The Outercurve Foundation

.NET Bio

документацијски зборник

2015

Бања Лука

.NET Bio:
документацијски зборник

Превео и приредио:
Димитрије Д. Чвокић

Технички уредник:
Димитрије Д. Чвокић

Прелом текста и обрада слика:
Димитрије Д. Чвокић

Издавач:
Ризница, Бања Лука

За издавача:
Предраг Адамовић

Е-издање

Предговор

Књига представља компилацију водича за коришћење апликација и програмских пакета који су дио .NET Bio пројекта. Материјал у књизи се односи на дио градива који студенти СП Биологија у Бањој Луци слушају на првој години из предмета Примјена рачунара у биологији.

.NET Bio представља *open source* пројекат, а и биоинформатички радни оквир, првенствено намијењен за истраживања у области геномике, тј. за ДНК и РНК секвенцирање. *.NET Bio Comparative Assembly*, као дио .NET Bio пројекта, представља оруђе које омогућује дјелотворно састављање великих и сложених генома за које је већ познат сродан, односно сличан геном. *.NET Bio Sequence Assembler* демонстрира могућности *.NET Bio Framework*-а када је у питању развој сложених и „богатих" апликација за биоинформатичка истраживања. Користи различите елементе корисничког прочеља да би омогућио пластичан приказ, а и елегантану обраду геномских података. *.NET Bio Extension for Excel* омогућује рад са геномским секвенцама, метаподацима и интервалним подацима унутар *Excel*-а, што представља посебну погодност имајући у виду да је Excel најраспрострањенији софтвер за табеларну обраду података. Штавише, *Extension for Excel* се може додатно проширити ради искоришћавања осталих могућности *.NET Bio Framework*-а. У дијелу *Технички водич кроз .NET Bio Framework Parallel De Novo Assembler* је описана програмска класа ParallelDeNovoAssembler (Padena), која представља програмску реализацију de novo секвенцирања, заснованог на de Brujin-овим графовима. На крају, представљен је примјер BioDemo.py који демонстрира комбиновање .NET Bio Framework-а са програмским језиком IronPython.

Димитрије Д. Чвокић

Садржај

Преглед .NET Bio Framework-а

Верзија 1.01 – новембар 2011

Сажетак

.NET Bio Framework је .NET библиотека с доступним кодом, намјењена за континуирану употребу, као и апликацијско програмско прочеље (API) испројектовано за истраживања у области биоинформатике.

У овом документу је дат преглед .NET Bio Framework-а, његових компоненти, и неколико примјера употребе.

Увод

У овом документу је дат преглед .NET Bio Framework-а – .NET библиотеке са јавно доступним кодом, намјењене за континуирану употребу, као и апликацијског програмског прочеља (API) за биоинформатичка истраживања. Концептуално, .NET Bio Framework омогућује надоградњу, прилагођавање, и вишекратну употребу. Штавише, развој .NET Bio Framework-а зависи управо од доприноса програмерске (и биоинформатичке) заједнице кроз Open Source Initiative (OSI).

Шта је то .NET Bio Framework?

.NET Bio Framework је, у суштини, биоинформатичко оруђе изграђено на бази Microsoft .NET Framework-а 4.0, чиме је омогућено сопствено надограђивање, тј. конструкција неких других биоинформатичких оруђа. Посебно је пројектован да омогући рад и руковање са великим скуповима података помоћу скалабилних алгоритама, који користе предности вишејезгарне организације рачунара, обезбјеђујући самим тим широк дијапазон биолошких анализа као на примјер:

- парсере/форматере за читање/образовање датотека стандардних биоинформатичих формата
- подршку за рад са ДНК, РНК и протеинским секвенцама
- алгоритамски радни оквир за анализу и трансформације
- додатни веб-оријентисани радни оквир који омогућује садејство са веб-сервисима.

Сам .NET Bio Framework је првенствено намијењен рјешавању проблема из геномике код којих је потребна:

- континуирана употреба одговарајућих структура података за представљање генетских секвенци и симбола
- У/И радни оквир за учитавање и снимање секвенци
- алгоритамск радни оквир за обраду учитаних секвенци.

Open Source пројекат отворен за програмерску заједницу

Један од основних циљева је да се за пројекат што више заинтересује биоинформатичка заједница, а тиме би се створили бољи услови за разумијевање разних техничких проблема као што су рачунарско моделовање, проширивост, развој софтвера, и многи други. Због свега тога, .NET Bio Framework је доступан под open source дозволом са два нивоа учествовања, описана у одјељку Учествовање у развоју Framework-а. Извршне датотеке, изворни код, демо апликације, и документација могу се бесплатно преузети путем Интернета са адресе http://bio.codeplex.com/:

- изаберите downloads картицу да бисте инсталирали извршне датотеке
- изаберите Source Code картицу за изворни код
- одговори на најчешће постављана питања, корисне хипервезе, и примери разних апликација су доступни већ на почетној страни.

Молимо вас да повратну информацију о .NET Bio Framework-у оставите на дискусионој групи http://bio.codeplex.com/discussions.

Имајући поменуто у виду, развој пројекта је био заснован на сљедећим захтјевима:

Захтјеви пројекта

Захтјев	Опис
Проширивост	Проширивост је саставни дио пројекта. Кључне двије ствари – прочеља и (технички) алфабети, омогућују елегантан развој алтернативних имплементација, или пак проширивање функционалности саме апликације.
Неутралност језика	Пројекат је изграђен у оквиру .NET Framework-а, што омогућује коришћење било ког језика који .NET Framework подржава, укључујући и динамички типизование језике попут IronPython-а.
Најбоља искуства	Најбоља досадашња искуства се прате и примијењују током читавог његовог развоја. Сам изворни код је добро документован, искоментарисан, а за коришћене алгоритме су наведене, као референце, и одговарајуће научно-стручне публикације.
Интероперабилност	bio.Silverlight библиотека омогућује покретање апликација под Silverlight-ом, којег подржава неколико најпопуларнијих ОС. Нпр. за Линукс ОС постоји Mono имплементација Silverlight-а, која се зове Moonlight, за коју је, штавише, доступан и изворни код.

Почетак рада са .NET Bio Framework-ом

.NET Bio Framework је доступан као open source пројекат. Извршне датотеке, иворни код, демо апликације, као и сва документација, могу се у потпуности бесплатно преузети.

За почетак, да би сте уопште могли да користите могућности .NET Bio Framework-а, требате преузети/покренути најновије верзије потребних инсталатера (укупно три). Успут, било би пожељно да преузмете и одговарајућу документацију са сајта CodePlex-а.

Framework

.NET Bio Framework је, биоинформатички гледано, језички неутралан инструментаријум, изграђен на темељу Microsoft®-овог .NET Framework-а. У свом саставу садржи парсере за најпопуларније биоинформатичке формате датотека, алгоритме за рад и руковање са ДНК, РНК, и протеинским секвенцама, као и скуп одговарајућих спојница на биолошке веб-сервисе као што је то нпр. NCBI BLAST.

Страница за преузимање .NET Bio Framework-а је

http://bio.codeplex.com/releases.

.NET Bio Sequence Assembler

.NET Bio Sequence Assembler представља, с концептуалне стране, једну лијепу демонстративну апликацију која користи бројне могућности .NET Bio dll-а, .NET Framework-а, и Windows® Presentation Foundation-а. Користећи богатство елемената корисничког прочеља (UI), .NET Bio Sequence Assembler на елегантан начин омогућује визуализацију и рад са геномским подацима. Страница за преузимање .NET Bio Sequence Assembler-а је http://bio.codeplex.com/releases.

Документацији .NET Bio Sequence Assembler-а можете приступити пратећи хипервезу, кликом на навигационо дугме **Documentation,** на страници http://bio.codeplex.com/.

.NET Bio Extension for Excel

.NET Bio Extension for Excel је додатак Microsoft Office Excel-у 2007 и Excel-у 2010, која омогућује једноставан и флексибилан начин рада са геномским секвенцама, мета-подацима и интервалним подацима у Excel-овом документу. .NET Bio Biology Extension add-in је у себе инкорпорирао неколико битних елемената .NET Bio Framework-а: парсере за најпопуларније формате геномских датотека; секвенционе алгоритме за формирање секвенце усаглашености за ДНК ланац; и спојнице на неколико Basic Local Alignment Search Tool (BLAST) веб-сервиса за геномску идентификацију.

Страница за преузимање .NET Bio Extension for Excel-а је
http://bio.codeplex.com/releases

До странице за преузимање документације .NET Bio Extension for Excel-а можете доћи путем хипервезе **Documentation** (у облику навигационог дугмета), на http://bio.codeplex.com/.

Програмирање и Framework

Сам Framework омогућује његово додатно проширивање за разне потребе. На примјер, ако вам затребају програмске функције којих нема у стандардној библиотеци, можете их сами испрограмирати, при чему ћете успут примијетити да их је веома лако додати већ постојећим функцијама Framework-а. Штавише, The Outercurve Foundation охрабрује програмере који проширују Framework и да омогуће приступ свом коду и другим програмерима, како би истраживачка и академска заједница, као једна велика цјелина, могле имати конкретне користи од њиховог рада.

За приступ изворном коду Framework-а, иницирању пројекта, или пак за више информација о писању самог кода, погледајте:

- .NET Bio Programming Guide: до ког долазите путем хипервезе на навигационом дугмету **Documentation,** на страници http://bio.codeplex.com/.

Уколико сте заинтересовани да својим кодом допринесете .NET Bio Framework пројектима, погледајте:

- .NET Bio Code Contribution Guide: до ког долазите путем хипервезе на навигационом дугмету **Documentation,** на страници http://bio.codeplex.com/

- .NET Bio C# Coding Standards: до ког долазите путем хипервезе на навигационом дугмету **Documentation,** на страници http://bio.codeplex.com/

.NET Bio Commenting Conventions: до ког долазите путем хипервезе на навигационом дугмету **Documentation,** на страници http://bio.codeplex.com/.

Упознавање са .NET Bio ресурсима

.NET Bio Framework је историјски произашао из Microsoft Biology Foundation-а (MBF) и Microsoft Biology Tools-а (MBT).

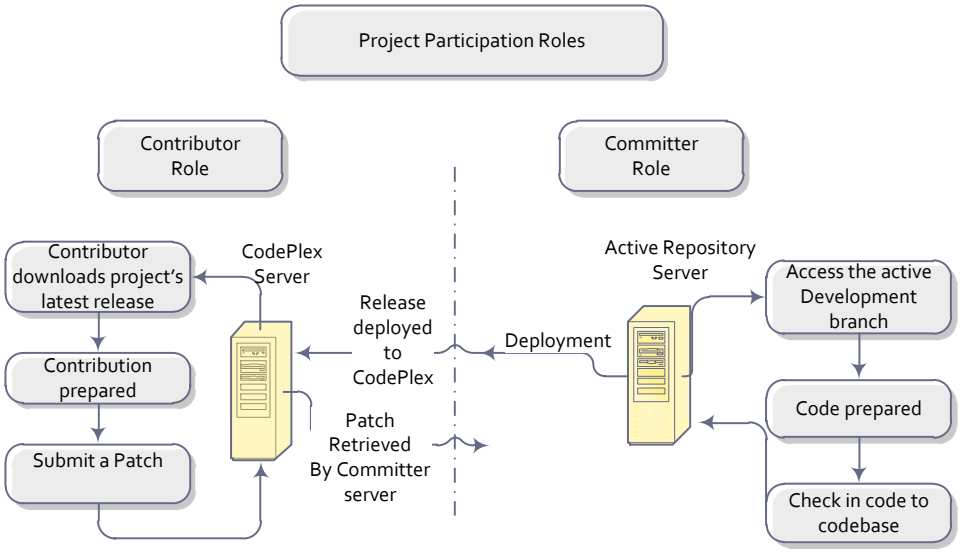
- главни веб-сајт: .NET Bio на Codeplex-у, http://bio.codeplex.com/, одакле можете преузети стабилне верзије Framework-а, неке битне примјере, изворни код, документацију, а и пратити форумске дискусије.
- Microsoft Biology Tools http://research.microsoft.com/bio/mbt.aspx, који представља збирку истраживачких биоинформатичких оруђа.
- MBF/.NET Bio Training, са сајта http://bio.codeplex.com, под training menu-ом, одакле можете преузети материјале за учење, који укључују и практичне лабораторијске вјежбе, а које вас уводе у тајне кодирања/програмирања у оквиру Framework-а.

Учествовање у развоју Framework-а

Постоје два начина како да учествујете у пројекту:

- као сарадник (енг. Contributor) – преузимањем кодне базе са Codeplex-а и дистрибуирањем својих достигнућа/доприноса кроз CodePlex мрежу; да би могли дистрибуирати своја достигнућа/доприносе требате отворити налог на CodePlex-у.
- као извршилац (енг. Committer) – непосредно приступајући кодној бази у репозиторијуму и провјеравајући сопствене измјене репозиторијума; наравно, морате имати одговарајућа извршна права.

Следећа слика илуструје двије поменуте улоге.



Док сарадници имају приступ само периодично постављеном коду и могу само предлагати измјене и допуне кода искључиво користећи Codeplex-ову **Upload a patch** опцију, извршиоци имају непосредан приступ кодној бази – и за check-in и за check-out кода.

Детаљније о овим улогама можете пронаћи у Contributor Guide-у и Committers Guide- у на http://bio.codeplex.com/documentation.

Сврха .NET Bio Framework-а

Биоинфоратика је још увијек релативно младо научно поље. Сам термин „биоинформатика“ датира из средине 1990-их, а његови конструктори и пропагатори су били људи са различитим научним позадинама: биологија, физика, хемија, а повремено и информатика.

Нажалост, могуће је да је то и разлог зашто сама биоинформатичка заједница није досљедна у коришћењу индустријских стандарда и већ поприлично нагомиланог искустава у програмирању, пројектовању и развоју софтвера. Формати података још увијек нису добро дефинисани, иако већ постоји велики број, што је најгоре, новоразвијених шема (које додуше покривају само неке од потребних ствари). Повећањем количине биоинформатичких података скалабилност постаје озбиљан проблем. Многе, чак поједностављене, софтверске имплементације нису у могућности обављати своје задатке на једнопроцесорским системима. Што се тиче искористивости вишејезгарних и вишепроцесорских рачунарских архитектура, још увијек нису ни смијернице развоја дефинисане.

Вишекратно искористиве библиотеке

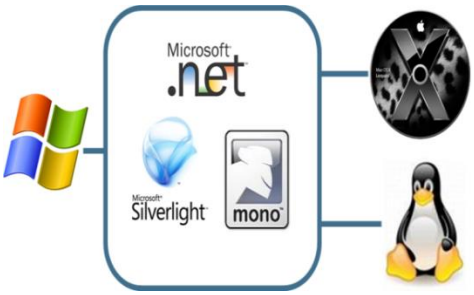
Претходно наведени проблеми су наметнули потребу за континуирано искористивим библиотекама са висококвалитетним биоинформатичким кодом. Циљ .NET Bio Framework-а је да обезбиједи развојни оквир који ће бити од користи читавој биоинформатичкој заједници и који ће осигурати високе пројектантске и кодне стандарде, неопходне за обезбјеђивање проширивости апликације и њену дуготрајност.

Коришћење било које апликацијске форме

Могуће је развијати конзолске апликације, NT-сервисе, графичка корисничка прочеља (GUI) користећи WIN Forms, динамичне и интерактивне ASP .NET веб-странице, апликације у cloud-у преко Azure cloud рачунарске платформе, и сервисне апликације које су веб-оријентисане користећи Silverlight, управо како је то и приказано на сљедећој слици.

Развој на различитим платформама

За коришћење Framework-а у другим окружењима препорука је Silverlight као апликацијска форма, а онда за обраду кода искористите Mono/Moonlight, или пак Silverlight plug-in на OS X платформи. За IDE на Windows платформи можете користити Visual Studio, или ако радите са изворним кодом можете користити и Mono-заснован IDE као што је MonoDevelop или пак SharpDevelop.



Напомена: Mono је open source имплементација Microsoft-овог .NET Framework-а која се извршава на не-Windows оперативним системима. Silverlight је додатак за прегледник који користи подскуп .NET-а, а који подржава већина популарних прегледника, укључујући Internet Explorer, Chrome, Firefox и Safari. Moonlight је изграђен над Mono-ом, као основом.

Широк спектар употребе

Framework је могуће користити за обављање веома широког спектра задатака, укључујући:

- састављање генома
- убацивање ДНК, РНК, или протеинских секвенцији из датотека, укључујући Fasta, FASTQ, GFF, и GenBank формате.
- формирање секвенци
- манипулисање секвенцама, као што је издвајање сегмената, генерисање одговарајућих допуна, или пак обртања саме секвенце
- анализу секвенце користећи алгоритме као што су Smith-Waterman и Needleman-Wunsch
- просљеђивање података о секвенци удаљеним веб-сервисима на анализу (као што је Basic Local Alignment Search Tool (BLAST))
- испис података о секвенци на било ком од подржаних формата, без обзира на улазни формат.

Реализација на било ком .NET компатибилном језику

.NET Bio Framework апликације могу бити кодиране/развијане на било ком од преко 70 .NET компатибилних језика, укључујући C#, F#, Visual Basic® .NET, и IronPython. Одговарајуће програмске водиче можете наћи на http://bio.codeplex.com/documentation, гдје је описан развој .NET Bio Framework апликације користећи C# и IronPython.

Шта је додато, а шта је само измијењено?

Коријени .NET Bio Framework су Microsoft Biology Foundation (MBF) и Microsoft Biology Tool (MBT). Следеће карактеристике и оруђа су инкорпорирани, одстрањени, или пак промијењени на путу од MBF до .NET Bio Framework-а.

Листа промјена

Табела садржи сажетак листе промјена за ово издање .NET Framework-а. Више детаља, као и комплетнију листу API промјена можете погледати у Programming Guide-у.

Листа промјена за .NET Bio Framework 1.0

Промјена	Опис
AzureBlast	Одстрањен.

Промјена	Опис
Bio.Silverlight	Додато. Bio.Silverlight је .dll имплементација Silverlight-функционалности за .NET Bio и омогућује развој апликација на различитим платформама користећи управо Silverlight методологију.
Comparative Assembly	Нове технике састављања геномских секвенци, које омогућују ресеквенцирање и упоредно састављање генома, када се то односи на исте или сличне врсте.
ComparativeUtil	**Ново** - ComparativeUtil покреће процес састављања геномских секвенци заснован на сличности са неки претходно задатим референтни геном.
Padena assembly algorithm	Повећан капацитет за састављање већих и сложенијих генома. Побољшане перформансе генерисања De Bruijn-ових графова.
PadenaUtil	Оруђе које са командне линије омогућава de novo састављање геномских секвенци.
Source tree changes	MBF\Source\MBF -> Bio\Source\Framework MBF\Source\MBF - > Bio\Source\Tools
Wiggle format support	Wiggle формат је геномски формат датотека, пројектован за прика густих и континуираних података као што су GC постотак, вриједности разних вјероватноћа, и транскриптомски подаци. За више информација пратите хипервезу.
ConsensusUtil	**Ново** – користи се за ComparativeUtil корак 4. Корисници могу манипулисати подацима прије њихове употребе у следећем кораку.
LayoutRefinementUtil	**Ново** – користи се за ComparativeUtil корак 3. Корисници могу манипулисати подацима прије њиховог уношења у следећем кораку.
LISUtil	**Ново** – услужно оруђе за најдужи растући низ mummer-а.
MUMmer	Оптимизација при сложенијим састављањима генома.
New License	Читава библиотека је пресељена из MS-PL у шире прихваћену дозволу Apache 2.0 OSI.
New namespace	**MBF** преименован у **Bio**.
NucmerUtil	**Ново** – користи се током ComparativeUtil корака 1. Корисници могу манипулисати подацима прије њихове употребе у следећем кораку.
Optimization work	a) Профилација меморије; оптимизована анализа Framework-а. b) Padena меморијска оптимизација. c) Оптимизације секвенци, укључујући non-string и non-character секвенце. d) MUMmer оптимизација заснована на суфиксном стаблу; побољшани линкови. e) Оптимизација Object Model-а. f) Више сценарија за профилацију меморије и перформанси.
Parser and formatter Encoding	**Одстрањено** – парсери и форматери више не примају кодирања. Одстрањена је читава кодна класа.

Промјена	Опис
RepeatResolutionUtil	**Ново** – користи се за ComparativeUtil корак 2. Корисници могу манипулисати подацима прије њихове употребе у следећем кораку.
SAMUtils	Оруђе за анализу покривености секвенце.
ScaffoldUtil	**Ново** – користи се за ComparativeUtil корак 5. Корисници могу манипулисати подацима прије њихове употребе у следећем кораку.
Sequence Object Model	a) Реконструисан ради бољег заузимања меморије. b) Употреба густе бинарне представе ДНК, РНК и протеинских секвенци, умјесто знаковних ниски. c) Повећан капацитет кориштењем **IEnumerable<byte>**-а. **Dictionary** и **hashset** се користе за похрану ставки као што су вишезначни алфабети. d) Кодирање се учинковитије реализује. e) Кодирање одстрањено из објектног модела секвенци (парсери и форматери). f) Измјене у употреби **ISequence** : **IList<byte>.**
Data Virtualization	**Одстрањено**

Инсталација .NET Bio Framework-a

У овом дијелу су описани системски захтјеви и кораци приликом инсталације .NET Bio Framework-a.

Предуслови

За потпуно искоришћавање капацитета Framework-a, морате имати основно знање о:

- геномичким и биоинформатичким методама и номенклатури
- раду са софтвером као што је Microsoft Office Excel.

Да би могли искористите могућности програмирања и проширивања, потребне су вам бар:

- основна знања из рачунарског програмирања
- познавање коришћења Microsoft Visual Studio®-а за развој .NET апликација у C#
- концептуално разумијевање веб-технологија.

Системски захтјеви

- Windows® XP Service Pack (SP) 3 и касније верзије Windows-а

- .NET Framework Version 4.0

Додатни софтверски захтјеви за развој и имплеметнацију .NET Bio Framework апликација су описани у „Programming Guide-у“.

Инсталација

У оквиру .NET Bio Framework пројеката повремено се на Codeplex-у објављују стабилни прикази стабла изворног кода, тачније на веб-страници http://bio.codeplex.com/. Текућу верзију стабла изворног кода можете добити просто преузимањем приказа .

Ако сте заинтересовани за .NET Bio Framework, али не желите отпремити ваш код у репозиторијум, само покрените .NET Bio Framework инсталатор – Bio.msi, и изаберите опцију **Complete**, како би инсталирали одговарајући софтверски инструментаријум (SDK). Овом опцијом се инсталира све што вам треба за развој .NET Bio Framework апликација, укључујући и све .NET Bio DLL-ове, под $\Program Files\.NET Bio директоријумом.

Успут, региструјте се на CodePlex-у да бисте користили било коју од датих опција. Не требају вам статуси извршиоца, нити сарадника, јер је ово све доступно за преузимање било ком заинтересованом кориснику.

Више података, као и сам инсталер – .NET Bio.msi, доступни су путем хипервезе http://bio.codeplex.com/releases.

Инсталација .NET Bio Framework-а

1. Преузети .NET Bioinstaller и .NET Bio.msi путем хипервезе http://bio.codeplex.com/releases; смјестити под директоријум на магнетном диску (могуће је, такође, преузети и покренути са Codeplex-овог сајта).
2. У прозору датог директоријума двокликом покрените **.NETBio.msi**, који покреће чаробњака за инсталацију.
3. Слиједите упуства чаробњака да би инсталирали .NET Bio.

   Напомена: Изаберите **Complete** инсталацију ако желите изворни код и бинарне датотеке, што ће укључити и инструментаријум.

За комплетну инсталацију the .NET Bio Framework-а инсталатор формира директоријум под називом C:\Program Files (x86)\.NET Bio\1.0\SDK са следећим садржајем:

\Docs

Bio.chm

Coding_Conventions.docx

Commenting_Conventions.docx

Comparative Assembly Technical Guide

Committer_Guide

Contribution_Documentation_Template.docx

Contribution_Guide.docx

Getting_Started.docx

Becoming_A_Committer

Onboarding.docx

Overview.docx

PaDeNa Technical Users Guide.docx

Programming_Guide.docx

IronPython_Programming Guide.docx

Testing_Guide.docx

Напомена: како би вам се API документација приказивала у Intellisense искачућим окнима, морате направити пројекат за формирање XML документацијске датотеке, а онда провјерити да ли се датотека налази под истим директоријумом гдје је и Bio.dll инсталиран.

\Framework

\Add-ins

\Bio.Comparative.dll

\Bio.Padena.dll

\Bio.Pamsam.dll

\Bio

\Bio.Hpc

\Bio.Hpc.distrubuteApp

\Bio.Silverlight

\ Bio.WebServiceHandlers

\Tools

\Bedstats

\ComparativeUtil

\ConsensusUtil

\Fileformatconverter

\FilterReadsutil

\IronPython

\LayoutRefinementUtil

\LISUtil

\MumUtil

\NucmerUtil

\PadenaUtil

\ReadSimulator

\RepeatResolutionUtil
\SampleClusterApp
\SAMUtils
\ScaffoldUtil
\Tools.VennTo.NodeXL
\TridentWorkflows
\VennTool
Readme.txt

Прелазак на новије верзије

Они који имају старије верзије .NET Bio Framework-а, не требају их деинсталирати да би инсталирали новију верзију. Конкретно, верзије могу бити инсталирене једна поред друге. Исто је ствар и при преласку са Microsoft Biology Foundation-а (MBF) на .NET Bio Framework.

Старије верзије MBF-а:
v1.0
тренутна .NET Bio Framework верзија
v1.0

Понашање инсталатера

Приликом инсталације новије верзије уз постојање старијих, очекује се да ће се збити:

- тиха надоградња било коју мање промјене верзије
- инсталација поред већ постојеће верзије (након приказивања поруке да постоји старија верзија софтвера и приједлога за њену деинсталацију) за било какве веће промјене.

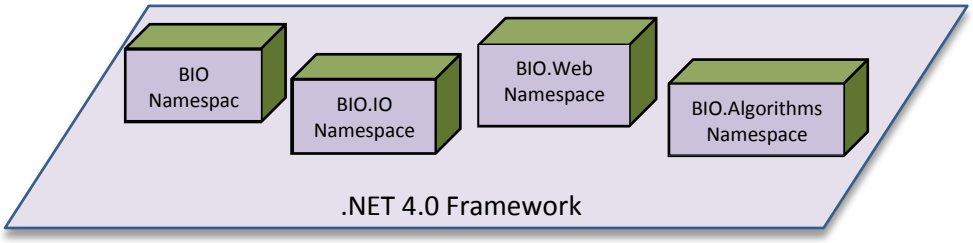
DLL верзионирање

Важи сљедећи договор о dll верзионирању:

- .NET Bio DLL-ови тренутне верзије су 1.0.0.0
- било која DLL верзија унутар пакета нема никакве везе са верзијом свог продукта. На примјер, MyProduct v3.0 може имати DLLове који су v1.0 / v5.0.

Архитектура .NET Bio Framework-a

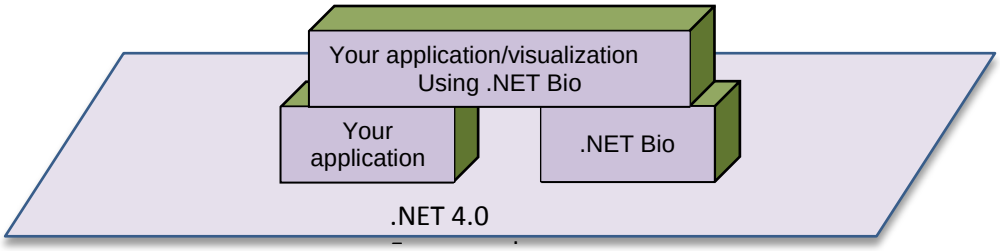
Framework је у основи биоинформатички скуп оруђа изграђен на бази .NET Framework-a 4.0, а који при том омогућује изградњу и развијање других оруђа. Садржи библиотеке за континуирано коришћење биоинформатичких функција и алгоритмима заснованих на .NET Framework-у. Свака библиотека има свој простор назива и одговарајуће програмске класе. Следећа слика приказује простор називе сваке од четири библиотеке.



Простори назива пружају следећим компонентама подршку у процесу биолошке анализе. Погледајте Bio.chm у директоријуму SDK\docs за референцу на API.

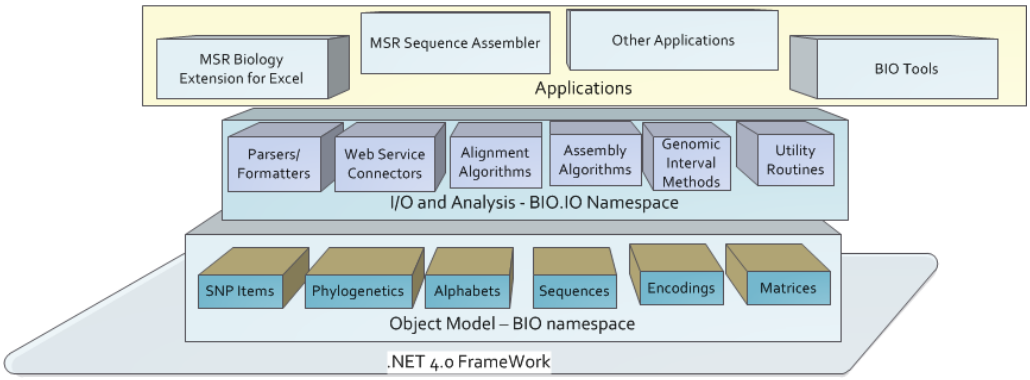
- **BIO:** објектни модел за похрањивање података о секвенцама, метаподатака, и кодираног материјала.
- **BIO.Web**: прочеље за веб-сервисе које омогућује повезивање објектног модела са различитим веб-оријентисаним компонентама. BLAST и ClustalW су подразумијеване имплементације.
- **BIO.Algorithms**: алгоритми за превођење, вишесеквенцно сравњивање по паровима, и састављање секвенци.
- **BIO.IO**: анализатори и форматери за различите типове геномских података.

Framework није апликација сама за себе. Не омогућује визуализацију података, већ пружа основу за пројектовање визуелизације, као што је приказано на следећој слици.



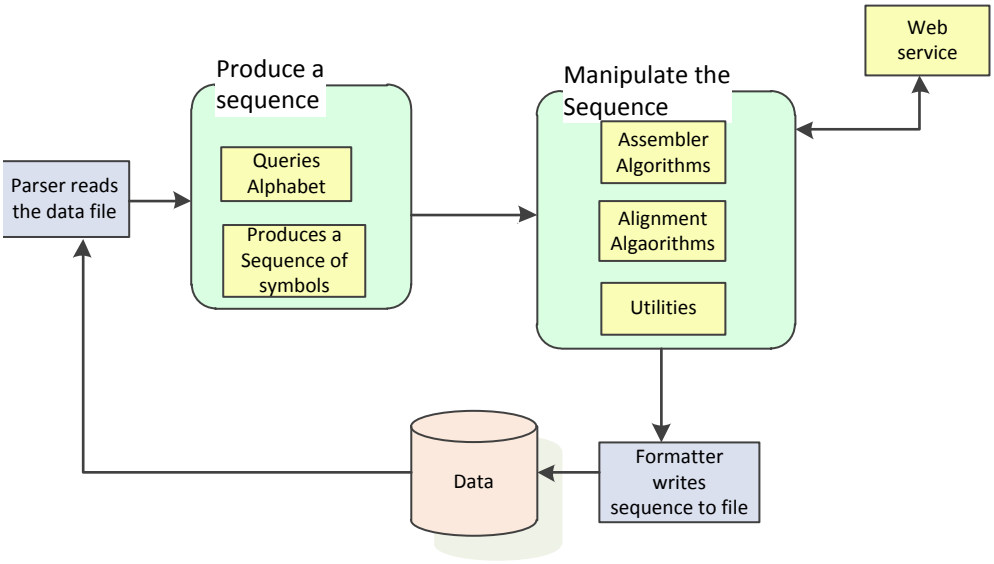
Кориштење .NET Bio Framework-a је једноставно попут додавања референце Bio.dll на ваш пројекат. Тада можете почети користити доступне типове података/објеката. Користите bio.Silverlight.dll за развиој Silverlight апликација.

Секвенце представљају кључни концепт Framework-a. Садрже [симболе] засноване на геномској азбуци, предмет су обраде парсера и форматера, просљеђују се као као аргументи програмским функцијама, а и враћају се као резултујуће вриједности послије обраде. На сљедећој слици је приказана архтектура пројекта.



Напомена: Састављачи и сравњивачи (енг. aligners) су подржани као додаци; програмску класу декоришите атрибутима и додацима.

Сљедећи дијаграм илуструје типичан процес обраде података.



Компоненете .NET Bio Framework-а

Имплементација Framework-а укључује:

- објектни модел за представљање геномских података
- парсере за стандардне биоинформатичке формате датотека
- алгоритаме за манипулисање ДНК, РНК, и протеинским секвенцама
- скуп софтверских спојница на биолошке веб-сервисе као што је нпр. NCBI BLAST.

Такође, могуће је радити са секвенцама користећи више понуђених оруђа у покренутом пројекту: .NET BioExtension за Excel (додатак за Microsoft Excel) и .NET BioSequence Assembler (једна од .NET апликација). За више информација погледајте пратеће документе на http://bio.codeplex.com/documentation или у **..\Bio\Doc** директоријуму документацијског стабла изворног кода.

Оруђа

Следећа два оруђа су доступна за било који започети биоинформатички пројекат:

Алат	Опис
.NET Bio Sequence Assembler	Обезбеђује графичко прочеље (GUI) за састављање секвенци. .NET Bio_Sequence_Assembler_User_Guide.docx
.NET Bio Biology Extension for Excel	Обезбеђује Excel toolbar ленту (изборник) за елегантно искоришћавање функционалности .NET Bio Framework-а. .NET Bio_Biology_Extension_User_Guide.docx

Парсери и форматери

Следећи парсери и форматери су доступни за било који започети пројекат (на ..\Bio\Source\Framework\Bio\IO):

Formats	Parser or Formatter	Description
FastA	Парсер и форматер	Sequence format
FastQ	Парсер и форматер	Sequence format
GenBank	Парсер и форматер	Sequence format
GFF	Парсер и форматер	Sequence format
Newick	Parser and Formatter	Филогенетика
Nexus	Парсер	Сравњивање секвенци
Phylip	Парсер	Филогенетика
SAM and BAM	Парсер и форматер	Sequence alignment
BED	Парсер и форматер	Sequence format
ClustalW	Парсер	Сравњивање секвенци
snpParser and SimplesnpParser	Парсер	
Wiggle	Парсер и форматер	Supports annotations.
XSV related Parser and formatters		
		XsvTextReader XsvSparseReader XsvSparseParser XsvSparseFormatter XsvSnpReader XsvContigParser XsvContigFormatter

Веб-сервиси

Следећи веб-сервиси и њихови (софтверски) руковаоци су доступни за било који започети пројекат:

Web Услуге	Опис
Azure	..\Bio\Source\Framework\Bio.WebServiceHandlers
BioHPC	..\Bio\Source\Framework\Bio.WebServiceHandlers
EBI	..\Bio\Source\Framework\Bio.WebServiceHandlers

Web Услуге	Опис
NCBI	..\Bio\Source\Framework\Bio.WebServiceHandlers
BLAST	Handler Bio.Web.Blast.IBlastServiceHandler at ..\Bio\Source\Framework\Bio\Web.
ClustalW	Handler Bio.Web.ClustalW.IClustalWServiceHandlerat ..\Bio\Source\Framework\Bio\Web.

Уграђени алгоритми за сравњивање

Неколико стандардних алгоритама су одмах доступни кроз своје програмске реализација, за било који започети пројекат, укључујући и алгоритме на

..\Bio\Source\Framework\Bio\Algorithms\Alignment:

Алајнери алгоритама	Опис
PairwiseOverlapAligner	Публикацијска програмска реализација простог алгоритма за сравњивање двије секвенце заснованог за 2-по-2 преклапању.
NeedlemanWunschAligner	Опште сравњивање (упоређује се читава секвенца) засновано на Needleman-Wunsch алгоритму.
SmithWatermanAligner	Локално сравњивање (поређење дијелова секвенци) засновано на Smith-Waterman алгоритму.
MUMmerAligner	Алгоритми за сравњивање читавих генома или веома великих протеинских ланаца. Заузврат позива MUMmer.
NucmerPairwiseAligner	Алгоритми за сравњивање читавих генома или веома великих ДНК ланаца.

Укључени примјери

Пројекат укључује примјере са кодом и одговарајућим датотекама који могу бити од користи почетницима.

Апликација-примјер	Опис
AlignSequences	Налази се у Programming Guide.docx. Демонстрира сравњивање секвенце и употребу **SequenceStatistics** за итерирање кроз саму секвенцу.
Bio.Workflow	Обично је смјештен под директоријумом C:\Program Files (x86)\.NET Bio\1.0\Tools, а изворни код под $/BIO/SourceSamples.

Апликација-примјер	Опис
BioDemo.py	IronPython-демонстарација неких од тренутних не-GUI карактеристика. Обично се налази под директоријумом C:\Program Files (x86)\.NET Bio\1.0\Tools, а изворни код под директоријумом $/BIO/SourceSamples.
BlastRequest	Налази се у Programming Guide.docx-у. Демонстрира коришћење сервиса Blast и **WebRequest**.
GenBank Data file	Примјер датотеке која је укључена у Programming Guide. Демонстрира рад са комадом секвенце, укључујући њено обртање, а и комплементирање.
ManipulateSequence	У Programming Guide.docx-у.
ReadSimulator	Под директоријумом C:\Program Files (x86)\.NET Bio\1.0\.\Tools, а изворни код под директоријумом $/BIO/SourceSamples.

Напомена: документација за сваки од наведених примјера се налази под истим директоријумом под којим је и конкретни примјер.

Такође, постоји и Starter Project и неколико примјера датотека у материјалима за обуку на адреси http://research.microsoft.com/bio, који вас води кроз нови C# пројекат.

Извори

Овај дио документације се односи на хипервезе ка страницама са додатним информацијама о .NET Bio Framework-у.

Microsoft

IronPython
http://www.codeplex.com/IronPython/

Microsoft Biology Foundation at Microsoft Research
http://research.microsoft.com/en-us/collaboration/tools/mbf.aspx

Visual Studio 2010 and .NET Framework 4
http://msdn.microsoft.com/vstudio/

CodePlex

.NET Bio Framework

- http://bio.codeplex.com/
  .NET Bio Framework: Overview
  .NET Bio Framework: Programming Guide
  .NET Bio Sequence Assembler: User Guide
  Padena: Parallel DeNovo Assembler
- Training Workshop Material - http://bio.codeplex.com/wikipage?title=Training&referringTitle=Home&ANCHOR#home

.NET Bio Extension for Excel User's Guide
http://bio.codeplex.com/wikipage?title=bioexcel&referringTitle=sampleapps&ANCHOR#sampleapps
.NET Bio Extension for Excel User Guide

Sandcastle
Sandcastle - Documentation Compiler for Managed Class Libraries
http://sandcastle.codeplex.com/
Sandcastle Help File Builder
http://www.codeplex.com/SHFB

Биоинформатичке референце

BLAST
http://blast.ncbi.nlm.nih.gov/Blast.cgi

EBI BLAST Service
http://www.ebi.ac.uk/Tools/blast2/index.html

FASTA format description
http://www.ncbi.nlm.nih.gov/blast/fasta.shtml

FASTQ format description
http://maq.sourceforge.net/fastq.shtml

GenBank

Overview
http://www.ncbi.nlm.nih.gov/Genbank/
Sample GenBank Record
http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html

GFF Specification

http://www.sanger.ac.uk/resources/software/gff/spec.html

International Nucleotide Sequence Database Collaboration

http://insdc.org/favicon.ico

National Center for Biotechnology Information

http://www.ncbi.nlm.nih.gov

Технички водич кроз .NET Bio Comparative Assembly

Верзија 1.0 јун 2011

Сажетак

Референтно састављање генома је процес у којем се послије очитавања *циљног* генома, користи већ секвенциран сродни геном као *референца* за спајање поклапајућих контига, чиме се значајно добија на времену. Comparative Assembly је веома дјелотворан у формирању великих и сложених генома за које је већ познат сродан, тј. сличан геном.

У овом документу је описана програмска класа **ComparativeGenomeAssember** и оруђе **ComparativeUtil**. **ComparativeGenomeAssember** и Padena представљају програмске реализације два алгоритма за састављање генома.

.Net Bio Framework програм и документација су доступни на адреси:
http://bio.codeplex.com

Увод

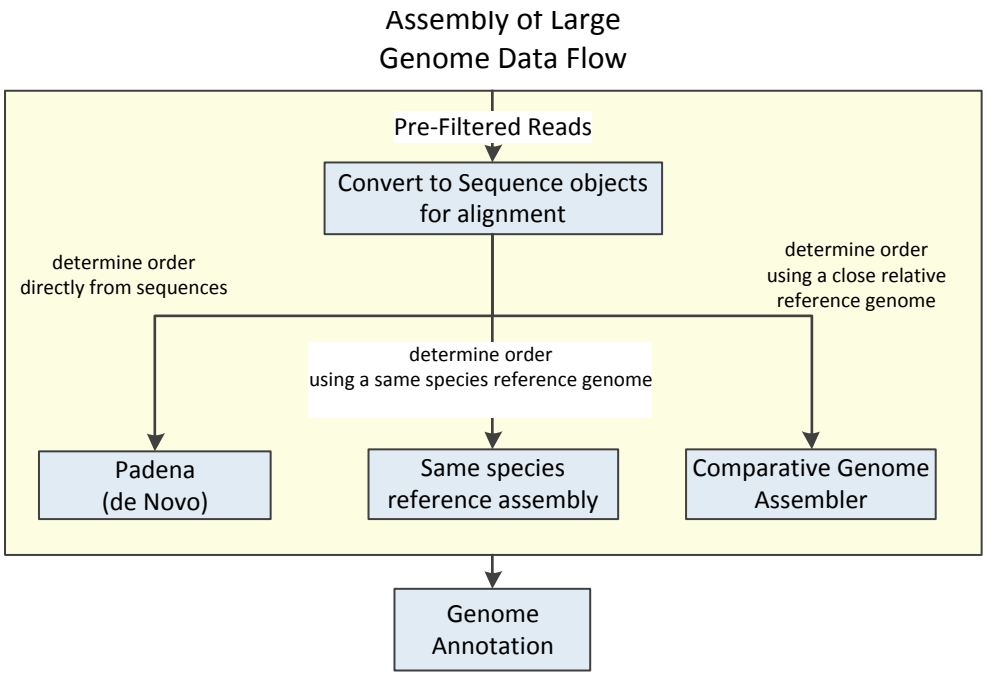
Са све већим бројем генома за које је публикована тзв. референтна секвенца, све више и више врста имају тзв. блиске рођаке са познатим референтним секвенцама. Стога се због заједничког еволутивног поријекла очекује да су велики дијелови блиско сродних генома веома слични. Имајући у виду ову претпоставку, може се значајно добити на временау, а и квалитету, у процесу састављања генома. Штавише, de-novo састављање је неколико редова величине спорије од референтног састављања, а и захтијева много више меморије. Ради искоришћавања свих предности које пружа референтно формирање генома, .NET Bio Framework библиотека „на свом репертоару" нуди софтверски додатак Comparative Assembly. Додатак се може искористити за спајање очитавања једног генома (тзв. циљног) на основу секвенце сродног/сличног генома (тзв. референце), као обрасца. Ова техника се показује нарочито дјелотворном у ситуацијама када је потребно секвенцирати велике и сложене геноме. Поново, de-novo техника, код које се веома много преклапајућих секвенци користи за одређивање редослиједа нуклеотида, и то непосредно на основу самих тих (под)секвенци, је неколико редова величине спорија и меморијски захтјевнија него референтно састављање.

Постоје три главна приступа у састављању геномских секвенци:

- **de-novo:** спајање се врши само на основу добијених очитавања циљног генома; De Novo секвенцирање је почетно секвенцирање чији резултат је примарна генетичка секвенца организама.

- **референтно састављање**: спајају се очитавања циљног генома користећи као референцу неки сродан/сличан геном, а с циљем груписања блиских очитавања и на основу чега би се формирала секвенца чији би већи дијелови требали бити скоро идентични референтној секвенци; концептуално, референца има улогу темеља у процесу самог састављања.

- **преспајање (re-assembly)**: спајају се очитавања циљног генома користећи као референцу већ једном одређени геном дате врсте; резултат је формирање индивидуалне секвенце дате врсте која је јако слична, али не и идентична референтној секвенци; у овом случају референтни геном представља примјер генетског кода исте врсте; сама референца је, иначе, довољно добра апроксимација ДНК неког конкретног живог бића.



.NET Bio Framework садржи реализације алгоритама за de novo и референтно формирање генома.

Напомена: Постоји програм PadenaUtil, који се позива са командне линије. Њиме се дефинише формирање суперконтига.

За више информација о .NET Bio Framework Parallel de Novo Assembler-у (Padena) погледати Bio Parallel de Novo Assembler Technical Guide.docx на Codeplex или на $**..\Bio\Doc**

Могуће ситуације

Геном велике биљке и сродна/слична референца

Могу се искористити геноми двију биљака које су генетски веома сличне једна другој, при чему је један много већи од другог.

Информације из мањег генома се могу искористити за реконструкцију већег генома када је мањи већ секвенциран и формиран de novo методом. Што се тиче сложености и потрошеног времена, састављање је спорије неколико редова величине и захтијева много више меморије од референтног формирања. Разлог томе је што алгоритам за формирање генома мора да упореди свако очитавање са свим осталим очитавањима (сложеност класе $O(n^2)$, мада се може свести на $O(n \log(n))$.

Преспајање (Re-Assembly) с циљем бољег увида у мутације и SNP

Референтно формирање генома се користи у формирању генома већ секвенцираних организама како бисмо што више сазнали о мутацијама и SNP-у. Нацрт људског генома је био доступан већ 2001. г. Но, и даље је много питања отворене, на које није могуће дати одговор само на основу једне копије људског генома. Због тога је и започет пројекат 1000

генома, којим научници желе да дешифрују фенотипске варијације изазване мутацијама и SNP-ом. De Novo секвенцирање, у овом случају, не долази у обзир ни као опција. Боље је искористити већ секвенцирани људски геном као референцу, сравнити очитавања на основу ње, и наћи варијације међу различитим копијама људског генома.

Формирање секвенци различитих сојева

Више сојева Mycobacterium tuberculosis, Streptococcus pneumoniae и Staphylococcus aureus су секвенцирани како би се разумјела вируленца, резистенција на лијекове и многе друге фенотопске разлике између њих. Ако је секвенца једног соја доступна, референтно формирање генома се може користити како би се даље настављале секвенце других сојева.

Преглед референтног састављања генома

Сравњивање представља методологију за аранжирање секвенци ДНК, РНК и протеина, с циљем идентификације сличних дијелова који могу бити важни за функционалну, структурну и еволутивну везу међу секвенцама.

Помоћу **ComparativeGenomeAssembler** можете да искористите већ секвенционисан блиско родни геном као референцу на основу које се траже поклапања у циљном геному.

1. Read Alignment (сравњење очитавања)
    a. Позива NUCmer
    b. NUCmer позива MUMmer
2. Repeat Resolution (позиционирање понављајућих секвенци)
3. Layout Refinement (побољшавање диспозиције)
4. Consensus Generation (формирање усаглашености (контига))
5. Scaffold Generation (формирање суперконтига)

Процес референтног састављања почиње сравњивањем очитавања према референтном блиско родном геному користећи NUCmer. Даље, NUCmer користи MUMmer како би сравнио свако shotgun-очитавање са референтним геномом. По завршетку процеса, повратне вриједности су делта сравњивања.

У другом кораку референтног састављања настоје се разријешити проблеми око очитавања за које није одређена само једна позиција. Овај корак захтијева информације mate-pair како би се позиционирале понављајуће секвенце.

Током референтног састављања очитавања циљног генома се само дјелимично поклапају са блиско родним геномом. Ово је резултат геномских разлика које су довеле до постојања двије различите врсте. Очекује се да процес мора да рачуна о новим SNP-овима, инделима (**in**sertion и **del**etion), транслокацијама, хромозомском дуплирању, и разним реконструкцијама. Током трећег корака референтна диспозиција, између циљног и референтног генома, побољшава се анализом индела и реконструкција, успут смањујући грешке у секвенцирању, укључујући и корекцију индела за које се сматра да су фиктивни. Током овог поступка се користи mate-pair информација.

За сваку групу преклапајућих очитавања при побољшаној диспозицији, врши се *вишеструко сравњивање* ради формирања секвенце усаглашености за геномски регион, покривен датим очитавањима. Током овог референтног састављања алгоритам за сравњивање-усаглашавање је коришћен да одреди секвенцу усаглашености новог генома.

Алгоритам пролази кроз сва делта сравњивања, код сваког индекса контига са делтом, и налази сравњење усаглашености. У последњој етапи формирања усаглашености, састављач прави сравњење свих очитавања које покривају геном и на основу тога, као усаглашеност сравњених очитавања, одређује оригиналну секвенцу генома.

Резултат формирања усаглашености је скуп контигних ДНК секвенци, чије релативно мјесто у геному није дефинисано. Процедура scaffolding се користи да посложе и усмјере контизи користећи информације добијене упареним очитавањем. Послије формирања контига, референтни састављач користи информацију mate-pair да посложи и усмјери контиге, а и да их уклопи у веће структуре зване scaffolds или суперконтизи.

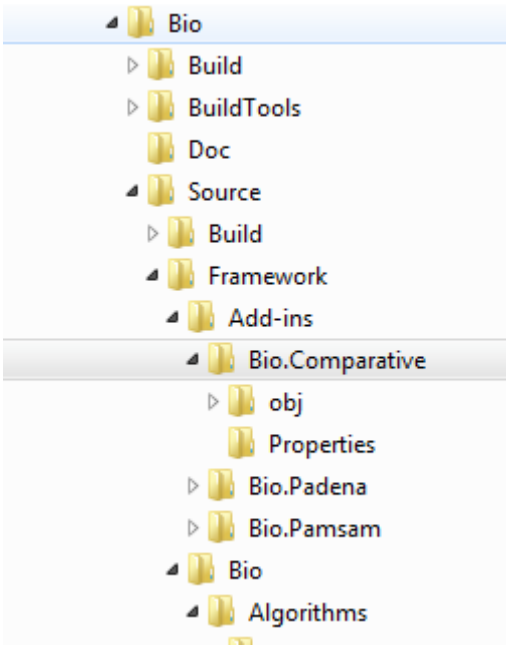
Контизи се повезују користећи:

- преклапања
- клонове
- сравњивање према референтном геному
- сравњивање према одговарајућим физичким мапама
- похраном генетске синтеније (енг. synteny)

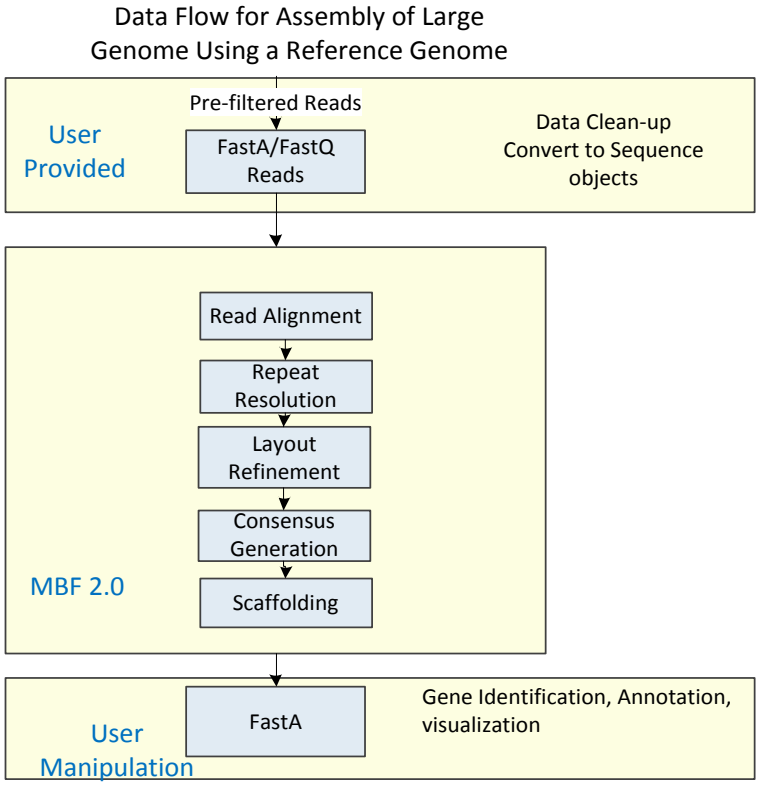
Образац референтног састављања генома

ComparativeGenomeAssembler класа је програмска реализација референтног састављача генома за састављање ДНК секвенци. Формирање одговарајућег програмског објекта се врши у пет корака. Штавише, сваки корак је независан један од другог, дозвољавајући тиме корисницима да рукују са подацима прије него што их искористе у наредном кораку.

ComparativeGenomeAssembler класа је доступна као додатак.



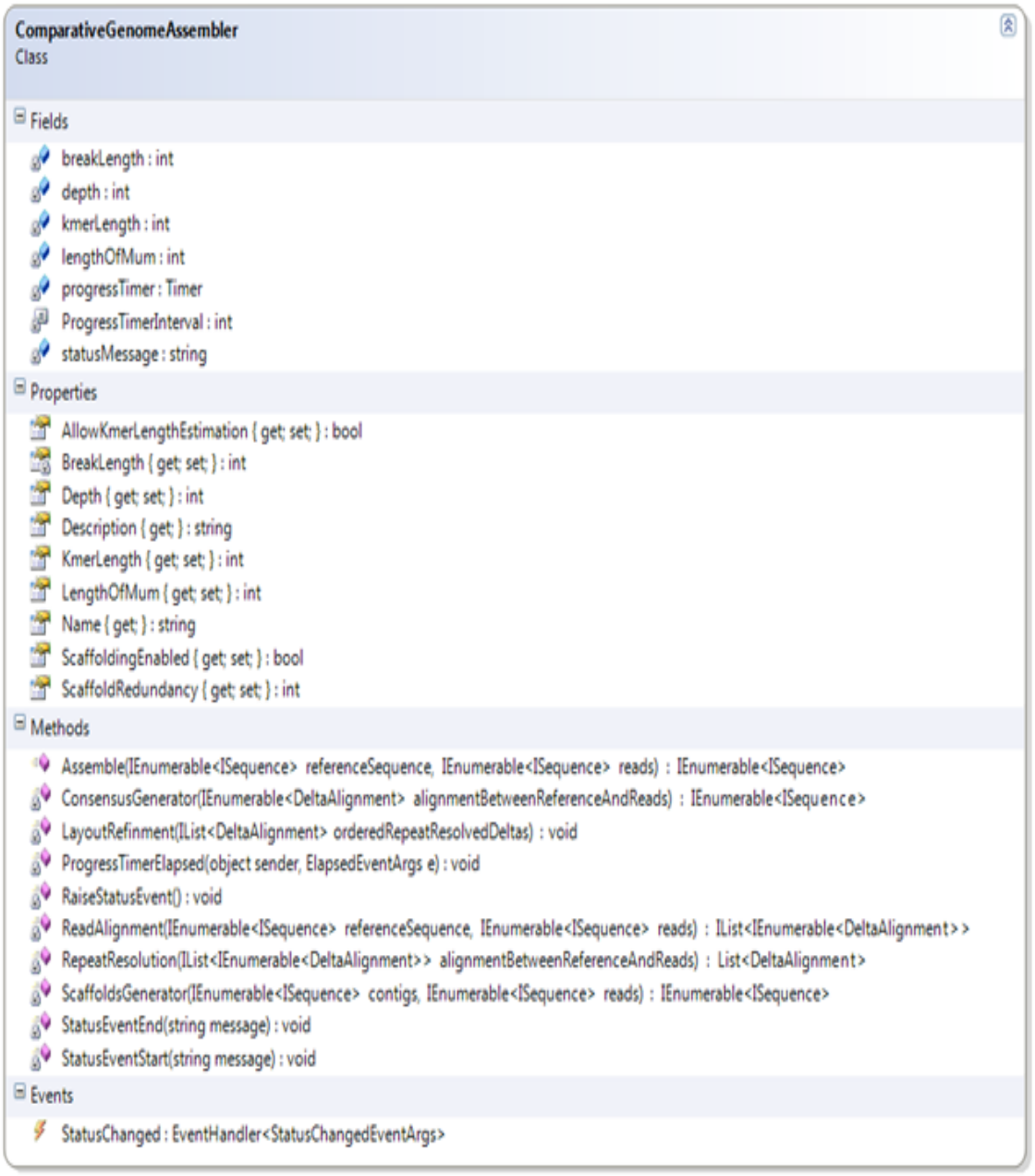
Сљедећа слика приказује кораке референтног састављања.



Напомена: референтно састављање се може покренути са командне линије, користећи оруђе ComparativeUtil, као што је описано у Референтно састављање са командне линије.

Class Diagram ComparativeGenomeAssembler

ComparativeGenomeAssembler је програмска реализација референтног састављача ДНК секвенци.



Процеси референтног састављања

За извођење референтног састављања неопходни су сљедећи процеси и морају се редом извршавати:

ComparativeGenomeAssembler Processes

Процеси	Опис

Процеси	Опис
ReadAlignment	Први корак. Користи се како би се сравнила очитавања, према референтном геному, користећи NUCmer. ReadAlignment може наћи више позиција гдје се одређена очитавања „уклапају" према референци. Проблематична позиционирања се раз-рјешавају у сљедећем кораку. Корисник може штошта још да ради са подацима, прије него што их искористи у сљедећем кораку, употребом оруђа NucmerUtil са командне линије.
RepeatResolution	Други корак референтног састављања. Користи се за рјешавање проблематичних ситуација настализ због лошег позиционирања неких очитавања. Корисник може додатно искористити податке, прије него што их употријеби у сљедећем кораку, користећи RepeatResolutionUtil са командне линије.
LayoutRefinement	Трећи корак референтног састављања. Користи се за побошљање диспозиције између циљног и референтног генома, анализом индела и (текућих) реконструкција. Корисник може додатно искористити податке, прије него што их употријеби у сљедећем кораку, користећи LayoutRefinementUtil са командне линије.
ConsensusGeneration	Четврти корак упоредног састављања. Користи се при формирању секвенце усаглашености на основу груписаних преклапајућих очитавања, за геномску област коју покривају дата очитавања. Корисник може додатно искористити податке, прије него што их употријеби у сљедећем кораку, користећи ConsensusUtil.exe са командне линије.

Процеси	Опис
ScaffoldGeneration	Пети корак упоредног састављања. Користи се за формирање суперкконтига. Корисник може додатно искористити податке, прије него што их употријеби у сљедећем кораку, користећи ScaffoldUtil са командне линије.

За више информација о оруђима командне линије погледајте Референтно састављање са командне линије, при крају документа.

Метод класе ComparativeGenomeAssembler

Као додатак овим атомичним операцијским компонентама, метод **ComparativeGenomeAssembler.Assemble()** се може искористити извршавање операција у погодно вријеме. **ComparativeGenomeAssembler.Assemble()** метод представља програмску реализацију алгоритма референтног састављања генома за склапање највећих могућих контига на основу узлазних секвенци.

```
int KmerLength = 11;
int MumLength = 20;
ComparativeGenomeAssembler asmblr = new ComparativeGenomeAssembler();
asmblr.ScaffoldingEnabled = false;
asmblr.KmerLength = KmerLength;
asmblr.LengthOfMum = MumLength;
IEnumerable<ISequence> assemblerResult = asmblr.Assemble(referenceSequences, sequences);
```

Гдје је

- **referenceSequence** је секвенца која се користи као *референца* за референтно упаривање очитавања *циљног* генома.
- **sequence** је очитавање из *циљног* генома
- а резултат је FastA датотека са суперконтизима и несравњеним секвенцама контига.

Ово за резултат даје и инстанцу **IComparativeAssembly**, која садржи листу састављених секвенци.

```
            //Comparative Assembly Steps
            //1) Read Alignment (Calling NUCmer for aligning reads to reference sequence)
            StatusEventStart(Properties.Resources.ReadAlignmentStarted);
            IList<IEnumerable<DeltaAlignment>> alignmentBetweenReferenceAndReads =
this.ReadAlignment(referenceSequence, reads.Where( a => a.Count >= LengthOfMum));
            StatusEventEnd(Properties.Resources.ReadAlignmentEnded);

            // 2) Repeat Resolution
            StatusEventStart(Properties.Resources.RepeatResolutionStarted);
            IList<DeltaAlignment> repeatResolvedDeltas =
this.RepeatResolution(alignmentBetweenReferenceAndReads);
            StatusEventEnd(Properties.Resources.RepeatResolutionEnded);

            StatusEventStart(Properties.Resources.SortingResolvedDeltasStarted);
```

```
            List<DeltaAlignment> orderedRepeatResolvedDeltas = repeatResolvedDeltas.OrderBy(a =>
a.FirstSequenceStart).ToList();
            StatusEventEnd(Properties.Resources.SortingResolvedDeltasEnded);

            // 3) Layout Refinement
            StatusEventStart(Properties.Resources.LayoutRefinementStarted);
            LayoutRefinment(orderedRepeatResolvedDeltas);
            StatusEventEnd(Properties.Resources.LayoutRefinementEnded);

            // 4) Consensus Generation
            StatusEventStart(Properties.Resources.ConsensusGenerationStarted);
            IEnumerable<ISequence> contigs =
this.ConsensusGenerator(orderedRepeatResolvedDeltas.OrderBy(a => a.FirstSequenceStart));
            StatusEventEnd(Properties.Resources.ConsensusGenerationEnded);

            if (ScaffoldingEnabled)
            {
                // 5) Scaffold Generation
                StatusEventStart(Properties.Resources.ScaffoldGenerationStarted);
                IEnumerable<ISequence> scaffolds = ScaffoldsGenerator(contigs, reads);
                StatusEventEnd(Properties.Resources.ScaffoldGenerationEnded);

                return scaffolds;
            }
            else
            {
                return contigs;
            }
```

Корак 1 – Read Alignment (сравњивање очитавања)

У овом кораку се позива **ReadAlignment** да се сравне очитавања за референтни геном, помоћу NUCmer-а. Исход корака је делта сравњење.

Напомена. Comparative Assembly формира (програмски) изузетак када наиђе на проблематично очитавања.

Свако очитавање се пореди са референтним геномом користећи MUMmer ради проналажења заједничких поднизова одређене дужине. Ово даје листу потенцијалних локација за груписање очитавања. Еволутивне промјене између референтног и циљног генома могу произвести ситуације да MUMmer има једно или више контигних поклапања, више дјелимичних поклапања, или да нема никаквих поклапања. Понављајуће секвенце и разнообразност између циљног и референтног генома производи код неких очитавања неконтигна сравњења. Користи се прилагођена верзија алгортима Longest Increasing Subsequence (LIS) за формирање ланца међусобно конзистентних поклапања између сваког очитавања и одговарајуће референце. Као додатак најдужем конзистентном ланцу формира се и скуп скоро оптималних ланаца, ради идентификације очитавања обиљежених у понављањима. Очитавања која нису изричито позиционирана у геному (један или више ланаца су унутар 2% од оптималне позиције) су класификована као понављајућа, а дати проблем се касније рјешава (у неким случајевима) користећи информацију mate-pair.

```
IList<IEnumerable<DeltaAlignment>> alignmentBetweenReferenceAndReads =
this.ReadAlignment(referenceSequence, reads.Where( a => a.Count >= LengthOfMum));
```

При чему су очитавања у FastA или FastQ формату.

што заузврат користи MUMmer са референтном секвенцом. NUCmer позива MUMmer да потпомогне његов high performance алгоритам за одређивање максимума потпуних поклапања.

```
List<IEnumerable<DeltaAlignment>> deltaAlignments = new
    List<IEnumerable<DeltaAlignment>>();
Parallel.ForEach(referenceSequence, sequence =>
{
    NUCmer nucmer = new NUCmer((Sequence)sequence);
    …
    foreach (ISequence qrySequence in reads)
     {
          deltaAlignments.Add(nucmer.GetDeltaAlignments(
               qrySequence, false));
     }
}
```

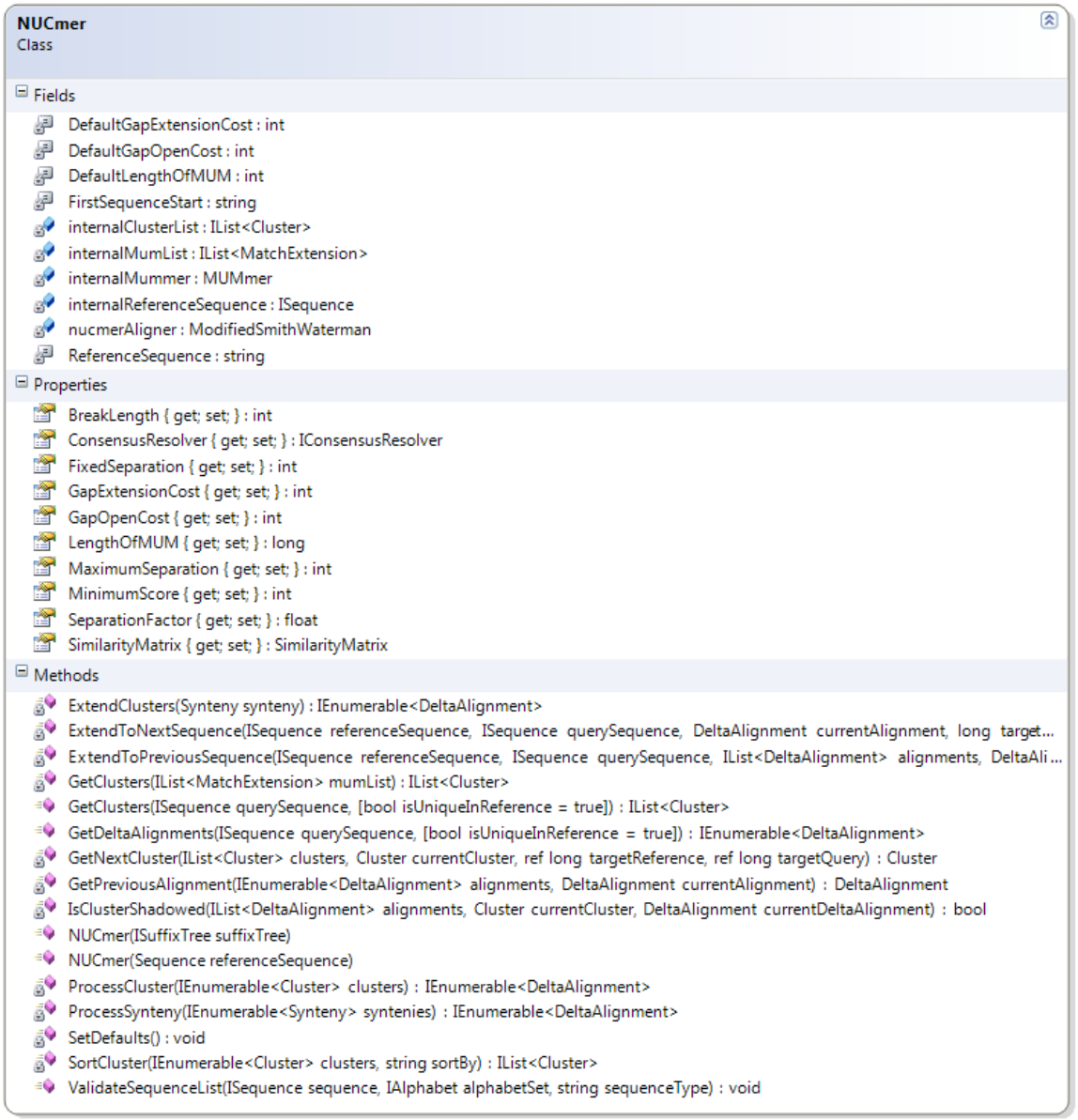
```
internalMummer = new MUMmer.MUMmer(referenceSequence);
```

NUCmer (NUCleotide MUMmer) дозвољава сравњивање скупа секвенци са скупом референци према m:m обрасцу. Конкретно, постоје три корака:

1. максимално потпуно поклапање
2. груписање поклапања
3. проширење сравњења

Почиње се коришћењем MUMmer-а који налази сва максимална јединствена поклапања задате дужине, између двије задате секвенце. У сљедећем кораку, појединачна поклапања се групишу у блиско груписане скупове **mgaps**-ом. Коначно, непотпуне секвенце између поклапања се сравњују модификованим Smith-Waterman алгоритмом, а груписања се проширују ради повећања свеукупне покривености сравњења.

Дијаграм класе NUCmer



NUCmer је систем за убрзано сравњивање цијелих генома или изразито великих ДНК секвенци. Дозвољава сравњивање скупа референтних секвенци са скупом циљних секвенци. Ово се обично користи за идентификацију позиције и смијера скупа контига неке секвенце, према већ састављеној секвенци. Оквирно дефинише и NUCmer алгоритам, при чему се неки кораци препуштају методима изведених класа.

Напомена: велике бинарне датотеке са сравњењима су подржане само на 64-битним машинама – 32-битне машине дају обавјештење о недостатку меморије.

MatePair

MatePair класа омогућује похрањивање очитавања са одговарајућом информацијом из програмске библиотеке, а **MapPairMapper** класа омогућује претварање улазне листе очитавања у упарена очитавања на основу информација доступних из FASTA заглавља.

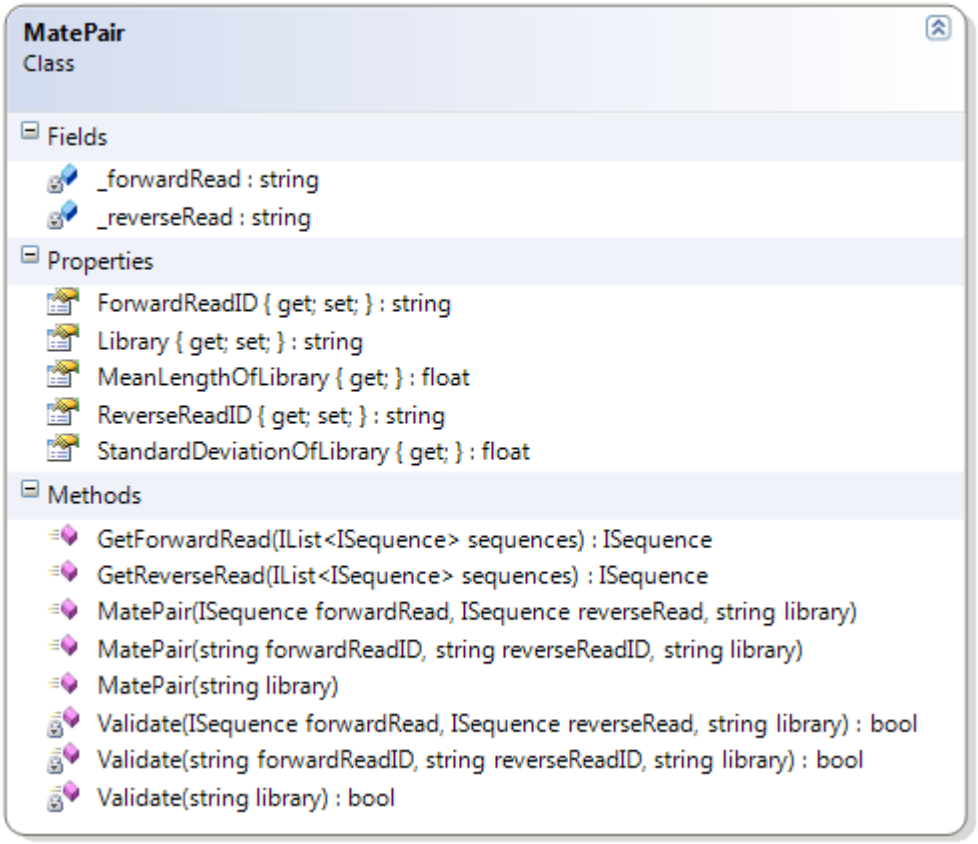
Подржани mate-pair формати

Формат	**опис**
chrI0.X1:abc ATGC	регуларна очитавања
chrI0.Y1:abc TACG	обратна очитавања
chrI0.F:abc ATGC	регуларна очитавања
chrI0.R:abc TACG	обратна очитавања
chrI0.1:abc ATGC	регуларна очитавања
chrI0.2:abc TACG	обратна очитавања

Где:

- X1,F,1 означавају регуларна очитавања
- Y1,R,2 означавају обратна очитавања
- abc означава назив библиотеке
- chrI0 је ID секвенце

Дијаграм класе MatePair



Корак 2 – Repeat Resolution (позиционирање понављајућих секвенци)

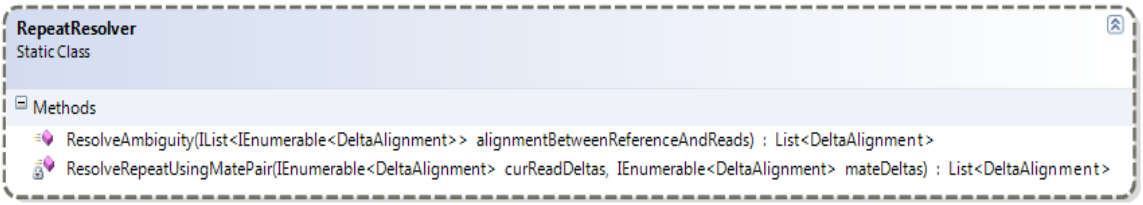
У овом кораку се рјешава проблем са очитавањима којима нису прецизно одређени положаји. Овај корак захтијева информацију о спаривању за разрјешавање поменутих недоумицака Информација о спаривању се користи на сљедећи начин:

1. Ако је paired-end секвенца јединствено обиљежена у самом геному, репетитивно очитавање је позиционирано тако да задовољава ограничења узрокована mate-pair информацијом.
2. Ако размјештај самих парњака није прецизан, настоји се открити да ли информација о упаривању дозвољава смијештање оба парњака. У неким случајевима, постоји само једно мјесто за обојицу – очитавање и његов пар, а који задовољава ограничења упаривања.
3. Ако постоји више повољних локација, нека од њих се наусмично издваја.

```
private List<DeltaAlignment> RepeatResolution(IList<IEnumerable<DeltaAlignment>>
alignmentBetweenReferenceAndReads)
{
    return RepeatResolver.ResolveAmbiguity(alignmentBetweenReferenceAndReads);
}
```

гдје:

- **alignmentBetweenReferenceAndReads** представља сравњење између референтног и очитаног генома.
- **referenceSequence** представља секвенцу референтног генома.
- **reads** представља листу очитавања секвенце у FastA или FastQ формату.



Корак 3 – Layout Refinment (побољшање диспозиције)

Пошто се очитавања циљаног генома тек дјелимично поклапају са референтним геномом, треба порадити на инделима (**in**sertions и **del**etions) и реконструкцији. Овај корак побољшава диспозицију између циљног и референтног генома, имајући у виду инделе и реконструкције засноване на информацији о сравњењу, садржане у делта датотеци. Mate-pair информација се користи за позиционирање понављајућих секвенци. Иначе, размјештај може бити и насумичан, чиме се симулира разномјерна покривнеост.

Побољшавање диспозиције користи кораке из de novo састављања за спајање несравњених очитавања. Претходни корак помаже у доношењу одлуке која се тиче поклапања самих очитавања, која се могу спојити користећи de novo технику за генерисање оног дијела који недостаје, док се каснији scaffolding кораци користе за груписање контига. Овај корак је паралелизован за сваки скуп еволутивних догађаја као што су индели и реконструкције. Тешкоће могу настати због парцијалног поклапања очитавања циљног генома са референцом, или пак због сусједних одјељака очитавања који се могу поклопити са удаљеним комадима референтног генома. У оба случаја прекида се алгоритам.

Трећи корак садржи **LayoutRefinment** метод:

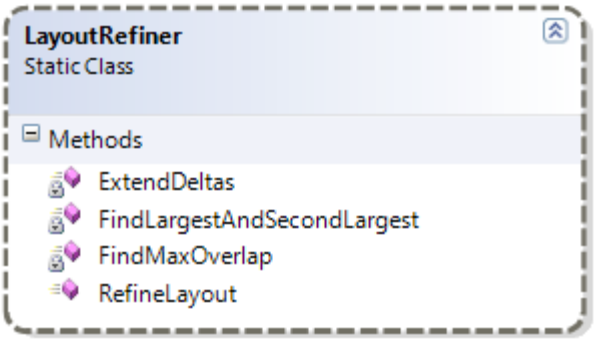
Код:

```
LayoutRefinment(IList<DeltaAlignment> orderedRepeatResolvedDeltas)
```

У суштини се позива **LayoutRefiner.RefineLayout()** метод.

```
private void LayoutRefinment(IList<DeltaAlignment> orderedRepeatResolvedDeltas)
        {
            LayoutRefiner.RefineLayout(orderedRepeatResolvedDeltas);
        }
```

Дијаграм класе LayoutRefiner



Заглавље класе **RefineLayout**:

```
public static void RefineLayout(IList<DeltaAlignment> orderedDeltas)
```

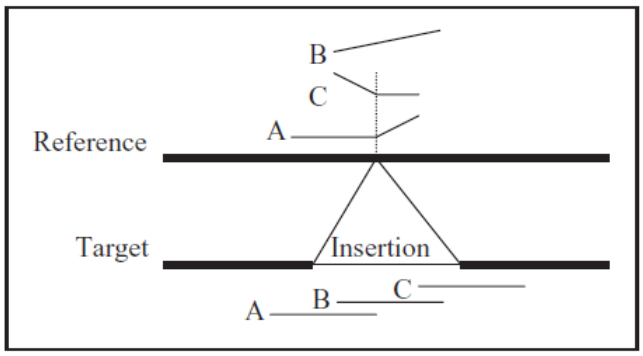
Током LayoutRefinement процеса разматрају се сљедећи индели и реконструкције:

- Уметања у циљ
- Уметања у референцу (уклањања из циља)
- Реконструкције
- Дивергентна ДНК

- Грешке и разнообразност

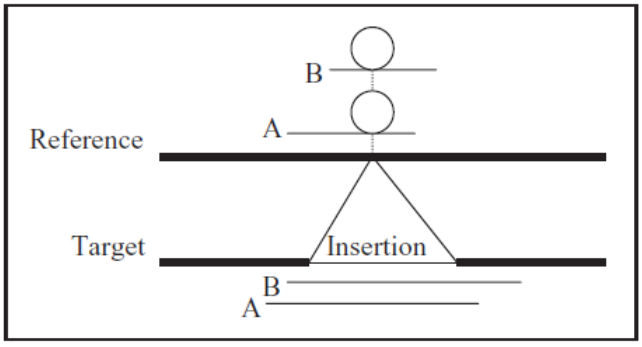
Уметања у циљ

Прва врста уметања се јавља када се дијелови очитавања не поклапају са референтним геномом. Резултат је приказан на сљедећој слици. У таквим случајевима MUMmer не даје никакву информацију о односу међу очитавањима A, B, и C. Ипак, пошто знамо њихове позиције, њихов однос може бити прорачунат, а сам проблем рјешен коришћењем *de novo* метода.



Уметање у циљни геном. Доњи дио показује коректну диспозицију очитавања A, B и C, док косе линије указују на дјелове ишчитавања који се не подударају.

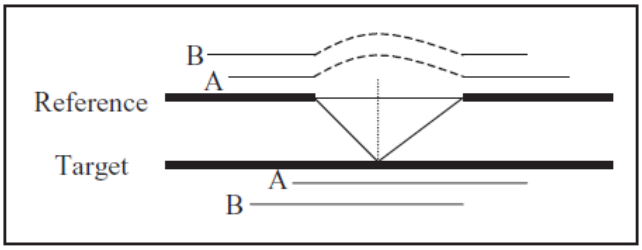
Ако је уметање довољно мало да се укључи у једно очитавање, средина очитавања неће одговарати референци, али зато хоће крајеви (као на слици). Примјер појаве сравњивања које се ломи и наставља на истом мјесту у референци, али на различитим мјестима у очитавању, је својеврсно свједочанство о оваквим уметањима као појавама.



Кратка уметања у циљни геном. Кружићи представљају дјелове који нису сравњени са референцом.

Уметања у референцу (уклањања из циља)

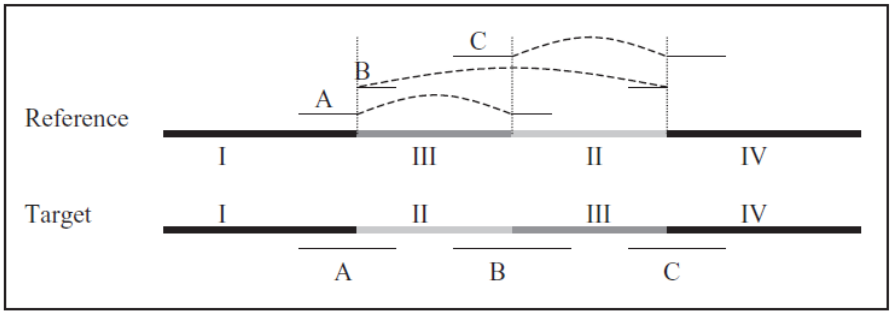
У случају уметања у референцу, очитавања која захватају тачку уметања повезују двије растављене области саме референце, као што је приказано на слици. Њихов релативни положај, као и положај околних очитавања, лако може бити одређен.



Уклањање из циља. Испрекидане линије указују на „протезање" очитавања потребних за сравњивање са референцом.

Реконструкције

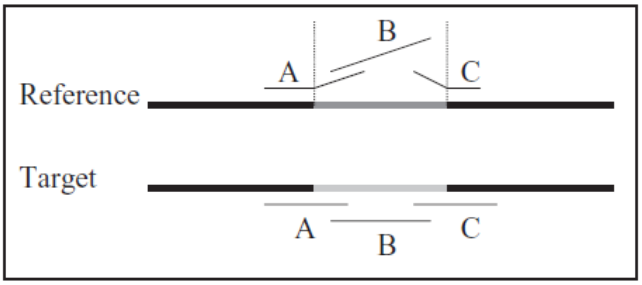
Транслокације (биле оне реципрочне или не) и инверзи су поприлично изазовне ситуације за референтно састављање. Алгоритам се заснива на конзервативном приступу, ограничавајући се на рјешавање проблема у областима које имају *ознаку* уметања у референцу, о чему је било ријечи у претходној тачки. Сљедећа слика илуструје такву ситуацију: редослијед појављивања области II и III у циљу и референци је различит. Очитавање A идентификује област III као уметање у референцу, између области II и III. Очитавање C идентификује област II као уметање између области III и IV. Очитавање B нема ознаку уметања у референцу, стога је резултат алгоритма пар контига (I + II и III + IV), који ће бити спојени у scaffolding кораку .



Реконструкције – области II и III из циља се појављују у различитом редослиједу у референци.
Сва очитавања одговарају референци, али на различитим мјестима (испрекидане линије), али само очитавања A и C имају ознаку уметања у референцу.

Дивергентна ДНК

Ако су оба генома имала довољно времена за неке веће мутације, секвенце се више не могу сравнити јер је разлика између њих већа од прага сравњивања (сљедећа слика). Пошто је ово у суштини уметање у циљ, резултат се састоји од два контига.

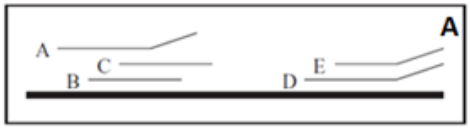


Значајна разлика између два генома. Косе линије означавају дијелове очитавања који се не поклапају са референцом.

Грешке и разнообразност

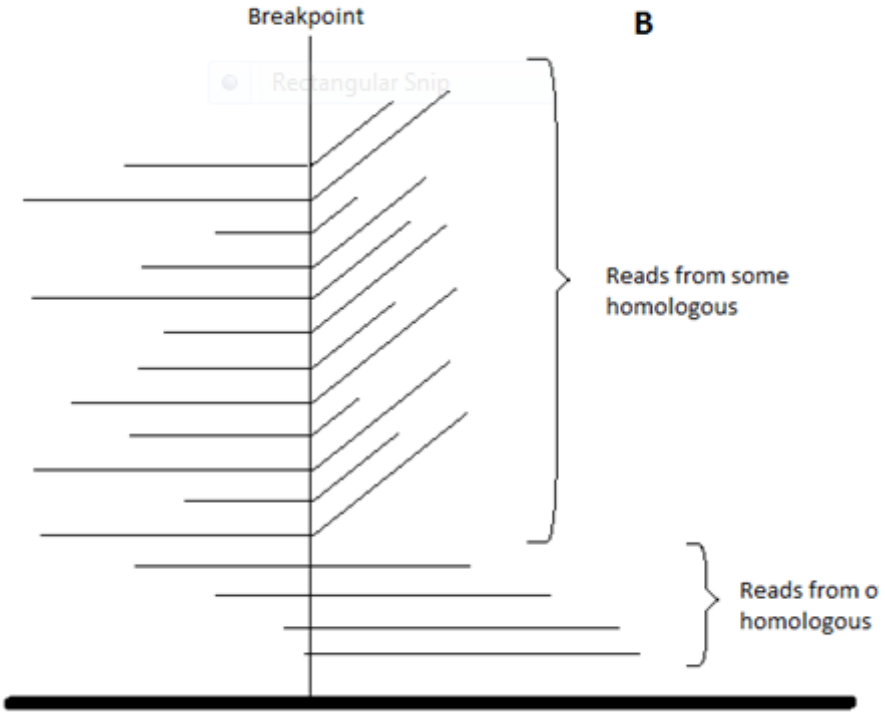
Грешке настале у фази припреме, или пак самог секвенцирања, имају сличне ознаке као и оне које су изазване конкретном разнообразношћу међу геномима. Прије сваког састављања, потребно је проценити квалитет свих добијених очитавања.

Након идентификације очитавања са најмање једном тачком прелома, сравњивање очитавања са референцом се састоји од више непрекидних одсјечака. Такође се идентификују и друга очитавања која имају, или пак захватају, исту тачку прелома. У оваквим ситуацијама нам проста статистика даје одговор на питање да ли се ради о грешки, или пак реконструкцији (видјети слику).



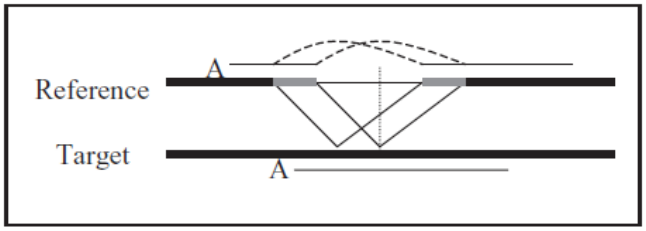
A) Уочавање грешке. Очитавање A вјероватно није тачно, јер B и C нису сагласни са A; очитавања D и E указују, вјероватно, на полиморфизам.

У случајевима када имамо више од два упарена (хомологна) скупа хромозома (полиплоидија), неопходна је нарочита пажња при одлучивању, јер један или више хомолога може да има преломну тачку, која је обухваћена неким другим хомологом (погледати слику).



Б) С обзиром да имамо много копија истих хромозома, није лако одлучити да ли је нешто грешка или пак полиморфизам.

Ако је разнообразност окружена кратким понављањима, као на наредној слици, LIS – алгоритам мора допустити преклапање међу сусједним сравњивањима са референцом. У овом случају, позиција очитавања које прате уметања су прилагођене тако да допуштају присуство краћег понављања.



Кратка понављања око сравњења очитавања са референцом. Испрекидане линије повезују дијелове очитавања A које се два пута појављује у референци, али једном у A (самим тим и у циљном геному).

Корак 4 – Consensus Generation (формирање усаглашености (контига))

У овом кораку се генерише секвенца усаглашености за област генома која је обухваћена очитавањима. За сваку групу преклапајућих очитавања у побољшаној диспозицији, *вишеструко сравњивање* генерише претходно поменуту секвенцу усаглашености. Због овога се врши 2-по-2 сравњивање сваког очитавања са текућом секвенцом усаглашености. Резултат процеса се користи за генерисање нове секвенце усаглашености, па се овај корак

састоји и од читаве серије 2-по-2 сравњивања Progressive Consensus Generation алгоритмом.

Код:

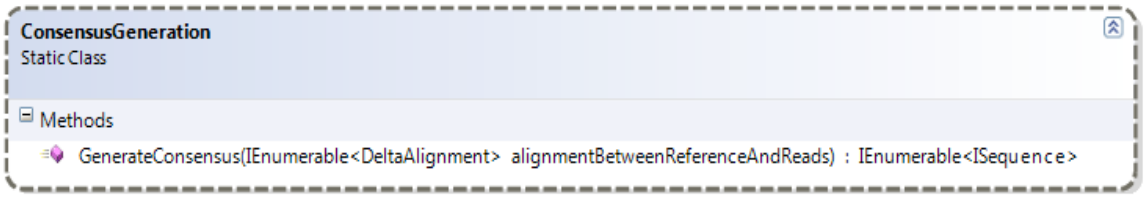
```
IEnumerable<ISequence> GenerateConsensus(IEnumerable<DeltaAlignment>
alignmentBetweenReferenceAndReads)
```

при чему:

- **alignmentBetweenReferenceAndReads** је улазна листа очитавања.
- повратна вриједност (резултат) је датотека са листом генерисаних секвенци контига; сваки елемент листе се састоји од одступања сравњења с почетка и саме секвенце.

```
outputSequences.Add(currentAlignmentStartOffset, new
Sequence(AmbiguousDnaAlphabet.Instance, currentContig.ToArray(), false));
```

Дијаграм класе ConsensusGeneration



Корак 5 – Scaffold Generation (формирање суперконтига)

Четврти корак, генерисање усаглашености, резултира низом контига. Даље, искориштава се информација о спаривању за откривање дислокације контига и њихових усмјерења, како би се формирао суперконтиг, за шта се обично користи други софтвер.

Суперконтиг је састављен од великог броја контига и тзв. расцјепа који представљају дијелове геномске секвенце реконструисане на основу shotgun-очитавања. Сваки контиг представља непрекидно парче геномске секвенце, код којег је степен повјерења за редослијед база јако велик. Расцјепи су резултат преклапајућих очитавања, тј. очитавања из два секвенцирана завршетка бар једног фрагмента који се преклапа са другим очитавањем у два различита контига који могу бити и сусједни.

Резултат преклапајућих суперконтига може бити размијештање секвенце на више позиција, на више суперконтига, иако се у ствари ради само о једној јединој локацији. Ово се дешава јер је сасвим уобичајено да већи проценат гена буде формиран на основу организације контига у непотпуни мозаик састава зван „суперконтиг”. Непрецизности у редослиједу и

усмјерењу контига могу значајно повећати број могућих размјештаја. Дате нејасноће могу, наравно, замаглити стварну структуру гена.

Референтно састављање генома користи референтни геном за дефинисање контигних спојева. Укупна покривеност генома скупом контига непосредно утиче на тачност резултата овог процеса (одређивања гена), заснованог на сличности са блиско сродним геном. Већа покривеност, тј. што ближе 100% покривености, омогућује већу вјероватноћу тачног лоцирања контига.

У овом кораку се формирају суперконтизи на основу контига и упарених очитавања у тзв. Padena Step 6 састављању. Padena Step 6:

```
PadenaAssembly assemblyResult = (PadenaAssembly)this.Assemble(inputSequences);
if (includeScaffolds)
{
    // Step 6: Build _scaffolds
    this.BuildScaffoldsStarted();
    IList<ISequence> scaffolds = this.BuildScaffolds(assemblyResult.ContigSequences);
    this.BuildScaffoldsEnded();

    if (scaffolds != null)
    {
        assemblyResult.AddScaffolds(scaffolds);
    }
}

return assemblyResult;
```

Напомена: референтни састављач захтјева **Bio.Padena.dll** за извоеђење ове операције, пошто генерисање суперконтига референтним састављањем садржи позив метода **GenerateScaffolds**, који је дефинисан у **Bio.Padena.dll**-у.

У овом кораку **ScaffoldsGenerator**() позива **Bio.Algorithms.Assembly.Padena.Scaffold.GraphScaffoldBuilder**, метод класе **BuildScaffold()**, преко које се у сштини обавља сав посао.

```
IEnumerable<ISequence> ScaffoldsGenerator(IEnumerable<ISequence> contigs, IEnumerable<ISequence>
reads)
```

Гдје је:

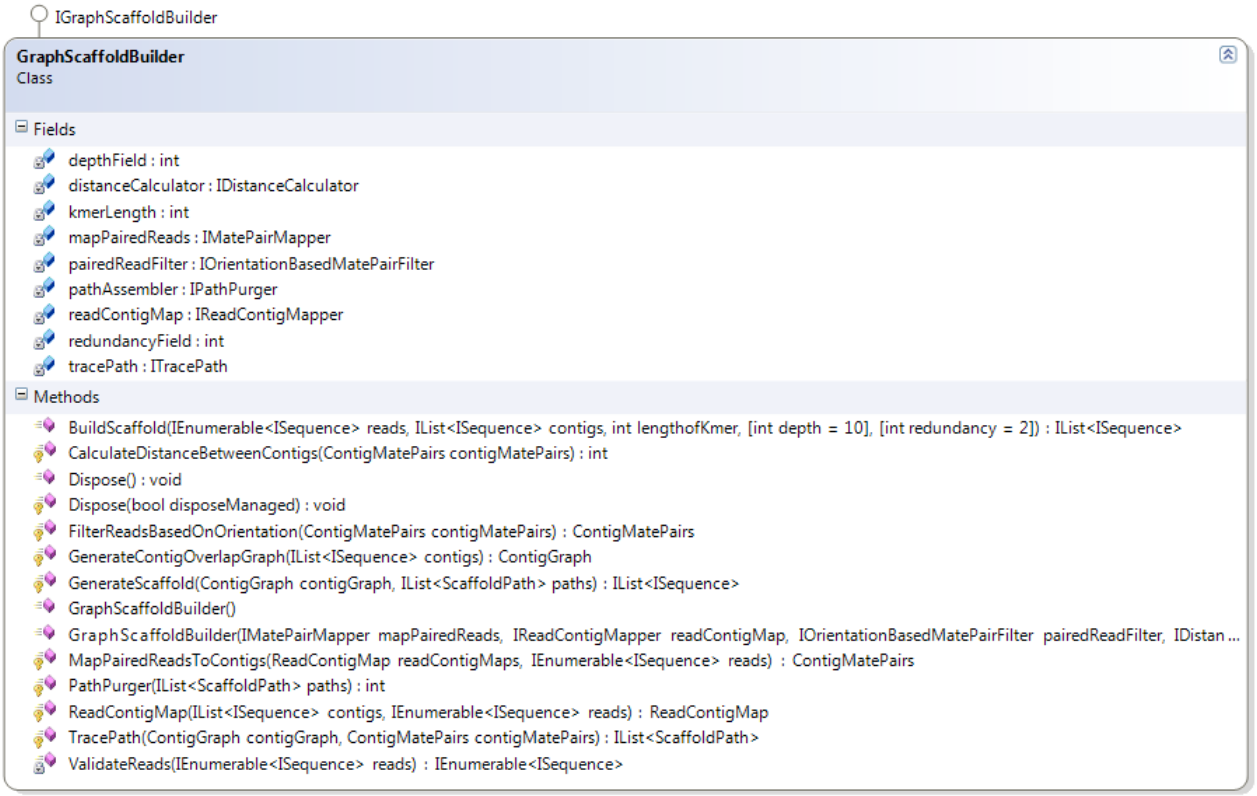
- **contigs** листа контига.
- **reads** листа упарених очитавања
- а резултат је FastA датотека са суперконтизима и несравњеним секвенцама контига

```
public IList<ISequence> BuildScaffold(
    IEnumerable<ISequence> reads,
    IList<ISequence> contigs,
    int lengthofKmer,
    int depth = 10,
    int redundancy = 2)
```

гдје је:

- **reads** је листа читача
- **lengthofKmer** theKmer дужина
- **depth** дубина обиласка графа
- **redundancy** број mate-pair-ова потребних за формирање везе међу контизима
- а резултат листа секвенци суперконтига.

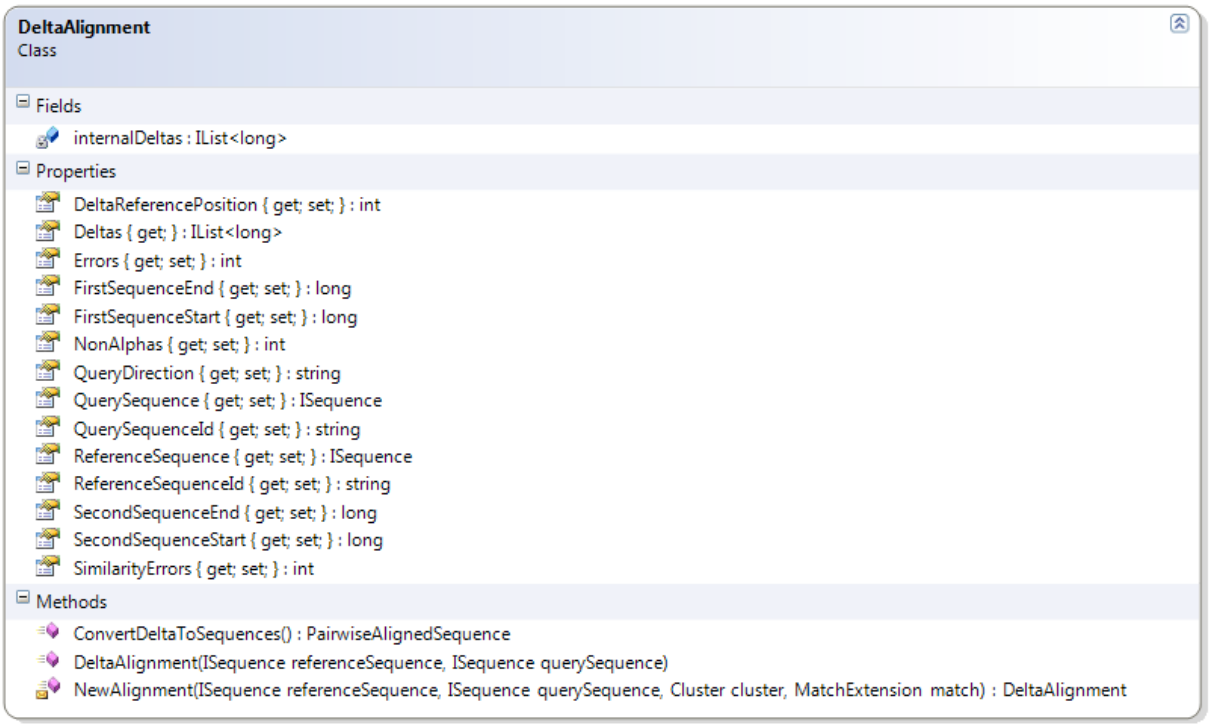
Дијаграм класе GraphScaffoldBuilder



Делта сравњивање

Делта сравњивање, као поступак, врши се у односу на тзв. делту – кодирану представу сравњења улазних секвенци. Садржи почетне и крајње индексе сравњења у референци и циљној секвенци, послије којих иду одступања (грешке) и листа цијелих бројева (у наредним линијама). Сваки цио број представља уметање (+ve) у референтну секвенцу, али и уклањање (-ve), такође из референтне секвенце.

Ова класа представља опис датог сравњивања (обиљежја програмских објеката и придружени методи).




```
public DeltaAlignment(ISequence referenceSequence, ISequence querySequence)
```

Референтно састављање путем командне линије

У овом дијелу је описана општа употреба програма (оруђа) ComparativeUtil, који се позива са командне линије, за референтно састављање генома.

Оруђа (командне линије) за референтно састављање

Оруђа	Опис
ComparativeUtil	Оруђе командне линије чија је сврха да иницијализује поступак референтног састављања и њиме започиње свих пет поменутих корака.
NucmerUtil	Користи се у 1. кораку ComparativeUtil-а. Подаци се додано могу употребити и у неке друге сврхе, прије него што се искористе у сљедећем кораку.

Оруђа	Опис
RepeatResolutionUtil	Користи се у опционом 2. кораку ComparativeUtil-а. Подаци се додано могу употребити и у неке друге сврхе, прије него што се искористе у сљедећем кораку.
LayoutRefinementUtil	Користи се у опционом 3. кораку ComparativeUtil-а. Подаци се додано могу употребити и у неке друге сврхе, прије него што се искористе у сљедећем кораку.
ConsensusUtil	Користи се у 4. кораку ComparativeUtil-а. Подаци се додано могу употребити и у неке друге сврхе, прије него што се искористе у сљедећем кораку.
ScaffoldUtil	Користи се у 5. кораку ComparativeUtil.-а Подаци се додано могу употребити и у неке друге сврхе, прије него што се искористе у сљедећем кораку.

ComparativeUtil

ComparativeUtil покреће поступак референтног састављања генома (одређивање редослиједа нуклеотида циљне секвенце). Сљедећа линија приказује синтаксу позива програма са командне линије:

```
ComparativeUtil.exe <options> <referencefilename> <readsfilename>
```

Сама употреба и рад ComparativeUtil-а је заснована на употреби (програмска) оруђа описаних у наставку.

Употреба ComparativeUtil.exe оруђа

По покретању ComparativeUtil-а, поступак референтног састављања пролази кроз свих пет основних корака.

```
ComparativeUtil.exe /s referenceFile  readsFile
```

Аргумент	Опис
ComparativeUtil	Команда којом се позива на извршавање програм за референтно састављање секвенце.
ComparativeUtil /s	Команда којом се при позиву програма налаже пролазак кроз свих пет корака састављања генома, с крајњим циљем добијања суперконтига.

Аргумент	Опис
referenceFile	FastA датотека која садржи референтне седвенце.
readsFile	FastA датотека који садржи очитавања.

Слиједе командне опције:

Options

Аргумент	Опис
h	„help" – приказује опис и употребу наредбе (програма). Подразумијевна врије-дност је **false**.
i	„meanlengthofinsert" – Mean Length од клон-библиотеке. Подразумијевана врије-дност је 0.
k	„kmerlength" – поставља дужину kmer-a. Подразумијевана вриједност је 10.
m	„Mumlength" – Mum Length. Подразумијевана вриједност је 20.
n	„clonelibraryname" – Clone Library Name. Подразумијевана знаковна ниска је празна ниска.
o	„outputFile" – излазна датотека.
s	„scaffold" – покретање суперконтиг-корака. Подразумијевана вриједност је **false**.
sd	„standarddeviationofinsert" – стандардно одступање од клон-библиотеке. Подразумијевана вриједност је 0.
v	„verbose" – опширни приказ записника вођеног током саме обраде. Под-разумијевана вриједност је **false**.

Такође, постоји и опција за појединачно извршавање сваког од корака.

NUCmer

ReadAlignment корак

NUCmer (NUCleotide MUMmer) дозвољава сравњивање више референци и више циљних секвенци.

Напомена: иако се оба, MUMUtil и NUCmerUtil, позивају у самом коду програма MUMera, MUMUtil је у суштини програмска реализација алгоритма Maximum Unique Match, док је NUCmerUtil инкорпорирао у себе још пар алгоритама, као што је генерисање кластера (групација) и сл.

```
NucmerUtil.exe [options] ReferenceFile QueryFile
```

Аргумент	Опис
NucmerUtil	Наредба за сравњивање више референци и више циљних секвенци.
ReferenceFile	FastA датотека која садржи референтне секвенце.
QueryFile	FastA датотека која садржи циљне секвенце.

Опције

Аргумент	Опис
b	“breaklength” - Distance an alignment extension will attempt to extend poor scoring region before giving up. The default value is 200.
c	„mincluster” – минимална величина кластера (групације). Подразумијевана вриједност је 65.
d	„diagfactor” – максимални дијагонални чинилац разлике у кластеровању (тј. груписању). Подразумјевана вриједност је 0.12.
e	„reverse” – сравњивање, али обрнутих (под)ниски циљне секвенце са (нормалним) (под)нискама референце. Подразумјевана вриједност је **false**.
f	„forwardAndReverse” – сравњивање, али само нормалних и обрнутих (под)ниски референтне секвенце. Подразумјевана вриједност је **false**.
g	„maxgap” – максимални расцијеп између два сусједна упаривања у кластеру (групацији). Подразумјевана вриједност је 90.
h	„help” – приказује опис и употребу наредбе (програма). Подразумјевна вриједност је **false**.
l	„minmatch” – минимална дужина максималног потпуног поклапања. Подразумјевана вриједност је 20.
m	„mum” – коришћење повезаних поклапања, јединствених и за референцу и за циљну секвенцу. Подразумјевана вриједност је **false**.

Аргумент	Опис
n	„extend” – искључује спољно вански-оријентисано проширење сравњења према из припадајућег кластера (групације); ако је NotExtend = true, биће спријечено сравњивање проширења, али ће ипак сравнити ДНК са кластерованим поклапањима и формирати .delta датотеку. Подразумјевана вриједност је **NotExtend** = **false**.
o	„outputFile” – излазне датотека.
r	„mumreference” – коришћење повезаних поклапања, јединствених за референцу, али не и за циљну секвенцу. Подразумјевана вриједност је **true**.
x	„maxmatch” – коришћење свих поезаних поклапања без обзира на њихову јединственостт. Подразумјевана вриједност је **false**.
v	„verbose” – опширни приказ записника вођеног током саме обраде. Подразумјевана вриједност је **false**.

RepeatResolutionUtil

Употреба RepeatResolution.exe оруђа

Позиционирати циљну секвенцу у односу на референтну, према информацији о сравњивању која се налази у .delta датотеци. Овај услужни програм може искористити информацију о упаривању за позиционирање понављајућих секвенци, или пак за насумично позиционирање којим би се симулирала равномјерна покривеност.

```
RepeatResolutionUtil.exe [options] InputDeltaAlignmentFile InputReadsFile
```

Аргумент	Опис
RepeatResolutionUtil	Команда за рјешавање репетиција
InputDeltaAlignmentFile	Датотека који садржи делта сравњења
InputReadsFile	FastA датотека са очитавањима

Опције

Аргумент	Опис
-h	Приказ описа и употребе команде
-o	Излазна датотека
-v	Опширан приказ записника, вођеног током саме обраде

LayoutRefinementUtil

Употреба LayoutRefinement.exe оруђа

```
LayoutRefinement.exe [options] InputdeltaAlignmentFile
```

Аргумент	Опис
LayoutRefinement	Команда за побољшање диспозиције
InputDeltaalignmentfile	Датотека која садржи делта сравњења

Опције

Аргумент	Опис
-h	Приказ описа и употребе команде
-o	Излазна датотека
-v	Опширан приказ записника, вођеног током саме обраде

ConsensusGenerationUtil

Употреба ConsensusGenerationUtil.exe оруђа

Прочитати информацију о диспозицији из контиг-датотека у којима су назначене позиције очитавања, а потом формирати вишеструка сравњења и/или секвенце усаглашености за њих.

```
ConsensusGenerationUtil.exe [options] InputDeltaAlignmentFile InputReadsFile
```

Аргумент	Опис
Концензус генерације	Команда за формирање контиг-секвенци.
Унос делта поравнања	Датотека која садржи делта сравњења.
Улаз чита датотеку	FastA датотека са очитавањима.

Опције

Аргумент	Опис
-h	Приказ описа и употребе команде
-o	Излазна датотека
-v	Опширан приказ записника, вођеног током саме обраде

ScaffoldUtil

Употреба ScaffodlUtil.exe оруђа

У овом кораку се искориштава излаз из претходног корака (формирање секвенце усаглашености) за улазни скуп података – ContigFile.

```
ScaffoldUtil.exe [options] ContigFile ReadsFile
```

Аргумент	Опис
ScaffoldUtil	Команда за формирање суперконтига на основу mate-pair информације.
ContigFile	FastA датотека са формираним контиг-секвенцама.
ReadsFile	FastA датотека са очитавањима.

Потребни параметри

Потребни параметри	Опис
-k:<int>	Дужина k-mer-a.

Опције

Аргумент	Опис
-d	Досег у обиласку графа. Подразумијевана вриједност је 10.
-h	Приказ информације о коришћењу команде. Подразумијевана је **false**.
-k	Дужина k-mer-a. Подразумијевана вриједност је 10.
-o	Излазна датотека. Подразумијевана је **null**.
-r	Број упарених очитавања, неопходан за повезивање два контига. Подразумијевана вриједност је 2.
-v	Опширан приказ записника, вођеног током саме обраде. Подразумјевана вриједност је **false**.

Рјечник

Дефинисани су неки основни биоинформатички термини, релевантни за сам пројекат. Наравно, није у питању комплетна листа тремина, већ само они који се користе у овом документу.

Састављач

Алгоритми за састављање секвенци се користе за спајање краћих секвенци и очитавања, а све с циљем одређивања оригиналне, односно базне секвенце.

Анотација

Процес дописивања биолошки значајне информације самој секвенци. Обухвата идентификовање елемената генома, процес који се зове генска прогноза, и приписивање биолошке информације овим елементима.

BAM

Бинарни еквивалент SAM-у.

BED

Browser Extensible Display. Формат за датотеке са подацима који описују области секвенци.

Биоинформатика

Дисциплина заснована на математичким, статистичким, и рачунарским техникама анализе ДНК, секвенци аминокиселина, и осталих података који тичу претходно побројаног.

BLAST

Basic Local Alignment Search Tool (BLAST) пореди секвенце нуклеотида или читавих протеина са подацима из своје базе и на основу поређења одређује да ли је упаривање статистички значајно и колико. BLAST се може користити и за одређивање функцијских и еволутивних односа међу секвенцама, као и за идентификацију фамилије гена.

Преломна тачка

Ситуација у којој се сравњење очитавања са референцом састоји од више контигних сегмената или једног сегмента који не обухвата крај очитавања.

Референто састављање

Састављање генома на основу блиско сродног (референтног) генома.

Усаглашеност

Реконструисана секвенца нуклеотида или аминокиселина добијена на основу сравњивања више (под)секвенци. Такође је познат као контиг.

Контиг

Скуп нуклеотида или протеинских секвенци – обично као дио већег молекула – који су већ сравњени и међусобно се поклапају. Нередундантне секвенце настале спајањем једне или више мањих преклапајућих секвенци. Мање секвенце могу бити индивидуална очитавања (трагови) или читаве клониране секвенце. Не би требале да постоје икакви расцјепи у контигу. Број регистрованих контига и спектар њихових величина су важни параметри у самој анализи гена.

ДНК (дезоксирибонуклеинска киселина)

Молекул који се састоји од двоструког ланца нуклеотида, којим је кодирана генетска информација свих живих организама.

ЕБИ (Европски биоинформатички институт)

Истраживачки биоинформатички институт. Једна од институција која обезбјеђује функционисање BLAST-сервиса.

FASTA

FASTA, познат и као фромат Pearson-а, представља организацију података у текстовним датотекама чија је сврха представљање секвенци. Базни парови и аминокиселине су представљени једним знаком (словом), при чему је додатно дозвољено уписивање назива испред саме секвенце, као и додавање коментара.

FASTQ

Текстовни формат за похрањивање података о секвенцама код кога се комбинују FASTA уписи са квалитативним подацима о секвенци.

GFF

Формат датотека за описивање ДНК, РНК, и протеинских секвенци.

Расцјеп

Област објекта за коју није позната одговарајућа секвенца. Генерално су представљени низом слова „N”.

GenBank

GenBak база података је јавно доступна збирка свих јавно познатих секвенци нуклеотидао и њихових протеинских транслација. Смјештена је на сервере NCBI-а, као дио пројекта International Nucleotide Sequence Database Collaboration (INSDC).

Геномика

Дисциплина која се односи на проучавање генетских секвенци.

Хомолози

Специфични умношци ДНК, као када имамо по два умношка свих аутозомалних хромозома – један урађен на основу мајке, а другу на основу оца. Такав пар се назива хомологним.

k-mer

Идентификује област унутар молекула као што су то ДНК молекули.

Дужина N50

Статистика која указује на дужину контига или суперконтига, који систематски премашују просјечну дужину – N50 дужину (максимална дужина за коју се 50% свих базних парова налази у контизима који су ове или пак веће дужине).

NCBI

National Center for Biotechnology Information.

нуклеотид

Основна структурна јединица ДНК и РНК. Називи се, обично, односе на њихове пуринске базе. ДНК користи четири нуклеотида: аденин (A), гуанин (G), тимин (T), цитозин (C). РНК такође користи A, G, и C, али замјењује T са урацилом (У).

Филогенетика

Филогенетско стабло предочава еволуционе везе међу врстама.

Разнообразност

Природне варијације унутар гена, ДНК секвенце, или хромозома.Најчешћи тип разнообразности укључује варијацију на једном пару база.

Полиплидија

Јавља се у ћелијама са више од два упарена (хомологна) скупа хромозома.

Протеин

Молекул – ланац аминокиселина.

РНА (рибонуклеинска киселина)

Једноструки ланац нуклеотида.

Секвенца

Дефинише структуру полимера као што су ДНК, РНК, и протеини.

SAM (мапирање сравњења секвенце)

Формат датотека за похрану сравњења нуклеотида.

Суперконтиг

Нередундантна секвенца настала спајањем више контига.Разлика је у томе што није потребно преклапање секвенци за конструисање веће секвенце. Додатне информације могу да потпомогну везивање. Расцјепи су врло могући.

Shotgun-секвенцирање

Познато и као shotgun-клонирање, метод секвенцирања дугачких ДНК ланаца. ДНК се насумично цијепа на велики број мањих сегмената, који се секвенцирају методом ланчане терминације (ради генерисања очитавања). Вишеструка преклапања очитавања циљне ДНК се добијају понављањем овог поступка у неколико рунди.

SNP (1-нуклеотидна разнообразност)

Ставке представљају варијације између врста или упарених хромозома.

Synteny

Ситуација када се два или више гена налазе на истом хромозому, без обзира да ли постоји доказана веза између њих.

.NET Bio Sequence Assembler: водич за кориснике

Верзија 1.0 - јул 2011.

Сажетак

.NET Bio Sequence Assembler представља демонстрацију могућности .NET Bio Framework-а по питању развоја апликација за биоинформатича истраживања. .NET Bio Sequence Assembler користи елементе корисничког прочеља (UI) да омогући рад и приказ геномских података. Конкретно, од .NET Bio Framework-а је преузео:

колекцију парсера за стандардне формате геномских датотека.

колекцију алгоритама за сравњивање и/или састављање ДНК, РНК, или протеинских ланаца.

колекцију споница на неколико различитих Basic Local Alignment Search Tool (BLAST) веб-сервиса за геномску идентификацију .

Извјештаји BLAST сервиса се могу преузети као једнолинијска знаковна ниска или пак могу бити интегрисани, као компонента, у Queensland University SilverMap приказу.

Веб-адреса за преузимање .NET Bio Sequence Assembler-а је

http://bio.codeplex.com/.

Увод

.NET Bio Sequence Assembler је намијењен биолозима или лаборантима чији посао захтјева рад са новом генерацијом геномских података (сравњивање, састављање, и/или BLAST идентификација). Иако и многе друге апликације обезбјеђују сличну функционалност, циљ .NET Bio Sequence Assembler-а је да у ствари демонстрира могућности .NET Bio Framework-а.

Богато корисничког прочеља (UI) засновано на Windows Presentation Foundation, обезбјеђује .NET Bio Sequence Assembler-у јединствену комбинацију напредног апликацијског прочеља, команди и визуелизације података. Тиме се настоји бити од користи научницима, истраживачима, па и здравственим радницима, који раде са геномским подацима, а што је још важније .NET Bio Sequence Assembler је примјер потпуно функционалане апликације која се према захтјевима може додатно проширивати и дорађивати.

Користећи .NET Bio Sequence Assembler можете:

- **учитавати датотеке са подацима о секвенцама; подржани формати датотека су**:

  FASTA GenBank
  FASTQ GFF

- **сравњивати цијеле секвенце, или само њихове дијелове;** подржани алгоритми су:

  MUMmer 3.0 Pairwise-Overlap (Reference Implementation)
  Needleman-Wunsch PAMSAM
  NUCmer 3.0 Smith-Waterman

- **саставити приказ свеквенце усаглашености на основу сравњених секвенци**
- **слати приказ секвенце усаглашености биолошким веб-сервисима на идентификацију;**
  подржани биолошки веб-сервиси су:

  NCBI QBLAST
  EBI WU-BLAST

- **приказати резултате BLAST упита/захтјева као једнолинијске знаковне ниске или пак као дио SilverMap приказа.**

За више информација погледајте одговарајуће документе у документацијском директоријуму:

Bio\Doc: An Overview [.NET Bio Overview.docx]
Programming Guide [.NET Bio Programming_Guide.docx]

Погледајте такође Додатак А: „Подржане секвенце и формати датотека.”

Инсталација .NET Bio Sequence Assembler-а

Овдје су описание предуслови, системски захтјеви и кораци при инсталацији .NET Bio Sequence Assembler-а.

Предуслови

Да би могли користити .NET Bio Framework-а, требате посједовати елементарно познавање геномичких и биоинформатичких метода и номенклатуре.

За проширивање функционалности апликације потребна су:

- основна знања о рачунарском програмирању
- познавање Microsoft Visual Studio®-а ради (до)програмирања .NET Framework апликација на језику C#.
- посједовање елементарног знања о програмирању оријентисаном ка веб-сервисима.

Системски захтјеви

- Windows® XP Service Pack (SP) 3 или новије верзије Windows-а
- Microsoft .NET Framework Version 4.0, доступна на

  http://www.microsoft.com/downloads/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992.

Инсталација

Инсталер за .NET Bio Sequence Assembler је BioSequenceAssembler.msi, доступан на веб-адреси http://bio.codeplex.com/.

Кораци инсталације .NET Bio Sequence Assembler-а

1. Умножити .NET Bio Sequence Assembler под неким директоријумом на магнетном диску.
2. Отворити прозор претходно споменутог директоријума, а затим два пута кликнути на **BioSequenceAssembler.msi**, који покреће тзв. чаробњака за инсталацију.
3. Слиједити упутства чаробњака.

Инсталатер умножава датотеке под директоријумом C:\Program Files\.NET Bio\1.0\Tools\.NET Bio Sequence Assembler

Послије инсталације .NET Bio Sequence Assembler се појављује на списку **Add and Remove Programs** (Додај и уклони програме) као „.NET Bio Sequence Assembler.”

Преглед корисничког прочеља (UI)

.NET Bio Sequence Assembler има три прошириве картице и ленту са изборницима, као што је приказано на Слици 1:

- Лента садржи сљедеће изборнике:
  - **File**: за отварање и снимање датотека са подацима секвенци.
  - **Options**: за бојење секвенци и промјену одговарајућих формата датотека.

- **Help**: није функционалан у овој верзији.

- **Sequences картица**: за састављања и сравњивање секвенци.
- **Contigs картица**: приказује контиге састављених секвенци.
- **Blast** картица: приказује резултате BLAST упита/захтјева као једнолинијску знаковну ниску или као дио SilverMap приказа.

Слика 1. .NET Bio Sequence Assembler

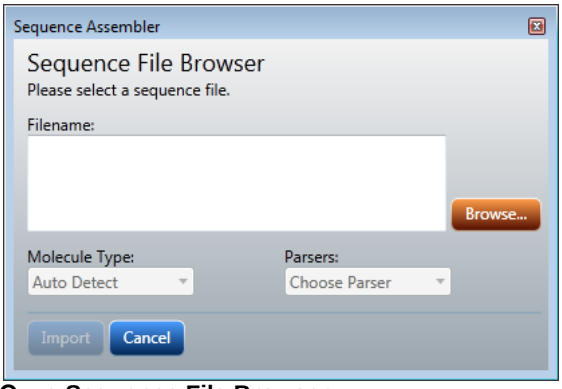
Уношење података о секвенцама

.NET Bio Sequence Assembler подржава сљедеће формате записа геномских података:

- ДНК, РНК или протеинске секвенце: FASTA, FASTQ, и GenBank формате
- Метаподатке о секвенцама: GFF формат

Кораци за унос података о секвенцама

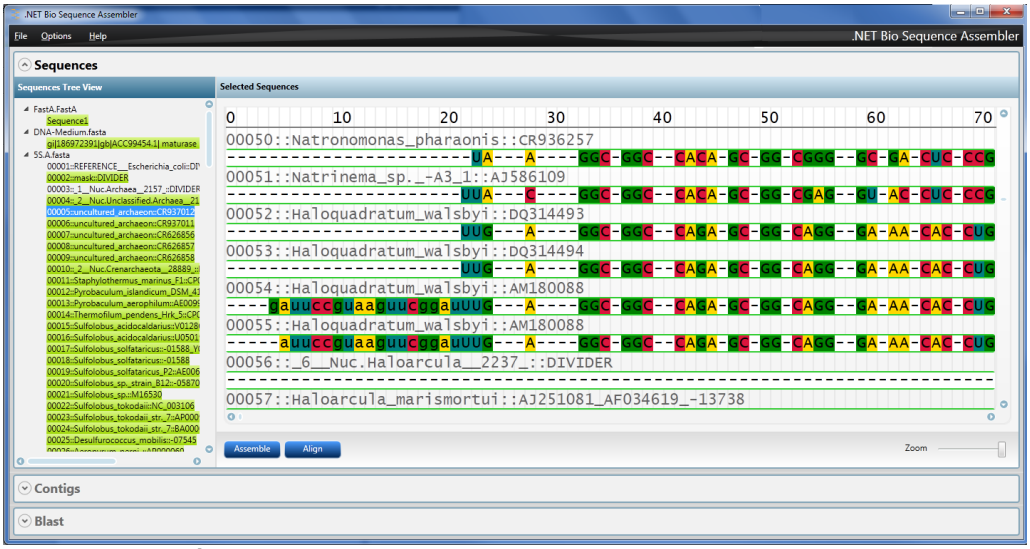
1. Са субдиректоријума C:\Program Files\.NET Bio\1.0\Tools\.NET Bio Sequence Assembler покренути BioSequenceAssembler.exe.
2. Кликнути на **File**, а потом на **Open**.
3. Кликнути на **Browse** у **Sequence File Browser-у**, као што је приказано на сљедећој слици.



Окно Sequence File Browser

4. Потражити датотеку секвенци изабраног формата, а потом је маркирати.
5. Изабрати формат секвенце из падајућег изборника **Parsers**, па кликнути на **Import**.

   Доступне опције су **FASTA**, **FASTQ**, **Genbank** и **GFF**. Сљедећа слика приказује резултате уноса двије FASTA секвенце.



Sequence Tree View

Избацивање секвенце

- Десни клик на секвенцу у **Sequence Tree View-у па клик на Unload у искачућем изборнику**.

Сравњивање секвенци

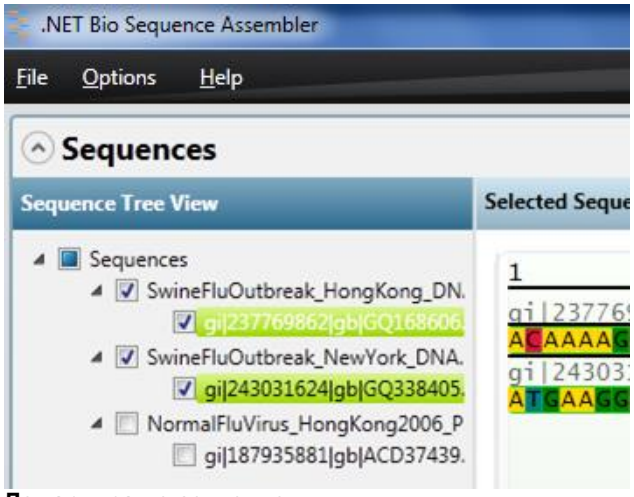
ДНК, РНК и протеинске секвенце се могу сравњивати сљедећим методима:

MUMmer 3.0
Needleman-Munsch
NUCmer 3.0
PAMSAM
Pairwise-Overlap (Reference Implementation)
Smith-Waterman

По одабиру алгоритма и секвенци, подешавате параметре за сравњивање, који укључују и одабир матрице сличности. Резултат сравњивања је приказан на **Contigs** картици .NET Bio Sequence Assembler-а.

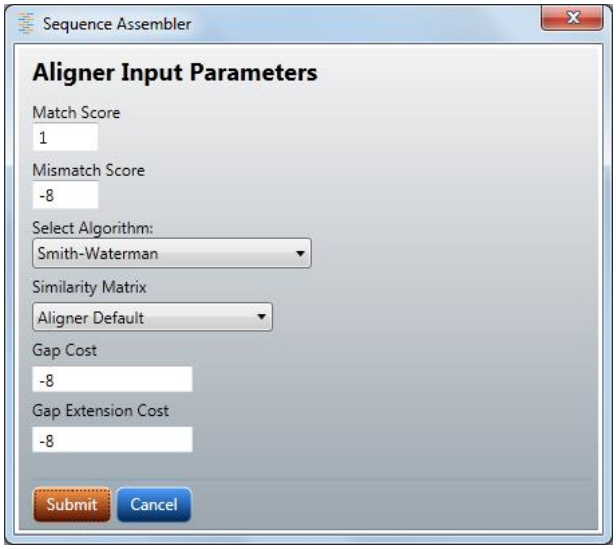
Кораци сравњивања секвенци

1. Унијети двије или више секвенци истог типа, као што је описано у „Уношење података о секвенцама”. Секвенце су већ по учитавању маркиране.
2. На **Sequences** картици уклонити квачице поред било које секвенце коју желите да изоставите из сравњивања, као што је приказано на сљедећој слици.



Демаркирање секвенце

3. На **Sequences** картици, кликнути на **Align** и подесити **Aligner Input Parameters** као што је приказано на сљедећој слици.



Aligner Input Parameters

4. Кликнути на **Submit**. Резултати су приказани на **Alignment** картици као на сљедећој слици.

Поравнате секвенце

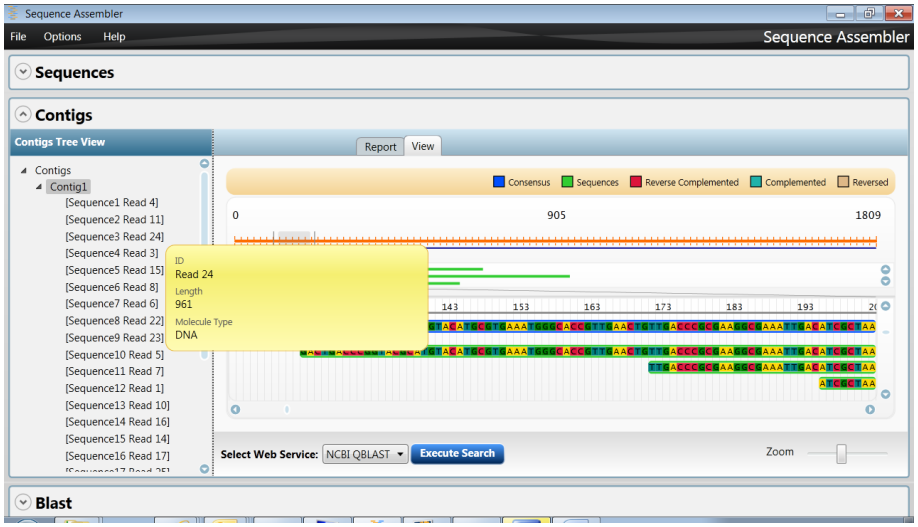
Кораци током састављања секвенце

1. Подесити двије или више секвенци као што је описано у дијелу у претходном дијелу.
2. На **Sequences** картици кликнути на **Assemble**.

   Резултат је приказан на **Consensus** картици, као на сљедећој слици.

Consensus картица

3. Ако контиг постоји, приказан је на **Consensus** картици. Можете прећи мишем изнад секвенце у контигу да се виде њени подаци у балону, као на слици.



Искачући прозор за контиг секвенцу

4. За похрану контига, десни клик на **Consensus** картицу, а потом **Save**.

Слање приказа секвенце усаглашености BLAST сервису

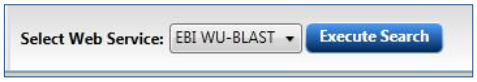
Можете користити .NET Bio Sequence Assembler за слање приказа секвенце усаглашености сљедећим биолошким веб-сервисима на провјеру:

EBI WU-BLAST
NCBI QBLAST

Послије одабира секвенце и избора сервиса, постављате параметре упита/захтјева, који су различити за сваки тип услуге. Резултати су приказани у **Web Service** картици.

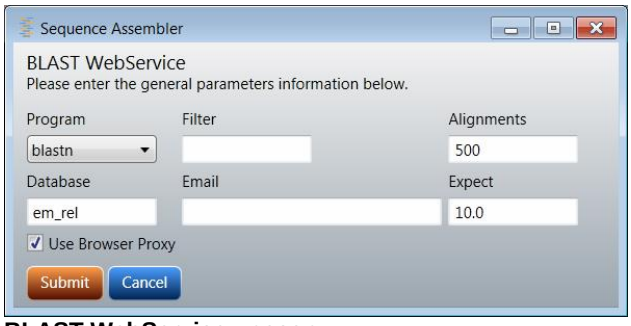
Кораци за слање приказа секвенце усаглашености BLAST сервису

1. Означити услугу користећи **Select Web service** као што је приказано на сљедећој слици, а онда кликнути на **Execute Search**.



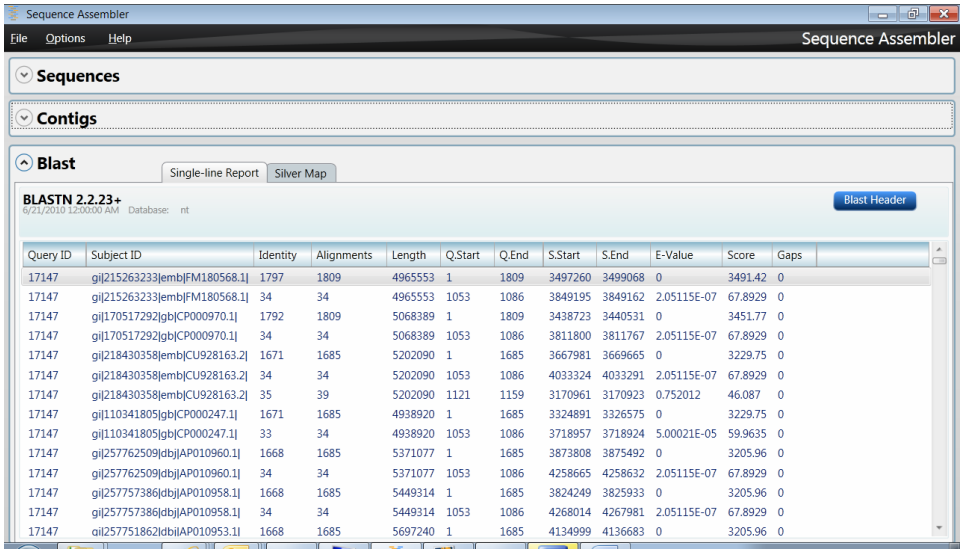
Select Web service падајући изборник

2. Подесити параматре упита/захтјева у **BLAST WebService** прозору, као што је приказано на сљедећој слици, а потом кликнути на **Submit**.

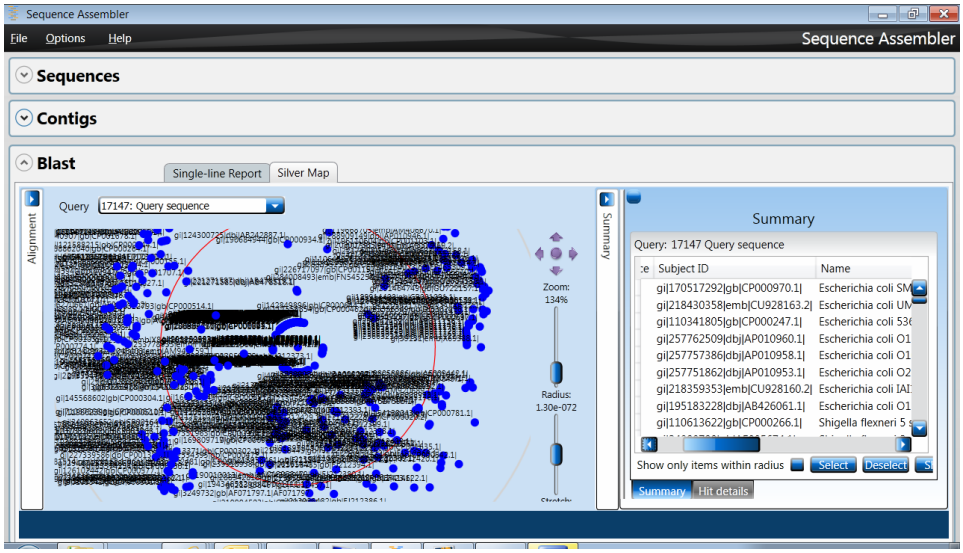


BLAST WebService прозор

Резултати су приказани у **Blast pane** картици као једнолинијске знаковне ниске или у SilverMap приказу, као на слици.



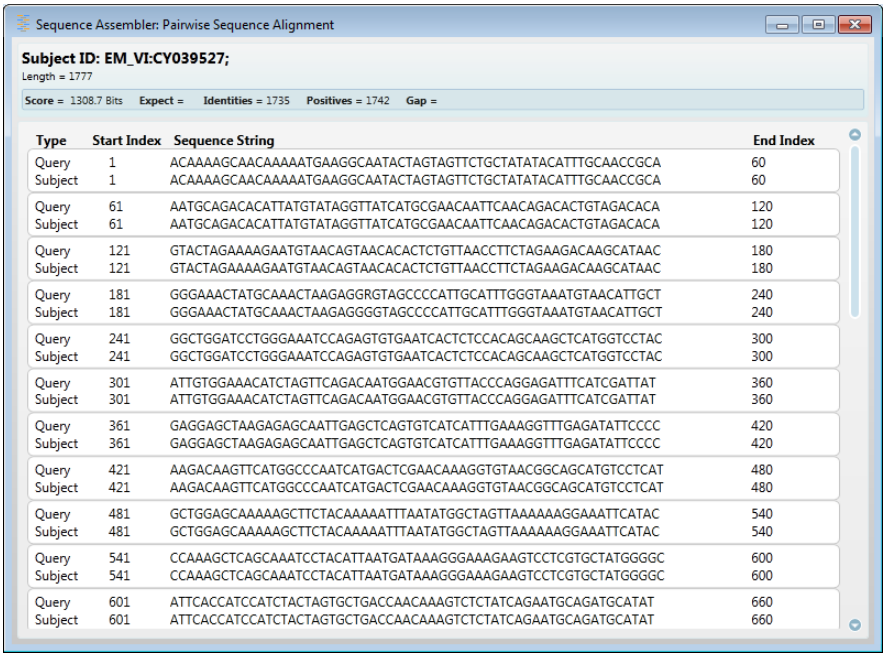
Једнолинијски извјештај BLAST сервиса



SilverMap приказ

За информације о раду са SilverMap приказом, погледајте http://qutbio.codeplex.com/.

3. За приказ 2-по-2 сравњивања секвенци, кад је овај ID у питању, и У једнолинијском извјештају, два пута кликнути на било који субјекат ID-а.



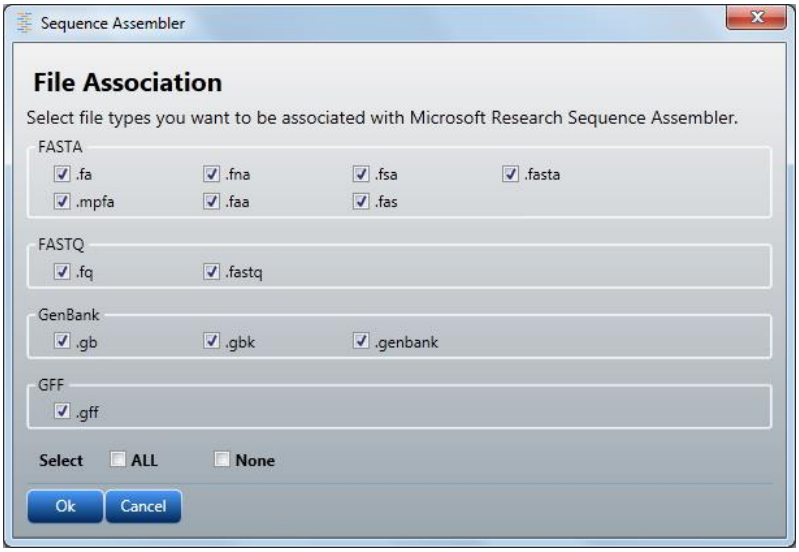
2-по-2 сравњивање секвенци

Конфигурисање .NET Bio Sequence Assembler-а

.NET Bio Sequence Assembler има двије конфигурацијске опције: повезивање датотечних формата са .NET Bio Sequence Assembler-ом и образац за бојење података о секвенцама.

Кораци за повезивање датотечних формата са .NET Bio Sequence Assembler-ом

1. У **Options** изборнику кликнути на **Associate File Types**.

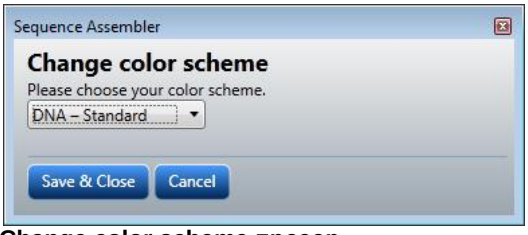


File Association прозор

2. У **File Association** прозору одабрати датотечне формате ради њиховог повезивања са .NET Bio Sequence Assembler-ом, а потом кликнути на **OK**.

Кораци у конфигурисању обрасца за бојење

1. Кликнути на **Change Colors** или на **Sequence View** картици или на **Consensus View** картици.
2. Означити **DNA – Standard** или **Protein – Standard** у **Change color scheme** прозору, као што је приказано на сљедећој слици.



Change color scheme прозор

3. Кликнути на **Save & Close** да снимите промјене.

Кораци за формирање сопственог обрасца за бојење

1. Отворити Sequenceassembler.exe.config датотеку која се налази под C:\Program Files\NET Bio\1.0\Tools\.NET Bio Sequence Assembler директоријумом.
2. У **<Colors>** дијелу промијенити вриједности боја за један или више симбола користећи уобичајене називе боја.
3. Додати нови образац формирањем новог **<ColorScheme>** одјељка. Увјерити се да су форматирање и структура постојећег **<ColorScheme>** одјељка у потпуности дуплирани.
4. Снимити Sequenceassembler.exe.config датотеку.

Listing 1. The Sequenceassembler.exe.config file

```
<?xml version="1.0"?>
<configuration>
  <configSections>
    <section name="Colors" type="SequenceAssembler.ColorSchemeConfigHandler,
SequenceAssembler"/>
  </configSections>

    <Colors>
      <!--Represents the DNA\RNA color scheme-->
      <ColorScheme Name="DNA - Standard">
        <Symbol Char="A" Color="Red"/>
        <Symbol Char="T" Color="Red"/>
        <Symbol Char="G" Color="Red"/>
        <Symbol Char="C" Color="Red"/>
        <Symbol Char="U" Color="Red"/>
        <Symbol Char="-" Color="Red"/>
        <Default Color="Red"/>
      </ColorScheme>

      <!--Represents the Protein color scheme-->
```

```
      <ColorScheme Name="Protein - Standard">
        <Symbol Char="A" Color="Blue"/>
        <Symbol Char="R" Color="Blue"/>
        <Symbol Char="N" Color="Blue"/>
        <Symbol Char="D" Color="Blue"/>
        <Symbol Char="C" Color="Blue"/>
        <Symbol Char="E" Color="Blue"/>
        <Symbol Char="Q" Color="Blue"/>
        <Symbol Char="G" Color="Blue"/>
        <Symbol Char="H" Color="Blue"/>
        <Symbol Char="I" Color="Blue"/>
        <Symbol Char="L" Color="Blue"/>
        <Symbol Char="K" Color="Blue"/>
        <Symbol Char="M" Color="Blue"/>
        <Symbol Char="F" Color="Blue"/>
        <Symbol Char="S" Color="Blue"/>
        <Symbol Char="T" Color="Blue"/>
        <Symbol Char="W" Color="Blue"/>
        <Symbol Char="Y" Color="Blue"/>
        <Symbol Char="V" Color="Blue"/>
        <Symbol Char="-" Color="Blue"/>
        <Default Color="Blue"/>
      </ColorScheme>
    </Colors>
<startup><supportedRuntime version="v4.0"
sku=".NETFramework,Version=v4.0"/></startup></configuration>
```

Додатак A: Подржани формати датотека

Овај додатак описује формате подржане у .NET Bio Framework-у, са хипервезама ка референцама и изворима за више информација.

FASTA: Текстовне ниске података

Текстовни формат за представљање пептидних или нуклеотидних ланаца, веома погодан за обраду на програмским језицима као што је Iron Python.

Технички, формат је низ линија. Најчешће има 80 слова по линији, али никако не више од 120.

Детаљан опис

FASTA формат спецификација
http://www.ncbi.nlm.nih.gov/blast/fasta.shtml

Извор

Преглед, хипервезе на конвертере и остале битне референце са Википедијине странице: http://en.wikipedia.org/wiki/FASTA_format.

FASTQ: Квалитативне ниске података

Текстовни формат који похрањује биолошке секвенце и Phred quality бодове у једну датотеку. Често се сматра *de facto* стандардом за похрану хеуристичких и бодовних података високо-пропусног анализатора (High-Troughput Computing).

Формат је дефинисан тако да четири линије датотеке чине слог (који одговара једној геномској нисци).

Уобичајени додаци датотекама укључују **.fq**, **.fastq**, **.txt**.

Детаљан опис

FASTQ формат спецификација
http://maq.sourceforge.net/fastq.shtml

Извор

Преглед, хипервезе на конвертере и остале битне референце са Википедијине странице: http://en.wikipedia.org/wiki/FASTQ_format

GenBank: Формат за геномске базе података

Flat-file формат који описује нуклеотиде и секвенце нуклеотида из GenBank-а – јавне базе података.

Детаљан опис

"Chapter 1, GenBank: The Nucleotide Sequence Database," Ilene Mizrachi; *NCBI Handbook*, 2007

http://www.ncbi.nlm.nih.gov/books/bookres.fcgi/handbook/ch1.pdf

Извори

GenBank на NCBI веб-страници базе података
http://www.ncbi.nlm.nih.gov/sites/entrez?db=nucleotide

Преглед, хипервезе на конвертере и остале битне референце са Википедијине странице: http://en.wikipedia.org/wiki/GenBank

GFF: Generic Feature Format

Линијски формат, при чему су поља раздвојена tab-размацима. Намијењен је за представљања слогова у геномској бази података. GFF слог представља субсеквенцу, као што је то на примјер ген или протеинска секвенца, истовремено дозвољавајући „умјерено детаљна" образложења.

Екстензија оваквих датотека је **.gff**.

Раније спецификације су преводиле скраћеницу као Gene-Finding Format.

Детаљан опис

Садашња верзија је n2. Формат су првобитно осмислили Richard Durbin и David Haussler, а посљедња верзија садржи измјене које су предложили Lincoln Stein, Suzanna Lewis, Anders Krogh и други.
http://www.sanger.ac.uk/resources/software/gff/spec.html

Извори

Сајт института Wellcome Trust Sanger за општи преглед формата
http://www.sanger.ac.uk/Software/formats/GFF/GFF_Spec.shtml

UCSC сајт пројекта Encode, такође за општи преглед формата
http://genome.ucsc.edu/goldenPath/help/customTrack.html#GFF

.NET Bio Extension for Excel: водич за кориснике

Сажетак

Описана је употреба додатка .NET Bio Extension for Excel за Microsoft® Office Excel 2007 и Excel 2010 који омогућује једноставан и елегантан рад са геномским секвенцама, метаподацима и интервалним подацима у Excel-документу.

Неке од могућности .NET Bio Extension-а су засноване на радном оквиру .NET Bio Framework, као на примјер: скуп парсера за уобичајене формате геномских датотека, скуп секвенционих алгоритама за формирање секвенце усаглашености ДНК, скуп (софтверских) спојница на неколико Basic Local Alignment Search Tool (BLAST) веб-сервиса за геномску идентификацију, као и интервалне геномске функције које омогућују коришћење BED датотека у Excel-у. Сам додатак .NET Bio Extension може се и програмски проширити ради искоришћавања неких додатних опција/функција NET Bio Framework-а.

Веб-адреса за преузимање .NET Bio Extension-а је http://bio.codeplex.com/.

Увод

.NET Bio Extension for Excel је намијењен биолозима који су заинтересовани или већ користе Microsoft Office Excel у својим истраживањима. Флексибилност MS Excel-а омогућује програмирање и уграђивање модула посебне намјене, а и формирање одговарајућег прочеља за модуле – картице на ленти – у овом случају „.NET Bio". Додатак, сам по себи, не није у конфликту са остатком Excel-а, тј. све уобичајене функције остају на располагању, при чему је биолозима мало повећан арсенал, биоинформатички-усмјерен.

Могуће додатно проширивати .NET Bio Extension користећи, као што је већ споменуто, .NET Bio Framework, али и неки други биониформатички радни оквир. На примјер, могуће је испрограмирати специфичне биоинформатичке апликације користећи библиотеке .NET Bio Framework-а и потом додати корисничком прочељу Biology Extension-а одговарајуће графичке елементе.

Генерално, Biology Extension омогућује:

- **Коришћење датотека које садрже податке о геномским секвенцама. Подржани формати датотека су**:

  BED GenBank
  FASTA GFF
  FASTQ

- **Сравњивање читавих геномских секвенци, или пак њихових дијелова.** Подржани алгоритми су:

  MUMmer 3.0 Pairwise-Overlap (Reference Implementation)
  Needleman-Wunsch Smith-Waterman
  NUCmer 3.0

- **Формирање приказа секвенце усаглашености послије вишеструког сравњивања геномских секвенци.**

- **Слање приказа секвенце усаглашености биолошким веб-сервисима на идентификацију.**
  Подржани сервиси су:

  NCBI QBLAST
  EBI WU-BLAST

- **Упис геномских секвенци у датотеке, као и њихову измјену.** Подржани формати датотека за упис су:

  FASTA
  FASTQ
  GenBank

- **Руковање и приказ података о геномским секвенцама:**
  - Руковање интервалним геномским подацима у формату UCSB BED.
  - Извршавање операција као што су Merge (унија), Intersect (пресјек) и Subtract (разлика) на интервалним геномским подацима.

- Формирање графикона са подацима о геномским секвенцама.
- Приказ Венових дијаграма интервалних геномских података користећи NodeXL за Excel.

За више информација, консултовати документацију .NET Bio Framework-а:

Bio\Doc: An Overview [.NET Bio Overview.docx]
Programming Guide [.NET Bio Programming_Guide.docx]

Такође, погледати и Прилог А – „Подржане секвенце и формати датотека"

Инсталација .NET Bio Extension-а

На почетку су описани предуслови, системски захтјеви и кораци за поставку (инсталацију) .NET Bio Extension-а.

Предуслови

Потребна су елементарна знања о:

- избору опсега ћелија у Excel-у

геномичким и биоинформатичким методама и номенклатури.

Системски захтјеви

.NET Bio Extension се може поставити на сваки рачунар који може покренути Microsoft Office 2007, што је сажето у чланку на адреси http://office.microsoft.com/en-us/products/HA101668651033.aspx

На самом рачунару се мора налазити сљедећи софтвер:

- било која верзија оперативног система Windows® који може покренути Office 2007, укључујући Windows XP Service Pack (SP) 3 и све новије верзије Windows-а
- Microsoft Office Excel 2007 или Excel 2010
- NodeXL шаблон за Excel, доступан на http://www.codeplex.com/NodeXL
- Microsoft .NET Framework Version 4.0, доступан на http://www.microsoft.com/downloads/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992

Инсталација

Програм за поставку .NET Bio Extension-а је доступан на веб-адреси http://bio.codeplex.com/.

Након успјешне поставке .NET Bio Extension се појављује у **Add and Remove Programs** као „.NET Bio Extension for Excel."

Инсталација .NET Bio Extension for Excel-а

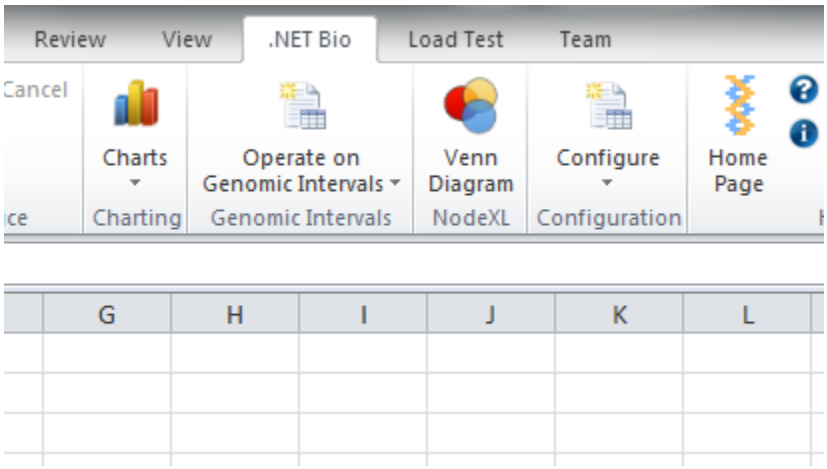
1. Затворити све Excel-документе.
2. Умножити .NET Bio Extension for Excel инсталатер на неком од директоријума (магнетног диска).
3. Двоклик на **BioExcel.msi**, што ће покренути тзв. чаробњака (који ће вас водити кроз остатак инсталације).

4. Пратити упутства чаробњака ради завршетка процеса инсталације Biology Extension-а.

Потврда успјешности инсталације

- Покренути Excel 2007 или Excel 2010.

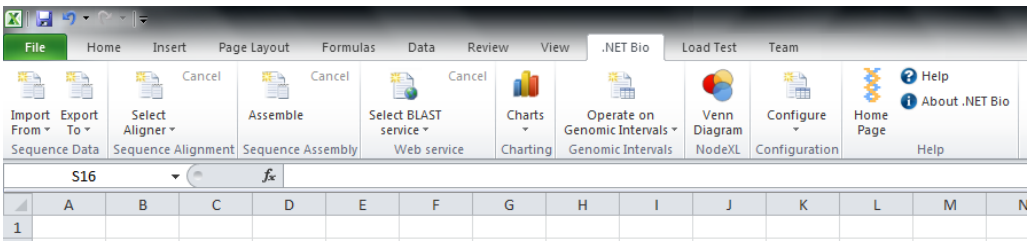
Лента би сада требала да садржи **.NET Bio** картицу, као што је приказано на Слици 1.



Слика 1. Лента Екселa са .NET Bio картицом.

Преглед корисничког прочеља

.NET Bio картица, као што је то приказано на Слици 2, садржи главно корисничко прочеље за Biology Extension.



Слика 2. .NET Bio картица

.NET Bio картица садржи седам група наредби:

Sequence Data

Убацивање и дистрибуција геномских секвенци користећи (подржане) формате датотека (наведени у Прилогу А).

Sequence Alignment

Сравњивање цијелих или пак дјелимичних геномских секвенци подржаним алгоритмима.

Sequence Assembly

Састављање двије или више геномских секвенци у приказ (секвенце) усаглашености.

Web Service
Слање приказа (секвенце) усаглашености веб-сервисима на идентификацију.

Charting
Графички приказ расподјеле нуклеотида у ДНК ланцима.

Genomic Intervals
Руковање интервалним геномским подацима у формату UCSB BED помоћу наредби (операција) као што су Merge (унија), Intersect (пресјек), Subtract (разлика).

NodeXL
Приказ Венових дијаграма на основу сравњења секвенци.

Configuration
Дефинисање сљедећих опција у Biology Extension-у:

промјена ширине колоне-оквира за податке о геномским секвенцама.

измјена шеме за бојење ДНК, РНК и протеинских молекула.

Учитавање датотека

Подржани су сљедеће типови и формати записа геномских података:

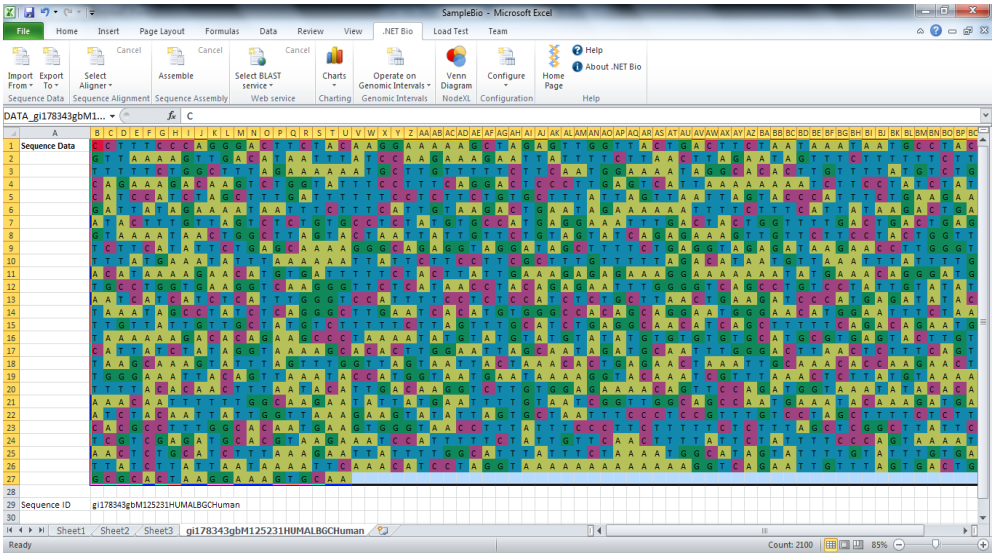
Тип	Формат
ДНК, РНК или протеинске секвенце	FASTA, FASTQ, GenBank
Метаподаци о секвенцама	GFF
Интервални геномски подаци	BED

Подаци се по учитавању датотеке могу преуређивати, мијењати, слати BLAST веб-сервису, или пак уписивати у нове датотеке.

Учитавање ДНК, РНК, или протеинских секвенци

6. Кликнути на **Import From** (Убаците из..) са **.NET Bio** ленте.
7. Одабрати један од формата геномских података: **FASTA**, **FASTQ**, или **GenBank**.
8. Пронаћи и одабрати жељену датотеку изабраног формата.
9. Кликнути на **Open**.

   Секвенца је учитана у нови радни лист, као на сљедећој слици.

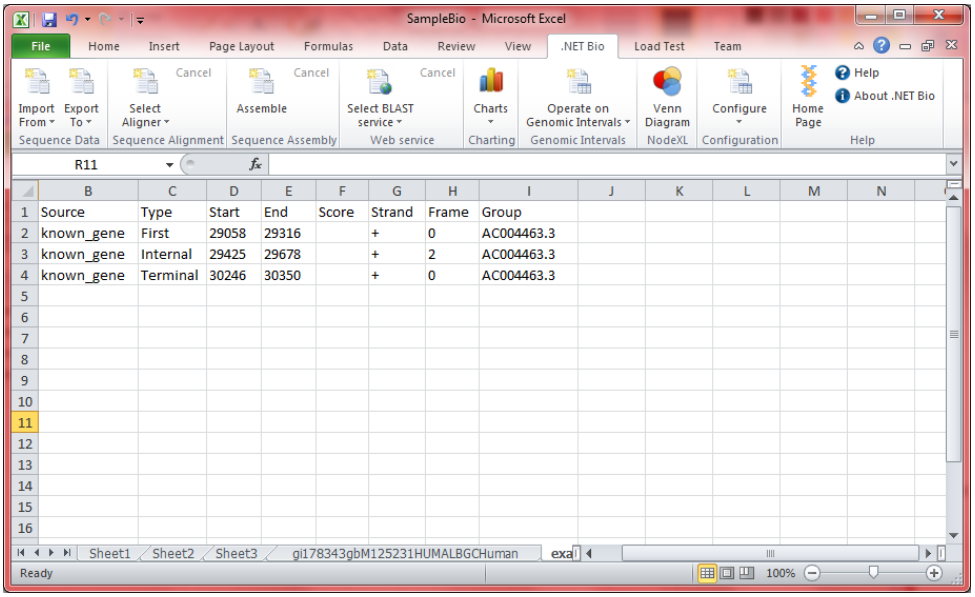


FASTA секвенца података у Excel-у

Учитавање GFF метаподатака о секвенцама

1. Кликнути на **Import From** са **.NET Bio** ленте.
2. Кликнути на **GFF** ради одабира формата GFF.
3. Пронаћи и одабрати жељену датотеку изабраног формата.
4. Кликнути на **Open**.

   GFF метаподаци су учитани у нови радни лист, као на сљедећој слици.



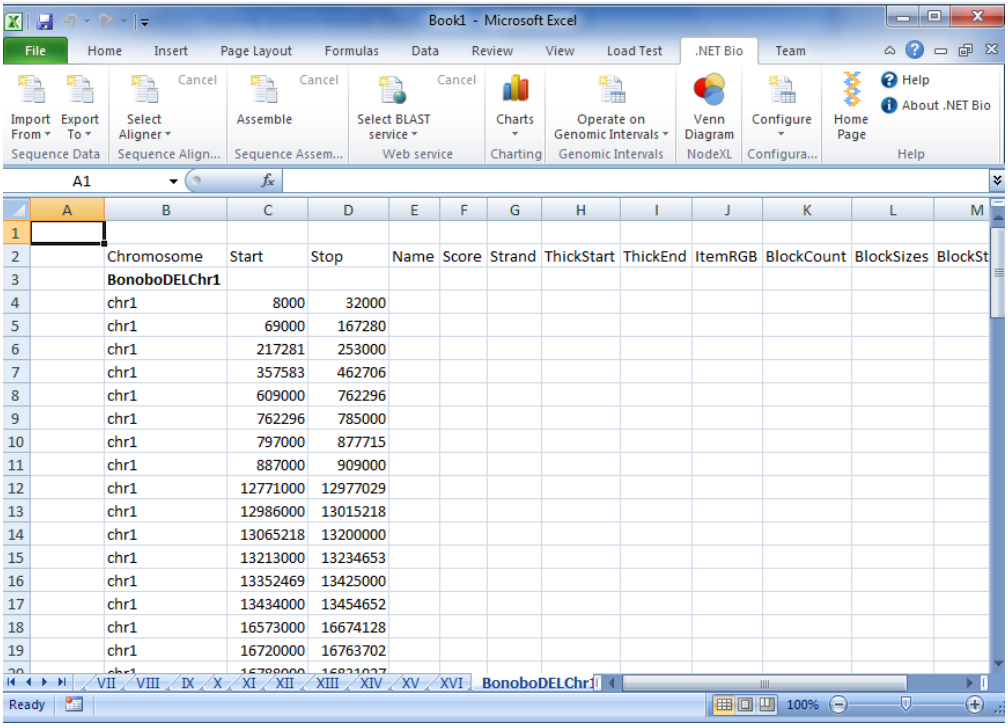
GFF метаподаци о секвенцама у Excel-у.

Учитавање BED интервалних геномских података

1. Кликнути на **Import From** са **.NET Bio** ленте.
2. Одабрати формат BED**.**
3. Пронаћи и одабрати датотеку са интервалним подацима изабраног формата.

4. Кликнути на **Open**.

   Интервални подаци су учитани у нови радни лист, као на сљедећој слици.



BED interval data in Excel

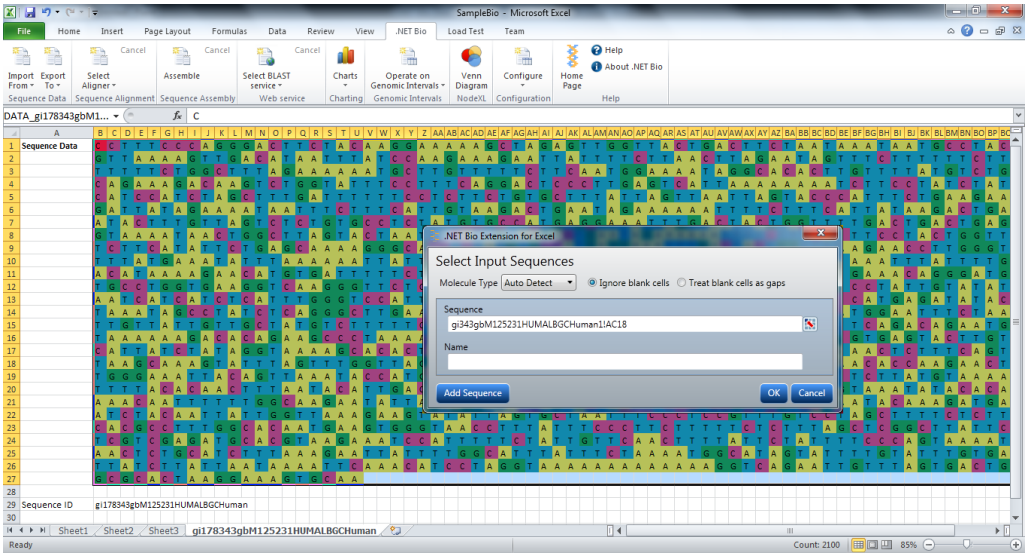
Упис у датотеку

.NET Bio Extension подржава упис ДНК, РНК или протеинских секвенци у датотеке FASTA, FASTQ и GenBank формата.

Након уноса података, могуће их је мијењати, сравњивати, слати на BLAST веб-сервис, или пак на основу њих формирати неке друге датотеке.

Уписивање ДНК, РНК или протеинских секвенци у датотеку

1. Кликнути на **Import From** са **.NET Bio** ленте.
2. Кликнути на жељени формат записа секвенци: **FASTA**, **FASTQ** или **Genbank**.
3. Пронаћи и одабрати датотеку изабраног формата.
4. Кликнути на **Open**.
5. Начинити жељене измјене у датотеци и кликнути на **Export To** (Упиши у)**.**
6. Помоћу дијалошког окна **Select Input Sequences** (Изаберите секвенце за упис), као што је приказано на сљедећој слици, одабрати цијелу или пак само дио секвенце и кликнути на **OK**.



Дијалошко окно Select Input Sequences

7. Кликнуте на **Save As** на **File** картици и снимити датотеку под новим називом.

Сравњивање секвенци

ДНК, РНК и протеинске секвенце се могу сравњивати према следећим алгоритмима:

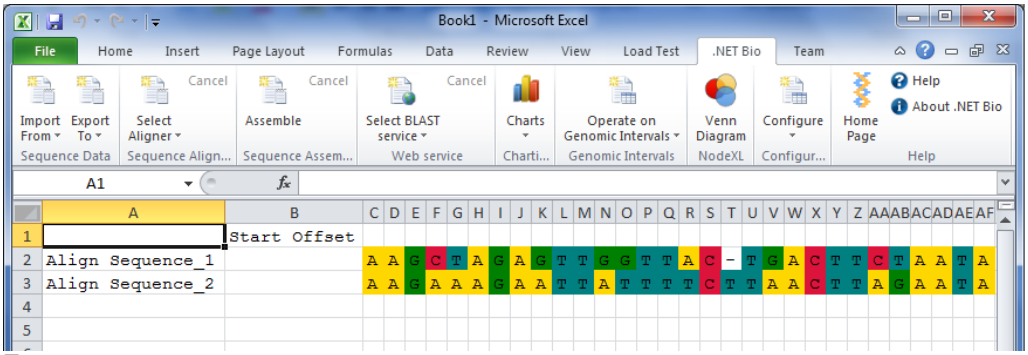
MUMmer 3.0 Pairwise-Overlap
Needleman-Wunsch Smith-Waterman
NUCmer 3.0

По одабиру алгоритма и двију или више секвенци, подешавате параметре за сравњивање, а који укључују и одабир матрице сличности. Резултат сравњивања бива приказан у новом радном листу.

Сравњивање секвенци

5. Убацити двије или више секвенци истог типа, као што је то претходно описано у поглављу „Учитавање датотека”.
6. Кликнути на **Select Aligner** (Изаберите поравнавање) са **.NET Bio** ленте.
7. Унутар дијалошког окна **Select Input Sequences** одабрати двије или више секвенци, а потом кликнути на **OK**.
8. Подесити параметре за сравњивање унутар дијалошког окна **Align Inputs Parameters** (Улазни параметри сравњивања) и кликнути на **OK**.

   Резултат сравњивања је приказан на новом радном листу, као на следећој слици.



Поравнате FASTA секвенце

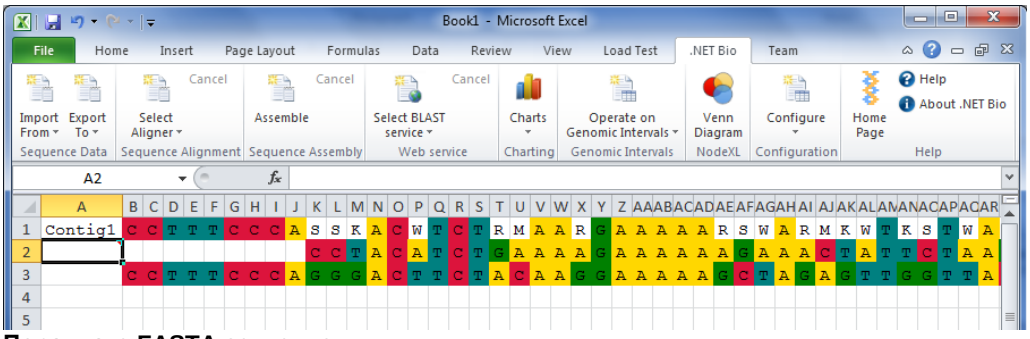
Агрегација секвенци

ДНК, РНК и протеинске секвенце могу бити агрегиране у тзв. приказе (секвенце) усаглашености.

Након одабира двију или више секвенци истог типа, подешавате параметре за агрегацију и бирате алгоритам за сравњивање. Резулат агрегације је приказан на новом радном листу.

Агрегирање секвенци

1. Унијети двије или више секвенци истог типа како је то претходно описано у поглављу „Упис у датотеку”.
2. Кликнути на **Assemble** (Агрегирај) са **.NET Bio** ленте.
3. Додати унешене секвенце користећи дијалошко окно **Select Input Sequences**, а затим кликнути на **OK**.
4. Подесити параметре за сравњивање унутар дијалошког окна **Align Inputs Parameters**, па потом кликнути на **OK**.



Поравнате FASTA секвенце

Слање секвенце BLAST веб-сервисима

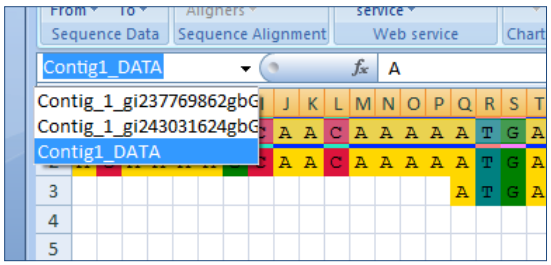
.NET Bio Extension се може искористити, послије вишеструких сравњивања, и за слање на провјеру приказа (секвенце) усаглашености сљедећим биолошким веб-сервисима:

EBI WU-BLAST
NCBI QBLAST

Након одабира цијеле секвенце или пак неког њеног дијелам, као и самог сервиса, подешавате параметре захтјева, који се разликују за сваки тип услуге. Резултат бива приказан на новом радном листу.

Слање цијеле секвенце BLAST-сервису

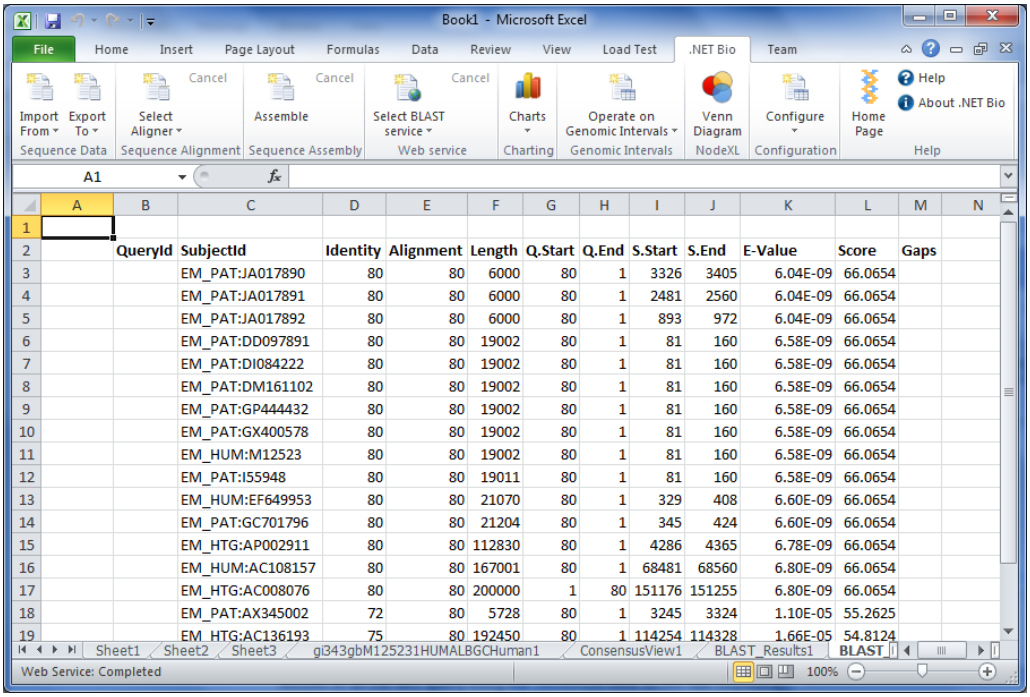
3. Одабрати секвенцу кликом на **Contig1_DATA** у падајућем изборнику **Name Box-a** (Називи поља), као што је приказано на сљедећој слици.



Одабир приказа усаглашености двије секвенце

4. Кликнути на **Select BLAST Service** (Изаберите BLAST-сервис) са **.NET Bio** ленте.
5. Кликнути на одговарајући BLAST-сервис ради формирања захтјева.
6. Подесити параметре захтјева у прозору **BLAST WebService** (BLAST Веб-сервис) и кликнути на **OK**.

   Резултати су приказани у новом радном листу, као на сљедећој слици.



	A	QueryId	SubjectId	Identity	Alignment	Length	Q.Start	Q.End	S.Start	S.End	E-Value	Score	Gaps
3			EM_PAT:JA017890	80	80	6000	80	1	3326	3405	6.04E-09	66.0654	
4			EM_PAT:JA017891	80	80	6000	80	1	2481	2560	6.04E-09	66.0654	
5			EM_PAT:JA017892	80	80	6000	80	1	893	972	6.04E-09	66.0654	
6			EM_PAT:DD097891	80	80	19002	80	1	81	160	6.58E-09	66.0654	
7			EM_PAT:DI084222	80	80	19002	80	1	81	160	6.58E-09	66.0654	
8			EM_PAT:DM161102	80	80	19002	80	1	81	160	6.58E-09	66.0654	
9			EM_PAT:GP444432	80	80	19002	80	1	81	160	6.58E-09	66.0654	
10			EM_PAT:GX400578	80	80	19002	80	1	81	160	6.58E-09	66.0654	
11			EM_HUM:M12523	80	80	19002	80	1	81	160	6.58E-09	66.0654	
12			EM_PAT:I55948	80	80	19011	80	1	81	160	6.58E-09	66.0654	
13			EM_HUM:EF649953	80	80	21070	80	1	329	408	6.60E-09	66.0654	
14			EM_PAT:GC701796	80	80	21204	80	1	345	424	6.60E-09	66.0654	
15			EM_HTG:AP002911	80	80	112830	80	1	4286	4365	6.78E-09	66.0654	
16			EM_HUM:AC108157	80	80	167001	80	1	68481	68560	6.80E-09	66.0654	
17			EM_HTG:AC008076	80	80	200000	1	80	151176	151255	6.80E-09	66.0654	
18			EM_PAT:AX345002	72	80	5728	80	1	3245	3324	1.10E-05	55.2625	
19			EM_HTG:AC136193	75	80	192450	80	1	114254	114328	1.66E-05	54.8124	

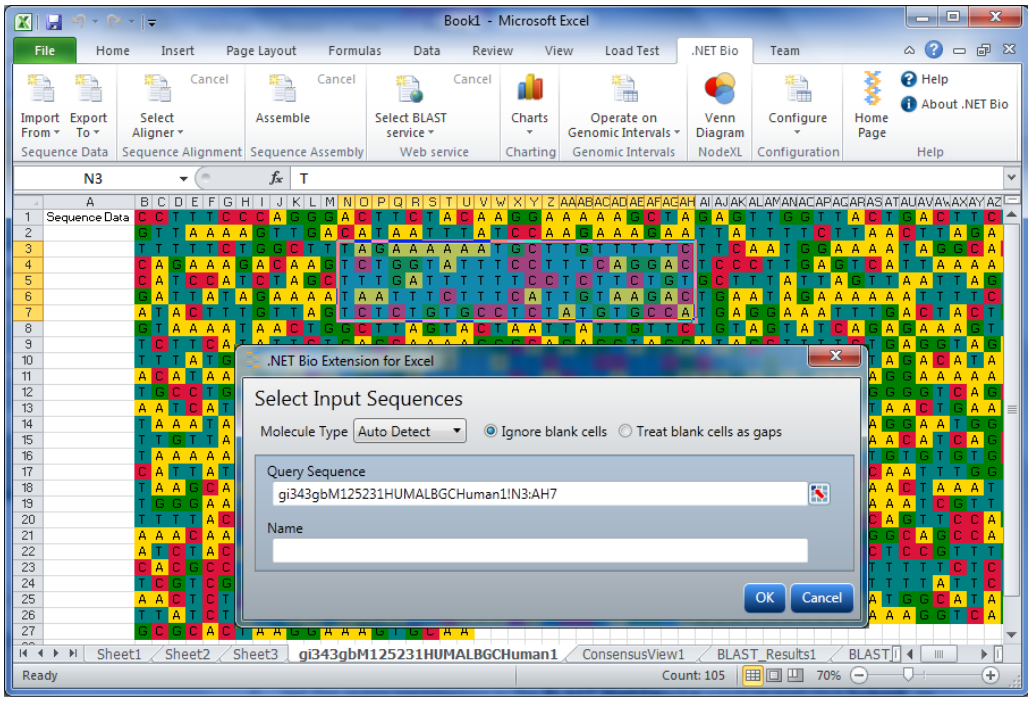
Резултати EBI-WU захтјева користећи приказ (секвенце) усаглашености за двије секвенце

Слање дијела секвенце BLAST-сервису

1. Одабрати показивачем одређена поља секвенце.
2. Кликнути на **Select BLAST Service** у **.NET Bio** ленти и одабрати BLAST-сервис.

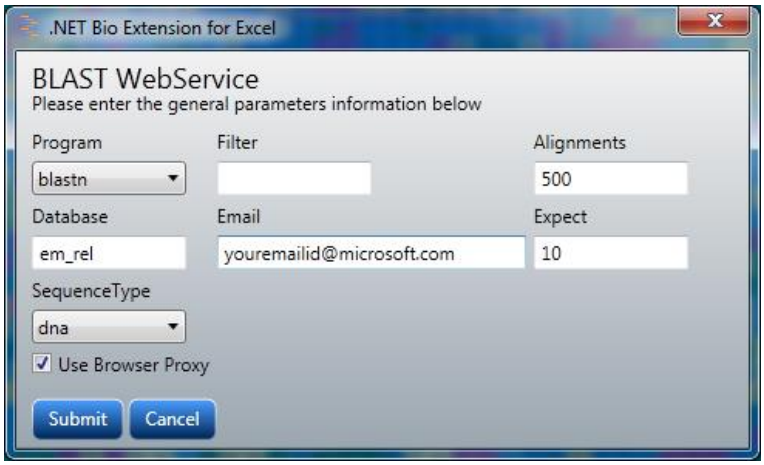
Дијалошко окно **Select Input Sequences** приказује одабир у пољу **Query Sequence** (Секвенца захтјева), као што је приказано на сљедећој слици.

Примијетићете да су посљедњи знаци секвенце захтјева у ствари референце на изабрана поља: **N3:AH7**.



Дијалошко окно Select Input Sequences

3. Подесити параметре захтјева у дијалошком окну **BLAST WebService** и кликнути на **Submit**, као што је приказано на сљедећој слици.

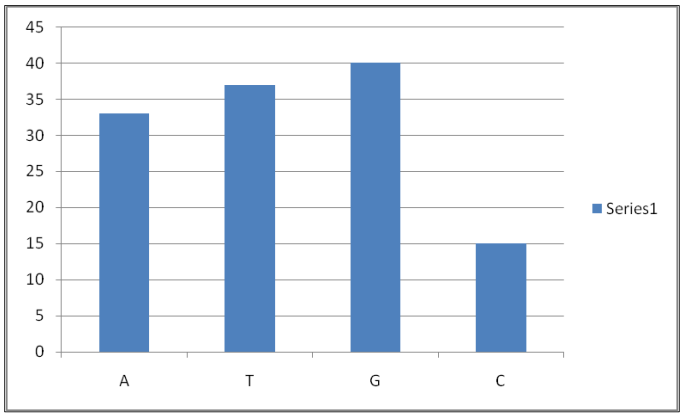


Дијалошко окно BLAST WebService за EBI WU-BLAST

Ако је обрада захтјева успјешна, резултат ће бити приказан на новом радном листу.

Графички приказ расподјеле нуклеотида ДНК

Charting-функцијом, на основу података из секвенце, могуће је формирати графикон расподјеле нуклеотида ДНК, као што је то приказано на Слици 3.



Слика 3. Хистограм расподјеле нуклеотида ДНК

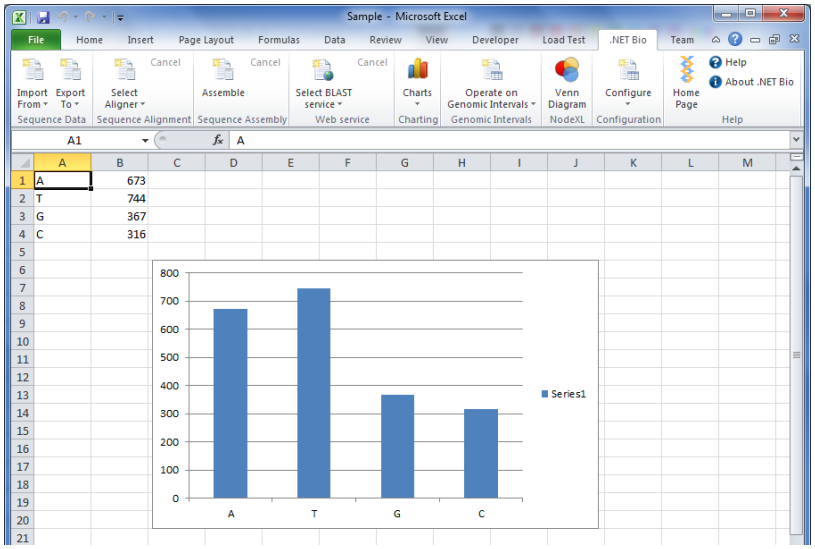
Да би могли користити Charting-функцију, морате омогућити макрое у Excel-у и додати одговарајући Excel-макро радном листу који користите са Biology Extension-ом. Назив макроа је DisplayDNASequenceDistribution.bas и долази уз инсталацију .NET Bio Framework-а.

Важно: слиједите процедуру из Прилога Б „Одобравање макроа" прије коришћења оруђа **Charts** (Графикони).

Формирање графикона за ДНК секвенцу

1. Отворити Excel-ову радну свеску у којој је допуштено коришћење макроа и која садржи макро **DisplayChart** (Прикажи графикон).
2. Одабрати радни лист који садржи податке о секвенци.
3. Кликнути на **Charts** иконицу са **.NET Bio** картице, а затим на **DNA Sequence Distribution Table** (Табела расподјеле ДНК секвенце).

   Графикон је приказан у новом радном листу, као на сљедећој слици.



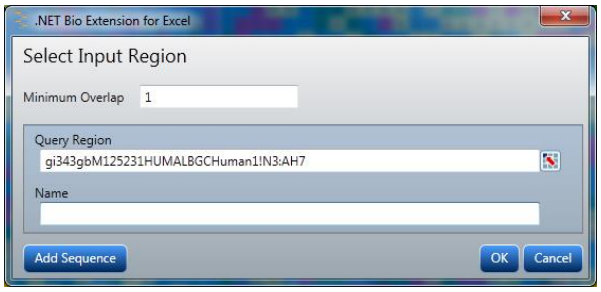
Нови радни лист са графиконом

Руковање са интервалним геномским подацима

Опцијом **Operate on Genomic Intervals** (Рад са геномским интервалима), могуће је извршити три основне (скуповне/интервалне) операције: **Merge** (унија)**, Intersect** (пресјек) и **Subtract** (разлика). Користећи датотеке у формату BED, можете дефинисати један или више захтјева користећи опсеге радног листа, а затим изабрати једну од наведене три операције. Одабрани опсези садрже једну или више базних парова координата хромозома. По обављеној операцији, резултат је исписан на новом радном листу.

Унија преклапајућих интервала

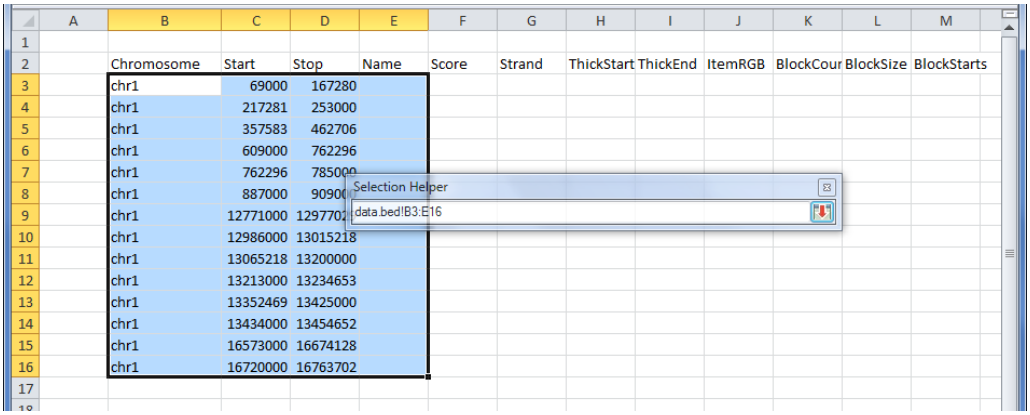
1. Кликнути **Import From**, а потом на **BED**.
2. Одабрати једну или више датотека и кликнути на **Open**.
3. Одабрати радни лист и кликнути на **Operate on Genomic Intervals**.
4. Кликнути на **Мере**. Приказује се дијалошко окно **Select Input Sequences Ranges** као на сљедећој слици.



Окно Select Input Sequence Ranges

5. Кликнути на иконицу за избор, која се налази уз десну страну поља **Reference Sequence** (Референтна секвенца), а потом изабрати опсег базних парова.

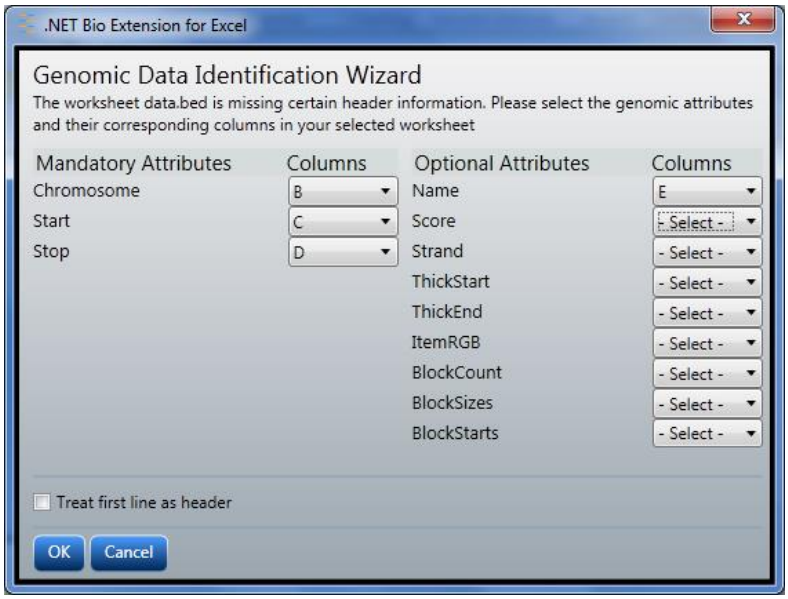
   Примјетићете да су у примјеру на сљедећој слици одабране четири колоне.



Избор опсега базних парова

6. Кликнути на иконицу за избор у прозору **Selection Helper** (Помоћ око избора), или пак притисните **Enter** да бисте се вратили на прозор **Select Input Sequences Ranges**.
7. Унесите назив захтјева и кликните на **OK**.

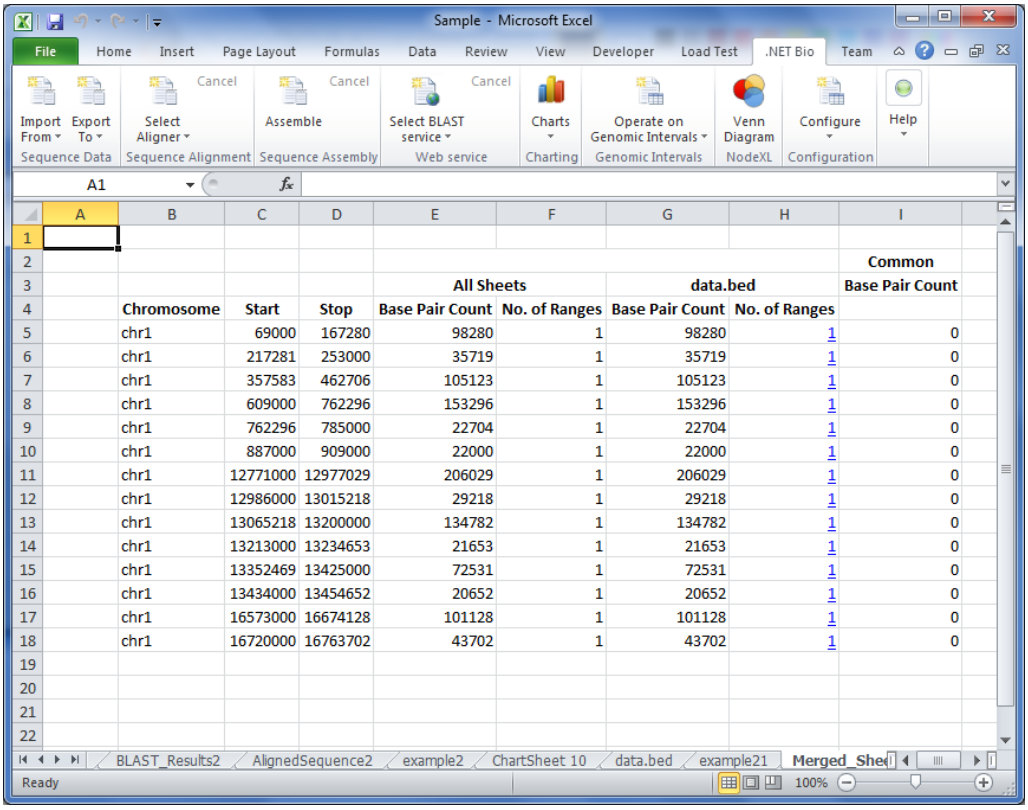
   Приказује се **Genomic Data Identification Wizard** (Чаробњак за идентификацију геномских података), као на сљедећој слици.



Чаробњак за идентификацију геномских података

8. Помоћу падајућих изборника подесите називе колона за четири изабране колоне — B, C, D, и E — потом кликните на OK.

   Речултати су приказани на новом радном листу под називом **Merged_Sheet1**, као на сљедећој слици.



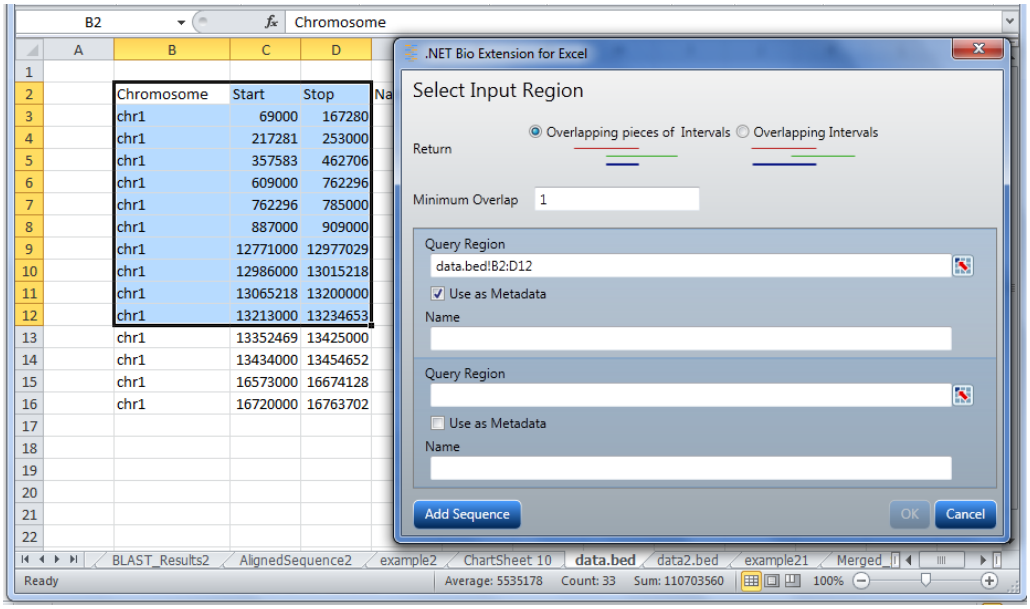
								Common
				All Sheets		data.bed		Base Pair Count
	Chromosome	Start	Stop	Base Pair Count	No. of Ranges	Base Pair Count	No. of Ranges	
	chr1	69000	167280	98280	1	98280	1	0
	chr1	217281	253000	35719	1	35719	1	0
	chr1	357583	462706	105123	1	105123	1	0
	chr1	609000	762296	153296	1	153296	1	0
	chr1	762296	785000	22704	1	22704	1	0
	chr1	887000	909000	22000	1	22000	1	0
	chr1	12771000	12977029	206029	1	206029	1	0
	chr1	12986000	13015218	29218	1	29218	1	0
	chr1	13065218	13200000	134782	1	134782	1	0
	chr1	13213000	13234653	21653	1	21653	1	0
	chr1	13352469	13425000	72531	1	72531	1	0
	chr1	13434000	13454652	20652	1	20652	1	0
	chr1	16573000	16674128	101128	1	101128	1	0
	chr1	16720000	16763702	43702	1	43702	1	0

Резултат уније

9. Кликнути на вриједности (хипервезе) у колони **No. of Ranges** да бисте видјели који су опсези из првобитног радног листа обухваћени унијом.

Пресјек интервала два захтјева

1. Кликнути на **Import From** и одабрати **BED**.
2. Означити једну или више датотека и кликнути на **Open.**
3. Означити радни лист и кликнути на **Operate on Genomic Intervals**.
4. Кликнути на **Intersect**, а потом искористити окно **Select Input Sequences Ranges** за избор два опсега са базним паровима: **Query Region** (Опсег захтјева)**,** као што је приказано на сљедећој слици.



Секвенца захтјева и референтна секвенца

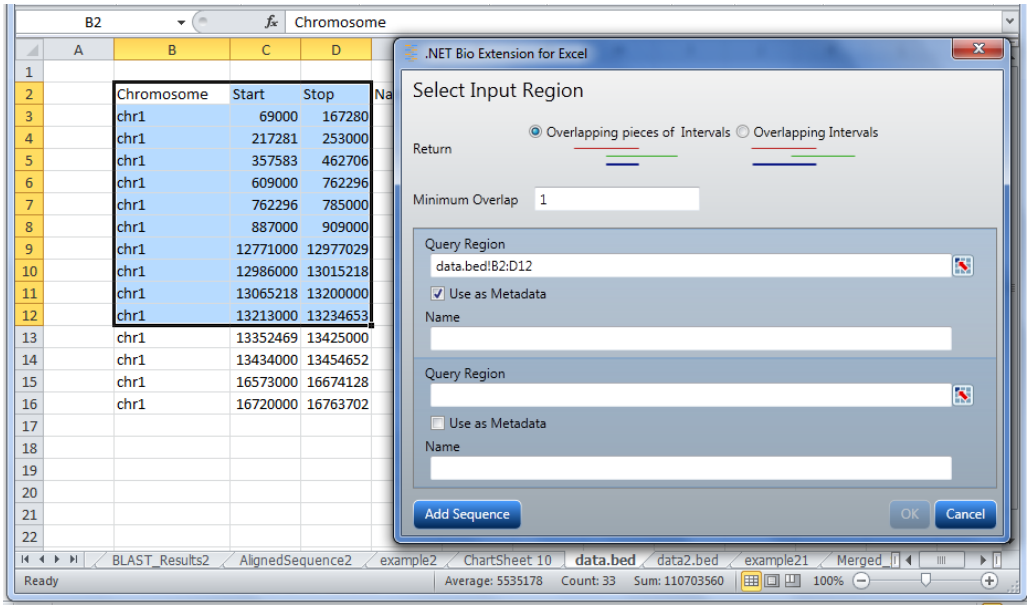
У овом примјеру се користе два радна листа, за сваку секвенцу по један.

Савјет: Додавањем заглавља избору (суб)секвенце, можете заобићи чаробњака за идентификацију геномских података. .NET Bio Extension аутоматски попуњава вриједности колоне.

5. Кликнути на **OK**. Резултати ће бити приказани на новом радном листу под именом **Intersect_Sheet1**.

Разлика два интервала

1. Кликнути на **Import From** и означити BED**.**
2. Означити једну или више датотека и кликнути на **Open**.
3. Означити радни лист и кликнути на **Operate on Genomic Intervals**.
4. Кликнути на **Intersect**, а потом искористити окно **Select Input Sequences Ranges** за избор два опсега са базним паровима: **Query Region** (Опсег захтјева), као што је приказано на сљедећој слици.



Chromosome	Start	Stop
chr1	69000	167280
chr1	217281	253000
chr1	357583	462706
chr1	609000	762296
chr1	762296	785000
chr1	887000	909000
chr1	12771000	12977029
chr1	12986000	13015218
chr1	13065218	13200000
chr1	13213000	13234653
chr1	13352469	13425000
chr1	13434000	13454652
chr1	16573000	16674128
chr1	16720000	16763702

Секвенца захтјева и референтна секвенца

5. Кликнуте на **OK**. Резултати ће бити приказани на новом радном листу под називом **Subtract_Sheet1**.

Приказ Венових дијаграма на основу (интервалних) геномских података

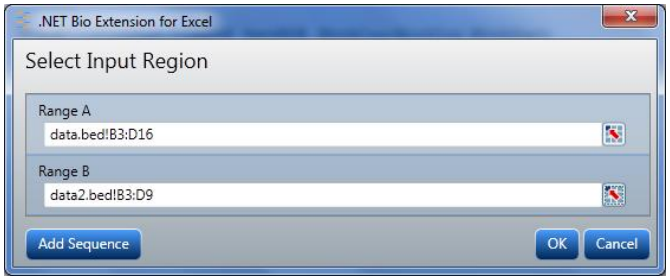
Опцијом **Venn Diagram** (Венов дијаграм)**,** можете направити 2- или 3-дјелне Венове дијаграме на основу BED (интервалних) геномских података, што омогућује визуелизацију односа међу областима, као и визуелизацију преклапања интервала.

Напомена: Venn Diagram захтијева NodeXL шаблон за Excel 2007 или Excel 2010, доступан на http://www.codeplex.com/NodeXL.

Формирање 2-дјелног Веновог дијаграма

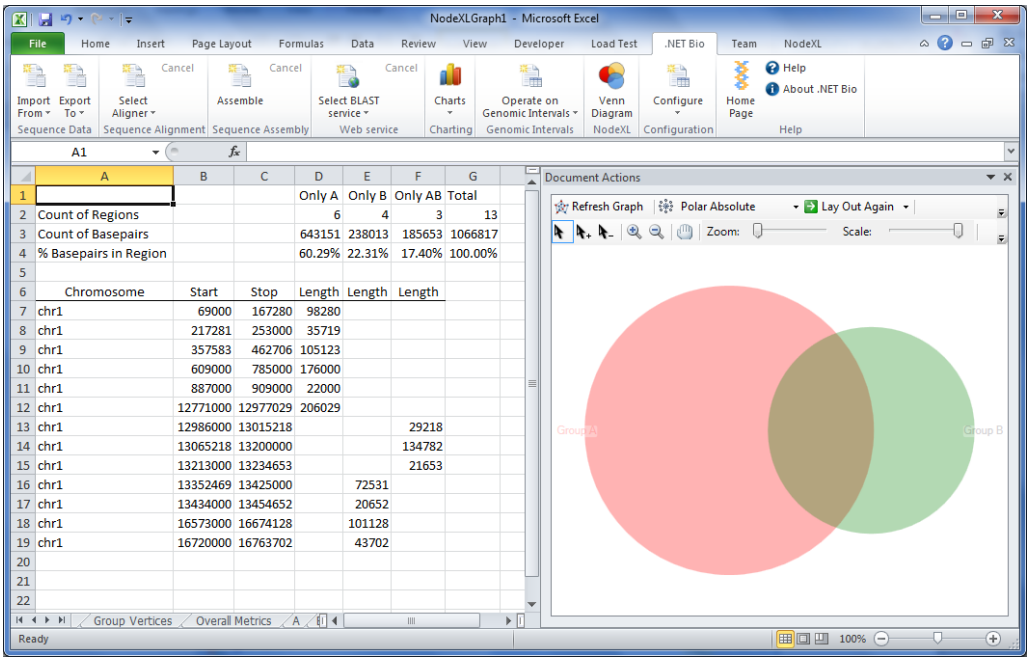
1. Кликнути на **Import From** и одабрати **BED**.
2. Означити једну или више датотека и кликнути на **Open**.
3. Означити радни лист и кликнути на **Venn Diagram**.
4. Унутар окна **Select Input Sequence Ranges**, као што је приказано на сљедећој слици, изабрати два опсега са базним паровима.

   Важно: Опсези базних парова се морају преклапати. Нпр. коришћење Chr1 за геноме горила и људи.



Дијалошко окно Select Input Sequence Ranges

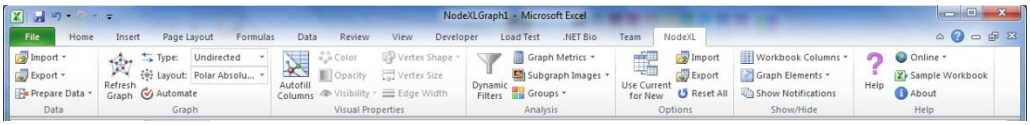
5. Резултујући Венов дијаграм је приказан у новој радној свесци под називом **NodeXLGraph1**, као што се може видјети на сљедећој слици.



Венов дијаграм генома Chr1код људи и горила

6. Кликнути на изборник NodeXL да бисте видјели ленту NodeXL, као што је приказано на сљедећој слици.

   Документација ових наредби се налази на адреси http://www.codeplex.com/NodeXL

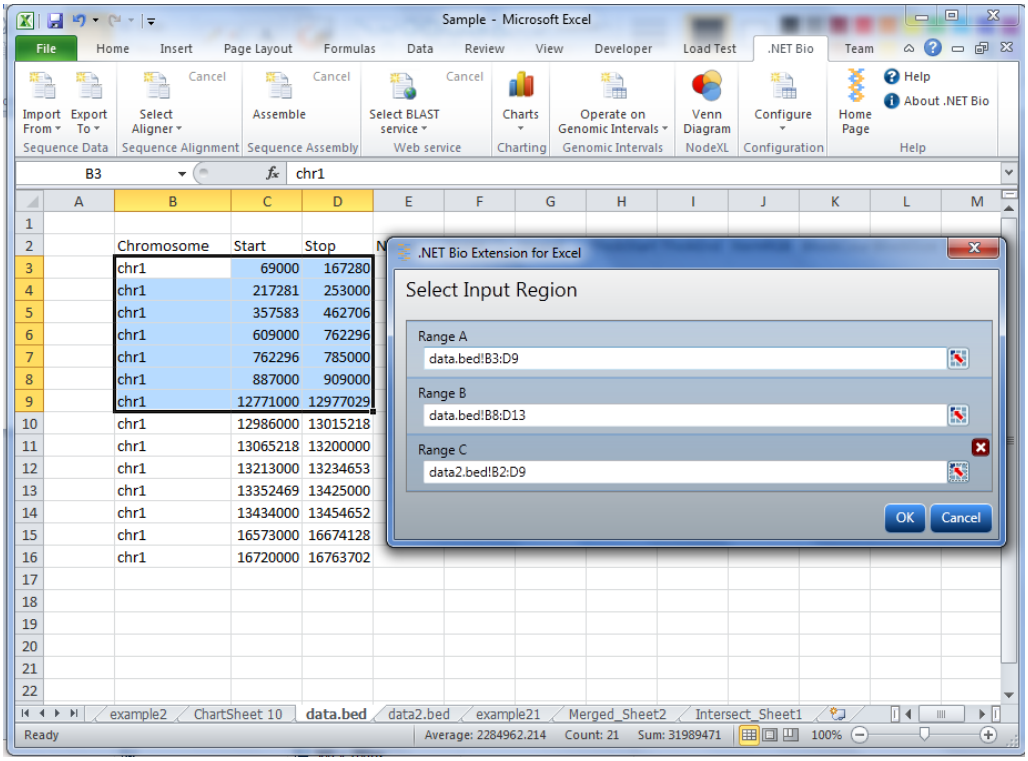


NodeXL лента

Формирање 3-дјелног Веновог дијаграма

1. Кликнути на **Import From** и одабрати **BED**.
2. Означити једну или више датотека и кликнути на **Open**.
3. Означити радни лист и кликнути на **Venn Diagram**.
4. Помоћу окна **Select Input Sequence Ranges,** као што је приказано на сљедећој слици, изаберите три опсега са базним паровима и кликните на **OK**.

   У овом примјеру, изабрана три опсега се преклапају.

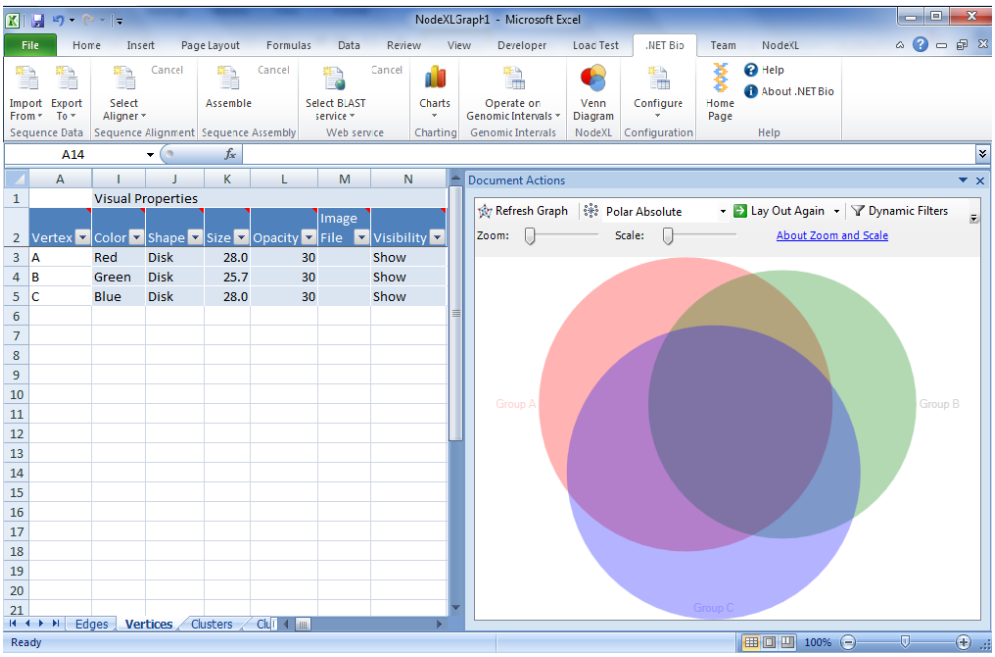


Три опсега са базним паровима који се преклапају

5. Подесите вриједност колона помоћу чаробњака **Genomic Data Identification Wizard**, као што је приказано на сљедећој слици, и кликните на **OK**.

Чаробњак за идентификацију геномских података

6. Резултати Веновог дијаграма су приказани у новој радној свесци под називом **NodeXLGraph1**, као што можемо видјети на сљедећој слици.



Тро-обласни Венов дијаграм

Промјена конфигурацијских опција

.NET Bio Extension има двије конфигурацијске опције:

- број колона-оквира за податке о геномским секвенцама;

  Мијења се начин на који Excel показује податке о секвенцама. Подразумјевани оквир је 80, а максимална вриједност је 16 000.

- шема за бојење молекула;

  Сваком молекулу се може придружити боја ради јасније представе секвенце. Само пет молекула, подразумјевано, има одређену боју: A, T, C, G, и U.

Конфигурисање колоне-оквира

1. Кликнути на **Configure** на ленти **.NET Bio**.
2. Кликнути на колону **Sequence Data Wraproud** (Оквир за податке секвенце).
3. Унијети нову вриједност у поље **Enter the maximum number of columns** (Унесите максималан број колона).

Конфигурисање шеме боја

1. Кликнути на **Configure** на ленти **.NET Bio**.
2. Кликнути на **Change Color Scheme for Molecules** (Измјена шеме за бојење молекула)**.**
3. Кликнути на дугме **Change Color** (Промијени боју) у дијалошком окну **Configure Color.**
4. Означити боју у дијалошком окну **Format Cells** (Формати поља) и кликнути на **OK**.

5. Кликнути на **OK** у прозору **Configure Color** да би се снимиле промјене.

Додатак A: Подржане секвенце и формати датотека

У овом додатку су описани подржани формати .NET Bio Framework-а, са одговарајућим хипервезама на референце, за више информација.

FASTA: Текстовне ниске података

Текстовни формат за представљање пептидних или нуклеотидних ланаца, веома погодан за обраду на програмским језицима као што је Iron Python.

Технички, формат је низ линија. Најчешће има 80 слова по линији, али никако не више од 120.

Технички опис

http://www.ncbi.nlm.nih.gov/blast/fasta.shtml

Извори

Википедијина страница http://en.wikipedia.org/wiki/FASTA_format – садржи општи преглед, хипервезе ка конверторима формата и још по неку битну референцу.

FASTQ: Квалитативне ниске података

Текстовни формат који похрањује биолошке секвенце и Phred quality бодове у једну датотеку. Често се сматра *de facto* стандардом за похрану хеуристичких и бодовних података високо-пропусног анализатора (High-Troughput Computing).

Формат је дефинисан тако да четири линије датотеке чине слог (који одговара једној геномској нисци).

Уобичајне екстензије датотека су: **.fq**, **.fastq**, **.txt**.

Технички опис

FASTQ формат спецификација
http://maq.sourceforge.net/fastq.shtml

Извори

Википедијина страница http://en.wikipedia.org/wiki/FASTQ_format – садржи општи преглед, хипервезе ка конверторима и још по неку битну референцу.

GenBank: Формат за геномске базе података

Flat-file формат који описује нуклеотиде и секвенце нуклеотида из GenBank-а – јавне базе података.

Технички опис

“Chapter 1, GenBank: The Nucleotide Sequence Database,” Ilene Mizrachi; *NCBI Handbook*, 2007

http://www.ncbi.nlm.nih.gov/books/bookres.fcgi/handbook/ch1.pdf

Извори

Веб-сајт NCBI базе података за опште информације о GenBank-у
http://www.ncbi.nlm.nih.gov/sites/entrez?db=nucleotide

Википедијина страница http://en.wikipedia.org/wiki/GenBank – садржи општи преглед, хипервезе ка конверторима и још по неку битну референцу.

GFF: Generic Feature Format

Линијски формат, при чему су поља раздвојена tab-размацима. Намијењен је за представљања слогова у геномској бази података. GFF слог представља субсеквенцу, као што је то на примјер ген или протеинска секвенца, истовремено дозвољавајући „умјерено детаљна“ образложења.

Екстензија за овај тип датотека је **.gff**.

Раније спецификације су преводиле акроним као Gene-Finding Format.

Технички опис

Садашња верзија је n2. Формат су првобитно осмислили Richard Durbin и David Haussler, а посљедња верзија садржи измјене које су предложили Lincoln Stein, Suzanna Lewis, Anders Krogh и други.

http://www.sanger.ac.uk/resources/software/gff/spec.html

Извори

Сајт института Wellcome Trust Sanger, за општи преглед формата
http://www.sanger.ac.uk/Software/formats/GFF/GFF_Spec.shtml

UCSC сајт пројекта Encode, такође за општи преглед
http://genome.ucsc.edu/goldenPath/help/customTrack.html#GFF

Browser Extensible Data (BED) Format

BED представља елегантан начин представљања геномских података и анотација, превасходно намијењен за приказ на UCSC Genome Browser-у. Поднесак за UCSC прегледник се обично састоји од главне датотеке са пољима која су одвојена tab-размацима и слоговима који су раздвојени простим празнинама.

Екстензија за овај тип датотека је **.bed**.

FAQ

Browser Extensible Data (BED) Format FAQ
http://genome.ucsc.edu/FAQ/FAQformat.html#format1

Извори

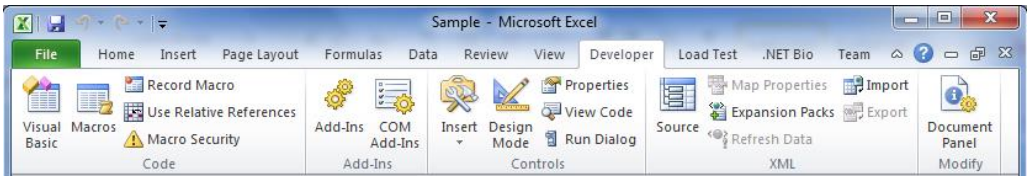
UCSS-ов сајт са општим прегледом базе података Genome Browser Database
http://users.soe.ucsc.edu/~kent/gbd.html#BED

Додатак Б: Одобравање макроа

Да бисте користили Charting функцију, морате одобрити макрое у Excel-у и додати Excel-макро радним свескама које користите у Biology Extension-у. Назив макроа је DisplayDNASequenceDistribution.bas, и долази инсталиран уз .NET Bio Framework.

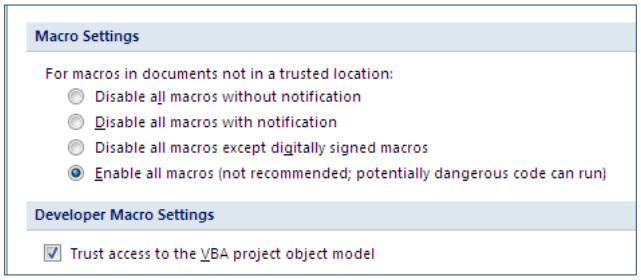
Одобравање макроа у Excel-у

1. Уз текућу отворену радну свеску, кликнути на **Office Button** у Excel-у и кликнути на **Excel Options**.
2. Кликнути на **Show Developer tab in the Ribbon**, а онда на OK.
3. Кликнути на картицу **Developer**, као што је приказано на сљедећој слици.



Developer картица

4. Кликнути на **Macro Security** у ленти **Developer**.
5. Кликнути на **Macro Settings** у прозору **Trust Center**.
6. Кликнути на **Enable all macros** и означите**Trust access to the VBA project object model** као што је приказано на сљедећој слици.

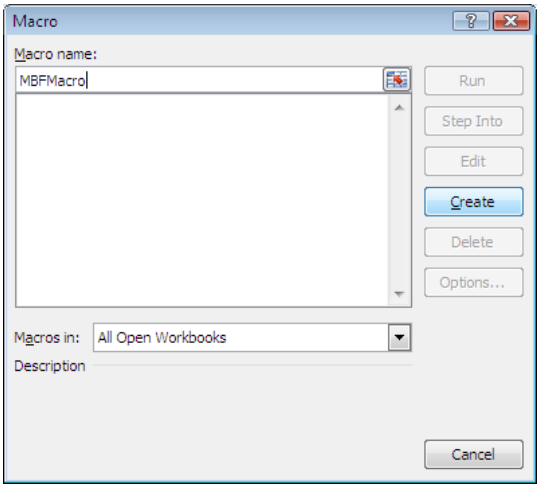


Macro Settings

7. Кликнути на **OK**.
8. Затворити и опет отворити радну свеску да би промјене ступиле на снагу.

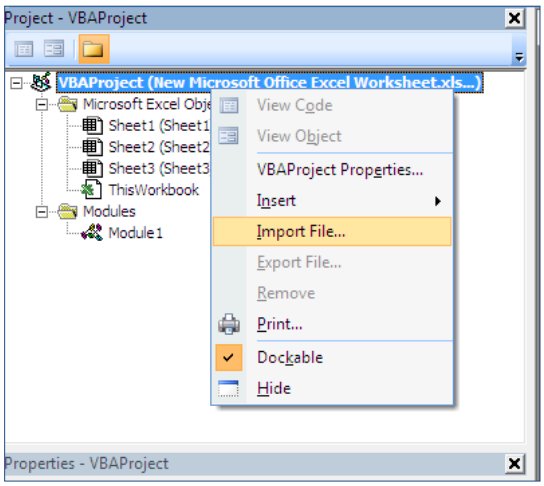
Додавање графикон-макроа

1. Кликнути на картицу **Developer**, а потом на **Macros**.
2. Унијети име макроа као што је на примјер „BioMacro” у поље **Macro name** у дијалошком окну **Macro** (погледати сљедећу слику) и кликнути на **Create**.



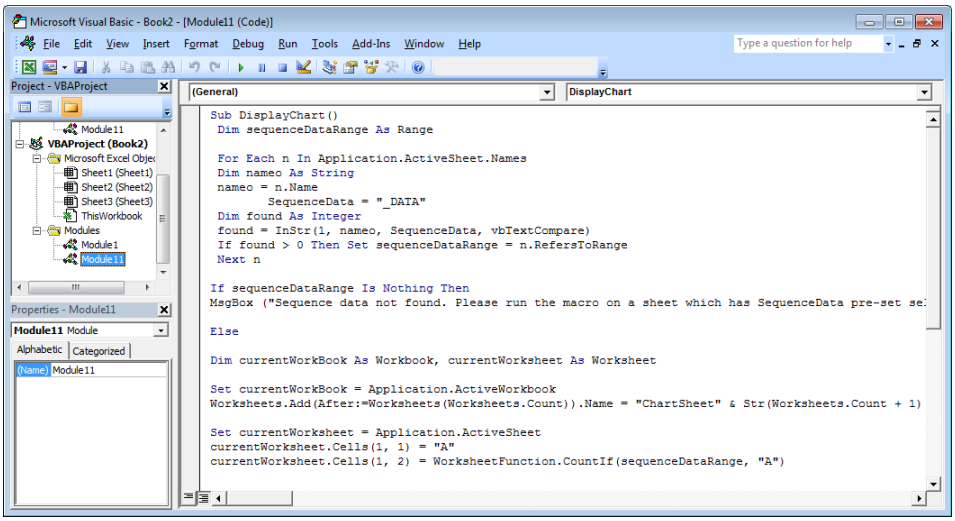
Дијалошко окно Макро

3. Десни клик на **VBAProject** у прозору **Microsoft Visual Basic**, а онда кликнути на **Import File…** као што је приказано на слици.



Наредба Import File…

4. Идите у C:\Program Files (x86)\.NET Bio\1.0\Tools\.NET Bio Extension for Excel. Означити **DisplayDNASequenceDistribution.bas** и кликнути на **Open.**
5. Двоструким кликом миша на **Module 11** да се прикаже макро, као што је приказано на слици.



Макро DisplayChart

6. Кликнути **Save** (Сачувај) и снимити као документ типа **Excel Macro-Enabled Workbook (*.xlsm)**.

Сада, када сте допустили макрое у Excel-у и додали марко DisplayDNASequence-Distribution.bas, можете користити Charting функцију у Biology Extension-у.

Технички водич кроз .NET Bio Framework Parallel De Novo Assembler

Верзија 1.0 Јун 2011

Сажетак

Описана је класа ParallelDeNovoAssembler (Padena). Ради се о програмској реализацији de novo секвенцирања, заснованог на de Bruijn-овим графовима.

Преглед

ДНК машине за секвенцирање обично раде са 500-1000 базних фрагмената (по очитавању). У посљедње вријеме све су популарније технике које омогућују кратка очитавања, дужине 25-100bp. Наравно, потребно је дате фрагменте повезати у једну непрекидну геномску секвенцу.

Постоје два најчешћа приступа:

коришћење референтне секвенце или одговарајућег низа маркера (формирање на основу референце)

кориштење великог броја преклапајућих секвенци за одређивање редослиједа нуклеотида, и то непосредно на основу дијелова саме секвенце (de novo техника).

Код de novo технике приступ је обично заснован на једном од сљедећа два концепта:

преклапање-диспозиција-усаглашеност

de Bruijn-ови графови, који математички моделују проблем формирања генома као проблем проналажења Ојлеровог пута у графу.

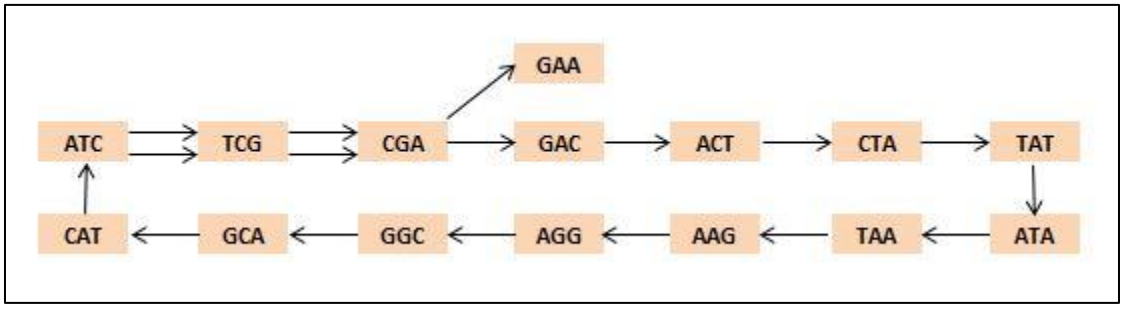
Већина најчешће кориштених de novo алгоритама су засновани на de Bruijn-овим графовима.

De Bruijn-ов граф је усмјерени граф код којег су чворови знаковне ниске, а краци указују на могућа преклапања.

Примјер: дата је ниска **ATCGACTATAAGGCATCGAA.**

Потребно је формирати de Bruijn-ов графа са чворовима дужине 3 (k-mer-и)

Усмјерени крак између два чвора се убацује ако је (k+1)-mer такав да обухвата два чвора (сусједни чворови увјек имају преклапање од (k-1) базе)



Алгоритми за састављање генома као што су ABySS, Velvet и EULER-SR засновани су на de Bruijn-овим графовима. Иначе, на вишем нивоу апстракције, ови алгоритми имају много заједничких корака.

Конструкција

Напомена. За сваки корак, постоји одјељак „Паралелизација” у којој се описани дијелови обраде података који се могу паралелизовати. Послије сваког паралелизираног корака, када је потребно повезати добијене вриједности, користићемо конкурентно-безбједне структуре података из `System.Collections.Concurrent`.

ParallelDeNovoAssembler програмска класа [простор назива Bio.Algorithms.Assembly]

ParallelDeNovoAssembler програмска класа представља реализацију de novo технике за састављање генома коришћењем de Bruijn-ових графова.

У овој имплементацији састављача секвенци (Padena), комбинује се неколико различитих приступа:

првих пет корака (формирање контига на основу очитавања улазне секвенце) de novo имплементације су преузете из ABySS алгоритма као што је описано у изворној публикацији: http://genome.cshlp.org/content/19/6/1117.full

посљедњи корак (формирање суперконтига), комбинује сљедећа два приступа:

ABySS алгоритам

Greedy Path Merging Algorithm

http://research.janelia.org/myers/Papers/greedy.path.merging.pdf

Улазни подаци за састављање геномске секвенце у Padena класи су:

листа очитавања за улазну секвенцу

информација о упареним очитавања: добија се mate-pair веза за очитавања секвенце, која представља пресликавање регуларних на обратна очитавања.

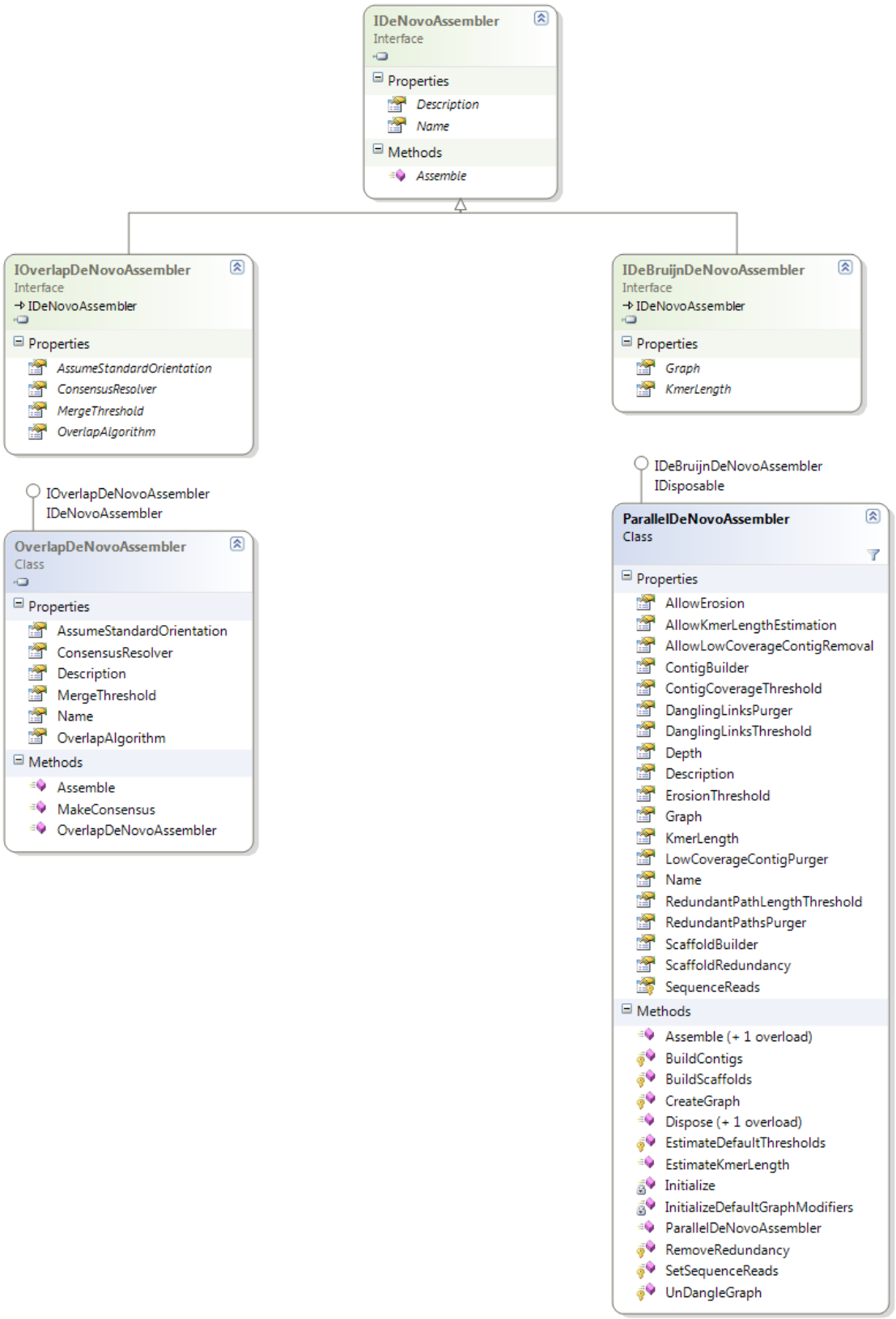
кориснички улазни параметри:

дужина k-mer-a: `_kmerLength`

сигнална вриједност за уклањање „висећих крајева”: `_dangleThreshold`

сигнална вриједност за уклањање сувишних путева: `_redundantPathLengthThreshold`

Class Diagram:
Namespace Bio.Algorithms.Assembly.Padena


IDeNovoAssembler
Interface
Properties
Description
Name
Methods
Assemble
IOverlapDeNovoAssembler
Interface
IDeNovoAssembler
Properties
AssumeStandardOrientation
ConsensusResolver
MergeThreshold
OverlapAlgorithm
IDeBruijnDeNovoAssembler
Interface
IDeNovoAssembler
Properties
Graph
KmerLength
IOverlapDeNovoAssembler
IDeNovoAssembler
OverlapDeNovoAssembler
Class
Properties
AssumeStandardOrientation
ConsensusResolver
Description
MergeThreshold
Name
OverlapAlgorithm
Methods
Assemble
MakeConsensus
OverlapDeNovoAssembler
IDeBruijnDeNovoAssembler
IDisposable
ParallelDeNovoAssembler
Class
Properties
AllowErosion
AllowKmerLengthEstimation
AllowLowCoverageContigRemoval
ContigBuilder
ContigCoverageThreshold
DanglingLinksPurger
DanglingLinksThreshold
Depth
Description
ErosionThreshold
Graph
KmerLength
LowCoverageContigPurger
Name
RedundantPathLengthThreshold
RedundantPathsPurger
ScaffoldBuilder
ScaffoldRedundancy
SequenceReads
Methods
Assemble (+ 1 overload)
BuildContigs
BuildScaffolds
CreateGraph
Dispose (+ 1 overload)
EstimateDefaultThresholds
EstimateKmerLength
Initialize
InitializeDefaultGraphModifiers
ParallelDeNovoAssembler
RemoveRedundancy
SetSequenceReads
UnDangleGraph

Интерпретација Class Diagram-а:

На самом врху је програмско прочеље за све de novo састављаче – IDeNovoAssembler. Даље, оно је подјељено на два прочеља која представљају сљедеће двије опште програмске класе (које одговарају различитим процесима при састављању генома):

- IOverlapDeNovoAssembler: програмско прочеље за концепт преклапање-диспозиција-усаглашеност.
- IDeBruijnDeNovoAssembler: програмско прочеље за концепт заснован на De Bruijn-вим графовима.

Програмска класа `ParallelDeNovoAssembler представља реализацију наше` de novo технике

- дата су поља у којима се похрањују вриједности улазних параметара (_kmerLength, _dangleThreshold, __redundantPathLengthThreshold, _sequenceReads)
- за сваки корак у de novo алгоритму постоји посебна програмска класа; 'ParallelDeNovoAssembler' декларише поље које се односи на класу и спроводи сваки од ових корака: `(_danglingLinksPurger, _redundantPathsPurger, _contigBuilder, _scaffoldBuilder`); више детаља о свакој од ових класа даље у тексту
- метод '`Initialize`' служи за иницијализацију, а и омогућује рад састављача:

  у случају да корисник није задао _kmerLength, вриједност се процјењује на основу дужина улазних очитавања; отприлике, опсег дозвољених kmer-вриједности је од (дужина најдуже секвенце / 2) до дужине најкраће секвенце. Средња вредност се бира за за _kmerLength

  сигналним вриједностима које нису иницијализоване придружене су подразумијеване вриједности на основу

  ```
  _kmerLength:
  _dangleThreshold = _kmerLength + 1
  _redundantPathLengthThreshold = (int)Math.Ceiling(1.5f * _kmerLength)
  ```

  класе за измјену графа, као што су _danglingLinksPurger и _redundantPathsPurger, користе подразумијеване вриједности одређене конкретном реализацијом коју алгоритам користи у одговарајућем кораку

  поред тога, овај метод такође уклања сва очитавања која имају двосмислене симболе

- реализовани су методи програмског прочеља класе:
  `public IDeNovoAssembly Assemble(List<ISequence> inputSequences);`

  овај метод се састоји од низа корака за de novo састављање, а повратна (резултујућа) вриједност је **`PadenaAssembly`** (програмски) објекат у којем је похрањен резултат састављања генома.

- за сваки корак састављања користи се заштићени метод, што омогућује извођење класа и надјачавања међу истоименим методима када су у питању неки посебни кораци и захтјеви:

  Step 1, 2: CreateGraph() – формира k-mer на основу очитавања и конструише de Brujin-ов граф
  Step 3: UnDangleGraph()
  Step 4: RemoveRedundancy()
  Step 5: BuildContigs()
  Step 6: BuildScaffolds()

Паралелизација:
На овом нивоу нема паралелизације.

Излаз:
Резултат састављања је дат као програмски објекат класе **`PadenaAssembly`**:

- листа састављених контигних секвенци
- листа састављених суперкконтига

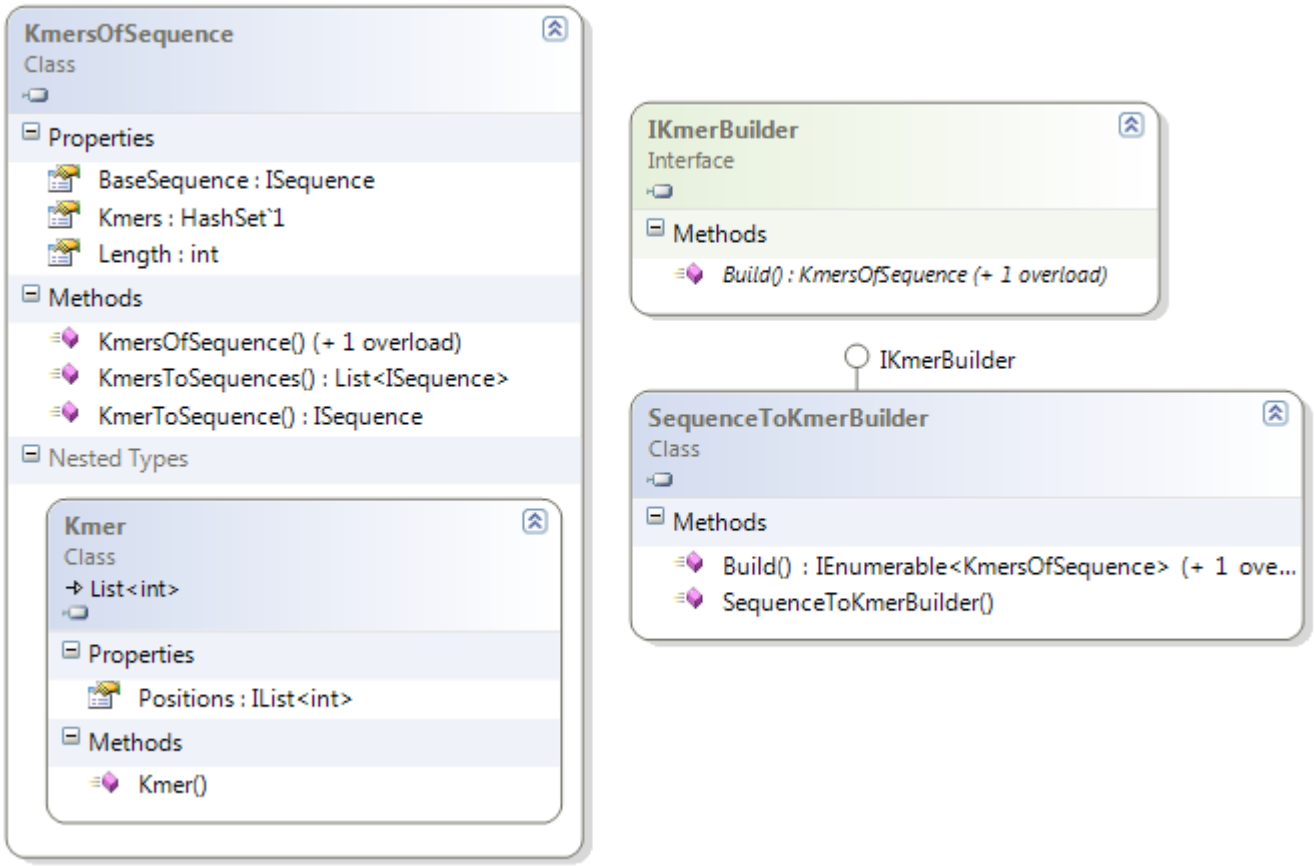
Корак 1, 2: Конструкција графа [namespace Bio.Algorithms.Assembly.Graph]

У првом подкораку свака улазна секвенца очитавања се разбија на секвенце дужине k (k-mer-и).

Напомена: овај скуп класа је укључен у било који Bio-пројекат, јер се користе и у Padena и у DiffSeq (EMBOSS).

Улаз:
Листа улазних очитавања секвенце, и k као дужина k-mer-а

Диаграм класе:
Namespace Bio.Algorithms



Интерпретација дијаграма:

- сваки објекат класе ***KmersOfSequence*** представља једну улазну секвенцу и при том садржи списак k-mer-а који су дио секвенце.
- угнијежђена класа ***Kmer:*** представља k-mer у оквиру основне секвенце; похрањивање ISequence за сваки Kmer-а је „разбацивање" са расположивом меморијом, пошто има доста преклапања између k-mer-ских секвенци; другим ријечима, похрањује са само листа стартних позиција у односу на базну секвенцу.
- ***KmersOfSequence:***

  похрањује улазну секвенцу у baseSequence.

  држи на окупу инстанце класе Kmer које представљају k-mer-е придружене објекту baseSequence; k-mer-и су одређени почетном позицијом у базној секвенци; на тај начин, уз базну секвенцу, потпуно дефинишу скуп k-mer-а придружених секвенци

  пошто се похрањују само почетне позиције, Kmer не може самостално формирати ISequence / знаковну ниску k-mer-а; но постоје помоћни методи класе KmersOfSequence којима је обезбјеђен приступ k-mer-има у облику знаковне ниске / секвенце:

```
public ISequence KmerToSequence(Kmer k)
public IEnumerable<ISequence> KmerToSequences()
```

  Прати се Builder образац; процес формирања k-mer-а на основу секвенце се раздваја од њене представе, што омогућује различите реализације самог процеса формирања k-mer-а.
- прочеље ***IKmerBuilder***: обезбеђује прочеље за формирање k-mer-а на основу дате секвенце.

класа ***SequenceToKmerBuilder*** представља програмску реализацију претходно поменутог прочеља; с обзиром на конкретну секвенцу, она приказује прозор величине _length дуж саме секвенце, издвајајући у задатом опсегу њене чланове ради формирања k-mer-а; реализовани су методи за формирање k-mer-а на основу једне секвенце или пак листе улазних секвенци; значи, постоје два могућа преоптерећења метода – рад са листом секвенци и ради са једном секвенцом.

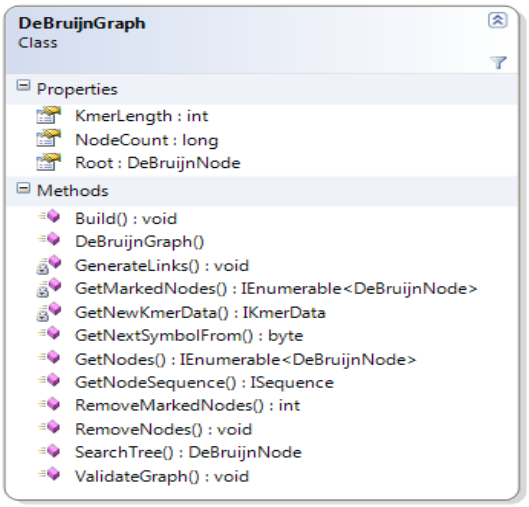
Паралелизација:

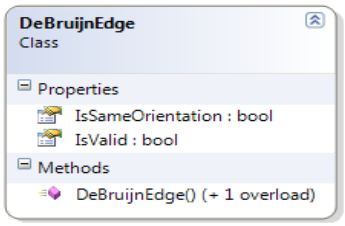
За свако очитавање улазне секвенце, формира се посебан `Task[1]` (.NET паралелизацијски конструкт). Сваки Task ће као повратну (резултујућу) вриједност дати одговарајући `KmersFromSequence`.

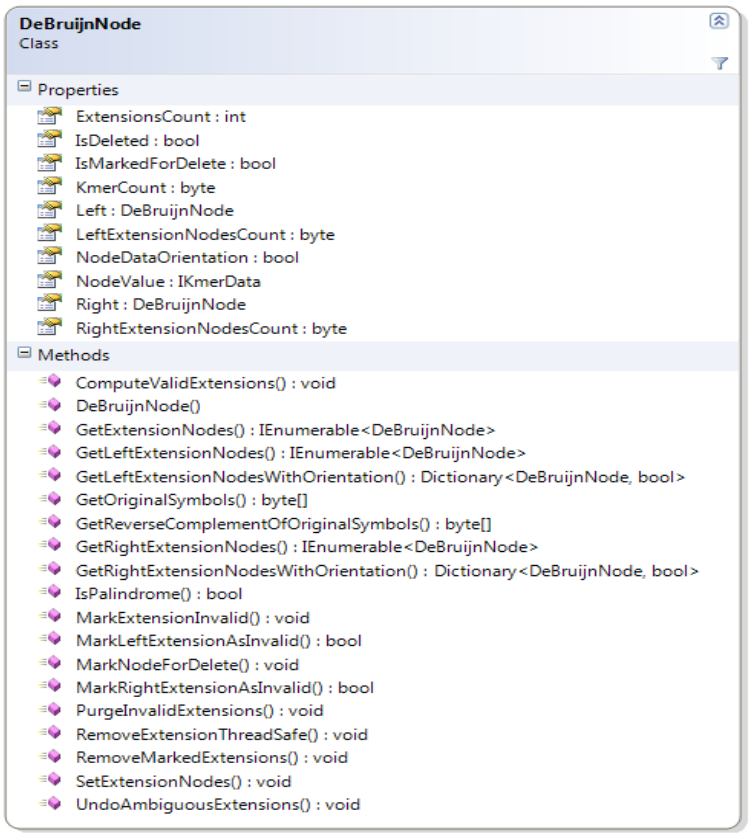
[1] Задатак, у слободном преводу.

У наредном подкораку се конструише de Brujin-ов граф на основу k-mer-а. Одређују се чворови и краци графа, на основу којих ће бити реализован процес састављања.

Диаграм класе
Namespace Bio.Algorithms.Assembly.Graph







Интерпретација диаграма:

За de Brujin-ов граф, поново се може слиједити Builder образац. Три класе су доступне:

- класа ***DeBruijnNode*** описује чвор у графу.

  сваком чвору је придручен k-mer; чвор садржи почетну позицију придруженог k-mer-а; овај индекс има референцу на базну секвенцу у графу DeBrujinGraph који садржи сам чвор.

  `LeftExtensionNodes, RightExtensionNodes` представљају чворове који су повезани с лијеве и с десне стране текућег чвора, као придружене краке.

  лијеви крак биће смјештен између чворова A и B, ако су последњих (k-1) ставки секвенце чвора B исти као и првих (k-1) ставки чвора A; слично томе, у десни крак биће смјештен између чворова A и B, ако су посљедњих (k-1) ставки секвенце у A исти као првих (k-1) ставки чвора B.

`_countNormalOrientation`, `_countReverseComplement` се односе на број појављивања k-mer-а придруженог чвору или његовог обрнутог комплемента у улазној листи секвенци; својство `KmerCount` представља збир датих вредности.

дати су методи омогућују додавање/уклањање лијевих/десних проширујућих чворова и ажурирање одговарајућих вриједности; користе се током конструкције графа.

- класа ***KmerIndexer*** одржава индекс k-mer-а као индекс секвенце, тј. колико пута се k-mer јавља у секвенци и информацију о усмјерењу (обрнути комплемент или пак не); ова информација се комбинује са листом позиција за тачну локацију k-mer-а у очитавању; дато пресликавање (k-mer у очитавање) се користи у 6. кораку de novo технике
- класа ***DeBruijnGraph*** представља граф-структуру формирану на основу DeBruijnNodes.

  служи за похрањивање скупа чворова у `_kmerNodes`

  садржи поље `_baseSequence` које служи за похрану сегментиране секвенце формиране узастопним надовезивањем свих улазних секвенци; метод `GetNodeSequence` се користи за конструкцију знаковне ниске чвора, наравно, на основу базне секвенце и доступних индекса

  дефинисани су методи који омогућују уклањање чворова и тиме измјене самог графа (`RemoveNodes`)

  дефинисан је метод који брише чвор из графа.

  `Build()` формира чвор за сваки k-mer и додаје информацију о његовом суседству

  `BuildContigGraph()` је дио 6. корака; конструише граф контига на основу k-mer-графа, с циљем формирања супекронтига.

Паралелизација:

Формирање чвора и његових кракова је паралелизовано. За сваки KmersOfSequence посебно се води рачуна о формирању чвора за k-mer-е секвенце, тј. започиње се нови Task; након формирања чвора, сваки k-mer-чвор провјерава постојање својих сусједних k-mer-а; стога, Task-ови се могу поново искористити – засебан Task је придружен сваком k-mer-чвору.

Излаз:

de Brujin-ов граф.

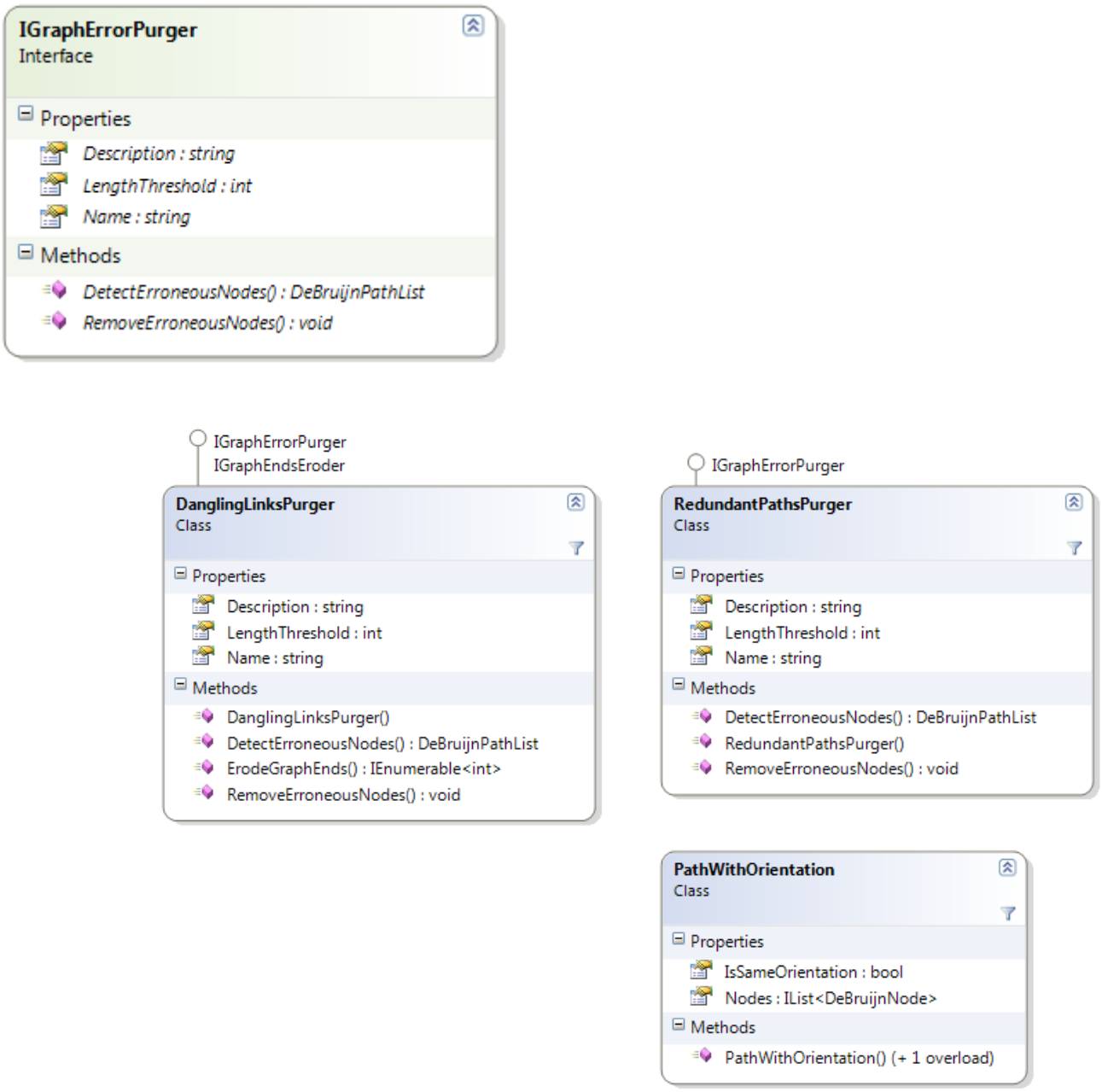
Корак 3, 4: Исправљање грешке [namespace Bio.Algorithms.Assembly.Padena]

Овај скуп класа обезбјеђује функционалност исправљања грешака, засновано на графовима, што одговара корацима за дотеривање и „дување мјехурића” (Steps 3, 4) у ABySS-у.

Улаз:

`DeBruijnGraph граф`

Диаграм класе:

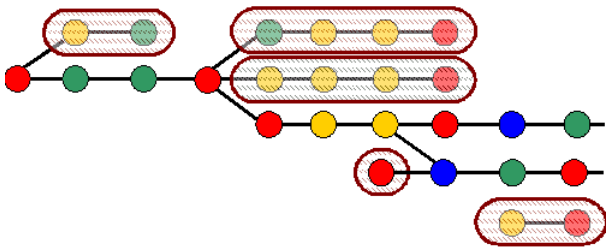


Интерпретација дијаграма:

IGraphErrorCorrection обезбјеђује прочеље за исправљање грешака, засновано на структури графа. Конкретније, обезбјеђује апстрактне методе за уочавање чворова који потпадају под неке критеријуме (`DetectErroneousNodes`), као и за накнадно уклањање тих чворова из графа (`RemoveErroneousNodes`).

Тренутно су подржана два механизма за исправљања грешака. Свака од ових класа (такорећи механизама) обезбјеђује претходно описано прочеље.

- DanglingLinksPurger: класа представља реализацију алгоритма за уклањање грешака заснованог на уочавању и уклањању висећих крајева. Ради се о реализацији AbySS-овог корака за „кресање сувих грана”.



Слика: На крају овог корака су уклоњени сви заокружени чворови [4]

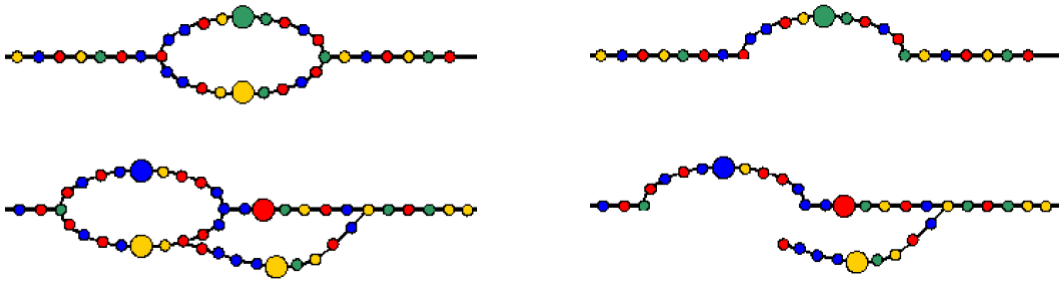
`LengthThreshold` похрањује сигналну вриједност (улазни параметар) која је потребна за идентификацију „висећих крајева”

`GetDanglingLinksLengths` метод се користи првије свега за одређивање дужине висећих крајева; управо ова информација се користи за утврђивање редослиједа самих уклањања.

`DetectErroneousNodes` метод идентификује чворове који се налазе на висећим крајевима, а који не прекорачују сигналну вриједност.

`RemoveErroneousNodes` метод уклања чворове идентификоване у претходном кораку, при чему се успут ажурира информација о крацима сусједног чвора.

- RedundantPathsPurger: Уочава и уклања сувишне путеве. Реализује се AbySS-ов корак „дување мјехурића”.



Примјери улаза, de Brujin-ових графова, за овај корак [4] **Исход, за графове с лијева [4]**

`_lengthThreshold` за похрану сигналног улазног параметра.

`DetectErroneousNodes` метод идентификује кракове који имају заједничку почетну и крајњу тачку. Потом, на основу параметара самог графа, идентификује кракове (и одговарајуће чворове) који се требају уклонити.

`RemoveErroneousNodes` метод уклања чворове идентификоване у претходном кораку.

Класа ***PathWithOrientation***: Сваки пут има смијер, који се ажурира у зависности од усмјерења крака који се додаје графу. Обиљежја (објеката) ове класе су листа чворова на датом путу и његов смијер.

Паралелизација:

Обоје, уочавање и уклањање, могу се паралелизовати. Засебни Task-ови биће дефинисани за сваки чвор који потенцијално може указати на висећи крај или сувишни пут. Послије тога, у оквиру Task-а, разматра се могуће проширивање и уклањање

елемената графа. Током уклањања, Task-ови се поново односе на скупове чворова – ажурирање сусједних чворова је паралелизовано.

Излаз:
DeBruijn-ов граф са исправљеним грешкама

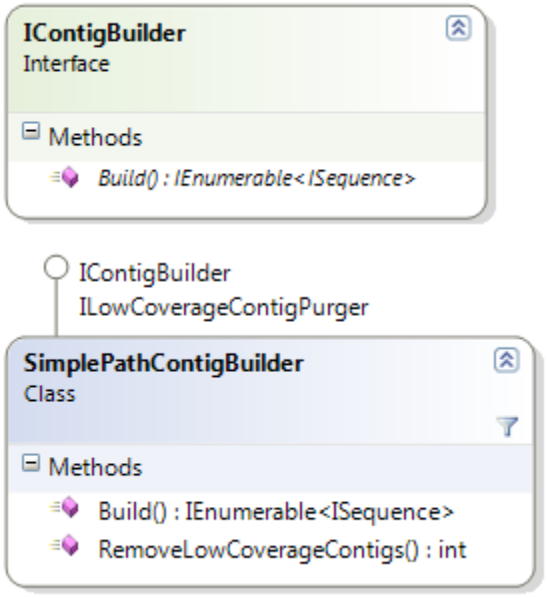
Корак 5: Формирање контига [namespace Bio.Algorithms.Assembly.Padena]

Ово одговара Vertex Merging (Assembly – SET) кораку у ABySS алгоритму:

- уклањају се краци који нису једнозначни (крак није једнозначан кад год тјемена повезана са краком имају више упадајућих/излазних кракова)
- обједињавање тјемена дуж једнозначних кракова ради формирања иницијалних контига; изградња секвенци дуж простих путева у графу, ради формирања контига.

Улаз:
De Bruijn-ов граф

Диаграм класе:
Namespace Bio.Algorithms.Assembly.Padena



Интерпретација дијаграма:

- прочеље ***IContigBuilder*** представља радни оквир за изградњу контига, тј. садржи апстрактни метод за формирање контига на основу de Brujin-овог графа.
- класа ***SimplePathContigBuilder*** представља реализацију поменутог прочеља; метод `Build` идентификује просте путеве (који имају један почетак и један крај), а потом форсира формирање одговарајућих контига.

  Одавде добијамо контиге као листу de Brujin-ових чворова (`ContigNodes`), што се користи за смањење обраде података у 6. кораку алгоритма.

Паралелизација:

За сваки чвор који има један одлазећи крак, формира се засебан Task. У сваком Task-у се провјерава да ли је пут од тог чвора прост. Ако јесте, дефинишу се одговарајуће контигне конструкције.

Излаз:

Листа контига (који су представљени као секвенце)

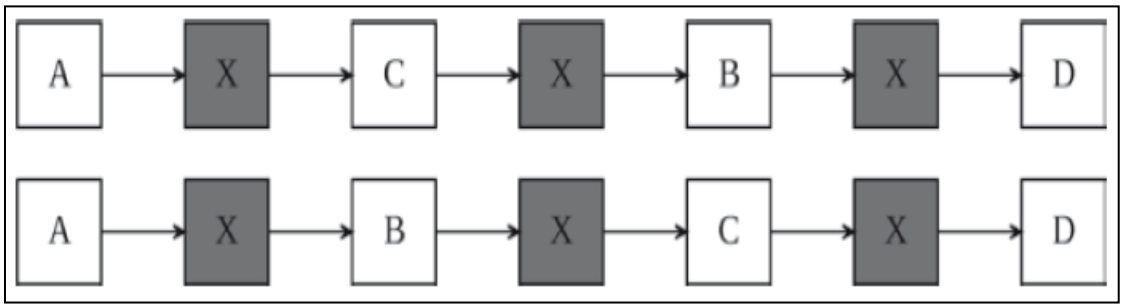
Корак 6: Грађење суперконтига [просто назива Bio.Algorithms.Assembly.Padena.Scaffold]

Позадина

Велики број понављања у ДНК секвенци може проузроковати својеврсне недоумице приликом састављања генома и на тај начин отежати читав процес.

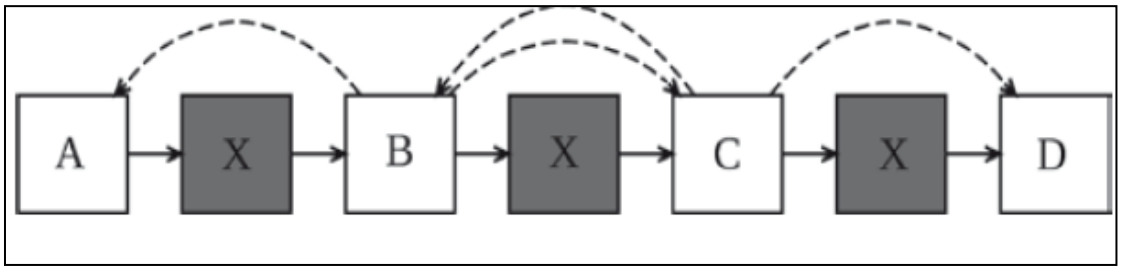
Сценарио

Знамо да су региони B и C окружени идентичним регионима X (има их више, понављају се), и да региони леже између региона A и D. Но, немамо довољно података да би открили тачан редослијед поменутих региона.[3]



Користи се paired-end секвенцирање, гдје су за оквирно познату величину, генерисани и секвенцирани фрагменти ДНК, са оба њена краја. Информација о овим паровима, као што је просечна величина фрагмента, и усмјерење очитавања у односу на пар очитавања, може бити укључена у сам процес састављања. Ако је растојање између paired-end-ова (функција величине уметања) довољно велико, онда постоји велика вјероватноћа да ће понављања бити захваћена паром очитавања (или mate-овима). што може расвјетлити нејасноће настале током самог процеса састављања.

На примјер, ако су анализирани paired-end подаци, и утврдило се да регион B има mate-ове у регионима A и C, али не и D, при чему регион C има mate-ове у региону B и D, али не и у A, онда се можемо извести закључак о каквом се редослиједу ради.



Испрекидане стрелице указују на информацију о упареним очитавањима. [3]

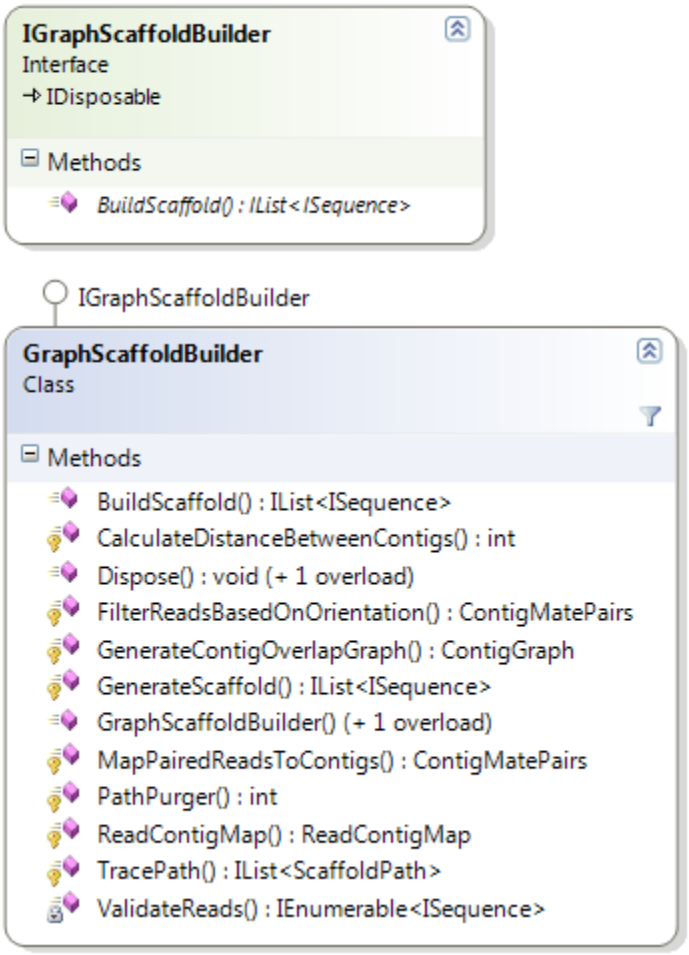
Улаз:

- Листа очитавање секвенце

- Граф (`DeBruijnGraph граф`)

Листа контига

Диаграм класе:



Интерпретација дијаграма:

Прочеље ***IScaffoldBuilder*** представља оквир за формирање суперконтига (уређене секвенце контига).

Омогућен је позив метода `BuildScaffold` чија је функција управо формирање суперконтига.

Класа ***GraphScaffoldBuilder*** представља програмску реализацију претходно поменутог прочеља

- дата су поља (обиљежја/својства) служе за похрањивање одговарајућих улазних вриједности/аргумената (_redundancy, _depth, _contigData, _contigGraph, _kmerLength)
- за сваки корак de novo алгоритма постоји посебна програмска класа; '***GraphScaffoldBuilder***' садржи поља која се односе на одговарајуће, претходно поменуте, програмске класе; више информација о свакој од ових класа је дато у даље у тексту.

- метод ‘Initialize’ иницијализује, тј. придружује почетне вриједности промјенљивим (пољима) послије њихове провјере.
- реализован је и метод прочеља:

  ```
  public IList<ISequence> BuildScaffold(IList<ISequence> reads)
  ```

  конкретно, састоји се од низа корака за формирање суперконтига (Scaffold Generation), а повратна вриједност му је објекат IList<ISequence> који садржи Scaffold секвенцу (суперконтиг).
- за сваки корак процеса се користе заштићени методи, што омогућује извођење класа и дефинисање надјачавајућих метода за специфичне кораке и посебне ситуације:

  Step 1: MapPairedReads()
  Step 2: ReadContigMap()
  Step 3: ModifyGraph()
  Step 4: FilterReadsBasedOnOrientation()
  Step 5: CalculateDistanceBetweenContigs()
  Step6: TracePath()
  Step 7: PathPurger()
  Step 8: GenerateScaffold()
- својства:

  Redundancy: број mate-pair-ова који се требају размотрити приликом формирања mate-pair везе међу контизима.

Паралелизација:

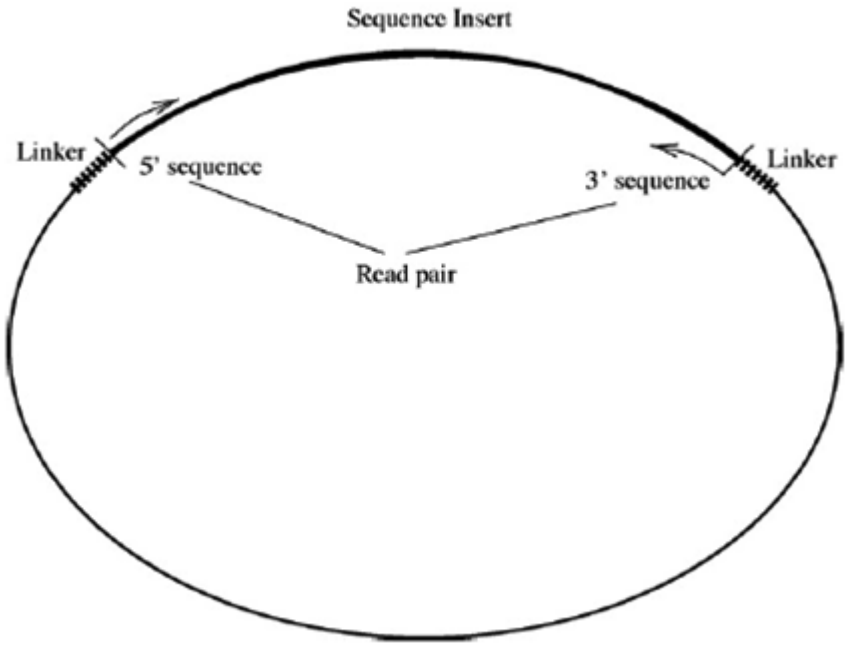
Сравњење очитавања са контизима и пресликавање очитавања на упарена очитавања су паралелизовани на нивоу корака. Дати Task-ови су међусобно независни.

Излаз:

Листа секвенци суперконтига (List<ISequence>)

Корак 1: Пресликавање очитавања у упарена очитавања

У paired-end секвенцирању очитавање са оба краја чини секвенцу уметања, при чему су за формирање очитавања познате локације, просторно заузеће, и релативна усмјерења.



Вектор секвенце. Шематска депикција вектора секвенце, као што је BAC (Bacterial Artificial Chromosome). Уметање може бити геномски фрагмент, или cDNA (за EST секвенцирање). У оба случаја, секвенцирање од било ког краја ће произвести пар очитавања који ће пружити допунске информације састављачима. [1]

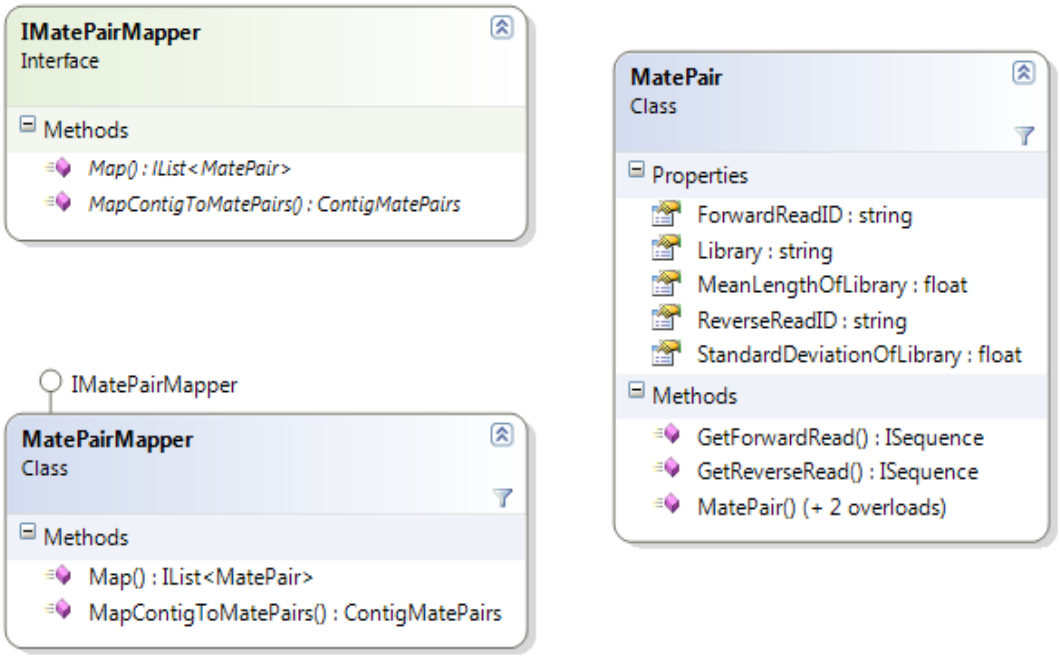
Улаз:

`IList<ISequence>` улазна листа очитавања

Излаз:

`IList<MatePair>`: листа mate-парова

Диаграм класе:



Интерпретација дијаграма:

- ReadPair класа има улогу концептуалног контејнера упарених очитавања, заједно са информацијом из програмске библиотеке; MapReadPairs класа преобраћа улазне листе очитавања у упарена очитавања на основу података доступних из FastA заглавља.
- прочеље ***IMatePairMapper*** представља радни оквир за пресликавање очитавања на mate-pair-ове, за шта је одговоран метод Map.
- класа ***MatePairMapper*** је програмска реализација претходно поменутог прочеља.
- Подржани формати mate-pair-ова:

  >chrI0.X1:abc
  ATGC
  >chrI0.Y1:abc
  TACG
  >chrI0.F:abc
  ATGC
  >chrI0.R:abc
  TACG
  >chrI0.1:abc
  ATGC
  >chrI0.2:abc
  TACG

при чему:

X1,F,1 означавају регуларна очитавања, а Y1,R,2 означавају обратна очитавања

abc означава назив библиотеке

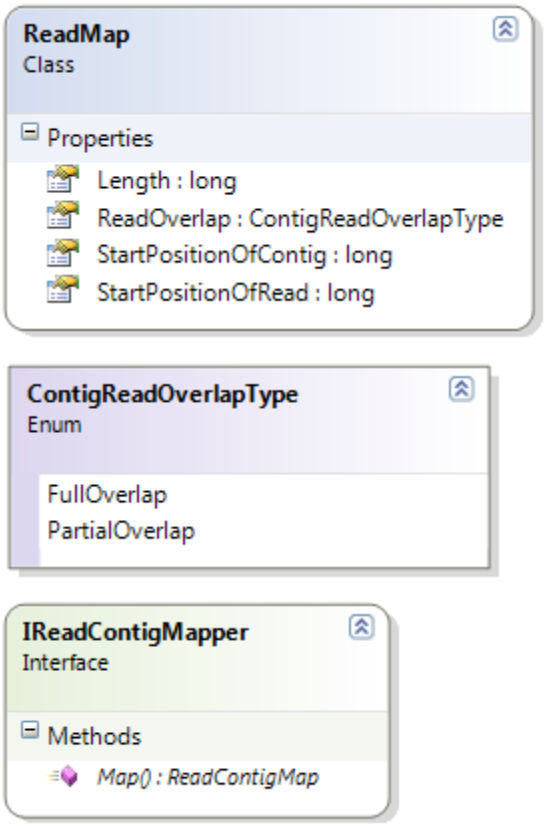
chrI0 је ID секвенције

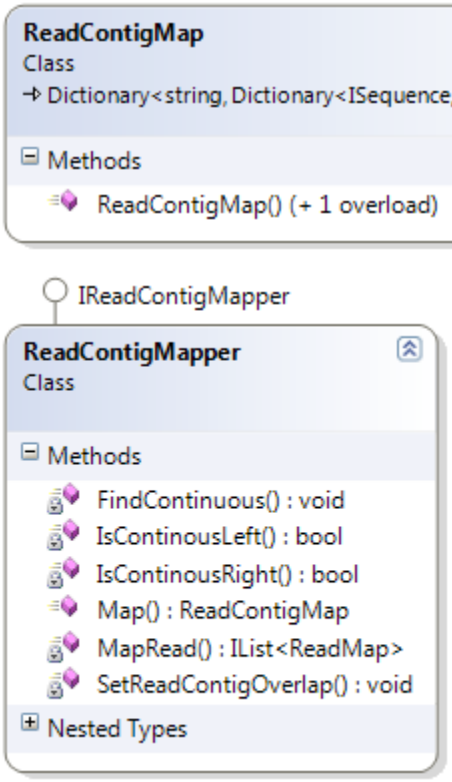
Корак 2: Пресликавање очитавања у контиге

Улаз:
`IList<ISequence>`: улазна листа очитавања, `IList<ContigData>`: улазна листа контига

Излаз:
ReadContigMap

Диаграм класе:





Интерпретација дијаграма:

- ReadContigMap класа обезбјеђује похрањивање пресликавања одговарајућих очитавања на контиге; прочеље ***IReadContigMapper*** представља радни оквир пресликавања, за шта је конкретно одговоран метод Map
- класа ***MatePairMapper*** је реализација претходно поменутог прочеља; метод Map додатно позива још неке методе ради формирања сравњења очитавањима и контига, на основу података похрањенх у DeBrujin-овим чворовима.

Паралелизација:

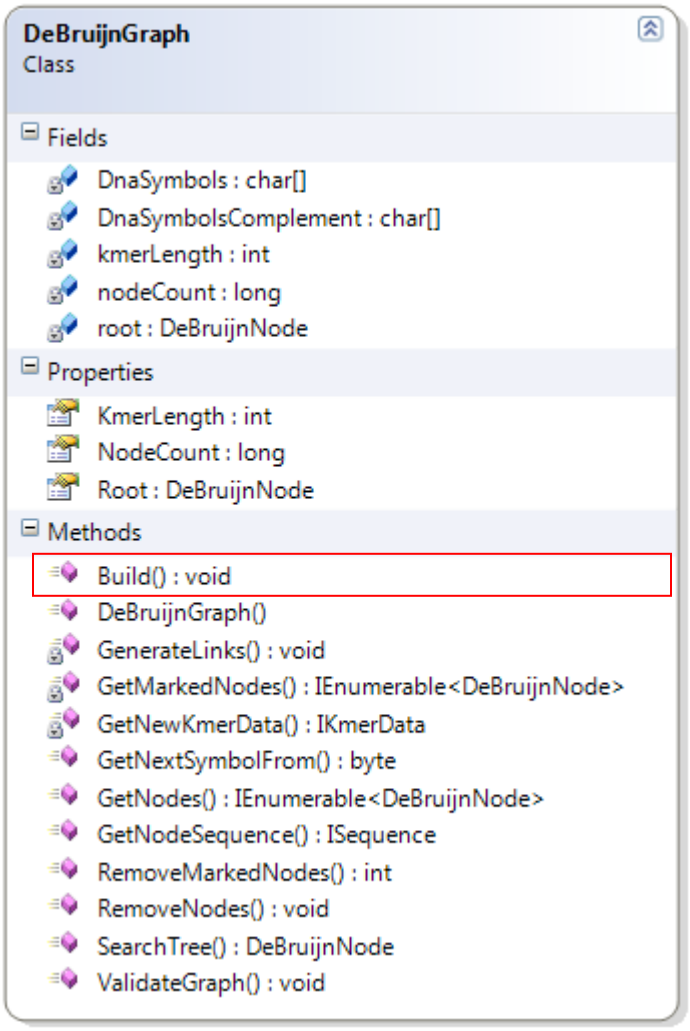
Пресликавање контига на очитавања је засебно за сваки контиг.

Корак 3: Измјена Графа:

Улаз:

`IList<ContigData>`: улазна листа контига

Диаграм класе:



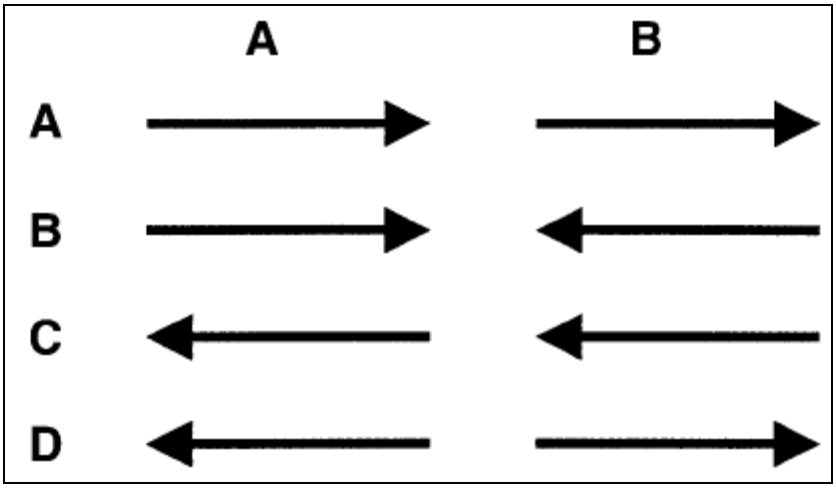
Метод BuildContigGraph мијења постојећи граф у граф контиг-преклапања. У овој фази, сваки чвор графа представља један контиг.

Паралелизација:

Чворови графа се ажурирају независно један од другога, а потом се стапају и формирају контиг-чворове.

Корак 4: FilterReadsBasedOnOrientation

За сваки пар контига, mate-pair-ови се групишу на основу њиховог усмјерења (четири могућа груписања, као на слици испод). Груписање са највећим бројем mate-pair-ова бива искориштено у наредним корацима, а остали mate-pair-ови бивају уклоњени из разматрања. Уколико је груписање са највећим бројем mate-pair-ова испод прага (кориснички дефинисаног: Default = 2), ниједан крак контига неће бити формиран и сви mate-pair-ови биће уклоњени (из даљег разматрања). Уколико два груписања имају једнак број mate-pair-ова и оба пролазе „цензус”, онда се ће бити задржани mate-pair-ови из обе групације, за касније разматрање.



Четири могућа усмјерења конгиг-парова [2]

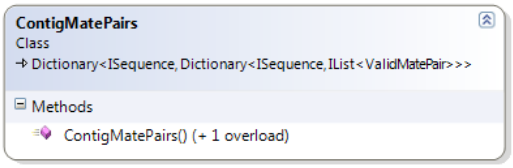
Улаз:

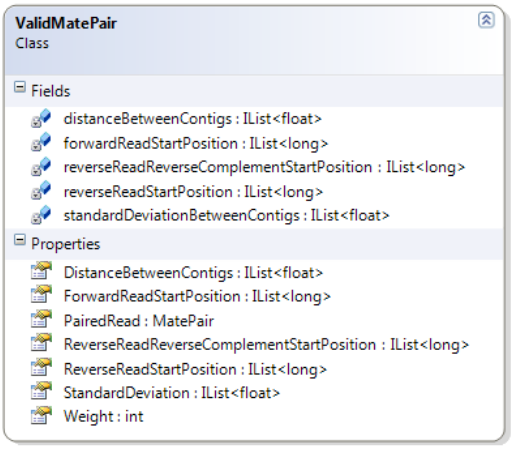
`ReadContigMap` и `IList<matePair>`:

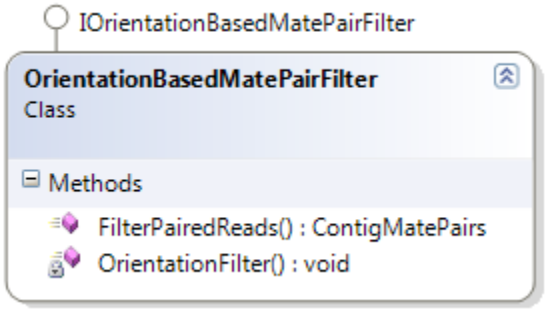
Излаз:

`ContigMatePairs`

Диаграм класе:







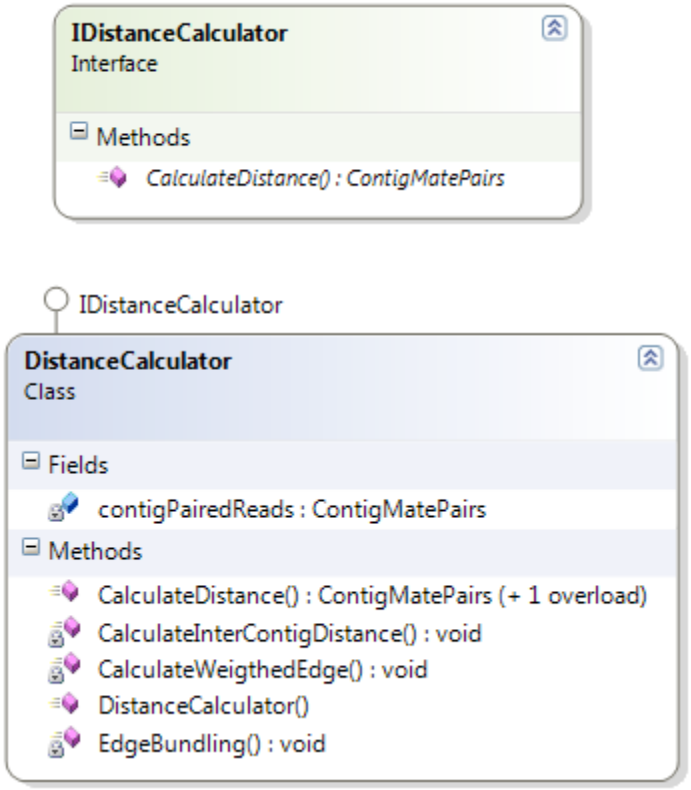
- Прочеље ***IOrientationBasedMatePairs*** представља оквир за пресликавање упарених очитавања на контиге, као и њихово просијавање засновано на усмјерењу mate-pair-ова.
- Класа ***OrientationBasedMatePairFilter*** представља реализацију претходно поменутог прочеља. FilterPairedReads метод омогућује пресликавање и просијеавање.

Паралелизација:
Пресликавања и просијавања се врше засебно за сваки пар контига.

Корак 5: CalculateDistanceBetweenContig

Улаз:
`ContigMatePairs`

Диаграм класе:



- Прочеље ***IDistanceCalculator*** обезбјеђује оквир за удаљености упарених очитавања у односу на контиге.
- Класа ***OrientationBasedMatePairFilter*** представља реализацију претходно поменутог прочеља. Садржи дефиницију метода за израчунавање удаљености између контига на основу спојева међу упареним очитавањима.

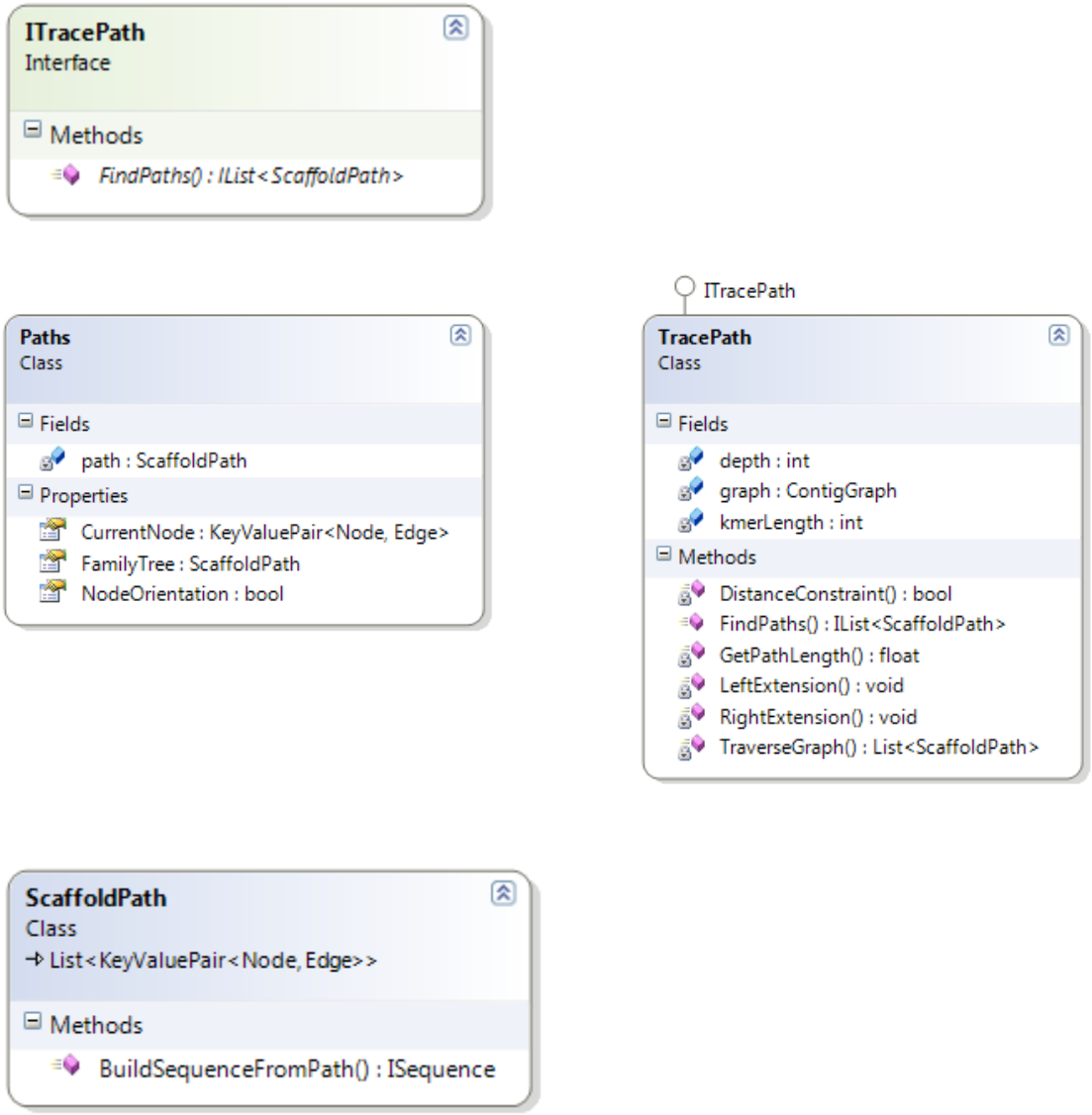
Паралелизација:
За сваки пар контига независно се израчунавају међусобне удаљености.

Корак 6: TracePath

Улаз:
`ContigMatePairs and DeBruijnGraph`

Излаз:
`IList<ScaffoldPaths>`

Диаграм класе:



- Прочеље ***ITracePath*** обезбјеђује радни оквир за обилазак контигно-преклапајућег графа користећи спојеве упарених очитавања међу контизима.
- Класа TracePath је реализација претходно поменутог прочеља. Метод FindPaths је одговоран за проналажење путева у графу. Одабрана алгоритамска стратегија је претраживање по дубини. Резултујућа вриједност метода је листа одговарајућих путева.
- Paths је унутрашња класа која која служи за похрањивање информација о путу током поменутог претраживања у дубину.

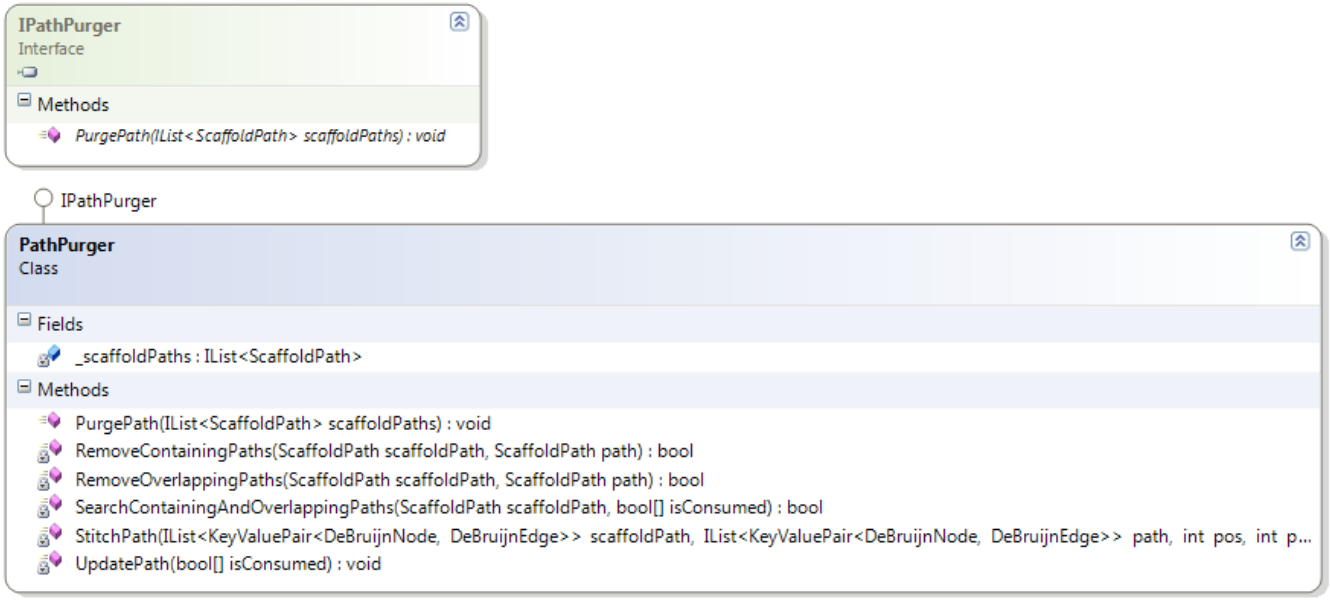
Паралелизација:

Свако дубински-оријентисано претраживање се врши независно од осталих, почевши од чворова са регуларним контизима.

Корак 7: PathPurger

Улаз:
`IList<ScaffoldPaths>`

Диаграм класе:



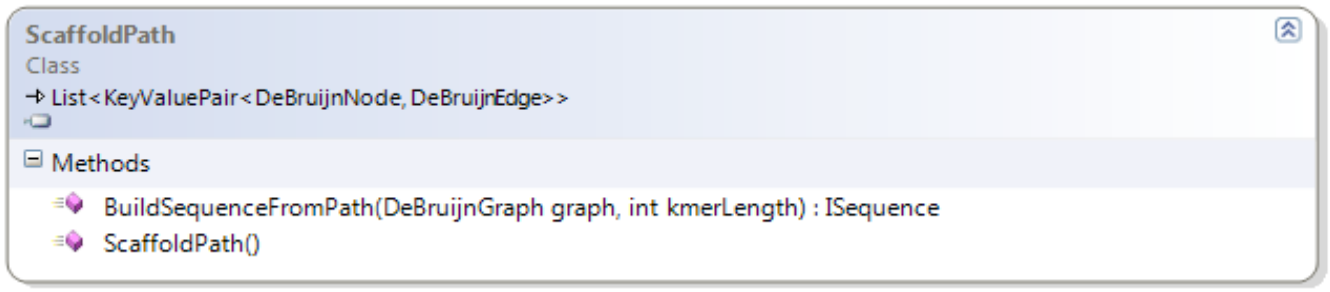
- Прочеље ***IPathPurger*** представља радни оквир како за уклањање сувишних путева, тако и за њихово међусобно надовезивање.
- Класа PathPurger представља реализацију претходно поменутог прочеља, при чему главну одговорност сноси метод PurgePath.

Корак 8: Формирање суперконтига

Улаз:
`IList<ScaffoldPaths>`

Излаз:
`IList<ISequence>`

Диаграм класе:

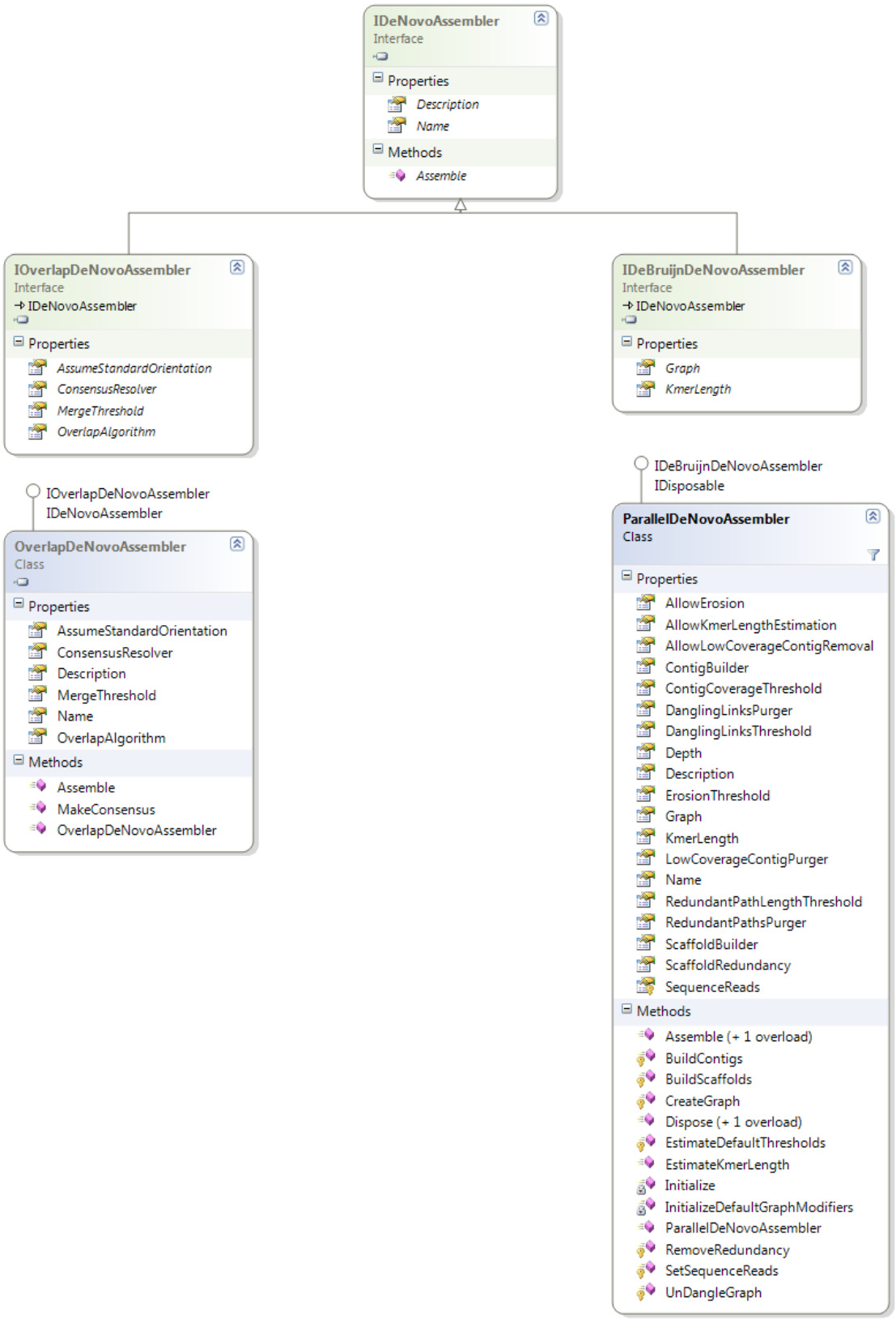


BuildSequenceFromPaths: гради секвенцу на основу путева користећи контиг-секвенце чворова (на датом путу).

Излаз:

Структура излаза је веома слична организацији структуре Assembler-а (састављача).

Namespace Bio.Algorithms.Assembly



На врху је прочеље за излаз свих de novo састављача – IDeNovoAssembly; резултат је списак састављених секвенци. Само прочеље је изведено из два прочеља која представљају излазе за двије врсте састављања:

1. IOverlapDeNovoAssembly: прочеље за излазе састављања заснованих на концепту преклапање-диспозиција-усаглашеност.

- додатно даје листу контига (заједно са информацијама о стапању за сваку контиг-секвенцу) и списак секвенци које нису стопљене.
- реализовано кроз програмску класу ***OverlapDeNovoAssembly***, што је резултат простог састављања секвенци заснованог на преклапањима (претходно реализовано кроз **`OverlapDeNovoAssembler`**).

2. IDeBruijnDeNovoAssembly: прочеље за излазе састављања на основу de Brujin-ових графова

- даје листу контига и листу суперконтига.
- програмска класа ***PadenaAssembly*** описује излаз (резултат) Padena процеса.

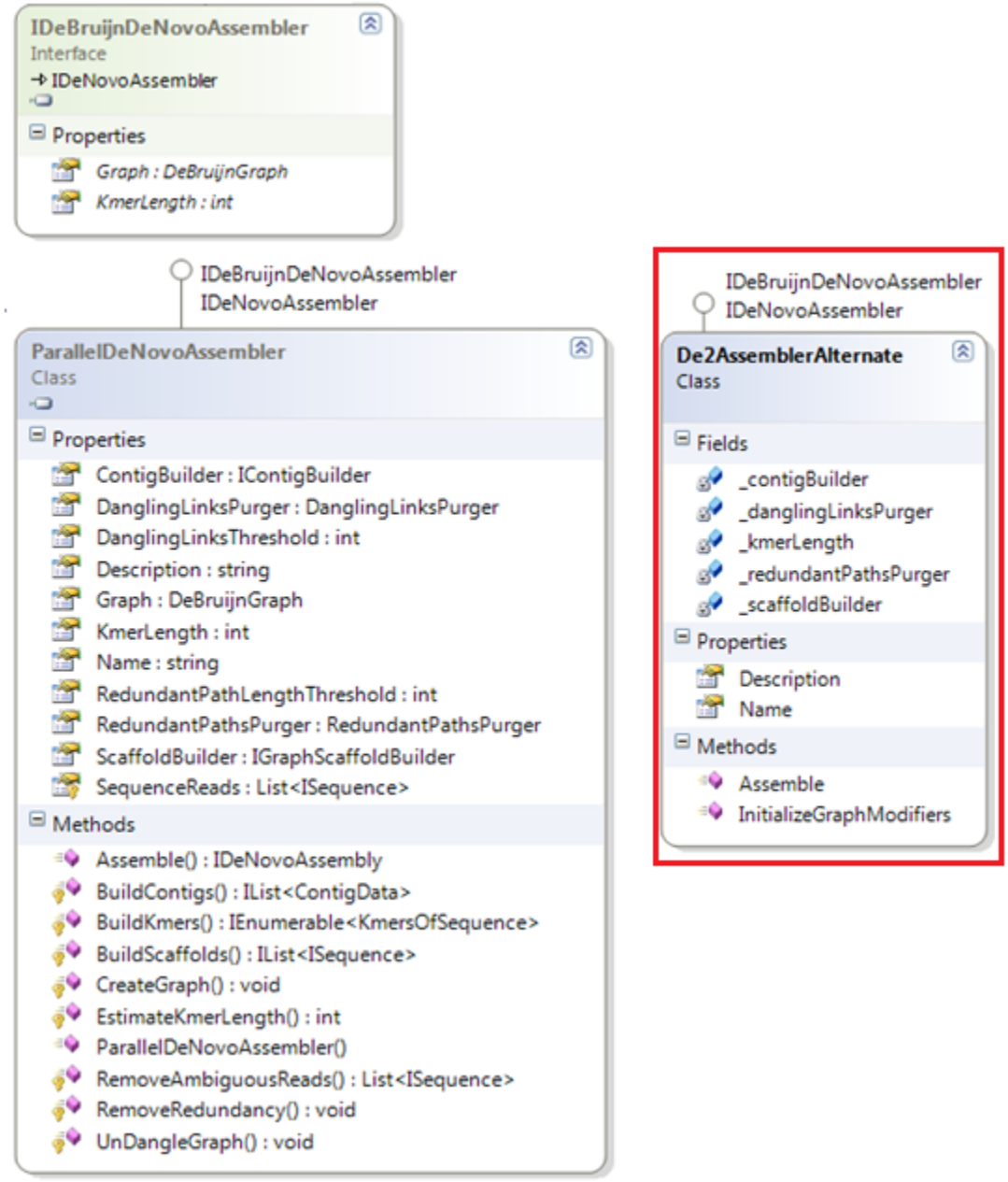
Додатак

Када имамо потпуну реализацију de novo састављача, могуће је додати и неке алтернативне имплементације претходно описаних корака. Овај plug-and-play модел за различите кораке даје кориснику неке додатне могућности. Дијелови дијаграма истакнути црвеном бојом указују на то како се алтернативне имплементације уклапају у текући пројекат.

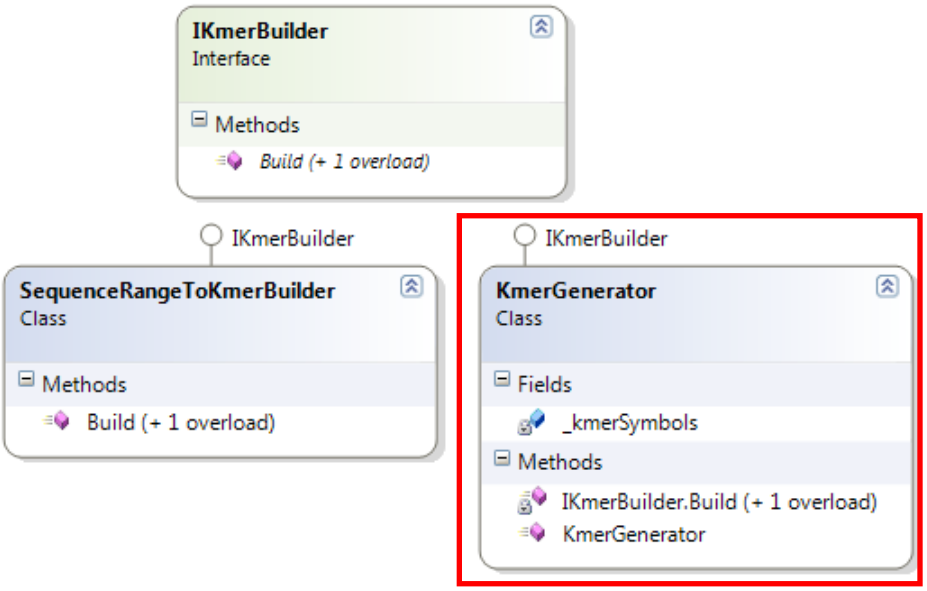
На примјер, размотрићемо спровођење корака сличних онима који су описани у публикацији која се тиче Velvet алгоритма. Velvet је у ствари користи различите алгоритме за отклањање грешака, а такође комбинује кораке у процесу састављања на један мало другачији начин. Као што је већ речено, дијаграми у наставку приказују могућности проширивања текућег пројекта с циљем премошћавања поменутих разлика.

Следећа слика (class De2AssemblerAlternate) у ствари говори о томе како пројекат може бити проширен новим алгоритмом који захтјева одређену комбинацију корака, или пак мало другачији редослијед.

class De2AssemblerAlternate



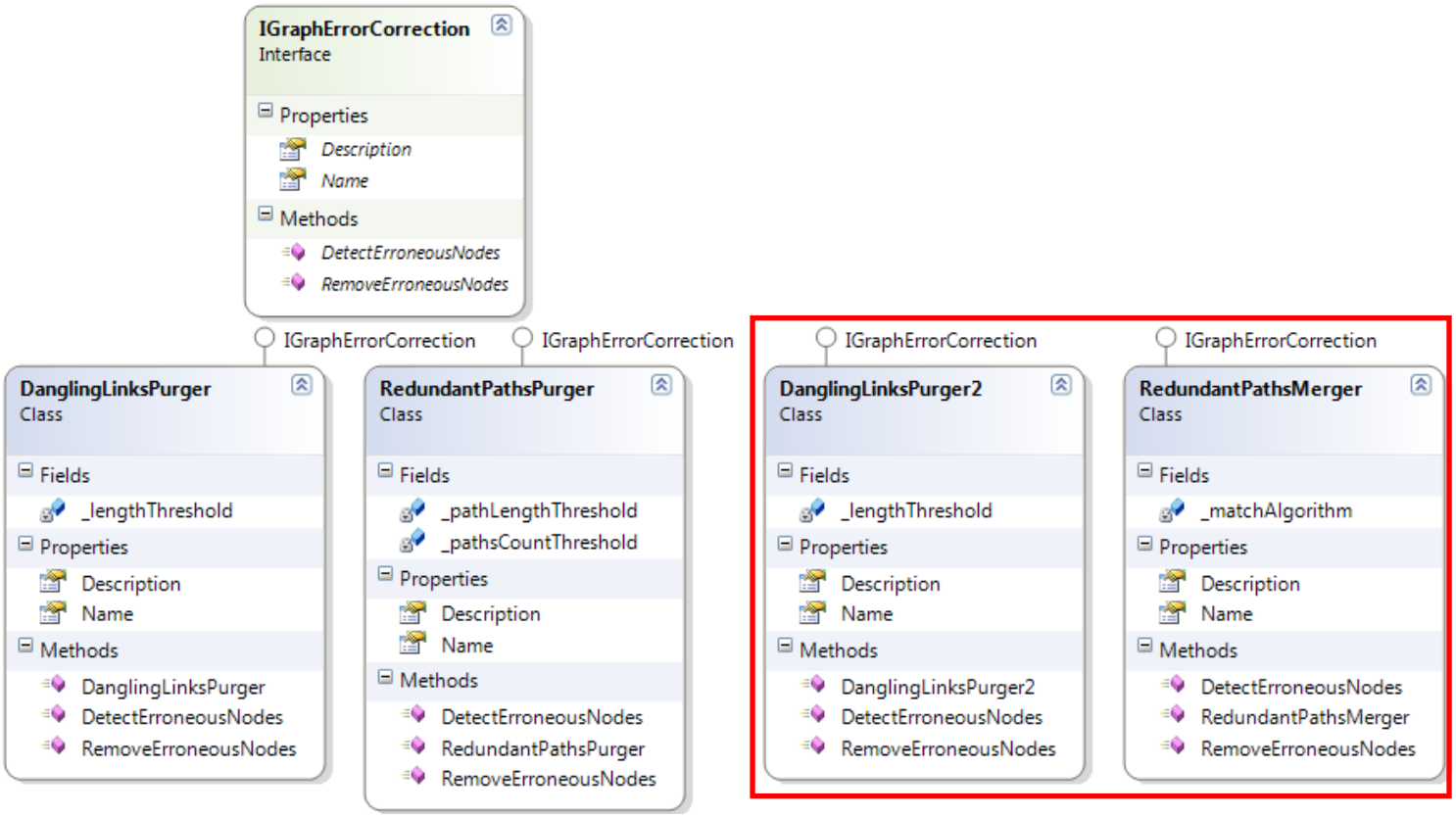
Алтернативна имплементације се може обезбједити, нпр. током формирања k-mer-а. Један веома популаран приступ у овом кораку је генерисање, и то на самом на почетку, цјелокупне листе свих могућих k-mer-а одговарајуће дужине. За ДНК треба генерисати (4^l) ниски (где је l дужина k-mer-а). Послије тога слиједи провјеравање егзистенције и позиција појављивања датих k-mer-а у улазним секвенцама



.

Што се метода за отклањање грешака тиче, Velvet је заснован на другачијем приступу од ABySS-а. На примјер, користи 2-по-2 алгоритам за препознавање сувишних путева или пак њихово надовезивање.

Следећа слика приказује како се неке друге реализације могу убацити међу поменуте механизме за отклањање грешака у графу. На исти начин се може додати и искористити било који други механизам за отклањање грешака.



Референце

1. K. Scheibye-Alsing, S. Hoffmann, A. Frankel, P. Jensen, P. F. Stadler, Y. Mang, N. Tommerup, M. J. Gilchrist, A. B. Nygård, S. Cirera 2009. Sequence assembly. Computational Biology and Chemistry, Vol. 33, No. 2. (April 2009), pp. 121-136.
2. Mihai Pop, Daniel S. Kosack and Steven L. Salzberg. Hierarchical Scaffolding with Bambus. Genome Res. 2004 14: 149-159
3. Imelfort M, Sequence Comparison Tools, In: Applied Bioinformatics. Ed Edwards D, Hanson D and Stajich J. Springer October 2008.
4. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJ, Birol I. ABySS: A parallel assembler for short read sequence data. Genome Research, 2009-June
5. Daniel H. Huson, Knut Reinert, Eugene Myers 2001. The greedy path-merging algorithm for sequence assembly. RECOMB '01: Proceedings of the fifth annual international conference on Computational biology (2001), pp. 157-163.
6. D. Hernandez, P. François, L. Farinelli, M. Osteras, and J. Schrenzel. 2008. De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. Genome Research. 18:802-809, 2008.
7. Chaisson MJ, Pevzner PA, "Short read fragment assembly of bacterial genomes", Genome Research Jan 2008.
8. D.R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Research 18:821-829.

Демонстрација могућности .NET Bio Framework-а на језику IronPython

Version 1.0 - June 2011

Сажетак

.NET Bio Framework је .NET библиотека са јавно доступним изворним кодом, као и биоинформатичко апликацијско програмско прочеље.

Овај документ представља анализу једног од примјера – BioDemo.py, који демонстрира неке од могућности Framework-а. Програмска реализације је на рачунарском програмском језику IronPython. За више информација о развоју апликација заснованих на Framework-у, али на неком другом рачунарском програмском језику, погледајте „.NET BioProgramming Guide" на сајту CodePlex или под ..\.NET Bio\Doc директоријумом.

.NET Bio Framework је доступан на адреси http://bio.codeplex.com.

Увод

.NET Bio Framework је .NET библиотека са јавно доступним изворним кодом, као и биоинформатичко апликацијско програмско прочеље. Концепт .NET Bio Framework-а омогућује проширивање, континуирану употребу, као и могућност да програмерска заједница допринесе његовом развоју у склопу Open Source Иницијативе (OSI).

Један од основних циљева је да се за пројекат заинтересује биоинформатичка заједница, чиме би се створили бољи услови за разумијевање разних техничких проблема као што су рачунарско моделовање, проширивост, развој софтвера, и многи други.

Молимо вас да повратну информацију у вези овог пројекта доставите на http://bio.codeplex.com.

Framework апликације могу бити реализоване на мноштву .NET језика, укључујући C#, F#, Visual Basic® .NET, и IronPython. IronPython је варијанта програмског језика Python, са јавно доступним изворним кодом, и веома добро интегрисана у .NET Framework-у. IronPython може користити .NET Framework и Python библиотеке, а што је најбоље и други .NET језици могу користити исто тако лако код написан на Python-у. Инсталатер за IronPython је доступан на адреси http://ironpython.codeplex.com/.

Генерално, документ даје корисне инструкције у вези са употребом IronPython-а и даје преглед једног од укључених примјера коришћења Frameworka – BioDemo.py. За више информација о развоју апликација заснованих на Framework-у, али на неком другом рачунарском програмском језику, погледајте „.NET BioProgramming Guide" на сајту CodePlex или под ..\.NET Bio\Doc директоријумом.

Такође, можете радити са секвенцама користећи два оруђа укључена у пројекат: .NET Bio Extension за Excel (додатак за Microsoft Excel) и .NET BioSequence Assembler (једна од .NET апликација).

За више информација погледајте сљедеће документе на сајту CodePlex или под ..\Bio\Doc директоријумом:

.NET Bio Programming Guide

.NET Bio Sequence Assembler: User's Guide

.NET Bio Biology Extension for Excel.

Коришћење IronPython Samples-а

IronPython је варијанта програмског језика Python са доступним кодом, који је веома добро интегрисан у сам .NET Framework. Омогућује коришћење .NET Framework и Python библиотека, при чему је омогућено њихово коришћење и

преко других .NET језика. Инсталатер за IronPython је доступан на адреси http://ironpython.codeplex.com/.

BioDemo.py је укључен у пројекат, при чему он демонстрира неке његове неграфичке (non-GUI) карактеристике.

Библиотека Bio.IronPython.dl

BioIronPython.dl обезбјеђује елегантан Python-ичан рад:

- отварање и снимање датотека са секвенцама, било ког формата којег подржавају парсери, користећи **BioIronPython.IO** модул
- насумично цијепање секвенци, помоћу **BioIronPython.Util** модула
- састављање секвенци, помоћу **BioIronPython.Algorithms** модула
- BLAST претраживања, помоћу **BioIronPython.Web** модула
- приступ C# пројектном коду, помоћу **BioIronPython.Util** модула.

Демо: BioDemo.py

Сад ћемо проћи кроз све дијелова BioDemo.py кода.

1. Важне препоруке за иницијализацију.

```
# Copyright Outercurve Foundation. All rights reserved.
import clr
import sys
import time
import os
from os import path

# Adding the dll reference will throw an exception if we're debugging in VS from the
Python
# development dir, instead of the standard non-dev method of running from the bin\Debug
dir or an
# installation dir.
try:
    clr.AddReferenceToFile("Bio.IronPython.dll")
except:
    default_filename = "bin\\Debug\\Small_Size.gbk"
else:
    default_filename = "Small_Size.gbk"

from BioIronPython.Algorithms import *
from BioIronPython.IO import *
from BioIronPython.Util import *
from BioIronPython.Web import *

build_dir = "bin\\Debug"

def deploy_file(filename):
    "Copies a file to the bin\Debug folder, replacing any file of the same name already
there."
    new_filename = build_dir + "\\" + filename[filename.rfind("\\") + 1 :]
    try:
        if File.Exists(new_filename):
            File.Delete(new_filename)
    except:
```

```
        # don't worry about replacing read-only files that we can't delete
        pass
    else:
        File.Copy(filename, new_filename)

try:
    # make build dir if needed
    if not path.exists(build_dir):
        os.mkdir(build_dir)

    # copy test file
    deploy_file("Data\\Small_Size.gbk")
except:
    print "An error occurred: " + `sys.exc_info()` + "\n"
    raw_input("Press enter to exit: ")

again = "y"
```

2. Захтјев за уносом назива датотеке са секвенцама.

   Формат може бити било који од подржаних, али који би, ипак, требали да садрже бар нешто (додатних) података о првој секвенци.

```
print "Welcome to the Bio IronPython Demo!"

while "yY".find(again[0]) != -1:
    try:
        # parse file
        filename = raw_input("\nPlease enter a sequence filename (defaults to " +
default_filename + "): ")
        if filename == "":
            filename = default_filename
        seq = open_seq(filename)[0]

        print "\nSuccessfully loaded sequence!"
        print "    ID     = " + seq.ID
        print "    Length = " + `seq.Count` + "\n"
```

3. Учитавање прве секвенце из датотеке.

   Приказивање ID-а и дужине секвенце.

```
        if seq.Count >= 500:
            # create fragments
            fragments = split_sequence(seq.Range(0, 500), 10, 50)

            print "A subsequence consisting of the first 500 nucleotides or amino acids
has been split into",
            print `len(fragments)` + " fragments, each of length 50."
            print "These will now be reassembled!  (This may take a minute.)\n"
```

4. Насумично цијепање секвенце на више дијелова једнаке дужине који се (дјелимично) преклапају, уз довољну покривеност да би се секвенца могла опет реконструисати (reassembly) (10x):

   приказивање броја и дужине дијелова

   састављање дијелова у контиге и растуће уређивање контига по дужини.

   приказивање број формираних contig-а и дужине најдужег contig-а.

```
        # assemble sequence and sort contigs by descending length
        assembly = assemble_pairwise(fragments)
```

```
        contig_list = sorted(assembly.Contigs, lambda c1, c2: c2.Length - c1.Length)

        print "The fragments have been assembled into " + `len(contig_list)` + "
contigs, with",
        print `len(assembly.UnmergedSequences)` + " unmerged fragments."
        print "The longest contig has a length of " + `contig_list[0].Length` + "."
        print "Let's do a BLAST search with it.  (This may also take a minute.)\n"
```

5. Покретање BLAST-претраживања користећи најдужи контиг и табеларно приказивање резултата.

```
            # run BLAST search
            job_id = submit_blast_search(contig_list[0].Consensus)

            # wait for response
            for i in range(1, 13):
                time.sleep(5)
                result_string = poll_blast_results(job_id)
                if result_string != None:
                    result_list = parse_blast_results(result_string)
                    if result_list != None:
                        print "\nThe following results were returned:\n"
                        print "ID".ljust(40), "Accession".ljust(20), "Length".rjust(10)
                        print "-----------------------------------------------------------
-----------------"
                        for result in result_list:
                            for record in result.Records:
                                for hit in record.Hits:
                                    print hit.Id.ljust(40), hit.Accession.ljust(20),
`hit.Length`.rjust(10)
                        print
                        break
```

6. При појави грешке исписује се извјештај о њој и прелази се на корак 7.

```
            elif i % 2 == 0:
                print "No response yet after " + `5*i` + " seconds..."
else:
            print "\nNo results have been returned from the BLAST search."
            print "Giving up on job ID " + `job_id` + "\n"
       else:
                print "Input sequence must have atleast 500 basepairs."
except:
        print "An error occurred: " + `sys.exc_info()` + "\n"
```

7. Кориснику се поставља питање поводом новог покретања програма, али сад за неку другу секвенцу.

```
    # prompt to go again
    again = " "
    while "yYnN".find(again[0]) == -1:
        again = raw_input("Would you like to enter another sequence? (y/n): ")
        if len(again) == 0:
            again = " "
```

Структура solution-а

Препоручује се да унесете IronPython код у Visual Studio Bio.sln solution. Код се тада може лако модификовати и дебаговати, у комбинацији са кодом Framework-а којем приступа. Visual Studio је препоручено Microsoft-ово развојно окружење за IronPython.

Visual Studio не посједује уграђену подршку за IronPython, тј. не постоји дефинисан тип пројекта за формирање, покретање, или дебаговање .py датотеке. Проширење, звано IronPython Studio, доступно свима путем Интернета (URL: http://ironpythonstudio.codeplex.com/), обезбјеђује основну функционалност, али сами пројекти формирани помоћу IronPython Studio-а имају неких битних недостатака:

- употреба ових пројектних типова би онемогућила потпуно отварање Framework solution-а без претходног инсталирања IronPython Studio-а.
- IronPython Studio је тренутно интегрисан само у верзији Visual Studio® 2008 и подржава једино IronPython 1.0; другим ријечима, може се десити да многи неријетко коришћени модули нису доступни

  Напомена: За Visual Studio® 2010, као и за касније верзије, потребно је ручно додати Bio.sln solution.
- DLL-ови засновани на IronPython Studio-у су помало мањкави када су у питању неке функционалне ствари.

Осим тога постоје и заобилазна решења која омогућавају формирање, покретање и дебаговање .py датотека без коришћења неког од уграђених пројектних типова Visual Studio-а или додавања екстензије.

Додавање IronPython пројекта у Visual Studio-у

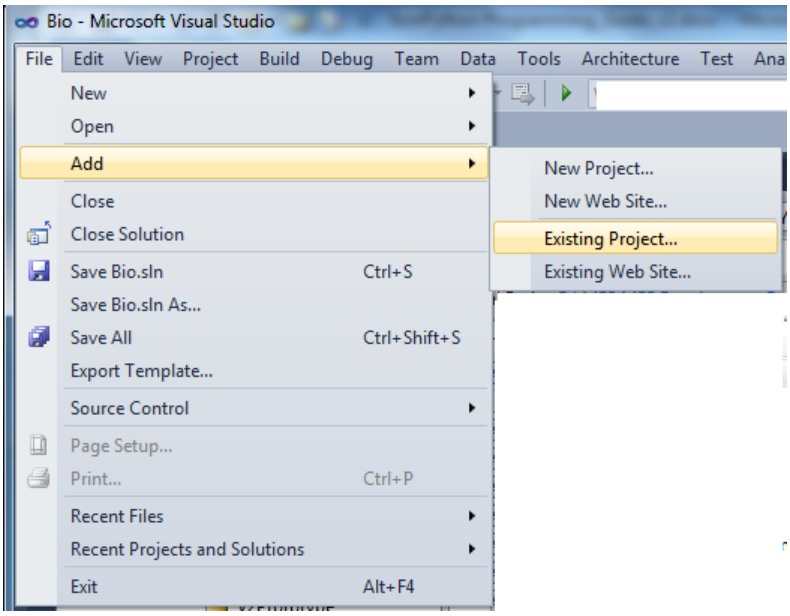
Можете унијети извршне датотеке у Visual Studio solution користећи наредбу **Add Existing Project**. Тада ваша IronPython апликација може бити дебагована на исти начин као и било који други пројекат.

Можете додати постојећи пројекат неком solution-у, а онда дорадити дати пројекат тако да испуњава захтјеве текућег solution-а.

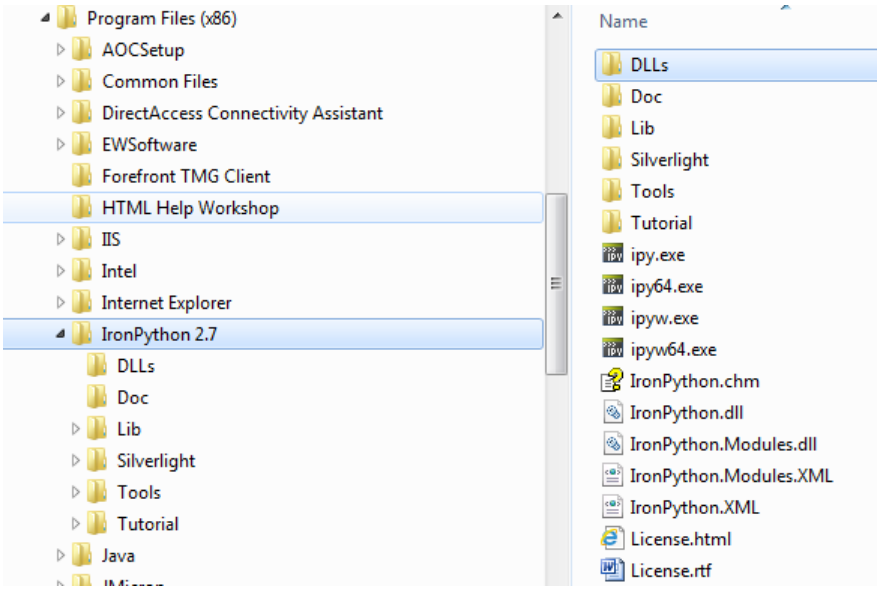
Додавање постојећег IronPython ipy.exe-а у Visual Studio solution

1. Отворити Visual Studio solution.
2. У опцији **Solution Explorer**, изабрати Bio.sln solution. Додати IronPython ipy.exe датотеку том solution-у.

   Напомена: претходно морате преузети IronPython са CodePlex-а. Погледати How to use the IronPython Samples.
3. **File → Add → Existing Project**.

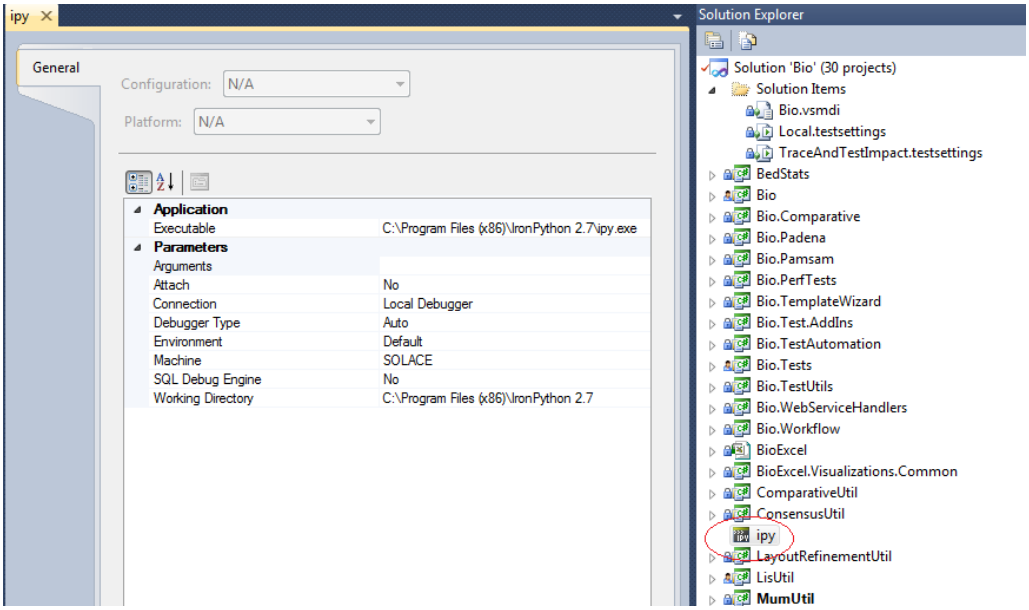


4. Пронаћи IronPython ipy.exe на мјесту гдје је инсталиран IronPython, као што је приказано на сљедећој слици и изабрати опцију додавања solution-у.



Напомена: наредбама **Add/New Project** и **Add/Existing Project** такође можете приступити десним кликом на сам solution у опцији Solution Explorer.

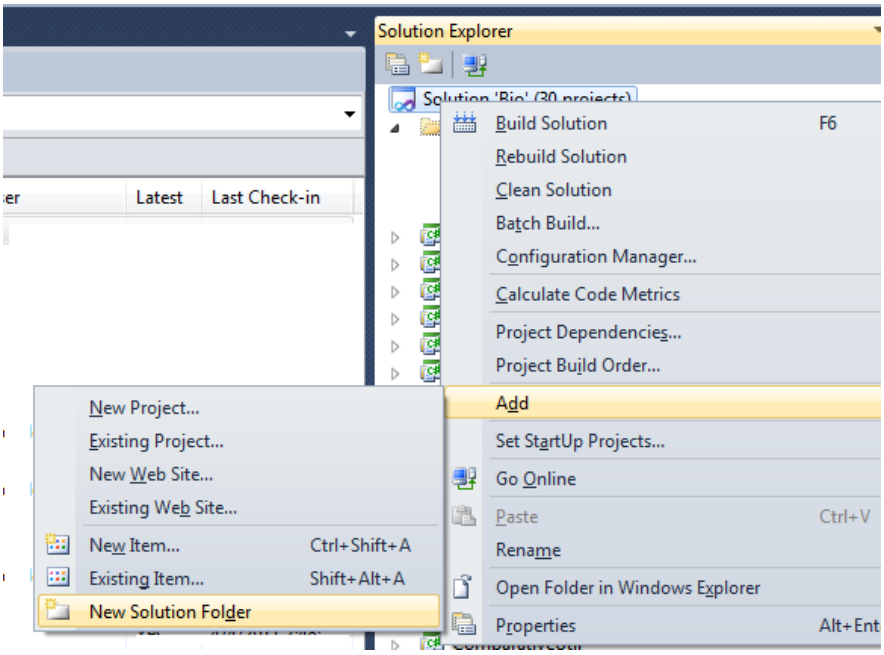
Такође, можете десним кликом на извршну иконицу приказати изборник са опцијама за промјену карактеристика пројекта, укључијући и циљ радног задатка, радни директоријум, а и аргументе командне линије.



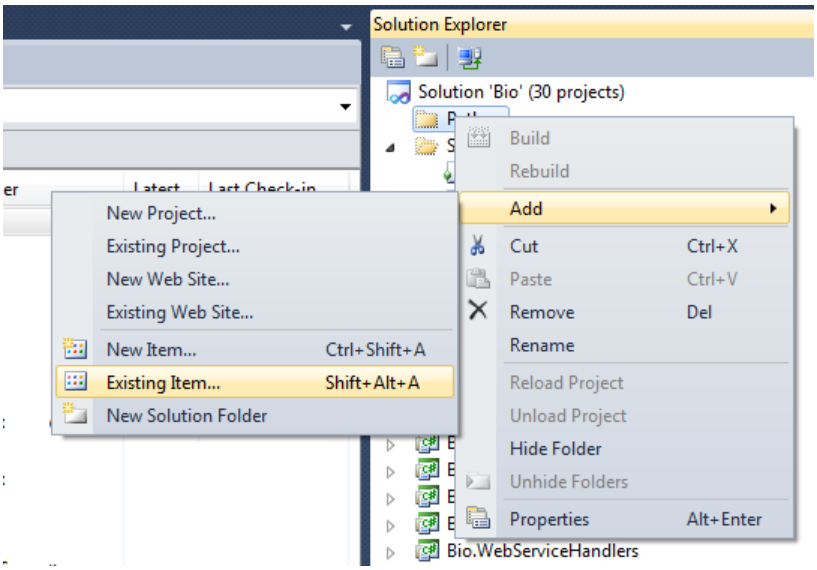
Сада додајте IronPython програмске скрипте достављене уз сам Framework. Формирајте два нова solution-директоријума у Visual Studio solution-у и попуните их IronPython датотекама.

Додавање постојећих датотека и директоријума Python-пројекта у Visual Studio solution

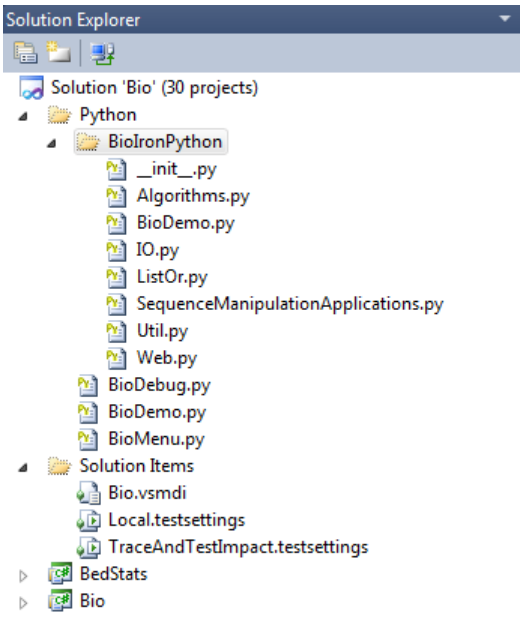
1. Десни клик на Bio.sln solution у директоријуму **Solution Items**, па преко опције/подизборника **Add** изабрати **New Solution Folder.** Назвати га „Python”.



2. Десни клик на директоријум **Python**, па преко опције/подизборника **Add** одабрати **Existing Item**.



3. Додати демо датотеке под директоријумом ..\Source\Tools\Python. Све их маркирати, а потом кликнути на **Add**.
4. Десним кликом на директоријум solution-а **Python** додати још један solution-директоријум. Назвати га „BioIronPython”.
5. Поновити корак 2 за директоријум **BioIronPython**. Додати демо текстове у изворни директоријум ..\\Tools\Python\BioIronpython.
6. Нови solution је илустрован сљедећом сликом.



Ваш нови solution сада посједује сљедеће карактеристике:

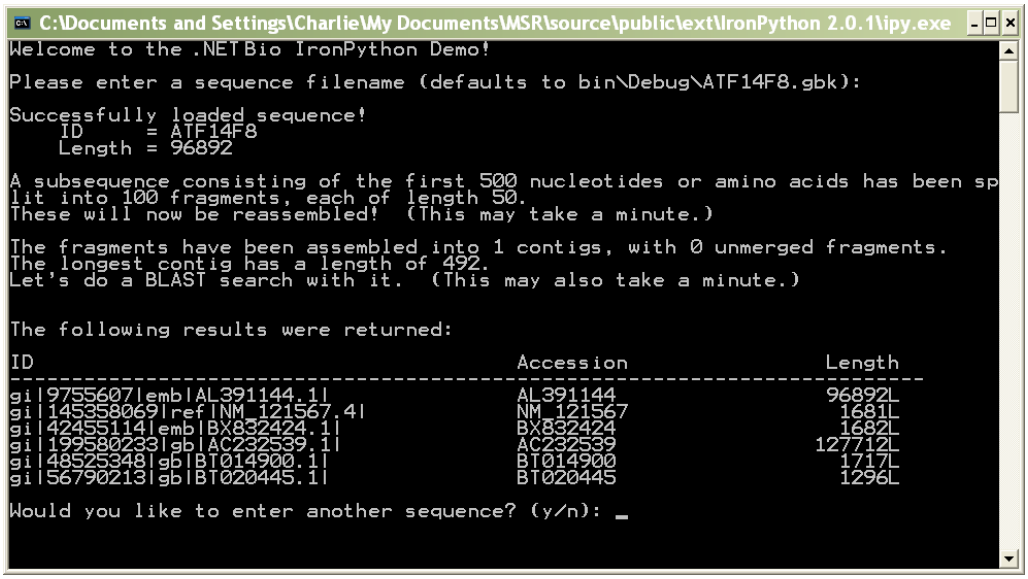
- датотеке IronPython-а су смјештене под директоријумом на истом нивоу као и C# пројекат
- демо-код је садржан у Python\BioDemo.py, модули библиотеке који обухватају BioIronPython.dll су под Python\BioIronPython, а програмске скрипте за формирање/дебаговање су у BioDebug.py

- текстовна љуска IronPython-а – ipy.exe, придружена је осталим .py датотекама
- међу својствима ipy.exe-а, радни директоријум је промијењен у директоријум Python, а аргументи су **-D BioDebug.py**. **-D** означава употребу дебагера. Други аргумент је датотека који чије извршавање треба бити позвано са командне линије
- када се ipy.exe подеси да буде startup-пројекат, BioDebug.py биће покренут кроз Visual Studio дебагер у Python-овој текстовној љусци
- покретањем програма BioDebug.py формира се BioIronPython.dll, неопходне радне датотеке се умножавају под директоријумом bin\Debug, а потом се у дебагеру покреће сам BioDemo.py, на исти начин на који се дебагује сваки други Visual Studio пројекат
- програмери који желе синтаксно обојен кода и неке другу погодности, могу инсталирати IronPython Studio, а адреса за преузиммање је

  http://ironpythonstudio.codeplex.com/.

Покретање и дебаговање кода

Демо може бити дебагован из Visual Studio-а (или из IDE-а по избору), покренут из текстовне љуске IronPython-а, или пак са командне линије CMD-а. Такође, BioIronPython.dll-у се може приступити непосредно из текстовне љуске IronPython-а. Излаз ће бити приказан као на слици испод.



Текстовно прочеље IronPython-а

Покретање дема из текстовне љуске IronPython-а

1. Да би покренули демо из текстовне љуске IronPython-а, прво умножите садржај директоријума Python\bin\Debug под свој радни директоријум, или пак пређите са текућег директоријума на Python\bin\Debug.
2. Покрените:

   >>> import BioDemo

Напомена: било која наредба на глобалном нивоу .py датотеке извршава се тек када је сама датотека учитана.

Покретање дема са командне линије

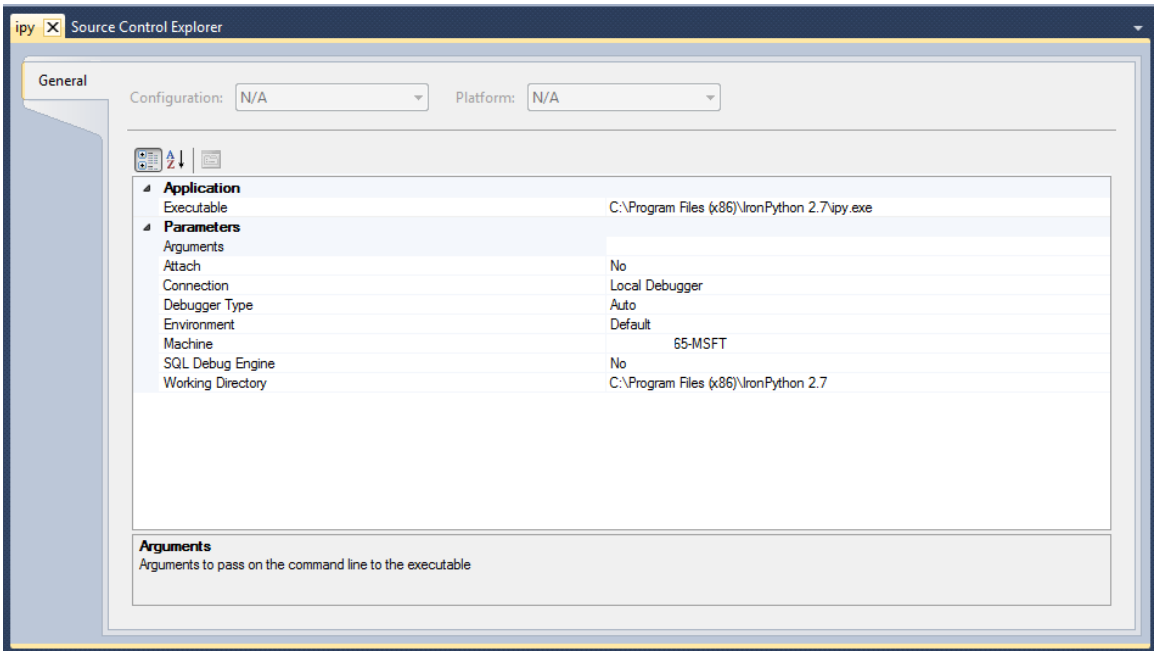
Покренути ipy.exe са пуном путањом до програма (датотеке)

Python\bin\Debug\BioDemo.py

као једини аргумент.

Дебаговање дема помоћу Visual Studio-а

1. Десни клик на иконицу **ipy.exe** Solution Explorer-а, а потом одабрати **Properties**.
2. Подесити својства према слици.



Својства ipy.exe

Напомена: радни директоријум треба да буде апсолутна путања.

3. Подесити ipy.exe као startup-пројекат и притиснути F5.

   Мјесто прекида (breakpoint) нека буде почетак кода програма BioDemo.py, ако желите да дебагер ту застане.

Напомена: када дебагујете помоћу Visual Studio-а, може вам се јавити извјештај **IronPython.Runtime.Exceptions.GeneratorExitException** при покретању програма BioDebug.py; игноришите то и притисните F5; наставиће се обрада кода као што је и предвиђено.

Дебаговање дема изван Visual Studio-а

Уколико код није исписан, урадите то постављањем ipy.exe као startup-пројекта и притисните F5.

Ако не желите да се демо покрене сваки пут када га направите, закоментаришите „**import BioDemo**” на крају кода BioDebug.py.

Извори

У овом дијелу се налазе хипервезе ка страницама са додатним информацијама о .NET Bio Framework-у и сродној тематици.

Референце које се односе на Microsoft-ов софтвер

IronPython

http://www.codeplex.com/IronPython/

Visual Studio 2010 и .NET Framework 4 Beta 2

http://msdn.microsoft.com/vstudio/

Референце на CodePlex

.NET Bio Framework

http://bio.codeplex.com/

.NET Bio Overview
.NET Bio Programming Guide
.NET BioSequence Assembler:User Guide
.NET BioParallel DeNovo Assembler technical Guide

.NET Bio Extension for ExcelUser's Guide

http://bio.codeplex.com/

.NET Bio Extension за Excel: User Guide

Sandcastle

Sandcastle - Documentation Compiler за Managed Class Libraries
http://sandcastle.codeplex.com/
Sandcastle Help File Builder
http://www.codeplex.com/SHFB

Референце за биоинформатичку проблематику

BLAST

http://blast.ncbi.nlm.nih.gov/Blast.cgi

EBI BLAST Service

http://www.ebi.ac.uk/Tools/blast2/index.html

FASTA format description

http://www.ncbi.nlm.nih.gov/blast/fasta.shtml

FASTQ format description

http://maq.sourceforge.net/fastq.shtml

GenBank

Overview
http://www.ncbi.nlm.nih.gov/Genbank/
Sample GenBank Record
http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html

GFF Specification

http://www.sanger.ac.uk/resources/software/gff/spec.html

International Nucleotide Sequence Database Collaboration

http://insdc.org

National Center за Biotechnology Information
http://www.ncbi.nlm.nih.gov

CIP - Каталогизација у публикацији
Народна и универзитетска библиотека
Републике Српске, Бања Лука

004.4:577.2(035)
577.2(035)

.NET Bio [Електронски извор] : документацијски зборник / [превео и приредио Димитрије Чвокић]. - Бања Лука : Ризница, 2015

Начин приступа (URL):
https://www.dropbox.com/s/dxr0mvaziwxpmek/NETBioZbornik.pdf?dl=0. - На насл. стр.: The Outercurve Foundation.

ISBN 978-99976-619-2-0
1. The Outercurve Foundation

COBISS.RS-ID 4709656