

Automated Testing TASSEL with JUnit – Getting Started

Terry Casstevens

Institute for Genomic Diversity, Cornell University

June 13, 2013

Objectives

- Exercise Core Functions
- Complex Analyzes are Hard to Verify Manually
- Regression Testing

TASSEL Test Repository

- Read Only: `git clone git://git.code.sf.net/p/tassel/maizegenetics4-test tasselmazegenetics4-test`
- Read / Write: `git clone ssh://<userid>@git.code.sf.net/p/tassel/maizegenetics4-test tasselmazegenetics4-test`

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window

<default conf... [Tools]


Projects Files Services ...java Utils.java TagCountToFastqPlugin.java KeepSpecifiedReads

tassel4

- New
- Build
- Clean and Build
- Clean
- Generate Javadoc
- Run
- Debug
- Profile
- Test ^{^F6}
- Set Configuration
- Open Required Projects
- Close
- Rename...
- Move...
- Copy...
- Delete ^{Delete}
- Find... ^{⌘F}
- Inspect and Transform...
- Git
- History
- Properties**
- net.maizegenetics.pal.report
- net.maizegenetics.pal.statistics

Source History [Navigation]

```
76     */
77     Depth,
78     /**
79     * This uses the allele frequency of a bas
80     * sort order of alleles. That Alignment i
81     */
82     Global_Frequency,
83     /**
84     * This sorts alleles based on the referen
85     */
86     Reference
87 };
88
89 /**
90 * Returns diploid value (genotype) for a give
91 *
92 * @param taxon taxon
93 * @param site site
94 *
95 * @return high four bits generally encode the
96 * lower four bits encode the less frequent al
97 */
98 public byte getBase(int taxon, int site);
99
100 /**
101 * Returns diploid values for given taxon and
102 * getBase(), except two values are already se
103 *
104 * @param taxon taxon
```



Project Properties - tassell4

Categories:

- Sources
- Libraries
- ▼ ○ Build
 - Compiling
 - Packaging
 - Documenting
- Run
- ▼ ○ Application
 - Web Start
 - Formatting

Project Folder: /Users/terry/NetBeansProjects/tassel4

Source Package Folders:

Package Folder	Label
/Users/terry/terry/maizegenetics4/src	Source Packages

- Add Folder...
- Remove
- Move Up
- Move Down

Test Package Folders:

Package Folder	Label
/Users/terry/terry/maizegenetics4_test/src	Test Packages
/Users/terry/terry/maizegenetics4/test	/Users/terry/terry/maizegenetics4/test

- Add Folder...
- Remove
- Move Up
- Move Down

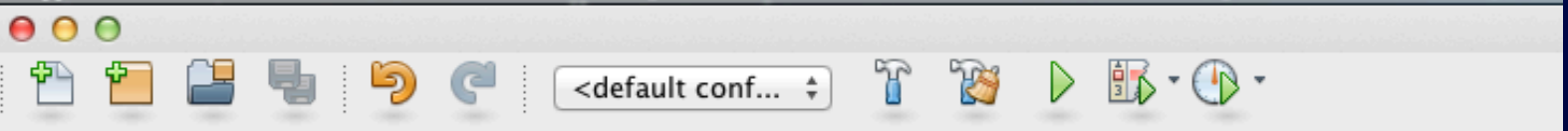
Source/Binary Format: JDK 6

Includes/Excludes...

Encoding: UTF-8

- Help
- OK
- Cancel



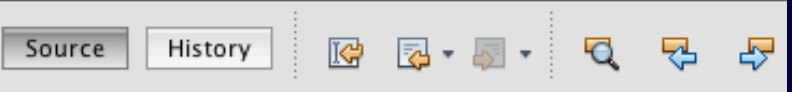


Projects Files Services

- tassel3
- tassel4
 - Source Packages
 - Test Packages
 - net.maizegenetics.baseplugins
 - DistanceMatrixPluginTest.java
 - net.maizegenetics.constants
 - GeneralConstants.java
 - TutorialConstants.java
 - net.maizegenetics.gbs.maps
 - net.maizegenetics.gbs.pipeline
 - net.maizegenetics.pal.alignment
 - AlignmentScopeTest.java
 - AlignmentTestingUtils.java
 - AlignmentUtilsTest.java
 - BitAlignmentHDF5Test.java
 - ExportUtilsTest.java
 - FilterAlignmentTest.java
 - MarkerPhenotypeAdapterTest.java
 - MutableNucleotideAlignmentHDF5Test.java
 - MutableNucleotideAlignmentTest.java
 - net.maizegenetics.pal.popgen
 - net.maizegenetics.pal.report
 - net.maizegenetics.stats.GLM
 - net.maizegenetics.testsuites
 - net.maizegenetics.util
 - /Users/terry/terry/maizegenetics4/test
 - Libraries
 - Test Libraries



...java Utils.java TagCountToFastqPlugin.java



```
76      */
77      Depth,
78      /**
79      * This uses the allele frequ
80      * sort order of alleles. That
81      */
82      Global_Frequency,
83      /**
84      * This sorts alleles based on
85      */
86      Reference
87  };
88
89  /**
90  * Returns diploid value (genotype
91  *
92  * @param taxon taxon
93  * @param site site
94  *
95  * @return high four bits generall
96  * lower four bits encode the less
97  */
98  public byte getBase(int taxon, int
99
100  /**
101  * Returns diploid values for give
102  * getBase(), except two values ar
103  *
104  * @param taxon taxon
105  * @param site site
```

AlignmentMaskGeneticDistance.java 88
AlignmentMaskReference.java 89
AlignmentUtils.java 90
BitAlignment.java 91
BitAlignment 92
BitNucleotide 93
BitNucleotide 94
BitPhasedAlignment 95
BitPhasedNucleotide 96
BitPhasedTe 97
BitTextAlignment 99
BitTextAlignment 100
CombineAlign 101
CombinePhen 102
DataTable.j 103

Members

- BitAlignment :: Abstract
- BitAlignment(IdGroup)
- BitAlignment(IdGroup)
- BitAlignment(Alignment)
- getAllelePresenceForTaxon(int taxon, int site): boolean
- getAllelePresenceForTaxon(int taxon, int site): boolean
- getAllelesSortedByAlleleFrequency(int taxon): List<Allele>
- getBase(int taxon, int site): byte
- getBaseArray(int taxon, int site): byte[]
- getBaseArraySBit(int taxon, int site): byte
- getBaseArrayTBit(int taxon, int site): byte
- getDiploidCounts(): Map<Allele, Integer>
- getDiploidssSortedByAlleleFrequency(int taxon): List<Allele>
- getHeterozygousCountForTaxon(int taxon): int
- getHeterozygousCountForTaxon(int taxon): int
- getHomozygousNucleotideInstance(Alignment a, Allele allele): Nucleotide
- getInstance(IdGroup idGroup, String[] data, Genotype genotype): Genotype
- getInstance(IdGroup idGroup, byte[][] data, Genotype genotype): Genotype

Open
Cut ⌘X
Copy ⌘C
Paste ⌘V
Compile File F9
Run File ⇧F6
Debug File ⇧⌘F5
Profile File
Test File ⌘F6
Debug Test File ⇧⌘F6
Profile Test File
Add
Delete Delete
Save As Template...
Find Usages ^F7
Refactor ▶
BeanInfo Editor...
File Members ⌘F12
File Hierarchy ^F12
History ▶
Git ▶
Tools ▶
Properties

```
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112
```

```
/**  
 * Returns diploid value (genotype) for a given taxon and site.  
 *  
 * @param taxon taxon  
 * @param site site  
 *  
 * @return high four bits generally encode the first allele  
 * lower four bits encode the less frequent allele  
 */  
public byte getBase(int taxon, int site);  
  
/**  
 * Returns diploid values for given taxon and site.  
 * getBase(), except two values are already separated.  
 *  
 * @param taxon taxon  
 * @param site site  
 *  
 * @return first byte (index 0) holds first allele  
 * bits. second byte (index 1) holds second allele  
 * bits.  
 */  
public byte[] getBaseArray(int taxon, int site)
```

find: calculateBitLDForInbred

Alignment >

Usages Output - tassell4 (compile-single) x

```
ant -f /Users/terry/NetBeansProjects/tassel4 -Djavac.in  
init:  
Deleting: /Users/terry/NetBeansProjects/tassel4/build/b  
file: /Users/terry/NetBeansProjects/t  
file: /Users/terry/NetBeansProject  
trap class path not set in conj  
total time: 0 seconds)
```

Apply Diff Patch...
Diff To...
Add to Favorites
Show In Finder
Create Tests
Analyze Javadoc
Add to Palette...



Class to Test: net.maizegenetics.pal.alignment.BitAlignment

Class Name:

Location:

Framework:

Code Generation

Method Access Levels	Generated Code
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Test Initializer
<input type="checkbox"/> Protected	<input checked="" type="checkbox"/> Test Finalizer
<input type="checkbox"/> Package Private	<input checked="" type="checkbox"/> Test Class Initializer
	<input checked="" type="checkbox"/> Test Class Finalizer
	<input checked="" type="checkbox"/> Default Method Bodies

Generated Comments

- Javadoc Comments
- Source Code Hints

Help OK Cancel


```
@Test
public void testHDF5Alignment() throws IOException {

    System.out.println("Testing HDF5 Alignment...");

    Alignment genoAlign = ImportUtils.readFromHapmap(TutorialConstants.HAPMAP_FILENAME, null);
    Alignment hdf5Align = BitAlignmentHDF5.getInstance(TutorialConstants.HDF5_FILENAME);

    // Compare Alignment from Hapmap and Alignment from HDF5
    AlignmentTestingUtils.alignmentsEqual(genoAlign, hdf5Align);
    genoAlign.optimizeForTaxa(null);
    AlignmentTestingUtils.alignmentsTaxaBitSetsEqual(genoAlign, hdf5Align);
    AlignmentTestingUtils.alignmentsSiteBitSetsEqual(genoAlign, hdf5Align);

    ExportUtils.writeToHDF5(genoAlign, GeneralConstants.TEMP_DIR + "testHDF5Align.hmp.h5");
    Alignment hdf5Align2 = BitAlignmentHDF5.getInstance(GeneralConstants.TEMP_DIR + "testHDF5Align.hmp.h5");

    // Compare Alignment from export to HDF5 and Alignment from existing HDF5
    AlignmentTestingUtils.alignmentsEqual(hdf5Align2, hdf5Align);
    AlignmentTestingUtils.alignmentsTaxaBitSetsEqual(hdf5Align2, hdf5Align);
    AlignmentTestingUtils.alignmentsSiteBitSetsEqual(hdf5Align2, hdf5Align);

}
```



```

public static void alignmentsEqual(Alignment align1, Alignment align2) {

    int siteCount = align1.getSiteCount();
    assertEquals("Expected Site Count: " + siteCount + " Actual: " + align2.getSiteCount(), siteCount, align2.getSiteCount());

    int taxaCount = align1.getSequenceCount();
    assertEquals("Expected Taxa Count: " + taxaCount + " Actual: " + align2.getSequenceCount(), taxaCount, align2.getSequenceCount());

    assertEquals("Max Number of Alleles: ", align1.getMaxNumAlleles(), align2.getMaxNumAlleles());

    int numAlleles = align1.getTotalNumAlleles();
    assertEquals("Total Number of Alleles: ", numAlleles, align2.getTotalNumAlleles());

    int[] physicalPositions = align1.getPhysicalPositions();
    assertEquals("Number of Physical Positions: ", siteCount, physicalPositions.length);

    String[] siteNames = align1.getSNPIDs();
    assertEquals("Number of Site Names: ", siteCount, siteNames.length);

    for (int s = 0; s < siteCount; s++) {

        assertEquals("Physical Position at Site: " + s, align1.getPositionInLocus(s), align2.getPositionInLocus(s));
        assertEquals("SNP ID at Site: " + s, align1.getSNPID(s), align2.getSNPID(s));
        Locus locus1 = align1.getLocus(s);
        Locus locus2 = align2.getLocus(s);
        assertEquals("Locus at Site: " + s, locus1.getName(), locus2.getName());
        assertEquals("Site at Physical Position: " + s, align1.getSiteOfPhysicalPosition(physicalPositions[s], locus1), align2.getSiteOfPhysicalPosition(physicalPositions[s], locus2));
        assertEquals("Site at Physical Position with SNP ID: " + s, align1.getSiteOfPhysicalPosition(physicalPositions[s], locus1, siteNames[s]), align2.getSiteOfPhysicalPosition(physicalPositions[s], locus2, siteNames[s]));
        assertEquals("Major Allele at Site: " + s, align1.getMajorAllele(s), align2.getMajorAllele(s));
        assertEquals("Minor Allele at Site: " + s, align1.getMinorAllele(s), align2.getMinorAllele(s));
        assertEquals("Major Allele Freq. at Site: " + s, align1.getMajorAlleleFrequency(s), align2.getMajorAlleleFrequency(s), 0.0);
        assertEquals("Minor Allele Freq. at Site: " + s, align1.getMinorAlleleFrequency(s), align2.getMinorAlleleFrequency(s), 0.0);
        assertEquals("Major Allele Count at Site: " + s, align1.getMajorAlleleCount(s), align2.getMajorAlleleCount(s), 0.0);
        assertEquals("Minor Allele Count at Site: " + s, align1.getMinorAlleleCount(s), align2.getMinorAlleleCount(s), 0.0);
        assertEquals("Heterozygous Count at Site: " + s, align1.getHeterozygousCount(s), align2.getHeterozygousCount(s));

        // getAllelesSortedByFrequency()
        int[][] align1SortedFreq = align1.getAllelesSortedByFrequency(s);
        int[][] align2SortedFreq = align2.getAllelesSortedByFrequency(s);
        assertEquals("getAllelesSortedByFrequency() length differ at Site: " + s, align1SortedFreq[0].length, align2SortedFreq[0].length);
        for (int i = 0, n = align1SortedFreq[0].length; i < n; i++) {
            assertEquals("getAllelesSortedByFrequency() values differ at Site: " + s, align1SortedFreq[0][i], align2SortedFreq[0][i]);
        }
    }
}

```

- ▶ build
- ▶ build.xml
- ▶ common
- ▼ dataFiles
 - ▶ ExpectedResults
 - ▶ MarkerPhenotypeAdapterTest
 - ▶ Tutorial
- ▼ dist
 - ▶ sTasselTest.jar
- ▶ lib
 - ▶ run_basic_suite.pl
 - ▶ run_pipeline_non_verbose.pl
- ▶ src
- ▼ tempDir
 - ▶ ExportUtilsTest
 - ▶ testFilterToHDF5_2.hmp.h5
 - ▶ testFilterToHDF5.hmp.h5
 - ▶ testHDF5Align.hmp.h5
 - ▶ update_tag.pl

- Test Packages
 - net.maizegenetics.baseplugins
 - DistanceMatrixPluginTest.java
 - net.maizegenetics.constants
 - GeneralConstants.java
 - TutorialConstants.java
 - net.maizegenetics.gbs.maps
 - TagsOnPhysMapHDF5Test.java
 - net.maizegenetics.gbs.pipeline
 - MinorWindowViterbilImputationPluginTest.java
 - net.maizegenetics.pal.alignment
 - AlignmentScopeTest.java
 - AlignmentTestingUtils.java
 - AlignmentUtilsTest.java
 - BitAlignmentHDF5Test.java
 - ExportUtilsTest.java
 - FilterAlignmentTest.java
 - MarkerPhenotypeAdapterTest.java
 - MutableNucleotideAlignmentHDF5Test.java
 - MutableNucleotideAlignmentTest.java
 - net.maizegenetics.pal.popgen
 - LinkageDisequilibriumTest.java
 - net.maizegenetics.pal.report
 - TableReportTestUtils.java
 - net.maizegenetics.stats.GLM
 - GLMTest.java
 - net.maizegenetics.testutils
 - BasicTassel.java
 - PrintResults.java
 - net.maizegenetics.util
 - BitSetTest.java