# GenomBilim Yazokulu Pratigi

Mikrosatelit calismalarinin en buyuk problem veri eksikligiyken yeni nesil dizileme (NGS) teknikleri bunun tam tersi bir sorunla karsi karsiya. NGS metotlari o kadar buyuk bir veri uretiyor ki bunu analiz etmek projeler icin onemli bir zorluk teskil ediyor. Arastirmacilar bu buyuklukteki verilerle basa cikabilmek icin oldukca guclu bilgisayarlara ve isletim sistemlerine ihtiyac duyuyorlar. Biyoinformatik bu ihtiyaclari karsilamak amaciyla cikmis bir araclar ve yaklasimlar butunudur. Biyoinformatikci, biyoloji ve bilgisayar bilimi arasinda kopru gorevi gorur. Gunumuzde pek cok universite bu konuda lisans veya yuksek lisans programlari sunsa da biyoinformatik aslen kendi cabamizla da ogrenebilecegimiz bir branstir.

Bu okulda son yillarda biyoinformatigin en ihtiyac duyuldugu alan olan dizilemedeki kullanimlarini ogrenecegiz. Bu amacla herkes kendisine verilen erisim numarasindaki bireyin kisa mitogenom dizilerini insan referans genomuna gore hizalayacak, mutasyonlari bulup her dizinin son halini hazirlayacak. Sonrasinda bu dizileri ve dunyanin pek cok yerinden baska dizileri kullanarak filogenomik haritalarini cikaracagiz.

Bu islemleri yapmak icin hal-i hazirda pek cok biyoinformatik araci zaten internette mevcut. Bu araclarin onemli bir kismi terminal dedigimiz komut satirlarindan calistirilabiliyor. Bu da ancak linux/unix bazli isletim sistemlerinde mumkun. Dolayisiyla okulumuz surecinde linux isletim sisteminde calisacagiz.

# Kisa Dizi Kutuphanesinden mtDNA’ni indir

## Calisma klasorunu olustur:

Once terminali ac:

**ssh egitim20@193.140.99.61 -p 1212**

**şifre: nN5Bb41JRLQ**

**Levrek1'e yönlendirilmek için;**

**ssh egitimXX@levrek1**

**Şifre: sana verilen şifre**

ve kendi hesabın icinde “Genomik” adi altinda bir klasor olustur. Proje boyunca bu klasoru kullanacaksin.

**mkdir Genomik**

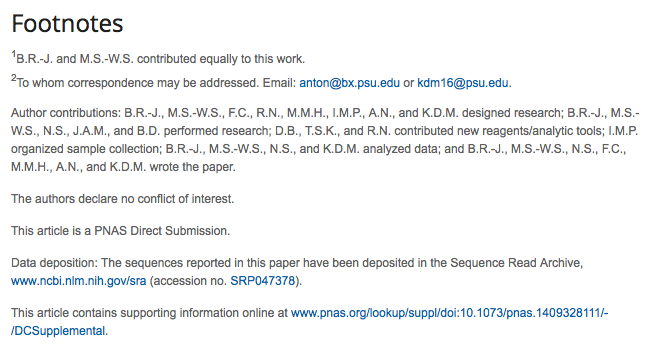
Bir de asagidaki kodu halistir ki egitim20’deki kütüphaneye koyduğumuz tum programları isimleri ile calistirabil:

**source /truba/home/egitim20/.bashrc**

## Kisa dizilerini kullanacagin makaleyi bul:

“Maternal age effect and severe germ-line bottleneck in the inheritance of human mitochondrial DNA” adli makaleyi asagidaki linkten indir :

<http://www.pnas.org/content/111/43/15474.full>



Figur 1. Makalenin Footnotes kismi

Makalede kac mitogenom dizisi uretilmis?

Nasil bir teknoloji kullanarak uretilmisler?

Vucudun hangi bolgesinden alinan ornekler kullanilmis?

Footnotes kisminda (Figur 1), erisim numarasi linkine (accession no.) tikla. Bu link seni arastirmanin uretip yayinladigi kisa dizilerin listesine goturecek (Figur 2).

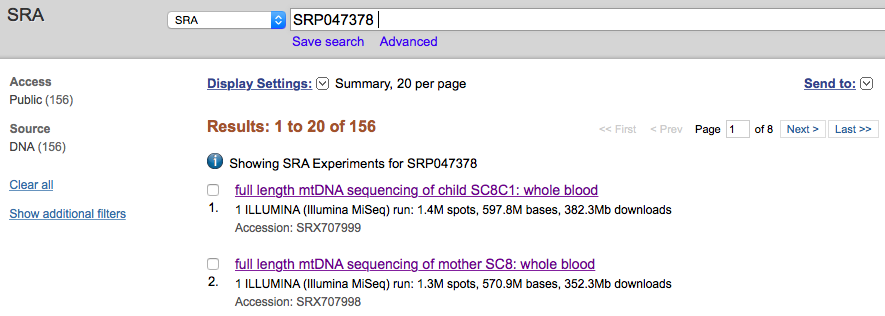
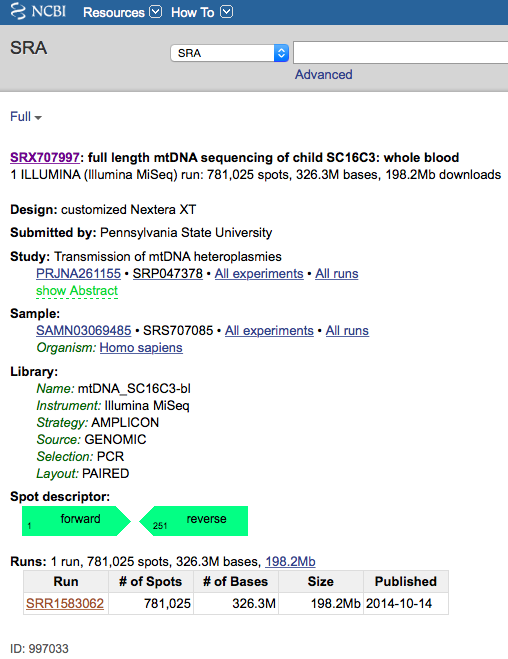


Figure 2. SRA sayfasi

SRA yani kisa dizi arsivi ozellikle yeni dizileme teknolojileri kullanilarak uretilmis genelde 1000 bazdan kisa dizilerin depolandigi bir veri tabanidir. SRA arastirmacilara kisa dizilerini kullanima acik olarak depolayabilecekleri bir yer gorevi gormekle kalmayip ayni zamanda aranilan diziye ve diziyle ilgili detayli deneysel bilgiye hizli bir erisime olanak verir.

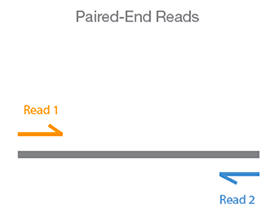
## Calisacagin kisa mitogenom dizilerini indir:

SRA icinde sana kendi eğitim hesap numarana denk dusen diziye tıkla. ’Run’ basliginin altindaki linkteki ‘SRR’ ile baslayan ismi kopyala.



Figur 3. Kisa dizileri indirmek

Kisa dizi dosyasini indirmek icin, SRA toolkit’i kullanacagiz. Dosyayi indirmek icin asagidaki komutu kullanacagiz, SRRin ardindan gelen uc noktayi indirmek istedigin dizinin numarasiyla degistirmeyi unutma:



Figur 4. paired-end okuma

**fastq-dump -I --split-files SRR…**

Bu komut iki tane dosya indirecek: one dogru diziler icin SRR…\_1.fastq ve arkaya dogru diziler icin SRR…\_2.fastq. Iki tane dosya olmasinin sebebi dizilerin paired-end teknolojisiyle uretilmis olmasi. Paired-end, Figur 4’de de gordugun gibi bir fragmanin iki ucundan dizilenmesini saglar, bu sayede cok daha yuksek kalitede bir dizi elde edilir. Biz kolaylik acisindan bu derste sadece one dogru uzanan dizilerle calisacagiz. O yuzden SRR…\_1.fastq dosyasinin adini infile\_1.fastq olarak degistirip SRR…\_2.fastq’yu dosyandan silebilirsin.

**cp SRR…\_1.fastq infile\_1.fastq**

Eger vaktin varsa fastq-dump komutunun nasil isledigini anlamaya calisabilirsin: <http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=fastq-dump>

# Filtreleme ve Kesme

## Kisa dizilerini kesfe cik:

Terminalde asagidaki komutu yazalim:

**head –n 8 infile\_1.fastq**

Bu komut infile\_1.fastq dosyasini acmadan sadece ilk 8 satirina goz atmamizi saglayacak. Dosyamiz cok buyuk oldugu icin **head** (ve son satirlari gosteren **tail**) cok ise yarayan bir komut. fastq dosyasinda bir dizi dort satirdan olusur ve asagidaki gibi gozukur:

@SRR1583062.1.1 1 length=183

AAATAATAGGATGAGGCAGGAATCAAAGACAGATACTGCGACATAGGGTGCTCCGGCTCCAGCGTCTCGCAATGCTATCGCGTGCACACCCCCCAGACGAAAATACCAAATGCATGAAGAGCTCCCGTGAGTGGTTAATAGGGTGATAGACCTGTGATCCATCGTGATGTCTTATTTAAGGGG

+SRR1583062.1.1 1 length=183

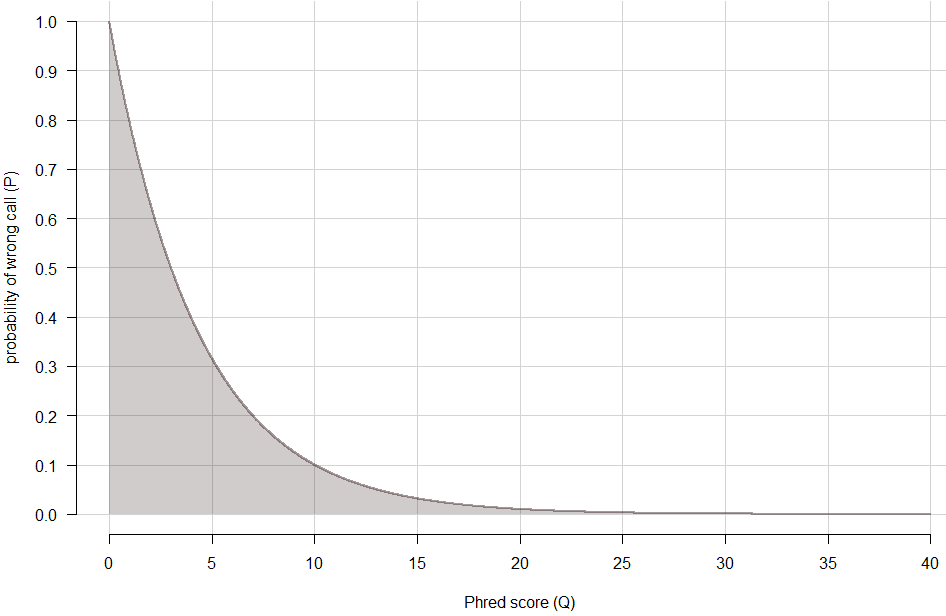
??A??BBBEEDDDDDDGGGGGGIIIIIIIIIIIIIIIIIHHHHHIIIIFHHIIIHHHHHIIIIHHHHIHIHHHIIIIIIHHHHEHHHHHHHHGGGGGGGGGHGGGGGGGGGGGGGGHGGGGGGGGGGCGEGEEGGGGGGGGGGCEGGGGGGGGGGGGGGGGGGGGGEGGEGGGGGGGGGGGGG

Ilk ve ucuncu satir dizinin adini ve uzunlugunu, ikinci satir dizinin kendisini gosterir. Son satirsa Phred kalite skorlarini gosterir. Bu satirdaki her karakter bir bazin kalite skoruna karsilik gelir.

Bir bazin Phred kalite skoru, , o bazin hatali okunma ihtimali *P*’nin logaritmasidir (Figur 5).

or

Mesela, Phred kalitesi 20 olan bir bazin yanlis okunmus olma ihtimali 0.01’dir (. Yani yuz okumadan biri bu bazi yanlis teshis edecektir. Bilimsel calismalar genelde kalite skoru en az 20 hatta mumkunse 30 olan bazlari dikkate aliyorlar.



Figur 5. Phred Kalite Verileri

## Dizinin kalite dagilimini kontrol et

Dizimizdeki okumaların kalite skorlarının dagilimlarina bakmak icin fastx\_toolkit’i kullanacagiz.

**fastx\_quality\_stats -i infile\_1.fastq**

Vaktin oldugunda buraya girip kullandigimiz komutlara ve parametereline göz at:

<http://hannonlab.cshl.edu/fastx_toolkit/commandline.html>

Bu dizileri analiz etmeden once dizilerimizi dusuk kaliteli okumalardan ve bazlardan temizlemeliyiz. Bunu iki asamada yapacagiz. Ilk once kalitenin genel olarak dusuk oldugu bas ve sonlardaki bazlari kesecegiz. Sonrasinda elimizeki dizileri filtreleyerek belli bir kalite skorunun altinda bazi cok olanlari atacagiz.

* Dusuk kaliteli bazlar genom analizi acisindan nasil bir risk olusturabilir? Bir dizinin kalitesini arttirmak icin ne yapmak gerekir?

## Kotu kalite bazlari kes:

Bas ve sonlardaki kotu kalite bazlari kesmek icin fastx fastx\_trimmer diye bir program kullanacagiz. Bunun icin, asagidaki komutu gir:

**fastx\_trimmer -f 20 -l 240 -i infile\_1.fastq -o infile.fastq**

Bu komutun bitmesini beklerken asagidaki linkten ne yaptigina bakabilirsin: <http://hannonlab.cshl.edu/fastx_toolkit/commandline.html>

Linki kullanarak kullandigimiz her bir parametrenin ne ise yaradigini not al:

* + **-f**
  + **-l**
  + **-i**
  + **-o**
* Dizininde infile.fastq isimli yeni bir dosya olusacak. Icerigine bir goz at, komut yapmasini bekledigin islemi yapmis mi diye bir kontrol et. Nasıl yapabiliriz bunu? Asagi kodu not et.
* Ne degismis, asagi yaz:
* Daha az veya çok kesseydik, dosyada ne değişecekti? Satir sayısı değişecek miydi?

## Kotu kalite dizileri filtrele:

Dizilerin genel olarak kalitesi iyilesti. Fakat hala aralarda kotu kalite bazlarimiz mevcut. Bunun icin, simdi “fastq\_quality\_filter” isimli programi kullanarak kotu kalite bazi fazla olan dizileri filtreleyecegiz.

**fastq\_quality\_filter -q 30 -p 95 -i infile.fastq -o outfile.fastq**

Filtreleme gerceklesirken bu komutun tam olarak ne yaptigina yandaki linkten bir goz at: <http://hannonlab.cshl.edu/fastx_toolkit/commandline.html>, ve asagiaki parametrelerin ne ise yaradigini not al:

* + **-q**
  + **-p**
  + **-i**
  + **-o**

Filtreleme bitince, kalite degerlerine tekrar bak, ne değişti?

**fastx\_quality\_stats -i outfile.fastq**

* Neden once kesip sonra filtreliyoruz? Tersini yapsaydik bir sey degisir miydi?
* Daha az veya çok filtreleseydik, dosyada ne değişecekti? Satir sayısı değişecek miydi?

# Insan mitogenomunu indir

Yerlestirme islemlerine baslamadan once referans dizimizi yani insan mitogenomunu indirmemiz lazim:

Indirmek icin, once USCS’ye gidip <https://genome.ucsc.edu/index.html> oradan Downloads/Genome Data/Human/Sequence data by Chromosome akisini takip ederek kromozom dizi listelerini bulalim. Oradan chrM.fa.gz dosyasına sag tıklayıp linkini kopyala. Bunu asagidaki kodun arkasına tırnak icinde gireceğiz:

**wget**

Son olarak dosyayi *ref.fasta* ismiyle kaydet.

**cp chrM.fa ref.fasta**

# Dizileri referans genomuna yerlestir

Dizileri yerlestirmek eslestirmece oyunu gibi dusunulebilir. Amac kisa dizilerin aynilarini (veya benzerlerini) buyuk dizide yani genomda bulmaktir. Bu islemi kolaylastirmak icin biyoinformatik araclari indexleme denilen bir teknik kullanirlar. Bu islem sayesinde cok hizli bir sekilde kisa diziler genom uzerinde yerlerini bulurlar. Biz yerlestirme icin bowtie2 programini kullanacagiz.

## Genomu indexle

Hizli yerlestirme icin ilk adim referans genomumuzu indexlemek. Bunun icin terminalden Genomik klasorune git ve asagidaki komutu gir:

**bowtie2-build ref.fasta ref**

## Dizileri yerlestir

Artik kisa dizileri genoma yerlestirmeye haziriz. Bunun icin asagidaki komutu gir:

**bowtie2 -x ref -q outfile.fastq -S outfile.sam**

Bu islem 5-10 dakika icinde bir SAM (**S**equence **A**lignment/**M**ap) dosyasi uretecek. SAM dosyasi insan gozunun okuyabilecegi text formatinda bir dosyadir ve icinde her bir dizi icin bir baslik bir de yerlestirmeyle ilgili detaylarin yer aldigin bir kisim vardir. Dosyanin ilk satirlarini acip icine bir goz atabilirsin. Yandaki link dosya icerigiyle ilgili daha detayli bilgi verir: <https://samtools.github.io/hts-specs/SAMv1.pdf>

Yerlestirme bitince ekranda kisa bir rapor belirecek.

* Kisa dizilerinden yuzde kaci yerlestirilebilmis? Asagi not et:

SAM dosyasini analiz etmek icin SAMTools programini kullanacagiz. Her programin kendi indexleme sekli oldugundan referans genomumuzu asagidaki komutu kullanarak tekrar indexlememiz gerek:

**samtools faidx ref.fasta**

Bu *ref.fasta.fai*. isimli bir dosya uretecek.

Simdi SAM dosyamizi BAM (**B**inary **A**lignment/**M**ap)’e cevirecegeiz. BAM SAM’in binary halidir. Bu islem icin asagidaki komutu kullanalim:

**samtools view -bt ref.fasta.fai outfile.sam > outfile.bam**

Asagidaki komutlarla bam dosyamizi da indexleyecegiz:

**samtools sort outfile.bam -o outfile.sorted.bam**

**samtools index outfile.sorted.bam**

## Genom Kapsami ve Derinligi

Peki elimizdeki diziler genomun yuzde kacini kapsiyor? Ve de kapsadigi yerler hakkinda ne kadar kesin konusabiliriz?

Bu sorularin cevabini vermek icin dizilerimizin kapsam (coverage) ve derinlik, yani bir noktayi kac kere kapsadigi (depth) degerlerini ogrenmemiz gerekir. Asagidaki konutlari kullanarak bu bilgilere erisebiliriz:

**samtools depth outfile.sorted.bam > depth**

**less depth**

* Diyelim ki bas ve sonlardaki kapsama oranlarını merak ediyoruz, nasıl yaparız?
* Ya da en dusuk kapsamalar kaç ve nerede diye bakmak istiyoruz, nasıl yaparız?

# Varyant Tespiti

Varyantlari tespit demek referans genom ve elimizdeki dizi arasindaki mutasyonlari bulacağız demek. Bunun icin yine SAMTools’u kullanacagiz.

## Referans genomu kullanarak mutasyonlari bul:

Kullandigimiz yerlestirme teknikleri kisa dizileri dogru yerlerine her zaman cok yuksek kesinlikte yerlestiremeyebilir. Ornegin, referans genomda uzun bir tekrar dizisinin icinde yer alan kisa bir dizi pekala tekrar dizisinin baska yerlerine de yerlesebilir. Bu durumda kullandigimiz program uygun yerlerden herhangi birini rastgele secmek durumunda kalir. Bu da yerlestirme guvenilirligini azaltir.

Yerlestirme programlari yerlestirme guvenilirliklerini tipki dizileme kaliteleri gibi skorlarlar. Yerlestirme skoru 10 olan bir dizinin yanlis yere yerlestirilmis olma ihtimali yuzde ondur. Dolayisiyla dizileme kalitesine gore degil yerlestirme kalitesine gore de filtreleme yapmamiz gerekir. Bunun icin yerlestirme kalitesi 10’dan dusuk olan dizileri terminalde, asagidaki komutla filtreleyecegiz:

**samtools mpileup outfile.sorted.bam -Q 30 -q 10 -uf ref.fasta > outfile.bcf**

Bu komutla yaklasik 10 dakika icinde outfile.bcf olarak adlandirdigin bir bcf dosyasi uretiyorsun. Bcf dosya formati tum mtDNA’n icin varyant bilgisini tutan binary bir formattir. Bu formatin icerigiyle ilgili daha fazla bilgi icin yandaki linke goz atabilirsin: <http://samtools.sourceforge.net/samtools.shtml>

## Varyant dosyasini okunabilir formata cevir:

Variant dosyasini bcf’ten, insan tarafindan okunabilir bir format vcf (Variant Call Format)’e cevirmen gerekir. Bunun icin BCFtools araclarini kullanacagiz:

**bcftools call -cV indels outfile.bcf > outfile.vcf**

Eger vaktin varsa, vcf dosyani acip yandaki link yardimiyla icerigini anlamaya calisabilirsin: <http://samtools.github.io/hts-specs/VCFv4.2.pdf>

## Varyant dosyasini indexle:

VCF dosyasini analiz etmek icin varyantlarin indexlenmesi gerekebilir. Bunun icin asagidaki konutlari girecegiz:

**bcftools view outfile.vcf -O z -o outfile.vcf.gz**

**bcftools index outfile.vcf.gz**

# Varyantlara Ilk Analiz

## Kac varyant var?:

VCF dosyasinin icinde ne var? Bcftools’un çok yararlı bir istatistik komutu bize bunu soyleyebilir:

**bcftools stats outfile.vcf.gz**

* Kac SNP var dizinde?
* Kac indel var, niye?

**cat outfile.vcf | cut -f 5| grep -v “\."**

## Varyantlar bir genle kesisiyor mu?:

Varyantlari bulduk. Peki bunlar neredeler? Bunun icin mitokondriyal DNA’daki koordinatlara ihtiyacımız var. Bu dosyayı elde etmenin pek çok yolu var. Biz size ENSEMBL Biomart uzerinden yapmayı gosterecegiz. Asagiya nasıl yaptigimizla ilgili notlar alabilirsin.

Neticede genes.bed isimli bir dosya elde edeceğiz. Bu dosya bed formatında. Bed format detaylarına buradan bakabilirsin: <https://genome.ucsc.edu/FAQ/FAQformat.html>

Elimizdeki varyantlarin kesisip kesilmediğini anlamak icin, ayni sekilde varyantlarimizdan da bir bed dosyası üretmemiz lazim. Hadi yapalım:

outfile.bed diye kaydettiğimiz bu dosya ile genes.bed dosyasında kesilen koordinatlar var mi diye bakabiliriz artik. Elbette bunu gözle yapmayacagiz. Her zamanki gibi bir araç kullanacagiz. Bu sefer kullanacagimiz arac oldukça eglenceli Isvicre cakisi gibi pek cok ise yarayan bir arac: bedtools.

Bunun icinden closest fonksiyonunu kullanacağız:

<https://bedtools.readthedocs.io/en/latest/content/tools/closest.html>

**bedtools closest -a outfile.bed -b genes.bed -d**

* Kac SNP bir gene yakin cikti dizinde?
* Kac tanesi genin icinde, nasıl anladin?
* O genler hangi genler?

## Dizilerimiz birbirine ne kadar yakin:

* Sizce bu diziler birbirlerine ne kadar yakin? Neden?
* Bazilari daha yakin olabilir mi? Hangileri?

Dizilerimiz arasındaki ortak SNP’leri bulmak icin vcftools paketinden vcf-compare aracini kullanacağız: <http://vcftools.sourceforge.net/perl_module.html#vcf-compare>

Bunun icin once bu araç ile vcf dosyamızı indexleyip, zipleyeceğiz:

**bgzip outfile.vcf**

**tabix -p vcf outfile.vcf.gz**

Sonrasinda dosyamızı ismimizle kaydedip (tugce.vcf.gz gibi) egitim20’nin icindeki genomik dosyasının icine yollayacagiz. Bunun icin asagidaki kodları kullan:

**cp outfile.vcf.gz <isim>.vcf.gz**

**cp <isim>.vcf.gz ../egitim20/genomik/**

En son olarak, egitim20’nin icinde biz asagidaki komutu kullanarak dizilerinizde bulduğunuz SNPlerin benzerliklerine bakacagiz:

**vcf-compare isim1.vcf.gz isim2.vcf.gz**