SPIRAL GENETICS

Spiral Sequencing Analysis Engine BioGraph™ QuickStart

Get up and running with Spiral Genetics®' BioGraph technology

Software Version 4.0.4

February 2019

Table of Contents

Overview

This walkthrough will show you how to:

- Install the BioGraph software
- Convert your data to BioGraph format
- Get up and running with the BioGraph tools that will allow you to produce a project level set of high-confidence structural variant calls.

The BioGraph Platform

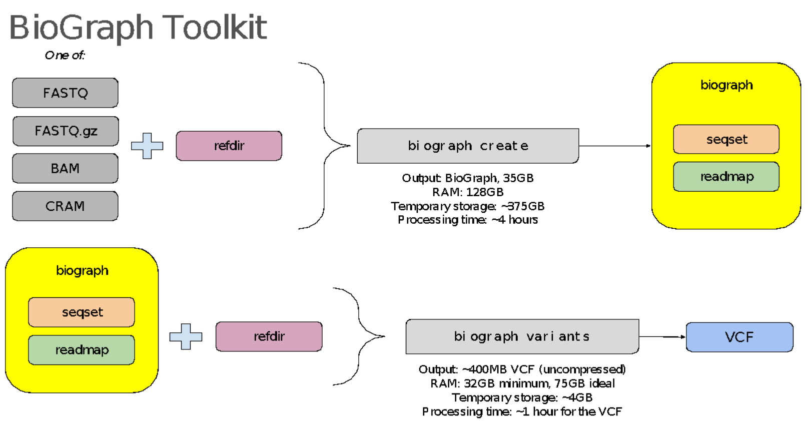
BioGraph is part of the Spiral Sequencing Analysis Engine. The BioGraph Platform consists of:

- BioGraph Format
- BioGraph Toolkit
    - Creates, merges, and calls variants from the BioGraph format
- BioGraph SDK
    - Builds custom analyses
    - Extracts/queries BioGraph files

What is the BioGraph Format?

- BioGraph is not a graph.
- Properties of the data structure allow us to do read-overlap-graph type operations
- BioGraph does not hold a reference or variants.
- Allows quick queries to look for coverage/divergence of reference sequence.
- Reference agnostic.
- BioGraph is a data structure that holds compressed and indexed NGS reads.
- Holds a minimum set of maximal kmers
- What's the fewest, longest, unique kmers we can store while still preserving all the data?
- BioGraph tools are like a reverse BWA type resequencing pipeline.

What is the BioGraph Toolkit?



Read Correction

- Kmers are generated from the reads.
- Erroneous (very rare) kmers are discarded.
- A de Bruijn graph is constructed from the remaining set of “true” kmers.
- A search algorithm (similar to A*) finds any base substitutions that would allow each read to align to the graph with no mismatches.
- Erroneous bases are automatically trimmed from the end of the read as necessary.
- Reads that cannot be corrected are dropped.
- The read correction process does not consider base quality scores, and does not require base quality score recalibration.
- The process is reference-free.

Installation of the BioGraph Software

You will be provided with two **`tar`** files:

```
   biograph-4.0.4.tgz
   biograph_sdk-4.0.4.tgz
```

You will need to extract and install the files. Use the following commands:

```
$ tar xzvf biograph-4.0.4.tgz
$ export PATH=`pwd`/biograph-4.0.4/:$PATH
$ tar xzvf biograph_sdk-4.0.4.tgz
$ export PATH=`pwd`/biograph_sdk-4.0.4/:$PATH
$ cd biograph_sdk-4.0.4.tgz
$ pip install BioGraphSDK-4.0.4.tar.gz
```

Notes:

- It is also possible to install BioGraph on a cluster with no access to the internet. For details, see the BioGraph User Guide.
- You may need admin/sudo privileges to run the ‘pip’ command.

Python Dependencies

Note: We currently support Python 2.7. Support for Python 3 is planned for a future release.

The SDK requires Python 2.7 compiled with ucs4 unicode support. This is the default for most platforms. To verify that your Python interpreter was compiled with ucs4, run the following Python script:

```
import sysconfig
print sysconfig.get_config_vars()['CONFIG_ARGS']
```

If the **`--enable-unicode=ucs4`** option is present, then your interpreter is supported. If **`--enable-unicode=ucs2`** is present, then you will need to recompile Python or download a compiled Python 2.7 with **`ucs4`** enabled.

Additional Software

When performing further analysis on BioGraph created variants (such as creating a project-level VCF), a mix of existing open source tools is leveraged to help properly manipulate the data. The following tools are required to be in the user's environment:

- vcftools -VCFtools (0.1.15) - https://vcftools.github.io/index.html
- bcftools - bcftools 1.9 - https://samtools.github.io/bcftools/
- gnu parallel, https://www.gnu.org/software/parallel/
- RTG - RTG Tools 3.8.2 - https://github.com/RealTimeGenomics/rtg-tools

High Level Overview of SV Analysis

Here's how to generate high-confidence SV calls using the BioGraph format:

1. Convert a reference for the analysis
2. Convert NGS data (whole genome) to the BioGraph format
3. Call variants on the BioGraph format files
4. Consolidate variants across all the individuals
5. Genotype for all the variants back across each individual
6. Produce a project level set of high-confidence calls.

These steps are covered in detail in the sections that follow.

Creating a Reference

To get started converting samples to the BioGraph format, we must first create a BioGraph reference. A reference needs to be created only once per genomic reference (one for grch37, one for grch38, and so on).

The BioGraph reference is used when converting files to the BioGraph format as a performance optimization but the reference used does not tie your sample to that reference. Samples converted to the BioGraph format are not related to a reference.

The BioGraph reference is also used when calling variants over samples in the BioGraph format and subsequent analysis steps.

To create a reference, use the following command:

```
$ biograph reference --in /path/to/ref.fasta --refdir /path/to/bg_reference
```

The created biograph reference will be a directory called **`bg_reference`**. It should take about 40 minutes to create and be about +/-13GB for a Human reference.

Converting Data to the BioGraph Format

Once we have a reference created, we can start converting whole genome read data to the BioGraph format. This file includes all reads that passed the correction procedure. By default, all reads containing kmers that occur fewer than four times in the data set are either filtered out or corrected.

```
 $ biograph create --out /path/to/sample123.bg --ref /path/to/my_reference --in
sample123.[bam/fastq/cram] --id sample123 --tmp /path/to/large_disk --max-mem 256
```

Calling Variants from Data in the BioGraph Format

Once we have successfully created BioGraph format files, we can call variants on them:

```
 $ biograph variants --in /path/to/sample.bg --ref /path/to/bg_reference/
--out /path/to/sample123.bg/sample123.vcf --tmp /path/to/large_disk
```

Note that performance issues will occur if **`--tmp`** is over network storage. After creating the **`--out`** VCF, use **`vcf-sort`** from VCFtools and **`bgzip`** to finalize the data.

```
$ vcf-sort /path/to/sample123.bg/sample123.vcf | bgzip >
/path/to/sample123.bg/sample123.vcf
```

Creating a Reference of Population Variation

After running **`biograph create`** and **`biograph variants`** on multiple individuals, the **`bgtools merge_vcfs`** program allows you to create a population-level VCF file (pVCF) containing the set of variants discovered across individuals.

For each sample, run **`biograph create`** and **`biograph variants.`** Then merge the resulting variants:

```
$ bgtools merge_vcfs --output /path/to/project-unfiltered.vcf.gz --reference
 /path/to/bg_reference /path/to/sample1.vcf /path/to/sample2.vcf ...
$ bgtools freq_filter --min-observations 10 --individuals 2 --variants
/path/to/project-unfiltered.vcf.gz | bgzip > /path/to/project.vcf.gz
$ tabix -p vcf /path/to/project.vcf.gz
```

Confirming Calls with BioGraph Genotype

For each sample, run BioGraph Genotype (pcmp). This genotypes events (reads covering the allele) from a pVCF file for project level analysis.

```
$ bgtools pcmp --variants /path/to/project.vcf.gz --reference /path/to/bg_reference --
sample sample123 --min-insert
200 --max-insert 700 --biograph /path/to/sample.bg --output /dev/stdout | vcf-sort |
bgzip > /path/to/sample123_pcmp.vcf.gz
$ tabix -p vcf /path/to/sample123_pcmp.vcf.gz
```

To get an estimate of your insert-size minimum/maximum, use **`bgtools stats`**:
`bgtools stats --biograph /path/to/sample.bg --reference /path/to/bg_reference`.`

Producing a Project Level Set of High-Confidence Calls

To recombine the individual results genotyped with **`pcmp`**, use **`bgtools vcf_sample_paste`** to create a project-level VCF. If your samples include familial data, provide a pedigree file (.ped) and **`bgtools meanno`** will annotate sites and report overall statistics of Mendelian error.

```
$  bgtools vcf_sample_paste /path/to/sample123_pcmp.vcf.gz
  /path/to/sample124_pcmp.vcf.gz ... | bgtools meanno --variants - --pedigree
project.ped --output /path/to/project_pcmp_meanno.vcf
$ bgzip /path/to/project_pcmp_meanno.vcf
$ tabix -p vcf /path/to/project_pcmp_meanno.vcf.gz
```

Expected Results of Running the BioGraph Pipeline

After running the pipeline in its entirety, for N number of samples, you will have the following:

- (N) Samples converted to the BioGraph format
- (N) VCFs generated using the BioGraph Variant Caller
- One project-level VCF, (pVCF) with all events genotyped on a per-sample basis.