Quiver practicum: building a high-quality consensus sequence from PacBio data

David Alexander, Patrick Marks

September 12, 2012

What is Quiver?

- A multiple-read consensus calling algorithm for PacBio reads
- Takes multiple reads of a given DNA template, outputs best guess of template's identity
- QV-aware hidden Markov model to model our sequencing errors; a greedy algorithm to find the maximum likelihood template.
- **Can achieve accuracy >Q50 (i.e. >99.999%) using pure PacBio raw reads.**

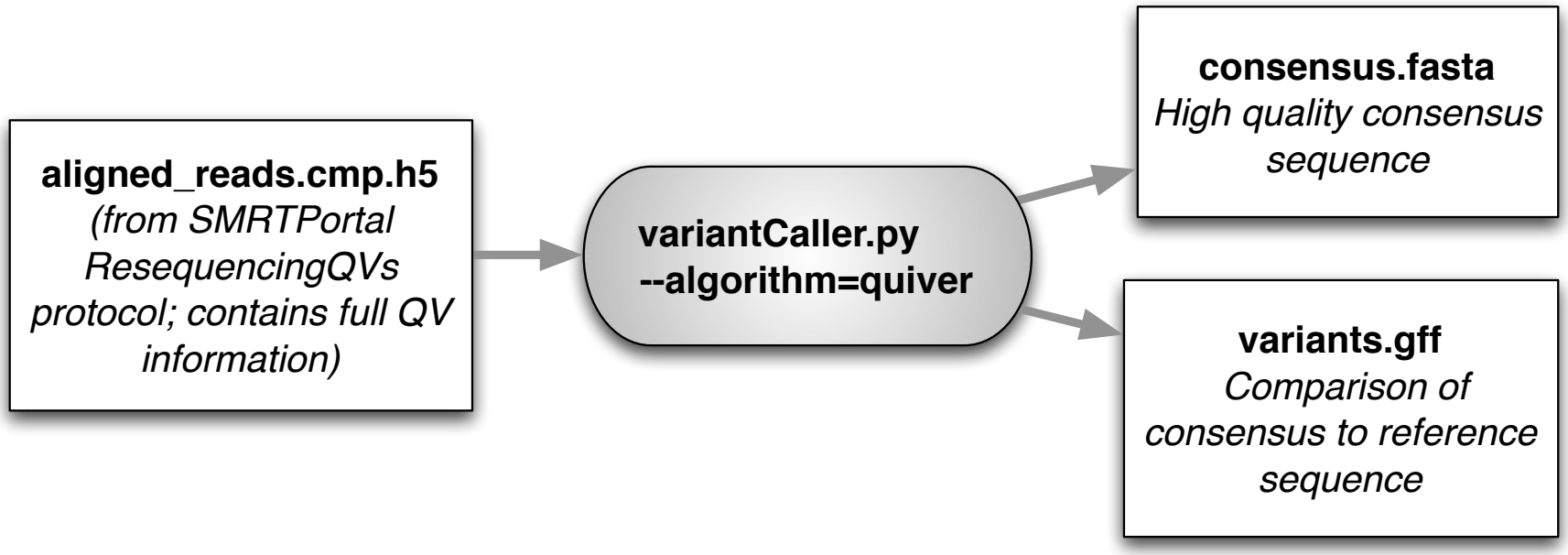
What Quiver isn't

- Quiver is NOT an "error-correction" algorithm
- Quiver is NOT YET a 100% polished piece of software
  - Has worked well in practice . . . but bugs may remain
  - Verification and testing process just beginning now

Quiver workflow

- Quiver takes a pile of PacBio reads (with QV information) that are from the same underlying DNA template, and infers the template identity.
- Needs to be fed a `cmp.h5` file (an alignment to a rough assembly, or an accurate reference).
- `cmp.h5` must have all QV data tracks: must be generated via `ResequencingQVs` protocol in SMRTPortal.

Quiver workflow diagram



Where does Quiver live?

- Driver program (`variantCaller.py`) for Quiver is in `GenomicConsensus` (Python)
  - requires the `pbcore` library for data file access
  - requires the `ConsensusCore` C++ library for core computational bits
- Will be available in 1.4 SMRTportal, but for now requires custom installation of these packages and command line work.

Let's install Quiver

(Read HowToQuiver document for details and system requirements; this is just the gist of it!)

Install (root privileges not required):

```
$ curl -L git.io/JR7TnQ | bash
```

Activate virtualenv via

```
$ source ~/VE-QUIVER/bin/activate
```

Example: Resequencing consensus on an Lambda job

Let's call resequencing-based consensus on a simple Lambda job! JobId 49002.

- 1 chip, 2 45" movies; 2Kb insert size. Mean coverage > 2800.
- I've already run it through the *ResequencingQVs* protocol

```
$ export PB=/mnt/secondary/Smrtanalysis/
$ variantCaller.py -j2 --algorithm=quiver                                  \
   $PB/userdata/jobs/049/049002/data/aligned_reads.cmp.h5                  \
   -r $PB/opt/smrtanalysis/common/references/lambda/sequence/lambda.fasta \
   -o consensus.fasta -o variants.gff
```

This command uses 100x coverage (subsamples) by default.

Examine the output

- `variants.gff` contains **zero** variant entries!
- The sequence in `consensus.fasta` is perfectly concordant with the lambda reference.

We have effectively assembled a perfectly accurate lambda genome from a fraction of a single PacBio chip, only using the true reference to segregate reads together — consensus is **de-novo**.

Sidebar: comparing genomic sequences

I recommend using MUMmer 3.0 for comparing two FASTA files purpose.

MUMmer includes a tool called `dnadiff`

```
$ dnadiff \
  $PB/opt/smrtanalysis/common/references/lambda/sequence/lambda.fasta \
  consensus.fasta
```

Examine:

- `out.report` for a summary of differences
- `out.snps` for SNPs;
- `gnuplot out.fg` for a zoomable dot-plot.

How was the reference used?

- Quiver only uses the alignment to the reference in order to group reads together.
- Quiver's consensus calls are completely independent of the reference—only use the reads (*Variant* calls still require reference for comparison.)

Quiver's accuracy potential

Indeed, we have found that Quiver can achieve accuracy >Q50 in real-world applications, even for true de-novo assemblies. Recent examples:

- *Meiothermus ruber*: Q37.5 before Quiver; Q50.6 after Quiver (including true variants)
- *Pedobacter heparinus*: QV51.5 after Quiver

Using Quiver for polishing an assembly

Same basic workflow as for resequencing before, only modifications are:

- User needs to upload rough assembly FASTA to SMRTPortal as a new reference;
- Then user needs to align against rough assembly in ResequencingQVs protocol
- Then, same as before!
- Output FASTA has same contigs as rough assembly, but accuracy will be higher.

This workflow is a hack

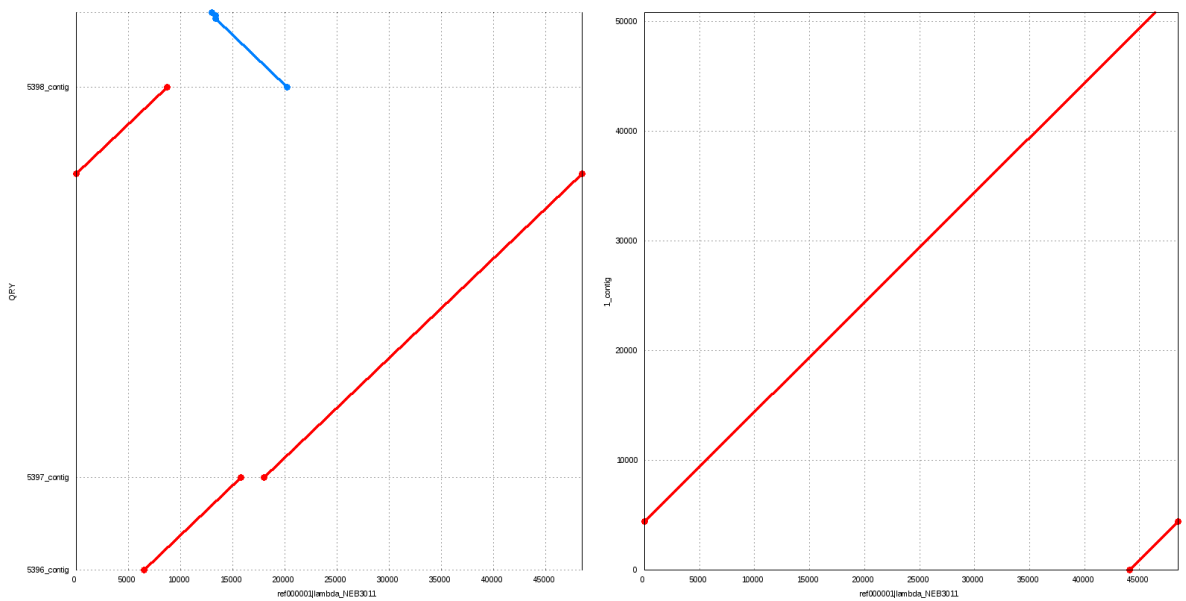
We will be making streamlining this workflow for the next SW release!

Step 1: Building the assembly

Step 1: build the best rough assembly (high N50, low # contigs) you can using PacBio or 3rd Party workflows.

Example: use *RS_Assembly* workflow with genome size set to good estimate.

- Important to manually inspect output. Bogus contigs?
- `mummerplot` can be used to compare assembly to a reference. Here are 3-contig and 1-contig lambda assemblies vs refererence.



Step 2: Uploading the assembly as a new reference

Upload the (curated) rough assembly FASTA file as a new reference to SMRTPortal

Step 3: Align to the rough assembly using *ResequencingQVs*

Select *ResequencingQVs*, and select the reference you have just uploaded

Example: My job was run and placed in Job 049061

Step 4: Invoke Quiver

```
$ export PB=/mnt/secondary/Smrtanalysis/
$ variantCaller.py -j2 --algorithm=quiver                                    \
   $PB/userdata/jobs/049/049061/data/aligned_reads.cmp.h5                    \
   -r assembled-1contig.fasta                                                \
   -o consensus.fasta -o variants.gff
```

Step 5: Examine accuracy

You can use `dnadiff` as before to give rough accuracy metrics, or you can use `blasr -m 0` to give text visualizations of alignments. Ex:

```
$ blasr lambda.fasta consensus.fasta -sa lambda.fasta.sa \
    -m 0 -out quiver-blasr.out
```

Output snippet:

```
...
  50  GGCGTTTCCGTTCTTCTTCGTCATAACTTAATGTTTTTATTTAAAATACC
   .  ||||||||||||||||||||||||||||||||||||||||||||||||||
4461  GGCGTTTCCGTTCTTCTTCGTCATAACTTAATGTTTTTATTTAAAATACC
...
```

Quiver accuracy should be substantially better than rough assembly!!

Effective coverage dips and `MapQV`

- As you would expect Quiver accuracy, scales with coverage.
- `variantCaller.py` filters out reads with low `MapQV`, so long repeats in the genome can end up inducing effective coverage deserts, and errors can pile up there.
- We will be adding diagnostic plots to show these...

Mixed samples

Quiver expects to be running on an unmixed, homogeneous sample

- Do not feed diploid data to Quiver! (yet)
- If there is suspicion of a mixed sample, examine Quiver output carefully.

Error profile

Quiver still makes occasional errors. By and large, these will be indels in homopolymer regions, but they should be very rare, given adequate coverage. We have found great results with coverage >50x.

```
  ...
 2200  AACTCTCACTCC-AAAAAAAAAAAAAAAAAAAAAAAAAGTCTAAATGCTT
    .  |||||||||||| ||||||||||||||||||||||||||||||||||||
32848  AACTCTCACTCCAAAAAAAAAAAAAAAAAAAAAAAAAGTCTAAATGCTT
  ...
```

Conclusions

- Quiver is used to improve accuracy of pure-PacBio consensus results: >Q50 attainable.
- Requires cmpH5 produced via *ResequencingQVs* worfklow
- **De-novo** consensus; reference not used to inform consensus calls.
- Quiver will be pre-packaged in SMRTPortal in the next major software release.