**The whole genome focused array SNP typing (*WG-FAST*) pipeline**

**Citation:**

**Contact:** Please address queries, concerns, improvements to jasonsahl at gmail dot com

**What does *WG-FAST* do?**

*WG-FAST* was designed as a tool to phylogenetically genotype unknown samples, even those with extremely low read coverage, in the context of a well-studied dataset.

**What does *WG-FAST* not do?**

*WG-FAST* is not intended to identify new SNPs in a dataset. If too many samples are processed with *WG-FAST*, a phylogenetic discovery bias can most certainly exist.

**Installation**

-The code is housed here: https://github.com/jasonsahl/wgfast.git

-Install the code with:

>git clone <https://github.com/jasonsahl/wgfast.git>

-The following line in “wgfast.py” must be edited to reflect your installation directory:

WGFAST\_PATH**=**"/Users/jsahl/wgfast"

**Unit tests**

-To test that all functions are working correctly in *WG-FAST*, type:

python tests/test\_all\_functions.py

If everything is correct, all tests should pass.

**Dependencies not included with *WG-FAST*:**

1. GATK – tested version is 2.72. This version requires Java 1.7. Should be back compatible with older versions. **Download Jar file and place in WGFAST\_PATH/bin**. Can be obtained from: <https://www.broadinstitute.org/gatk/download>
2. Samtools – tested version is 0.1.19. Must be in PATH as “samtools”. Can be obtained from: <http://samtools.sourceforge.net/>
3. BWA-MEM – tested version is 0.7.5a. Must be in PATH as “bwa”. Can be obtained from: <http://bio-bwa.sourceforge.net/>
4. BioPython – must be in PYTHONPATH. Can be obtained from: <https://github.com/biopython/biopython>

**Additional dependencies included with *WG-FAST***.

1. Trimmomatic. Included with *WG-FAST*. Can also be obtained from: <http://www.usadellab.org/cms/?page=trimmomatic>
2. Picard tools – tested version is 1.79. Included in “binary” directory. Can also be obtained from: <http://broadinstitute.github.io/picard/>
3. RAxML – tested version is 8.1.7. Must be in PATH as “raxmlHPC-SSE3” and “raxmlHPC-PTHREADS-SSE3”, if the sub-sample routine will be used. Can be obtained from: <https://github.com/stamatak/standard-RAxML>. The PTHREADS version does not support the ASC substitution models. Because RAxML is under constant development, a stable version is included with *WG-FAST* (see build directions in the standard-RAxML directory (“README”). As improvements are made to RAxML, the version will be updated in *WG-FAST*, as long as changes don’t affect performance.
4. DendroPy – tested version is 3.12.0, must be installed in PYTHONPATH. Can be obtained from: <https://github.com/jeetsukumaran/DendroPy>. Dendropy is also included with *WG-FAST* with the following included information:

**Copyright 2009-2010 Jeet Sukumaran and Mark T. Holder**

**All rights reserved.**

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JEET SUKUMARAN OR MARK T. HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**Required input files**

1. Directory of sequence reads. The reads must be named according to Illumina HiSeq or MiSeq conventions. Reads must be in the Illumina 1.9+ FastQ format. If you have old Illumina FastQ encodings, they must be converted before running *WG-FAST*. **Important: names must not have periods “.” in the header.**
2. SNP matrix. The easiest way to generate this is by using NASP (<https://github.com/TGenNorth/NASP>). If other SNP matrix formats are used, they must conform to having the first column including (contig::coordinate) and the column following the SNP calls must be (#SNPcall). For the sub-sampling routine to complete, a genome must be present in your dataset that is called ‘Reference’. **Important: sample names must not have periods “.” in the header.**
3. Phylogeny. A script is included with *WG-FAST* that can generate an appropriate phylogeny from a NASP matrix (see below). This script also generates a ‘Parameters’ file, which can be used with *WG-FAST* and cuts down on the computational time required for each subsequent run.
4. Reference genome in FASTA format. This should be the same FASTA that was used to call SNPs with NASP.

**Complete list of Arguments:**

-h, --help show this help message and exit

-m MATRIX, path to NASP-formatted SNP matrix [REQUIRED]

-t TREE, path to input tree in Newick format [REQUIRED]

-r REFERENCE, path to reference fasta [REQUIRED]

-d DIRECTORY, path to directory of fastq.gz files [REQUIRED]

-x PARAMETERS, parameters file for RAxML insertion [OPTIONAL]

-p PROCESSORS, # of processors to use - defaults to 2

-c COVERAGE, minimum SNP coverage required to be corrected, default = 3x

-o PROPORTION, proportion of alleles to be corrected, defaults = 0.9

-k KEEP, keep temporary files? Defaults to F (T or F)

-s SUBSAMPLE, run subsample routine? Defaults to T (T or F)

-n SUBNUMS, number of subsamples to process, defaults to 100

-g DOC, run depth of coverage on all files? Defaults to T (T or F)

-e TMP\_DIR, temporary directory for GATK, default = “/tmp” (Must be PATH)

-z insertion method (MP or ML), defaults to ML, MP not well tested

-f How close does a subsample have to be from true placement? Defaults to 0.1 (Float)

-y Only run sub-sample routine and exit? Defaults to F (T or F)

-j MODEL, which model to run with raxml (GTRGAMMA, ASC\_GTRGAMMA)

**Test Data:**

1. To give *WG-FAST* a run, to make sure everything is installed correctly, check out the test\_data directory.
2. The following command was run from a “run” directory created within the *WG-FAST* installation:

python ../wgfast.py -m ../test\_data/nasp\_matrix.tsv –t ../test\_data/nasp\_raxml.tree \ -r ../test\_data/reference.fasta -d ../test\_data/reads -x ../test\_data/nasp.PARAMS \ -c 1

**Output printed to screen**

1. Number of callable positions. These are all positions called in each sample, compared to the reference. This is the number prior to any filtering due to mixed SNP positions.
2. Number of SNPs. These are the number of observed polymorphisms, based on calls made by GATK.
3. Number of discarded SNPs. These are polymorphisms that were called by GATK, but were thrown out because they failed to meet the depth and/or proportion filters.
4. Insertion likelihood values. The higher the likelihood value and the fewer the number of possible insertion nodes, the more trusted the placement, although caveats exist (see Manuscript).
5. If sub-sample routine is invoked, information is also available for how often the sub-sample was placed correctly. A sub-sample is considered to be “correct” by comparing the patristic distance from the sub-sampled genome to the “Reference”, then comparing that to the distance between the un-sub-sampled genome to the “Reference”. If this ratio falls within the “fudge-factor” range, then it is considered to be correct. The number of times that the sub-sample falls within this range is divided by the total number of iterations and a p-value is reported.

Other scripts:

1. wgfast\_prep.py

-What does it do? Given a NASP matrix, this script will generate a maximum likelihood phylogeny with RAxML and will also generate a “parameters” file that can be used for future *WG-FAST* runs.

-What do you need for the script to run? Requirements include:

* Python > 2.7 < 3.0
* NASP matrix
* RAxML in your $PATH as “raxmlHPC-SSE3”

-What does the output look like? Two files are produced:

1. “nasp\_raxml.tree”. Your tree. Names have been fixed to work with WG-FAST
2. “nasp.PARAMS”. Parameters file. Use this with the “-X” flag described above.

$python wgfast\_prep.py -m nasp.matrix

1. subsample\_snps\_pearson.py

-What does it do? Given a NASP matrix, the script generates a new matrix over a given number of iterations at a given level of SNP sampling.

-What do you need for the script to run? Requirements include:

* Python > 2.7 < 3.0
* NASP matrix
* ‘mothur’ executable in your PATH. Mothur can be freely obtained from: <http://www.mothur.org/wiki/Download_mothur>

-What does the output look like? One file is generated “results.txt”, that is new-line delimited, with each line containing a Pearson correlation value (0 to 1).

$python subsample\_snps\_pearson.py -m nasp.matrix -s 100

3. Subsample\_

**Citations for all dependencies.**

**BioPython**: Cock, P.J. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422-3 (2009).

**Trimmomatic**: Bolger, A.M., Lohse, M. & Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30**, 2114-20 (2014).

**BWA-MEM**: Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXivorg. 2013(arXiv:1303.3997 [q-bio.GN])

**RAxML v8**: Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* (2014).

**RAxML evolutionary placement algorithm**: Berger SA, Krompass D, Stamatakis A. Performance, accuracy, and Web server for evolutionary placement of short sequence reads under maximum likelihood. Syst Biol. 2011;60(3):291-302.

**Samtools:** Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25(16):2078-9.

**DendroPy:** Sukumaran J, Holder MT. DendroPy: a Python library for phylogenetic computing. Bioinformatics. 2010;26(12):1569-71. Epub 2010/04/28. doi: 10.1093/bioinformatics/btq228. PubMed PMID: 20421198.

**GATK:** McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, DePristo MA. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome research. 2010;20(9):1297-303.