Mid Project Demonstration

Application of Machine Learning Techniques to Next Generation Sequencing Quality Control

Sam Nicholls
msn

Department of Computer Science
Aberystwyth University

March 19, 2014

Next Generation Sequencing

What is Next Generation Sequencing?

- Major and rapid advances in genetic sequencing hardware
- Massively parallel; billions of simultaneous chemical reactions
- Both time and cost of genetic analysis has reduced

Why this project?

- Processes are complex and open to error
- Quality control is an essential step
- Must be able to assure confidence for downstream results

Next Generation Sequencing

What is Next Generation Sequencing?

- Major and rapid advances in genetic sequencing hardware
- Massively parallel; billions of simultaneous chemical reactions
- Both time and cost of genetic analysis has reduced

Why this project?

- Processes are complex and open to error
- Quality control is an essential step
- Must be able to assure confidence for downstream results

Aims

Report on state of the current quality control system at Sanger

- Working with Sanger Institute's Human Gentics Informatics Team
- `auto_qc` classifies samples as pass, fail or warn
- Current classifier consists of hard-coded simple thresholds
- `auto_qc` also requires timely human intervention

Goals

- Apply learning techniques to replicate current human rules
- Attempt to improve efficiency of current "warning" handling
- Identify new or unused parameters that improve classification

Aims

Report on state of the current quality control system at Sanger

- Working with Sanger Institute's Human Gentics Informatics Team
- `auto_qc` classifies samples as pass, fail or warn
- Current classifier consists of hard-coded simple thresholds
- `auto_qc` also requires timely human intervention

Goals

- Apply learning techniques to replicate current human rules
- Attempt to improve efficiency of current "warning" handling
- Identify new or unused parameters that improve classification

Aims

Identify lanelet properties that affect downstream variant calling

- For better QC we need an idea of "good" and "bad"
- How does quality affect analyses performed after sequencing?

Goals

- Will leaving out a sample during variant calling affect the result?
- Select a "representative" region of the human genome for analysis
- Compare calls on whole genome results to GWAS "SNP chips"
- Determine what is actually "good" and "bad" for QC

Aims

Identify lanelet properties that affect downstream variant calling

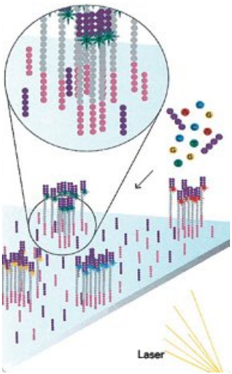
- For better QC we need an idea of "good" and "bad"
- How does quality affect analyses performed after sequencing?

Goals

- Will leaving out a sample during variant calling affect the result?
- Select a "representative" region of the human genome for analysis
- Compare calls on whole genome results to GWAS "SNP chips"
- Determine what is actually "good" and "bad" for QC

Analysis of Current QC System

Samples, Lanes and Lanelets (Oh my!)



Illumina DNA Synthesis

- Illumina HiSeq hardware; eight **lane** flowcell
- A **sample** is a distinct DNA specimen
- Samples are prepared with barcodes and amplified across multiple lanes
- The amplification process creates millions of clusters in each lane
- A lane thus contains more than one sample and samples can be spread across multiple lanes
- A **lanelet** represents the aggregate of all clusters in one particular lane that match the barcode of a particular sample

Image Harvard University Informatics and Scientific Applications: Illumina Sequencing Technology. http://bit.ly/1IMb4KG

Analysis of Current QC System

Data Access

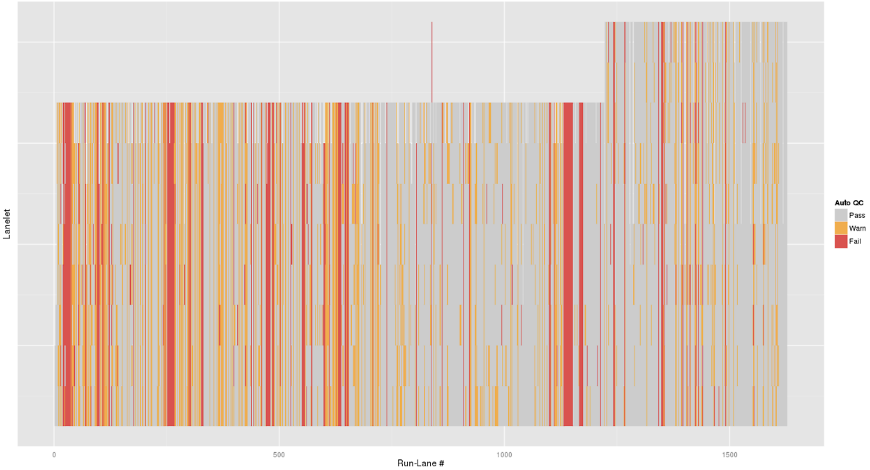
- Access to two of the largest studies at the institute
- 13,455 "lanelets"; aggregated clusters of a sample in one lane
- `auto_qc` **pass** 9,154 (68%), **fail** 1,542 (11%) **warn** 2,759 (21%)
- Possible access to another large data set on the horizon

Format

- Key-value statistical summary numbers from `samtools stats`
- `samtools stats` also generates tab-delimited dataframes measuring some metrics over cycle time
- Additional summary numbers gained by passing output of `samtools stats` through `bamcheckr`

Analysis of Current QC System

Brief Investigation of Classification Correlation



Analysis of Current QC System

Frontier

- My own Python script to read and process these data files
- Input formed by output of the current system's statistical data
- `Frontier`'s "StatPlexer" provides an API to access dataframe

Rule Extraction

- Utilising scikit-learn, a Python machine learning framework
- Training decision trees on key-value summary statistics
- Experimented with various parameter and data handling options
- Decision trees prone to overfitting but provide rules to follow

Analysis of Current QC System

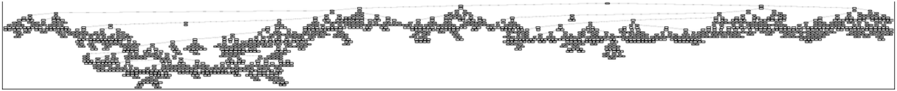
Frontier

- My own Python script to read and process these data files
- Input formed by output of the current system's statistical data
- `Frontier`'s "StatPlexer" provides an API to access dataframe

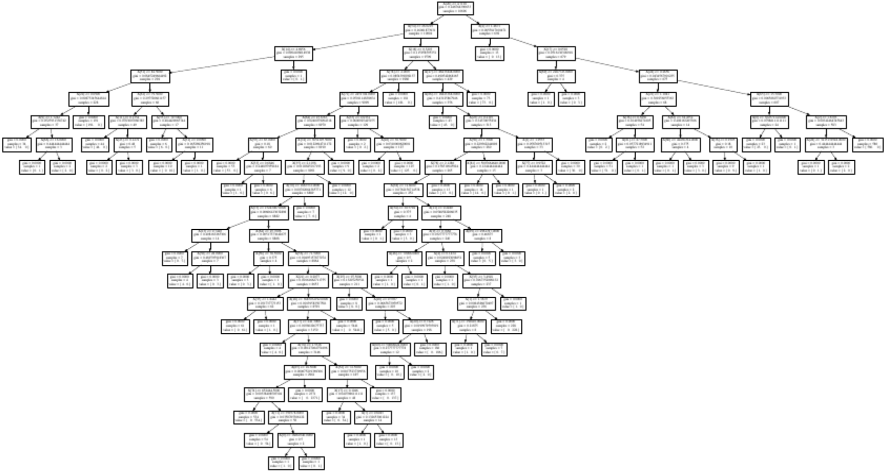
Rule Extraction

- Utilising scikit-learn, a Python machine learning framework
- Training decision trees on key-value summary statistics
- Experimented with various parameter and data handling options
- Decision trees prone to overfitting but provide rules to follow

Early Decision Tree



Improved Decision Trees



Contributions to Current QC System

- bamcheckr; an in-house tool written in R
- Supplements samtools stats key-value summary numbers which are then used by the current auto_qc system

What did I do?

- Patched a bug that prevented plotting of diagnostic graphs
- Writing additional routines to calculate percentage or ratio based parameters that should prove useful for training the decision tree

Downstream Progress

Variant Call Format (**VCF**)

- Stores called variant alleles at each location on the genome for each sample; a huge tab delimited file
- File also stores reference alleles at these locations and other meta-data including quality score and filters
- Downloaded and built a collection of tools; `vcftools` to generate indexes and query VCF files for particular columns

Latest Progress

- Locations of called variants across all "SNP chips" extracted
- Crude Python script to generate candidate regions

Project Issues: QC Report

Some issues encountered so far include:

Data Noise The "warn" classification seems to introduce a lot of noise to the generated decision trees

CV Confusion with performing cross-validation with scikit-learn, how best to stratify data we have?

Params The `bamcheckr` outputs are missing some percentage and ratio formatted parameters that auto_qc gains from another source

Pruning The scikit-learn framework does not support pruning of decision trees but analysis of the current trees indicate this would provide improvement for generalisation

Project Issues: Downstream Analysis

Comms. A misunderstanding wasted some time as I tried to recover strand data for some of the SNP chip data, whilst unnecessary it was interesting!

Region We need to locate a region of the genome that does not have too many or too few variants ("representative")

CPU Analysis will require use of Sanger computing clusters due to the intensive nature of the variant calling pipeline

Storage The size of the candidate region must be not so small as to hinder analysis but not too large to avoid computational time and storage limitations

What's next?

- Complete my code for selection of candidate genome regions
- Assist construction of Sanger pipeline to perform leave-one-out analysis, using candidate region as the target for variant calling
- Compare results of this pipeline to the known variants in the SNP chips; attempt to determine if there are any effects on accuracy
- Can we learn what QC parameters to look for in such cases?

What else?

- Refine attempts to replicate current QC rules
- Implement new weighting algorithm for cross-validation
- Implement pruning for the decision tree
- Complete contributions to `bamcheckr`

Future

- Opportunity to patch pruning algorithm in to scikit-learn
- Sanger Institute expressed desire to publish research

With more time...

- Investigate other machine learning algorithms and their application to the learning of quality control classification
- Increase the size or even use multiple extracted genomic regions to measure generalisation of what is learned through the leave-one-out methodology