Variant Calling and Clustering on RNA-Seq Data

by Paul Arndt and Karsten Tausche
03.02.2017
Agenda

● The basics
● History of DNA Sequencing Technologies
● RNA-Seq and Variant Calling Pipelines
● Clustering on mixed data
● Hands-on: Variants in practice
● Ethics discussion
The basics
The basics

- **RNA sequencing expression**
  - differences in mapped reads between different samples → compare the amount of specific genes
  - quantitative data

- **Variant Calling: DNA vs. RNA**
  - DNA sequencing: analytically complex and not very efficient
  - RNA sequencing: cheaper, and, because of the traditionally used RNA sequencing expression analysis, the data is already there
    - → Variant Calling on RNA Data
    - but: beware that RNA only contains genes expressed in the analyzed cells, not the whole genome
The basics

Variants

→ differences in genes, according to a reference genome

Natalie  ATA TGA TCA ACA CTT

Steven  ATA TGA TCA ACA GTT

● SNPs (Single Nucleotide Polymorphisms) vs CNVs (Copy Number Variant)
● Risk Variants vs Protective Variants
History of Genetics

Relatively short history is basis for our current understanding

- 1869: Nucleic acid
- 1919: Polynucleotide model: four bases, sugar, phosphate
- 1944: Genes
- 1954: Structure of the Deoxyribonucleic acid (DNA)
- 1984: Initial Idea of the “Human Genome Project”
- 2000: First Draft of HG
- 2003: HG completely sequenced
DNA Sequencing Technologies

Human Genome: 3.2 Gbp (Million basepairs)

- **First Generation Sequencing (ABI 2002): Human Genome Project**
  - Very high accuracy (> 99.99%)
  - Slow processing (1 run = 100kbp, 3h)

- **Next Generation Sequencing: Illumina (2006): Today’s Standard**
  - Acceptably high accuracy (> 99.9%)
  - 2006: 1Gbp / run, 2016: 1 Tbp / run (6 days)
  - **Short read length: 200-400bp, later up to 700bp → fragmented output!**

- **Pacific Biosciences: Third Generation Sequencing (2013)**
  - Long read sequencing: 60kbp (“DeNovo Alignment”)
  - Accuracy > 99% (!)
Illumina Sequencing Process (simplified)

1) Preparation
   ○ Fragmentation of DNA into chunks (“reads”)
   ○ Required to be able to read sequence
   ○ 200-800 bp (3.2 Gbp in Human Genome!)

2) Amplification
   ○ Generate readable DNA regions (clusters)

3) Sequencing
   ○ Light reflected differently by each nucleotide
   ○ Record laser light reflection image
   ○ Generate textual output from recorded image

Illumina Sequencing Process (simplified)

1) Preparation
   ○ Fragmentation of DNA into chunks (“reads”)
   ○ Required to be able to read sequence
   ○ 200-800 bp (3.2 Gbp in Human Genome!)

2) Amplification
   ○ Generate readable DNA regions (clusters)

3) Sequencing
   ○ Light reflected differently by each nucleotide
   ○ Record laser light reflection image
   ○ Generate textual output from recorded image

[An Introduction to Next-Generation Sequencing Technology, illumina, 2016]
RNA-Seq based Pipelines

Reference Genome

Alignment

Mapped Reads

Quantification

Expression Data

RNA-Seq Reads

Variant Calling

Variants
Short Read Alignment

- Sequenced RNA: Many small RNA chunks (reads)
- Locate related position in the reference genome
  - Could be anywhere in the coding regions
  - Many highly similar regions within the DNA
  - Related coding DNA part may contain non-coding (irrelevant) parts
  - Editing events occur at specific regions/genes
- Process aligned reads
  - Probably many reads for same locations
  - Partly overlapping reads
  - Contradictory information
  - Apply statistical methods
Proteine Quantification Pipeline

- Multiple input samples (e.g., two conditions, healthy, ill)
- **Transcriptome**: set of all mRNA in a cell
  ($\approx$ genes expressed in that cell)
- **Differential Expression**
  
  Differences of mRNA quantities between the samples
RNA-Seq based Variant Calling

- **Filtering**
  - Deduplication
  - Remove low-quality reads (defined by sequencing device)
  - Filter unmapped reads
  - Filter low quality reads/mappings

- **Variant Calling**
  - Find deviation from reference genome

- **Postprocessing**
  - Separate Variants from Indels
  - Filter low-quality variants
  - Filter false-positive variants
RNA-Seq based Variant Calling Pipelines

**SNPiR:** “Reliable Identification of Genomic Variants from RNA-Seq Data” [Piskol 2013]

- High sensitivity
  - Loose criteria in variant calling step
- High specificity
  - Extensive filtering to omit false-positives
- Based on tools optimized for DNA-Seq Data

**GATK Best-Practices:** “Calling variants in RNAseq” [2014-2017]

- Built on newer tools, specialized for RNA-Seq Data
- Including some concepts of SNPiR
DATA CLEANUP

Non-GATK

Raw RNAseq Reads
- Map to Reference
  - STAR 2-pass
- Mark Duplicates & Sort (Picard)
- Split’N’Trim + ReassignMappingQuality
- Indel Realignment
- Base Recalibration

Analysis-Ready RNAseq Reads

VARIANT DISCOVERY

Analysis-Ready RNAseq Reads
- Variant Calling
  - HC in RNAseq mode

Raw Variants
- SNPs
- Indels

Variant Filtering
- RNAseq-specific settings

Filtered Variants
- SNPs
- Indels

EVALUATION

Analysis-Ready Variants
- SNPs & Indels

Variant Annotation

Variant Evaluation
- look good?
  - troubleshoot
  - use in project

Phasing
Alignment Across Splice Junctions

- Genome consists of **exons** (coding) and **introns** (non-coding)
- **Splicing**: removal of introns, joining of adjacent exons
- Not all **splice junctions** are known
- How to align reads across splice junctions?
Alignment Across Splice Junctions

- **Alignment to genome only?**
  - Algorithm would probably find a similar (wrong) location

- **Alignment to transcriptome only?**
  - Transcriptome may not be complete

- **Combined approach!**
  - Align to Genome
  - + known parts of the transcriptome
GATK: Two-Pass Alignment

- Using **STAR** aligner
  - State-of-the-art for RNA-Seq data

- Option: “2-pass STAR”
  - Detect splice junctions in first run
  - Use generated information in second run
    → final alignment

- Not using previously known splice junctions
  - No additional data dependencies
  - Missing information?
Filtering based on Genome Annotation

[USCS Genome Browser: Genomes + Annotations]

RepeatMasker Annotation

- Genome contains highly repetitive regions
- Controlling transcriptions, immunity against foreign DNA, …
- Generally non-coding
- Difficult/impossible to correctly align reads to
Filtering based on Genome Annotation

**RNA Editing Sites**

- Nucleotide sequence differs from original sequence in DNA
- Complicates read alignment
- Differences must not be interpreted as variants

![RNA Editing Diagram](https://en.wikipedia.org/wiki/RNA_editing)

**Description:**

1. **DNA** (AGCTGCAATTCGGCAATTCGCGATACGCG)
2. **Transcription of DNA to Pre-edited RNA** (AGCUGCAAUUGCUGCAUUCCAAACCGGAUACGC)
3. **Guide RNA gives template for editing** (UCGACGGUUAACGAACGUAAGAGUUGGCCUAUG)
4. **Editing of pre-mRNA** (AGCUGCAAUUGCUGCAUUCCAAACCGGAUACG)
5. **Pre-edited RNA** (AGCUGCAAUUGCUGCAUUCCAAACCGGAUACG)
6. **Edited RNA** (AGCUGCAAUUGCUGCAUUCCAAACCGGAUACG)
7. **Translation to protein or other pathway**
Filtering based on Genome Annotation

- Heavily used by SNPiR
  - Pseudo-Chromosomes
  - Post-processing after variant calling

- Not part of the GATK-Pipeline
  - Relying on advanced, specialized tools
  - Not relying on previously known data

- Apply SNPiR filtering to GATK-Pipeline?
  - Focus on human genome: rich information available
  - Filtering most reliable variants based on all known data
Statistical Filtering Strategies

● Statistical decisions in whole pipeline
  ○ Quality scores for alignment (depth, certainty)
  ○ Quality scores for called variant
  ○ Uncertainties in reference genome, two DNA strands, …

● Quality score evaluation requires reference scores
  ○ “Base quality score recalibration”
  ○ Data available for DNA-Seq
  ○ Not yet available for RNA-Seq

● Evaluation using known DNA-Seq variants
  ○ Currently most reliable way to verify tools and pipelines
Raw Sequencing Data: FASTQ Files

@SRR831012.1 HWI-ST155_0742:7:1101:1284:1981/1
NGAGATGAAGCACTGTAGCTTGGAATTTCGCGTGCCCAAGGAACCTCCAGT
+
%1=DDDFHHHGGFIIIIIIIIIIIIIIIIIEHIIIIIIIIIFIIIIIIII

@SRR831012.2 HWI-ST155_0742:7:1101:2777:1998/1
NGAGATGAAGCACTGTAGCTTTGGAATTTCGCGTGCCCAAGGAACCTCC
+
%1=DDFFHHHHHHHHIIIIIIIIIIIIIIIIIGIIIIIIIIIIIIIIIIIG

Quality score (increasing from worst to best):
"#$%&'()*+,-./0123456789:;<=?>?ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^`abcdefghijklmnopqrstuvwxyz{|}~

Experimental Setup

In our setting:
- ~1.4 GB per file
- ~8 Mio reads per file
- 80 files

RNAseq Intro
Milena Kraus, Apr 19, 2016
VCF: Variant Call Format

Example

```vcf
##fileformat=VCFv4.0
##fileDate=20100707
##source=VCFtools
##reference=NCBI36
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">  
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">  
##FORMAT=<ID=GT,Number=1,Type=Integer,Description="Genotype">  
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality (phred score)">  
##FORMAT=<ID=GP,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA genotypes (R=ref,A=alt)">  
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">  
##ALT=<ID=DEL,Description="Deletion">  
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">  
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">  

#CHROM POS ID REF ALT QUAL FILTER INFO  
1  1 . ACG A,AT . PASS .  
1  2 rs1 C T,CT . PASS H2;AA=T  
1  5 . A G . PASS .  
1 100 T <DEL> . PASS SVTYPE=DEL;END=300
```

Mandatory header lines

Optional header lines (meta-data about the annotations in the VCF body)

Reference alleles (GT=0)

Alternate alleles (GT>0 is an index to the ALT column)

Phased data (G and C above are on the same chromosome)

How to make sense of the data

open question: What do newly sequenced genes do?

- infer correlations between different genes - allowing for example the building of classifiers to improve diagnosis, ...

other general use cases for clustering in bioinformatics:

- complexity reduction
How to make sense of the data
Clustering

Main Principles: Homogeneity, Separation

very intuitive for us in 2-D

Problem: n-dimensional data
  ● curse of dimensionality
Types of Clustering

Hierarchical
- Agglomerative (bottom-up)
- Divisive (Top-down)

Partitional
- Error Minimization
- Graph theoretic
- Density based
- Model based
  - minimal Spanning Tree
  - expectation maximization
  - Decision Tree
Types of Clustering

Hierarchical
- Agglomerative (bottom-up)
- Divisive (Top-down)
- Error Minimization
  - K-means
  - minimal Spanning Tree

Partitional
- Graph theoretic
- Density based
- Model based
  - expectation maximization
  - Decision Tree
Example Hierarchical Clustering

1. Every node is assigned its own cluster
2. Find the closest pair of nodes and merge them into a cluster
3. Repeat step 2, until all nodes in the network have been merged into a single large cluster
4. Choose a useful clustering threshold between the bottom and top levels
Example Hierarchical Clustering

How do you compute the distance between clusters?

- Single-link: merge two clusters with the smallest minimum pairwise distance
- Average-link: merge two clusters with the smallest average pairwise distance
- Maximum-link or Complete-link: merge the two clusters with the smallest maximum pairwise distance
Types of Clustering

Hierarchical
- Agglomerative (bottom-up)
- Divisive (Top-down)

Partitional
- Error Minimization
- Graph theoretic
- Density based
- Model based
  - minimal Spanning Tree
  - expectation maximizing
  - Decision Tree

K-means
Example K-means

The main idea is to define \( k \) centroids, one for each cluster.

1. Select \( k \) entities as the initial centroids
2. (Re)Assign all entities to their closest centroids
3. Recompute the centroid of each newly assembled cluster
4. Repeat step 2 and 3 until the centroids do not change or until the maximum value for the iterations is reached
Example K-means

advantages:

- simple, fast, efficient (O(n))

disadvantages:

- difficult to predict K, often produces clusters of uniform size, spherical assumption
Handling Mixed Data

Clustering so far is almost exclusively done on quantitative data

Now: adding Variants (qualitative data) → mixed Data

Main Problem: How to compute distances?
Clustering - Distance measures

**COR** Pearson sample metric

**EISEN** Cosine correlation

**SPEAR** Spearman sample correlation distance

\[ d_{spear}(x, y) = 1 - \frac{\sum_{i=1}^{m} (x'_i - \bar{x}')(y'_i - \bar{y}')}{\sqrt{\sum_{i=1}^{m} (x'_i - \bar{x}')^2 \sum_{i=1}^{m} (y'_i - \bar{y}')^2}} \]
Gower Similarity

compares two cases i and j

- $S_{ijk}$: contribution provided by the k-th variable
- $w_{ijk}$: 1 or 0 depending on the comparison

basically case distinction depending on variable type
Gower Similarity

\[ S_{ij} = \frac{\sum_{k} w_{ijk} S_{ijk}}{\sum_{k} w_{ijk}} \]

\[ S_{ijk} = 1 - \frac{|x_{ik} - x_{jk}|}{r_{k}} \]

ordinal/continuous variables:

rk is range of values for the k-th variable
Gower Similarity

nominal variables:

\[ S_{ijk} = 1 \text{ if } X_{ik} = X_{jk} \text{ or } 0 \text{ if } X_{ik} \neq X_{jk} \]

\[ w_{jk} = 1 \text{ if both cases have observed states for } k \]
Gower Similarity

binary values

\[
S_{ij} = \frac{\sum_{k=1}^{n} w_{ijk} S_{ijk}}{\sum_{k=1}^{n} w_{ijk}}
\]
Multiple Factor Analysis

It may be seen as an extension of:

- Principal component analysis (PCA) when variables are quantitative,
- Multiple correspondence analysis (MCA) when variables are qualitative,
- Factor analysis of mixed data (FAMD) when the active variables belong to the two types.
Multiple Factor Analysis

PCA

MCA: also a dimension reducing method; it represents the data as points in 2- or 3-dimensional space.

indicator matrix or burt table
Step 1: $K$ tables of $J_k$ variables collected on the same observations

Step 2: Compute generalized PCA on each of the $K$ tables (where $\gamma$ is the first singular value of each table)
Step 3: Normalize each table by dividing by its first singular value ($\tau$)

$$
\gamma_1^{-1} \times X_1 = Z_1
$$

$$
\gamma_k^{-1} \times X_k = Z_k
$$

$$
\gamma_K^{-1} \times X_K = Z_K
$$
Step 4: Concatenate the $K$ normalized tables

Step 5: Compute a generalized PCA on the concatenated table
Clustering results - now what?

We will hopefully see some patterns that we can associate with diseases / known issues

To prove this, we can, for example, look at the Variants that got clustered together and check whether they are associated with similar problems
Hands-On: Genome Browser