Introduction to Bioinformatics and Computational Biology

2021-02-16

Contents

Chapter 1

Course information

This is the course material for STAT115/215 BIO/BST282 at Harvard University.

1.1 Contributors

Xiaole Shirley Liu (lead instructor)
Joshua Starmer
Martin Hemberg
Ting Wang
Feng Yue

Ming Tang
Yang Liu
Bo Yuan
Jack Kang
Scarlett Ge
Jiazhen Rong
Phillip Nicol
Maartin De Vries

We thank many colleagues in the community, who helped Dr. Liu in prepare the STAT115/215 BIO/BST282 course over the years. Some of the lecture slides acknowledged their contributions, but these contributors are not individually acknowledged here.

Chapter 2

Introduction

2.1 Brief history of bioinformatics

2.1.1 Protein structure wave

2.1.2 Gene expression wave

2.1.3 Genome sequencing wave

2.1.4 Big data challenge from sequencing

2.2 Should I take this course?

2.2.1 Bioinformatics vs computational biology

2.2.2 Is this class for me?

2.3 Course information

2.3.1 Logistics

2.3.2 X Shirley Liu lab introduction

2.4 Lab 1

2.4.1 Introduction

2.4.2 Introduction to R

2.4.3 Introduction to Bash

2.4.4 Getting started with Cannon

Chapter 3

High throughput sequencing

3.1 Three generations of sequencing technologies

First generation sequencing is Sanger sequencing. It is the technology that was used to obtain the first human genome sequence.

Second generation sequencing is also called next generation sequencing (NGS) and is the start of high throughput sequencing. It is what scientists use most often nowadays, and Illumina is the market leader. Most of the rest of this course will cover data analysis using second generation sequencing.

Third generation sequencing is single-molecule sequencing. There are many new technologies still under active development, although none has reached market penetration.

3.2 FASTQ and FASTQC

NGS generates FASTQ files. FASTQC is an computational approach to evaluate the quality of your NGS data.

3.3 Early sequence alignment (1 with 1)

In the early days (1970s), scientists were not worried about having to align too many sequences. They wanted to find the best alignment between two

sequences. Many bioinformatics courses start with learning these, although it is not the main focus of our course. We included two videos in case you are interested.

The Needlemen-Wunsch algorithm is the earliest algorithm to find the alignment between two sequences and score their similarity.

When two sequences are long, and only a portion of them can align well with each other, the Smith-Waterman algorithm can find the best local sequence alignment. It is still considered the best alignment approach, although it is slow.

3.4 Sequence search algorihtms (1 with many)

With more and more sequences available in the public in the 1980s, scientists were interested in finding whether their newly sequenced string has been sequenced before in the public database. Therefore, the fast search algorithm BLAST was developed, using one sequence as the query to find similar sequences from a database.

3.5 Borrow-Wheeler Aligner (many with many)

With NGS, scientists need much faster search (aka mapping) algorithms in order to align the millions of sequences to the reference genome. The current best algorithm is called Borrow-Wheeler Aligner or BWA.

In order to understand BWA, we first need to introduce Borrows-Wheeler transformation and LF mapping

The basic idea of Borrows-Wheeler alignment

3.6 Alignment output

NGS raw data is in FASTQ. Alignment gives you SAM (alignment) or BAM (binary version of SAM) files which contain the sequence information in FASTQ and the mapping locations. BED file is the simpliest, although there is information loss.

Chapter 4

RNA-seq Quantification

4.1 Introduction to RNA-seq experiment

4.2 RNA quality control and experimental design

4.3 Alignment

4.4 RNA-seq QC

4.5 RNA-seq expression index

4.6 RSEM and Salmon

4.7 RNA-seq read distribution

4.8 Lab 2

4.8.1 STAR tutorial

4.8.2 RSeQC tutorial

4.8.3 RSEM/Salmon Tutorial

Chapter 5

Differential expression, FDR, GO, and GSEA

DESeq2 is a popular and accurate computational algorithm to detect differential gene expression from RNA-seq data. It includes many elegant quantitative considerations, such as:

- Normalize the gene read counts by library size and composition
- Model gene read counts with negative binomial distribution
- Use hierarchical modeling to stabilize the gene variance
- Use Benjamini-Hochberg to calculate control for false discovery rate of calling differentially expressed genes
- Filter lowly expressed genes to reduce the number of hypotheses to be tested

5.1 DESeq2 library normalization

5.2 DESeq2 variance stabilization

5.3 Multiple hypotheses testing and False Discovery Rate

5.4 DESeq2 gene filtering

5.5 Gene Ontology (GO analysis)

5.6 Gene Set Enrichent Analysis (GSEA)

5.7 DESeq2 tutorial

Chapter 6

Clustering

6.1 Heatmap and clustering quality

6.2 Hierarchical cluster

6.3 K means cluster

6.4 Pick K and consensus clustering

6.5 Batch effect removal

6.6 Lab3

6.6.1 PCA tutorial

6.6.2 Clustering tutorial

6.6.3 Combat tutorial

6.6.4 DESeq2 Tutorial

6.6.5 DAVID/GSEA Tutorial

Chapter 7

Dimension Reduction

RNA-seq samples have tens of thousands of genes, although many genes might not vary much between samples and many others have correlated gene expression. Dimension reduction techniques aim to reduce the dimension of representing each sample with tens of thousands of genes to much fewer demensions, e.g. 2 to 100.

7.1 Principal Component Analysis: idea behind PCA.

PCA / SVD automatically outputs PC1, PC2, PC3, etc, with earlier PCs capturing the highest level of variability in the original data. Each PC is a linear combination of raw gene expression, and is orthogonal to all other PCs.

7.2 Principal Component Analysis: PCA applications.

PCA is a widely used method to project samples with high dimensions (e.g. with gene expression data) onto two dimensions for better visualization. It is an intuitive way to identify sample clusters, and identify batch effect.

7.3 Multidimensional Scaling (MDS)

MDS can use differential ways to calculate pair-wise distance, then use lower dimensions to satisfy the pair-wise distance. PCA is a special case of MDS.

7.4 Linear discriminant Analysis (LDA)

LDA is not only a dimension reduction method, but also a supervised machine learning method.

Chapter 8

Classification

8.1 Introduction

Imagine you have RNA-seq of a collection of labeled normal lung and lung cancer tissues. Given a new sample of RNA-seq from the lung with unknown diagnosis, will you be able to predict based on the existing labeled samples and the expression data whether the new sample is normal or tumor? This is a sample classification problem, and it could be solved using **unsupervised** and **supervised** learning approaches.

Unsupervised learning is basically clustering or dimension reduction. You can use hierarchical clustering, MDS, or PCA. After clustering and projection the data to lower dimensions, you examine the labels of the known samples (hopefully they cluster into separate groups by the label). Then you can assign label to the unknown sample based on its distance to the known samples.

Supervised learning considers the labels with known samples and tries to identify features that can separate the samples by the label. Cross validation is conducted to evaluate the performance of different approaches and avoid over fitting.

StatQuest has done an amazing job with machine learning with a full playlist of well organized videos. While the full playlist is worth a full course, for the purpose of the course, we will just highlight a number of widely used approaches. They include logistic regression (this is considered statistical machine learning), K nearest neighbors, random forest, and support vector machine (these are considered computer science machine learning).

8.2 Supervised learning

8.3 Cross validation

8.4 Regression

8.5 Regularization

8.5.1 Ridge regression

8.5.2 LASSO regression

8.6 KNN

8.7 Decision trees

8.8 Random forest

8.9 SVM

Chapter 9

Module I Review

9.1 Gene Expression Module Summary

9.2 Gene Expression Analysis Scenarios

Chapter 10

Transcription Factor Motif Finding

Chapter 11

ChIP-seq, Expression Integration

11.1 ChIP-seq

11.2 BWA and MACS

11.3 ChIP-seq QC

11.4 TF interactions (motif)

11.5 TF target genes (expression integration)

Chapter 12

Epigenetics, DNA Methylation

12.1 Epigenetics

12.2 DNA methylation

12.3 Promoter function

12.4 Gene body function

12.5 Enhancer function

12.6 Repetitive region function

12.7 Early cancer detection

Chapter 13

Histone Modifications , Chromatin Accessibility

Chapter 14

Long Range Chromatin Interactions

Chapter 15

Hidden Markov Model

15.1 Intro to HMM

15.2 Pb1: Forward & backward procedure

15.3 Pb2: Viterbi algorithm

15.4 Pb3: Parameter estimation

15.5 HMM application

Chapter 16

Module II Review

16.1 Module II Review

16.2 Practive Questions

Chapter 17

SNP and GWAS

17.1 SNP and LD

17.2 Family-based vs case-control association studies

17.3 GWAS studies and catalog

17.4 GTEx and eQTL

Chapter 18

GWAS and Epigenomics

18.1 Find tissue / cell type

18.2 Identify causal SNPs and genes

18.3 Predict phenotypes

Chapter 19

Single-cell RNA-seq (1)

Chapter 20

Single-cell RNA-seq (2)

20.1 Annotate scRNA-seq clusters

20.2 Differential expression

20.3 Batch effect removal

20.4 Pseudotime

20.5 Overload 10X

20.6 Other applications (CITE-seq, multi-seq, spatial transcriptomics)

Chapter 21

scATAC-seq

Chapter 22

Module III Review

22.1 Module III Review

Chapter 23

Cancer Genome Sequencing , Mutation analyses

Chapter 24

Cancer Subtyping, Survival Analyses

Chapter 25

Targeted Therapy, Drug Resistance, Compound and Genetic Screens

Chapter 26

Cancer Immunotherapy (1)

26.1 Systemic immunotherapy

26.2 Personalized immunotherapy

26.3 HLA and neoantigens

26.4 Tumor immune deconvolution

26.5 T cell signaling (PD1/PDL1, etc)

26.6 Other immune-cells (scRNA-seq)

Chapter 27

Cancer Immunotherapy (2)

Chapter 28

CRISPR Screens

Chapter 29

Module IV Review and Course Review

29.1 Module IV Review

29.2 Course Review