diversity

MDPI

## Supplementary Materials

# Genetic Diversity of Northern Wheatgrass (*Elymus lanceolatus* ssp. *lanceolatus*) as Revealed by Genotyping-by-Sequencing

**These supplementary materials are divided into the following two sections:**

**Section A: List of Supplementary Materials (three files and five zip folders) and they are available online** (DOI://10.6084/m9.figshare.6057491)

A1. Table S1: List of 144 samples and sequencing information (Excel file)

A2. NWG_CbyT-207-50_SNP.txt (Text file for SNP data at 50% missing level)

A3. Explanations for Haplotag output files (pdf file)

A4. Four zip folders for Haplotag output files
(mergedAll-ABC.zip; HTML-A.zip; HTML-B.zip HTML-C.zip)

A5. One zip folder of 13 files for all the custom shell and perl scripts and related files

**Section B: Detailed procedure for analyzing FASTQ files using UNEAK and Haplotag to generate tag-level SNP data**

### 1. GBS data

This study assayed 144 individual northern wheatgrass plants of 12 accessions (Table 1 and Table S1). Total genomic DNAs were extracted and digested with HinfI and HpyCH4IV using the gd-GBS method (Peterson et al. 2014). Three MiSeq sequencing runs of 144 samples of 12 accessions, each using a 600 base version 3 kit, generated a total of 288 forward (R1) and reverse (R2) FASTQ sequence files (Table S1). All the raw pair-end sequencing data in FASTQ format were deposited into NCBI Sequence Read Archive (SRA) under BioProject ID PRJNA392957. The FASTQ data was trimmed with Trimmomatic (Bodger et al. 2014) to remove any sequenced-through Illumina adapters, low-quality sequence (sliding window of 10 bases, average Phred of 20), and fragments under 64 bases long.

Note that the bioinformatics and genetic diversity analysis for this study were made only on 119 samples from the 10 accessions listed in Table 1, after excluding the 24 samples for the accession TMP24017 of a different species and for the accession TMP24008 with octoploidy reading and one failed sample from TMP24016. Only forward (R1) sequence reads were used in the analysis.

### 2. Fragmenting FASTQ data

The custom Perl script fastq184CutandCode.pl was used to divide the input sequence into three parts: the first 64 bases containing the *Hin*fI residual restriction site and the next two 60-base portions. Only sequences with an intact *Hin*fI site at the beginning were considered. UNEAK is hard-coded to only recognize fragments with a barcode and a one of several restriction enzymes, (Lu et al. 2012) thus some modifications to the input sequence were required. The script replaced

the *Hin*fI residual restriction site (ANTC) with a *Sau*3AI residual restriction site (GATC), added a *Sau*3AI sequence to the second and third fragments, and added a pseudo barcode sequence (CATCAT) in front all three sequence fragments. The resulting three fragments were each 70 bases long and contained a barcode and a restriction site that would be recognized by the UNEAK software. After that, fragments had to meet the full-length requirement of each fragment to be processed. The relationship between the three fragments was not preserved going into UNEAK, and each fragment set was passed into UNEAK as an independent data set. In this analysis, we only examined the forward sequence (R1) reads.

**3. Run Fragmented FASTQ Data Using UNEAK**

UNEAK, available from https://tassel.bitbucket.io/TasselArchived.html (Accessed: 2018/03/13), (from the Tassel v 3.0 GBS pipeline (Glaubitz et al. 2013); Tassel v 3.0 UNEAK pipeline (Lu et al. 2012)) was executed with the following conditions:

- plugin -UFastqToTagCountPlugin to identify the Sau3AI enzyme (-e);
- plugin -UMergeTaxaTagCountPlugin set to collect a maximum of 250-million tags per tagCount file (-m) and each tag requiring a minimum of 10 reads (-c);
- plugin -BinaryToTextPlugin to convert the mergedAll and individual sample tagCounts from the UNEAK binary format to text format.

Once processed it was observed that UNEAK truncates fragments at a 3′ *Sau*3AI restriction site, thus not all tags passed onto Haplotag were 64 bases in length. Additionally, removing *Hin*fI with its variable second base may have created artificially combined tags. However, because *Hin*fI was not on the list of allowed enzymes, a substitution had to be made in order to run the software. The resulting mergedAll.txt and individual tagCount files were passed to the Haplotag software (Tinker et al. 2016).

**4. Analyze Fragmented FASTQ Data Using Haplotag**

Haplotag, available from http://haplotag.aowc.ca/ (Accessed: 2018/03/13), was run for each of the three sets of tagCount files with the following conditions in the HTindex.txt file:

- @DiploidSNPGenos, true, {show homozygotes as diploids (e.g., AA)}
- @Verbose, true, {set true for detailed model selection in HTML reports}
- @ThreePlus, false, {set true to limit HTML reporting to models with >2 haplotypes}
- @reportallpp, true, {report passports for cluster even if there is no model (default=false) }
- @MaxThreads, 99, {use 999 for maximum possible, 99 for Maximum minus 1}
- @MinTagCount, 10, {set high to inspect only deep-sequenced tags}
- @MaxBaseDif, 3, {Maximum number of base mismatches to join tags in a cluster}
- @MinPres, 0.015, {minimum minor allele frequency}
- @MaxPres, 0.99, {maximum major allele frequency}
- @MaxQ, 300000000, {maximum total tags to inspect - for low memory, testing, etc.}
- @MaxS, 100000000, {maximum total tag clusters to inspect}
- @MaxTagsToTest, 9, {maximum tags in a cluster, clusters with more are ignored}
- @RSite, HinfI, {restriction site HinfI, ApeKI, PstI-MspI}

- @ThreshGeno, 0.4, {Threshold for minimum complete genotypes (% of taxa) when selecting a model}
- @ThreshHet, 0.1, {Threshold heterozygote frequency}
- @ThreshMAHet, 0.4, {Max. het. freq. by allele., Excludes high-het rare allele., Set to 1 for bi-parentals}
- @ThreshTrihet, 0, {Threshold trizygote frequency (3 haplotypes in one taxon)}
- @ThreshMultiHet 0, {Threshold multizygote frequency (4 or more haplotypes in one taxon)}
- @HetRatio, 0.1, {threshold ratio of tag count for minor allele - below this ratio call a homozygote}

and the following program steps:

- !ReadTaxaIDFile, .\HTTaxa.txt
- !ClusterMergedAll, .\mergedAll.txt, Build HTclusters and HThaplos directly from UNEAK mergedall file
- !ReadClusters, .\output\HTClusters.txt, You need to read clusters after they are built.,
- !ReadHaplotypes, .\output\HTHaplos.txt
- !MakeTagByTaxa, .\tagCounts-txt\, Tagcounts from UNEAK,
- !IdentifyAlleles, build the models and report the genotypes and passports

Following this, the resulting SNP calls (contained in the 'HTSNPGenos files') were concatenated into a single set of SNP calls. The header row was removed from the second and third files prior to concatenation. Additional SNP filtering was then performed on this concatenated file using the in-house Character by Taxa (CbyT) program provided by N. Tinker. CbyT was run with three different filtering levels for minimum presence: 80% (representing 20% missing data), 70% (representing 30% missing data), 60% (representing 40% missing data), and 50% (representing 50% missing data). The CbyT output file generated from the Haplotag HTSNPGeno file was used as the final SNP data file. Diploid SNP calls were converted to haploid format by replacing all homozygous diploid values with the haploid equivalent using the Linux sed command: i.e.

```
sed 's/AA/A/g' all-HTSNPGenos.txt | sed 's/GG/G/g' | sed 's/CC/C/g' | sed
's/TT/T/g'| sed 's/--/-/g'> all-HTSNPGenos-singles.txt
```

Sample CbyT.bat batch file for 20% missing data:

```
set SNPRAW=.\all-HTSNPGenos-singles.txt
set HAPRAW=.\all-HTGenos.txt

CbyT %SNPRAW% httaxa.txt null ALL-HTSNPGenos_SNP.txt 7 0 10 1 80
CbyT %HAPRAW% httaxa.txt null ALL-HTGenos_HAP.txt 6 0 10 1 80

pause
```

This example of the CbyT.bat file is set for:

- `7` = HTSNPGenos input data (@DiploidSNPGenos = false) or `6` = HTGenos input data
- `0` = diversity data,
- `10` = Max Het% (maximum heterozygosity as a percent),

- `1` = Min MAF % (minor allele frequency as a percent),
- `80` = Min Pres (minimum completeness score as a percent (i.e., the reverse of "missing data"))

**5. Supportive Analysis**

5.1 UNEAK key.txt files and Haplotag HTTaxa.txt and HTinput.txt files were prepared using a combination of Notepad++ and MS Excel.

5.2 Supportive Perl and shell scripts were specifically written and used to assist in the preparation of the various files required to run UNEAK and Haplotag and run from the Windows command (cmd) terminal or Cygwin (mintty 1.2-beta1 (x86_64-pc-cygwin; 2013) terminal:

- *fastq184CutandCode.pl* was used to prepare the input MiSeq fastq files for UNEAK,
- *tagCountTXTmaker.sh* was used to make the batch file to convert tagCount binary files to text files
- *tagCountTXTbatch.bat* converted the tagCount binary files to text files.
- *autotrimNWG-SE.sh* to run Trimmomatic over the batch of all FASTQ R1 files.

The above four text files and the UNEAK and Haplotag batch and info files are available as online supplementary information as described in Section A.

5.3 UNEAK and Haplotag were run using Microsoft Windows 7 64-bit OS, using an Intel Core i7-3930K.

**6. References**

Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30: 2114–2120

Glaubitz J, Elshire R, Casstevens T, Harriman J, Buckler E (2013) TASSEL 3 Genotyping by Sequencing (GBS) pipeline documentation. https://bytebucket.org/tasseladmin/tassel-5-source/wiki/docs/TasselPipelineGBS.pdf (Accessed: 2013/03/13)

Lu F, Glaubitz J, Harriman J, Casstevens T, Elshire R (2012) TASSEL 3.0 Universal Network Enabled Analysis Kit (UNEAK) pipeline documentation. https://bytebucket.org/tasseladmin/tassel-5-source/wiki/docs/TasselPipelineUNEAK.pdf (Accessed: 2018/03/13)

Lu F, Lipka AE, Glaubitz J, Elshire R, Cherney JH, Casler MD, Buckler ES, Costich DE (2013) Switchgrass genomic diversity, ploidy, and evolution: novel insights from a network-based SNP discovery protocol. PLoS Genet 9(1): e1003215

Peterson GW, Dong YB, Horbach C, Fu YB (2014) Genotyping-by-sequencing for plant genetic diversity analysis: a lab guide for SNP genotyping. Diversity 6:665–680

Tinker NA, Bekele WA, Hattori J (2016) Haplotag: software for haplotype-based genotyping-by-sequencing analysis. G3 (Bethesda) 6:857-863