پاسخ سوالات تیوری پروژه‌ی مبانی بیوانفورماتیک

میلاد آقاجوهری ، احسان سلطان‌آقایی

۸ بهمن ۱۳۹۶

در این مستند به بیان قسمت‌های عملی و برنامه‌نویسی قضیه می‌پردازیم. امکان نمایش این قسمت در Rmd برای ما وجود نداشت زیرا ما از سیستم makefile برای انجام کارهای خود استفاده کرده‌ایم. دقت کنید که این سیستم بسیار حرفه‌ای و دقیق است. در واقع سیستم ما به صورت کامل اگر برای مثال یک خط از یک فایل تغییر کند تمام کارهای لازم برای بروزرسانی شدن نتایج را خود بخود انجام می‌دهد. کافیست که به پوشه‌ی Source بروید و دستور make clean را بزنید تا تمام نتایج پاک شوند و سپس با زدن یک دستور ساده‌ی make all خواهید دید که تنها با استفاده از داده‌های ژنوم‌های گونه‌ها و ماربرگ و ژن‌هایش تک تک نتایج یکی پس از دیگری ظاهر خواهند شد(حتی عکس‌ها). خود اجرا شدن این دستور در واقع یک مستند کامل از نحوه‌ی پیاده‌سازی این قسمت است اما آن رادر حال حاضر توضیه نمی‌کنیم زیرا ممکن است یک سری از کتابخانه‌هایی که در تولید نتایج استفاده کرده‌ایم در کامپیوتر شما موجود نباشد و دستور اجرا نشود و نتایج ناقص شوند اما می‌توانید پس از دیدن آن‌ها یکبار روند بالا را اجرا کنید و اگر مشکلی از لحاظ کتابخانه‌ها و میزان رم(بیش از چهار کافی است) پیش نیاید، از روند حاصل شدن نتایج لذت ببرید.

۱ یک توجه

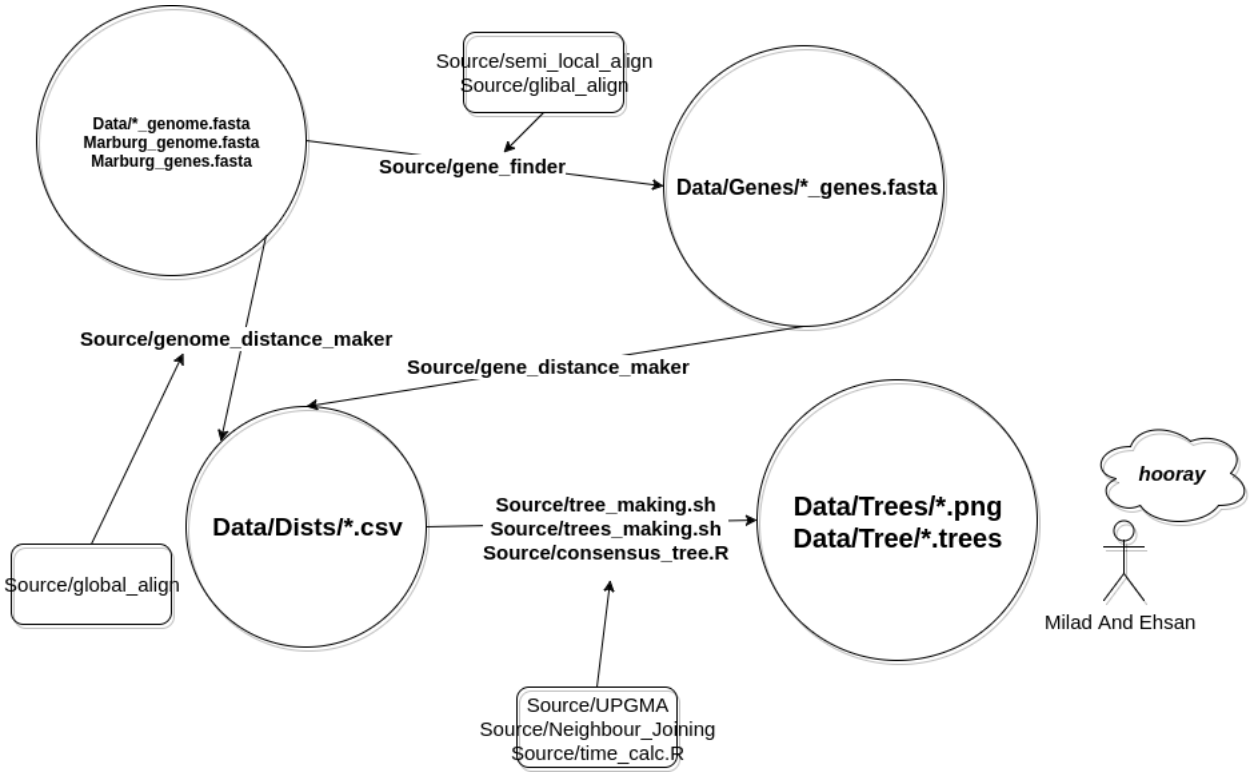
در این بخش ما تنها به توضیح بخش برنامه‌نویسی پرداخته‌ایم و از رسم نتایج خودداری کرده‌ایم. تمامی نتایج این بخش را می‌توانید در فایل theoritical_report.pdf pdf ببینید که تمامی نتایج خواسته شده را با رسم تصاویر پوشش داده است.

۲ سخنی با خواننده

می‌پذیریم که ارایه‌ی یک فایل پی‌دی‌اف و توضیح دادن یک فایل makefile به جذابیت یک فایل Rmd نیست. اما فایل‌های makefile مزیت‌های زیادی دارند زیرا خودشان نحوه و روند اجرا شدن را ذخیره کرده‌اند و در صورت هر تغییری از آن‌جا شروع به به‌روزسانی تمام نتایج می‌کنند و تمام عکس‌ها و نمودارها را بروز میکندو ذخیره می‌کنند.اما به هر حال توضیح این روند در یک فایل پی‌دی‌اف جذابیت و گیرایی خاص یک Rmd را ندارد. از شما خواهش‌مندیم از فایل‌های موجود در

بخش Data بازدید کنید که در زیر توضیح داده شده و البته نتایج حاصله را ببینید اگرچه در گزارش بخش تیوری به صورت کامل آمده است.

۳ توضیح روند اجرای کد



در ابتدا ما تنها ژنوم ماربرگ و ژن‌های آن و در پوشه‌ی Data/Genomes ژنوم گونه‌های ابولا را داریم.
در این جا تکه کد Source/gene_finder.cpp اجرا می‌شود که با بارها استفاده از Source/semin_local_align.cpp هر ژن ماربرگ را در ژنوم ابولا سرچ می‌کندو نتایج را به ازای هر گونه از ابولا به نام مثلا X در Data/Genes/X_gene.fasta می‌ریزد.

سپس تکه کد Source/gene_distance_maker.cpp از روی این فایل‌های ژن‌ها را خوانده و ماتریس فواصل را به دست آورده و در فایل‌هایی به نام Data/Dists/Y.csv میریزد که ماتریس حاصل از مقایسه برای ژنی به اسم مثلا Y است.

تکه کد Source/genome_distance_maker.cpp از روی ژنوم فواصل بین ژن‌ها را به دست می‌آورد
و در Data/Dists/genome.csv میریزد.
همین فواصل را به اضافه‌ی ماربرگ در Data/Dists/all_and_marburg.csv میریزد.
حال تکه کد Source/tree_making.sh می‌آید و از روی Data/Dists/Y.csv و با استفاده از

Source/UPGMA.cpp ، Neighbour_Joining.cpp دو فایل Data/Trees/Y_UMPGA.png و Data/Trees/Y_NJ.png را میسازد.

در نهایت Source/consensus_tree.R می‌آید و نتایج حاصل توسط NJ, UMPGA را بهم ترکیب می‌کند و در قالب دو عکس Data/Trees/UMPGA.png و Data/Trees/NJ.png در می‌آورد.

برای بدست آوردن فاصله‌ی زمانی جدا شدن این‌گونه‌ها از هم کد Source/time_calc.R با خواندن Data/Dists/all_and_marburg.csv و فرمول گفته شده در بخش تیوری ماتریس فاصله‌ای بر حسب زمان‌ها به دست می‌آوریم و سپس از UPGMA از R استفاده می‌کنیم و درخت حاصل را به نام Data/Tress/times.png ذخیره می‌کنیم. در این جا تمام خواسته‌های پروژه انجام شده است.

۴ قسمت دوم

۱.۴ استخراج ژن ها

ما در این قسمت از یک روش semi_local_alignment استفاده کرده‌ایم که کد آن در semi_local_alignment.cpp موجود است. این روش بهترین تطابق را ماتریس امتیازی که در آن شباهت امتیاز برابر با یک و تفاوت امتیاز ـ۱ و گپ امتیاز منفی یک دارد پیدا میکند gap = -1, dif = -1, equal = +1 و یک dp ساده به صورت زیر است.

```
dp[i][j] = max(dp[i-1][j]+gap, dp[i][j-1]+gap, dp[i-1][j-1]
+(genome[i]==gene[j]?equal:dif , if (j==0) 0)
```

است که البته جزییات در این کد ساده حذف شده است و میتوانید در خود فایل مشاهده کنید. اما قسمت آخر که اگر $j = ۰$ بود می‌توانیم هزینه را صفر بگذاریم دلیلش این است که می‌توانیم قسمتی از اول ژنوم را بدون دادن هزینه مپ نکنیم و بیندازیم یعنی می‌توانیم فرض کنیم اول ژنوم بدون هزینه به گپ مپ شده است تا جایی که شروع کنیم به مپ کردن به ژن.

۲.۴ هم‌ترازی ژن‌ها و به دست آوردن ماتریس فاصله

برای به دست آوردن ماتریس‌های فاصله ابتدا باید کدی داشته باشیم که فاصله‌ي ویرایشی دو موجود را به ما خروجی بدهد. البته این کد به سادگی با تغییر در همان کد قسمت قبل حاصل می‌شود، کافیست ارفاق به ازای $j == ۰$ را حذف کنیم و البته equal را نیز برابر صفر قرار دهیم. در این صورت فاصله‌ی ویرایش برابر با منفی یک ضربدر حاصل .استdp

```
dp[i][j] = max(dp[i-1][j]+gap, dp[i][j-1]+gap, dp[i-1][j-1]
+(genome[i]==gene[j]?0:dif)
```

این کد را در global_align.cpp می‌توانید مشاهده کنید. سپس کدی داریم به نام gene_distance_maker که به ازای هر جفت ژن‌ها الگوریتم بالا را اجرا می‌کند و ماتریس را تشکیل می‌دهد(در این کد بسیار کدینگ زیبایی را شاهد هستیم، بین پروسه‌ها پایپ کرده‌ایم و غیره...).

۵ قسمت سوم

۱.۵ تشکیل درخت زندگی برای هر ژن

دو الگوریتم وNJ UPGMA را الحمدلله قبلا در تمرین‌ها زده بودیم که از آن‌ها استفاده می‌کنیم و حاصل را توسط tree_drawer می‌کشیم. کدینگ این قسمت در فایل tree_making.sh موجود است و چندان نکته‌ي خاصی ندارد، صرفا دو برنامه‌ی ذکر شده اجرا و حاصلشان به tree_drawer برای کشیدن داده می‌شود.

۲.۵ ترکیب درخت‌ها و ارائه‌ی درخت نهایی

در اینجا از تابع consensus از پکیج ape در آر استفاده کرده‌ایم. این تابع را صدا زده‌ایم و سپس حاصل را ذخیره کرده‌ایم. کدینگ این قسمت را در فایل consesnsus_tree.R میتوانید مشاهده کنید و لذت ببرید.

۳.۵ مقایسه‌ی درخت ترکیبی و درخت حاصل از همترازی سراسری

در این‌جا برای همترازی سراسری ما از همان کد خودمان که در بالا توضیح دادیم که فاصله‌ی ویرایشی محاسبه می‌کند استفاده کرده‌ایم. درخت ترکیبی حاصل هم در بالا توضیح دادیم که توسط consesnsus_tree.R محاسبه می‌شود.

۴.۵ تعیین نقطه‌ی شروع

کدهای لازم برای این قسمت در بقیه‌ی بخش‌ها توضیح داده شده است.

۶ بخش چهارم

۱.۶ چه زمانی از هم جدا شده‌اند؟

قسمت‌های تیوری در گزارش بخش تیوری مشاهده شده است. در اینجا ما از روی فاصله‌هایی که در قبل حساب کرده‌ایم با استفاده از تکه کد time_calc.R آن‌ها را طبق فرمول گفته شده در بخش تیوری به ماتریس فاصله‌ی زمانی تبدیل و سپس از UPGMA استفاده کرده و در نهایت درخت حاصل را رسم و ذخیره می‌کنیم.