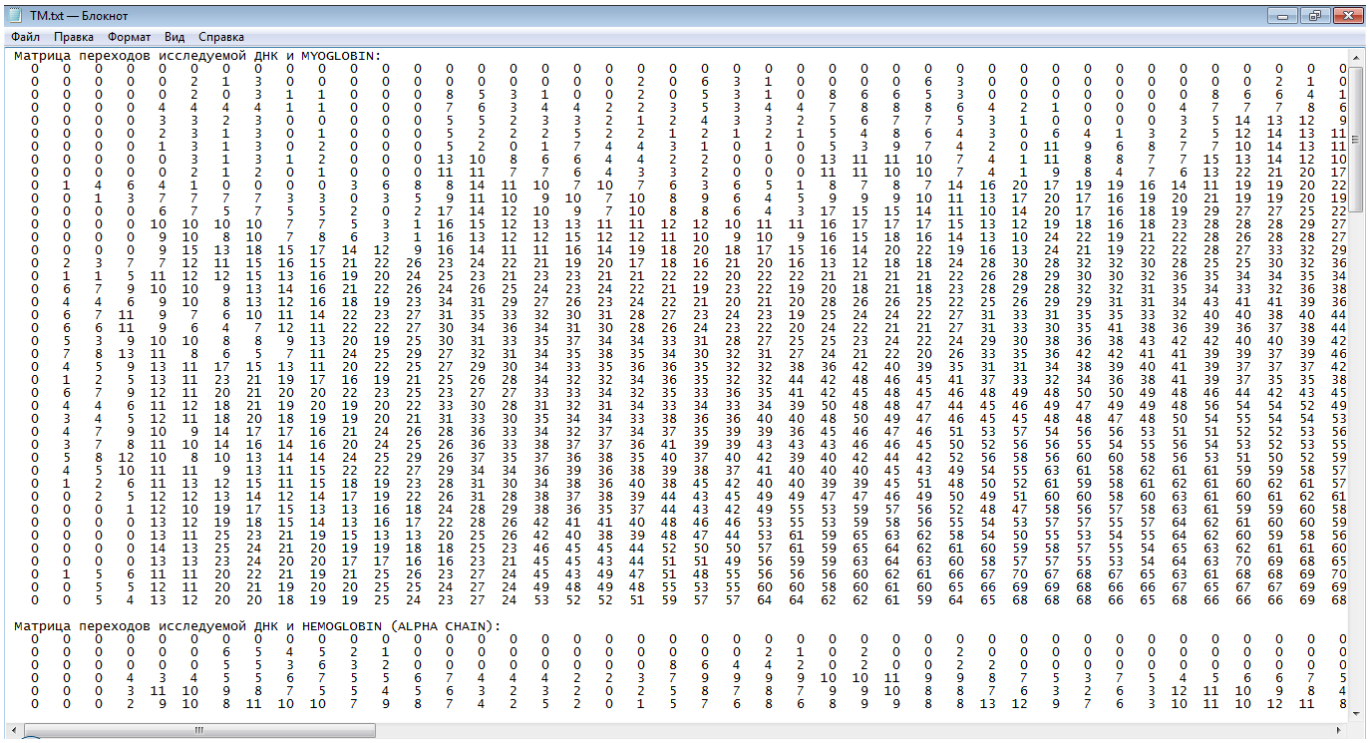


Рисунок 3.16 – Файл TM.txt с матрицами переходов

Степенью сходства последовательностей является наличие или измерение сходства и различия, независимо от источника сходства. За степень сходства двух биологических последовательностей принимается максимальный элемент матрицы переходов.

3.2.1.3 Поиск максимальной общей подстроки.

Заключительным этапом выполнения выравнивания является поиск наибольшей общей подстроки. Для её выявления необходимо заполнить двумерный массив str[n][m], где n и m – длины сравниваемых строк. Если элемент str1[i] первой последовательности равен элементу str2[j] второй последовательности, тогда для i=0 или j=0 элемент str[i][j] = 1; для i и j > 0 элемент str[i][j] = str[i-1][j-1] + 1. Максимальное значение массива str[i][j] является длиной наибольшей подстроки. В

ходе вычисления максимального значения maxValue производится запись индекса элемента массива maxI, которой позволяет вычислить стартовую позицию StartPos искомой подстроки по формуле (3.5).

$$StartPos = \max I + 1 - \max Vaue\,. \tag{3.5}$$

Все результирующие выравнивания вычисляются и выводятся после нажатия пользователем кнопки «Выполнить выравнивание строк» (рисунок 3.17).

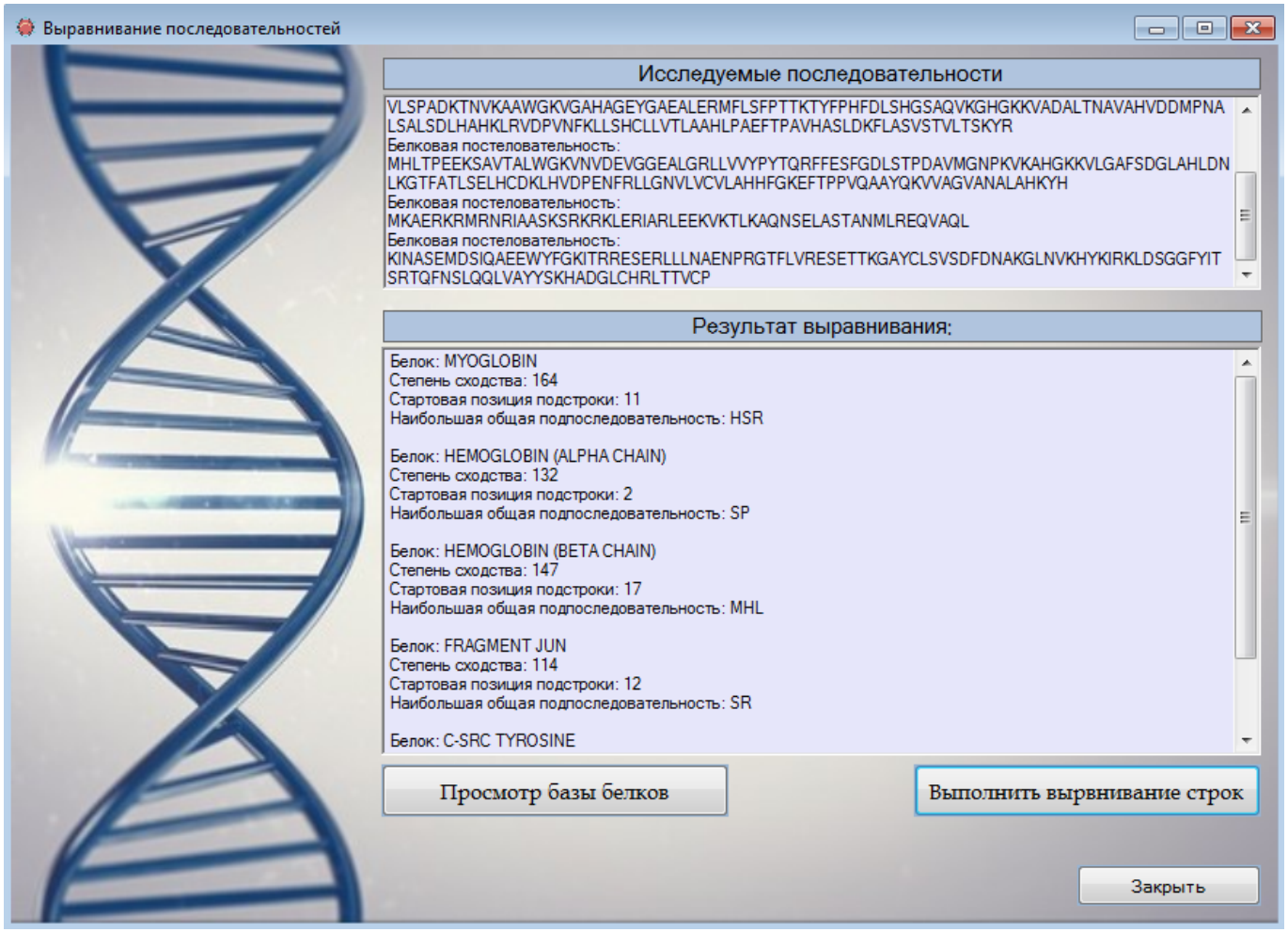


Рисунок 3.17 – Вывод результатов выравнивания

3.2.2 Выравнивание белков

Кнопка «Выравнивание белков» главного окна системы поиска открывает одноимённое окно (рисунок 3.18). Возможность осуществить выравнивание белков создана для сравнения последовательностей с целью:

− выявить принадлежность некоторых особей одному виду;

− выявить эволюционные отклонения;

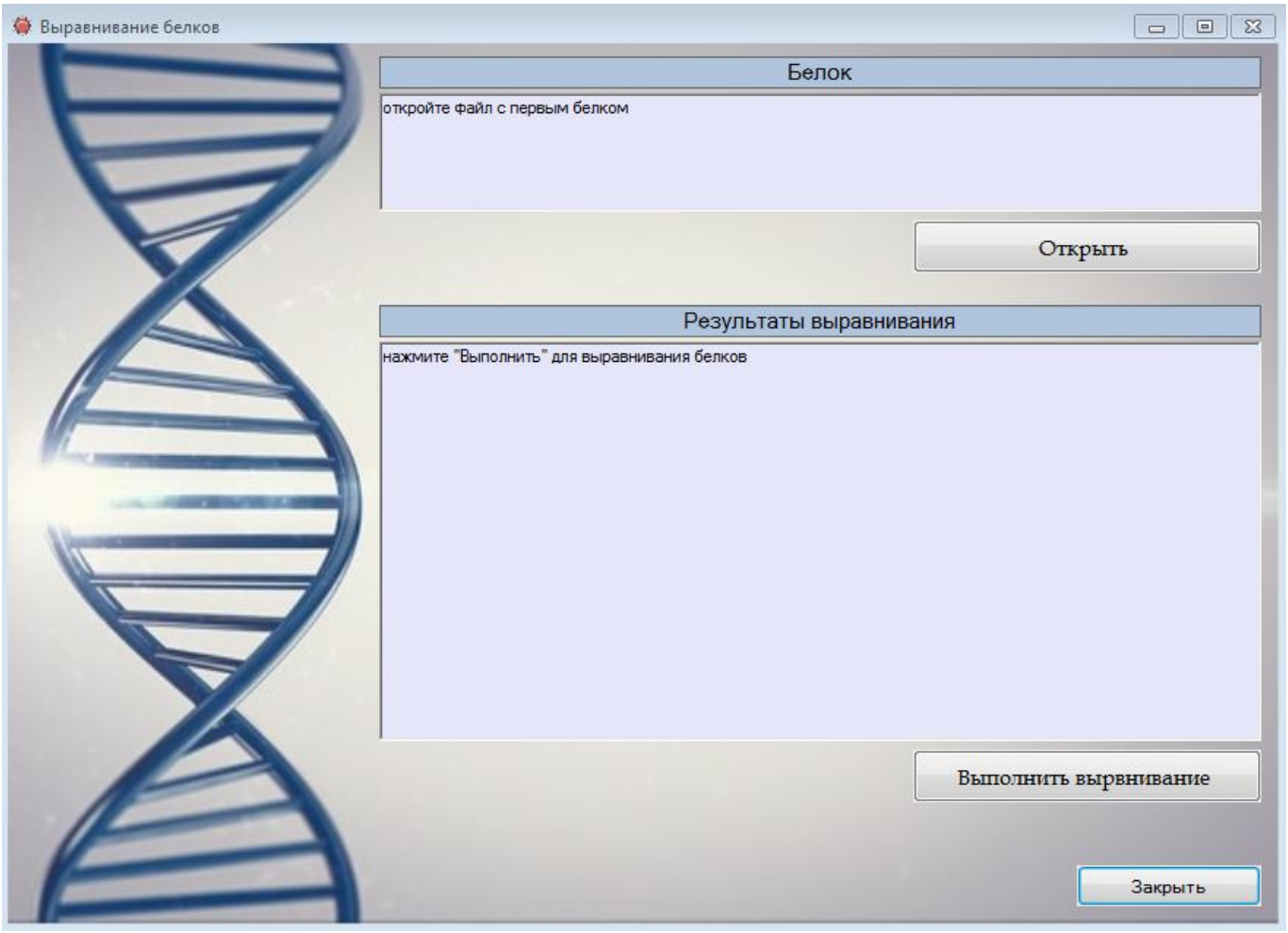


Рисунок 3.18 – Окно «Выравнивание белков»

На начальном этапе работы пользователем выбирается файл, в котором хранится исследуемая белковая последовательность (рисунок 3.19).

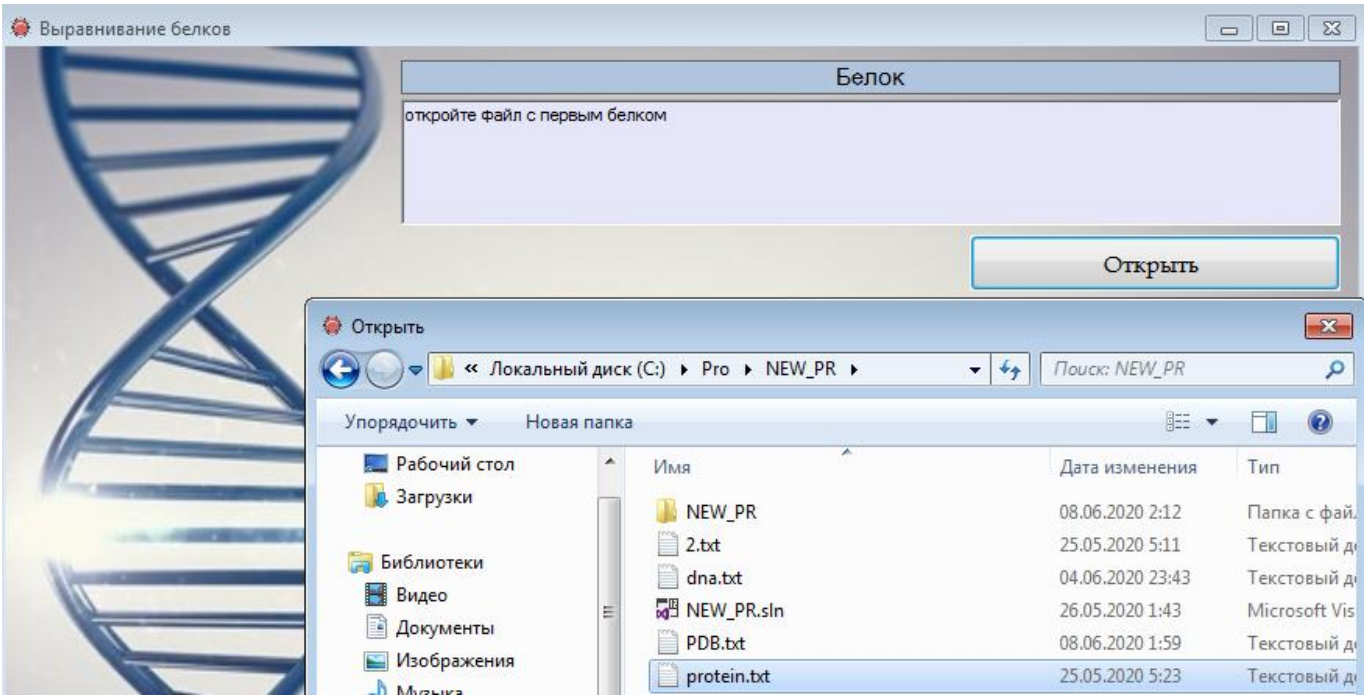


Рисунок 3.19 – Выбор файла с исследуемой белковой последовательностью

Перед тем, как выполнить выравнивание белков, система выводит сообщение о необходимости выбора файла с белками, с которыми предполагается произвести выравнивание (рисунок 3.20). После того, как файл выбран, система выполняет выравнивание и выводит результаты (рисунок 3.21). Матрицы весов и переходов выводятся в файлы WT1.txt TW1.txt соответственно.

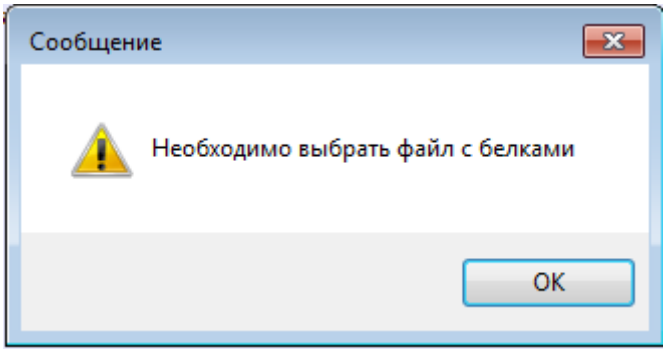


Рисунок 3.20 – Сообщение о необходимости выбора файла

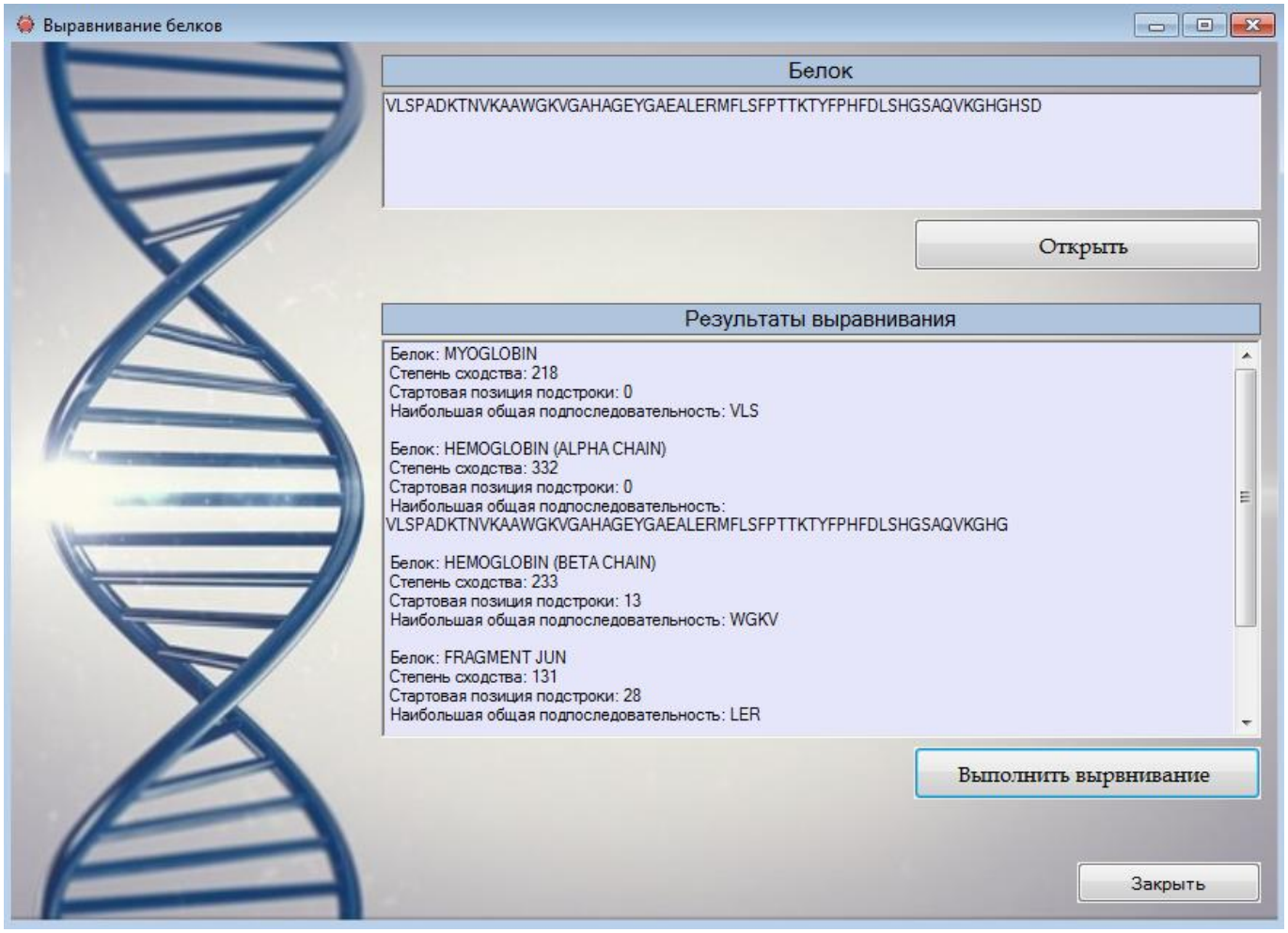


Рисунок 3.21 – Результаты выравнивания белков

ЗАКЛЮЧЕНИЕ

В настоящее время развитие биоинформатики открывает большие возможности в изучении генома и генов.

Работая с моделями биологических систем ею был внесён значительный вклад в сравнительную геномику, моделирование молекул биологических веществ и выявлении их мутаций.

Главной целью биоинформатики является исследование биологических цепочек, изучение свойств макромолекул ДНК, РНК и белков, определяемых некоторым расположением нуклеотидов и аминокислот друг относительно друга. Для преобразования информации, биологических последовательностей в символьные модели биоинформатиками используется метод секвенирования. Однако появление быстродейственных методов привело к накоплению большой базы данных.

Данную проблему можно решить посредством создания нового ПО для хранения, управления и анализа биологических данных.

В настоящее время наиболее распространенными алгоритмами анализа последовательностей являются алгоритмы Смита-Ватермана и Нидлмана-Вунша. Однако большинство реализаций этих алгоритмов требуют значительное количество времени и ресурсов.

В ходе выполнения работы была изучена структура генома, основные алгоритмы выравнивания последовательностей, проведен сравнительный анализ алгоритма Смита-Ватермана и Нидлмана-Вунша.

Реализованная система позволяет осуществить выравнивание последовательности ДНК, предварительно выполнив её подготовку: перевод в РНК, удаление некодирующих участков – интронов, приведение к аминокислотному виду. Также система позволяет выполнить сравнение белков, которое позволит определить принадлежат ли последовательности особям одного вида; являются ли различия эволюционными изменениями или белым шумом. Время требуемое на выполнение выравнивания составляет около 2-3 минут.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Назипова, Н.Н. Большое данные в биоинформатике / Н.Н. Назипова. – Москва, 2017. – Т. 12, №1. – С. 102–119.

2 Стефанов, В.Е. Биоинформатика: учебник для академического бакалавриата / В.Е. Стефанов. – Москва: Юрайт, 2017. – С. 10–219.

3 Зленко, Д.В. Анализ последовательностей: каф. Биофизики биологического ф-та МГУ / Д.В. Зленко. – Москва – С. 6–12.

4 Бочарников, А.В. Разработка программного пакета для анализа сходства регуляторных районов ткань-специфичных генов: маг. дис. ст. / Бочарников Андрей Васильевич – Новосибирск, 2010. – С. 6–18.

5 Огурцов, А.Н. Методы биоинформационного анализа: учеб. пособие по курсу «Биоинформатика и информационная биотехнология» / А.Н. Огурцов. – Харьков: НТУ «ХПИ», 2011. – С. 14–37.

6 Ахметов, И.З. Кластеризация биоинформатических данных: маг. дис. ст. / Ахметов Ильшат Зуфарович – Казань, 2019. – С. 7–9 .

7 Потапов, А.Н. Оптимизация алгоритма выравнивания биологических последовательностей на GPU / А.Н. Потапов, Е.А. Кольчугина // Молодой учёный. – 2014. – №3. – 62 с.

8 Лахно, В.Д. Математическая биология и биоинформатика / В.Д. Лахно // Вестник Российской академии наук. – 2011. – Т.81, № 9.

9 Порозов, Ю.Б. Биоинформатика и средства компьютерного анализа и визуализации макромолекул / Ю.Б. Порозов // Саратовский научно-медицинский журнал. – 2010. – Т.6, № 2. – 274 с.

10 Несговорова, Г.П. Биоинформатика: пути развития и перспективы / Г.П. Несговорова // Информатика в науке и образовании: сбор. ст. / под ред. В.Н.Касьянова. – Новосбириск, 2012. – С. 83–85.

11 Огурцов, А.Н. Введение в биоинформатику: учеб. пособие по курсу «Биоинформатика и информационная биотехнология» / А.Н. Огурцов. – Харьков: НТУ «ХПИ», 2011. – 207 с.

Приложение А

(обязательное)

Перечень иллюстрационно-графического материала ВКР

Перечень рисунков в ВКР

Перечень таблиц в ВКР

Приложение Б

(справочное)

Руководство пользователя

Б.1 Введение

1) Область применения

Система поиска используется для поиска оптимального локального выравнивания по алгоритму Смита-Ватермана.

2) Краткое описание возможностей

− перевод последовательности ДНК в РНК;

− удаление некодирующих участков из последовательности РНК;

− приведение последовательности к аминокислотному виду;

− поиск последовательности ДНК по базе данных белков;

− выравнивание белковых последовательностей.

3) Уровень подготовки пользователя

Пользователь системы должен иметь опыт работы с ОС Windows (XP/7), а также базовые знания генетики и биоинформатики.

Б.2 Назначение и условия применения

Система поиска предназначена для сокращения временных затрат на осуществление следующих видов операций:

− перевод последовательности ДНК в РНК;

− удаление некодирующих участков из последовательности РНК;

− приведение последовательности к аминокислотному виду;

− поиск последовательности ДНК по базе данных белков;

− выравнивание белковых последовательностей.

Минимальные программные и аппаратные требования, необходимые для корректной работы системы поиска приведены в таблице Б.1.

Таблица Б.1 – Минимальные программные и аппаратные требования к системе

Критерий	Требование
Операционная система	Windows XP и выше
ОЗУ	1 Гб
Свободное место на диске	1 Гб
Периферия	Стандартный монитор разрешением 1024x768 и выше; мышь.

Б.3 Подготовка к работе

Б.3.1 Подготовка входных данных.

Перед началом работы с системой поиска пользователю необходимо:

1) записать исследуемую последовательность ДНК или белка в текстовый файл;

2) записать последовательность исследуемого белка в текстовый файл;

3) при необходимости дополнить файл PDB.txt с базой белков (наименование и последовательность белка, формат записи аналогичен приведённому на рисунке Б.1).

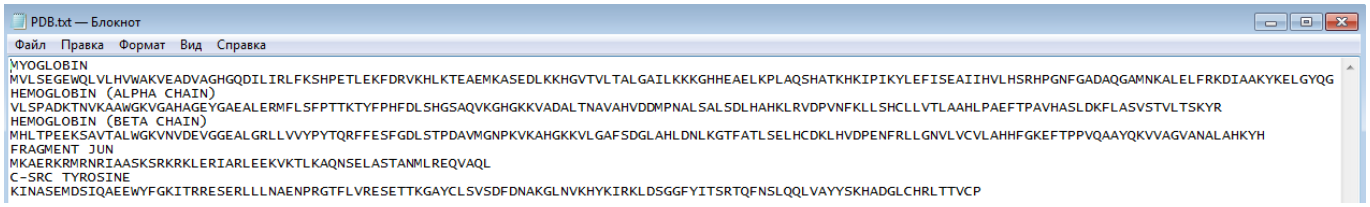


Рисунок Б.1 – Файл с базой белков PDB.txt

Перед началом работы с системой поиска пользователю необходимо выполнить следующие действия:

1) скопировать папку с проектом на компьютер;

2) записать исследуемую последовательность ДНК или белка в текстовый файл;

3) запустить исполняемый файл.

Б.4 Описание операций

Данный раздел содержит описание пользовательских операций для выполнения основных задач.

Б.4.1 Задача «Подготовка последовательности ДНК».

Выполнение данной задачи возможно при наличии входных данных:

1) текстового файла с исследуемой последовательностью ДНК;

2) подготовленного файла PDB.txt с базой белков.

Основные действия:

1) нажать на кнопку «Выравнивание ДНК», откроется окно «Подготовка последовательности ДНК» (рисунок Б.2);

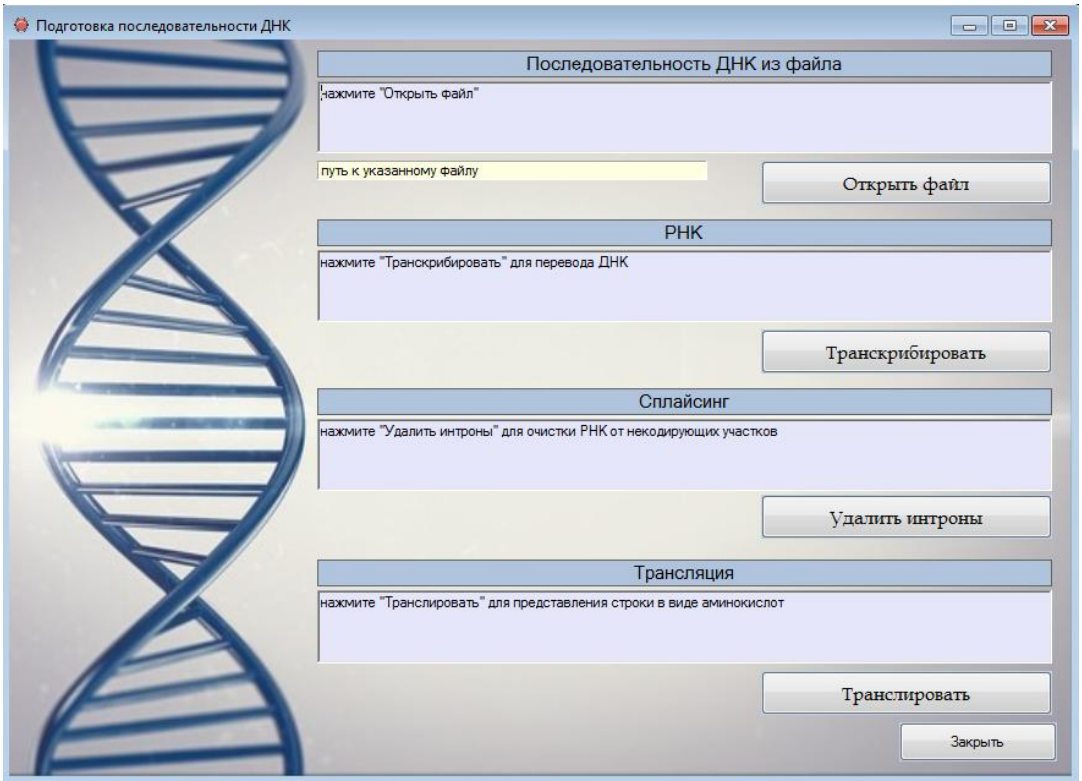


Рисунок Б.2 – Окно «Подготовка последовательности ДНК»

2) нажать на кнопку «Открыть файл», откроется диалоговое окно;

3) выбрать файл с исследуемой ДНК, в соответствующий TextBox выводится считанная последовательность (рисунок Б.3);

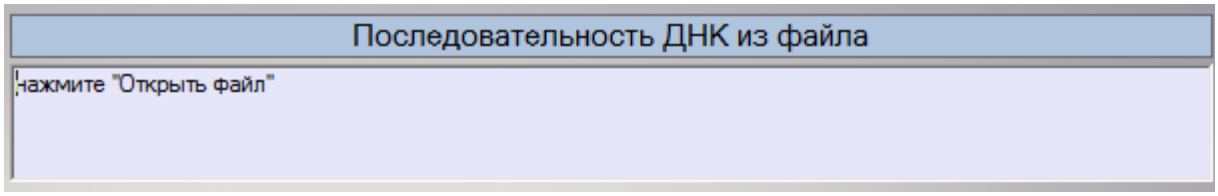


Рисунок Б.3 – TextBox просмотра последовательности ДНК

4) нажать на кнопку «Транскрибировать», в соответствующий TextBox выводится полученная последовательность РНК (рисунок Б.4);

Рисунок Б.4 – TextBox просмотра последовательности РНК

5) нажать на кнопку «Удалить интроны», в соответствующий TextBox выводится полученная последовательность (рисунок Б.5);

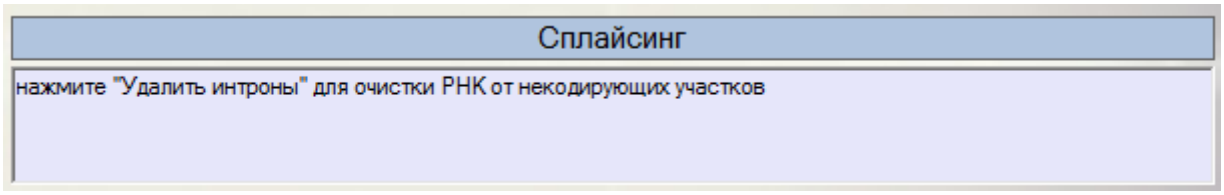


Рисунок Б.5 – TextBox просмотра очищенной последовательности

6) нажать на кнопку «Транслировать», в соответствующий TextBox выводится полученная аминокислотная последовательность (рисунок Б.6).

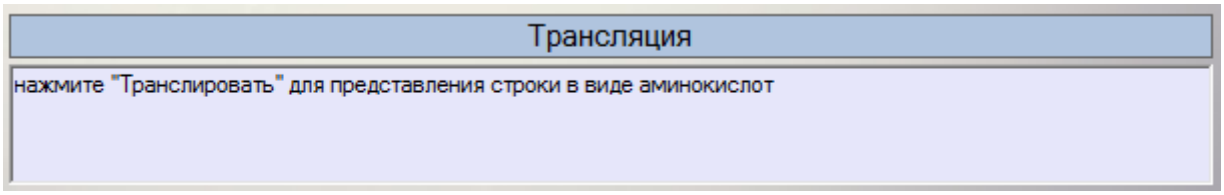


Рисунок Б.6 – TextBox просмотра аминокислотной последовательности

После выполнения всех вышеперечисленных действий системой поиска открывается окно «Информация» с сообщением об успешном завершении подготовки и переходе к выравниванию (рисунок Б.7).

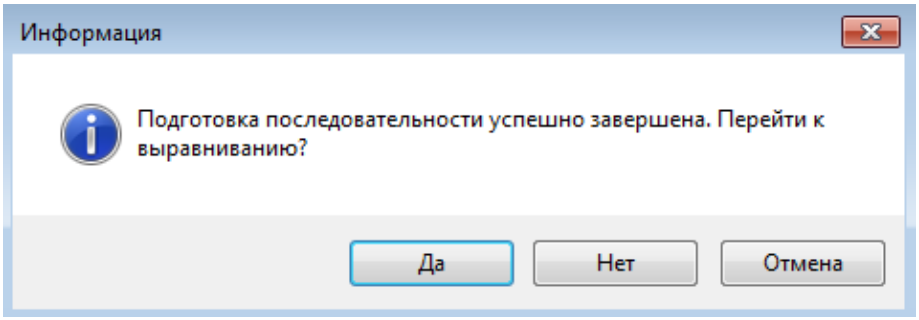


Рисунок Б.7 – Окно «Информация»

В зависимости от нажатой кнопки на окне «Информация» системой осуществляется одно из следующих действий:

− закрывает окно «Информация» при нажатии кнопки «Нет» или «Отмена»;

− закрывает окно «Информация» и открывает окно «Выравнивание последовательностей» при нажатии кнопки «Да».

Б.4.2 Задача «Поиск ДНК по базе данных».

Для выполнения данной операции необходимо:

1) выполнить действия, описанные в пункте Б.4.1, откроется окно «Выравнивание последовательностей» (рисунок Б.8);

2) нажать на кнопку «Выполнить выравнивание строк», в окне отобразятся исследуемые последовательности (рисунок Б.9) и результаты выравнивания (рисунок Б.10).

Для просмотра базы белков необходимо:

1) нажать кнопку «База белков», откроется окно с аналогичным названием (рисунок Б.11);

2) нажать кнопку «Показать», в окне отобразятся белки, имеющиеся в текстовом файле PDB.txt.

Рисунок Б.8 – Окно «Выравнивание последовательностей»

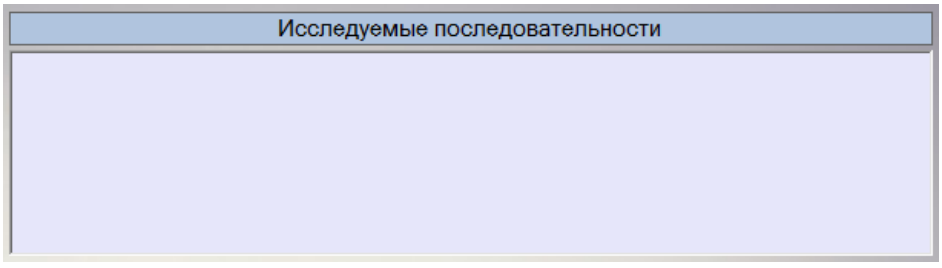


Рисунок Б.9 – TextBox просмотра исследуемых последовательностей

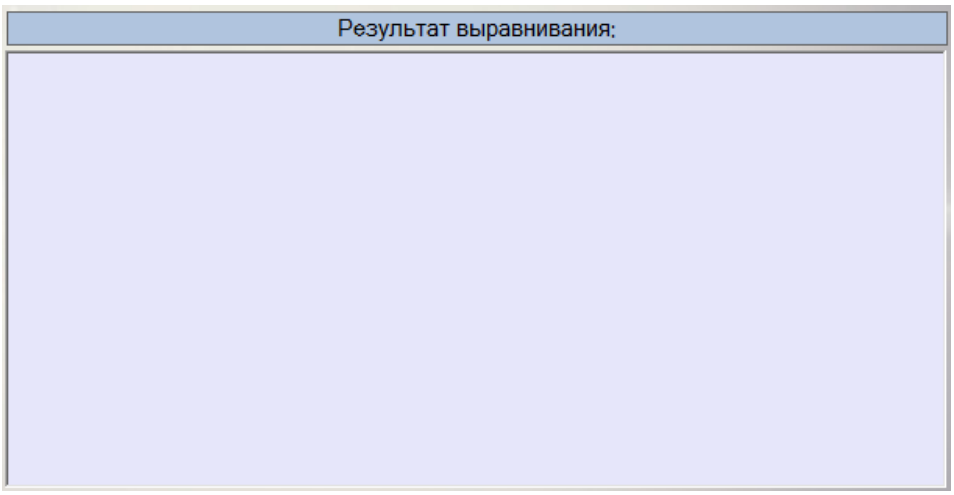


Рисунок Б.10 – TextBox просмотра результатов выравнивания

Рисунок Б.11 – Окно «База белков»

Б.4.3 Задача «Выравнивание белков».

Выполнение данной задачи возможно при наличии входных данных:

1) текстового файла с последовательностью исследуемого белка;

2) подготовленного файла PDB.txt с базой белков.

Основные действия:

1) нажать на кнопку «Выравнивание белков», откроется одноимённое окно (рисунок Б.12);

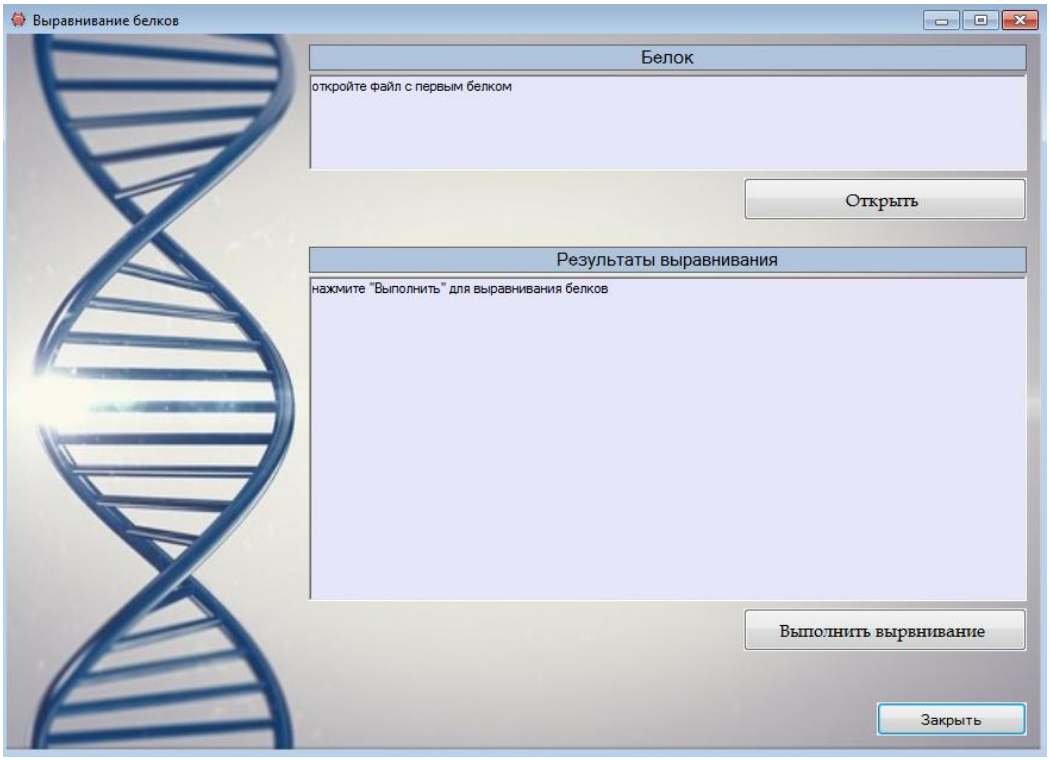


Рисунок Б.12 – Окно «Выравнивание белков»

2) нажать кнопку «Открыть», откроется диалоговое окно;

3) выбрать файл с исследуемым белком, в соответствующий TextBox выводится считанная последовательность (рисунок Б.13);

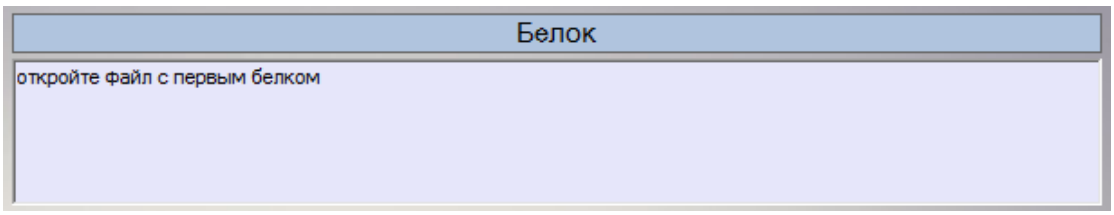


Рисунок Б.13 – TextBox просмотра считанной последовательности

4) нажать кнопку «Выполнить выравнивание», в соответствующий TextBox выводятся результаты выравнивания (рисунок Б.14).

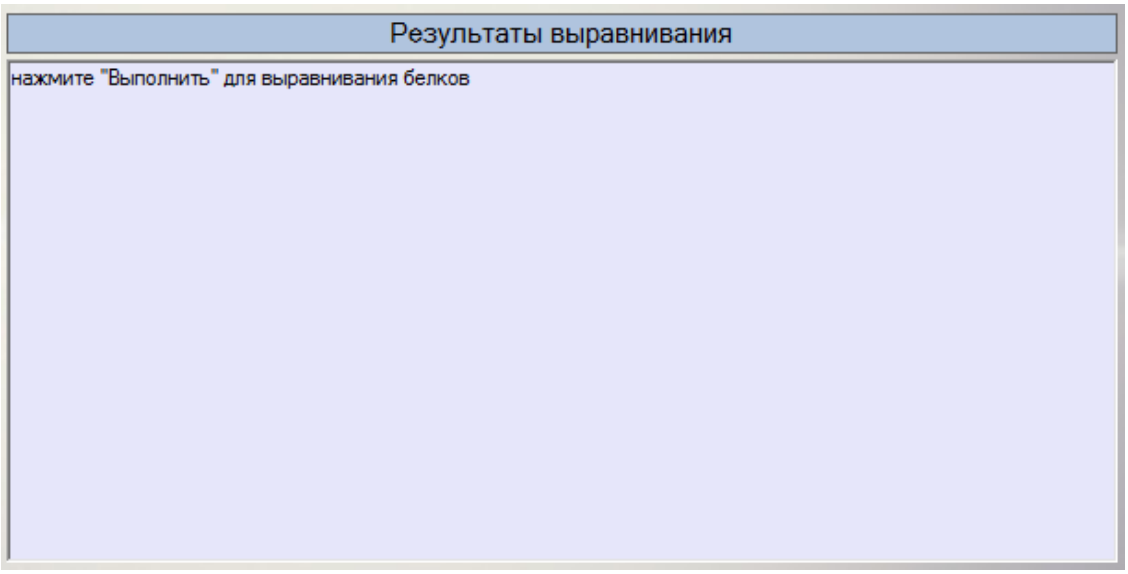


Рисунок Б.14 – TextBox просмотра результатов выравнивания

Б.4.4 Другие задачи.

Б.4.4.1 Просмотр матриц весов и матриц переходов.

При необходимости просмотра построенных в ходе выравнивания матриц весов и переходов необходимо перейти в папку с проектом. Матрицы записываются следующим образом:

− в файл WM.txt матрицы весов, полученные в ходе поиска ДНК по базе белков;

− в файл TM.txt матрицы переходов, полученные в ходе поиска ДНК по базе белков;

− в файл WM1.txt матрицы весов, полученные в ходе выравнивания белков;

− в файл TM1.txt матрицы переходов, полученные в ходе выравнивания белков.

Перед каждым следующим осуществлением выравнивания ранее построенные матрицы удаляются.

Б.4.4.2 Просмотр информации о системе.

Для ознакомления с основными свойствами системы необходимо нажать на кнопку «О программе» на главном окне системы поиска.

Откроется окно «Информация», имеющее три основных раздела: «О системе», «Входные данные», «Результаты» (рисунок Б.15).

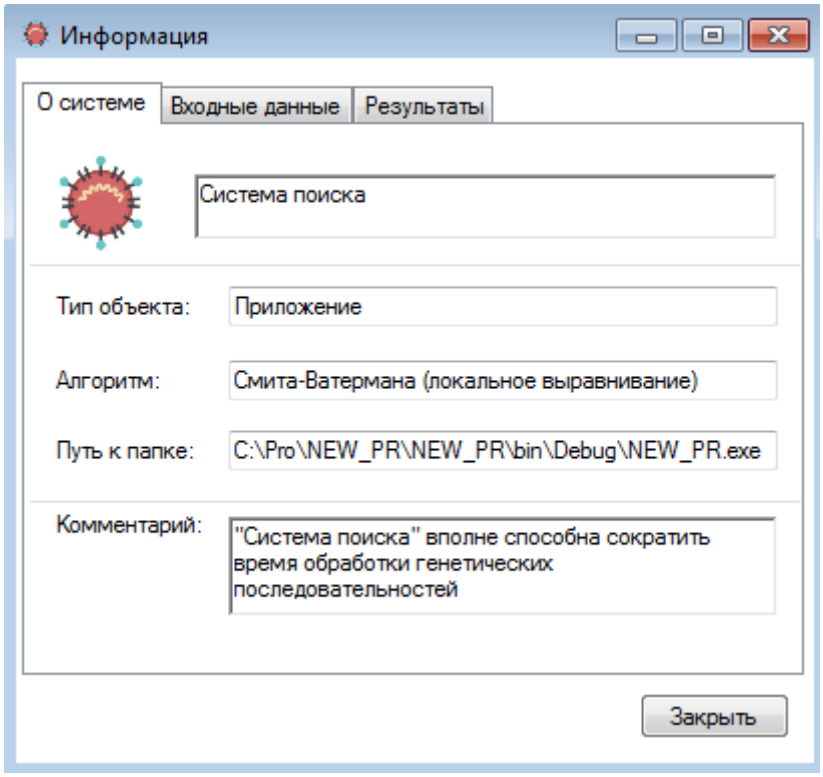


Рисунок Б.15 – Окно «Информация»

Б.5 Рекомендации по освоению

Для успешного освоения системы поиска необходимо иметь навыки работы с ПК и изучить данное «Руководство пользователя».

Приложение В

(справочное)

Руководство программиста

В.1 Назначение и условия применения

Система поиска предназначена для сокращения временных затрат на выполнение поиска оптимального выравнивания биологических последовательностей. Система осуществляет поиск максимальной общей подстроки посредством алгоритма локального выравнивания Смита-Ватермана.

Минимальные программные и аппаратные требования, необходимые для корректной работы системы поиска приведены в таблице В.1.

Таблица В.1 – Минимальные программные и аппаратные требования к системе

Критерий	Требование
Операционная система	Windows XP и выше
ОЗУ	1 Гб
Свободное место на диске	1 Гб
Периферия	Стандартный монитор разрешением 1024x768 и выше; мышь.

В.2 Характеристики программы

Система поиска должна обеспечивать возможность выполнения следующих видов операций:

− перевод последовательности ДНК в РНК;

− удаление некодирующих участков из последовательности РНК;

− приведение последовательности к аминокислотному виду;

− поиск последовательности ДНК по базе данных белков;

− выравнивание белковых последовательностей.

В.3 Обращение к программе

Для корректной работы системы необходимо установить на ПК программу Microsoft Visual Studio Professional 2013.

Для того, чтобы начать работу с системой необходимо запустить исполняемый файл NEW_PR.exe, расположенный в папке проекта.

В.4 Входные и выходные данные

Для корректной работы системы поиска необходимы следующие входные данные:

− последовательность исследуемой ДНК, записанная в текстовый файл;

− последовательность исследуемого белка, записанный в текстовый файл;

− база белков, хранящаяся в файле PDB.txt и имеющая формат записи представленный на рисунке В.1.

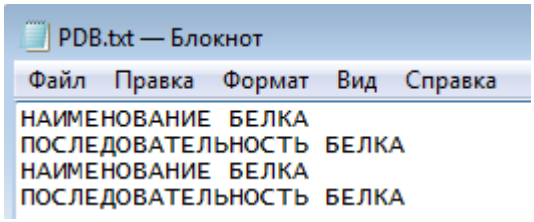


Рисунок В.1 – Формат записи базы белков

Выходные данные для каждой пары последовательностей:

− матрица переходов;

− матрица весов;

− степень сходства последовательностей;

− наибольшая общая подстрока;

− стартовая позиция наибольшей общей подстроки.

B.5 Сообщения

Подготовка последовательности ДНК к выравниванию должна осуществляться в строго определённом порядке:

1) чтение последовательности ДНК из файла;

2) перевод ДНК в РНК;

3) удаление некодирующих участков из РНК;

4) приведение последовательности к аминокислотному виду.

Если пользователь пропускает выполнение какого-либо этапа подготовки, система выводит сообщение (рисунок B.2).

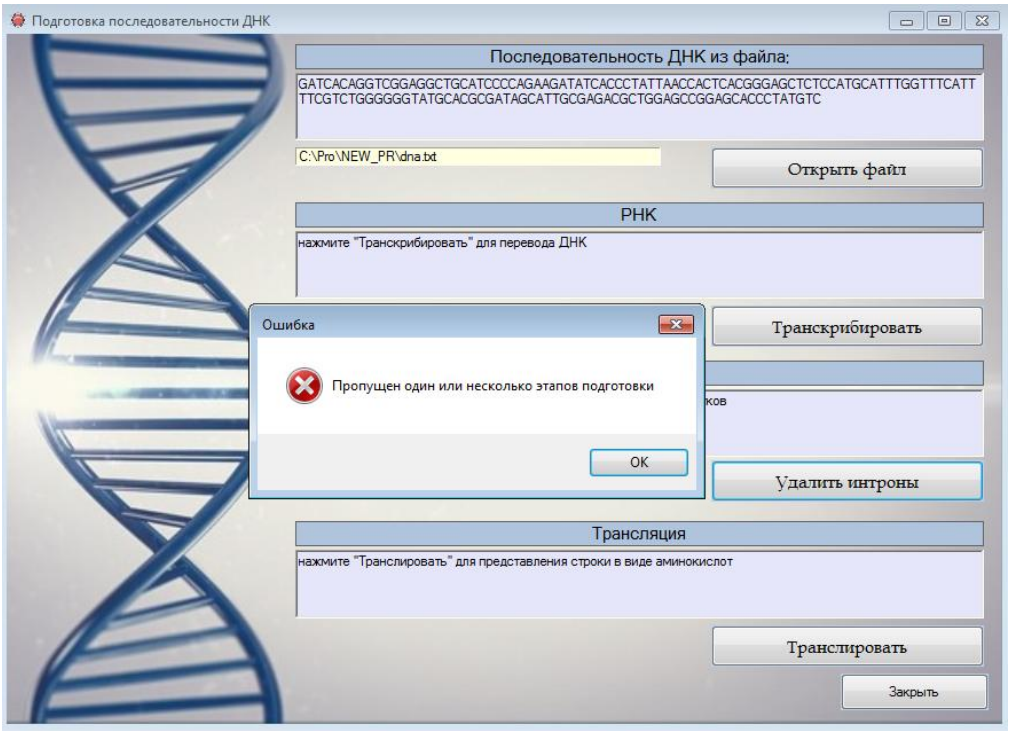


Рисунок B.2 – Сообщение о пропуске этапа подготовки

При удачном выполнении всех этапов подготовки, система выводит сообщение об этом и предлагает перейти к выравниванию (рисунок В.3).

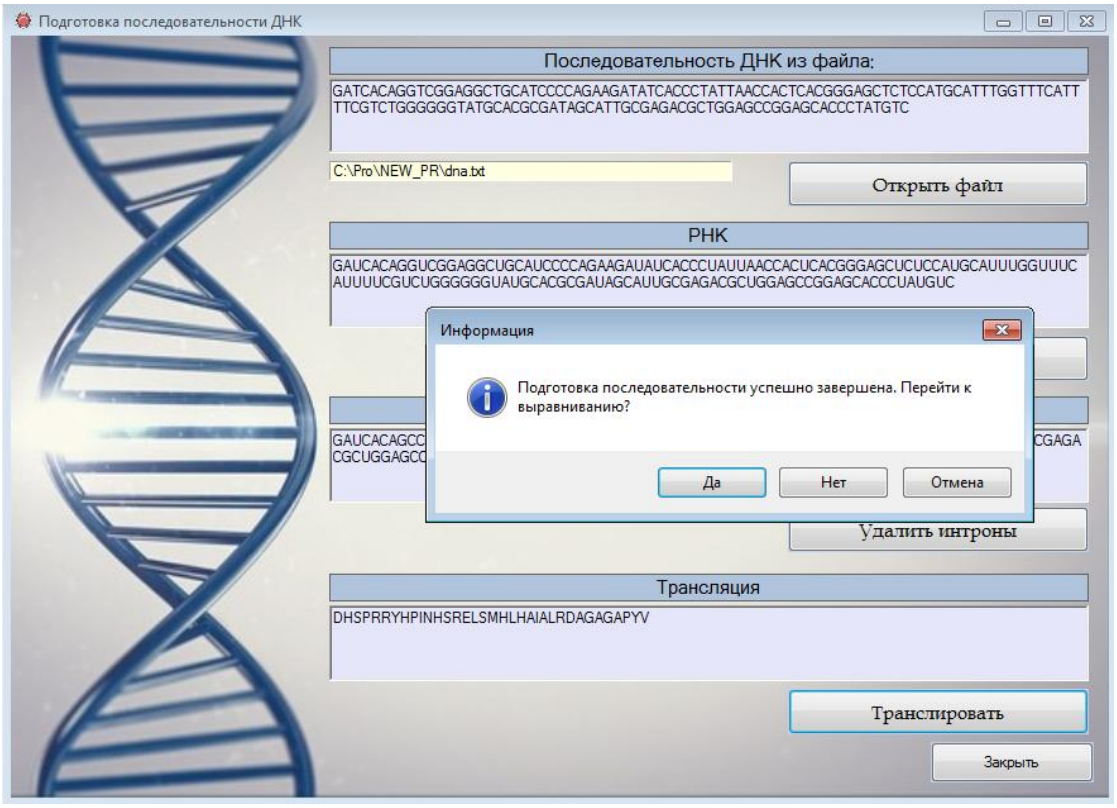


Рисунок В.3 – Сообщение об удачном завершении подготовки ДНК

Если при выравнивании белков пользователь системы нажимает на кнопку «Выполнить выравнивание» прежде, чем открыть файл с исследуемым белком, система выводит сообщение-ошибку (рисунок В.4).

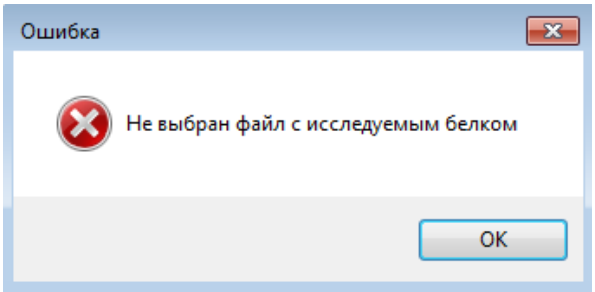


Рисунок В.4 – Сообщение-ошибка

Если файл с исследуемой последовательностью белка был выбран, перед тем как выполнить выравнивание система выводит сообщение о необходимости выбрать базу белков (рисунок В.5).

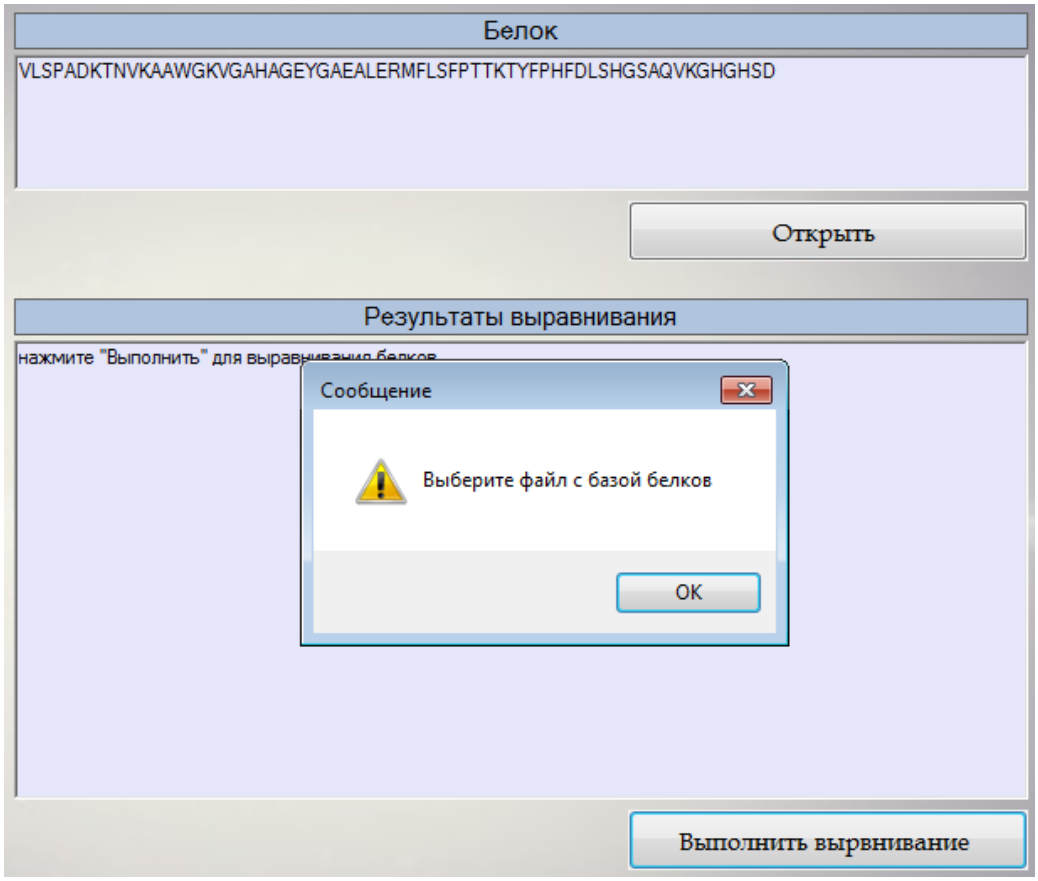


Рисунок В.5 – Сообщение о необходимости выбрать файл с базой белков