

به نام خدا

پروژه بيو اينفرماتيك با زبان جاوا:

برای باز کردن پروژه ابتدا باید برنامه Netbeans رو نصب کنیم. قبل از نصب باید JDK رو نصب کنیم. یک درایو دلخواه انتخاب میکنیم (در اینجا D) و صبر میکنیم تا نصب شود.

بعد از نصب JDK باید JRE هم در همان درایوی که JDK را نصب کردیم، نصب کنیم. حالا نوبت نصب Netbeans است. اگر JDK رو درست نصب کرده باشید موقع نصب Netbeans برنامه JDK را نیز شناسایی میکند. Netbeans را نیز در همان درایوی که برنامه های قبلی را نصب کردیم نصب میکنیم.

وقتی که Netbeans رو نصب کردیم گزینه ای تحت عنوان font داره که برای بهتر نمایش دادن نتیجه بهتره اونو در درایو C://windows/font کپی کنیم.

حالا برای اجرای پروژه Netbeans رو باز میکنیم.

پوشه ی bioinformaticporoject رو بهش میدیم.

بغلش یه علامت قهوه ای میاد یعنی اینکه Netbeans پروژه رو شناخته!

تو این پروژه از یه سری library استفاده شده که به صورت فایل jar هستند. قبل از اجرا باید مسیر این فایل هارو به پروژه بدیم.

برای اینکار روی گزینه add jar folder میریم و در library پوشه های مورد نظر را انتخاب میکنیم.

بین این library ها (فایل های common و Hadoop) تو قسمت ساخت گراف پروژه استفاده شده. برای

تولید گراف از یه library به نام ثابت استفاده شده که به این فایل ها نیاز داره. تو این پروژه از Hadoop استفاده نشده ولی چون بدون این فایل ها error بوجود میاد این فایل ها باید add بشن. فایل های jaligner و Myalignment

بعد از تولید گراف برای align کردن نتیجه و نشون دادن خروجی رشته DNA به صورت align شده زیر هم استفاده شده.

برای اجرای پروژه ابتدا bioinformatics.... رو انتخاب و اجرا میکنیم.

این پروژه ۳ قسمت داره:

۱- یک رشته DNA ساخته میشه. طول رشته و اینکه از بیس های A,C,T,G چند درصد باشه تو رشته وارد میشه. (مثلا هر کدوم ۲۵٪) جمع درصد ها باید ۱۰۰٪ بشه

طول رشته رو هم میدیم ۲۷.

بعد تو قسمت دوم رشته رو تیکه تیکه میکنیم به سری تیکه که طولشون بیشتر از ۵ باشه و از ۹ کمتر بشه.

تعداد کپی هم (coverage) هستن.

معمولا ۶ یا ۷ میگیریم.

رشته ی اول به ۶ تا رشته کپی میشه بعد هر کدوم به صورت تصادفی تیکه تیکه میشه. بعد تو قسمت بعدی از این تیکه ها سعی میکنیم با تولید گراف رشته رو بازسازی کنیم.

بعد از کلیک روی قسمت بازسازی رشته کاری که انجام میشه اینه که یه گراف؟؟؟(de bruijn) از تیکه های مرحله دوم ساخته میشه بعد از یه الگوریتم واسه پیدا کردن یه مسیر Euleri تو گراف استفاده شده.

مسیر Euleri مسیری که از همه ی یال ها یه بار رد میشه.

از یه راس شروع میشه و به همون راس ختم میشه.

در واقع الگوریتم به این صورت کار میکنه. حالا بعد از تشکیل گراف ممکنه بعضی اوقات مسیر کامل Euleri پیدانشه یا الگوریتم چند تا مسیر بده.

تو این مثال مسیر هایی که داده زیاده.

این قسمت حاصل مسیرهاست.

قسمت اخر هم align شده ی مسیر های پیدا شده با رشته ی اصلیه.

ممکنه تو بعضی از بیس ها هم خطا به وجود بیاد.

تو قسمت آخر رشته اول رشته ی اصلیه .

نکته: گرافی که موقع اجرا تشکیل میشه تو قسمت play که در بالای صفحه هست چاپ میشه. مثلا تو یه قسمت که میخایم رشته رو بازسازی کنیم (همون تولید گراف)

این این node ها گرافن که نوشته از کدوم نود به کدوم نود وجود داره.

تو این قسمت میخوایم قسمت ساخت گرافی که تو پروژه استفاده شده رو بگیریم.

ب این صورت که اگه رشته ی ما باشه همون رشته ای ه تو عکس هست رشته ی اول رشته های ۳ و ۲ هم کپی این رشته ها باشن

یا فرض کنیم اگه رشته به صورت تصادفی اینجوری تیکه تیکه شده باشه ینی مثلا کپی اول شده CCATTG یا تیکه دیگری شده ACCCGTTAC و تیکه سومش شده TGG ینی کپی های دیگه همینجوریه.

گراف به این صورت ساخته میشه که هر تیکه رو جداگونه در نظ میگیره مثلا تیکه اول CCATTG بعد اگر $k=3$ باشه ۳-mer، ۳-mer نود میسازه. ۳-mer اول این تیکه یال وصل میشه ازش به ۳-mer بعدی

AAT--->TTG<---CCA--->CAT

بعد ۳-mer دوم به ۳-mer بعدیش (یه خونه جلوتر)

باز دوباره همین ۳-mer AAT به ۳-mer بعدیش ینی TTG.

تا اینکه این تیکه تموم میشه.

گراف به این صورت ساخته میشه که هر تیکه رو جداگونه در نظ میگیره مثلا تیکه اول CCATTG بعد اگر $k=3$ باشه ۳-mer، ۳-mer نود میسازه. بعد مثلا تیکه ی بعدی ینی نارنجیه (ACCCGTTGAC) ۳-mer اول ACC به ۳-mer بعدیش ینی CCC

بعد تیکه های بعدی تا تموم شه تیکه ها همینجوری یال و نود ب گراف اضافه میشه. بعضی جاها بین ۲ تا یال یا نود هستن. در واقع زیاد پیش میاد که چن تا یال از یه نود به نود دیگه باشه. بعد از اینکه این گراف ساخته میشه باید توش یه مسیر euler پیدا شه (یا چند تا)

برای بخش ارائه این پروژه از فریم ورک spring mvc که یک فریم ورک جاوا با معماری mvc می باشد، استفاده شده است

پروژه نهایی روی اپلیکیشن سرور glassfish ، دیپلوی (deploy) شده است.

در پکیج com.web.configuration دو کلاس HelloWorldConfiguration و HelloWorldInitializer ساختیم که

وظیفه بوت استریپینگ و تنظیمات فریم ورک را بر عهده دارند.

در این پروژه تمام ریکوئست ها از سمت کلاینت به وسیله تکنیک ajax به سرور ارسال می شوند.

در این پروژه دو کنترلر (Controller) داریم. که یکی از آن ها AjaxController است و وظیفه رسیدگی به ریکوئست های ajax را به عهده دارد.

یعنی تمام ajax request ها به این کنترلر فرستاده می شوند تا این کنترلر آن ها را handle کند.

کلاس AjaxController حاوی ۳ متد می باشد

که هر یک از آن ها همانطور که از نامشان پیداست

وظیفه تولید خروجی مربوط به مراحل Assemble و Generate و Split را عهده دار هستند.

در پکیج `com.web.model` چهار کلاس وجود دارد که هر یک وظیفهٔ مدل کردن یک دیتا را دارند. مثلاً `getGenerate` و `getAssemble` متد `AjaxResponseBody` کلاسی است که ریسپانسی را که سه متد `getSplit` از کلاس `AjaxController` بر می‌گردانند را مدل میکند

کلاس `AjaxGenerateRequestBody`

ایجکس ریکوئستی را که به متد `getAssemble` می‌آید را مدل می‌کند.

کلاس `AjaxSplitRequestBody` همین کار را برای `getSplit` می‌کند

کلاس `SliceSession`

از آنجا که در کد برای تولید خروجی مرحلهٔ سوم نیاز به لیست آرایه‌ای (`slices`) که در مرحلهٔ دوم تولید می‌شود) بود، تصمیم گرفتیم که از `session` برای ذخیره این لیست آرایه‌ای استفاده کنیم و کلاس `SliceSession` دیتایی را که قرار است در `session` ذخیره شود (یعنی لیست آرایه‌ای از رشته‌ها) را مدل می‌کند.

در پکیج `com.web.logic` هم کلاس `bridge` وجود دارد که به مثابه پُلی میان بخش منطق برنامه یعنی پکیج‌های `alignment`, `bioinformaticproject`, `com.philiphubbard.digraph`, `com.philiphubbard.sabe` و بخش `resperesentation` (بخش) یعنی پکیج‌های دیگر می‌باشد.

خوب حالا قسمت مختلف پروژه:

که فایل شروع پروژه `Uform`

ظاهر پروژه و کدش:

کلاس هست چند - `Bioinformaticproject` داخل این پکیج

برای تولید رشته.. `Sequencegenera`:

برای تیکه تیکه کردن رشته.. `Sequencespi....`:

نوشته شده که زیاد مهم نیست thread که حالا ۲ تا هست از دومی استفاده شده که با

حالا تو کد میبینیم که کجا استفاده کردم ازش

برای تولید گراف از تیکه ها و پیدا کردن مسیر sequenceAsee

کنه بینه کدوم قسمتای رشته ی اصلی ساخته شده align برای اینکه اخر کار نتیجه رو با رشته ی اصلی

sequenceAlign.....

یه کلاس کمکی مهم نیست زیاد واسه اینکه یه عدد رندوم تولید کنه تو قسمت تولید رشته استفاده شده ازش

util

نقطه ابتدایی اجرا bioinformaticproject.java کلاس

میسازه بعد ویزیل میکنه تا منو پروژه نشون داده بشه بعد دیگه فایل اجرا میشه uiiform از کلاس object

۲ تا خط بیشتر نداره. یه

و sequenecegene بعد که فایل اجرا میشه تو این فابل با کلیک روی دکمه ها از کلاس های دیگه مثلا

اجرامیشن و استفاده میشه

اطلاعات ورودی از منو میگیره رو قسمت تولید رشته تولید کنیم کدش مشخص میشه uiiform مثلا اگه رو

قسمت طراح ۱

و ... ذخیره میکنه aperecent و طول رشته و طول متغیر ها ی actg منظور ورودی های

استفاده میکنه و اجراش میکنه sequenceGenerator بعد از کلاس

و این متغیر هارو به عنوان ورودی بهش میده

هست که اجرا میشه که sequenceGenerator نتیجه sequence و

رشته تولید شده هست که بعد تو خط بعدی تو کادر اول

ریخته میشه

رو نگاه کنیم این کلاس یه متد داره `sequenceGenerator` که بریم

که همون ورودی هارو میگیره و رشته ی تولید شدرو `generateSequence`

برمیگردونه این قسمت کد واسه یولید رشتس. توضیح دادنش زیاد طول میکشه

اگر خوندی و متوجه نشدی تو ویدیو دیگه توضیح میدم

درصد هارو میگیره تو این چند تا خط به تعداد تبدیل میکنه واسه اینکه دونه دونه یکیش رو انتخاب کنیم به صورت تصادفی

رندوم واسه تولید رندومه

رو به عنوان `ACTG` هم به تعداد طول رشته تکرار میشه تو هر تکرار یه بسته از `for` این تصادفی
برمیگردونه به رشته اضافه میکنه

`UIForm` برمیگردیم تو

بعد از اینکه رو دکمه تیکه تیکه کلیک کنیم این قسمت اجرا میشه

عین همون قبلی ورودی هارو میگیره چک میکنه حداقل تو از حداکثر کمتر باشه

اجرا میکنه رو `sequencespliter` اگر بود کلاس

رو اجرا میکرد `sequencegene....` عین قبلی که

رو اجرا میکنه `sequencep...اینجا`

ورودی هارو بهش میده

که تو مرحله قبلی ساخته شد `Sequence`

مین طول ماکس طول و تعداده کپی

رو اجرا میکنه استارت میشه بعد که تموم `thread` با `sequencep....` چون این قسمت

اجرا همیشه Onprogressfinish

از رشته ها که همون تیکه ها هستن رو برمیگردونه به اسم Array که یه

از رشته هست ریخته میشه array تو اسلایس که اون هم یه S این

بعد تو این قسمت این تیکه ها تو کادر دوم ریخته میشن

رو نگاه کنیم من بدون رو باز میکنم که کدش ...sequencespli... گه بریم

اینم یه متد داره .thread فرقی نداره فقط این با

از رشته هست رو بر میگردونه array اینارو میگیره رشته مین ماکس.... و اسلایس که

euleri قسمت اخر واسه تولید گراف و چیدا کردن مسیر

رو اگه باز کنیم تو این کلاس که بیشترش به صورت شی گراییseunceasse کلاس

هم استفاده شده library. نوشته شده از

تو این تیکه کد

میده میشه با طول های ...basicassem این کلاس یه متد داره که اول یه آبجکت از

دیگه ای هم اجرا کرد. مثالی هم که تو توضیح تولید گراف داشتیم طول نود ها ۳ بود

که مربوط به اینه که true هستش یا میتونه همین باشه یا fals مقدار سوم هم که

باشه یا نه که اینجا تو این حالت نیست multifdge گراف

یعنی اینکه بین ۲ تا نود یک یال بیشتر ایجاد میکنه

از رشته ها هستش واسه مسیر هایی که array هم مثل اسلایس یه Pathes

به ما برمیگردونه در واقع این تیکه کد تیکه هارو میگیره و یه assembler

هستن برمیگردونه که ممکنه بعضی اوقات euleri از رشته که مسیر های array

این یه عضو داشته باشه یا چندتا

بریم موس رو روی اون قسمت میگیریم euleri اگه بخوایم به قسمت تولید پاراگراف و

دکمه کنترل رو نگه میداریم و روش کلیک میکنیم

موس عوض میشه shel دکمه کنترل رو که نگه میداریم

بعد کلیک کنیم میره به اون قسمت کد

باز میشه همون کلاس که ازش تو جای قبلی یه basicassembler و کلاس

استفاده شده library ساختیم البته کد زیاد یه قسمتاییش هم از object

تو این قسمت اول تعداد نود های گراف رو حساب میکنه با این فرمول که عدد یک رو به تعداد همون طول نود که گفتیم ضبر ۲ به چپ شیفت میده بعد یه ابجکت از کلاس باشه singleedge گراف میسازیم و تعداد نود هارو بهش میده و اینکه گراف

رو اجرا میکنه اگه کنترل بگیریم روش کلیک کنیم papulatgraph بعد متد

میریم به قسمت این متد

تو این متد برای همه ی این تیکه های که ما تولید کردیم تواین حلقه همون

گفتیم انجام میده debruijn کاری که توضیح نحوه ساخته گراف

۳-mer و این قسمت به ازای هر کدوم از تیکه ها یه حلقه هست که

هارو میسازه. و تو این قسمت به ابجکت گراف یال اضافه میکنه اگه برگردیم به اون رو ایجاد کردیم بعد از اینکه ساختیمش و اجرا کردیم assembler قسمت که

اگه بریم تو این متد با همون روش کنترل و کلیک این قسمت میاد که کد هستش و کاری نداریم که چجوری کار میکنه

میسازه `eulerpath` از آبجکت های کلاس `array` ولی خوب این قسمت یه

(مسیرها `path`) از این آبجکت رو برای گرفتن `getpath` بعد با این آبجکت متد

اجرا میکنه این هم همون گراف ساخته شده بود که بهش دادیم

هستش `library` این هم کد