

۱-۲. همان طور که می بینیم ژنوم گونه های مختلف ویروس ابولا یک RNA تک رشته ای است که حدوداً از ۱۹۰۰۰ نوکلئوتاید ساخته شده است. این رشته ۷ پروتیین ساختاری را انکد میکند که عبارتند از: (NP) nucleoprotein ، polymerase ، (VP30) transcription activator ، GP ، VP24 و RNA-dependent RNA polymerase (L) [1]. ژنوم این ویروس یک پروتیین غیرساختاری هم کد می کند که soluble glycoprotein یا sGP نام دارد. [2]

با توجه به توضیحات بخش بعد آنچه از مطالعه های مختلف بدست آمد، پروتیین GP و VP24 نقش کلیدی تری نسبت به بقیه ی پروتیین ها در سرایت و تکثیر ویروس ابولا ایفا میکنند. [3][4]

۱-۳. دو کاندیدا برای پروتیین های entry در سلول میزبان وجود دارد که اولین آنها پروتیین cholesterol transporter به نام host-encoded Niemann–Pick C1 (NPC1) است که ظاهراً نقشی کلیدی در ورود ویریون های ابولا به ویروس میزبان و تکثیر نهایی آن دارد. مطالعه ای روی موش هایی که یک کپی از NPC1 را نداشته اند نشان می دهد که ۸۰ درصد از موش هایی که آلوده به ویروس ابولا بوده اند نجات یافته اند. مطالعه ای دیگر نشان می دهد مولکول های کوچکی اثر ویروس ابولا را از طریق جلوگیری از bind شدن NPC1 با viral envelope glycoprotein (GP) تضعیف میکنند. در نتیجه NPC1 باید نقش مهمی در ورود و تکثیر این filovirus دارد زیرا با bind شدن مستقیم به GP به صورت واسطه در سرایت ویروس نقش ایفا میکند.

کاندیدای دوم TIM-1 است. TIM-1 به بخش receptor binding domain گلیکوپروتیین ویروس ابولا (GP) میچسبد و قدرت پذیرش Vero cell ها را افزایش میدهد. TIM-1 در بافت ها توسط ابولا (trachea ,cornea ,conjunctiva) تحت تاثیر قرار میگیرد. یک آنتی بادی که علیه بخش IgV در TIM-1 عمل میکند از bind شدن و سرایت ابولا جلوگیری میکند. در نتیجه این مطالعات آنچه از شواهد پیداست این است که NPC1 و TIM-1 پتانسیل آن را دارند که مورد هدف داروی ضد ویروس ابولا باشند. [1]

1- [https://en.wikipedia.org/wiki/Ebola\\_virus](https://en.wikipedia.org/wiki/Ebola_virus)

2- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3094950>

3- <http://www.sciencemag.org/news/2014/08/what-does-ebola-actually-do>

4- <https://www.sciencedaily.com/releases/2014/11/141120141654.htm>

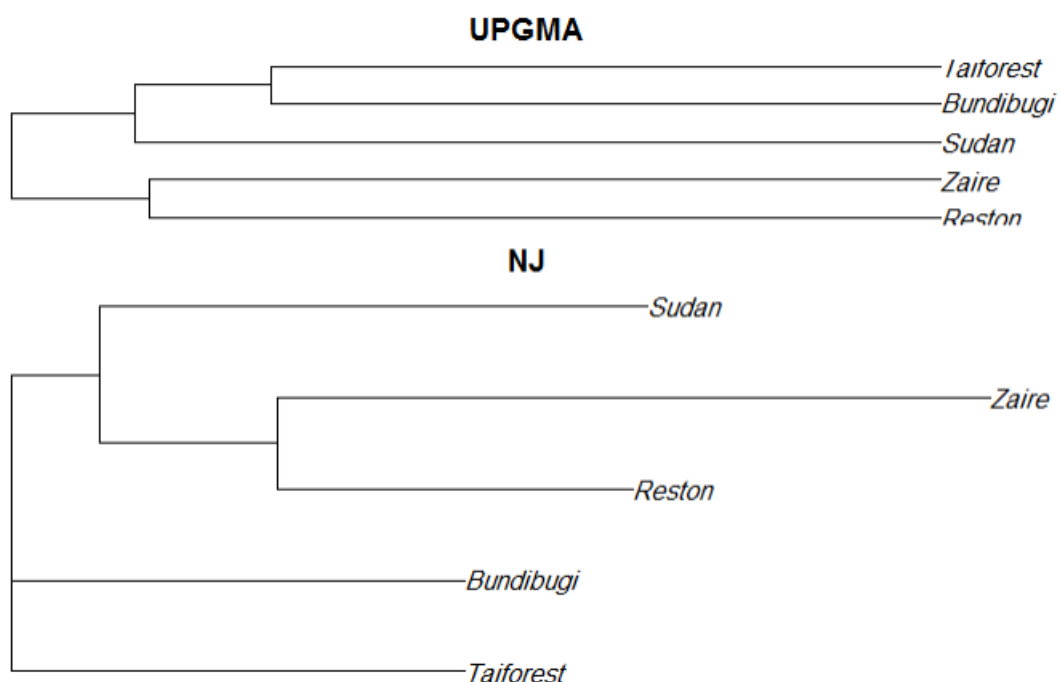
در بخش ۲-۲ با استفاده از الگوریتم هم تراز ی سراسری-محلی (myCode.py) ژن های ماربرگ را با ژنوم همه ی گونه ها هم تراز کردیم و ژن های هر گونه را استخراج کردیم که هر یک در فایل جداگانه در پوشه ی myGenes قرار دارد.

در بخش ۳-۲ با استفاده از الگوریتم نیدلمن-وونش (کد مربوط به این بخش ND\_WC.java است) ژن های بدست آمده از قسمت قبل را هم تراز کردیم و امتیازات (فاصله ویرایش) آنها را در ماتریس های جداگانه تحت نام های X\_Gene.CSV ذخیره کردیم. (که X نام ژن مربوطه است)

در بخش ۳-۱ با استفاده از کتابخانه phangorn در زبان R درخت های زندگی هر یک از ماتریس ها با استفاده از الگوریتم های UPGMA و NJ را ساختیم که خروجی کد (project.rmd) در تصاویر زیر دیده میشود:

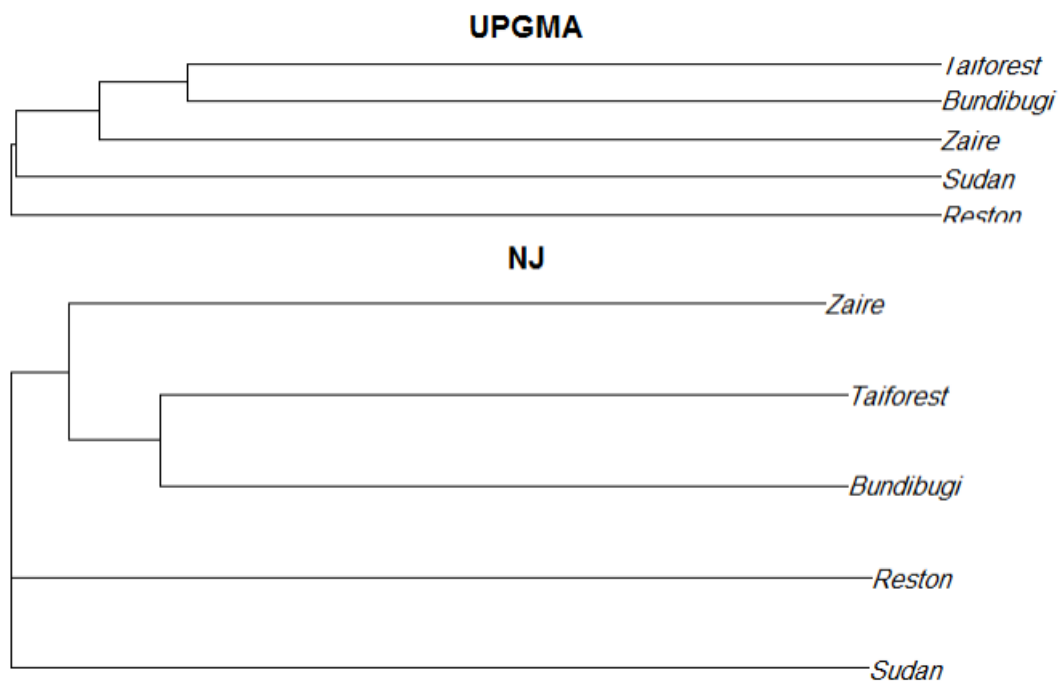
تصویر زیر مربوط به ژن GP است.

همانطور که در تصویر میبینید با توجه به این ژن گونه های Taiforest و Bundibugy و گونه های Zaire و Restton به هم شبیه ترند. درخت های بدست آمده از این دو الگوریتم در این ژن شباهت بالایی باهم دارند اما تفاوت هایی هم دارند که از جمله ی آنها میتوان به زمان جهش و جدا شدن گونه ی سودان اشاره کرد.

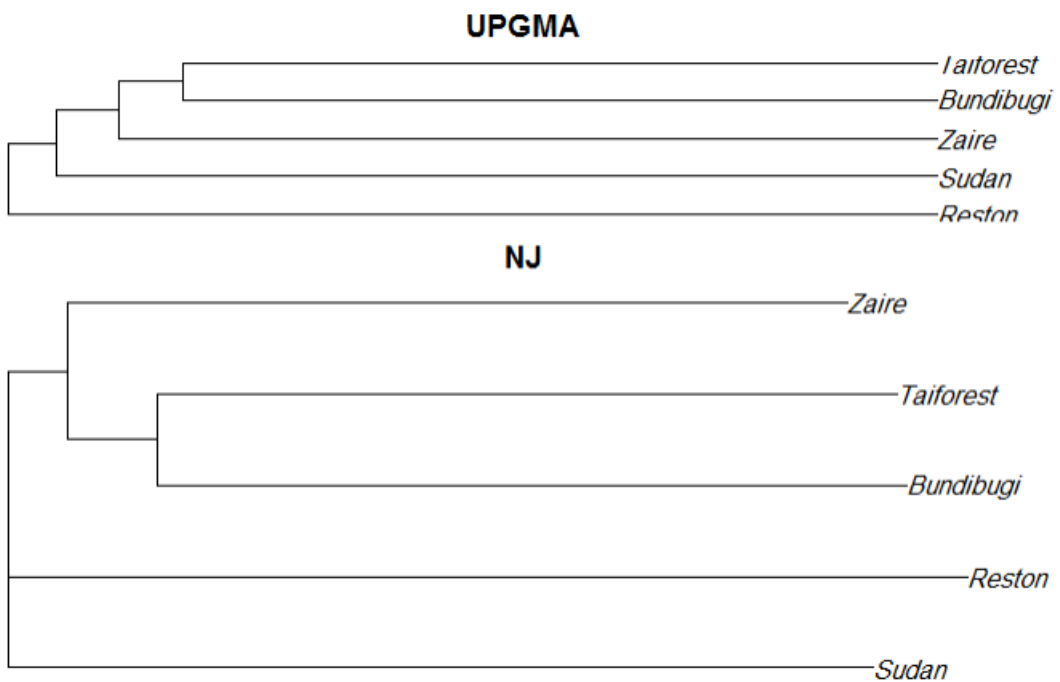


تصویر زیر مربوط به ژن L است. درخت های مربوط به این ژن هم بسیار به هم شبیه هستند و تقریباً تفاوتی ندارند تنها تفاوت آنها در زمان جهش جد مشترک سودان، رستون و بقیه ی گونه هاست که این تفاوت بسیار ناچیز است.

[Type here]

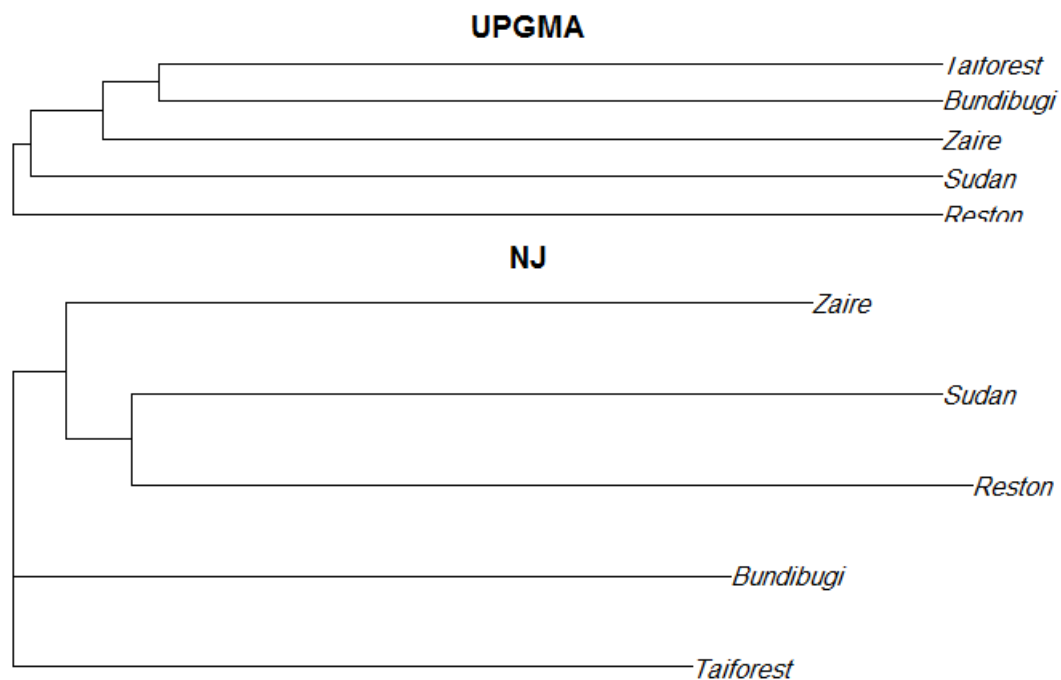


تصویر زیر مربوط به ژن NP است. شباهت ها و تفاوت های درخت های این ژن هم دقیقا مانند ژن قبلی است. به این صورت که درخت ها از لحاظ سلسله مراتبی بسیار شبیه اند و تنها تفاوتشان در زمان جهش جد مشترک سودان و رستون و جد بقیه گونه هاست.

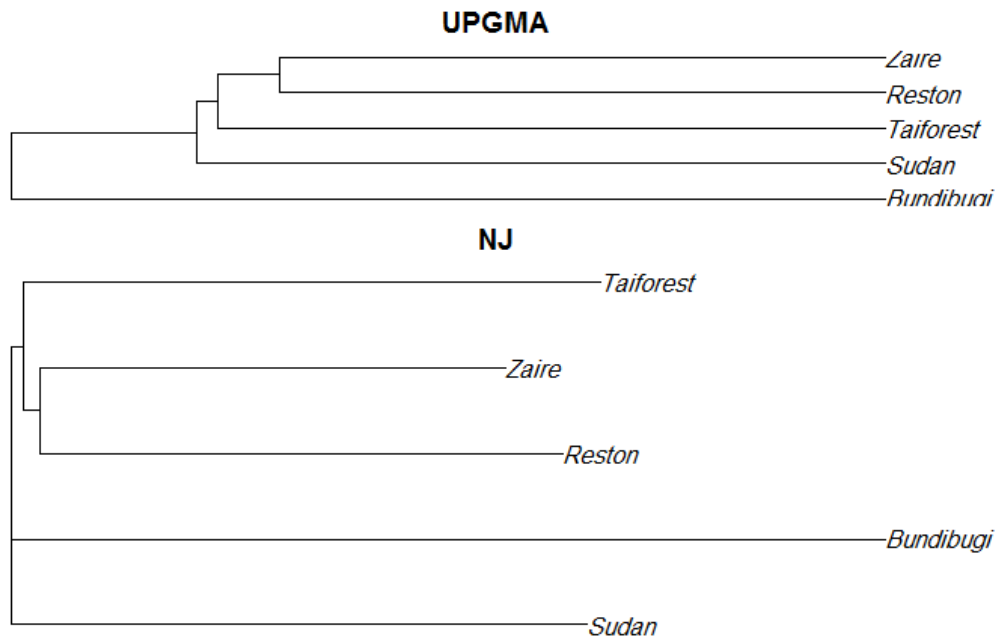


[Type here]

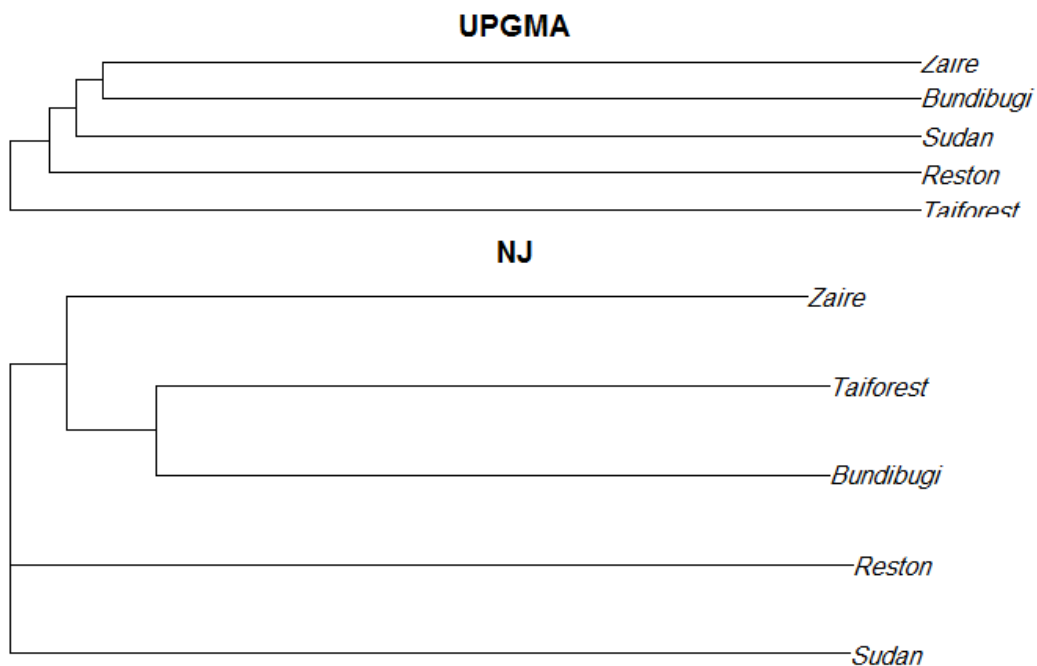
تصویر زیر مربوط به ژن VP24 است . شباهت درخت های این ژن در نزدیک بودن گونه های Taiforest و Bundibugy با هم و گونه های Reston و Sudan با هم است. اما در ترتیب سلسله مراتبی این دو درخت تفاوت وجود دارد. مثلاً در الگوریتم UPGMA جد گونه های Taiforest و Bundibugy در مرحله آخر جدا شده اند ولی در الگوریتم NJ خلاف این دیده میشود به طوریکه ابتدا این دو گونه از بقیه جدا میشوند.



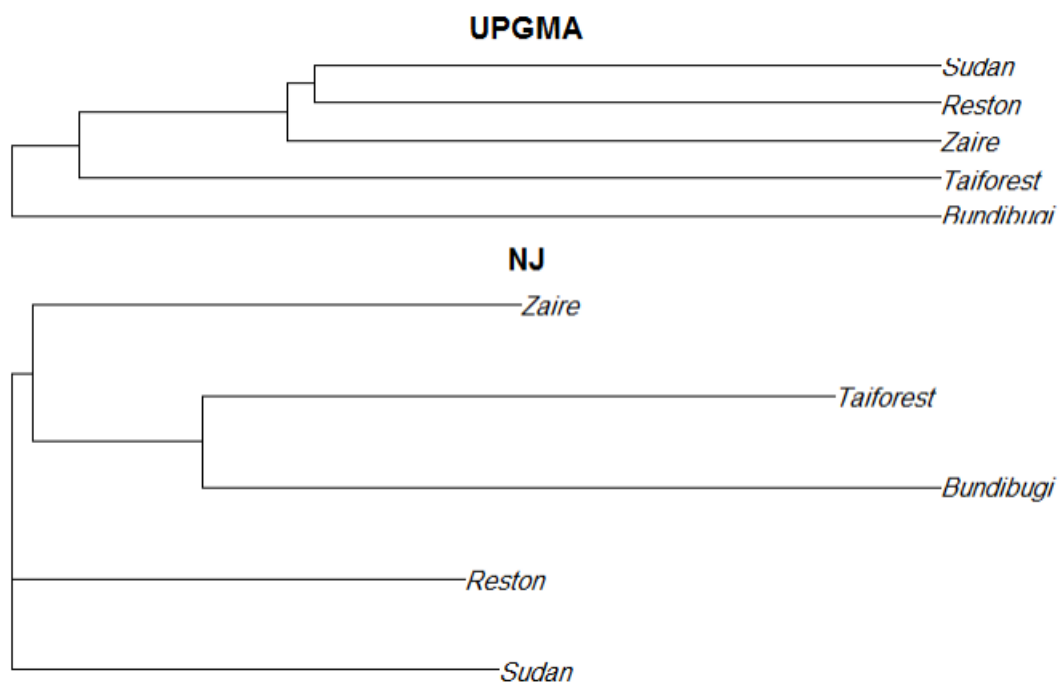
تصویر زیر مربوط به ژن VP30 است . شباهت ها و تفاوت های درخت های این ژن هم دقیقاً مانند ژن NP و L است. به این صورت که درخت ها از لحاظ سلسله مراتبی بسیار شبیه اند و تنها تفاوتشان در زمان جهش جد مشترک سودان و Bundibugy و جد بقیه گونه هاست.



تصویر زیر مربوط به ژن VP35 است. در این دو درخت نسبت به ژن های دیگر تفاوت های چشمگیری مشاهده میشود. همانطور که میبینیم در الگوریتم UPGMA گونه های Zaire و Bundibugyi به هم نزدیکترند ولی در NJ گونه ی Taiforest و Bundibugyi به هم نزدیکترند که مانند نتیجه ی دیگر ژن هاست. (با توجه به نتایج دیگر ژن ها حدس میزنم که الگوریتم NJ در اینجا نتایج قابل اعتماد تری داشته باشد)



تصویر زیر مربوط به ژن VP40 است. شباهت و تفاوت درخت های این ژن شبیه ژن VP24 است. شباهت درخت های این ژن در نزدیک بودن گونه های Taiforest و Bundibugy با هم و گونه های Reston و Sudan با هم است. اما در ترتیب سلسله مراتبی این دو درخت تفاوت وجود دارد. مثلاً در الگوریتم UPGMA جد گونه های Taiforest و Bundibugy در مرحله آخر جدا شده اند ولی در الگوریتم NJ خلاف این دیده میشود به طوریکه ابتدا این دو گونه از بقیه جدا میشوند.



تفاوت هایی که دیده میشود را در بخش توضیحات مربوط به هر ژن بیان کردیم. در حالی که شباهت های معنی داری بین نتایج وجود داشت گاهی تفاوت های چشمگیری نیز در بین آنها دیده شد که با توجه به اینکه ژن های بدست آمده دقیقاً ژن های اصلی نیستند و در گزینش آنها خطا وجود دارد انتظار چنین تفاوت هایی میرود. به علاوه این که خود الگوریتم های ساختن درخت ها هم سطح اطمینان ۱۰۰ درصدی ندارند و دارای خطا می باشند. لذا انتظار چنین تفاوت هایی را داشتیم.

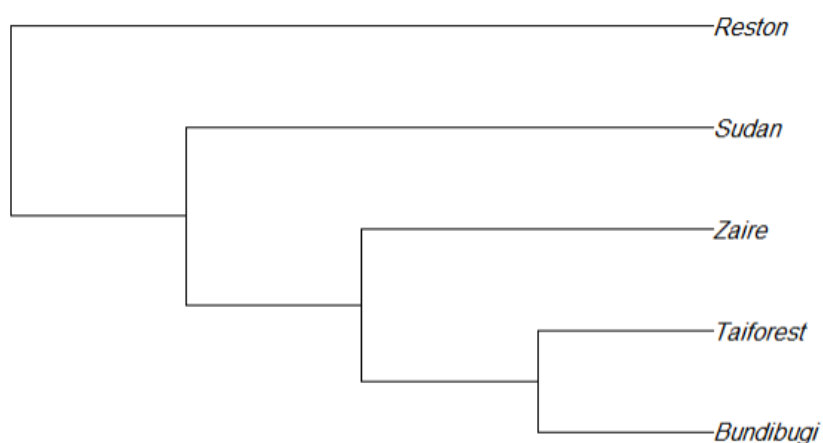
در الگوریتم UPGMA نرخ evolution را یکسان در نظر میگیریم در نتیجه branch tip ها یکسان اند ولی در الگوریتم NJ لزوماً نرخ ها یکسان نیستند و در نتیجه طول branch ها با مقدار تغییرات متناسبند. اگر نرخ هر branch به صورت مشخص شده ای غیر یکسان نباشند ترتیب branch ها در هر دو الگوریتم یکسان خواهد بود. اگر evolution از یک ساعت ژنی<sup>[1]</sup> سختگیرانه ای تبعیت نکند و اگر از فاصله های درست و واقعی برای evolution استفاده کنیم UPGMA روش خوبی است اما در کل نسبت به بقیه روش ها غیر قابل اعتماد است. در غیر این صورت توپولوژی ممکن است نادرست باشد حتی در صورتی که از فاصله های واقعی و مدل های واقعی استفاده کنیم. در کل اگر evolution clock like نباشد روش UPGMA روش قابل اعتمادی نخواهد بود. لذا در کل الگوریتم NJ الگوریتم بهتری است و با توجه به مشاهداتی که در بخش

1. [https://fa.m.wikipedia.org/wiki/%D8%B3%D8%A7%D8%B9%D8%AA\\_%D9%85%D9%88%D9%84%DA%A9%D9%88%D9%84%DB%8C](https://fa.m.wikipedia.org/wiki/%D8%B3%D8%A7%D8%B9%D8%AA_%D9%85%D9%88%D9%84%DA%A9%D9%88%D9%84%DB%8C)

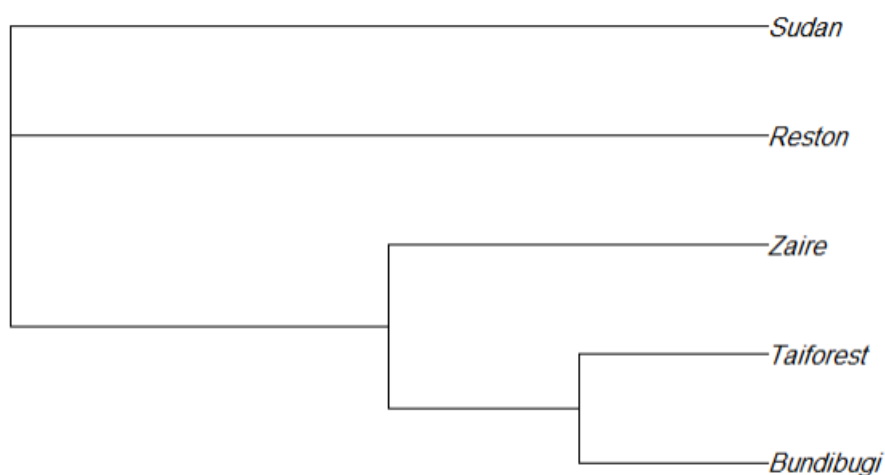
های قبل مربوط به ژن ها دیدیم الگوریتم NJ را به UPGMA ترجیح دادیم.

در بخش بعد ۲-۳ درخت ادغام شده ی ۷ درخت بدست آمده توسط الگوریتم UPGMA و درخت ادغام شده ی ۷ درخت بدست آمده توسط الگوریتم NJ را توسط دستور consensus بدست آوردیم که در تصاویر زیر مشاهده میکنید:

تصویر زیر مربوط به درخت ادغام شده ی UPGMA است:



و تصویر زیر مربوط به درخت ادغام شده ی الگوریتم NJ است:

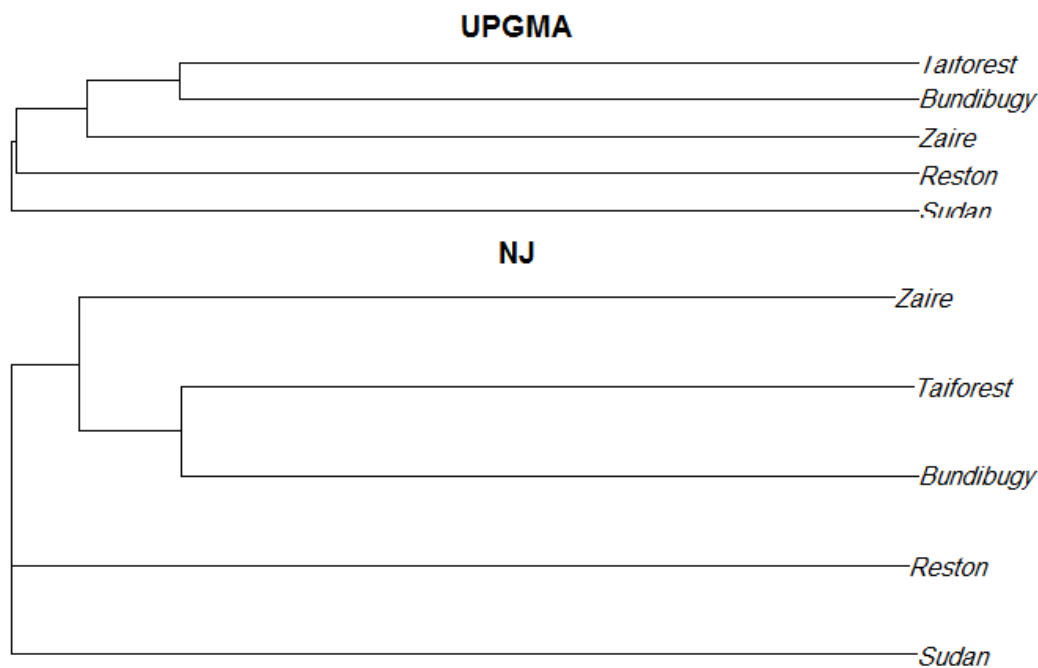


در این قسمت از این روش استفاده کردیم زیرا با این روش قادر هستیم چندین درخت را که از داده های مختلف بدست آورده ایم با هم ادغام کنیم، علاوه بر این با محاسبه ی درخت های جداگانه برای هر ژن میتوانیم درخت های مربوط به هر کدام را داشته باشیم، از طرفی در این روش از درخت های از قبل درست شده به صورت موازی سازی شده استفاده کردیم که خود

[Type here]

آنها از داده های کم حجم مربوط به ژن ها درست شده بودند و در نتیجه با مشکل منابع محاسباتی و مشکل زمان اجرای الگوریتم مواجه نشدیم، این در حالیست که اگر بخواهیم یک درخت را از روی یک supermatrix (!) که از رشته های طولانی ساخته شده اند استفاده کنیم با محدودیت روبرو میشویم.<sup>[1]</sup>

در بخش ۳-۳ با استفاده از کد editDistance.py فاصله ی ویرایش ژنوم های گونه های مختلف را بدست آوردیم و در ماتریس فاصله 5Species.CSV ذخیره کردیم و درخت آن را با روش های UPGMA و NJ رسم کردیم که نتیجه در تصویر زیر قابل مشاهده است:

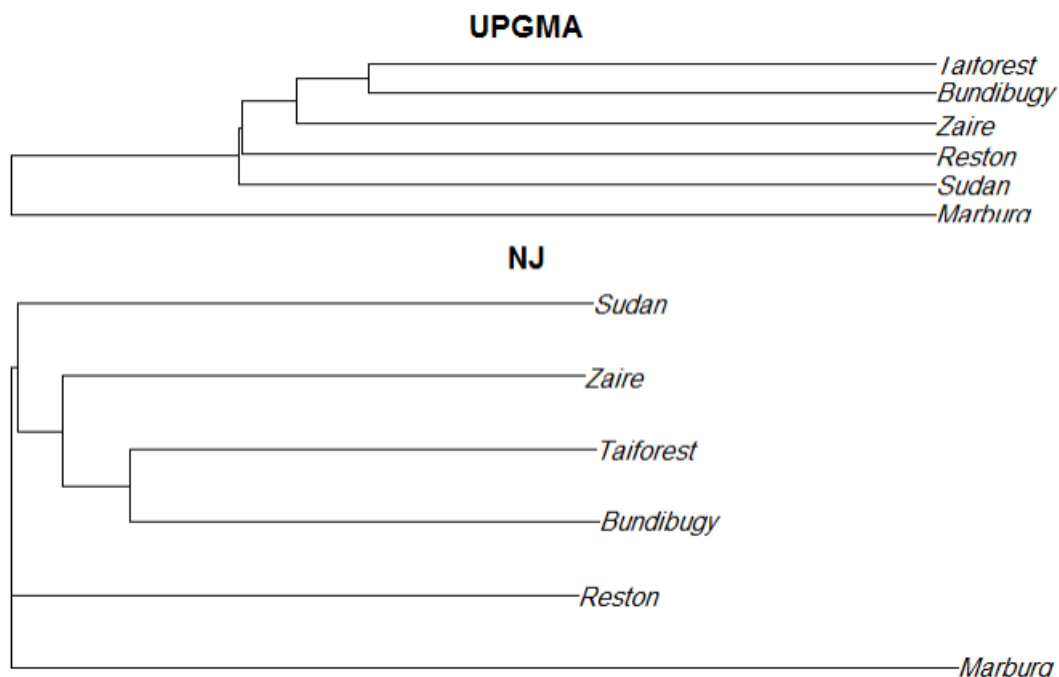


درخت های بدست آمده در این قسمت با درخت های بدست آمده در قسمت قبلی تفاوتی چندانی ندارد و ترتیب برنج ها نیز یکسان است این مساله نشان میدهد درخت بدست آمده از این دو متد (ادغام درخت مربوط به ژن ها و درخت بدست آمده از هم ترازی سراسری ژنوم ها) درخت قابل قبولی است. تفاوتی که در این دو وجود دارد محدودیتی است که هم ترازی سراسری برای ما ایجاد میکند لذا متد ادغام درخت ها متد بهینه تری میباشد.

در بخش ۳-۴ ماتریس فاصله ۶ گونه را در 6Species.CSV میبینیم. با توجه به این ماتریس و با استفاده از روش UPGMA و NJ درخت زندگی این ۶ گونه را بدست می آوریم. با توجه به توضیحات بخش قبل الگوریتم NJ را استفاده میکنیم. تصویر زیر درخت زندگی این ۶ گونه را نشان میدهد.(فاصله ویرایش و کشیدن درخت ها با کمک کد های پیاده سازی شده در بخش قبل صورت گرفته است)

1- <https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/consensus-tree>





در بخش ۱-۴ برای محاسبه ی فاصله ی زمانی گونه های مختلف از روش Jukes-Cantor استفاده کرده ایم زیرا این روش، روشی ساده برای تخمین فاصله ی زمانی بین دو گونه مبتنی بر فاصله ی ویرایش ساختار ژنوم آنهاست. علاوه بر این دلیل در این روش فرض بر این است که احتمال جهش یک نوکلئوتاید به هر نوکلئوتاید متفاوت از آن یکسان است. در نتیجه نرخ جهش در همه ی جهات برابر می باشد. همچنین با توجه به اینکه جهش نوکلئوتایدی یک فرایند مارکف برگشت پذیر است با انجام محاسباتی (که ما براساس گفته شما نتیجه را در اینجا آورده ایم و از انجام محاسبات صرف نظر کرده ایم) میتوان نتیجه گرفت اگر نرخ جهش هر دو گونه یکسان باشد (این نرخ طبق فرض صورت مساله برابر با ۰.۰۰۱۹ است) خواهیم داشت:

$$\lambda t = -\frac{3}{4} \ln(1 - \frac{4}{3}P)$$

که در این رابطه  $t$  فاصله ی زمانی بین دو گونه و  $\lambda$  نرخ جهش و  $P$  ماتریس حاصل از تقسیم فاصله ویرایش دو گونه بر میانگین تعداد کل نوکلئوتاید های رشته هاست. (در این قسمت طول ژنوم گونه ها را به طور میانگین برابر با ۱۹۰۰۰ نوکلئوتاید در نظر گرفته ایم)

نتیجه ی این بخش در ماتریس زیر دیده میشود:

	Bundibugy	Reston	Sudan	Taiforest	Zaire
[1,]	0.0000	255.8448	256.7993	193.8701	227.7396
[2,]	255.8448	0.0000	255.3685	255.1041	250.5837
[3,]	256.7993	255.3685	0.0000	256.2158	252.8374
[4,]	193.8701	255.1041	256.2158	0.0000	225.8698
[5,]	227.7396	250.5837	252.8374	225.8698	0.0000

(واحد های زمانی به سال است.)

[Type here]

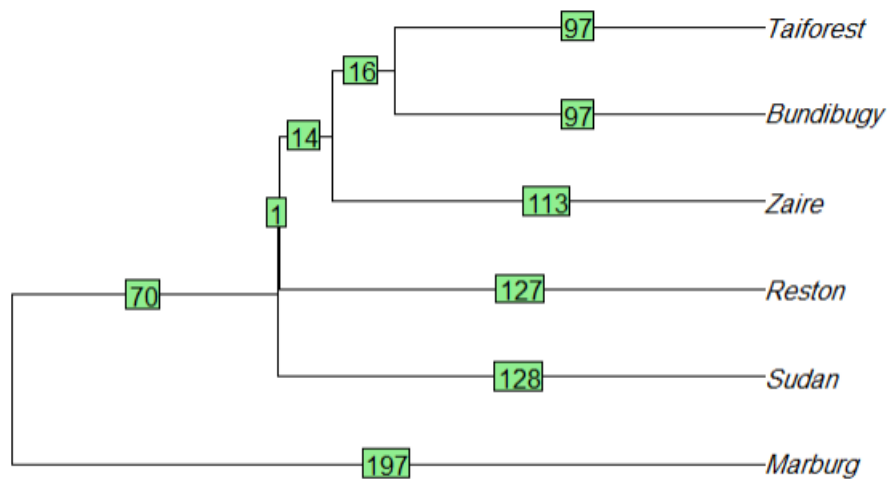
حال می‌خواهیم بدانیم جد مشترک کی می‌زیسته است.

مانند بخش قبلی ماتریس فاصله زمانی را این بار برای ۶ گونه پیدا می‌کنیم. این ماتریس را در تصویر زیر مشاهده می‌کنید:

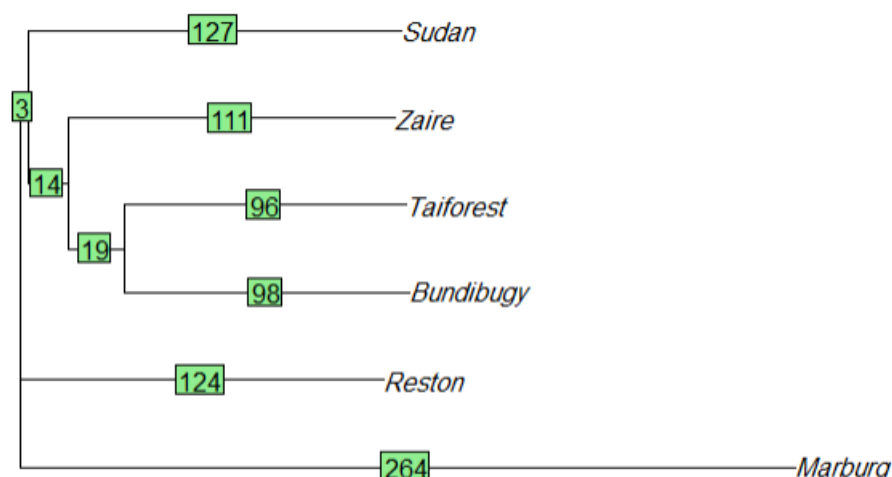
	Bundibugy	Reston	Sudan	Taiforest	Zaire	Marburg
[1,]	0.0000	255.8448	256.7993	193.8701	227.7396	398.9977
[2,]	255.8448	0.0000	255.3685	255.1041	250.5837	388.6311
[3,]	256.7993	255.3685	0.0000	256.2158	252.8374	393.3299
[4,]	193.8701	255.1041	256.2158	0.0000	225.8698	397.6300
[5,]	227.7396	250.5837	252.8374	225.8698	0.0000	393.1049
[6,]	398.9977	388.6311	393.3299	397.6300	393.1049	0.0000

در تصاویر زیر درخت‌های مربوط به بخش ۳-۴ را مشاهده می‌کنید که اندازه‌ی یال‌ها نیز روی آن مشخص است.

تصویر زیر مربوط به الگوریتم UPGMA است



و تصویر دیگر که در ادامه می‌بینید مربوط به الگوریتم NJ است.



واحد روی یال ها به سال است.

الگوریتم ترجیحی ما NJ میباشد. و با توجه به الگوریتم NJ ماربرگ از دیگر گونه های ابولا حدود ۲۶۴ سال پیش جدا شده است. سپس حدود ۱۲۴ سال پیش که جد مشترک همه گونه های ابولا می زیسته است طی یک جهش رستون از بقیه گونه ها جدا میشود. پس از ۳ سال جهشی در جد مشترک ۴ گونه رخ میدهد و گونه ی سودان از جد مشترک ۳ گونه ی دیگر جدا میشود. پس از ۱۴ سال گونه ی زیر از جد مشترک دو گونه ی دیگر جدا میشود و پس از ۱۹ سال دیگر یعنی حدود ۹۷ سال پیش (بین ۹۶ تا ۹۸ سال) گونه های تایفورست و bundibugy هم جدا میشوند. لذا جد مشترک ۵ گونه حدود ۱۲۴ سال پیش می زیسته است.

با توجه به الگوریتم UPGMA ماربرگ از گونه های ابولا حدود ۱۹۷ سال پیش جدا شده است. از همان موقع تا ۷۰ سال پس از آن یعنی حدود ۱۲۷ سال پیش جد مشترک همه گونه های ابولا می زیسته است. که طی یک جهش جد مشترک ۴ گونه به جز سودان از گونه ی سودان جدا میشود. پس از ۱ سال از این جهشی در جد مشترک ۴ گونه رخ میدهد و گونه ی رستون از جد مشترک ۳ گونه ی دیگر جدا میشود. پس از ۱۴ سال گونه ی زیر از جد مشترک دو گونه ی دیگر جدا میشود و پس از ۱۶ سال دیگر یعنی حدود ۹۷ سال پیش گونه های تایفورست و bundibugy هم جدا میشوند. لذا جد مشترک ۵ گونه حدود ۱۲۷ الی ۱۹۷ سال پیش می زیسته است.

۴-۲. یکی از روش های آماری برای پیش بینی جهشی که در آینده رخ خواهد داد روشی است که در آن از الگوریتم bootstrap استفاده میشود. این روش بر اساس پیدا کردن جهش های epistatic در محل های leading و trailing و تخمین فاصله ی بین آنها در این درخت است.