‘Deneysel evrim sonucu ortaya cikan mutasyonlari bulmak’

Deneysel Evrim – Bolum II, Uygulama Egitimi, 7 Subat 2019

Evrimsel Genombilim Uygulamalı Eğitim 2019

Hazirlayan: Gonensin Ozan Bozdag

http://egenombilim.wixsite.com/home

Ozet: Vcf dosyasi filtrelemek, istatistik cikarmak, vcf dosyalarini karsilastirma icin hazirlamak (zip, index), dosyalari karsilastirip sadece evrilmis bireydeki SNP ve indel'leri ayiklamak, "annotation" adimi icin kromozom isimlerini dosyada duzeltmek, "annotate" etmek.

Not: Asagida, '$' isareti ile paslayan satirlar bash komutlarini iceriyor
'#' ile baslayan satirlar ilgili komut hakkinda yorumlari/bilgileri iceriyor.

Baslamadan once: Kullanacaginiz programlarin manual'lerini ve uyguladiginiz islemlerin mantigini okuyup anlamaniz bilimsel/etik olarak sart. O nedenle bu asagida bulunan komutlari ve programlari kendi verinize uygulamadan once, ilgili dokumanlari okumanizi oneririm.

Ornegin vcffilter aracin hakkinda bilgi sahibi olma yontemleri:

#bash uzerinden vcffilter'in komutlarina hizla bakmak icin:
$ vcffilter –h

vcffilter ve vcflib hakkinda bilgi:
https://github.com/vcflib/vcflib#vcffilter

Yapilacak islemlerin mantigi: Elimizde iki ayri bireyden gelen sekanslanmis genom datasi var. Bireyler Saccharomyces cerevisiae maya turunun Y55 soyundan geliyor. Bu genom sekanslari .vcf dosyasi olusturana kadar islemlerden gecirildi. Biz bu uygulamada.vcf dosyasi adimindan itibaren kalan birkac islemi yapacagiz.

.vcf dosyasindan itibaren, varyantlara goz atacagiz, filtreleyecegiz (kalite ve derinlik bazli filtreleme). En son evrilmis genomdaki varyantlar ile atasal genomdaki varyantlari karsilastirip, sadece evrilmis olandaki varyantlari/mutasyonlari ("derived variants") ayiklayacagiz. Son olarak "annotate" ederek mutasyonlarin ne anlama geldigini daha anlasilir hale getirecegiz.

1. VCF dosyasini inceleyelim. Hem Unix-bash araclarini kullanarak .vcf dosyasina bakabiliriz (less, head, cut, vb. araclar bu is icin uygun). Ayrica gorsel olarak IGV isimli programi kullanarak da varyantlara bakabilirsiniz (IGV'nin nasil kullanildigini Sibel hocanizin verdigi derste gordunuz) *.

VCF dosyasi hakkinda detayli bilgi:
http://www.internationalgenome.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-40/

https://samtools.github.io/hts-specs/VCFv4.2.pdf

$ less sample1_lanes_merged_sorted_dedub_fixed_HC.vcf

sample1 atasal bireyin genomunda bulunan varyantlari iceriyor. Bu “varyantlar” deneyde kullanilan Y55 soyunun en iyi referans genomuna sahip maya soyu olan S288c soyla karsilastirildiginda bulunan “varyantlar”.

$ less sample5_lanes_merged_sorted_dedub_fixed_HC.vcf
# sample5 dosyasi evrilmis bireye ait varyantlari gosteriyor.

2. Filtreleme adimi:

VCF dosya formati tab ile ayrilmis 8 sutundan olusan bir tur “text” dosyasidir. O nedenle ilgili filtreleme islemlerini kendiniz, cut, awk, vb. bash araclarini kullanarak yazdiginiz komutlarla filtreleyebilirsiniz. Fakat vcf dosyasi uzerinde calisan vcffilter (vcflib’e ait bir arac), vcftools vb. araclari kullanmak daha sofistike filtreleme islemleri yapmanizi saglar. Biz vcflib’in icinde bulunan vcffilter aracini kullaniyoruz.

Filtreleme islemini “mapping quality” ve “genotype quality” gibi kalite degerlerine gore filtrelemek gerekiyor. Daha fazla bilgi icin VCF dosyasinda bulunan “mapping quality” ve “genotype quality” gibi terimlerin ne anlama geldigini VCF dosya tipini anlatan linklerden okuyun. Diger bir filtreleme secenegi “depth based filtering” (derinlige gore filtreleme) islemi yapmak. VCF dosyasinda DP olarak kisaltilan degerlere bakarak ilgili genomik pozisyona denk gelen olasi (!) varyantin derinliginine gore filtreleme yapilabilir. Ornegin ilgili olasi variantin (ornegin SNP) bulundugu bolgede 10 adetten az okuma var ise, 10’dan cok okumaya sahip genomic pozisyonlara denk gelen olasi varyantlari tutup, 10 okumadan az bolgelere denk gelen olasi varyantlari eleyebilirsiniz.

Filtreleme hakkinda bilgiler:
https://www.biostars.org/p/51439/

“Depth ve quality based filtering” islemi komutu:

Oncelikle atasal birey uzerinde kalite temelli filtreleme yapalim:

$ vcffilter -f “QUAL > 30 & DP > 10”
sample1_lanes_merged_sorted_dedub_fixed_HC.vcf >
sample1_lanes_merged_sorted_dedub_fixed_HC_qf.vcf

“>” ile yeni dosyaya filtreden sonra kalan dosyalari kaydediyoruz. Dosya isminin sonuna ‘qf’ (quality filtered) ekledik.
Depth 10’dan kucuk varyantlari eliyoruz.

Ardindan evrilmis bireyden gelen genomdan cikan olasi varyantlar uzerinde filtreleme yapalim:

$ vcffilter -f “QUAL > 30 DP >10 ” sample5_lanes_merged_sorted_dedub_fixed_HC.vcf > sample5_lanes_merged_sorted_dedub_fixed_HC_qf.vcf

Islemler calisti mi diye bash araclarini kullanarak kontrol edelim – filtrelemeden once ve sonraki VCF dosyalarini istediginiz gibi karsilastirin. Ayrica filtreleme islemi gercekten basarili oldu mu diye daha incelikli kontrol ederseniz cok daha tutarli verilerle sonraki adimlara devam edersiniz! Ornegin depth degeri '10'un altinda variant kaldi mi? awk'I kullanabilirsiniz.

Cok onemli not: Filtreleme vb. adimlardan sonra aslinda "dogru variant" olmayan bircok variant vcf dosyasinda kaliyor. Referans genoma siralama hatasi, sekanslarken olusabilecek hatalar vb. bircok adimda aslinda variant olmayip yine de referans genomdan farkli gibi gorunen nukleotid bolgeleri kalacka. O nedenle filtreleme adimi daha kaliteli bir vcf dosyasi elde etmenizi saglasa da, yine de bircok biyolojik olarak dogru olmayan variant vcf dosyanizda bulunacaktir. Ozetle son elde ettiginiz vcf dosyanizdaki varyant'larin dogruluguna hep suphe ile yaklasmalisiniz.

3. Varyant istatistigi: Filtrelemeden once/sonra vcf dosyalarin istatistiklerini karsilastirmaniz islemlerin ve verinin ne durumda olduguna dair size daha anlasilabilir bilgi verecektir. Bcftools'un stats aracini/komutunu kullanacagiz:

# Ilk olarak filtrelemeden onceki dosyaya bakalim:
$ bcftools stats sample1_lanes_merged_sorted_dedub_fixed_HC.vcf > sample1.log

#Filtrelemeden sonraki dosya:
$ bcftools stats sample1_lanes_merged_sorted_dedub_fixed_HC_qf.vcf > sample1_qf.log

Karsilastir:

$ less sample1.log

$ less sample1_qf.log

$ bcftools stats sample5_lanes_merged_sorted_dedub_fixed_HC.vcf > sample5.log

$ bcftools stats sample5_lanes_merged_sorted_dedub_fixed_HC_qf.vcf > sample5_qf.log

Karsilastir:

$ less sample5.log

$ less sample5_qf.log

4. Varyantlarin filtrelendigi VCF dosyalarini ZIP'leyelim: Bu islem sonraki adimda kullanacagimiz 'bcftools isec' isimli program zip'lenmis dosya uzerinde calistigi icin gerekli.

$ bgzip sample1_lanes_merged_sorted_dedub_fixed_HC_qf.vcf

$ bgzip sample5_lanes_merged_sorted_dedub_fixed_HC_qf.vcf

5. INDEX the zipped VCF file:

#zip'lenmis dosyayi indexleyelim ('bcftools isec' zip'lenmis dosyaya ve index bilgisine bakarak karsilastirma yapiyor).

$ tabix sample1_lanes_merged_sorted_dedub_fixed_HC_qf.vcf.gz

$ tabix sample5_lanes_merged_sorted_dedub_fixed_HC_qf.vcf.gz

6. Sadece evrilmis bireydeki mutasyonlari/varyantlari elde etmek: Su ana kadar variant listelerimiz Y55 maya soyu ve S288c referans genom olarak kullandigimiz maya soyu arasindaki genetic farkliliklari gosteriyor (atasal birey vs. S288c & evrilmis birey vs. S288c). Bizim asil ilgilendigimiz varyantlar "de novo" olarak da adlandirilan deneysel evrim sonucunda evrilmis bireyde ortaya cikmis yeni mutasyonlar (atasal vs. evrilmis veya sample1 vs sample5). Bu mutasyonlari ortaya cikarmak icin iki vcf dosyasini (sample1 & sample5) karsilastirip sadece sample5'de (evrilmis) bulunanlari bir sonraki adim icin kullanacagiz.

$ bcftools isec -p isec_output -Oz
sample1_lanes_merged_sorted_dedub_fixed_HC_qf.vcf.gz
sample5_lanes_merged_sorted_dedub_fixed_HC_qf.vcf.gz

$ less README.txt

Hangi dosyayi kullanacagiz, README.txt dosyasina bakarak karar verin.

Sadece sample5 (evrilmis) bireye ozgu olan varyantlari kopyalayip, bir ust klasore geri atalim (bcftools isec, sonuclari 'isec output' adinda yeni olusturdugu klasore kaydetmisti).

$ cp ./0001.vcf.gz ../

bu dosya isimleri (orn. 0001.vcf.gz) anlamsiz geliyorsa, istediginiz gibi yeniden adlandirin.

$ cd ..

#Kopyaladik mi?

$ ls

7. *Kromozom isimlerini* sadece evrilmis bireyde olan varyantlari iceren VCF dosyasinda *duzeltelim*: VCF dosyasinda kromozom isimleri, son adimda “annotation” yapacagimiz (snpEff) programinin database’inde chrI, chrII, chrIII, chrIV, olarak kayitli. Bu nedenle “annotation” islemini dogrudan 0001.vcf dosyasina yaparsak, “annotation” islemi kromozom isimleri bulunamadigindan calismiyor. Once kromozom isimlerini degistirelim (sed isimli bash aracini kullanacagiz):

$ sed 's/ref|NC_001133|/chrI/' 0001.vcf > test.vcf

# “/ref|NC_001133|” etiketleri “chrI” etiketi ile degisti mi, kontrol edelim:
$ grep “chrI” test.vcf

Simdi de kromozom 2 (chrII) icin ayni islemi yapalim (bu islemi 16 kromozom ve mitochondrial DNA –chrM olarak etiketlenmis- icin yapmak zorundayiz):
$ sed 's/ref|NC_001134|/chrII/' test.vcf > test1.vcf
.
.
.
.
.
$ sed 's/ref|NC_001224|/chrM/' test15.vcf > test16.vcf

Bu sed islem adimlarini tek tek yapmak yerine bir shell script’e cevirip tek seferde yapmaniz cok daha akillica olacaktir.

En son mitkondrial DNA etiketini de vcf dosyamizda duzelttikten sonra annotation icin haziriz.

bu dosya isimleri (orn. test16.vcf) anlamsiz geliyorsa, istediginiz gibi yeniden adlandirin (orn. ‘test16.vcf’ yerine ‘evrilmis_de_novo.vcf’). test16.vcf dosyasinda tum 16+1 kromozomun ismide duzeltilmis olmali.

9. Annotate: snpEff isimli programi kullanacagiz. Oncelikle ilgili ture (Saccharomyces cerevisiae) ait database'i indirmemiz gerekiyor.

A) Annotation dosyasi database'de var mi:

$ snpEff. databases | grep "cerevisiae"

B) Download annotation database

Ilgili canli turune ozgu annotation dosyasi database'ini indirin:

$ snpEff download Saccharomyces_cerevisiae

C) Annotate vcf file:

$ snpEff -I vcf -o vcf -canon -dataDir /truba/home/egitim1/Admin/Programs/snpEff/snpEff/data/ Saccharomyces_cerevisiae test16.vcf > test16_annot.vcf

#Yukaridaki komutta "-dataDir /truba/home/egitim1/Admin/Programs/snpEff/snpEff/data/" ile database'in truba'da nerede oldugunu gosteriyor.

Sonucu inceleyin:

$ less test16_annot.vcf

Ayrica .html dosyasi olarak cok kolay incelenebilecek bir "output" veriyor snpEff. Bu .html dosyasini websayfasi uzerinde inceleyin.

#snpEff annotation aracinin mantigini ve elde ettiginiz dosyada ne bilgiler var ogrenmek icin:

http://snpeff.sourceforge.net/SnpEff_manual.html

http://snpeff.sourceforge.net/VCFannotationformat_v1.0.pdf

Ilgili “annotate” edilmis vcf dosyasi cok karisik gelebilir. Istediginiz ozellikleri (orn. Kromozom, pozisyon, etkisi, gen adi vb.) secip daha sade hale getirebilirsiniz:

$ snpEff extractFields -s "," -e "." test16_annot.vcf CHROM POS REF ALT AF DP "ANN[*].EFFECT" "ANN[*].GENEID" "ANN[*].BIOTYPE" > test16_annot_sade.vcf

-snpEff’i bash komutu ile calistirmak ile ilgili onemli not:

Can (Elverici) snpEff’i dogrudan calistirabilmemiz icin alias olusturmustu. Eger snpEff’I kendi bilgisayariniza indirip alias olusturmazsaniz, java uzerinden calistirmaniz gerekir. Ornegin az onceki komutun java komutu eklenerek calistirilmasi:

$ **java -jar** /usr/bin/genome_tools/snpEff/**SnpSift.jar** extractFields -s "," -e "." test16_annot.vcf CHROM POS REF ALT AF DP "ANN[*].EFFECT" "ANN[*].GENEID" "ANN[*].BIOTYPE" > test16_annot_sade.vcf

Yukaridaki komutta “/usr/bin/genome_tools/snpEff/” “path”i benim bilgisayarimda snpEff’in yuklu oldugu klasoru gosteriyor. Sizin bilgisayarinizda neredeyse oranin “path”ini vermeniz gerekiyor.

10. Sadece SNP’lere veya indel’lere bakmak isterseniz: Sadece SNP’leri incelemek isterseniz, ilgili “annotate” edilMEmis dosyadan SNP’leri (veya indel’leri) ayiklayabileceginiz vcftools’un guzel bir secenegi var.

Vcftools kullanarak:

$ vcftools --vcf test16.vcf --remove-indels --recode --recode-INFO-all --out test16_snps_all &

* IGV: https://software.broadinstitute.org/software/igv/download

SON