Biyoinformatik · Uygulama notları

Toplu Transkriptomik

Ham SRA Okumalarından Diferansiyel Gen İfadesi,
Zenginleşme, Ağlar ve Şekillere

Bu not, yeniden üretilebilirliği gözeten, referansa dayalı bir toplu RNA dizileme (bulk RNA-seq) iş akışını açıklar. Veri alımı, kalite kontrolü, seçilen kırpma aracı, isteğe bağlı rRNA uzaklaştırma ve kontaminasyon taraması, hizalama ve gen düzeyinde nicelendirme ele alınır. Alternatif hizalayıcı ve sayım araçları ile hizalamasız bir yol da sunulur. Diferansiyel gen ifadesi için DESeq2, edgeR ve limma-voom karşılaştırmaları ile eşdeğerlik testi açıklanır. Model ve model olmayan organizmalarda işlevsel zenginleşme (GO, KEGG, GSEA, g:Profiler ve özel gen kümeleri), protein ilişki ağları ve yaygın sonuç şekilleri kapsanır. Sayım matrisi, diferansiyel ifade tablosu veya GEO mikrodizi serisiyle üç alternatif başlangıç, aynı sonraki analizlere bağlanır. Her aşama kararları, girdileri, çıktıları ve sık hata kaynaklarını açıklar. Yardımcı kod okunabilir betikler ve ortam tanımları sağlar; doğrulama durumu ile çalıştırma sınırları açıkça belirtilir.

Kod ve yazılım ortamları:
github.com/tunabirgun/bioinformatics-notes

Tuna Birgun
github.com/tunabirgun

Orijinal Not: 2 Temmuz 2026 · Türkçe: 29 Ekim 2026

İçindekiler

1 Genel bakış ve kapsam **3**
- 1.1 Araç seçiminden önce verilmesi gereken kararlar . . . 3

2 Bu çalışma notunun kullanımı **4**

3 İş akışına genel bakış **5**

4 Yazılımlar ve paketler **5**

5 Ortam kurulumu **7**
- 5.1 İşletim sistemi: Linux, WSL2 veya macOS . . . 7
- 5.2 Miniconda kurulumu ve kanal yapılandırması . . . 8
- 5.3 Ayrı bir ortam oluşturma ve araçları kurma . . . 8
- 5.4 Proje dizin yapısının oluşturulması . . . 9
- 5.5 Yardımcı betiklerin çalıştırma sözleşmesi . . . 9

6 Ham okumalardan sayım matrisi oluşturmaya **10**
- 6.1 Terminalden veri kümesi bulma ve alma . . . 10
- 6.2 Ham veriyi indirme (SRA Toolkit) . . . 11
- 6.3 Başlangıç kalite kontrolü (FastQC ve MultiQC) . . . 12
- 6.4 Kontaminasyon taraması (FastQ Screen) İsteğe bağlı . . . 12
- 6.5 Adaptör ve kalite kırpma (fastp) . . . 13
- 6.6 Kırpma sonrası kalite kontrol . . . 13
- 6.7 Alternatif kırpma araçları: Trim Galore ve Trimmomatic . . . 14
- 6.8 rRNA uzaklaştırma (SortMeRNA) İsteğe bağlı . . . 15
- 6.9 Referanssız rRNA uzaklaştırma (RiboDetector) İsteğe bağlı . . . 16
- 6.10 Okumaları yeniden eşleştirme (BBMap `repair.sh`) İsteğe bağlı . . . 17
- 6.11 Genom indeksini oluşturma (STAR) . . . 17
- 6.12 Hizalama (STAR ile sıralı BAM) . . . 18
- 6.13 Alternatif hizalayıcı: HISAT2 . . . 19
- 6.14 Kütüphanenin iplik yönlülüğü bilgisini belirleme . . . 20
- 6.15 Gen düzeyinde nicelendirme (featureCounts) . . . 20
- 6.16 Kalite kontrolü özeti: Eşleme ve atama oranları . . . 21
- 6.17 Hizalama sonrası RNA QC (RSeQC) İsteğe bağlı . . . 22
- 6.18 Alternatif sayım aracı: htseq-count . . . 23
- 6.19 Alternatif sayım: STAR gen sayımları (`--quantMode GeneCounts`) . . . 24
- 6.20 Hizalamasız nicelendirme (Salmon) . . . 25
- 6.21 Tek uçlu okumalar . . . 25

7 Alternatif başlangıç noktaları **26**
- 7.1 Bir sayım matrisi ile başlama . . . 26
- 7.2 Bir diferansiyel gen ifadesi sonuç tablosuyla başlama . . . 26
- 7.3 limma ile GEO mikrodizi serisi . . . 27

8 DESeq2 ile diferansiyel gen ifadesi **28**
- 8.1 Sayımları içe alma ve örnek üst veri tablosu oluşturma . . . 28
- 8.2 Veri kümesini kurma, ön filtreleme ve referans düzeyini seçme . . . 29
- 8.3 DESeq2 çalıştırma ve sonuçları alma . . . 30
- 8.4 Eşdeğerlik testi: Anlamlı bir sınır içindeki etkiler İsteğe bağlı . . . 31
- 8.5 Sayımları dönüştürme ve dışa aktarma . . . 32
- 8.6 Gen anotasyonu (AnnotationDbi ve biomaRt) . . . 32
- 8.7 İlgilenilen genlerin incelenmesi . . . 33
- 8.8 DE yöntemlerini karşılaştırma: edgeR ve limma-voom . . . 34

9 İşlevsel zenginleşme **36**
- 9.1 Gen kimliklerini eşleme ve arka plan gen kümesi tanımlama . . . 36
- 9.2 GO ve KEGG aşırı temsil analizi (ORA) . . . 37

9.3 Sıralı gen listesinde GSEA . . . 38
9.4 OrgDb olmadan zenginleşme (model olmayan organizmalar) . . . 38
9.5 Kendi gen kümesi kaynağınızı kullanma . . . 40
9.6 Zenginleşmeyi görselleştirme . . . 41
9.7 Diğer gen kümesi kaynakları ve uzunluk yanlılığı . . . 41
9.8 Örnek başına gen kümesi etkinliği (GSVA) İsteğe bağlı . . . 42

10 Protein etkileşim ağı **42**

11 Transkriptomik için şekiller **44**
11.1 PCA (örnek yapısı) . . . 44
11.2 UMAP (isteğe bağlı; çok örnekli yapı) . . . 45
11.3 Örnek uzaklığı ısı haritası ve dendrogram . . . 45
11.4 MA grafiği: Etki büyüklüğü ve ifade . . . 46
11.5 Volkan grafiği: Anlamlılık ve etki büyüklüğü . . . 47
11.6 İfadesi farklılaşan öne çıkan genlerin ısı haritası . . . 49

12 Ek bilgiler **50**
12.1 Notun ötesi: *De novo* birleştirme ve varyant çağırma . . . 50
12.2 Genişletilmiş QC ve yardımcı araçlar . . . 51
12.3 Döngüler ve toplu işleme . . . 51
12.4 İş parçacığı, bellek ve kritik parametreler: Hızlı başvuru . . . 52
12.5 Yeniden üretilebilirlik . . . 52
12.6 Sorun giderme . . . 53
12.7 Örnek korelasyonları ve duyarlılık kontrolleri . . . 53
12.8 Çalışmalar arası sentez . . . 54
12.9 Anotasyon aktarımı ve elde edilemeyen sonuçlar . . . 54

Kaynaklar **55**

1 Genel bakış ve kapsam

Bulk RNA-seq, her biyolojik örnekte hücrelerden bir araya getirilen RNA'yı ölçer. Örneğin bağımsız tedavi ve kontrol hayvanlarını karşılaştırmak, gruplar arasında göreli gen bolluğunun değişip değişmediğini sorgular; doku düzeyindeki fark hücre bileşiminden, hücre içi ifadeden veya her ikisinden kaynaklanabilir. Okumaları toplamadan ve yazılım seçmeden önce bağımsız biyolojik birimleri tanımlayın.

Bu çalışma notu, temel bir iş akışını ve açıkça ayrılmış alternatiflerini izler. Kabuk aşamaları (Bölüm 5–6), ham dizileme verilerini bir *sayım matrisi* hâline getirir; R aşamaları (Bölüm 8–11) bu matristen diferansiyel gen ifadesi sonuç tabloları, zenginleşen yolaklar ve şekiller üretir. İsteğe bağlı aşamalar **İsteğe bağlı** işareti taşır. Bir kırpma aracı, bir hizalama/nicelendirme yolu, bir sayım aracı ve birincil diferansiyel gen ifadesi (DE) yöntemi seçin; numaralı betiklerin tümünü sırayla çalıştırmayın.

Uygulama notu

Çalışma örneği. Not boyunca **çift uçlu Illumina** okumaları, iki gruplu bir tasarım (`treated` ve `control`) ve organizma için bir **referans genom (FASTA) ile anotasyon (GTF)** bulunduğu varsayılır. Komutlar Linux üzerinde `bash` kabuğuna göre yazılmıştır. `<angle brackets>` içindeki yer tutucuları kendi değerlerinizle değiştirin.

İş akışının kapsamı. Bu, *referansa dayalı, gen düzeyinde* bir iş akışıdır: Okumalar genoma hizalanır ve gen başına özetlenir. Uygun bir referans bulunduğunda ve amaç gen düzeyinde diferansiyel ifade olduğunda kullanılan temel yoldur.

Kapsam dışında kalanlar. İş akışı transkript/izoform düzeyinde nicelendirme veya *de novo* birleştirme yapmaz. Transkript düzeyinde kestirim ya da genom hizalaması olmadan bir yol gerekiyorsa Salmon veya kallisto kullanın ve sonuçları `tximport` ile alın [22]; aşağıdaki DESeq2, zenginleşme ve şekil bölümleri, Bölüm 7'de açıklanan girdiye özgü uyarlamalarla gen düzeyindeki çıktıya uygulanır. Referans genomu bulunmayan bir organizmada transkriptleri *de novo* birleştirin (Bölüm 12.1).

Yöntemler arasındaki yeri. Aşamaların sırası, yaygın RNA-seq iyi uygulama derlemeleri ve topluluk iş akışlarının genel yaklaşımını izler [24, 25, 26, 27]: Deney tasarımı ve kütüphane seçimi, kalite kontrolü, okuma kırpma, hizalama veya hizalamasız nicelendirme, gen düzeyinde sayım, normalleştirme ve diferansiyel ifade, ardından işlevsel yorumlama. Araçlar yaygın kullanılan seçeneklerdir; kararın önemli olduğu yerlerde alternatifler ayrıca belirtilir (Bölüm 6.20 ve 9.7).

Başlangıç noktaları. Varsayılan yol ham okumalardan başlar; hazır bir sayım matrisi, önceden hesaplanmış bir diferansiyel ifade tablosu veya GEO mikrodizi serisiyle hizalamadan sonraki aşamalara da girilebilir (Bölüm 7).

Uygulama notu

Önce tasarım. İstatistiksel çıkarımın geçerliliği tasarıma bağlıdır. Teknik tekrar yerine biyolojik tekrar kullanın; tekrar sayısını beklenen değişkenlik ve etki büyüklüğü üzerinden planlayın. Grup başına üç örnek burada öğretici bir alt sınırdır, istatistiksel güç garantisi değildir. Belirli bir noktadan sonra diferansiyel ifade gücünü artıran temel unsur dizileme derinliğinden çok biyolojik tekrardır. Gerekli okuma derinliği organizmaya, kütüphane karmaşıklığına, ifade düzeyine ve soruya bağlıdır; evrensel bir hedef yerine çalışmaya özgü güç ve doygunluk kanıtını kullanın. Kütüphaneyi biyolojiyle eşleştirin: Poly-A seçimi olgun mRNA'yı zenginleştirir; rRNA uzaklaştırma uygulanan total-RNA kütüphaneleri lncRNA, pre-mRNA ve histon mRNA'larını da korur ve bozulmuş ya da FFPE girdilere daha elverişli olabilir. [24, 27]

1.1 Araç seçiminden önce verilmesi gereken kararlar

Gen düzeyindeki sayım modeli, hücre bileşimindeki değişimleri hücre içi ifade değişimlerinden ayıramaz. Yazılımı seçmeden önce biyolojik birimi, hedef popülasyonu ve karşılaştırmayı tanımlayın. Örnekleri ekstraksiyon, kütüphane hazırlama ve dizileme partilerine rastgele dağıtın; bilinen değişkenlik kaynakları için bloklama uygulayın ve atama kaydını saklayın. Dizileme hatları ve aynı donörden alınan tekrarlı ölçümler

bağımsız biyolojik tekrar oluşturmaz. Eşleştirilmiş donörlerde eşleştirmeyi modele ekleyin; eşleştirilmemiş test uygulamayın. Bu ilkeler, güncel deney tasarımı perspektif yazısında da vurgulanır [77].

2016 tarihli iyi uygulamalar derlemesi ve 2019 tarihli RNA-seq derlemesi temel kaynak olmayı sürdürür [24, 25]. 2026 tarihli deney planlama derlemesi, RNA sınıfları, örnek kalitesi ve platform seçimini güncel bağlamda ele alır [76]; tek bir hizalayıcının veya DE yönteminin her durumda üstün olduğunu gösteren bir karşılaştırma değildir. Kütüphane hazırlama yöntemini hedef RNA sınıfına ve materyalin kalitesine göre seçin; okuma düzenini, iplik yönlülüğünü, referans derlemeyi ve anotasyon sürümünü kaydedin. Total RNA ve poli(A) seçilimli kütüphaneler farklı RNA topluluklarını örnekler. Yalnızca kütüphane büyüklüğünün normalleştirilmesi bu farkları gidermez.

Hedeflenen niceliği seçin. Gen düzeyinde diferansiyel ifade, tanımlanmış bir karşılaştırmada gen bolluğunun değişip değişmediğini sorgular. Diferansiyel transkript kullanımı, kesilip birleştirme olayları ve transkript keşfi farklı yanıt değişkenleri, anotasyonlar ve yöntemler gerektirir. Gen düzeyindeki iş akışının başarılı olması bu analizleri doğrulamaz. Düzeltilmiş p değeri de etki büyüklüğü veya belirli bir genin yanlış bulgu olma olasılığı değildir. Test edilen sıfır hipotezini, etki yönünü ve çoklu test ailesini sonuçla birlikte belirtin.

DE bölümünde sıfır etkiye karşı olağan test, gözlenen etkiye göre keşif amaçlı süzme ve eşik hipotezi birbirinden ayrılır (Bölüm 8.3); yöntemlere özgü girdi sözleşmeleri ise alternatiflerin yanında açıklanır (Bölüm 8.8).

2 Bu çalışma notunun kullanımı

Kodlar iki tür kutuda gösterilir: **SHELL** komut satırı, **R** ise R kodu içindir. R kodu, nesnelerin incelenip yeniden çalıştırılmasını kolaylaştıran ardışık adımlardan oluşur. Açıklık için gövdede *tek* örneğe ait komutlar verilir; Bölüm 12.3, bunları döngüye almayı ve iş parçacığı, referans ile yol değerlerini parametreleştirmeyi gösterir. Aşamalara ayrılmış okunabilir yardımcı betikler ve ortam tanımları `github.com/tunabirgun/bioinformatics-notes` deposundadır.

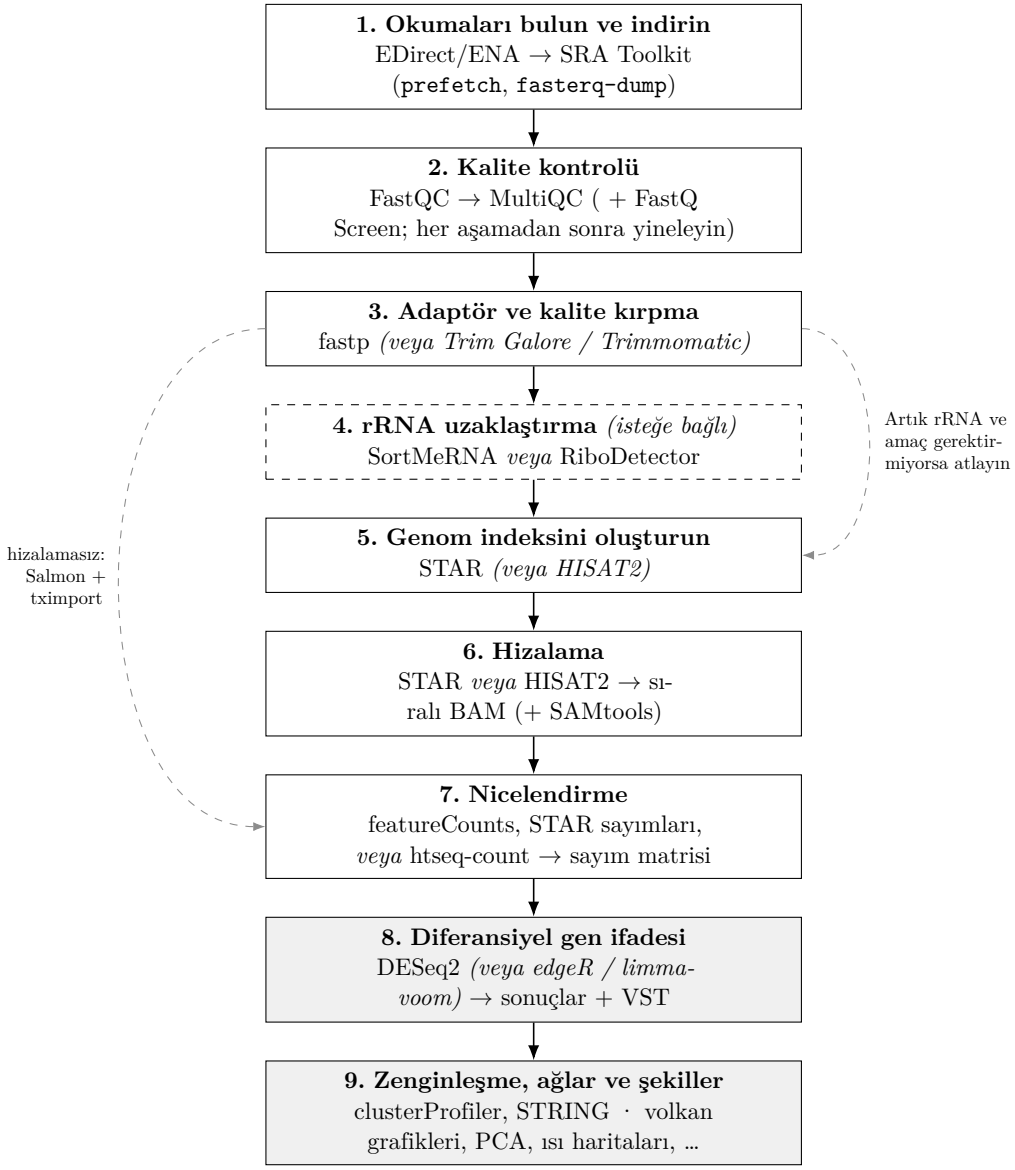
Dikkat edilmesi gerekenler

Doğrulama durumu. Gözden geçirilen betiklerin sözdizimi denetlendi; girdi sözleşmeleri küçük sentetik örneklerle sınandı. Sabit tohumlu sentetik sayım matrisi yolu, DESeq2/apeglm ve tanımlanabilir teknik parti eş değişkeniyle çalıştırıldı; yapılandırılmış tasarım, özgün çıkarım sütunları, eksiklik ve şekil girdisi üyeliği denetimlerini geçti. Bu sınırlı bir yürütme kontrolüdür; biyolojik doğrulama değildir. Bu baskı için analiz yazılım yığınının tamamı, gerçek veriyle yeniden üretim ve şekil üretimi çalıştırılmadı. Sürüm kısıtları sağlanır; çözülmüş bir conda kilidi veya sınanmış bir `renv.lock` dosyası pakete dâhil değildir. İsteğe bağlı komşu iş akışlarını, belirtilen ön koşulları karşılanana kadar açıklayıcı örnek olarak değerlendirin.

Uygulama notu

Komut satırına yeni başlıyorsanız. SHELL kutusundaki kod terminale yazılır: macOS Terminal, Windows üzerinde Ubuntu/WSL2 kabuğu (Bölüm 5.1) veya bir Linux terminali. **R** kutusu R ya da RStudio içinde çalıştırılır. Terminalde bir *istem* görünür; bu çoğunlukla `$` veya `(rnaseq) $` biçimindedir. Bir komutu yapıştırıp Enter'a basın ve sonraki komuta geçmeden bitmesini bekleyin. Satır sonundaki `\`, tek komutun birkaç satırda sürdüğünü gösterir; bloğu birlikte yapıştırın. `#` ile başlayan satırlar açıklamadır, çalıştırılmaz. `<angle brackets>` yer tutucularını değiştirin; araçların PATH üzerinde bulunması için önce `conda activate rnaseq` komutunu çalıştırın.

3 İş akışına genel bakış



4 Yazılımlar ve paketler

Komut satırı araçları **conda** (Miniconda) aracılığıyla **Bioconda** ve **conda-forge** kanallarından kurulur [23]; R paketleri CRAN ve Bioconductor kaynaklıdır. Aşağıdaki sürümler notun hazırlanmasında esas alınan sürümlerdir; ortamda gerçekten kurulan sürümleri kaydedin (Bölüm 12.5). Tablo, hedeflenen araç kümesini gösterir; birden fazla platformda doğrulanmış bir ortam çözümü değildir. Tam künyeler Kaynaklar bölümündedir.

Komut satırı araçları

Araç	Sürüm	Amaç
SRA Toolkit (prefetch, fasterq-dump)	—	NCBI SRA ham dizileme kayıtlarını indirir ve FASTQ biçimine dönüştürür. [1]
Entrez Direct (esearch, efetch)	—	Kabuktan NCBI SRA/GEO sorgular; erişim numarası ve üst veri kayıtlarını çözümler. [28]
FastQC	0.12.1	Örnek başına okuma kalitesi ölçütleri sağlar: Baz kalitesi, adaptör içeriği ve duplikasyon. [2]
MultiQC	1.30	Çeşitli araç ve örneklerin kalite kontrolü (QC) günlüklerini bir HTML raporunda toplar. [3]
fastp	1.0.1	Adaptör kırpma, kaliteye göre filtreleme ve okuma kalite kontrolünü tek geçişte yapar. [4]
Trim Galore İsteğe bağlı	—	Cutadapt sarmalayıcısıdır; otomatik adaptör saptama ile adaptör ve kalite kırpma uygular. [56]
Trimmomatic İsteğe bağlı	0.39	Belirtilen adaptör FASTA dosyasına göre adaptör keser; kayan pencere kalite kırpması yapar. [57]
FastQ Screen İsteğe bağlı	—	Okuma alt kümesini bir genom paneline karşı tarayarak kontaminasyon ve tür karışımını araştırır. [58]
SortMeRNA İsteğe bağlı	4.3.4	Küratörlü rRNA veri tabanlarına göre ribozomal RNA okumalarını filtreler. [5]
RiboDetector İsteğe bağlı	—	CPU üzerinde çalışan sinir ağı sınıflandırıcısıyla referanssız rRNA uzaklaştırır. [59]
BBMap (repair.sh) İsteğe bağlı	38.96	rRNA filtrelemesinden sonra çift uçlu okumaları yeniden eşleştirir. [6]
STAR	2.7.11b	İntron çıkarılmasını gözeten hizalayıcı; RNA-seq okumalarını referans genoma eşler. [7]
Salmon İsteğe bağlı	—	Genom hizalaması olmadan transkript nicelendirmesi yapar; hızlı ve düşük bellek gereksinimlidir. [40]
SAMtools	1.22.1	SAM/BAM hizalama dosyalarını sıralar, indeksler ve özetler. [8]
RSeQC (infer_experiment) İsteğe bağlı	—	RNA-seq kalite kontrolü: İplik yönlülüğü, okuma dağılımı ve gen gövdesi kapsamı. [34]
UCSC (gtfToGenePred, genePredToBed) İsteğe bağlı	—	GTF dosyasını RSeQC için gereken 12 sütunlu BED gen modeline dönüştürür. [68]
Cytoscape İsteğe bağlı	—	Dışa aktarılan protein etkileşim ağını GraphML biçiminde görüntüleyen masaüstü uygulamasıdır; cytoscape.org kaynağından edinilir. [67]
Subread (featureCounts)	2.1.1	Hizalanmış okumaları genlere atar ve sayım matrisi oluşturur. [9]
Miniconda + Bioconda/conda-forge	—	Araçların kurulmasında kullanılan paket yöneticisi ve kanallardır. [23]

R / Bioconductor paketleri

Paket	Amaç
`DESeq2`	Ham sayımlardan negatif binom genelleştirilmiş doğrusal model (GLM) ile diferansiyel gen ifadesi analizi yapar. [10]
`apeglm`	Log_2 kat değişimi kestirimlerine shrinkage (sıfıra doğru büzülme) uygular; `lfcShrink` kullanır. [11]
`edgeR`	Yarı olabilirlik testli negatif binom GLM modelleri sağlar; alternatif DE yöntemidir. [42]
`limma`	Doğrusal modeller kurar ve ampirik Bayes yaklaşımıyla düzenlenmiş t istatistikleri üretir; RNA-seq için limma-voom ve mikrodizi yolunu yürütür. [60]
`clusterProfiler`	GO, KEGG ve özel gen kümeleri için aşırı temsil ve GSEA zenginleşme analizleri sağlar. [12]
`enrichplot`	Zenginleşme sonuçlarını görselleştirir: dotplot, cnetplot, emapplot ve ridgeplot. [13]
`fgsea`	Önceden sıralanmış liste üzerinde hızlı gen kümesi zenginleşme analizi (GSEA) yapar. [45]
`msigdbr`	MSigDB Hallmark, küratörlü ve ontoloji gen kümelerini düzenli bir tablo olarak verir. [44]
`gprofiler2`	g:Profiler hizmeti üzerinden OrgDb gerektirmeyen GO/KEGG/Reactome zenginleşmesi sağlar. [63]
`GSVA`	Bir ifade matrisi üzerinden örnek başına gen kümesi etkinlik skoru üretir. [48]
`STRINGdb`	STRING sorgular ve genleri protein-protein ilişkilerine eşler. [64]
`igraph`	Graf oluşturma, topluluk saptama ve merkezîlik ölçütleri sağlar. [65]
`ggraph`	Ağ çizimi için grafik dil bilgisi tabanlı yerleşimler sağlar. [66]
`org.Hs.eg.db`	AnnotationDbi aracılığıyla genom çapında anotasyon ve gen kimliği eşleme sağlar. [14]
`biomaRt`	Çevrim içi Ensembl anotasyonu ve kimlik eşleme sağlar: Biyotip, koordinatlar ve uzunluklar. [33]
`GEOquery`	GEO serilerinin ifade matrislerini ve örnek üst verilerini R ortamına indirir. [30]
`affy` İsteğe bağlı	Affymetrix CEL dosyalarını okur ve RMA normalleştirmesi uygular. [61]
`tximport`	Salmon/kallisto transkript kestirimlerini DESeq2 için gen düzeyine aktarır. [22]
`ggplot2`	Özel şekillerin temelini oluşturan grafik dil bilgisi sistemidir. [15]
`ggrepel`	Saçılım grafiklerindeki metin etiketlerinin, örneğin gen adlarının, çakışmasını azaltır. [16]
`ggtext`	Grafik metninde Markdown/HTML biçimlendirmesini, italikleri ve alt simgeleri işler. [17]
`EnhancedVolcano`	Tek çağrıyla biçimlendirilmiş volkan grafiği üretir. [18]
`pheatmap`	Satır ölçekleme, anotasyon ve kümeleme içeren ısı haritası üretir. [19]
`ComplexHeatmap`	Birden fazla anotasyon şeridiyle ayrıntılı biçimde özelleştirilebilen ısı haritaları sağlar. [20]
`uwot`	Örnek yapısı için doğrusal olmayan boyut indirgeme yöntemi UMAP sağlar. [21]

5 Ortam kurulumu

5.1 İşletim sistemi: Linux, WSL2 veya macOS

Komut satırı aşamaları Unix kabuğu varsayar; yalnızca Linux için ayrılmış bir bilgisayar gerektirmez. Aşağıdaki hedef platformlarda ortamın ayrı ayrı çözülmesi gerekir; tek bir paketin bulunması, tüm yazılım yığınının kurulabildiğini göstermez.

- **Linux** (yerel kurulum veya laboratuvar sunucusu/HPC): Referans ortamdır. Gerekli paket sürümlerinin makine mimarisinde bulunduğunu denetleyin.
- **WSL2 üzerinden Windows**: Bir dağıtım kurun (`wsl --install -d Ubuntu`); projeyi `/mnt/c/` altında değil, Linux ev dizininde (`~/`) tutun. `/mnt/c` üzerinden dosya sistemleri arası giriş/çıkış daha yavaştır. Bioconda araçları böylece yerel Linux ortamındaki gibi çalışır. WSL2 varsayılan olarak ana

makine RAM'inin 50%'sini ayırır; memeli STAR indeksi daha fazla bellek gerektiriyorsa `.wslconfig` içindeki `memory =` ayarını yükseltin.

- **macOS** (Intel veya Apple Silicon): R/Bioconductor ve komut satırı paketlerinin tam mimari ve sürüm için bulunabilirliğini denetleyin. Gerekli bir araç yalnızca x86-64 için varsa ayrı, uyumlu bir ortam veya desteklenen Linux makinesi kullanın; mimarileri aynı ortamda karıştırmayın.

Komut satırı

```
# Apple Silicon: most tools install natively (see the main env in the next subsection).
# For a tool with no arm64 build (e.g. SortMeRNA), make a small emulated x86-64 env for it:
CONDA_SUBDIR=osx-64 conda create -n rrna_x86 -y sortmerna
conda activate rrna_x86
conda config --env --set subdir osx-64        # keep this env on osx-64 for later installs
```

Dikkat edilmesi gerekenler

Sınırlayıcı etken çoğunlukla işletim sisteminden çok bellektir: STAR'ın insan genomu indeksi yaklaşık ∼30 GB RAM gerektirir. Birincil genom derlemesinin FASTA dosyası ve `--genomeSAsparseD 2`, indeks boyutunu azaltabilir; ancak hizalama ve koordinata göre sıralamanın 24 GB içine sığacağını garanti etmez. Sıralama ayrıca bellek ayırır. Derlemeyi bilimsel soruya göre seçin, en yüksek bellek kullanımını ölçün ve işletim sistemine pay bırakın. [72] Gerekirse STAR'ı ortak bir sunucuda çalıştırın veya birkaç GB bellek kullanan hizalamasız nicelendirme yolunu seçin (Bölüm 6.20). Emüle osx-64 yolu uyumlu Rosetta kurulumu da gerektirir; gerçek macOS sürümündeki desteği doğrulayın.

5.2 Miniconda kurulumu ve kanal yapılandırması

Conda yalıtılmış ortamlar sağlar; açık sürüm kısıtları ve çözülmüş ortam dışa aktarımı, seçilen sürümleri kaydeder. Kanalları *bir kez* yapılandırın; kanal sırası ve `strict` önceliği hangi derlemelerin seçileceğini belirler.

Komut satırı

```
# 1. Download and install Miniconda (accept the licence, allow conda init)
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda.sh
bash ~/miniconda.sh
source ~/.bashrc
conda --version                        # verify it works

# 2. Add channels in the recommended order, then enforce strict priority
conda config --add channels bioconda
conda config --add channels conda-forge
conda config --set channel_priority strict
```

Dikkat edilmesi gerekenler

Önce `bioconda`, *ardından* `conda-forge` ekleyin; en son eklenen kanal en yüksek önceliği alır. `channel_priority strict`, çalışma sırasında sorun yaratabilecek uyumsuz kanal karışımlarının seçilmesini engeller. Bioconda belgelerinde önerilen öncelik sırası budur. Yardımcı kurulum betiği, genel ayarları değiştirmeden YAML dosyasındaki kanal sırasını kullanır. [70]

5.3 Ayrı bir ortam oluşturma ve araçları kurma

Önce temel yolu kurun; yalnızca kullanacağınız alternatifleri ekleyin. Yardımcı dizinde temel YAML dosyası ile isteğe bağlı araçların ayrı bir listesi bulunur. Bunların hiçbiri çözülmüş bir kilit dosyası değildir.

Komut satırı

```
conda create -n rnaseq -y \
  sra-tools entrez-direct fastqc fastp star samtools subread
conda activate rnaseq

# MultiQC is happiest installed with pip inside the same environment
pip install multiqc

# RiboDetector pulls PyTorch (~2 GB); install it only if you use the reference-free rRNA
    ↪ route
conda install -n rnaseq -y ribodetector
```

Uygulama notu

`multiqc`, sıkı conda Python sürüm kısıtlarıyla çakışabilir; etkin ortam içinde pip ile kurmak bu tür çakışmaları azaltabilir. Kurulumdan sonra her aracı uygun `<tool> -version` komutuyla doğrulayın.

5.4 Proje dizin yapısının oluşturulması

Tutarlı bir dizin yapısı girdileri, ara dosyaları ve sonuçları ayırır; döngülerin davranışını öngörülebilir kılar.

Komut satırı

```
mkdir -p rnaseq_project/{raw_data,trimmed,rrna_clean,ref,mapped,counts,results,config} \
         rnaseq_project/qc/{pre_trim,post_trim,post_rrna,fastq_screen,rseqc} \
         rnaseq_project/results/ppi
cd rnaseq_project
```

Proje dizinleri

```
rnaseq_project/
|-- raw_data/            # SRA downloads + FASTQ (.fastq.gz)
|-- qc/
|   |-- pre_trim/        # FastQC before trimming
|   |-- post_trim/       # FastQC after trimming
|   |-- post_rrna/       # OPTIONAL: FastQC after rRNA removal
|   |-- fastq_screen/    # OPTIONAL: contamination-screen reports
|   `-- rseqc/           # OPTIONAL: read distribution + gene-body coverage
|-- trimmed/             # trimmer output (clean reads)
|-- rrna_clean/          # OPTIONAL: SortMeRNA or RiboDetector output
|-- ref/                 # genome FASTA + GTF + STAR index (+ genes.bed)
|-- mapped/              # STAR sorted BAM files (+ .bai)
|-- counts/              # featureCounts matrix + summary
`-- results/             # DESeq2 tables, enrichment, figures, ppi/
```

5.5 Yardımcı betiklerin çalıştırma sözleşmesi

Kod deposunu analiz dizininden ayrı tutun. `BULK_CODE` değişkenini yardımcı kod klasörüne ayarlayın, dizin kurulumunu bir kez çalıştırın; ardından kopyalanan `config/samples.tsv` ve `config/analysis.R` dosyalarını düzenleyin. Her örnek kimliği bir biyolojik analiz birimini göstermelidir; teknik dizileme hattı verilerini çıkarımdan önce birleştirin veya birleştirme biçimini belgeleyin. Üst veri dosyası koşulu ve tasarımdaki eş değişken alanlarını sağlar. Sayımlar üst veriye sütun konumuyla değil, örnek kimliğiyle eşleştirilir.

Komut satırı

```
export BULK_CODE=/path/to/main_scripts/bulk_transcriptomics
export BULK_PROJECT="$HOME/rnaseq_project"
bash "$BULK_CODE/scripts/00_setup/01_project_dirs.sh"
cd "$BULK_PROJECT"
# Edit config/samples.tsv and config/analysis.R before analysis.
# Set after inspecting protocol and strand evidence:
export BULK_STRANDED=2
Rscript "$BULK_CODE/scripts/05_differential_expression/01_deseq2.R"
```

Son komut, mevcut bir featureCounts tablosundan başlar; önceki kabuk aşamalarını çalıştırmaz. Başka bir biçimden başlıyorsanız ilgili başlangıç betiğini seçin. Sağlanan R yardımcısı, kilit dosyası bulunduğunda yardımcı kod dizinindeki renv projesini yükler; böylece çalışma dizini verileri içerirken kodun paket kitaplığı kullanılabilir. R aşamasından önce bu kitaplığı kurup doğrulayın. Ortam tanımını veya üretilmiş bir kilidi başarılı yeniden üretimin kanıtı saymayın.

6 Ham okumalardan sayım matrisi oluşturmaya

6.1 Terminalden veri kümesi bulma ve alma

Açık RNA-seq verileri birbirine bağlı üç arşivde bulunur: NCBI GEO (seri `GSE`, örnek `GSM`), NCBI SRA (çalışma `SRP`, dizileme kaydı `SRR`) ve SRA içeriğini yansıtıp hazır FASTQ sunan EMBL-EBI ENA. Üçü de kabuktan sorgulanabildiği için veri alımı betiklerle yürütülebilir ve yeniden üretilebilir.

NCBI EDirect ile üst veri sorgulama. EDirect, NCBI E-utilities araçlarını komut satırından kullanır: `esearch` sorguyu çalıştırır, `efetch`/`esummary` kayıtları alır, `xtract` XML içeriğini sütunlara dönüştürür [28].

Komut satırı

```
conda install -c bioconda entrez-direct          # installs esearch, efetch, xtract, ...

# resolve a GEO series to its SRA run accessions + a metadata table (RunInfo CSV)
esearch -db sra -query "GSE183947" | efetch -format runinfo > runinfo.csv
python "$BULK_CODE/scripts/01_data_retrieval/03_parse_runinfo.py" runinfo.csv SRR_Acc_List.
    ↪ txt

# check layout/strategy/platform before committing disk and compute
esearch -db sra -query "PRJNA670204" | efetch -format runinfo > runinfo.csv
python "$BULK_CODE/scripts/01_data_retrieval/03_parse_runinfo.py" runinfo.csv SRR_Acc_List.
    ↪ txt
#   -> Run, LibraryStrategy, LibraryLayout (PAIRED/SINGLE), Platform
```

Dizileme kayıtlarını indirme. Depolanmış dizileme verileri iki yoldan alınabilir; dosyaların bayt düzeyinde aynı olduğunu varsaymak yerine kökenini ve okuma düzenini doğrulayın. SRA Toolkit, `.sra` kapsayıcısını indirip açar (sonraki alt bölüm); ENA ise doğrudan gzip ile sıkıştırılmış FASTQ sağlar. İkinci yol dönüştürme aşamasını atlar ve çoğu zaman daha hızlıdır.

Komut satırı

```
# ENA Portal API: turn an accession into its FASTQ URLs (and md5 checksums)
curl -s "https://www.ebi.ac.uk/ena/portal/api/filereport?accession=SRR16356854&result=
    ↪ read_run&fields=run_accession,fastq_ftp,fastq_md5,library_layout&format=tsv"

# fetch every FASTQ the report lists (loop this over SRR_Acc_List.txt)
curl -s "https://www.ebi.ac.uk/ena/portal/api/filereport?accession=SRR16356854&result=
    ↪ read_run&fields=fastq_ftp&format=tsv" \
  | tail -n +2 | tr ';' '\n' | sed 's#^#https://#' | wget -i - -P raw_data/
```

Bazı yardımcı araçlar erişim numarasını çözümleyip tek komutla indirme yapar:

- `pysradb`, erişim numarası türlerini dönüştürür ve dosyaları alır. Örnek komutlar: `pysradb gse-to-srr GSE183947` veya `pysradb metadata SRP324458`. [31]
- `ffq`, JSON üst verisi ve doğrudan bağlantılar döndürür: `ffq GSE183947`, `ffq --ftp SRR16356854`. [32]
- `fastq-dl -a PRJNA670204`, bir projedeki tüm FASTQ dosyalarını ENA veya SRA üzerinden alır.
- `nf-core/fetchngs`, erişim numarası listesinden FASTQ dosyaları ve düzenli bir örnek tablosu üretir. [26]
- R içinden `GEOquery`, GEO serisinin üst verisini ve yazarların depoladığı işlenmiş dosyaları, örneğin bir sayım matrisini okur. [30]

R

```r
library(GEOquery)
gse   <- getGEO("GSE183947", GSEMatrix = TRUE)     # series metadata (+ matrix if present)
pheno <- pData(gse[[1]])                            # sample-level phenotype table
getGEOSuppFiles("GSE183947")                        # author files, e.g. a counts matrix
```

Dikkat edilmesi gerekenler

GEO `GSE` kaydı işlenmiş veriyi ve üst veriyi içerir; ham okumalar bağlantılı SRA `SRP`/`SRR` kayıtlarında bulunduğundan çoğu iş akışı `SRR` listesiyle başlar. İndirmeden önce `LibraryLayout` (`PAIRED` veya `SINGLE`) ve `LibraryStrategy` (`RNA-Seq`) alanlarını doğrulayın. Her ENA dosyasını yayımlanan `fastq_md5` değeriyle `md5sum` kullanarak karşılaştırın.

6.2 Ham veriyi indirme (SRA Toolkit)

`prefetch`, sıkıştırılmış `.sra` kapsayıcısını alır; `fasterq-dump` bunu FASTQ biçimine açar. Çift uçlu kayıtlar için `<SRR>_1.fastq` ve `<SRR>_2.fastq` dosyalarını yazar.

Komut satırı

```
# one accession (repeat / loop for more -- see Supplementary)
prefetch SRR1234567 --max-size 50g -O raw_data/
fasterq-dump raw_data/SRR1234567 -e 8 -O raw_data/ --split-files
gzip raw_data/SRR1234567_*.fastq          # compress to .fastq.gz
```

Temel parametreler

`--max-size 50g`, büyük kayıtların atlanmaması için `prefetch` varsayılan boyut sınırını 20 GB'den yükseltir. `-e 8` (`= --threads`), `fasterq-dump` işlemini hızlandırır. `--split-files`, çift uçlu veride R1/R2'yi ayrı dosyalara yazar. `-O` çıktı dizinini belirler.

Girdiler ve çıktılar

Girdi: Bir SRA erişim numarası; örneğin `SRR1234567`. **Çıktı:** `raw_data/SRR1234567_1.fastq.gz` ve `_2.fastq.gz`.

Dikkat edilmesi gerekenler

`fasterq-dump`, `.sra` boyutunun yaklaşık ∼10× katına kadar geçici disk alanı gerektirebilir; `-t <tmp>` için yeterli boş alanı bulunan hızlı bir disk seçin. Kaydın gerçekten çift uçlu olduğunu doğrulayın: Tek çıktı dosyası tek uçlu veriye işaret eder; iş akışının geri kalanı varsayılan olarak bunu kullanmaz.

6.3 Başlangıç kalite kontrolü (FastQC ve MultiQC)

Başlangıç kalite kontrolü, kırpma öncesinde okuma kalitesini ve olası sorunları görmeyi sağlar. Her FASTQ dosyasında FastQC çalıştırın; raporları MultiQC ile birleştirerek örnekleri birlikte karşılaştırın.

Komut satırı

```
fastqc raw_data/*.fastq.gz -o qc/pre_trim/ -t 8
multiqc qc/pre_trim/ -o qc/pre_trim/ -n multiqc_pre_trim
```

Temel parametreler

FastQC için `-o`, önceden var olması gereken çıktı dizinidir; `-t` dosya başına bir iş parçacığı çalıştırır, dosya sayısını aşan iş parçacıkları boşta kalır. MultiQC için `-o` çıktı dizinini, `-n` rapor adını belirler; üzerine yazmak için `-f` ekleyin.

Uygulama notu

Önce *Per-base sequence quality (baz başına dizi kalitesi)*, *Adapter content (adaptör içeriği)* ve *Over-represented sequences (fazla temsil edilen diziler)* modüllerini inceleyin; kırpmanın hangi sorunu gidermesi gerektiğini gösterirler. MultiQC; fastp, STAR ve featureCounts günlüklerini de ayrıştırır. İş akışının tamamını özetlemek için sonunda yeniden çalıştırın.

6.4 Kontaminasyon taraması (FastQ Screen) İsteğe bağlı

FastQ Screen, küçük bir okuma alt kümesini referans genom paneline hizalar ve her referansa hizalanan oranı bildirir. FastQC'den farklı olarak kütüphanenin beklenen türe ait olup olmadığını araştırır. Fare/insan çapraz kontaminasyonunu, adaptör veya vektör kalıntılarını ve yüksek rRNA oranını gösterebilir. Tanısal bir araçtır; filtreleme yapmadığı için okumaları uzaklaştırmaz.

Komut satırı

```
# one-time: download the bundled genome panel (human, mouse, rRNA, adapters, vectors, ...)
fastq_screen --get_genomes --outdir ref/fastq_screen_db

# screen one sample's R1 with an initial 100k-read diagnostic subset
fastq_screen --aligner bowtie2 \
  --conf ref/fastq_screen_db/FastQ_Screen_Genomes/fastq_screen.conf \
  --subset 100000 --threads 8 \
  --outdir qc/fastq_screen/ \
  raw_data/SRR1234567_1.fastq.gz
```

Temel parametreler

`--get_genomes`, referans panelini indirir ve eşleşen `fastq_screen.conf` dosyasını bir kez yazar. `--conf`, genom listesi ve bowtie2 indekslerini içeren bu yapılandırmayı gösterir. `--aligner bowtie2`, kullanılacak hizalayıcıdır; ortamda bowtie2 bulunmalıdır. `--subset 100000`, 100 bin okumayı örnekler; `--threads` iş parçacığı sayısıdır, `--force` eski raporun üzerine yazar. Yalnızca R1'i taramak ekonomik bir ilk kontroldür; asimetrik kontaminasyon veya teknik artefakt şüphesi varsa iki ucu karşılaştırın.

Girdiler ve çıktılar

Girdi: Ham FASTQ. **Çıktı:** `qc/fastq_screen/*_screen.txt` ve HTML sütun grafiği. MultiQC için yerleşik bir modül bulunur.

Dikkat edilmesi gerekenler

Genom paneli birkaç GB boyutundadır; bir kez indirip aynı yolu yeniden kullanın. Kontaminasyonu kırpma işlemi gizlemeden görebilmek için taramayı *ham* okumalar üzerinde yapın. rRNA satırındaki yüksek oran, rRNA uzaklaştırmayı değerlendirmek için bir işarettir (Bölüm 6.8); beklenmeyen bir türün yüksek oranı, kırpmanın düzeltemeyeceği örnek karışması veya kontaminasyonu düşündürür.

6.5 Adaptör ve kalite kırpma (fastp)

fastp, çift uçlu veride adaptörleri saptar, baz kalitesi içeriğine göre okumaları filtreler, çok kısa okumaları eler ve tek geçişte HTML/JSON raporu üretir.

Komut satırı

```
fastp \
  -i raw_data/SRR1234567_1.fastq.gz -I raw_data/SRR1234567_2.fastq.gz \
  -o trimmed/SRR1234567_1.trim.fq.gz -O trimmed/SRR1234567_2.trim.fq.gz \
  --detect_adapter_for_pe \
  -q 15 -u 40 -l 36 \
  -h qc/post_trim/SRR1234567.html -j qc/post_trim/SRR1234567.json \
  --thread 8
```

Temel parametreler

`-i`/`-I`, girdi R1/R2; `-o`/`-O`, çıktı R1/R2 dosyalarıdır. `--detect_adapter_for_pe` çift uçlu adaptör saptamayı açar. `-q 15` en düşük baz kalitesi, `-u 40` izin verilen düşük kaliteli baz yüzdesi üst sınırı, `-l 36` ise kırpma sonrası 36 bp'den kısa okumaları eleme sınırıdır. `-h`/`-j` raporları, `--thread` iş parçacığı sayısını belirtir; v1.0.1 için üst sınır 64'tür. İki renkli kimyada (NextSeq/NovaSeq), poly-G kuyruklarını kırpmak için `-g` ekleyin.

Gösterilen değerler öğretici başlangıç ayarlarıdır. Yalnızca bu kalite kuralı düşünüldüğünde, 100 bazlık bir okumada Q15 altındaki 41 baz `-u 40` sınırını aşar; 40 baz aşmaz. `-q` değerini yükseltmek veya `-u` değerini düşürmek daha çok okumayı eler; bu seçenekler düşük kaliteli bazları tek tek kırpmaz. `-l` değerini yükseltmek, kırpma sonrası kısa kalan okumaları daha çok dışlar; belirsiz kısa okumalarla birlikte hizalanabilir parçalar da kaybedilebilir. Değişiklikten sonra eşli okuma korunmasını ve hizalama/atama sonuçlarını inceleyin; daha katı ayar her zaman daha çok bilgi sağlamaz. [4]

Girdiler ve çıktılar

Girdi: Ham çift uçlu FASTQ. **Çıktı:** `trimmed/*_1.trim.fq.gz`, `*_2.trim.fq.gz` ve örnek başına HTML/JSON.

Dikkat edilmesi gerekenler

Kırpılmış çıktıyı girdi dosyalarının üzerine yazmayın. Bu komut adaptörleri kırpar ve kaliteye göre okuma filtreler; kayan pencere uç kırpması ayrı seçenekler gerektirir. Aşırı kırpmadan kaçının; fastp raporunda önceki/sonraki kaliteyi ve uzunluk filtresini geçen okuma oranını inceleyin.

6.6 Kırpma sonrası kalite kontrol

Adaptörlerin giderildiğini ve kalitenin iyileştiğini doğrulamak için kırpılmış okumalarda FastQC ve MultiQC işlemlerini yineleyin.

Komut satırı

```
fastqc trimmed/*.fq.gz -o qc/post_trim/ -t 8
multiqc qc/post_trim/ -o qc/post_trim/ -n multiqc_post_trim
```

6.7 Alternatif kırpma araçları: Trim Galore ve Trimmomatic

Varsayılan araç fastp'dir; Trim Galore ve Trimmomatic de yaygın kullanılır ve laboratuvar standardı ya da değerlendirme süreci bunlardan birini gerektirebilir. Üç araç da adaptörleri ve düşük kaliteli bazları uzaklaştırıp çok kısa okumaları eler; ancak ayarları ve okuma tutma kuralları sayımları değiştirebilir. Seçtiğiniz aracın çıktısını standart `trimmed/<sample>_{1,2}.trim.fq.gz` biçimine yeniden adlandırın; böylece hizalama döngüsü değişmez (Bölüm 12.3).

Trim Galore, Cutadapt ve FastQC'yi birleştiren bir sarmalayıcıdır; adaptörü otomatik saptar ve ayrı adaptör FASTA dosyası gerektirmez.

Komut satırı

```
trim_galore --paired --gzip --cores 4 \
  --quality 15 --length 36 \
  --basename SRR1234567 --output_dir trimmed/ \
  raw_data/SRR1234567_1.fastq.gz raw_data/SRR1234567_2.fastq.gz

# Trim Galore names its output *_val_1.fq.gz / *_val_2.fq.gz; rename to the canonical form
mv trimmed/SRR1234567_val_1.fq.gz trimmed/SRR1234567_1.trim.fq.gz
mv trimmed/SRR1234567_val_2.fq.gz trimmed/SRR1234567_2.trim.fq.gz
```

Temel parametreler

`--paired`, çift uçlu kırpmayı açar; uçlardan biri `--length` sınırının altına düşerse çift elenir. `--quality 15`, Phred eşiğidir; `--length 36`, korunan en düşük uzunluktur. `--cores`, yaklaşık ∼4 çekirdeğe kadar etkilidir. `--basename` çıktı ön ekini belirler.

Trimmomatic, belirtilen adaptör FASTA dosyasına göre kesim ve kayan pencere kırpması uygular. Argümanların *sırası*, işlemlerin sırasını belirler. Her uç için eşli ve eşi bulunmayan olmak üzere dört çıktı dosyası yazar.

Komut satırı

```
# Set BULK_ADAPTER_FASTA to the actual bundled or study-selected adapter FASTA first.
trimmomatic PE -threads 8 \
  raw_data/SRR1234567_1.fastq.gz raw_data/SRR1234567_2.fastq.gz \
  trimmed/SRR1234567_1.trim.fq.gz trimmed/SRR1234567_1.unpaired.fq.gz \
  trimmed/SRR1234567_2.trim.fq.gz trimmed/SRR1234567_2.unpaired.fq.gz \
  "ILLUMINACLIP:$BULK_ADAPTER_FASTA:2:30:10" LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN
    ↪ :36
```

Temel parametreler

Çıktı sırası: *R1 eşli, R1 eşi bulunmayan, R2 eşli, R2 eşi bulunmayan*; sonraki aşamalara yalnızca iki *eşli* dosyayı verin. `ILLUMINACLIP:<fa>:2:30:10`, sırasıyla adaptör dosyası, başlangıç eşleşmezliği (seed mismatch), palindrom kırpma ve basit kırpma eşikleridir. `LEADING`/`TRAILING` 3, uçlarda kalite 3 altındaki bazları kırpar. `SLIDINGWINDOW:4:15`, dört bazlık pencerenin ortalama kalitesi 15 altına düşünce keser; `MINLEN:36`, 36'dan kısa okumaları eler.

Girdiler ve çıktılar

Girdi: Ham çift uçlu FASTQ. **Çıktı:** `trimmed/*_1.trim.fq.gz` ve `*_2.trim.fq.gz`; yeniden adlandırmadan sonraki standart adlar veya eşli çıktılar.

Dikkat edilmesi gerekenler

Trim Galore'un `*_val_1.fq.gz` adları değiştirilmezse hizalama döngüsü dosyaları bulamaz. Trimmomatic için adaptör FASTA dosyasının gerçek yolunu belirtin; conda ortamıyla gelen dosya kullanılabilir. Standart çift uçlu sayım yolunda eşi bulunmayan okumalar kullanılmaz. İşlemler yazıldıkları sırada uygulandığından `ILLUMINACLIP` ilk sırada olmalıdır. fastp varsayılanlarına yakın, otomatik adaptör saptama için **Trim Galore**; adaptör kümesi ve kalite penceresini açıkça denetlemek için **Trimmomatic** seçin.

6.8 rRNA uzaklaştırma (SortMeRNA) İsteğe bağlı

Laboratuvarda poli(A) seçimi veya rRNA uzaklaştırma uygulanmış olması, dizilenen kütüphanede artık rRNA kalmadığını göstermez. Kalite kontrolünde görülen artık rRNA, hedeflenen rRNA dışı analizi etkiliyorsa hesaplamalı uzaklaştırmayı değerlendirin; rRNA araştırma sorusunun parçasıysa koruyun. Yeni bir geçişe karar vermeden önce daha önce hesaplamalı filtre uygulanıp uygulanmadığını ve korunma raporunu kaydedin. Yalnızca protokolün adı filtrelemek veya atlamak için yeterli değildir. rRNA olarak sınıflandırılan çiftleri çıkarmak, kalan parça havuzunu değiştirir; hizalamadan önce kaybı ve eşlerin uyumunu inceleyin.

SortMeRNA **4.3.4** sürümünün varsayılan veri tabanını (`database.tar.gz`) indirip `smr_v4.3_default_db.fasta` dosyasını kullanın. Veri tabanı sürümünü ve kökenini kaydedin. Doğruluk kestirimleri karşılaştırmalı değerlendirme dizilerine ve hata koşullarına bağlıdır; sürüm sayfasındaki bir ölçüm, bu organizma veya kütüphane için garanti değildir.

Komut satırı

```
DB=~/sortmerna/db/smr_v4.3_default_db.fasta
WORK=sortmerna_work/SRR1234567   # a new, sample-specific directory

sortmerna \
  --ref "$DB" \
  --reads trimmed/SRR1234567_1.trim.fq.gz \
  --reads trimmed/SRR1234567_2.trim.fq.gz \
  --paired_in \
  --fastx \
  --aligned rrna_clean/SRR1234567_rRNA \
  --other    rrna_clean/SRR1234567_norRNA \
  --out2 \
  --threads 8 \
  --workdir "$WORK"

# Keep this work directory; use a different one for the next sample.
```

Temel parametreler

`--ref`, rRNA veri tabanıdır; çift uçlu veri için `--reads` *iki kez* verilir. `--paired_in`, uçlardan *biri* rRNA ise ikisini de `--aligned` çıktısına gönderir; bu korumacı bir uzaklaştırmadır. Uçlardan biri rRNA dışı olduğunda çifti korumak için `--paired_out` kullanılır ve daha çok okuma tutulur. `--fastx` FASTA/FASTQ yazar. `--aligned`/`--other`, rRNA ve tutulan okuma ön ekleridir. `--out2`, eşleri *ayrı* R1/R2 dosyalarına yazar; `--threads` iş parçacığını, `--workdir` çalışma dizinini belirler.

Girdiler ve çıktılar

Girdi: Kırpılmış çift uçlu FASTQ. **Çıktı (`--out2` ile):** Korunan okumalar için `*_norRNA_fwd.fq[.gz]` ve `*_norRNA_rev.fq[.gz]`; uzaklaştırılan rRNA için `*_rRNA_*`.

Dikkat edilmesi gerekenler

`--workdir`, indeksleri ve örneğe özgü çalışma durumunu içerir. Her örnek için yeni, ayrı bir dizin kullanın; ortak dizini özyinelemeli biçimde temizlemeyin. Belgelenmiş sürdürme yolunu veya yeni çıktı konumunu seçmeden önce mevcut çalışmayı inceleyin. `--out2` zaten düzenli R1/R2 dosyaları verdiği için sonraki BBMap yeniden eşleme adımı yalnızca *iç içe* çıktı üretildiyse veya eşlerin sırası bozulduysa gerekir.

6.9 Referanssız rRNA uzaklaştırma (RiboDetector) İsteğe bağlı

RiboDetector, okumaları rRNA veri tabanına hizalamak yerine eğitilmiş bir sinir ağı ile rRNA/rRNA dışı olarak sınıflandırır. Referans indirme gerektirmez, CPU üzerinde hızlı çalışır. Organizmanız için küratörlü bir rRNA veri tabanı yoksa veya böyle bir kaynağı yönetmek istemiyorsanız SortMeRNA yerine kullanın. Doğrudan düzenli eşli çıktı yazdığı için ardından BBMap yeniden eşlemesi yapılmaz.

Komut satırı

```
# -l is the read length; set it to your data (here 150 bp)
ribodetector_cpu -t 8 -l 150 \
  -i trimmed/SRR1234567_1.trim.fq.gz trimmed/SRR1234567_2.trim.fq.gz \
  -e norrna --chunk_size 256 \
  -o rrna_clean/SRR1234567_1.fq.gz rrna_clean/SRR1234567_2.fq.gz
```

Temel parametreler

`-i`, R1 ve R2 girdileridir; tek uçlu veri için tek dosya verilir. `-o`, karşılık gelen rRNA dışı çıktılardır. `-l`, **okuma uzunluğudur**; model gerçek uzunluğu bekler. `-e norrna`, güvenle rRNA dışı sınıflandırılan okumaları koruyan korumacı *ensure* kipidir. `--chunk_size`, yığın başına okuma sayısıdır; bellek kısıtlıysa azaltın. `-t`, iş parçacığı sayısıdır.

Girdiler ve çıktılar

Girdi: Kırpılmış çift uçlu FASTQ. **Çıktı:** Zaten eşli olan rRNA dışı okumalar; `rrna_clean/*_1.fq.gz` ve `*_2.fq.gz`.

Dikkat edilmesi gerekenler

`-l` değerini gerçek okuma uzunluğuna ayarlayın; yanlış değer sınıflandırıcı başarımını düşürür. RiboDetector yaklaşık ∼2 GB PyTorch yığını kurar; yalnızca bu yolu kullanacaksanız ekleyin. Çıktı zaten eşli ve uç başına adlandırılmış olduğundan BBMap yeniden eşlemesini atlayın ve `rrna_clean/` içeriğini doğrudan STAR'a verin. Yardımcı betikler iki rRNA yolunu da `<sample>_1.trim.fq.gz`/`_2.trim.fq.gz` biçimine getirir. Veri tabanına dayalı, izlenebilir uzaklaştırma için **SortMeRNA**; uygun rRNA referansı olmadığında **RiboDetector** tercih edilebilir. Önce kontaminasyon taraması ile rRNA içeriğini inceleyin (Bölüm 6.4).

6.10 Okumaları yeniden eşleştirme (BBMap `repair.sh`) İsteğe bağlı

Bu adımı yalnızca rRNA filtresinden geçen okumalar iç içe veya sıra dışıysa kullanın. `repair.sh`, eşleşen R1/R2 sırasını düzeltir; eşi bulunmayan okumaları tekil okumalar dosyasına gönderir.

Komut satırı

```
repair.sh \
  in=rrna_clean/SRR1234567_norRNA.fq.gz \
  out1=rrna_clean/SRR1234567_R1.fq.gz \
  out2=rrna_clean/SRR1234567_R2.fq.gz \
  outs=rrna_clean/SRR1234567_singletons.fq.gz \
  overwrite=t
```

Temel parametreler

`in =`, iç içe girdidir; iki dosya için `in1 =/in2 =` kullanılır. `out1 =/out2 =`, yeniden eşleştirilmiş R1/R2; `outs =`, eşi bulunmayan okumalar; `overwrite =t`, üzerine yazma iznidir. Büyük dosyalarda Java belleğini `-Xmx` ile ayarlayın; örneğin `repair.sh -Xmx8g ...`.

Dikkat edilmesi gerekenler

SortMeRNA'yı atladıysanız bu adımı da atlayıp `trimmed/` okumalarını STAR'a verin. Hizalama döngüsünün R1/R2'yi otomatik bulabilmesi için dosya adlarını tutarlı tutun.

6.11 Genom indeksini oluşturma (STAR)

Hizalama için referans genom ile anotasyonun birbiriyle eşleşmesi gerekir. NCBI veya Ensembl üzerinden genomu FASTA, anotasyonu GTF biçiminde indirin; ardından genom başına bir kez STAR indeksi oluşturun.

Komut satırı

```
# Set READLEN from the protocol and SAINDEX from FASTA length (formula below).
# The companion index script computes SAINDEX from the uncompressed FASTA.
STAR --runMode genomeGenerate \
  --genomeDir ref/ \
  --genomeFastaFiles ref/genome.fna \
  --sjdbGTFfile ref/genome.gtf \
  --sjdbOverhang "$((READLEN-1))" \
  --genomeSAindexNbases "$SAINDEX" \
  --runThreadN 12
```

Temel parametreler

`--genomeDir`, indeks çıktısı; `--genomeFastaFiles`, genom FASTA dosyası; `--sjdbGTFfile`, ekzon birleşim noktası anotasyonudur. `--sjdbOverhang`, *okuma uzunluğu* −1 olmalıdır; 150 bp okuma için 149. Varsayılan 100, yaklaşık ∼100–150 bp için kullanılabilir. `--genomeSAindexNbases`, **küçük genomlarda azaltılmalıdır**: $\min(14,\ \lfloor \log_2(\text{GenomeLength})/2 - 1 \rfloor)$; örneğin 35 Mb mantar genomunda yaklaşık ∼11, memelilerde 14. `--runThreadN`, iş parçacığı sayısıdır.

Girdiler ve çıktılar

Girdi: `genome.fna` ve `genome.gtf`. **Çıktı:** `ref/` içindeki indeks dosyaları: SA, SAindex, Genome ve diğerleri.

Dikkat edilmesi gerekenler

Memeli dışı küçük genomlarda `--genomeSAindexNbases` değerini 14 bırakmak uyarı ve hizalama sorunları yaratabilir; değeri genom uzunluğundan hesaplayın. GTF standart dışı öznitelik adları kullanıyorsa `--sjdbGTFtagExonParentGene` ayarını belirtin; GFF3 için `--sjdbGTFtagExonParentTranscript Parent` gerekir. Bellek gereksinimi derlemeye, indeks ayarlarına, iş parçacığına ve sıralamaya bağlıdır. Seyrek sonek dizisi indeks belleğini azaltabilir; hizalama ve sıralama tepe değerlerini ayrı değerlendirin (Bölüm 5.1).

6.12 Hizalama (STAR ile sıralı BAM)

Her örneğin çift uçlu okumalarını indekslenmiş genoma hizalayın ve koordinata göre sıralı BAM yazın. İsteğe bağlı rRNA adımlarını kullandıysanız `--readFilesIn` için `trimmed/` yerine `rrna_clean/` okumalarını seçin. STAR hızlıdır ancak yüksek bellek ister; daha düşük bellekli intron çıkarılmasını gözeten alternatif için HISAT2'ye bakın (Bölüm 6.13).

Komut satırı

```
STAR --runMode alignReads \
  --genomeDir ref/ \
  --readFilesIn trimmed/SRR1234567_1.trim.fq.gz trimmed/SRR1234567_2.trim.fq.gz \
  --readFilesCommand zcat \
  --outSAMtype BAM SortedByCoordinate \
  --quantMode GeneCounts \
  --runThreadN 12 \
  --outFileNamePrefix mapped/SRR1234567_

samtools index mapped/SRR1234567_Aligned.sortedByCoord.out.bam
samtools flagstat mapped/SRR1234567_Aligned.sortedByCoord.out.bam
```

Temel parametreler

`--readFilesIn` sırası R1, ardından R2'dir. `--readFilesCommand zcat`, `.gz` dosyalarını işlem sırasında açar. `--outSAMtype BAM SortedByCoordinate`, doğrudan sıralı BAM üretir. `--quantMode GeneCounts`, featureCounts ile karşılaştırma veya doğrudan sayım matrisi için kullanılabilen `*ReadsPerGene.out.tab` dosyasını da yazar (Bölüm 6.19). `--outFileNamePrefix`, **yolu ve sonda ayırıcıyı** içermelidir; burada son karakter `_` biçimindedir. Yeni birleşimleri daha duyarlı saptamak için `--twopassMode Basic` eklenebilir.

Girdiler ve çıktılar

Girdi: Temiz çift uçlu FASTQ ve STAR indeksi. **Çıktı:** `mapped/SRR1234567_Aligned.sortedByCoord.out.bam` ve `.bai`; hizalama istatistiklerini MultiQC'ye veren `*_Log.final.out`; ayrıca `*_SJ.out.tab`.

Dikkat edilmesi gerekenler

Uyumsuz genom/GTF sürümleri veya yanlış `--readFilesCommand`, düşük eşleme oranının sık nedenleridir. `*_Log.final.out` içindeki *Uniquely mapped % (tekil hizalanan yüzde)* alanını inceleyin. IGV'de görüntülemeden önce BAM'ı indeksleyin. `samtools merge`, aynı örneğin teknik tekrarlarını veya dizileme hattı dosyalarını birleştirmek içindir; R1/R2'yi birleştirmek için kullanılmaz.

6.13 Alternatif hizalayıcı: HISAT2

HISAT2, STAR gibi intron çıkarılmasını gözeten bir hizalayıcıdır; hiyerarşik graf FM indeksi kullanır. Temel uygulama avantajı bellektir: İnsan genomu hizalaması STAR'ın yaklaşık ∼30 GB gereksinimine karşı birkaç GB RAM ile yürütülebilir; bu nedenle daha düşük bellekli donanıma uygun olabilir [49]. `conda install -c bioconda hisat2` ile kurun. Linux/WSL2'de yerel çalışır; Apple Silicon için arm64 çözümü yoksa Bölüm 5.1 içindeki emüle `osx-64` ortamını kullanın. Genoma hizalanmış BAM, STAR yolundaki gibi featureCounts veya htseq-count girdisi olur.

Genom yolu (önerilen). İntron çıkarılma bölgesi bilgisini GTF'den bir kez çıkarın, daha az RAM isteyen sade indeksi oluşturun; bölge bilgisini hizalama sırasında verin. Böylece büyük genomlarda tam graf indeksi oluşturmanın çok yüksek bellek maliyetinden kaçınılır.

Komut satırı

```
# 1. splice sites from the annotation (script ships with HISAT2)
hisat2_extract_splice_sites.py ref/genome.gtf > ref/genome.ss

# 2. build a plain genome index -> ref/genome_idx.1.ht2 ... .8.ht2 (8 files, current dir; no
    ↪ folder)
hisat2-build -p 8 ref/genome.fna ref/genome_idx

# 3. align (paired-end, dUTP reverse); supply known junctions here, sort in one pipe
hisat2 -p 8 --rna-strandness RF \
  --known-splicesite-infile ref/genome.ss \
  -x ref/genome_idx \
  -1 trimmed/SRR1234567_1.trim.fq.gz -2 trimmed/SRR1234567_2.trim.fq.gz \
  --summary-file mapped/SRR1234567.hisat2.summary \
  | samtools sort -@ 4 -o mapped/SRR1234567.sorted.bam -
samtools index mapped/SRR1234567.sorted.bam
```

Temel parametreler

`--dta`, StringTie/Cufflinks birleştirmesi için birleşim bildirimini ayarlayan isteğe bağlı bir seçenektir; yalnızca featureCounts/DESeq2 sayım yolunda kullanmayın. `--rna-strandness RF`, dUTP/reverse çift uçlu veriye karşılık gelir; FR ileri, R/F tek uçlu içindir. `--known-splicesite-infile`, bilinen birleşimleri hizalama sırasında ekler; indekse gömmekten daha düşük maliyetlidir. `--summary-file`, MultiQC'nin okuyabildiği raporu yazar. Çıktıyı doğrudan `samtools sort` işlemine yönlendirmek büyük ara SAM dosyasını önler.

Kompakt genomlar için ayrı örnek yol. Transkriptom koordinatlarındaki hizalama, kompakt genomlar için de ayrı bir açıklayıcı seçenektir. Kullanışlılığı anotasyonun tamlığına ve transkript belirsizliğine bağlıdır; intronların kısa olması tek başına genom hizalamasına eşdeğerlik göstermez. Transkript bolluğu kestirimi için desteklenen Salmon/tximport yolunu tercih edin. Ensembl üzerinden `*.cdna.all.fa` dosyasını indirdikten sonra şu komutlar kullanılabilir:

Komut satırı

```
gunzip Fusarium_graminearum.ASM24013v3.cdna.all.fa.gz
hisat2-build Fusarium_graminearum.ASM24013v3.cdna.all.fa cdna_idx      # 8 .ht2 files,
    ↪ working dir
hisat2 -x cdna_idx -1 trim_R1.fastq -2 trim_R2.fastq \
  | samtools sort -o kont1.sorted.bam -
```

Dikkat edilmesi gerekenler

`hisat2-build`, sekiz `.ht2` dosyasını verdiğiniz temel adla **geçerli dizine** yazar; klasör oluşturmaz. cDNA'ya hizalı BAM *transkript* koordinatlarındadır; genom GTF dosyasıyla sayılamaz, aksi hâlde sayımlar sıfıra yakın olur. Transkript başına sayıp `tx2gene` eşlemesiyle gen düzeyinde toplayın veya omurgalı/intron bakımından zengin genomlarda genom yolunu kullanın. Bu kısa yol yeni birleşimler de keşfedemez. Memeli genomunda `--ss`/`--exon` ile tam graf indeksi oluşturmak çok yüksek RAM ister; insan için yaklaşık ∼160 GB. 24 GB makinede hazır HISAT2 indeksi veya yukarıdaki sade indeks ile `--known-splicesite-infile` yolunu seçin.

6.14 Kütüphanenin iplik yönlülüğü bilgisini belirleme

İplik yönlü protokoller her okumanın hangi genomik zincirden geldiğini kaydeder. Sayım aracına doğru kural verilmezse okumaların çoğu kaybolabilir; aşağıda açıklanan sessiz hata budur. Yaygın Illumina kimyasında (TruSeq Stranded, NEBNext), ikinci cDNA zinciri dUTP ile işaretlenip parçalanır; dizilenen parça mRNA'nın *ters tamamlayıcısını* bildirir. Sayım araçlarında bu, ters iplik yönlülüğüne karşılık gelir: `featureCounts -s 2`, `htseq-count --stranded reverse`, Salmon ISR. İplik yönsüz kütüphaneler `-s 0`, daha seyrek ileri yönlü (FR) protokoller `-s 1` kullanır. İplik yönlülüğü bilgisi, karşı zincirlerde örtüşen genlerin doğru ayrılmasını da sağlar.

Ayarı tahmin etmek yerine hizalanmış bir örnekten kanıt toplayın.

Komut satırı

```
# Option A -- RSeQC: compare each read's strand to the annotated transcript strand
infer_experiment.py -r ref/genes.bed -i mapped/SRR16356854_Aligned.sortedByCoord.out.bam

# Option B -- read it off STAR's per-gene counts (needs --quantMode GeneCounts):
#   col2 = unstranded, col3 = forward (-s 1), col4 = reverse (-s 2)
#   Compare col3 versus col4; do not choose the largest of all three.
grep -v "^N_" mapped/SRR16356854_ReadsPerGene.out.tab \
  | awk '{u+ =$2; f+ =$3; r+ =$4} END{print "unstranded="u" forward="f" reverse="r}'
```

Temel parametreler

`infer_experiment.py`, her yön kuralıyla açıklanan okuma oranını verir. Yaklaşık ∼50/50 dağılım iplik yönsüz veriyi destekler. Çift uçlu veride `"1++,1--,2+-,2-+"` örüntüsünün baskınlığı ileri yönü (`-s 1`), `"1+-,1-+,2++,2--"` örüntüsünün baskınlığı ters yönü (`-s 2`) destekler. Belirlenemeyen oranı ve kütüphane protokolünü birlikte inceleyin. [71] `-r`, 12 sütunlu BED gen modeli ister; bunu GTF'den UCSC `gtfToGenePred`, ardından `genePredToBed` ile bir kez oluşturun.

Uygulama notu

Salmon bu seçimi otomatik çıkarabilir: `salmon quant -l A`, kütüphane türünü kestirip `lib_format_counts.json` içine kaydeder. Kullanılan yöntem ne olursa olsun çalışmanın yön ayarını belirleyin ve tüm örneklerde aynı kurala göre kullanın.

6.15 Gen düzeyinde nicelendirme (featureCounts)

featureCounts, hizalanmış okuma *çiftlerini* GTF anotasyonundaki genlere atayarak gen × örnek sayım matrisi oluşturur. Tüm örnekleri tek matriste toplamak için BAM dosyalarını aynı çağrıya verin. Aynı aşama için yaygın bir alternatif htseq-count aracıdır (Bölüm 6.18).

Komut satırı

```
featureCounts \
  -a ref/genome.gtf \
  -o counts/counts.txt \
  -T 6 \
  -p --countReadPairs \
  -t exon -g gene_id \
  -s 2 \
  -Q 10 \
  mapped/*_Aligned.sortedByCoord.out.bam
```

Temel parametreler

`-a`, anotasyon (GTF); `-o`, çıktı matrisi; `-T`, iş parçacığı sayısıdır. `-p`, girdinin çift uçlu olduğunu belirtir. Subread ≥2.0.2 sürümlerinde okuma yerine *parça* saymak için `--countReadPairs` ekleyin. `-t exon -g gene_id`, ekzonları gene göre gruplar; bunlar varsayılan değerlerdir. `-s`: 0 iplik yönsüz, 1 ileri iplik yönlü, 2 ters iplik yönlü. `-Q 10`, en düşük eşleme kalitesi eşiğidir.
Bu öğretici eşik, baz çağrısının kalitesini değil hizalamanın güvenini değerlendirir. Yükseltilmesi belirsiz atamaları azaltabilir; zor bölgelerdeki bilgi taşıyan parçaları da dışlayabilir. Düşürülmesi, kökeni belirsiz daha fazla hizalamayı korur. Değişiklikten sonra atama kategorilerini ve kalan sayımları karşılaştırın. Eşleme kalitesi ölçekleri ve özel değerler hizalayıcıya bağlıdır; sabit bir eşik evrensel olarak tekil eşleme garantisi vermez.

Girdiler ve çıktılar

Girdi: Sıralı BAM dosyaları ve GTF. **Çıktı:** `counts/counts.txt`; önce altı anotasyon sütunu (Geneid, Chr, Start, End, Strand, Length), ardından BAM başına bir sayım sütunu. `counts/counts.txt.summary`, MultiQC'ye aktarılan atama istatistiklerini içerir.

Dikkat edilmesi gerekenler

İplik yönlülüğü, önemli bir sessiz hata kaynağıdır. Yanlış `-s`, okumaların çoğunu `Unassigned_NoFeatures` kategorisine gönderebilir. Çoğu Illumina dUTP kütüphanesi ters yönlüdür (`-s 2`); emin değilseniz RSeQC veya STAR ileri/ters toplamlarını protokolle birlikte karşılaştırın. İplik yönsüz sayım sütunu, yönlü bir kütüphanede bile en büyük toplamı verebilir; 0/1/2 arasından en yüksek değeri seçmek yönlülüğü belirlemez. [71] R içine aktarırken ilk altı sütunun anotasyon olduğunu unutmayın.

6.16 Kalite kontrolü özeti: Eşleme ve atama oranları

Genel hizalama oranı, hizalanan okumaların oranıdır. *Atama oranı* ise sayım kipine göre işlenen parça veya okumalar arasından genlere atananların oranını gösterir. STAR hizalama istatistikleri `*_Log.final.out` dosyasında bulunur.

`counts.txt.summary`, featureCounts atama istatistiklerini içerir. Bunları tabloya almak için:

Komut satırı

```
# uniquely + multi-mapped % from each STAR log -> a tab-separated table
for f in mapped/*_Log.final.out; do
  s=$(basename "$f" _Log.final.out)
  uniq=$(grep "Uniquely mapped reads %" "$f" | grep -o "[0-9.]*%")
  multi=$(grep "mapped to multiple loci" "$f" | grep -o "[0-9.]*%" | head -1)
  printf "%s\t%s\t%s\n" "$s" "$uniq" "$multi"
done | sort | column -t

# overall mapping rate straight from a BAM
samtools flagstat mapped/SRR1234567_Aligned.sortedByCoord.out.bam | grep "mapped ("

# featureCounts assignment per sample (Assigned counts; MultiQC converts these to %)
grep -E "^(Status|Assigned)" counts/counts.txt.summary | column -t
```

Uygulama notu

MultiQC, FastQC, fastp, STAR ve featureCounts örnek ölçütlerini elektronik tabloya uygun `multiqc_data/multiqc_general_stats.txt` dosyasında toplar. Sonunda `multiqc .` çalıştırıp her günlüğü ayrı ayrıştırmak yerine bu dosyayı veya HTML raporunu okuyun. Düşük tekil eşleme oranı genom/GTF uyumsuzluğu veya kontaminasyon; düşük atama oranı ise çoğu zaman yanlış `-s` ayarı açısından incelenmelidir.

6.17 Hizalama sonrası RNA QC (RSeQC) İsteğe bağlı

RSeQC, eşleme oranına ek olarak RNA ve kütüphane kalitesine ilişkin tanısal bilgiler sağlar.

Okuma dağılımı, `read_distribution.py` ile raporlanır. Okumalar ekzon, intron, UTR veya genler arası olarak sınıflanır; yüksek intronik veya genler arası oran DNA kontaminasyonu ya da yaygın pre-mRNA için bir işaret olabilir.

Gen gövdesi kapsamı, `geneBody_coverage.py` ile raporlanır. 5' ucundan 3' ucuna kapsamın düzgünlüğünü ölçer; 3' yanlılığı bozulmuş RNA'yı düşündürebilir. İki komut da BAM ve 12 sütunlu BED gen modeli gerektirir.

Komut satırı

```
# build the 12-column BED once from the GTF (the same model infer_experiment.py used, the
    ↪ strandedness section)
gtfToGenePred ref/genome.gtf ref/genome.genePred
genePredToBed ref/genome.genePred ref/genes.bed

read_distribution.py -i mapped/SRR1234567_Aligned.sortedByCoord.out.bam -r ref/genes.bed \
  > qc/rseqc/SRR1234567.read_dist.txt
geneBody_coverage.py -i mapped/SRR1234567_Aligned.sortedByCoord.out.bam -r ref/genes.bed \
  -o qc/rseqc/SRR1234567
multiqc qc/rseqc/ -o qc/rseqc/ -n multiqc_rseqc
```

Temel parametreler

`-i`, sıralı ve indekslenmiş BAM'dır. `-r`, UCSC `gtfToGenePred` → `genePredToBed` ile bir kez oluşturulan 12 sütunlu BED gen modelidir. `geneBody_coverage.py` için `-o`, örnek eğrisinin ve MultiQC özetinin çıktı ön ekidir.

Girdiler ve çıktılar

Girdi: Sıralı BAM, `.bai` ve `ref/genes.bed`. **Çıktı:** `qc/rseqc/` içinde okuma dağılımı tabloları ve gen gövdesi kapsam eğrileri; MultiQC bunları özetler.

Dikkat edilmesi gerekenler

Bunlar filtre değil, rapordur; 3' yanlılığıyla RNA bozulmasını veya yüksek genler arası oranla genomik kontaminasyonu araştırmaya yardım eder. `geneBody_coverage.py` yavaştır; tüm örnekler yerine temsili birkaç örnekte çalıştırılabilir. BED dosyasını hizalama ve sayımda kullanılan *aynı* GTF'den oluşturun.

6.18 Alternatif sayım aracı: htseq-count

htseq-count, DESeq yaklaşımından gelen birleşim kipinde çalışan sayım aracıdır ve featureCounts'a yaygın bir alternatiftir [50, 51]. Sıralı BAM ile GTF okur, gen başına sayım yazar. Bir okuma yalnızca tek gene belirsizlik olmadan eşlenirse sayılır; iki genle örtüşenler `__ambiguous`, birden fazla konuma eşlenenler `__alignment_not_unique` kategorisine gider. Böylece sayımlar DESeq2 modeline uygun girdi olarak kalır.

Komut satırı

```
htseq-count -f bam -r pos -s reverse -t exon -i gene_id -m union \
  mapped/SRR1234567_Aligned.sortedByCoord.out.bam ref/genome.gtf \
  > counts/SRR1234567.tsv

# if you assemble the matrix by hand, drop the trailing summary rows first:
grep -v "^__" counts/SRR1234567.tsv > counts/SRR1234567.clean.tsv
```

Temel parametreler

`-f bam -r pos`, koordinata göre sıralı BAM okur; sabit bellekle eşli sayım için ada göre sıralı BAM'da `-r name` kullanın. `-s reverse`, dUTP/TruSeq yönüdür ve `featureCounts -s 2` ile eşleşir; `-s yes`, `-s 1`; `-s no`, `-s 0` karşılığıdır. `-t exon -i gene_id`, özellik türü ve gruplama özniteliğidir. `-m union`, örtüşme kipidir; `intersection-strict` ve `intersection-nonempty` seçenekleri de vardır. HTSeq 2, çok çekirdekli sayım için `-n <N>` sağlar.

R

```
# read straight into DESeq2 (strips the trailing __ rows automatically)
library(DESeq2)
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
cfg <- read_analysis()
samples <- read_samples(cfg = cfg)
samples$sampleName <- samples$sample
samples$fileName <- paste0(samples$sample, ".tsv")
samples <- samples[, c("sampleName", "fileName", "condition",
                       setdiff(names(samples), c("sampleName", "fileName", "condition")))]
dds <- DESeqDataSetFromHTSeqCount(sampleTable = samples,   # cols: sampleName, fileName,
                                                           # condition
                                  directory   = "counts",
                                  design      = cfg$design)
```

Dikkat edilmesi gerekenler

`htseq-count` varsayılanı iplik yönsüz değil, `-s yes` biçimindedir. Ters yönlü veride bu ayar bırakılırsa okumaların önemli bir bölümü `__no_feature` kategorisine gidebilir; featureCounts'ta olduğu gibi `-s` seçimini açıkça yapın. `DESeqDataSetFromHTSeqCount`, sondaki özet satırlarını (`__no_feature`, `__ambiguous`, `__alignment_not_unique` ve diğerleri) otomatik çıkarır. Matrisi kendiniz oluşturuyorsanız bu satırları yukarıdaki gibi kaldırın. htseq-count, featureCounts'tan yavaştır; Linux ve macOS üzerinde, Windows'ta ise WSL2 ile çalışır.

6.19 Alternatif sayım: STAR gen sayımları (`--quantMode GeneCounts`)

STAR hizalaması zaten `--quantMode GeneCounts` ile çalıştırıldı (Bölüm 6.12); her örnek üç yön kuralında gen sayımlarını içeren `*_ReadsPerGene.out.tab` dosyasına sahiptir. Bu tabloları birleştirmek featureCounts geçişini atlar. STAR, htseq `union` sayım kurallarını kullanır; çıktı DESeq2 girdisidir ve featureCounts ile karşılaştırılabilir.

R

```
#### Build the count matrix from STAR ReadsPerGene tables ####
files <- list.files("mapped", "_ReadsPerGene.out.tab$", full.names = TRUE)
names(files) <- sub("_ReadsPerGene.out.tab$", "", basename(files))

scol <- 4                                    # 2 = unstranded, 3 = forward (-s 1), 4 = reverse
                                              # (-s 2)
read_star <- function(f) {
  x <- read.delim(f, header = FALSE, skip = 4)   # drop N_unmapped/N_multimapping/N_
    ↪ noFeature
                                                  # /N_ambiguous
  setNames(x[[scol]], x[[1]])
}
mats <- lapply(files, read_star)
stopifnot(length(unique(lapply(mats, names))) == 1)   # identical gene order in every sample
cts  <- do.call(cbind, mats)
library(DESeq2)
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
cfg <- read_analysis()
cts <- check_counts(cts)
coldata <- read_samples(colnames(cts), cfg)
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata, design = cfg$design)
```

Temel parametreler

`skip = 4`, genlerden önceki dört STAR özet satırını çıkarır. `scol`, yön sütununu seçer ve Bölüm 6.14 kararına uymalıdır: 2 iplik yönsüz, 3 ileri, 4 ters. `check_counts()`, saklama kipini değiştirmeden önce tamsayı sayımları denetler. `read_samples()`, bağımsız hazırlanmış üst veriyi eşler ve yapılandırılmış tasarımı denetler.

Dikkat edilmesi gerekenler

STAR indeksinin `--sjdbGTFfile` ile oluşturulmuş olması gerekir; aksi durumda `ReadsPerGene.out.tab` boş kalır. Bu tablolarda anotasyon sütunu *yoktur*. İlk altı sütunu atan featureCounts içe aktarıcısını kullanmayın; gerçek örnek sütunlarını silmiş olursunuz (Bölüm 8). Çalışma boyunca tek bir `scol` seçin. Sayım araçlarının atama ve çoklu eşleme kuralları farklıdır; karşılaştırmayı eşleşen anotasyon ve ayarlarla yapın, önemli ayrılıkları araştırın.

6.20 Hizalamasız nicelendirme (Salmon)

Salmon ve kallisto, baz düzeyinde genom hizalamasını atlar; okumaları *transkriptoma* eşleyip transkript bolluklarını doğrudan kestirir. Bu yol hızlıdır ve STAR'ın onlarca GB belleğine karşı birkaç GB kullanabilir. Salmon, parça GC ve dizi yanlılığı etkilerini de modeller. Transkript kestirimlerini tximport ile gen düzeyine aktarın; uygun DESeq2 oluşturucusu transkript uzunluğu ofset değerlerini sağlar. Yardımcı kodda ayrı bir Salmon başlangıç betiği bulunur.

Komut satırı

```
# Transcript-only illustration; use a decoy-aware transcriptome for selective alignment.
# Pass --decoys decoys.txt with a matching transcriptome-plus-decoy FASTA.
salmon index -t ref/transcripts.fa -i ref/salmon_idx -k 31
salmon quant -i ref/salmon_idx -l A \
  -1 trimmed/SRR16356854_1.trim.fq.gz -2 trimmed/SRR16356854_2.trim.fq.gz \
  --gcBias -p 8 -o quant/SRR16356854
```

R

```
library(tximport); library(DESeq2)
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
cfg <- read_analysis()
coldata <- read_samples(cfg = cfg)
files <- setNames(file.path("quant", coldata$sample, "quant.sf"), coldata$sample)
tx2gene <- read.delim("config/tx2gene.tsv", colClasses = "character")
txi <- tximport(files, type = "salmon", tx2gene = tx2gene)   # tx2gene: transcript -> gene
                                                              # map
dds <- DESeqDataSetFromTximport(txi, colData = coldata, design = cfg$design)
```

Dikkat edilmesi gerekenler

Diferansiyel ifade için *ham sayımlar* kullanın veya transkript uzunluğunu düzelten ofseti `tximport` aracılığıyla sağlayın. Salmon, seçici hizalama sırasında yanıltıcı eşleşmeleri denetlemek için genomik dizilerin (decoy) kullanılmasını önerir; yalnızca transkript içeren indeks daha basit bir örnektir. [73] Sayım temelli DE analizini TPM veya FPKM üzerinde yapmayın: Bunlar örnek içi normalleştirmelerdir ve modellerin gerektirdiği sayım düzeyi bilgisini kaybettirir.

6.21 Tek uçlu okumalar

Gövdede çift uçlu veri varsayılır. Tek uçlu (SE) kütüphaneler okuma işleme seçeneklerini değiştirir; sayım matrisinden sonraki DESeq2, zenginleşme ve şekil aşamaları aynı çerçeveyi kullanır. Aşama başına değişiklikler şöyledir:

- **İndirme.** SE kaydı tek FASTQ üretir; `_1/_2` ayrımı yoktur. fasterq-dump varsayılan `--split-3` kipi bunu işler.
- **fastp.** `--detect_adapter_for_pe` seçeneğini kaldırın; bu seçenek çift uçlu örtüşme saptaması içindir. Yalnızca `-i/-o` kullanın; `-I/-O` eklemeyin. SE adaptör kırpma varsayılan olarak açıktır; fastp fazla temsil edilen 3' adaptörünü arar. Otomatik saptama kaçırıyorsa `--adapter_sequence <SEQ>` verin. `-q/-l` değişmez.
- **STAR.** `--readFilesIn reads.fq.gz` için tek dosya verin.
- **HISAT2.** `-1/-2` yerine `-U reads.fq.gz`, yön için `--rna-strandness R` veya `F` kullanın.
- **featureCounts.** Çift uçlu veriye özgü `-p` ve `--countReadPairs` seçeneklerini kaldırın; `-s` kalsın. htseq-count değişiklik gerektirmez.
- **İplik yönlülüğü.** RSeQC `infer_experiment.py` çalışmaya devam eder; SE için read1/read2 örüntüleri yerine `"++,--"` ve `"+-,-+"` bildirir.

Dikkat edilmesi gerekenler

Tek uçlu BAM üzerinde `-p/--countReadPairs` bırakmak featureCounts sayımını bozabilen yaygın bir hatadır. SE, gen düzeyinde DE için ekonomik bir seçenektir; çift uçlu veri izoformları, ekzon birleşim noktası bölgelerini ve çoklu eşlenen okumaları daha iyi çözebilir. İzoform düzeyindeki sorularda çift uçlu tasarımı tercih edin.

7 Alternatif başlangıç noktaları

Önceki bölümlerde ham okumalardan sayım matrisi elde edildi. Elinizde işlenmiş veri varsa aynı analizlere uygun bir ara aşamadan başlayabilirsiniz. Hazır sayım matrisi veya GEO mikrodizi serisi, diferansiyel gen ifadesi aşamasının girdisidir (Bölüm 8). Önceden hesaplanmış sonuç tablosu ise gen başına istatistikleri zaten içerdiği için doğrudan zenginleşme ve uygun şekil aşamalarına bağlanır (Bölüm 9 ve 11).

7.1 Bir sayım matrisi ile başlama

Gen × örnek sayım matrisi önceden üretilmiş veya indirilmişse kabuk aşamalarını atlayıp doğrudan DESeq2 içine alın. Düz matriste bulunmayan anotasyon sütunlarını yanlışlıkla silmemek için featureCounts içe aktarıcısından değil, *veri kümesini kurma, ön filtreleme ve referans düzeyi ayarlama* adımından devam edin (Bölüm 8).

R

```
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
cfg <- read_analysis()
cts <- read_count_matrix("counts/matrix.tsv")
coldata <- read_samples(colnames(cts), cfg)
dds <- DESeqDataSetFromMatrix(cts, coldata, design = cfg$design)
```

Dikkat edilmesi gerekenler

Ham tamsayı sayımlar kullanın. Ondalıklı değerler normalleştirilmiş nicelikler veya kestirilmiş sayımlar olabilir; kökenlerini belirleyin. Desteklenen transkript kestirimlerini `tximport` ile alın (Bölüm 6.20); TPM'yi yuvarlayıp DESeq2'ye vermeyin. Önce sondaki htseq/STAR özet satırlarını çıkarın. Gen kimliği türünü (Ensembl, symbol, Entrez) bu aşamada kaydedin; anotasyon ve zenginleşme buna bağlıdır (Bölüm 8.6 ve 9).

7.2 Bir diferansiyel gen ifadesi sonuç tablosuyla başlama

Bir çalışma arkadaşının DESeq2/edgeR/limma tablosu veya bir çalışmanın ek dosyası, diferansiyel ifade istatistiklerini hazır sağlayabilir. Bu durumda istatistikleri yeniden hesaplamadan zenginleşme, ağ ve yalnızca sonuç tablosu gerektiren şekil aşamalarına geçilebilir. Önce sütun adlarını sonraki kodun beklediği biçime getirin.

R

```
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
cfg <- read_analysis()
res_df <- read.delim("input/de_table.tsv", check.names = FALSE)
stopifnot("gene_id" %in% names(res_df))
check_ids(res_df$gene_id, "Result gene IDs")
rownames(res_df) <- res_df$gene_id
# rename to the canonical columns used from here on
names(res_df)[names(res_df) == "logFC"]     <- "log2FoldChange"   # edgeR / limma name
names(res_df)[names(res_df) == "adj.P.Val"] <- "padj"             # limma name
names(res_df)[names(res_df) == "FDR"]       <- "padj"             # edgeR name
stopifnot(all(c("log2FoldChange", "padj") %in% names(res_df)))
res <- check_results(res_df)  # retain gene identities; use the companion result-only
    ↪ adapter
```

Dikkat edilmesi gerekenler

Çalıştırılabilir tablo alma yolu açık bir `gene_id` sütunu, belirtilmiş kimlik ad uzayı ve geçerli sayısal istatistikler ister. Kat değişimi *yönünün* amaçlanan karşılaştırma, yani treated/control yönüyle eşleştiğini doğrulayın; işaretin ters olması sonraki tüm yön yorumlarını tersine çevirir. Sayım gerektiren PCA, örnek uzaklığı, ifade ısı haritası ve seçilmiş gen incelemesi sayımlar olmadan çalışmaz; bunları atlayıp volkan grafiği ve zenginleşmeyi kullanın. MA ayrıca uyumlu bir bolluk ölçüsü ister; genel bir DE tablosu bunu içermeyebilir. Kimlikler gen sembolüyse eşlemede `fromType = "SYMBOL"` seçin (Bölüm 9) ve Ensembl sürüm eki kaldırma satırını kullanmayın.

7.3 limma ile GEO mikrodizi serisi

GEO mikrodizi serisi DESeq2 yerine limma ile analiz edilir ve aynı `res` sonuç yapısını üretir. Yardımcı betikler ifade matrisiyle örnek üst verisini şekillere uyarlar; sayımlara özgü DESeq2 yöntemleri bu matrise uygulanmaz. Normalleştirilmiş seri matrisini alın, belgelenmiş ön işlemeyi ve ifade ölçeğini belirleyin, tek platform seçin, belirsiz olmayan probları genlere özetleyin; sonra belirtilen tasarım ve karşılaştırmayı modelleyin.

R

```
library(GEOquery); library(limma)
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
cfg <- read_analysis()
series <- getGEO("GSE5583", GSEMatrix = TRUE, AnnotGPL = TRUE)
platform <- "<GPL accession>"  # inspect annotation for each returned platform
selected <- vapply(series, function(x) annotation(x) == platform, logical(1))
stopifnot(sum(selected) == 1)
gse <- series[[which(selected)]]
expr <- exprs(gse)                          # probes x samples (usually already normalised)

# log2 only if the values look like linear intensities (GEO2R heuristic)
scale <- "log2"  # explicitly set from the deposit/platform documentation
if (scale == "linear") {
  stopifnot(all(is.finite(expr)), all(expr > 0))
  expr <- log2(expr)
}

# collapse probes to gene symbols, keeping one value per gene
sym <- fData(gse)$"Gene symbol"               # column name varies by platform
keep <- !is.na(sym) & nzchar(sym) & !grepl("///|;|,", sym)
expr <- avereps(expr[keep, , drop = FALSE], ID = sym[keep])

coldata <- read_samples(colnames(expr), cfg) # explicit sample IDs and condition
design <- model.matrix(cfg$design, coldata)
fit    <- eBayes(lmFit(expr, design), trend = TRUE, robust = TRUE)   # limma-trend
res    <- topTable(fit, coef = "conditiontreated", number = Inf) # verify against colnames(
    ↪ design)
names(res)[names(res) %in% c("logFC", "adj.P.Val")] <- c("log2FoldChange", "padj")
```

Log2 ölçeğindeki `expr` matrisi, Bölüm 11'deki örnek yapısı şekillerinde varyansı dengelenmiş matris yerine kullanılır.

Dikkat edilmesi gerekenler

Çoğu GEO seri matrisi zaten normalleştirilmiştir; tekrar normalleştirmeyin veya ikinci kez log dönüşümü yapmayın. Veri sağlayıcının ön işleme kaydını ve platform belgelerini kullanın. Değer aralıkları tanısal ipucudur; tek başına dönüşüm kuralı değildir. Gen sembolü sütununun adı değişebilir: `Gene symbol`, `Gene Symbol`, `gene_assignment`. `fData(gse)` içeriğini inceleyin. Birden çok probu bulunan genleri her gen bir kez görünecek biçimde birleştirin. Ham Affymetrix CEL dosyaları için `affy::ReadAffy()`, ardından `affy::rma()` ile normalleştirilmiş matris elde edilir.

8 DESeq2 ile diferansiyel gen ifadesi

8.1 Sayımları içe alma ve örnek üst veri tablosu oluşturma

Modelde her sayım sütunu, ilgili örneğin koşul ve tasarım bilgileriyle birlikte değerlendirilir. Bu nedenle featureCounts matrisini okuduktan sonra altı anotasyon sütununu çıkarın ve satır sırası sayım sütunlarıyla *tam olarak* eşleşen bir `colData` tablosu oluşturun.

R

```
#### Setup (run once) ####
if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager")
BiocManager::install(c("DESeq2", "apeglm"))
library(DESeq2)
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
cfg <- read_analysis()

#### Import the featureCounts matrix ####
fc            <- read.delim("counts/counts.txt", comment.char = "#", check.names = FALSE)
rownames(fc) <- fc$Geneid
cts           <- as.matrix(fc[, -(1:6)])     # drop Geneid,Chr,Start,End,Strand,Length
cts           <- check_counts(cts)          # reject decimals, missing/negative counts first

# featureCounts names columns after the BAM paths -- tidy them
colnames(cts) <- gsub("_Aligned.sortedByCoord.out.bam$", "", basename(colnames(cts)))

#### Sample metadata: identity, design, and covariates ####
coldata <- read_samples(colnames(cts), cfg)
stopifnot(identical(rownames(coldata), colnames(cts)))
```

Dikkat edilmesi gerekenler

Bağımsız olarak hazırlanmış örnek üst verisini sayım sütunlarına kimlikle eşleyin. Koşullar sütun konumuna göre atanırken üst veri satır adları da aynı sütunlardan oluşturulmuşsa sıra kontrolü işlevsizdir. Yardımcı kod, model kurmadan önce yinelenen/eksik kimlikleri, farklı örnek kümelerini, eksik eş değişken alanlarını ve etkileri karışmış tasarım durumunu reddeder. Sayımlar TPM/FPKM/CPM değil, ham tamsayı olmalıdır; DESeq2 sayımları modeller ve normalleştirmeyi kendi içinde yapar.

8.2 Veri kümesini kurma, ön filtreleme ve referans düzeyini seçme

Varsayılan `~ condition` tasarımı, ifadeyi gruba göre modeller. İçe alma, model kurma ve dışa aktarma boyunca aynı `cfg$design`, referans ve karşılaştırmayı kullanın; tanımlanabilir bir teknik parti veya eşleştirilmiş donör terimi gerektiğinde yapılandırmayı değiştirin. Ön filtreleme, model kurulmadan önce neredeyse boş genleri çıkarır.

R

```
dds <- DESeqDataSetFromMatrix(countData = cts,
                              colData   = coldata,
                              design    = cfg$design)

# keep genes with > = 10 reads in at least the smallest group's size
smallestGroup <- min(table(coldata$condition))
keep          <- rowSums(counts(dds) > = 10) > = smallestGroup
dds           <- dds[keep, ]

# define the baseline level explicitly
dds$condition <- relevel(dds$condition, ref = cfg$reference)
```

Temel parametreler

`cfg$design`, incelenmiş modeli sağlar; örneğin `~ batch + condition`, tanımlanabilir teknik parti etkisini denetler. On sayım kuralı öğretici bir başlangıçtır. Eşiği yükseltmek modele giren gen sayısını ve hesaplama yükünü azaltır; gerçekten değişen düşük bolluklu genleri de dışlayabilir. Eşiği düşürmek daha az bilgi taşıyan testleri korur. Test ailesi değiştiğinden, gücün sürekli artacağını varsaymak yerine kalan genleri ve düşük sayım tanılarını inceleyin. `cfg$reference`, paydadaki referans düzeyini belirler. [69]

Uygulama notu

Bilinen bir teknik parti/dizileme hattı etkisi varsa tasarımda ilgilenilen değişkenden *önce* yer verin: `~ batch + condition`. Tasarım tanımlanabilir olduğunda DESeq2, teknik partiye göre düzeltilmiş koşul etkisini kestirir. Koşulla tamamen karışmış bir teknik parti etkisi modele terim ekleyerek ayrılamaz. VST, PCA'daki teknik parti etkilerini kaldırmaz; yalnızca görüntüleme için yapılan düzeltme biyolojik tasarımı korumalıdır. [69]

8.3 DESeq2 çalıştırma ve sonuçları alma

Kütüphane derinliği ve RNA bileşimi, gen bolluğu aynı ölçüde değişmeden de sayımları değiştirebilir. DESeq2, yeterli sayıda kararlı gen veya gerekçelendirilmiş bir kontrol kümesi varsayımıyla ölçek faktörlerini kestirerek aynı genin örneklerini ortak bir göreli ölçeğe taşır. Bu işlem hücre başına mutlak RNA ölçümü sağlamaz. Tekrarlar, sayım örneklemesinin ötesindeki değişkenliği gösterir; negatif binom modelindeki saçılım bu ek değişkenliği tanımlar. Tasarım beklenen sayımları eş değişkenlere bağlar; koşul katsayısı, tanımlanan $\log_2$ kat değişimini belirsizliğiyle birlikte kestirir. Normalleştirme, teknik partiyle tamamen karışmış koşul etkisini geri kazandıramaz. `DESeq()` bu bileşenleri kestirip modeli kurar; `results()`, sıfır etkiye karşı olağan Wald testini, `lfcShrink()` ise görüntüleme için daha kararlı etki kestirimlerini verir. [69]

R

```
dds <- DESeq(dds)
resultsNames(dds)                          # inspect the coefficient names

#### Wald test: treated vs control ####
res <- results(dds,
               contrast = cfg$contrast,
               alpha    = cfg$alpha)

#### Shrunken log2 fold changes (for ranking, MA & volcano) ####
stopifnot(cfg$coefficient %in% resultsNames(dds))
resLFC <- lfcShrink(dds,
                    coef = cfg$coefficient, res = res,
                    type = "apeglm")

summary(res)
sum(res$padj < cfg$alpha, na.rm = TRUE)         # number of significant genes
```

Temel parametreler

`cfg$contrast`, pay/payda yönündeki karşılaştırmayı tanımlar; varsayılan yön treated/control biçimindedir. Değeri 0,05 olan `cfg$alpha`, hem bağımsız filtreleme hedefini hem de raporlama eşiğini sağlar. Büzülme aşamasının farklı bir çıkarım tablosu üretmemesi için eşleşen `res` nesnesini ve yapılandırmadaki tam katsayıyı `lfcShrink(type = "apeglm")` işlevine verin. Testler için `res` korunur; görüntülemeye yalnızca büzülmüş LFC değerleri aktarılır. [11, 69]

Dikkat edilmesi gerekenler

`padj = NA`, bağımsız filtreleme veya aykırı değer kuralından kaynaklanabilir; bu eksikliği koruyun. Olağan DE seçimi `res$padj < cfg$alpha` koşuluna dayanır. Buna eklenen gözlenen `abs(res$log2FoldChange) > cfg$lfc_cutoff` koşulu, burada sonraki listeler için kullanılan keşif amaçlı süzmedir; kalan alt küme başlangıçtaki FDR güvencesini kendiliğinden taşımaz. Bu seçim, aşağıda açıklanan gerçek etkinin bir marjı aşıp aşmadığı testinden farklıdır. Büzülmüş değerler grafikteki konumları belirler; özgün etkiler ve standart hatalar çıkarım ve çalışmalar arası birleştirme için korunur.

Varsayımsal sonuç. treated/control karşılaştırmasında bir genin ham $\log_2$FC değerinin 1,5, `lfcSE` değerinin 0,3 ve `padj` değerinin 0,004 olduğunu varsayalım. Kestirilen göreli bolluk tedavi örneklerinde yaklaşık $2^{1,5} = 2{,}83$ kattır; karşılaştırma ters çevrildiğinde yaklaşık 0,35 kat olur. Standart hata $\log_2$ ölçeğindeki belirsizliği gösterir; aynı etkide daha büyük standart hata, daha az kesin kestirim demektir. Varsayımsal büzülmüş LFC değeri 1,2 ise özgün test değişmeden grafikteki nokta yer değiştirir. Bu satır olağan 0,05 eşiğini ve gözlenen etki için 1 sınırını geçer; ancak gerçek LFC değerinin 1'i aştığını kanıtlamaz.

Gerektiğinde etki eşiğini hipoteze katın. Olağan DE testinde `padj < 0.05` koşulunu sağlayan genleri seçip gözlenen $|\log_2 FC| > 1$ ile süzmek, kalan alt kümeye başlangıçtaki FDR güvencesini kendiliğinden taşımaz. Bilimsel soru önceden belirlenen bir marjı aşan etki gerektiriyorsa buna uygun eşik testi kullanın; DESeq2'de `lfcThreshold` ve `altHypothesis = "greaterAbs"`, edgeR'de `glmTreat` bu amaçla kullanılabilir. Etkinin bir marjın içinde kaldığına ilişkin kanıt ise eşdeğerlik testiyle değerlendirilir. İki yaklaşım da değişimin tam olarak sıfır olduğunu göstermez. Birincil testi değiştirmeden önce güncel yöntem belgelerini inceleyin [69, 78].

8.4 Eşdeğerlik testi: Anlamlı bir sınır içindeki etkiler İsteğe bağlı

Sıradan bir testin anlamlı olmaması, *değişim için yeterli kanıt bulunmadığı* anlamına gelir; *değişim olmadığının* kanıtı değildir. DESeq2, belirtilen tolerans içindeki etkileri belirlemek için iki tek yönlü teste dayanan eşdeğerlik testi (TOST) sunar. Test, $\log_2$ kat değişiminin $\pm$ `lfcThreshold` bandının *dışında* değil, *içinde* olup olmadığını sorgular [62].

R

```
# Choose this margin from biological relevance before inspecting results.
res_eq <- results(dds, contrast = cfg$contrast,
                  lfcThreshold = cfg$equivalence_margin,
                  altHypothesis = "lessAbs", alpha = cfg$alpha)
equivalent <- subset(as.data.frame(res_eq), padj < cfg$alpha)
write.csv(equivalent, "results/equivalent_within_margin.csv")
```

Temel parametreler

`lfcThreshold = 1`, örnek eşdeğerlik bandıdır: $|\log_2 FC| < 1$, yani iki kat değişim sınırı. Biyolojik olarak gerekçelendirilmiş toleransı analizden önce seçin. `altHypothesis = "lessAbs"`, mutlak etkinin eşikten *küçük* olduğunu sınar; TOST alternatif hipotezi budur. `padj < alpha`, seçilen karşılaştırmanın belirtilen sınır içinde kaldığına dair kanıtı gösterir; etkinin tam sıfır olduğuna dair kanıt değildir. [69]

Girdiler ve çıktılar

Girdi: Modeli kurulmuş `dds`. **Çıktı:** Kat değişimi anlamlı biçimde bant içinde kalan genleri içeren `results/equivalent_within_margin.csv`.

Dikkat edilmesi gerekenler

İki ölçütü de karşılamayan bir gen için bu testlerle sonuç kesin değildir; belirsizlik veya sınıra yakın etki buna yol açabilir. Kaç genin geçtiğini gördükten sonra tolerans bandını genişletmeyin. Sınırı, tekrar sayısını ve kesinliği birlikte planlayın; çalışma ilgili etkileri ayıramıyorsa belirsizliği koruyun. Örneğin temel hücresel işlevlerde görevli bir geni kontrol olarak kullanırken ifadesinin anlamlı bir tolerans içinde eşdeğer kaldığını göstermek için bu testi kullanın; sıradan büyük bir p değeri bu iddiayı desteklemez.

8.5 Sayımları dönüştürme ve dışa aktarma

Varyans dengeleme dönüşümü (VST), sayımları PCA, kümeleme ve ısı haritaları için yaklaşık eş varyanslı, log benzeri bir ölçeğe taşır. Ardından sonuç tablosunu diske yazın.

R

```
vsd <- varianceStabilizingTransformation(dds, blind = FALSE) # also works with small gene
    ↪ panels

# ordered results table to disk
res_out <- as.data.frame(res)
res_out <- res_out[order(res_out$padj), ]
write.csv(res_out, "results/DE_results.csv")
save_analysis(dds, res, resLFC, assay(vsd), coldata, cfg)
saveRDS(vsd, "results/vsd.rds")
```

Uygulama notu

`vst()`, eğilimi kestirmek için genlerin alt kümesini kullanan hızlı bir sarmalayıcıdır; varsayılan alt küme çok küçük gen panellerine uygun değildir. Tam `varianceStabilizingTransformation()` bu alt küme boyutu kısıtından kaçınır. `rlog()`, özellikle ölçek faktörleri çok değiştiğinde yararlı başka bir dönüşümdür; örnek sayısı arttıkça yavaşlar. Dönüşümün tasarımı kullanması için `blind = FALSE` seçin.

8.6 Gen anotasyonu (AnnotationDbi ve biomaRt)

Sayımlar kararlı gen kimlikleriyle anahtarlanır: Ensembl `ENSG...` veya Entrez numaraları. Gen sembolleri sonuç tablolarını daha okunabilir kılabilir; zenginleştirme analizi ise seçilen veri tabanının veya gen kümesinin beklediği kimlik türünü gerektirir. Ek anotasyonlar biyotip ve uzunluk bilgilerini içerebilir. İki arayüz bu gereksinimlerin çoğunu karşılar.

OrgDb ile AnnotationDbi (çevrim dışı, hızlı). `mapIds()`, önceden hazırlanmış organizma veri tabanını sorgular. İnsan için `org.Hs.eg.db`; diğer organizmalarda uygun tür paketi kullanılır.

R

```
library(AnnotationDbi)
library(org.Hs.eg.db)

ids <- sub("\\..*$", "", rownames(res))      # strip Ensembl version suffix: ENSG...".7" ->
    ↪ ENSG...
single_target <- function(x) {
  x <- unique(x[!is.na(x)])
  if (length(x) == 1L) x else NA_character_
}
res$symbol <- mapIds(org.Hs.eg.db, keys = ids, keytype = "ENSEMBL",
                     column = "SYMBOL", multiVals = single_target)
res$entrez <- mapIds(org.Hs.eg.db, keys = ids, keytype = "ENSEMBL",
                     column = "ENTREZID", multiVals = single_target)
columns(org.Hs.eg.db)                          # all fields you can map to
```

biomaRt (çevrim içi Ensembl, kapsamlı öznitelikler). OrgDb içinde bulunmayan gen biyotipi, kromozom ve koordinatlar, GC içeriği, transkript uzunlukları ve türler arası homoloji alanları için kullanın.

R

```
library(biomaRt)
mart <- useEnsembl(biomart = "genes", dataset = "hsapiens_gene_ensembl")
ann <- getBM(
  attributes = c("ensembl_gene_id", "external_gene_name", "gene_biotype",
                 "chromosome_name", "start_position", "end_position"),
  filters    = "ensembl_gene_id",
  values     = ids,
  mart       = mart)
```

Dikkat edilmesi gerekenler

Eşlemeden önce Ensembl sürüm ekini, örneğin `ENSG00000141510.7` içindeki .7'yi çıkarın; aksi hâlde anahtarlar eşleşmeyebilir. Gen biyotipi, `protein_coding` seçimi gibi özellik sınıflandırmalarını destekler. rRNA veya mitokondri bolluğu kestirimi ayrıca uygun özellik anotasyonları ve ifade sayımları gerektirir. Bire çok ve çoğa bir eşleşmeler olabilir. `multiVals = "first"`, bir kaynağın birden çok hedefinden ilkini seçer; biyolojik belirsizliği çözmez. Yardımcı kod tüm eşleşmeleri dışa aktarır ve keyfî hedef seçmek yerine belirsiz tek-değerli anotasyonu eksik bırakır. Genel Ensembl sunucusu zaman zaman erişilemeyebilir; `useEnsembl(..., version = 110)` ile sürüm sabitleyin veya ayna sunucu kullanın. `AnnotationHub`/`ensembldb` (`EnsDb`), sürümlenmiş çevrim dışı yedek olabilir.

8.7 İlgilenilen genlerin incelenmesi

Genom çapındaki tabloya ek olarak belirteç, yolak üyesi veya aday gibi birkaç genin ifadesi incelenebilir. Farklı derinlikteki kütüphaneleri karşılaştırabilmek için *normalleştirilmiş* sayımlardan çalışın.

R

```r
#### One gene as a labelled count plot ####
goi_id <- rownames(res)[match("TP53", res$symbol)]    # symbol -> Ensembl ID (from the
    ↪ annotation step)
plotCounts(dds, gene = goi_id, intgroup = "condition") # base-R dotplot of normalised counts

#### A panel of genes: normalised counts (genes x samples) ####
goi <- c("TP53", "MDM2", "CDKN1A")
ids <- rownames(res)[match(goi, res$symbol)]
counts(dds, normalized = TRUE)[ids, ]

#### Their effect sizes and significance, from the results table ####
res[ids, c("log2FoldChange", "pvalue", "padj")]
```

Uygulama notu

Ölçek faktörüne bölünmüş normalleştirilmiş sayımlar, ham sayımlardan farklı olarak aynı genin örnekler arasındaki değerlerini karşılaştırılabilir kılar. Şekil üretmek için `plotCounts` çağrısında `returnData = TRUE` kullanıp grafiği ggplot2 ile yeniden kurun; koşula göre `geom_jitter` ve gerekirse log ölçek kullanılabilir. Anlamlılığı gruplar arasındaki görsel aralıktan değil, `padj` değerinden okuyun.

8.8 DE yöntemlerini karşılaştırma: edgeR ve limma-voom

Aynı sayım matrisi DESeq2 yerine edgeR veya limma-voom ile de analiz edilebilir; üçü yerleşik yöntemlerdir ve çoğu durumda benzer sonuçlar verir. edgeR, yarı olabilirlik testli negatif binom GLM kurar [42]; limma-voom, sayımları kesinlik ağırlığı içeren log-CPM değerlerine dönüştürüp doğrusal model kurar [43]. Yöntemi modeline, tasarım desteğine ve tanısal sonuçlarına göre seçin; örnek sayısı tek başına uygun yöntemi belirlemez. İkisi de Bölüm 8'deki aynı `cts` ve `coldata` girdilerinden başlar.

Paketin yetenekleri ve bu kodun girdi sözleşmesi farklıdır. edgeR v4, kesirli sayımları destekler ve küçük sayımlar için yarı olabilirlik kestirimini yeniler [78]. Bu özellik, herhangi bir normalleştirilmiş ifade matrisinin sayım olarak kullanılabileceği anlamına gelmez. Buradaki ortak tamsayı matrisi giriş noktası kesirli değerleri bilinçli olarak reddeder; tahmini transkript sayımları belgelenmiş `tximport` yolundan alınır. TPM değerlerini yuvarlamak veya girdi sözleşmesini sessizce değiştirmek yerine nicelendirme aracını, ofset yaklaşımını ve kullanılan modeli kaydedin.

edgeR (yarı olabilirlik F testi).

R

```
library(edgeR)
design <- model.matrix(cfg$design, data = coldata)
coefficient <- paste0(cfg$contrast[1], cfg$contrast[2])
stopifnot(coefficient %in% colnames(design), cfg$contrast[3] == cfg$reference)
y    <- DGEList(counts = cts, group = coldata$condition)    # raw integer counts
keep <- filterByExpr(y, design = design)                   # filter for the declared model
y    <- y[keep, , keep.lib.sizes = FALSE]
y    <- normLibSizes(y)                                     # TMM normalisation (older edgeR
    ↪ : calcNormFactors)

y      <- estimateDisp(y, design)                           # gene-wise + trended
    ↪ dispersions
fit    <- glmQLFit(y, design)                               # quasi-likelihood negative-
    ↪ binomial GLM
qlf    <- glmQLFTest(fit, coef = coefficient)               # declared condition effect
topTags(qlf, n = 20)                                        # ranked table, BH-adjusted FDR
```

limma-voom (yukarıda filtrelenen, TMM ile normalleştirilen `y` ve `design` nesnelerini kullanır).

R

```
library(limma)
v   <- voom(y, design, plot = TRUE)       # mean-variance trend -> per-observation precision
    ↪  weights
fit <- lmFit(v, design)                   # weighted least-squares linear model per gene
fit <- eBayes(fit, robust = TRUE)         # moderated t-statistics, robust to hypervariable
    ↪ genes
topTable(fit, coef = coefficient, number = 20)      # ranked results
```

Temel parametreler

`filterByExpr`, tasarıma göre zayıf ifade edilen genleri çıkarır; edgeR/limma için önerilen ön filtredir. `normLibSizes` (eski adı `calcNormFactors`), TMM ölçek faktörlerini hesaplar. `glmQLFit` ve `glmQLFTest`, küçük n durumlarında da kullanılan edgeR yarı olabilirlik testidir. `coefficient`, yapılandırılan pay düzeyini referansına karşı adıyla seçer; böylece eş değişken eklendiğinde test edilen sütun sessizce değişmez. `voom`, ortalama–varyans ilişkisini modelleyerek limma doğrusal modelini sayım verisine uyarlar.

Dikkat edilmesi gerekenler

Bu yöntemlere ham sayımlar veya hizalamasız yol için `tximport` uzunluk ofsetleri verin; TPM/FPKM kullanmayın. Bu örnek içi normalleştirmeler modelin dayandığı sayım bilgisini kaybettirir. Yöntemlerin uyumu tasarıma, filtrelemeye, saçılım kestirimine ve veriye bağlıdır. Yöntemler arası uyum bir duyarlılık kontrolü sağlar; deneyin amaçlanan etkiyi belirlediğinin kanıtı değildir. DESeq2 karşılaştırmasında olduğu gibi `coef`/`contrast` seçimini tasarımla tam eşleştirin.

9 İşlevsel zenginleşme

Zenginleşme analizinde kullanılan gen listesi, sorulan soruyu belirler. *Aşırı temsil analizi (ORA)*, anlamlı gen *listesinin* bir kategoride zenginleşip zenginleşmediğini sorgular. *Gen kümesi zenginleşme analizi (GSEA)* ise sert eşik uygulanmamış *sıralı* gen listesini kullanır; bir kategorinin genlerinin listenin üst veya alt uçlarında toplanıp toplanmadığını inceler. İkisi de clusterProfiler ile yürütülür [12]. GO aşırı temsil ve GSEA ile hastalık ontolojisi için organizma kapsamı denetlenmiş bir Bioconductor `OrgDb` gerekir. KEGG bir KEGG genom kodu kullanır; bunların bulunmadığı organizmalarda g:Profiler veya özel gen kümeleri kullanılabilir (Bölüm 9.4).

9.1 Gen kimliklerini eşleme ve arka plan gen kümesi tanımlama

Zenginleşme işlevleri ENTREZ kimlikleri ve uygun bir *arka plan gen kümesi* bekler: Bu karşılaştırmanın seçim kuralına uygun tüm genler. Eşlemeden önce Ensembl sürüm eklerini çıkarın.

R

```
#### IDs ####
library(clusterProfiler)
library(org.Hs.eg.db)                        # swap for your organism's OrgDb

res_df         <- as.data.frame(res)
res_df$ENSEMBL <- sub("\\..*$", "", rownames(res_df))   # ENSG000..."."7" -> ENSG000...

# map every tested gene to ENTREZ (this becomes the universe)
map_all <- bitr(res_df$ENSEMBL, fromType = "ENSEMBL",
                toType = "ENTREZID", OrgDb = org.Hs.eg.db)
write.csv(map_all, "results/enrichment_mapping_all.csv", row.names = FALSE)
one_target <- !duplicated(map_all$ENSEMBL) & !duplicated(map_all$ENSEMBL, fromLast = TRUE)
map <- map_all[one_target, , drop = FALSE]
res_map <- merge(res_df, map, by = "ENSEMBL")
universe <- unique(res_map$ENTREZID[!is.na(res_map$padj)])

# Exploratory observed-effect screen, not a threshold-null test
sig_entrez <- unique(res_map$ENTREZID[which(res_map$padj < cfg$alpha &
                                      abs(res_map$log2FoldChange) > cfg$lfc_cutoff)])
```

Dikkat edilmesi gerekenler

Test edilen genleri her zaman `universe =` ile verin; bu alanı atlamak veri tabanı arka planını kullanır ve bilimsel karşılaştırmayı değiştirebilir. `keyType` kimlik türüyle eşleşmeli, `OrgDb` ve organizma kodu doğru olmalıdır: İnsan `hsa`, fare `mmu` vb. Küçük bir `bitr` eşleme kaybı olağandır; büyük kayıpta türü, sürüm kapsamını ve ad alanı bilgisini araştırın.

Seçilen liste, Bölüm 8.3 içinde açıklanan olağan padj ve gözlenen etki sınırına dayalı keşif amaçlı süzmeyi kullanır; bu bir eşik hipotezi testi değildir ve gen düzeyinde yeni bir FDR güvencesi sağlamaz. Süzme kuralını değiştirmek ORA sorusunu değiştirir. Eşlenmiş uygun arka planı koruyun; seçilen ve uygun genlerdeki eşleme kayıplarını bildirin.

9.2 GO ve KEGG aşırı temsil analizi (ORA)

R

```r
#### GO biological process ####
ego <- enrichGO(gene          = sig_entrez,
                universe      = universe,
                OrgDb         = org.Hs.eg.db,
                keyType       = "ENTREZID",
                ont           = "BP",          # BP / MF / CC
                pAdjustMethod = "BH",
                pvalueCutoff = 0.05,
                qvalueCutoff = 0.20,
                readable      = TRUE)          # show gene SYMBOLs

#### KEGG pathways ####
ekegg <- enrichKEGG(gene         = sig_entrez,
                    organism     = "hsa",
                    universe     = universe,
                    pvalueCutoff = 0.05)
ekegg <- setReadable(ekegg, OrgDb = org.Hs.eg.db, keyType = "ENTREZID")

head(as.data.frame(ego))
```

Temel parametreler

`ont`, GO alt ontolojisini seçer: BP biyolojik süreci, MF moleküler işlevi, CC hücresel bileşeni gösterir. `pAdjustMethod = "BH"`, Benjamini–Hochberg FDR düzeltmesidir. `pvalueCutoff`/`qvalueCutoff`, anlamlılık eşikleridir. `readable = TRUE`/`setReadable()`, ENTREZ kimliklerini gen sembollerine dönüştürür.

`minGSSize`/`maxGSSize` küme boyutu sınırları, kullanılabilir örtüşmeden sonraki büyüklüğe uygulanır. Çok küçük kümelerde birkaç gen sonucu belirleyebilir; geniş kümeler farklı süreçleri bir araya getirebilir. Yardımcı koddaki en az 2 gen sınırı oyuncak GMT içindir; en çok 500 sınırı geniş kümeleri dışlar. İkisi de her analiz için doğrulanmış seçim değildir. Daha dar sınırlar hipotezleri azaltır; kalanları mutlaka iyileştirmez. `pvalueCutoff`/`qvalueCutoff`, raporlanan kategorileri süzer ve sınanmış bir terimi gizleyebilir. Boş tabloda örtüşmeyi, boyut filtrelerini ve kanıtı inceleyin; bundan etkinlik sonucu çıkarmayın. [12]

Varsayımsal ORA sayımları. Eşleme ve kaynağın anotasyon kısıtlarından sonra 1.000 uygun gen kaldığını, bunların 100'ünün keşif amaçlı süzmeyi geçtiğini varsayalım. Bir terimin 50 uygun geninden 10'u seçilmiş olsun. `GeneRatio` 10/100, `BgRatio` ise 50/1000 olur. Rastgele seçimde beklenen örtüşme beştir; gözlenen örtüşme bunun iki katıdır. Bu sayımlar tek başına düzeltilmiş p değerini vermez; test ve sınanan terimler arasında çoklu test düzeltmesi yine gereklidir. Paydalar, ilgili kaynak için gerçekten uygun olan genlerden hesaplanmalıdır.

9.3 Sıralı gen listesinde GSEA

GSEA, ENTREZ adları taşıyan *tüm* genlerin işaretli bir istatistikle büyükten küçüğe sıralanmasını ister. Burada ham işaretli $\log_2$FC kullanılır; analiz boyunca belirtilen aynı ölçütü koruyun.

R

```
#### Build the ranked list ####
unique_rank <- !duplicated(res_map$ENTREZID) & !duplicated(res_map$ENTREZID, fromLast = TRUE
    ↪ )
ranked <- res_map[unique_rank & is.finite(res_map$log2FoldChange), ]
geneList <- sort(setNames(ranked$log2FoldChange, ranked$ENTREZID), decreasing = TRUE)
# Report excluded ambiguous mappings; do not select an arbitrary duplicate row.

#### Run GSEA (reproducible) ####
set.seed(42)
gse <- gseGO(geneList     = geneList,
             OrgDb        = org.Hs.eg.db,
             ont          = "BP",
             keyType      = "ENTREZID",
             pvalueCutoff = 0.05,
             eps          = 0,              # accurate small p-values
             seed         = TRUE)
```

Dikkat edilmesi gerekenler

Sıralama ölçütü yönü korumak için *işaretli*, liste yalnızca anlamlı genlerden oluşmaması için *tam* olmalıdır. `set.seed()` ile `seed = TRUE`, permütasyon p değerlerinin yeniden üretilebilirliğini destekler. `eps = 0`, çok küçük p değerlerinin alt sınıra sabitlenmesini önler.

Ham LFC sıralaması, kestirimin kesinliğine göre ağırlıklandırmadan etki büyüklüğünü öne çıkarır. Varsayımsal olarak LFC değeri 2, standart hatası 2 olan gen; LFC değeri 1,5, standart hatası 0,25 olan genin üstünde yer alır. Oysa Wald istatistikleri sırasıyla 1 ve 6 olur. İşaretli Wald istatistiğiyle sıralama bu iki genin sırasını değiştirir ve belirsizliği dikkate alan farklı bir soru sorar. Ölçütü yorumlamadan önce belirtin; istenen yolağı elde etmek için değiştirmeyin. [69]

9.4 OrgDb olmadan zenginleşme (model olmayan organizmalar)

Yukarıdaki yollar Bioconductor `OrgDb` gerektirir; kapsam organizmaya ve anotasyon kaynağına bağlıdır. Yolu seçmeden bulunabilirliğini denetleyin. Organizmanın kaynaklarına göre üç seçenek vardır:

1. **KEGG genom kodu varsa** (organizma kataloğunu denetleyin): KEGG ORA ve GSEA, organizmanın kendi gen kimlikleriyle çalışır.
2. **g:Profiler desteği varsa** (çok sayıda Ensembl/EnsemblGenomes organizması): OrgDb olmadan web hizmeti üzerinden GO, KEGG ve Reactome kullanılır.
3. **İkisi de yoksa**: Kendi gen kümelerinizi sağlayın (Bölüm 9.5).

Önce kimlikleri hizmetin beklediği ad alanına dönüştürün. KEGG, NCBI Gene kimlikleri veya organizmaya özgü lokus etiketi kullanır; bitkilerle mantarlar için evrensel bir dönüştürme kuralı yoktur. RefSeq LOC+sayı, pirinç LOC_Os kimlikleri ve mantar lokus etiketleri farklı kurallardır. İncelenmiş bir dönüştürme tablosu kullanın, ad alanını belirtin, eşlenmeyen ve belirsiz kayıtları koruyun. [74]

R

```r
# A complete reviewed one-to-one map for this example; NA explicitly marks unmapped genes.
res_df <- as.data.frame(res)
res_df$gene_id <- rownames(res_df)
map <- read.delim("config/kegg_mapping.tsv", colClasses = "character")
ids <- map$kegg_id[match(res_df$gene_id, map$gene_id)]
sig <- res_df$gene_id[which(res_df$padj < cfg$alpha & abs(res_df$log2FoldChange) > cfg$lfc_
    ↪ cutoff)]
kegg_ids <- ids[match(sig, res_df$gene_id)]
kegg_ids <- kegg_ids[!is.na(kegg_ids)]
universe_kegg <- unique(ids[!is.na(res_df$padj) & !is.na(ids)])
keep <- !is.na(ids) & is.finite(res_df$log2FoldChange)
ranklist <- sort(setNames(res_df$log2FoldChange[keep], ids[keep]), decreasing = TRUE)
```

Kodu bulunan organizmalar için KEGG (pirinç `osa`, *Fusarium graminearum* `fgr` vb.):

R

```r
ekegg <- enrichKEGG(gene = kegg_ids, organism = "osa", keyType = "ncbi-geneid",
                    universe = universe_kegg, pvalueCutoff = 0.05)
set.seed(42)
kgse <- gseKEGG(geneList = ranklist, organism = "osa", keyType = "ncbi-geneid", # ranklist
    ↪ in same ID namespace
                eps = 0, seed = TRUE, pvalueCutoff = 0.05)
```

g:Profiler, desteklenen organizmalarda GO/KEGG/Reactome için kullanılabilir:

R

```r
library(gprofiler2)
gp <- gost(query    = sig,
           organism = "osativa",                    # g:Profiler organism id
           sources = c("GO:BP", "KEGG", "REAC"),
           custom_bg = res_df$gene_id[!is.na(res_df$padj)],             # the tested genes
    ↪ as the background
           user_threshold    = 0.05,
           correction_method = "g_SCS")             # g:Profiler's own multiple-testing
    ↪ correction
head(gp$result[, c("source", "term_name", "p_value", "intersection_size")])
```

Temel parametreler

`organism = "osa"`, KEGG kodudur. Bu örnekte `keyType = "ncbi-geneid"`, NCBI Gene kimliği kullanıldığını belirtir; `"kegg"` yalnızca doğrulanmış yerel KEGG anahtarlarında kullanılır. g:Profiler için `organism`, hizmetin kendi organizma kimliğidir; örneğin `osativa`. `sources`, veri tabanları; `custom_bg`, test edilmiş gen arka planı; `correction_method = "g_SCS"`, varsayılan çoklu test düzeltmesidir. Kimlikler tanınmıyorsa önce `gconvert()` ile dönüştürün.

Dikkat edilmesi gerekenler

Sıfır terim bildirilmesi geçerli bir negatif sonuç, küme boyutu/anlamlılık filtresi etkisi veya kimlik uyumsuzluğu olabilir. Yorumlamadan önce eşleme kapsamını ve uygun küme boyutlarını inceleyin. KEGG anahtar türleri organizmaya özgüdür; g:Profiler gönderilen ad alanını tanımalıdır. Gerektiğinde `gconvert()` sonucunu inceleyin. KEGG organizma kodunu, daha sonraki STRING aşamasında ise seçilen sürümün desteklediği taksonu doğrulayın (Bölüm 10). Başarılı eşleme zenginleşmeyi garanti etmez; negatif zenginleşme de eşlemenin başarısız olduğunu kanıtlamaz.

9.5 Kendi gen kümesi kaynağınızı kullanma

Organizma için uygun veri tabanı yoksa veya kendi küratörlü kaynağınız varsa, örneğin GMT dosyası ya da kimlik-terim tablosu, aynı clusterProfiler işlevleriyle sınayın. Yöntem organizmadan bağımsızdır; gen kümeleriyle gen listenizin aynı kimlik ad uzayını kullanması yeterlidir.

Bu bağımsız dal, kaydedilmiş sonuç tablosunu yeniden okur; sorguyu, uygun gen evrenini ve sıralamayı `custom_cfg$gene_namespace` ad alanında tanımlar. Ensembl sürüm ekleri yalnızca ENSEMBL kuralında çıkarılır. Önceki ENTREZ/KEGG nesneleri yerine analiz projesindeki `config/genesets/example_sets.gmt` kullanılır. Oyuncak GMT yerine aynı ad alanında incelenmiş bir dosya sağlayın ve kökenini kaydedin. Örtüşme yoksa kontrol testten önce kimlik uyumsuzluğunu bildirir; kısmi kapsam yine incelenmelidir.

R

```
library(clusterProfiler)
source(file.path(Sys.getenv("BULK_CODE"), "R/contracts.R"))
obj <- read_analysis_object(); custom_res <- obj$res; custom_cfg <- obj$config
ids <- native_gene_ids(rownames(custom_res), custom_cfg$gene_namespace)
sig <- ids[match(significant_ids(custom_res, custom_cfg), rownames(custom_res))]
universe_custom <- ids[!is.na(custom_res$padj)]
rank_custom <- ranked_genes(custom_res, ids)
gmt <- read.gmt("config/genesets/example_sets.gmt")   # term, native gene ID
if (!length(intersect(gmt$gene, universe_custom))) {
  stop("Gene sets and eligible genes have no shared IDs; check the namespace")
}
eora <- NULL
if (length(sig)) eora <- enricher(sig, TERM2GENE = gmt, universe = universe_custom,
    minGSSize = custom_cfg$geneset_min_size, maxGSSize = custom_cfg$geneset_max_size)
set.seed(42)
egse <- GSEA(rank_custom, TERM2GENE = gmt, minGSSize = custom_cfg$geneset_min_size,
             maxGSSize = custom_cfg$geneset_max_size, eps = 0, seed = TRUE)
```

Dikkat edilmesi gerekenler

Gen kümeleri ve gen listeniz *aynı* kimlikleri kullanmalıdır; denetlenmeyen uyumsuzluk boş sonuca yol açabilir; bu dalda uygun kimlikler hiç örtüşmüyorsa işlem durur (Bölüm 9.4). `read.gmt` iki sütun verir: term, gene. İki sütunlu `TERM2GENE` tablosunda terim ilk, gen kimliği ikinci sütunda olmalıdır. Paketteki dört genli oyuncak örnekler, gösterim için açıkça en az iki genlik küme sınırı kullanır; gerçek analizde biyolojik gerekçeli sınırlar seçin. Boş seçilmiş-gen listesi ORA sonucu üretmez; bu bir ad alanı hatası değildir. Testin tüm genomla karşılaştırılmaması için uygun `universe` tanımlayın.

Varsayımsal bir GSEA satırında `NES` = 1,8 ve düzeltilmiş $p = 0{,}02$ bulunması, küme üyelerinin treated/control sıralamasının pozitif ucunda yoğunlaştığını gösterir. NES, zenginleşme skorunu sıfır dağılımına göre normalleştirir; kat değişimi değildir. Pozitif zenginleşmede `core_enrichment`, yürüyen skorun tepe noktasına kadar karşılaşılan ve sinyali taşıyan öncü alt küme üyelerini gösterir. Negatif NES diğer uca yönelir. Bu genleri ve sıralama ölçütünü inceleyin; zenginleşme tek başına yolağın biyokimyasal olarak etkinleştiğini göstermez. [12]

9.6 Zenginleşmeyi görselleştirme

R

```
library(enrichplot)

dotplot(ego, showCategory = 15)                          # top categories: gene ratio + size
barplot(ego, showCategory = 15)                          # counts per category
ego2 <- pairwise_termsim(ego)                            # needed before emapplot
emapplot(ego2, showCategory = 25)                        # network of related terms
cnetplot(ego, showCategory = 5, foldChange = geneList) # genes <-> categories
ridgeplot(gse, showCategory = 15)                        # GSEA: score distributions
gseaplot2(gse, geneSetID = 1:3)                          # GSEA running-score curves
```

Uygulama notu

`dotplot`, tek grafikte özet sağlar: Kategoriye karşı gen oranı, gen sayısını gösteren nokta boyutu ve düzeltilmiş p değerini gösteren renk. `emapplot`, birbirini tekrar eden GO terimlerini temalar içinde toplar; `cnetplot`, kategorilere hangi genlerin katkı verdiğini gösterir.

9.7 Diğer gen kümesi kaynakları ve uzunluk yanlılığı

GO ve KEGG başlangıç kaynaklarıdır. Aynı seçilmiş-gen kümesi ve sıralı liste, yaygın başka kaynaklara da verilebilir. Bunların hepsi yukarıdaki `sig_entrez`, `universe` ve `geneList` nesnelerini kullanır.

R

```
#### MSigDB Hallmark sets + fast pre-ranked GSEA (fgsea) ####
library(msigdbr); library(fgsea)
h        <- msigdbr(species = "Homo sapiens", collection = "H")     # 50 Hallmark gene sets
pathways <- split(h$ncbi_gene, h$gs_name)                            # sets as lists of
    ↪ ENTREZ ids
fgseaRes <- fgsea(pathways = pathways, stats = geneList)             # geneList = signed,
    ↪ ranked, named

#### Reactome pathways ####
library(ReactomePA)
react <- enrichPathway(gene = sig_entrez, universe = universe,
                       organism = "human", readable = TRUE)
```

R

```
#### Any MSigDB collection through clusterProfiler (e.g. C2 canonical pathways) ####
library(clusterProfiler)
c2    <- msigdbr(species = "Homo sapiens", collection = "C2", subcollection = "CP:REACTOME")
emsig <- enricher(sig_entrez, TERM2GENE = c2[, c("gs_name", "ncbi_gene")],
                  universe = universe)
```

Temel parametreler

`msigdbr`, MSigDB içeriğini düzenli tablo olarak sağlar [44]: `H`, geniş ve gürültüsü azaltılmış süreçleri içeren Hallmark; `C2`, küratörlü Reactome/KEGG/BioCarta yolakları; `C5`, GO ontolojileridir. `fgsea`, adlandırılmış, işaretli ve sıralı istatistik üzerinde hızlı permütasyon GSEA yapar [45]. `enricher`/`GSEA`, özel `TERM2GENE` tablolarına ORA/GSEA uygular. Yeni `msigdbr` sürümlerinde ($\geq$10), `category`/`subcategory` yerine `collection`/`subcollection`, `entrez_gene` yerine `ncbi_gene` kullanılır. Eski kurulumlarda önceki adlar geçerlidir; `colnames()` ile denetleyin.

Dikkat edilmesi gerekenler

Gen uzunluğu yanlılığı. Okuma sayısı transkript uzunluğuyla ilişkili olduğundan uzun genler daha kolay anlamlılığa ulaşabilir; düz GO aşırı temsil testi uzun gen bakımından zengin kategorileri fazla bildirebilir. Önemli olduğunda, testi uzunluğa bağlı seçim olasılığıyla ağırlıklandıran `goseq` kullanın: `nullp()` → `goseq()` [47]. Sıralı yöntemler sert DE-listesi eşiğini kaldırır; ancak sıralama istatistikleri ve anotasyon kapsamı yine yanlılık yaratabilir. Bunlar genel bir uzunluk yanlılığı düzeltmesi değildir.

9.8 Örnek başına gen kümesi etkinliği (GSVA) İsteğe bağlı

Yukarıdaki testler gen kümesinin *gruplar arasında* farklılaşıp farklılaşmadığını sorar. GSVA, her *örneği* her gen kümesi için puanlar; ifade matrisini küme × örnek etkinlik matrisine dönüştürür. Bu skorlar bir eş değişkenle ilişkilendirilebilir, kümelenebilir, limma gibi sürekli skora uygun yöntemle modellenebilir veya çizilebilir [48]. GSVA betimseldir; tek başına anlamlılık testi değildir.

R

```
library(GSVA)
mat <- assay(vsd)                                # a normalised matrix (samples in columns)
gene_sets <- read.gmt("sets/my_pathways.gmt") # two columns: term, gene
gene_sets <- split(gene_sets$gene, gene_sets$term)   # -> a named list of gene-id vectors

par    <- gsvaParam(mat, gene_sets, kcdf = "Gaussian", minSize = 2, maxSize = 500)
scores <- gsva(par)                              # gene sets x samples
write.csv(scores, "results/gsva_scores.csv")

# heatmap of the most variable pathways
library(pheatmap)
v <- apply(scores, 1, var)
sel <- head(order(v, decreasing = TRUE), 40)
pheatmap(scores[sel, , drop = FALSE], scale = "row", cluster_rows = length(sel) > 1)
```

Temel parametreler

`gsvaParam(...)` ardından `gsva()`, iki adımlı GSVA API'sidir; sürüm ≥1.50 için kullanılır. Önceki sürümler doğrudan `gsva()` çağırır. Log ölçekli sürekli matris için, örneğin VST, log-CPM veya log2 yoğunlukta, `kcdf = "Gaussian"`; ham tamsayı sayımlarda `"Poisson"` seçilir. `minSize`/`maxSize`, küme boyutu sınırlarıdır.

Dikkat edilmesi gerekenler

GSVA skoru örnek başına etkinliktir; ham sayım veya diferansiyel test değildir. DESeq2 girdisi olarak kullanmayın. Grupları karşılaştırmak için skor matrisine uygun limma modeli veya t-testi uygulayın. Gen kümeleri matrisle aynı kimlik ad uzayını paylaşmalıdır; aksi hâlde kullanılabilir örtüşme kalmayabilir (Bölüm 9.4). `kcdf` seçimini veri ölçeğiyle eşleştirin.

10 Protein etkileşim ağı

STRING, seçilen genleri işlevsel ilişkileri içinde incelemeyi sağlar. Bölüm 8'deki seçilmiş gen listesini çevrim içi hizmete eşleyerek bir ağ oluşturun; bu ağda bağlı modüller ve yüksek bağlantılı düğümler incelenebilir [64]. Burada zenginleşmedekiyle aynı düzeltilmiş p değeri ve gözlenen etki sınırına dayalı keşif amaçlı süzme kullanılır; bu bir eşik hipotezi testi değildir.

R

```
library(STRINGdb); library(igraph)
sig_sym <- res_map$symbol[which(res_map$padj < cfg$alpha &
                                abs(res_map$log2FoldChange) > cfg$lfc_cutoff)]
sig_sym <- unique(sig_sym[!is.na(sig_sym)])

# Use a taxon supported by the selected STRING release (see the watchbox).
sdb    <- STRINGdb$new(version = "12.0", species = 9606, score_threshold = 400)
mapped <- sdb$map(data.frame(gene = sig_sym), "gene", removeUnmappedRows = TRUE)
edges <- sdb$get_interactions(mapped$STRING_id)
```

R

```
# build the graph, find modules and hubs
weighted <- data.frame(from = edges$from, to = edges$to, weight = edges$combined_score /
    ↪ 1000)
g <- graph_from_data_frame(weighted, directed = FALSE,
                           vertices = data.frame(name = unique(mapped$STRING_id)))
g <- simplify(g, edge_attr_combine = list(weight = "max", "ignore"))

set.seed(42)
comm <- cluster_louvain(g)                      # functional modules
V(g)$module <- membership(comm)
V(g)$degree <- degree(g)
hubs <- names(sort(V(g)$degree, decreasing = TRUE))[1:15]
```

R

```
# export for Cytoscape, then draw a figure
write_graph(g, "results/ppi/string_ppi.graphml", format = "graphml")

library(ggraph)
set.seed(42)
p <- ggraph(g, layout = "fr") +
       geom_edge_link(alpha = 0.3) +
       geom_node_point(aes(size = degree, colour = factor(module))) +
       theme_void()
ggsave("results/ppi/string_ppi.png", p, width = 7, height = 6, dpi = 300)
ggsave("results/ppi/string_ppi.svg", p, width = 7, height = 6)
```

Temel parametreler

`version = "12.0"`, STRING sürümünü sabitler. `species`, NCBI taxid değeridir. `score_threshold = 400`, orta güvenli kenar kayıtlarını korur; yüksek güven için 700 kullanılır. `combined_score / 1000`, STRING'in 0–1000 skorunu 0–1 kenar ağırlığına dönüştürür. `cluster_louvain`, modülerlik üzerinden modülleri bulur; düğümün derecesini veren `degree` değeri, merkez düğümleri sıralamakta kullanılır.

Aynı eşlenmiş başlangıç genleri için STRING eşiğini 400'den 700'e yükseltmek daha az ilişkiyi korur; dereceler, bağlı bileşenler ve modüller değişebilir, yalıtılmış başlangıç düğümleri ise korunabilir. Bu, kapsam ile kanıt güveni arasında bir seçimdir; ağın biyolojik olarak daha geçerli olduğunu kanıtlamaz. Eşiği sürümle birlikte kaydedin. [75]

Girdiler ve çıktılar

Girdi: Seçilen gen sembolleri. **Çıktı:** Cytoscape ile açılabilen `results/ppi/string_ppi.graphml` [67] ve PNG/SVG ağ şekli.

Dikkat edilmesi gerekenler

Seçilen sürümün desteklediği taksonu kullanın. STRING suşları ayrı anahtarlarla tutabilir; tür düzeyindeki kimlik 404 hatası döndürebilir. *Fusarium graminearum* PH-1 için tür kimliği 5518 yerine **229533** kullanılır. Gen sembolü olmayan, RefSeq lokus etiketli genomlarda başlangıç kimliği olarak gen kimliğini kullanın. STRING canlı bir web hizmetidir; yeniden üretilebilirlik için `version` sabitleyin ve eşlenen kenarları önbelleğe alın. Kimlik eşlemelerini aynen koruyun; tüm sembolleri genel bir kuralla büyük/küçük harfe dönüştürmeyin. Varsayılan ağ işlevsel ilişkileri gösterir ve dolaylı kanıt içerebilir. Derece, düğümleri yalnızca seçilen indüklenmiş graf içinde sıralar; nedensel önem göstermez. Kenarı bulunmayan eşlenmiş başlangıç düğümlerini koruyun, eşleme kayıplarını bildirin. [75]

11 Transkriptomik için şekiller

Aşağıdaki sayım yolu örnekleri `dds`, `res`, `resLFC` ve `vsd` nesnelerini kullanır. Yardımcı betikler ortak sonuç/ifade/üst veri çıktısını okur; örnek yapısı grafiklerinde log2 mikrodizi ifadesi, volkan grafiği için dışarıdan DE tablosu kullanılabilir. Yalnızca sonuç içeren girdi, ifade temelli şekilleri sağlayamaz. MA için uyumlu ortalama bolluk ölçüsü gerekir: DESeq2 baseMean, log eksende gösterilir; limma AveExpr ve edgeR logCPM zaten log ölçeğindedir. Her şekil sonunda PNG (raster; sunumlar için) ve SVG (vektör; düzenleme için) biçiminde dışa aktarılır. Başlığı şeklin içine koymayın; açıklama metninde verin.

11.1 PCA (örnek yapısı)

VST çıktısındaki en değişken genlere uygulanan temel bileşen analizi (PCA), örneklerin koşula göre ayrılmasını gösterir; aykırı örnekleri ve teknik parti etkilerini incelemeyi sağlar.

R

```r
#### PCA ####
library(ggplot2)
pca <- plotPCA(vsd, intgroup = "condition", ntop = 500, returnData = TRUE)
pv <- round(100 * attr(pca, "percentVar"))

p_pca <- ggplot(pca, aes(PC1, PC2, colour = condition)) +
  geom_point(size = 3, alpha = 0.9) +
  ggrepel::geom_text_repel(aes(label = name), size = 3, seed = 1, show.legend = FALSE) +
  labs(x = paste0("PC1 (", pv[1], "%)"),
       y = paste0("PC2 (", pv[2], "%)"), colour = NULL) +
  coord_fixed() +
  theme_bw(base_size = 12)

ggsave("results/pca.png", p_pca, width = 6, height = 5, dpi = 300)
ggsave("results/pca.svg", p_pca, width = 6, height = 5)
```

Temel parametreler

`ntop = 500`, en değişken 500 geni seçer. `returnData = TRUE`, grafiğin biçimini değiştirmek için veri çerçevesi döndürür. `percentVar` özniteliği, her eksenin açıkladığı varyansı verir. `coord_fixed()`, eksenlerde karşılaştırılabilir ölçeği korur.

Dikkat edilmesi gerekenler

PCA'yı ham sayımlara değil, `vst/rlog` çıktısına uygulayın. PC1 koşul yerine teknik partiye göre ayırıyorsa DE modeline tanımlanabilir bir parti terimi ekleyin. Bu, çıkarımı düzeltir ancak VST/PCA'daki parti değişkenliğini kaldırmaz. Yalnızca görüntüleme için parti etkisi kaldırılıyorsa tasarım koşul etkisini korumalıdır. [69]

`ntop = 500`, VST genleri arasında değişkenliği en yüksek olanları seçen öğretici bir ayardır. Daha küçük küme daha az sayıda baskın örüntüyü öne çıkarır; daha büyük küme ek, bazen zayıf değişkenliği de içerir ve eksenleri ya da açıklanan varyansı değiştirebilir. PCA için özellik seçimi bu tanısal görünümü değiştirir; kurulmuş DE modelini değiştirmez. Bir ayrışmayı açıklamadan önce makul özellik kümelerini ve üst veriyi karşılaştırın.

11.2 UMAP (isteğe bağlı; çok örnekli yapı)

UMAP, doğrusal olmayan gömme yöntemidir [21]. Bilgi verici kullanım için *çok sayıda* örnek gerekir; tipik küçük toplu transkriptomik tasarımlarında PCA ve kümeleme daha güvenilirdir.

R

```r
#### UMAP (uwot) ####
library(uwot)
library(matrixStats)

v   <- rowVars(assay(vsd))
sel <- head(order(v, decreasing = TRUE), 1000)    # top 1000 variable genes
mat <- t(assay(vsd)[sel, ])                   # samples x genes

set.seed(42)
um <- umap(mat, n_neighbors = 15, min_dist = 0.1, metric = "euclidean")

um_df <- data.frame(UMAP1 = um[, 1], UMAP2 = um[, 2],
                    condition = vsd$condition, sample = colnames(vsd))
ggplot(um_df, aes(UMAP1, UMAP2, colour = condition)) +
  geom_point(size = 3) + theme_bw(base_size = 12)
```

Dikkat edilmesi gerekenler

`n_neighbors` **örnek sayısından küçük olmalıdır**. Az sayıda örnekte UMAP kararsız ve yanıltıcıdır; burada PCA'yı tercih edin. UMAP stokastik olduğu için daima `set.seed()` kullanın. `min_dist`, noktaların ne kadar sıkı gruplanacağını belirler.

11.3 Örnek uzaklığı ısı haritası ve dendrogram

Örnekler arası uzaklıkların ısı haritası ve eşleşen dendrogram, seçilen uzaklık ve bağlantı yöntemi altında örnek ilişkilerini özetler.

R

```
#### Sample distances ####
library(pheatmap)
library(RColorBrewer)

sampleDists <- dist(t(assay(vsd)), method = "euclidean") # between samples
mat         <- as.matrix(sampleDists)
rownames(mat) <- colnames(mat) <- paste(vsd$condition, colnames(vsd), sep = " | ")
cols        <- colorRampPalette(rev(brewer.pal(9, "Greys")))(255)

pheatmap(mat,
         clustering_distance_rows = sampleDists,   # cluster on the SAME metric
         clustering_distance_cols = sampleDists,
         clustering_method        = "ward.D2",
         col = cols)

# stand-alone dendrogram
plot(hclust(sampleDists, method = "ward.D2"),
     labels = colnames(vsd), main = "", xlab = "", sub = "")
```

Temel parametreler

`dist(t(assay(vsd)))`, örnekler arası Öklid uzaklığı hesaplar; örneklerin satır olması için transpoz alınır. `clustering_distance_* = sampleDists`, ısı haritasıyla dendrogramın tutarlı olması için *önceden hesaplanmış* uzaklığı verir. `clustering_method = "ward.D2"`, yaygın kullanılan kompakt bir bağlantı yöntemidir.

11.4 MA grafiği: Etki büyüklüğü ve ifade

MA grafiği, ortalama ifadeye karşı $\log_2$FC gösterir. Büzülme, düşük sayımlı genlerin grafik uçlarına aşırı hâkim olmasını azaltır.

R

```
#### MA plot ####
ma_display <- res                              # original inference and missingness
ma_display$log2FoldChange <- resLFC$log2FoldChange[match(rownames(res), rownames(resLFC))]
plotMA(ma_display, ylim = c(-4, 4), alpha = cfg$alpha,
       colSig = "black", colNonSig = "grey70")     # base-R version
abline(h = c(-1, 1), col = "grey50", lty = 2)

# ggplot version
ma <- as.data.frame(ma_display)
ma <- ma[!is.na(ma$padj), ]
ma$sig <- ma$padj < cfg$alpha
ggplot(ma, aes(baseMean, log2FoldChange, colour = sig)) +
  geom_point(size = 1, alpha = 0.6) +
  geom_hline(yintercept = 0, colour = "grey40") +
  scale_x_log10() +
  scale_colour_manual(values = c("FALSE" = "grey75", "TRUE" = "black"),
                      name = paste("padj <", cfg$alpha)) +
  labs(x = "mean of normalised counts", y = "log2 fold change") +
  theme_bw(base_size = 12)
```

Dikkat edilmesi gerekenler

Görüntüleme tablosunu `res` nesnesinden kurun; düzeltilmiş p değerlerini ve NA durumunu koruyup yalnızca eşleşen büzülmüş LFC değerini gen kimliğiyle aktarın. Böylece tanımlanan test değişmeden düşük sayımlı genlerin uç görüntüleme etkileri azalır. x ekseni ortalama normalleştirilmiş sayımları log ölçekte gösterir.

11.5 Volkan grafiği: Anlamlılık ve etki büyüklüğü

Volkan grafiği etki büyüklüğünü ve istatistiksel kanıtı birlikte gösterir: x ekseninde $\log_2$FC, y ekseninde $-\log_{10}$ düzeltilmiş p ve seçilen genlerin etiketleri. Notun en ayrıntılı çizim örneğidir; `ggrepel` etiketleri okunabilir tutar, `ggtext` alt simge ve italik biçimlendirmesini sağlar.

R

```r
#### Volcano plot (ggplot2 + ggrepel + ggtext) ####
library(ggplot2)
library(ggrepel)
library(ggtext)

res_df              <- as.data.frame(res)              # original inference table
res_df$raw_lfc      <- res_df$log2FoldChange
res_df$log2FoldChange <- resLFC$log2FoldChange[match(rownames(res_df), rownames(resLFC))]
res_df$gene         <- rownames(res_df)
res_df              <- res_df[!is.na(res_df$padj), ]   # drop NA padj (filtered)
res_df$neglog10padj <- -log10(pmax(res_df$padj, .Machine$double.xmin))

# exploratory screen using original adjusted p-values and raw effects
res_df$direction <- "Other"
res_df$direction[res_df$padj < cfg$alpha & res_df$raw_lfc > cfg$lfc_cutoff] <- "Up"
res_df$direction[res_df$padj < cfg$alpha & res_df$raw_lfc < -cfg$lfc_cutoff] <- "Down"

# label the 15 lowest-padj genes passing the exploratory screen
lab <- res_df[res_df$direction != "Other", ]
lab <- head(lab[order(lab$padj), , drop = FALSE], 15)

# greyscale-friendly: encode direction by BOTH colour and shape
pal <- c(Up = "black", Down = "grey55", "Other" = "grey80")
shp <- c(Up = 17,      Down = 16,       "Other" = 16)

p_volcano <- ggplot(res_df, aes(log2FoldChange, neglog10padj)) +
  geom_hline(yintercept = -log10(cfg$alpha), linetype = "dashed",
             colour = "grey60", linewidth = 0.3) +
  geom_point(aes(colour = direction, shape = direction),
             size = 1.6, alpha = 0.8) +
  geom_text_repel(data = lab, aes(label = gene),
                  size = 3, fontface = "italic",
                  box.padding = 0.4, point.padding = 0.2,
                  min.segment.length = 0, max.overlaps = Inf,
                  segment.size = 0.3, segment.colour = "grey50",
                  seed = 42) +
  scale_colour_manual(values = pal, name = NULL) +
  scale_shape_manual(values = shp, name = NULL) +
  scale_x_continuous(breaks = scales::pretty_breaks(6)) +
  labs(x = "shrunken log<sub>2</sub> fold change",
       y = "-log<sub>10</sub> adjusted *p*") +
  theme_classic(base_size = 12) +
  theme(axis.title.x = element_markdown(),
        axis.title.y = element_markdown(),
        legend.position = "top")

ggsave("results/volcano.png", p_volcano, width = 6, height = 5, dpi = 300)
ggsave("results/volcano.svg", p_volcano, width = 6, height = 5)
```

Temel parametreler

ggrepel: `box.padding`/`point.padding`, etiket ve nokta çevresindeki boşluğu belirler. `min.segment.length = 0`, bağlantı çizgisini daima çizer; `max.overlaps = Inf`, istenen tüm etiketleri korur. `seed`, yeniden üretilebilir etiket yerleşimini sabitler. **ggtext:** `element_markdown()`, eksen başlıklarında `<sub>` alt simgelerini ve `*...*` italiklerini işler. **scales:** `pretty_breaks(6)`, uygun x işaretleri seçer. **Gösterim:** Yön hem renk hem şekille kodlanır; böylece gri tonlu baskıda da ayırt edilir.

Dikkat edilmesi gerekenler

Up/Down etiketleri, özgün `padj` ve ham LFC üzerinde keşif amaçlı ortak süzmeyi gösterir; bu bir eşik hipotezi testi veya ek FDR güvencesi değildir. Other, süzme kuralının geçilmediği anlamına gelir; anlamlı ama küçük etkileri de içerebilir. Büzülme yalnızca grafikteki konumları değiştirdiğinden ham etki sınırları burada dikey sınır oluşturmaz. Grafiğe hangi genlerin gireceğini belirlerken özgün p değerlerini ve eksikliği koruyun. Etiket sayısını sınırlayın; burada 15 kullanılır. `EnhancedVolcano` için de p değerlerini ayrı üretilmiş bir büzülme tablosundan almak yerine çıkarımı koruyan aynı görüntüleme tablosunu verin. [18]

11.6 İfadesi farklılaşan öne çıkan genlerin ısı haritası

Gen başına z skoru hesaplanarak oluşturulan ısı haritası, örnekler arasındaki ifade *örüntülerini* gösterir ve benzer olanları gruplar.

R

```r
#### Top-DE-gene heatmap (z-scored VST) ####
library(pheatmap)

ids <- rownames(res)[which(res$padj < cfg$alpha)]
ids <- head(ids[order(res[ids, "padj"])], 30)
mat <- assay(vsd)[ids, , drop = FALSE]
mat <- mat[apply(mat, 1, var) > 0, , drop = FALSE]
stopifnot(nrow(mat) > 0)
mat <- t(scale(t(mat)))                       # z-score per gene (row)
ann <- as.data.frame(colData(dds)[, "condition", drop = FALSE])

pheatmap(mat,
         scale            = "none",           # already z-scored
         annotation_col   = ann,
         show_rownames    = TRUE,
         clustering_method = "ward.D2", cluster_rows = nrow(mat) > 1,
         color = colorRampPalette(c("#2C7BB6", "white", "#C0392B"))(255),
         border_color = NA, fontsize_row = 7)
```

Uygulama notu

Birden fazla anotasyon şeridi veya bölünmüş satır/sütun için `ComplexHeatmap` kullanın [20]: `Heatmap(mat, name = "z", col = circlize::colorRamp2(c(-2,0,2), c("#2C7BB6","white","#C0392B")), top_annotation = HeatmapAnnotation(condition = vsd$condition))`.

Dikkat edilmesi gerekenler

Farklı ifade ölçeklerindeki genleri karşılaştırmak için z skorunu her gen için örnekler arasında hesaplayın. Veri zaten ölçeklendiyse çift ölçeklemeyi önlemek üzere `scale = "none"` kullanın. Gen kümesini okunabilir boyutta tutun; burada 30 gen vardır. Sıfır merkezli iki yönlü renk paleti kullanın.

Satır z skoru, her genin dönüştürülmüş değerlerinden kendi ortalamasını çıkarıp kendi örneklem standart sapmasına böler. Varsayımsal VST değerleri 8, 10 ve 12 için ortalama 10, örneklem standart sapması 2, z skorları ise $-1, 0, 1$ olur. Dolayısıyla 1 değeri, o genin ortalamasının bir standart sapma üstünü gösterir; iki kat değişim veya başka bir genden daha yüksek ifade anlamına gelmez. En üstteki genler aynı koşul karşılaştırmasıyla seçildiği için ısı haritası betimseldir; grupların ayrışması bağımsız doğrulama sayılmaz. Kod yalnızca olağan düzeltilmiş p değerine göre seçer; ek gözlenen etki süzmesini uygulamaz.

Odaklanmış ısı haritası (ilgilenilen genler). Öne çıkan DE genleri yerine bir yolak, belirteç paneli veya aday listesi gibi küratörlü bir kümeyi incelemek için `top` seçimini kendi kimliklerinizle değiştirin ve sıralarını koruyun.

R

```
#### Heatmap of a chosen gene panel ####
goi <- c("CD8A", "GZMB", "IFNG", "FOXP3", "IL2RA")          # genes of interest (symbols)
ids <- rownames(res)[match(goi, res$symbol)]                # res$symbol from the annotation
    ↪ step
ids <- ids[!is.na(ids)]
m   <- scaled_rows(assay(vsd)[ids, , drop = FALSE])                      # z-score per gene
    ↪ (row)
rownames(m) <- res$symbol[match(rownames(m), rownames(res))] # label rows by symbol
pheatmap(m, scale = "none", annotation_col = ann,
         cluster_rows = FALSE,                              # preserve the panel's order
         color = colorRampPalette(c("#2C7BB6", "white", "#C0392B"))(255),
         border_color = NA)
```

12 Ek bilgiler

12.1 Notun ötesi: *De novo* birleştirme ve varyant çağırma

Sık karşılaşılan iki komşu görev, burada izlenen referansa dayalı gen düzeyi kapsamının dışındadır.

Referans genom yoksa. Transkriptleri *de novo* birleştirin. Trinity, RNA-seq için yerleşik araçtır; rnaSPAdes yaygın bir alternatiftir [52]. MEGAHIT hızlıdır ancak RNA-seq'e özgü değil, metagenom/genom birleştirme aracıdır [53]. Tipik sıra: Birleştirme, TransDecoder ile ORF kestirimi, birleştirilmiş transkriptlere karşı Salmon nicelendirmesi ve örneğin Trinotate ile anotasyon.

Komut satırı

```
# Trinity (paired-end)
Trinity --seqType fq --left trimmed/R1.fq.gz --right trimmed/R2.fq.gz \
  --max_memory 20G --CPU 8 --output trinity_out
```

RNA-seq üzerinden varyantlar. BAM içinden SNP/indel çağrılabilir; ancak yalın `mpileup` yolu RNA-seq için uygun değildir. Aşağıdaki DNA tipi sözdizimi komutların gösterimidir; doğrulanmış RNA-seq varyant iş akışı değildir:

Komut satırı

```
bcftools mpileup -f ref/genome.fna mapped/SRR1234567.sorted.bam \
  | bcftools call -mv -Ob -o calls.bcf
bcftools view calls.bcf | less
```

Dikkat edilmesi gerekenler

Güvenilir RNA-seq varyantları için GATK iyi uygulamalarını izleyin: STAR 2-pass, duplikat işaretleme, ardından `HaplotypeCaller` öncesinde `SplitNCigarReads` [54]. Son araç, intronların oluşturduğu N CIGAR boşluklarında okumaları böler ve çağırıcının tutabilmesi için STAR MAPQ 255 değerini 60'a dönüştürür. Ekzon birleşim noktası, alele özgü ifade ve RNA düzenleme (A-to-I, okumada A → G), yalın DNA tipi çağırıcıyı karıştırabilir.

12.2 Genişletilmiş QC ve yardımcı araçlar

Ek komut satırı araçları, iş akışları ve değerlendirmelerde kullanılan kalite kontrollerini sağlar. Seçilen her aracın platforma özgü kurulumunu ve MultiQC modül desteğini denetleyin; her çıktı otomatik ayrıştırılmaz.

Araç	**Sağladığı bilgi**	**Örnek çağrı**
SAMtools [8]	Eşleme oranı (`flagstat`); mitokondri oranı dâhil kromozom başına sayımlar (`idxstats`); özet istatistik (`stats`); duplikat işaretleme (`markdup`).	`samtools idxstats`
Picard [36]	`MarkDuplicates`: PCR/optik duplikatlar, özellikle düşük girdili kütüphaneler. `CollectRnaSeqMetrics`: Kodlayan/UTR/intron/genler arası yüzdeler ve 5'/3' yanlılığı.	`picard CollectRnaSeqMetrics`
RSeQC [34]	İplik yönlülüğü (`infer_experiment.py`); ekzon/intron/UTR okuma dağılımı ve RNA bozulmasını gösterebilen 5'→3' gen gövdesi kapsamı.	`read_distribution.py`
Qualimap [35]	Tek HTML RNA-seq kalite kontrolü: Okumaların genomik kökeni, kapsam yanlılığı, transkript doygunluğu.	`qualimap rnaseq`
preseq [37]	Kütüphane karmaşıklığı/doygunluk eğrisi: Daha derin dizileme daha çok farklı molekül sağlar mı?	`preseq lc_extrap`
deepTools [38]	Genom tarayıcısı için normalleştirilmiş bigWig kapsam izleri (`bamCoverage`); örnekler arası korelasyon (`multiBamSummary` ve `plotCorrelation`).	`bamCoverage`
gffread [39]	GTF/GFF dönüşümü; Salmon veya kallisto için transkript FASTA çıkarma.	`gffread -w tx.fa`

Uygulama notu

Gen gövdesi kapsamı (RSeQC `geneBody_coverage.py`) ve `CollectRnaSeqMetrics`, bozulmuş RNA'yı incelemek için hızlı kontrollerdir. Kısmen bozulmuş poly-A kütüphanelerinde ters transkripsiyon poly-A kuyruğundan başladığı için 3' kapsam yanlılığı görülebilir.

Dikkat edilmesi gerekenler

DNA-seq'ten farklı olarak standart RNA-seq diferansiyel ifade analizinde okumalar otomatik tekilleştirilmez. Yüksek ifade edilen gen gerçekten çok sayıda aynı parça üretebilir; duplikat oranı hem gerçek ifadeyi hem teknik çoğaltmayı yansıtabilir. İfadeye bağımlılığı dupRadar ile inceleyin. UMI kütüphaneleri molekül farkındalıklı uygun bir tekilleştirme yöntemi gerektirir. Çalışmaya özgü gerekçe olmadan UMI taşımayan duplikatları otomatik kaldırmayın.

12.3 Döngüler ve toplu işleme

Gövde tek örnek üzerinden ilerler. Uygulamada `basename` ile örnek adını türetip döngü kurun; taşınabilirlik için *iş parçacığı, referans ve yol* değerlerini betiğin başında değişkenlerde tutun.

Komut satırı

```
#### Parameters (edit once) ####
THREADS=8
REF=ref/genome.fna
GTF=ref/genome.gtf

#### fastp over every paired sample ####
for r1 in raw_data/*_1.fastq.gz; do
  s=$(basename "$r1" _1.fastq.gz)          # sample name from the R1 file
  echo ">> trimming $s"
  fastp \
     -i raw_data/${s}_1.fastq.gz -I raw_data/${s}_2.fastq.gz \
     -o trimmed/${s}_1.trim.fq.gz -O trimmed/${s}_2.trim.fq.gz \
     --detect_adapter_for_pe -q 15 -l 36 \
     -h qc/post_trim/${s}.html -j qc/post_trim/${s}.json \
     --thread "$THREADS"
done
```

Komut satırı

```
# Same idea with GNU parallel (run 2 samples at a time, 8 threads each)
ls raw_data/*_1.fastq.gz | sed 's#_1.fastq.gz##' | \
  parallel -j 2 'fastp -i {}_1.fastq.gz -I {}_2.fastq.gz \
    -o trimmed/$(basename {})_1.trim.fq.gz -O trimmed/$(basename {})_2.trim.fq.gz \
    --detect_adapter_for_pe -q 15 -l 36 --thread 8'
```

Dikkat edilmesi gerekenler

CPU kapasitesini aşmayın: *paralel iş sayısı* × *iş başına iş parçacığı*, çekirdek sayısını geçmemelidir. STAR yüksek bellek istediğinden birden fazla genom indeksini bellekte tutacak RAM yoksa aynı anda *tek* STAR işi çalıştırın.

12.4 İş parçacığı, bellek ve kritik parametreler: Hızlı başvuru

Araç	Temel seçenekler	Not
fasterq-dump	`-e` <threads>, `-t` <tmp>	geçici alan yaklaşık ∼10× .sra boyutu
FastQC	`-t` <n>	dosya başına bir iş parçacığı
fastp	`--thread` (≤64; v1.0.1), `-q` `-l`, `-g`	NextSeq/NovaSeq poly-G için `-g`
SortMeRNA	`--threads`, `--out2`, `--paired_in/out`	örnek başına ayrı çalışma dizini
STAR index	`--runThreadN`, `--sjdbOverhang`, `--genomeSAindexNbases`	overhang = readlen−1; küçük genomda SA indeksini azaltın; insan için ∼30 GB RAM
STAR align	`--runThreadN`, `--readFilesCommand` zcat, `--outSAMtype` BAM SortedByCoordinate	ön ekin sonunda yol ayırıcısı bulunmalı
featureCounts	`-T`, `-p` `--countReadPairs`, `-s` 0/1/2	`-s` ayarını protokole göre seçin; çoğunlukla 2

12.5 Yeniden üretilebilirlik

Başka bir araştırmacının ortamı yeniden kurup sonuçların uyumunu değerlendirebilmesi için kullanılan sürümler ve oturum bilgileri kaydedilmelidir. Sürümleri sabitleyin, oturumları kaydedin ve her stokastik aşamada rastgelelik başlangıç değeri belirtin. Bu değer tek başına sürümler ve platformlar arasında sayısal özdeşlik garanti etmez.

Komut satırı

```
# Capture the exact tool versions of the conda environment
conda env export -n rnaseq > environment.lock.yml      # recreate with: conda env create -f
    ↪ environment.lock.yml

# A lighter, human-readable record of the key tools
conda list -n rnaseq | grep -E "star|fastp|subread|samtools|sortmerna|sra-tools|multiqc"
```

R

```
#### Record the R environment + fix seeds ####
set.seed(42)              # before UMAP / GSEA / anything random
writeLines(capture.output(sessionInfo()), "results/sessionInfo.txt")
# For full project-level R reproducibility, consider renv::init() / renv::snapshot()
```

Uygulama notu

`environment.yml`, `sessionInfo.txt`, betikler ve kısa README dosyasını sürüm kontrolünde (`git`) tutun. "Latest" seçimine dayanmak yerine araç sürümlerini açıkça belirtin; hizalayıcı veya anotasyon veri tabanı sürümleri sonuçları değiştirebilir. Genom/GTF derlemesinin tam kaynağını, sürümünü ve tarihini kodun yanında kaydedin.

12.6 Sorun giderme

Belirti	Olası neden	Çözüm
STAR: `genomeSAindexNbases` fazla büyük	Küçük genomda varsayılan 14 kullanılmış	`--genomeSAindexNbases` değerini genom uzunluğundan hesaplayın
featureCounts `Assigned` % çok düşük	Yanlış yönlülük veya anotasyon uyumsuzluğu	Protokolü ve ileri/ters kanıtı denetleyin; 0/1/2 arasında en yükseği seçmeyin
SortMeRNA mevcut durum bildiriyor	Çalışma dizini yeniden kullanılmış	Önceki durumu inceleyin; örnek için yeni dizin seçin
fasterq-dump disk alanını tüketiyor	Geçici alan yetersiz	`-t` ile .sra boyutunun ∼10× katı boş alanı olan disk seçin
STAR'da çok okuma eşlenmiyor	Genom/GTF uyumsuz veya adaptör kalmış	Derlemelerin eşleşmesini ve fastp/MultiQC raporlarını denetleyin
DESeq2: *every gene has at least one zero*	Seyrek veride ölçek faktörleri hesaplanamıyor	Seyrekliği ve tasarımı inceleyin; `type="poscounts"` açık bir alternatif normalleştirme seçimidir
`bitr` genlerin çoğunu kaybediyor	Yanlış `keyType` veya kimlik biçimi	Ensembl sürüm eklerini çıkarın; kimlik türünü ve `OrgDb` seçimini denetleyin

12.7 Örnek korelasyonları ve duyarlılık kontrolleri

Pearson korelasyonu örnek ifade profillerinin doğrusal benzerliğini, Spearman korelasyonu ise sıralarının benzerliğini karşılaştırır. PCA'da kullanılan sonlu dönüştürülmüş ifade matrisini kullanın; örnek etiketlerini ve bilinen ortak değişkenleri inceleyin. Eşlik eden `07_figures/05_sample_correlations.R` betiği, gen-satır ve örnek-sütun düzenindeki TSV dosyasından iki matrisi de üretir. Yüksek korelasyon, güçlü tedavi etkisiyle veya örneklerin karışmasıyla birlikte görülebilir; tek başına kimliği ya da modelin geçerliliğini doğrulamaz. Bu işlem örnek düzeyinde kalite değerlendirmesidir; WGCNA gibi bir gen eş ifade ağı analizi değildir.

Dönüştürülmüş ifadeye uygulanan eşleştirilmemiş Wilcoxon sıra toplamı testi, iki bağımsız grup için sınırlı bir keşifsel duyarlılık kontrolü sağlayabilir. Parti düzeltmesini, eşleştirmeyi veya sayımların ortalama-varyans modelini korumaz. `stats::wilcox.test` işlevini yalnızca örnekleme varsayımları soruya uygunsa kullanın; eşit sıraların nasıl ele alındığını belirtin ve test edilen tüm genler için çoklu test düzeltmesi uygulayın. DESeq2 ile uyuşmazlık, yöntemler arasında çoğunluk oylaması yapmayı değil tasarımın ve verilerin incelenmesini gerektirir. Burada birincil modelin yerine Wilcoxon aşaması konulmaz.

12.8 Çalışmalar arası sentez

`~ dataset + condition` gibi ortak bir model, eklemeli çalışma terimini hesaba katarak ortak koşul karşılaştırmasını kestirir. Tanımlanabilir tasarım ve karşılaştırılabilir ölçüm/anotasyon gerektirir. Çalışmayla tamamen karışan bir koşulun etkisini ayıramaz. Çalışma kökeni alanı, dizileme hatlarını veya yinelenen veri gönderimlerini değil bağımsız çalışmaları göstermelidir. Açık veri kümelerini birleştirmeden önce ortak donörleri ve kontrolleri inceleyin; aynı örneğin yeniden kullanılması çalışmaların bağımsızlığı varsayımını bozar.

Meta-analizde her çalışma ayrı modellenir ve uyumlu istatistikler birleştirilir. `HTSFilter`, tekrarlı sayım verileri için süzme yöntemi sağlar; `metaRNASeq`, ters normal yöntem dâhil p değeri birleştirme yöntemlerini uygular [79]. Kanıtların birleştirilmesi ortak etki büyüklüğünü kestirmez ve bulgunun her çalışmada tekrarlandığını kanıtlamaz. Aynı yönlü etkiler, her çalışmanın anlamlı olduğunu göstermez. Yöne göre seçilen bir test ailesi gerekçelendirilmiş yönlü çıkarım gerektirir; yalnızca aynı işaretleri seçmenin olağan BH kontrolünü sağladığını varsaymayın.

Etki sentezinde `metafor::rma.uni`, büzülmemiş log2 etkileri ve standart hatalarını birleştirir. Ortak etki (`FE`) ve rastgele etkiler (örneğin `REML` veya `DL`) modelleri farklı varsayımlara dayanır; rastgele etkiler uyumsuz biyolojiyi veya çalışmalar arası bağımlılığı gidermez. Az sayıda çalışmada heterojenliği ve belirsizliği kestirmek güçtür. Hedef modeli anlamlılık sonuçlarına bakmadan seçin; çalışma düzeyindeki etkileri ve aralıkları inceleyin.

`08_cross_study/01_pool_effects.R`, ham okumalardan değil gözden geçirilmiş çalışma tablolarından başlar. TSV dosyası `gene_id`, `study`, `numerator`, `denominator`, `log2FoldChange` ve `lfcSE` alanlarını içerir. Çalışmalarda organizma, gen kimliği uzayı, karşılaştırma yönü ve log2 ölçeği aynı olmalıdır; standart hata büzülmemiş kestirime ait olmalıdır. Anlamlı gen listelerinin birleşimi yerine gözden geçirilmiş ortak test gen kümesinin tamamını sağlayın. Betik eksik çalışma satırlarını, yinelenen kayıtları, uyumsuz yönleri ve pozitif olmayan standart hataları reddeder. İki yönlü birleşik testlerin BH düzeltmesi sağlanan tüm genleri kapsar; işaret uyumu sütunu betimseldir. FE, temel R ile çalışır; REML/DL için `metafor` gerekir ve bu öğretim yolu en az üç çalışma ister.

FE/REML/DL, yerel macOS aarch64 üzerinde R 4.6.1 ve `metafor` 5.2-1 ile bağımsız sentetik kontrolleri geçti; gerçek veri veya tüm yazılım yığını doğrulanmadı.

12.9 Anotasyon aktarımı ve elde edilemeyen sonuçlar

Küratörlü OrgDb bulunmayan organizmalarda dışarıdan alınan GO/KEGG veya STRING ortoloji aktarımı anotasyonları `TERM2GENE` ya da GMT biçiminde özel zenginleştirme yoluna verilebilir. Taksonu, sürümü, kaynak kanıtını, ortoloji ilişkisini ve tüm eşleme kayıplarını saklayın. Aktarılan üyelik, hedef türde doğrudan deneysel anotasyon değil eşlemeye dayalı bir hipotezdir. `msigdbr` aracılığıyla alınan MSigDB kümelerinde de tür ve koleksiyon belirtilmelidir; türler arası izdüşüm yorumu değiştirir. Test edilen evren ve çıkarım modeli bir p değerini desteklemiyorsa küme örtüşmesi betimseldir.

Başarıyla hesaplanıp eşik sonrasında terim bırakmayan zenginleştirmeyi; eksik eşleme, desteklenmeyen tür, ağ hizmeti hatası veya başarısız model nedeniyle elde edilemeyen sonuçtan ayırın. Boş tablo veya yer tutucu şekil bu durumları ayıramaz. Tam test tablosu, etkin anotasyonlu arka plan ve eşleme kaydının yanında yürütme durumunu saklayın. Elde edilemeyen analizi bu şekilde bildirin; başarısızlığı yolak katılımının yokluğu olarak yorumlamayın.

Genel karşılaştırmalarda büzülme. `apeglm` seçilen model katsayısını büzer; `ashr`, DESeq2'nin `lfcShrink` işlevi üzerinden desteklenen genel karşılaştırmalarda kullanılabilen alternatif bir büzülme yöntemidir. Eşlik eden kod, tanımlanmış katsayısı için `apeglm` kullanır; karmaşık karşılaştırma yalnızca görünen etiketi değiştirmeyi değil açık uygulama ve incelemeyi gerektirir. Büzülme, birincil hipotez testinden ayrı tutulur [69].

Kaynaklar

[1] Leinonen R, Sugawara H, Shumway M; INSDC. The Sequence Read Archive. *Nucleic Acids Res.* 2011;39(Database issue):D19–D21. https://doi.org/10.1093/nar/gkq1019

[2] Andrews S. *FastQC: A Quality Control Tool for High Throughput Sequence Data.* Babraham Bioinformatics; 2010. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

[3] Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics.* 2016;32(19):3047–3048. https://doi.org/10.1093/bioinformatics/btw354

[4] Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics.* 2018;34(17):i884–i890. https://doi.org/10.1093/bioinformatics/bty560

[5] Kopylova E, Noé L, Touzet H. SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics.* 2012;28(24):3211–3217. https://doi.org/10.1093/bioinformatics/bts611

[6] Bushnell B. *BBMap: A Fast, Accurate, Splice-Aware Aligner.* Lawrence Berkeley National Laboratory; 2014. LBNL-7065E.

[7] Dobin A, Davis CA, Schlesinger F, et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics.* 2013;29(1):15–21. https://doi.org/10.1093/bioinformatics/bts635

[8] Danecek P, Bonfield JK, Liddle J, et al. Twelve years of SAMtools and BCFtools. *GigaScience.* 2021;10(2):giab008. https://doi.org/10.1093/gigascience/giab008

[9] Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics.* 2014;30(7):923–930. https://doi.org/10.1093/bioinformatics/btt656

[10] Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 2014;15(12):550. https://doi.org/10.1186/s13059-014-0550-8

[11] Zhu A, Ibrahim JG, Love MI. Heavy-tailed prior distributions for sequence count data: removing the noise and preserving large differences. *Bioinformatics.* 2019;35(12):2084–2092. https://doi.org/10.1093/bioinformatics/bty895

[12] Wu T, Hu E, Xu S, et al. clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. *The Innovation.* 2021;2(3):100141. https://doi.org/10.1016/j.xinn.2021.100141

[13] Yu G. *enrichplot: Visualization of Functional Enrichment Result.* R/Bioconductor package. https://doi.org/10.18129/B9.bioc.enrichplot

[14] Carlson M. *org.Hs.eg.db: Genome wide annotation for Human.* R/Bioconductor package. https://doi.org/10.18129/B9.bioc.org.Hs.eg.db

[15] Wickham H. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York; 2016. https://doi.org/10.1007/978-3-319-24277-4

[16] Slowikowski K. *ggrepel: Automatically Position Non-Overlapping Text Labels with ggplot2.* R package. https://cran.r-project.org/package=ggrepel

[17] Wilke CO, Wiernik BM. *ggtext: Improved Text Rendering Support for ggplot2.* R package. https://cran.r-project.org/package=ggtext

[18] Blighe K, Rana S, Lewis M. *EnhancedVolcano: Publication-ready volcano plots with enhanced colouring and labeling.* R/Bioconductor package. https://doi.org/10.18129/B9.bioc.EnhancedVolcano

[19] Kolde R. *pheatmap: Pretty Heatmaps.* R package. https://doi.org/10.32614/CRAN.package.pheatmap

[20] Gu Z, Eils R, Schlesner M. Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics.* 2016;32(18):2847–2849. https://doi.org/10.1093/bioinformatics/btw313

[21] McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426; 2018. (R: uwot, Melville J.)

[22] Soneson C, Love MI, Robinson MD. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research.* 2015;4:1521. https://doi.org/10.12688/f1000research.7563.2

[23] Grüning B, Dale R, Sjödin A, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat Methods.* 2018;15(7):475–476. https://doi.org/10.1038/s41592-018-0046-7

[24] Conesa A, Madrigal P, Tarazona S, et al. A survey of best practices for RNA-seq data analysis. *Genome Biol.* 2016;17:13. https://doi.org/10.1186/s13059-016-0881-8

[25] Stark R, Grzelak M, Hadfield J. RNA sequencing: the teenage years. *Nat Rev Genet.* 2019;20(11):631–656. https://doi.org/10.1038/s41576-019-0150-2

[26] Ewels PA, Peltzer A, Fillinger S, et al. The nf-core framework for community-curated bioinformatics pipelines. *Nat Biotechnol.* 2020;38(3):276–278. https://doi.org/10.1038/s41587-020-0439-x

[27] ENCODE Project Consortium. *ENCODE Guidelines and Best Practices for RNA-Seq.* ENCODE Data Coordinating Center. https://www.encodeproject.org/data-standards/rna-seq/long-rnas/

[28] Kans J. *Entrez Direct: E-utilities on the Unix Command Line.* In: Entrez Programming Utilities Help. Bethesda (MD): NCBI; 2013 (updated regularly). https://www.ncbi.nlm.nih.gov/books/NBK179288/

[29] Burgin J, Ahamed A, Cummins C, et al. The European Nucleotide Archive in 2022. *Nucleic Acids Res.* 2023;51(D1):D121–D125. https://doi.org/10.1093/nar/gkac1051

[30] Davis S, Meltzer PS. GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. *Bioinformatics.* 2007;23(14):1846–1847. https://doi.org/10.1093/bioinformatics/btm254

[31] Choudhary S. pysradb: A Python package to query next-generation sequencing metadata and data from NCBI Sequence Read Archive. *F1000Research.* 2019;8:532. https://doi.org/10.12688/f1000research.18676.1

[32] Gálvez-Merchán Á, Min KHJ, Pachter L, Booeshaghi AS. Metadata retrieval from sequence databases with ffq. *Bioinformatics.* 2023;39(1):btac667. https://doi.org/10.1093/bioinformatics/btac667

[33] Durinck S, Spellman PT, Birney E, Huber W. Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. *Nat Protoc.* 2009;4(8):1184–1191. https://doi.org/10.1038/nprot.2009.97

[34] Wang L, Wang S, Li W. RSeQC: quality control of RNA-seq experiments. *Bioinformatics.* 2012;28(16):2184–2185. https://doi.org/10.1093/bioinformatics/bts356

[35] Okonechnikov K, Conesa A, García-Alcalde F. Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics.* 2016;32(2):292–294. https://doi.org/10.1093/bioinformatics/btv566

[36] Broad Institute. *Picard Toolkit.* GitHub repository; 2019. https://broadinstitute.github.io/picard/

[37] Daley T, Smith AD. Predicting the molecular complexity of sequencing libraries. *Nat Methods.* 2013;10(4):325–327. https://doi.org/10.1038/nmeth.2375

[38] Ramírez F, Ryan DP, Grüning B, et al. deepTools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Res.* 2016;44(W1):W160–W165. https://doi.org/10.1093/nar/gkw257

[39] Pertea G, Pertea M. GFF Utilities: GffRead and GffCompare. *F1000Research.* 2020;9:304. https://doi.org/10.12688/f1000research.23297.2

[40] Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods.* 2017;14(4):417–419. https://doi.org/10.1038/nmeth.4197

[41] Bray NL, Pimentel H, Melsted P, Pachter L. Near-optimal probabilistic RNA-seq quantification. *Nat Biotechnol.* 2016;34(5):525–527. https://doi.org/10.1038/nbt.3519

[42] Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics.* 2010;26(1):139–140. https://doi.org/10.1093/bioinformatics/btp616

[43] Law CW, Chen Y, Shi W, Smyth GK. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* 2014;15(2):R29. https://doi.org/10.1186/gb-2014-15-2-r29

[44] Liberzon A, Birger C, Thorvaldsdóttir H, et al. The Molecular Signatures Database Hallmark gene set collection. *Cell Syst.* 2015;1(6):417–425.

[45] Korotkevich G, Sukhov V, Budin N, et al. Fast gene set enrichment analysis. *bioRxiv.* 2021;060012 (version 3, posted 1 February 2021). https://www.biorxiv.org/content/10.1101/060012v3

[46] Yu G, He QY. ReactomePA: an R/Bioconductor package for reactome pathway analysis and visualization. *Mol BioSyst.* 2016;12(2):477–479. https://doi.org/10.1039/c5mb00663e

[47] Young MD, Wakefield MJ, Smyth GK, Oshlack A. Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biol.* 2010;11(2):R14. https://doi.org/10.1186/gb-2010-11-2-r14

[48] Hänzelmann S, Castelo R, Guinney J. GSVA: gene set variation analysis for microarray and RNA-seq data. *BMC Bioinformatics.* 2013;14:7. https://doi.org/10.1186/1471-2105-14-7

[49] Kim D, Paggi JM, Park C, Bennett C, Salzberg SL. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nat Biotechnol.* 2019;37(8):907–915. https://doi.org/10.1038/s41587-019-0201-4

[50] Anders S, Pyl PT, Huber W. HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics.* 2015;31(2):166–169. https://doi.org/10.1093/bioinformatics/btu638

[51] Putri GH, Anders S, Pyl PT, Pimanda JE, Zanini F. Analysing high-throughput sequencing data in Python with HTSeq 2.0. *Bioinformatics.* 2022;38(10):2943–2945. https://doi.org/10.1093/bioinformatics/btac166

[52] Grabherr MG, Haas BJ, Yassour M, et al. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat Biotechnol.* 2011;29(7):644–652. https://doi.org/10.1038/nbt.1883

[53] Li D, Liu CM, Luo R, Sadakane K, Lam TW. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics.* 2015;31(10):1674–1676. https://doi.org/10.1093/bioinformatics/btv033

[54] Van der Auwera GA, O'Connor BD. *Genomics in the Cloud: Using Docker, GATK, and WDL in Terra.* O'Reilly Media; 2020. (GATK Best Practices for RNA-seq short variant discovery.)

[55] Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal.* 2011;17(1):10–12.

[56] Krueger F. *Trim Galore: A Wrapper Around Cutadapt and FastQC.* Babraham Bioinformatics. https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/

[57] Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics.* 2014;30(15):2114–2120. https://doi.org/10.1093/bioinformatics/btu170

[58] Wingett SW, Andrews S. FastQ Screen: a tool for multi-genome mapping and quality control. *F1000Research.* 2018;7:1338. https://doi.org/10.12688/f1000research.15931.2

[59] Deng ZL, Münch PC, Mreches R, McHardy AC. Rapid and accurate identification of ribosomal RNA sequences via deep learning. *Nucleic Acids Res.* 2022;50(10):e60. https://doi.org/10.1093/nar/gkac112

[60] Ritchie ME, Phipson B, Wu D, et al. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* 2015;43(7):e47. https://doi.org/10.1093/nar/gkv007

[61] Gautier L, Cope L, Bolstad BM, Irizarry RA. affy—analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics.* 2004;20(3):307–315. https://doi.org/10.1093/bioinformatics/btg405

[62] Schuirmann DJ. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *J Pharmacokinet Biopharm.* 1987;15(6):657–680. https://doi.org/10.1007/bf01068419

[63] Kolberg L, Raudvere U, Kuzmin I, et al. g:Profiler—interoperable web service for functional enrichment analysis and gene identifier mapping (2023 update). *Nucleic Acids Res.* 2023;51(W1):W207–W212. https://doi.org/10.1093/nar/gkad347

[64] Szklarczyk D, Kirsch R, Koutrouli M, et al. The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Res.* 2023;51(D1):D638–D646. https://doi.org/10.1093/nar/gkac1000

[65] Csardi G, Nepusz T. The igraph software package for complex network research. *InterJournal Complex Systems.* 2006;1695.

[66] Pedersen TL. *ggraph: An Implementation of Grammar of Graphics for Graphs and Networks.* R package. https://cran.r-project.org/package=ggraph

[67] Shannon P, Markiel A, Ozier O, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 2003;13(11):2498–2504. https://doi.org/10.1101/gr.1239303

[68] Kent WJ, Sugnet CW, Furey TS, et al. The human genome browser at UCSC. *Genome Res.* 2002;12(6):996–1006. https://doi.org/10.1101/gr.229102

[69] Love MI, Anders S, Huber W. *Analyzing RNA-seq data with DESeq2.* Bioconductor package vignette. https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

[70] Bioconda contributors. *Bioconda documentation: usage and channel priority.* https://bioconda.github.io/

[71] Dobin A. *STAR stranded gene-count conventions and unstranded totals.* Upstream explanations, issues 842 and 1589. https://github.com/alexdobin/STAR/issues/1589

[72] STAR developers. *STAR parameter reference: index and BAM-sort memory settings.* https://github.com/alexdobin/STAR/blob/master/source/parametersDefault

[73] Salmon developers. *Salmon documentation: selective alignment and decoy-aware transcriptomes.* https://salmon.readthedocs.io/en/stable/salmon.html

[74] KEGG. *KEGG GENES database and identifier conventions.* https://www.genome.jp/kegg/genes.html

[75] STRING consortium. *What is a STRING network?* https://en.string-db.org/help/scores/

[76] Antoszewski K, Chmielewska K, Jagiello K, Puzyn T. Designing RNA sequencing experiments: A practical guide to reproducible gene expression analysis. *Comput Struct Biotechnol J.* 2026;31:101–119. Published online 18 December 2025. https://doi.org/10.1016/j.csbj.2025.12.015

[77] Wagner MR, Kleiner M. How thoughtful experimental design can empower biologists in the omics era. *Nat Commun.* 2025;16:7263. https://doi.org/10.1038/s41467-025-62616-x

[78] Chen Y, Chen L, Lun ATL, Baldoni PL, Smyth GK. edgeR v4: powerful differential analysis of sequencing data with expanded functionality and improved support for small counts and larger datasets. *Nucleic Acids Res.* 2025;53(2):gkaf018. https://doi.org/10.1093/nar/gkaf018

[79] Rau A, Marot G, Jaffrézic F. Differential meta-analysis of RNA-seq data from multiple studies. *BMC Bioinformatics.* 2014;15:91. https://doi.org/10.1186/1471-2105-15-91