8-2013

A Review of Apomixis and Differential Expression Analyses Using Microarrays

Jonathan Harris Cardwell
Utah State University

A REVIEW OF APOMIXIS AND DIFFERENTIAL EXPRESSION ANALYSES USING MICROARRAYS

By

Jonathan Cardwell

A Plan B paper submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Plant Science

Approved:

_____	_____
John G. Carman Major Professor	Paul Johnson Committee Member
_____	
John Stevens Committee Member	

UTAH STATE UNIVERSITY
Logan, Utah

2013

ACKNOWLEDGEMENTS

I would like to thank my wife, Katy, for all of her support and encouragement throughout my degree. Without her, I would not have ever finished.

Thanks to my committee members, Drs. John Stevens and Paul Johnson for their assistance and willingness to help, and special thanks to my major professor, Dr. John Carman, for his constant motivation to help me to finally complete my degree.

Thanks also to Gao Lei, Mayelyn Bautista, and David Sherwood, for all of their help the lab, which was integral to my experience at USU.

Jonathan Cardwell

CONTENTS

Abstract

Apomixis is a complex trait of great interest to the agricultural community, as it has the potential to fix hybrid vigor in many agriculturally significant crops. Although apomixis has been studied extensively morphologically, the genetic and epigenetic factors responsible for apomixis are still very poorly understood. As no apomictic species has been sequenced and annotated, various low-cost tools and techniques are being utilized to begin profiling the trait. These include cross-species microarrays using probe masking, which deletes information from array probes that do not hybridize to the genomic DNA of the cross species. Despite their limitations, these tools are providing a strong informatics foundation for which future, more robust profiling procedures can be conducted.

Introduction

Apomixis

Apomixis in flowering plants is a process in which viable seeds develop from maternal cells that are not products of genetic recombination and the fusion of male and female gametes (Ozias-Akins & van Dijk, 2007). Apomixis occurs in all major clades of eukaryotes where sex predominates (Carman et al., 2011). There are two main phases of apomixis in eukaryotes. The first is apomeiosis, which is the formation of a genetically unreduced, parthenogenetically competent egg from cells that would normally undergo female meiosis, or from cells closely associated with those that undergo female meiosis. The second major phase is parthenogenesis, which is embryo formation without fertilization.

Apomixis is of major interest because conferring the apomixis phenotype to domesticated plants, especially to agriculturally significant hybrids, would revolutionize global agriculture (Spillane et al., 2004). Apomixis in the agriculture industry is predicted to slash hybrid seed production costs by 80%, and increase yield of existing inbred crops, by converting them to high yielding hybrids, by 15-30% (Carman, 2004; Ozias-Akins & van Dijk, 2007).

Apomixis gives rise to viable embryos based completely on the maternal genome and appears visually to be quite similar to the normal sexual pathway with some important differences in how the final embryo is formed (Tucker & Koltunow, 2009). Apomixis in angiosperms is generally divided into three

categories, apospory, diplospory, and adventitious embryony. These categories are based on the method the species utilizes to form the embryo.

Types of Apomixis

In apospory, the megaspore mother cell (MMC) generally undergoes a normal meiosis as per usual in the sexual pathway. However, all products of this meiosis degenerate. Concurrently, one or more somatic cells in the surrounding nucellus produce a functional 2*n* embryo sac (megagametophyte) that contains a parthenogenetically competent 2*n* egg, which is genetically identical to the mother plant. Diplospory, on the other hand, involves the production of a functional 2*n* embryo sac from an ameiotic MMC. This may occur directly from the MMC, a process called *Antennaria* type diplospory, or after a modified meiosis that results in two unreduced spores that are genetically identical to the mother plant. In the latter case, the embryo sac forms from one of the two 2*n* megaspores and the other one degenerates. This is called *Taraxacum* type diplospory and is the form that occurs in the common dandelion. Mature genetically unreduced embryo sacs of aposporous or diplosporous species are either 4-nucleate, with an egg, two synergids and a single polar body, or 8-nucleate, with an egg, two synergids, two polar bodies (central cell), and three antipodals, depending on the species. Fertilization of the egg is not required for embryo development in apomictic plants. However, fertilization of the single polar body, in species that produce a 4-nucleate embryo sac, or of the central cell, in species that produce an 8-nucleate embryo sac, is often required for

endosperm development, a condition called pseudogamy. Without endosperm development, parthenogenesis usually aborts. However, some apomicts produce both embryos and endosperm without fertilization. This is referred to as autonomous apomixis (Carman, 1997; Tucker & Koltunow, 2009).

The genetic and molecular control of apomixis remains largely an enigma (Schranz et al., 2006; Tucker & Koltunow, 2009), although it is generally thought that apomixis has repeatedly arisen across time and space from sexual progenitors in multiple genera, families and kingdoms (Carman, 1997; Hörandl, 2009; Lampert, 2008). The pursuit to elucidate the functional control of apomixis has been a crop scientist goal for the past 100 years but with little or no success (Asker & Jerling, 1992). As a result, there are at least three theories in the current literature that attempt to explain apomixis: 1) it is a mutation-based anomaly involving a simple or possibly complex genetic locus (Ozias-Akins & van Dijk, 2007), 2) it is caused by developmental asynchronies that result from hybridization or polyploidization (Carman, 1997), and 3) it is an ancient epigenetically regulated alternative to sexual reproduction with remnant capacities more or less conserved across eukaryotes (Carman et al., 2011).

The *Het* Chromosome

A traditional approach to the apomixis question has been the idea that apomixis is controlled by a single locus, such as the *Het* chromosome (Ozias-Akins & van Dijk, 2007). These *Het* chromosomes, or heterochromatic chromosomes, are intriguing because of their presence in *Boechera* and are

reminiscent of apomixis-conferring chromosomal regions found in some grasses (Ozias-Akins & van Dijk, 2007). However, when Schranz et al. crossed 92 different lines of *Boechera*, with successful hybridizations of sexual and apomictic species, weak F1 seed production suggested that apomixis was not transferred in any of these crosses, including those that would have contained the *Het* chromosome (Schranz et al., 2005). Although the *Het* chromosome may be involved in the apomixis phenotype, this is a strong argument against the single locus theory.

Mutation of Sexual Pathway

During most of the last century, apomixis was thought to be a result of the mutation of meiotic/syngamy genes in sexual species (Mogie, 1992; Savidan, 2000). In 1997, Carman proposed that most of the reproductive anomalies observed in angiosperms, such as bispory, tetraspory and apomixis, evolved through hybridization and asynchronous expression of the resulting combined transcriptomes. Apomixis would have thus arisen as a sexual dysfunction created by duplicate genes in hybrids (Carman 1997; 2007). Despite elements of Carman's asynchrony hypothesis having recently been extended to include sequence variation in sRNAs and transposable elements that regulate aspects of sporogenesis, gametogenesis and embryogenesis (Tucker & Koltunow, 2009; Rodrigues et al., 2010; Armenta-Medina et al., 2011), there remains a fundamental weakness to this hypothesis. As addressed by Carman et al., apomictic species do not occur randomly among angiosperms as would be

expected if apomixis is an "anomaly of sex" created *de novo* by hybridization-induced asynchrony or epigenetic miss-communications (Carman et al., 2011).

Ancient Mechanism

A third hypothesis, based partially on preliminary data important to the research proposed herein and the results of a recent study of *Sorghum bicolor* (Carman et al., 2011), suggests that apomixis is actually an ancient mechanism conserved in its molecular components to a greater or lesser extent across eukaryotes. Similar to what has been observed in cyclically apomictic animals, i.e., those that are apomictic during favorable conditions and sexual during stress, e.g., aphids and water fleas (Suomalainen et al., 1987), and facultatively apomictic plants (Knox & Heslop-Harrison, 1963), it is postulated that environmental control and stress signaling (either perceived or actual) could be the trigger that switches reproduction between the sexual and the asexual pathways. This switching mechanism would be under epigenetic control and more or less conserved across eukaryotes (Carman et al., 2011). If this is correct, it could represent a major paradigm shift in understanding the evolutionary relationship between sex and apomixis.

Recent Genetic Profiling of Apomixis

Genetic profiling research relative to apomixis is being studied in many species. Zeng et al. recently used 454 sequencing to analyze differentially expressed genes between *Pennisetum squamulatum* and an apomictic derived

backcross line containing only one foreign chromosome that confers apomixis via a region of the chromosome referred to as the Apospory Specific Genomic Region (ASGR). Interestingly, 6 of the 7 significant GO terms found when comparing all libraries were all related to ribosomal or translational functions (Zeng et al., 2011). Also in the *Pennisetum* genus, Sahu et al. performed a transcriptome analysis of *P. glacum* and its apomeiotic interspecific hybrid (BC1GO) by using suppression subtractive hybridization (Sahu et al., 2011). Although the majority of differentially expressed transcripts (40%) were found to be of unknown function, the next largest significant categories were stress response (11%), metabolism related (10%), and translational functions (8%).

In an expressed sequence tag (EST) and differential display analysis, Cervigni et al. compared sexual and apomictic genotypes of *Eragrostis curvula*, including across ploidy levels, through differential display. Interestingly, in both apomictic vs. sexual comparisons, regardless of ploidy, both groups produced overexpressed genes of ribosomal and translational function in the apomictic groups. Also in the apomicts were metabolism related genes, such as AMP and ATP synthase beta chains (Cervigni et al., 2008).

Laspina et al. investigated differential expression between sexual and apomictic inflorescences in *Paspalum notatum*. Results indicated that apomixis in *Paspalum* could be related to silencing of large genomic regions and altered expression of a signal transduction cascade (Laspina et al., 2008). Ochogavia et al. continued this by investigating expression levels of retrotransposons in sexual

and apomictic *P. notatum* . Two retrotransposons in particular, N17 and N22, were found to be highly expressed in the sexuals compared to the apomicts (Ochogavia et al., 2011).

Further support of epigenetic control of the apomixis phenotype came from another differential expression study, also using differential display, by Polegri et al. Two hundred and two cDNA-AFLP amplicons were generated from sexual and apomictic *Paspalum simplex* flowers, and of all apomixis-linked alleles, the most frequent biological functions were related to signal transduction and were interpreted as epigenetic regulators. This lead the authors to suggest that even if the key genes to trigger apomixis were few, the downstream effects of these apomixis-linked factors could be great (Polegri et al., 2010).

Based on these recent publications, it would be reasonable to assume that epigenetic regulators and signaling play a large role in the apomixis phenotype. In addition, ribosomal genes and translational processes appear to be heavily involved, which would also relate to significant metabolism functions. Future research in the study could benefit from broader expression profile techniques, such as microarrays and next generation sequencing, to provide a larger picture of the differing transcriptomes of sexual and apomictic species. Further work is needed to better challenge the existing hypotheses concerning the evolution of apomixis.

Boechera as a Model Genus

Commonly used species for studying apomixis can be found in the genus, *Boechera*. *Boechera*, until recently, was referred to as *Arabis* (Böcher 1951; Naumova et al., 2001), but was later renamed after the person who first demonstrated apomixis in the genus, Tyge W. Böcher. It is considered a model genus for studying apomixis as it contains obligate to facultative apomicts and plants that are fully sexual. The apomicts appear to be of hybridogenous origins (Sharbel et al., 2009; Beck et al., 2011). *Boechera* is perhaps the only genus in which naturally occurring diploid apomicts exist (Rushworth et al., 2011). Essentially all other naturally occurring apomicts are polyploids (Carman, 1997). This makes it an ideal plant material for conventional genetic analyses and for gene expression comparisons (Sharbel et al., 2010). Two of the aforementioned types of apomixis exist in the genus, apospory and diplospory. Ovules of *B. microphylla* are nearly 100% diplosporous and about 30% aposporous, i.e., both mechanisms are observed to occur in about 30% of all ovules. *B. lignifera*, on the other hand, is nearly 100% diplosporous (Carman et al., 2007). *Boechera* also has a broad geographic distribution across North America, making it easily accessible for a large research community (Rushworth et al., 2011).

Additionally, helping to supplement genetic research, the morphological process of apomixis in *Boechera* has been well documented. *Boechera*, although containing the only documented natural apomicts in the Brassicaceae (Schranz et al., 2006), is closely related to the heavily studied model organism *Arabidopsis*

thaliana (Al-Shehbaz et al., 2006; Beilstein et al., 2006). The two genera are thought to be separated by only 12-19 million years (Arabidopsis Genome Initiative, 2000). Consequently, this close relationship provides an abundant wealth of molecular data and genetic annotation that can be extrapolated, with caution, to *Boechera* for molecular analyses. The impending release of the *Boechera* genome sequence will further increase the relevance of apomixis research involving *Boechera* (http://genome.jgi.doe.gov/genome-projects/).

Expression Profiling and Analyses

Microarrays and Preprocessing

Miniaturized microarrays are powerful, high-throughput tools that have been used over the last decade to study entire transcriptomes of various well-known genomes. Microarrays have multiple uses, including the study of gene expression profiles, nucleotide polymorphisms, and genotyping through the targeting of nucleotides such as DNA, RNA, or cDNA. Although various types of microarrays and manufacturers exist, Affymetrix has become a popular, cost-effective provider of arrays over recent years. Affymetrix GeneChip® arrays are one such type of microarray that have proven to be an effective, useful, and easily reproducible approach to studying gene expression, especially at the scale of the entire genome. GeneChip® arrays are typically on a 1.28 cm^2 chip containing over 500,000 locations that represent every transcribed gene in an

organism's entire transcriptome, and are designed to follow a simple and effective protocol for optimal reproducibility (Lipschutz et al., 1999).

Genes on a GeneChip® array are represented by sets of "probes," which are short oligonucleotides, typically 25 base-pairs that have been specifically designed to target unique regions of each gene (Lipschutz et al., 1999). Although the number of probes per set can vary by organism, a typical GeneChip® probe set can contain 11-20 probe "pairs" (Bolstad et al., 2005). A probe pair is a duo of probes that are designed to target the same region of a specific gene but are different from each other in that one contains a single-nucleotide polymorphism at the thirteenth base-pair (Mismatch probe, or MM probe), while the other matches the gene region perfectly (Perfect Match probe, or PM probe) (Lipschutz et al., 1999). Thus, each gene in the organism is represented by a unique set of PM and MM probe pairs that target unique regions of the gene. As an example, for each gene in the *Arabidopsis thaliana* transcriptome, there are 11 PM and 11 MM probes that target each of the approximate 22,800 genes on the chip (Affymetrix, 2010). This specific targeting of a probe pair to a unique region of only one gene is intended to strengthen the results by allowing each probe set to detect only a single gene. When the chips are manufactured, probe sets are attached and randomly scattered across the array.

For gene expression analyses, RNA is derived from the sample and is labeled with biotin. Once prepped, labeled RNA is then washed across the array and hybridization begins to occur. Hybridization refers to the matching and

binding of the array-attached probe sets to their complementary regions of the sample RNA. More RNA of a particular gene means more hybridization to the complementary probe pairs on the array. Once the washing is completed, a laser is used to induce biotin fluorescence. Light detectors and filters are used to collect the emitted signal and a computer then reads and interprets the intensity emitted from each probe (Lipschutz et al., 1999). Resulting intensities for each individual probe are then recorded and written into a large data storage (CEL) file, which can then be used for expression analysis (Gautier et al., 2004).

Before microarray data can be interpreted however, it is necessary to preprocess the data – especially when comparing across multiple samples. Various algorithms exist, such as the Robust Multichip Average (RMA), GCRMA, MAS 5.0, or PLIER. Typical steps for most algorithms include correcting for background noise, normalization to account for systematic and technical differences between arrays, and summarization of the probe intensities to the gene expression level (Bolstad et al., 2005). Many studies have been done as an attempt to distinguish a "best" algorithm, though there is no definitive answer (Bolstad et al., 2003; 2005). For example, the RMA algorithm was shown to be superior over the commonly used MAS 5.0 algorithm in a classic statistical study (Irizarry et al., 2003). Nevertheless, which normalization algorithm that is used is typically dependent on the data being studied or simply by user preference.

To preprocess microarray data, probe intensity files (CEL) are first read into bioinformatic software, such as R with Bioconductor packages, GeneSpring,

or HarVEST (Gautier et al., 2004; Morinaga et al., 2008). Quality control of the data can be done, followed by the use of any elected preprocessing algorithm. Regardless of the algorithm chosen, during preprocessing the algorithm utilizes an important file called the Chip Description File (CDF), which is what allows the software to interpret the raw probe intensity file. A Chip Description File is a key or map to the CEL files that annotate and describe each and every probe in the half-million locations on the entire chip (Gautier et al., 2004). CDFs are designed by the manufacturer and are specific to each type of chip made. For example, the *Arabidopsis* ATH1121501 GeneChip® has its own designated CDF, while each human genome GeneChip® has a separate, distinct CDF (Affymetrix, 2010). As is discussed with probe masking, modification of the CDF can also cause specific probes to be ignored completely by the preprocessing algorithm. The results of preprocessing are a conversion from a raw probe intensity file to a data set of summarized gene expression levels for each gene on the array.

Once microarray data has been preprocessed, analysis of the expression levels and subsequent interpretation can finally be done. In the case of testing multiple species or treatment groups, a very common and informative analysis is a differential expression test, which is a method of identifying genes that differ significantly in their expression across treatments (Scholtens & von Heydebreck, 2005). There are many types of differential expression algorithms, ranging from linear model based approaches such as *limma* (Smyth, 2005), to permutation-based approaches like *SAM* (Significance Analysis of Microarrays, Tusher et al.,

2001) and *maxT* (Pollard et al., 2005), and just as with preprocessing, selection of an algorithm can simply be user preference. Regardless of choice, differential expression tests are powerful tools for discovering meaningful genes at a statistical level, even finding significant genes that might have been missed in biological assays. For example, Tusher et al. identified genes that were being expressed in human lymphoblastoid cell lines as a direct result of exposure to ionizing radiation by utilizing a differential expression analysis. Microarrays were performed for the lymphoblastoid cells at both exposed and non-exposed treatment levels, and the resulting gene expression levels were compared against each other using SAM. As a result of the analysis, previously unrecognized genes were identified that participate in repair of damaged DNA (Tusher et al., 2001).

As powerful, easily available, and cost-effective microarrays are, there is still a severe limitation to the use of microarrays to a large part of the scientific community. Chips are designed by organism, and are limited only to well-known genomes that are mostly or fully sequenced. This fact increases their power, but also limits studies to a select number of organisms, thus making non-sequenced genomes, even those with high economic or scientific value, very difficult to study at the entire transcriptome level.

More recent platforms have helped to overcome weaknesses inherit in the microarray technology, such as next-generation (NGS) pyrosequencing techniques. Unlike microarrays, platforms such as 454 Sequencing, Illumina, and

SOLiD can evaluate absolute transcript levels of sequenced and unsequenced organisms, detect novel transcripts and isoforms, identify previously annotated 50 and 30 cDNA ends, map exon/intron boundaries, reveal sequence variations (e.g. SNPs) and splice variants and many more (Mutz et al., 2012). Although not as powerful as NGS techniques, this microarray transcriptome data will certainly provide useful and insightful information moving forward and will supplement future next-generation and third generation platform experiments.

Cross-Species Arrays and Probe Masking

Due to the wealth of gene expression information that can be obtained from a microarray, many have recently tried utilizing cross-species hybridization (CSH) as a cost-effective approach to allow the study of organisms for which no microarray is designed. CSH microarray analyses are done by applying sample RNA to a target array for which the sample was not designed, usually a closely related species (Bar-Or et al., 2007). There are many successful examples of cross-species microarrays, such as bovine, pig, and dog on human arrays (Ji et al., 2004), *Brassica oleracea* on *Arabidopsis thaliana* arrays (Hammond et al., 2005), canine on human (Grigoryev et al., 2005), two *Thlapsi* species on *A. thaliana* (Hammond et al., 2006), chimpanzee on human (Toleno et al., 2009), banana on rice (Davey et al., 2009), horse on human (Graham et al., 2009), and many others (Spiewak Rinaudo & Gerin, 2004; Becher et al., 2004; Hudson et al., 2007; Morinaga et al., 2008; NASCArrays:Xspecies, 2010).

There are fundamental problems that must be solved before conducting CSH studies. These include genome dissimilarity, cross-hybridization, and sequence polymorphisms, which can lead to bias. For example, if the sample RNA has amply diverged in terms of sequence, not necessarily in function, from the oligonucleotide probes on the array, that particular gene may be called as low expressed or not present at all. Even with good hybridization in some probes, if a sufficient amount of probes in a set do not hybridize well enough, a gene will be classified as low or not present, leading to its possible exclusion from the analysis. As an example, Benovoy et al. showed that simply a single nucleotide polymorphism in a probe's target sequence is enough to disrupt hybridization, with increasing severity depending on the SNP's position in the sequence (Benovoy et al., 2008). To counter this problem, various microarray procedure modifications can be taken to help insure that the quality of the expression data is good and not misinterpreted such as increasing the amount of repetitive samples, using longer oligonucleotide probes, or bioinformatic data filtration (Bar-Or et al., 2007). Data filtration is perhaps the most cost-effective example, due to the ease of massive data manipulation using bioinformatic software.

Although various methods of data filtration have been used over the last few years, gDNA-based filtration, first introduced by the Nottingham *Arabidopsis Stock Centre* (Craigon et al., 2004), has become an increasingly effective and simplistic data filtration method, and is considered an excellent choice for researchers that study non-sequenced species (Bar-Or et al., 2007). This is due to

the fact that gDNA-based filtration does not require any prior genomic knowledge of the species being studied, including its relation to the target microarray being used, though any information known increases the power of any data-filtration approach. gDNA-based data filtration is a three step process: 1) gDNA is used for an initial cross-species array, 2) probe masking is conducted based on the gDNA test using Xspecies procedures, and 3) results from probe masking are used in subsequent CSH array analyses.

In the first step, gDNA is applied directly to the target microarray. For example, before testing expression levels of drought and stress response genes in *Musa* (banana) on rice microarrays, gDNA samples were first taken from *Musa* leaves and applied to a rice array (Davey et al., 2009). Genomic DNA, where every gene is present, represents the best possible scenario for the CSH array and judges hybridization efficiency of the sample to the target array. If a probe has good hybridization to the genomic DNA, then that probe will be used in the gene expression analysis. On the contrary, probes with less hybridization affinity to the gDNA are removed from the study though masking. gDNA arrays are performed for each species being studied, and are done on the same array that the subsequent gene expression analysis will be done on. Intensities for the gDNA arrays are read as a standard microarray and are stored as a CEL file, which are then used to calibrate chips for probe-masking (Hammond et al., 2005).

Probe-masking, the next step in gDNA-based filtration, involves reducing the number of array probes used in microarray analysis, and has been utilized by

groups and methods other than just gDNA-based approaches. Probe reduction can be done by a variety of criteria, ranging from known percent-sequence similarity between probe and target, to comparative performance of homologous-species probes (Ji et al., 2004; Grigoryev et al., 2005) without losing specificity of detection or sensitivity (Antipova et al., 2002). In the case of gDNA filtration, probe masking is individual probe intensity-based, which determines a probe's inclusion to the analysis by how well each probe hybridized to gDNA. For example, if a particular probe did not hybridize well to the target gDNA sample, then it will be filtered out as probe masking begins.

As outlined by Hammond et al., probes are analyzed and removed by a Perl script called "Xspecies," which reads a gDNA CEL file. Xspecies is given a user-defined intensity threshold, and removes all probes with individual intensities below that defined threshold. If all probes of a particular probe set are removed, the probe set is also removed. After analysis of the gDNA CEL is completed, a new Chip Description File (CDF) is produced that only contains the remaining probe pairs and probe sets, which becomes a species-specific CDF that optimizes cross-species array analyses. As threshold selection is user-defined and therefore very subjective, masking at multiple thresholds is typically done to maximize probe set retention while optimizing the number of significant genes (Hammond et al., 2005).

The final step to gDNA-based filtration is the substitution of the Xspecies created CDF into the cross-species array analyses (NASCArrays:Xspecies 2010).

For gene expression studies, rather than the actual designed CDF for the array, the custom CDF is used to interpret the probe intensities, limiting the study only to those probes with good intensities, or in other words, utilizing only probes with good hybridization (Chain et al., 2008). RMA is typically used for probe masking as it excludes all MM probes in the analysis, as MM probes have been shown to disrupt microarray analyses by detecting true signal (Irizarry et al., 2003), and sometimes hybridizing more efficiently to cross-species arrays than their PM counterparts (Grigoryev et al., 2005).

Retaining only probes with good hybridization intensities through gDNA-based masking has been well demonstrated in cross-species arrays to increase the array sensitivity, detect greater numbers of significant genes, and provide expression levels consistent with other methods of measuring gene expression for a wide range of organisms that vary in similarity to the cross-species array targets (Hammond et al., 2005; Morinaga et al., 2008; Davey et al., 2009; Graham et al., 2009). gDNA-based masking has even been shown to improve homologous-species arrays (Graham et al., 2007). For example, when Hammond et al. studied shoot transcriptomes of two different *Thlapsi* species, gDNA-based data filtration was done utilizing unique CDFs for both of the *Thlapsi* species on cross-species *Arabidopsis thaliana* microarray results. Not only was detection of significant genes increased in both species, results were more consistent with RT-PCR expression levels (Hammond et al., 2006).

Although presenting its own unique set of challenges, cross-species hybridization microarrays are a viable and cost-effective option in the studying of species that do not have pre-manufactured chips. Through additional measures such as data filtration using gDNA-based probe masking, cross-species arrays can be further strengthened, and novel traits and organisms can be explored at the entire transcriptome level.

Gene Ontologies and Enrichments

Another useful tool for the bioinformatic study of transcriptomes is the use of known genetic annotation, such as Gene Ontology (GO). GO is a structured hierarchical classification system that categorizes genes into groups based on certain criteria, which are of three types: Molecular Function (MF), Biological Process (BP), or Cellular Component (CC) (Gene Ontology, 2000). GOs span across organisms, and allow for database interoperability. For example, if a gene was found to be part of the electron transport chain, it would be assigned a GO of GO:0006118, and would be considered a part of a Biological Process (BP). Thus, any genes found to be involved in the electron transport chain would also be assigned this same GO value thus demonstrating a clear relationship of any genes involved in similar processes no matter the organism. The same could be said of any gene found in similar cellular locations or molecular functions (Gene Ontology, 2000).

GOs can also be used as information in filtering and useful statistical tests as well. For example, it could be valuable to test for differential expression of

genes in a specific ontology only. Even GOs themselves can be used as a test of significance, as categories of genes can be tested for over-representation in a transcriptome. Since many complex phenotypes can involve the use of multiple genes or gene products, GOs can be used to identify significant processes or functions that may be strongly correlated to the resulting phenotype of interest. By using one of many global testing algorithms, such as the publicly available AmiGO (Gene Ontology, 2000), GOrilla (Eden et al., 2009), or GOEAST (Zheng & Wang, 2008), specific groups of genes can be found to be significant, even in large datasets with large expression profiles.

With the establishment of the Gene Ontology Consortium in 2000, the number and specificity of GOs have continued to expand and evolve, providing a powerful, multi-organismal reference. Using GOs not only increases data interoperability between variable research projects but also provides additional biological relevance to informatics results. All Gene Ontology information can be found on the consortium's website at www.geneontology.org.

Development for Analyses of Microarray Data

gDNA-based Probe Selection

Probe-pairs from the *A. thaliana* ATH1-121501 Gene-Chip® array (Affymetrix, 2010) were selected for transcriptome analysis of *Boechera* species using a gDNA-based probe-selection strategy based on the hybridization of gDNA to the PM probe. Total genomic DNA was extracted from *B. formosa, B.*

microphylla, and *B. lignifera* leaves and mailed to the Nottingham Arabidopsis Stock Center, UK. In return, a .CEL file containing the gDNA hybridization intensities between genomic DNA fragments of the three *Boechera* species was generated. Probe-pairs from the .CEL file were selected for subsequent transcriptome analysis using a .CEL file parser script (Xspecies) written in the Perl programming language. The Perl script (NASCArrays:Xspecies, 2010) was designed to create probe mask (.CDF) files compatible with a range of microarray analysis software packages. A probe-set was selected when it was represented by one or more PM probe-pair(s) per probe-set. Modification of the Xspecies Perl code was also done to allow multiple thresholds of "minimum probe" restrictions, so as to test masking against a minimum of 2-7 probes per probe set, including restrictions on odd or even numbers (Appendix).

There is no a priori restriction of a suitable gDNA hybridization intensity threshold for probe mask file generation in a target species. Thus, the algorithm allows for a user-specified gDNA hybridization intensity threshold for probe mask file generation. Files (.CDF) were generated using a range of gDNA hybridization intensity thresholds (from 0 to 1000). Thresholds generating the best balance of significant gene results and gene set retention were used (Appendix). After an optimal threshold was chosen for each species, a combined-species CDF was selected that utilizes all probes retained from every optimal threshold file (Appendix). This was done to insure that no artificial

significance would be introduced due to probes being masked in one species' array, and not the other.

Differential Expression Analyses

All differential expression analyses were done in R using multiple Bioconductor packages including those for quality control (Appendix). For the Affymetrix ATH1 arrays, the *affy* package (Gautier et al., 2004) was used to load expression data. In order to use the custom .CDF files for probe masking, the *makecdfenv* package (Irizzary et al., 2006) was used. For array quality control, the *affyPLM* package (Bolstad et al., 2005) was used. The RMA (Robust Multi-chip Average) algorithm using default parameters was used to normalize chip data. This normalization algorithm does not use the mismatch (MM) probes and is more appropriate for probe masking. The *samr* package (Tibshirani et al., 2010), which implements the Significance Analysis of Microarrays (Tusher et al., 2001) permutation algorithm, was used to determine significance for each differential analysis. For additional confirmation of significant genes, a permutation-based, nested factorial model was also used, called *affyNFM* (Stevens et al., 2010). For all analyses default parameters and a Benjamini-Hochberg false discovery rate (BH-FDR) of 0.05 was used (Benjamini & Hochberg, 1995).

Python Data Organization

Various scripts were written to organize and analyze data sets in both *Boechera* and Sorghum. Manipulation of results delivered in R, GOEAST, and TAIR (Lamesch et al., 2011) was important and common throughout analyses.

Most Python scripts for *Boechera* projects were created *ad hoc* to fulfill various reporting and data manipulation needs. In the case of the Sorghum project, a Kashiwa BioImaging script (KBI: Kashiwa BioImaging, 2007) was modified to run my own created module, allowing for large gene sets to be BLASTed against the TAIR database. A Python script was also written to utilize the online TAIR gene search tool. Excerpts and examples of the Python code can be found in the appendix.

Results

Utilizing Affymetrix ATHI *Arabidopsis thaliana* GeneChip arrays, differential expression was tested between various species of *Boechera* of both sexual and apomictic reproductive type, testing at comparable stages of ovule and whole pistil development. Ovules of sexual *B. formosa* and apomictic *B. microphylla* and *B. lignifera* were collected by the Carman lab from 2005-2010 and processed using microarrays. For consistency between studies, additional data involving collected pistils, instead of ovules, were obtained for apomictic *B. lignifera* and sexual *B. stricta* using the microarray platform. Five separate tests were conducted, utilizing 28 microarrays, between stages 1 and 3 of sexual *B. formosa* and *B. stricta,* and apomictic *B. lignifora* and *B. microphylla*, with 2 reps per stage. Two stages of development with two reps of whole pistils were also compared – between apomictic *B. lignifora* and obligate sexual *B. stricta*. Every

analysis was repeated using optimal masking thresholds (Appendix), and results can be seen in Table 1.

Table 1 Results for all SAM analyses, both masked and non-masked. L – *Boechera lignifora*, M – *Boechera microphylla*, F – *Boechera formosa*, S – *Boechera stricta*.

Ovule Analyses							
		No Masking			Masking		
Stages	**Species**	**Up**	**Down**	**Total**	**Up**	**Down**	**Total**
1	L vs. M	2	0	2	1	0	1
1-2	L	0	0	0	0	0	0
1-2	M	0	1	1	0	1	1
2	F vs. L	368	378	746	695	778	1473
2	F vs. M	482	441	923	1009	913	1922
2	L vs. M	5	3	8	16	7	23
2-3	L	0	0	0	0	0	0
2-3	M	0	0	0	0	0	0
2-3	F	0	1	1	0	0	0
3	F vs. L	415	1495	1910	622	1676	2298
3	F vs. M	1038	1971	3009	1433	2102	3535
3	L vs. M	0	0	0	3	1	4
3-4	L	1487	2309	3796	1646	2027	3673
3-4	M	1142	993	2135	1375	1319	2694
4	L vs. M	2932	1202	4134	2920	2059	4979
1-3	L	0	0	0	0	0	0
1-3	M	2	1	3	4	0	4
Pistil Analyses							
1-1	L vs. S	995	969	1964	1172	1298	2470
1-2	L vs. S	595	817	1412	408	439	847
1-2	S vs. L	968	699	1667	949	781	1730
1-2	L	148	89	237	117	3	120
1-2	S	311	328	639	82	154	236
2-2	L vs. S	320	497	817	92	374	466

A nested factorial model was additionally used, and results were compared against the SAM results for additional confirmation of significance, and as a gauge of the efficacy of masking. NFM results can be found in Table 2.

Table 2: Comparison of similarities between significant genes produced by SAM and NFM results. L – *Boechera lignifora*, M – *Boechera microphylla*, F – *Boechera formosa*, S – *Boechera stricta*.

Stage	Analysis	NFM	NFM Masked
2	F vs. L	2981	2918
2	F vs. M	3064	3008
3	F vs. L	3253	3054
3	F vs. M	3303	3154
1	L vs. S	3365	3184
2	L vs. S	2938	2655

Similarity between the methods, both with masking and without, can be seen in Table 3.

Table 3 Comparison of results between NFM and SAM. L – *Boechera lignifora*, M – *Boechera microphylla*, F – *Boechera formosa*, S – *Boechera stricta*.

SAM & NFM				
Stage	**Analysis**	**SAM**	**In Common**	**% Similar**
1	L vs. S	1964	1402	71.4%
2	L vs. S	817	717	87.8%
2	F vs. L & M	304	286	94.1%
3	F vs. L & M	1184	892	75.3%
1-3	Sex vs. Apomixis	55	40	72.7%
1-3	Sex vs. Apomixis	17	13	76.5%
SAM & NFM Masked				
1	L vs. S	2470	1661	67.2%
2	L vs. S	466	413	88.6%
2	F vs. L & M	815	666	81.7%
3	F vs. L & M	1602	1093	68.2%
1-3	Sex vs. Apomixis	132	74	56.1%
1-3	Sex vs. Apomixis	20	12	60.0%

All significant genes were then uploaded to GOEAST for Gene Ontology enrichment. GOs were generated for both masked and non-masked significant gene sets, and totals can be found in Table 4. Overall, masking showed an increase in the total number of significant genes, but a lower number of enriched gene ontologies.

Table 4 Number of enriched Gene Ontologies per stage-by-stage masked comparison. L – *Boechera lignifora*, M – *Boechera microphylla*, F – *Boechera formosa*, S – *Boechera stricta*.

Stage	Analysis	Non-Masked GOs	Masked GOs
1	L vs. S	247	127
2	L vs. S	118	91
2	F vs. L	99	103
2	F vs. M	182	194
3	F vs. L	327	279
3	F vs. M	336	327
1-3	S vs. A	45	47
1-3	S vs. A	25	12

Resources Cited

Affymetrix. **2010**. http://www.affymetrix.com/

Al-Shehbaz A, Beilstein MA, Kellogg EA. **2006**. Systematics and phylogeny of the Brassicaceae (Cruciferae): an overview. *Pl. Syst. Evol.* 259: 89-120

Antipova AA, Tamayo P, Golub TR. 2002. A strategy for oligonucleotide microarray probe reduction. *Genome Biology* 3(12):research0073.1–0073.4

Armenta-Medina A, Demesa-Arevalo E, Vielle-Calzada J-P. 2011. Epigenetic control of cell specification during female gametogenesis. *Sexual Plant Reproduction* DOI 10.1007/s00497-011-0166-z (online April 12, 2011)

Arabidopsis Genome Initiative. 2000. Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. *Nature* 408: 796-815

Asker SE, Jerling L. 1992. Apomixis in Plants. CRC Press, Florida, 2000, 298

Beck JB, Alexander PJ, Allphin L, Al-Shehbaz IA, Rushworth C, Bailey CD, Windham MD. 2011. Does Hybridization Drive The Transition To Asexuality In Diploid *Boechera*? *Evolution* 66(4): 985–995

Bar-Or C, Czosnek H, Koltai H. 2007. Cross-species microarray hybridizations: a developing tool for studying species diversity. *TRENDS in Genetics* 23(4): 200-207

Becher M, Talke IN, Krail L, Kramer U. 2004. Cross-species microarray transcript profiling reveals high constitutive expression of metal homeostasis genes in shoot of the zinc hyperaccumulator *Arabidopsis halleri*. *The Plant Journal* 37: 251-268

Beilstein MA, Al-Shehbaz IA, Kellogg EA. 2006. Brassicaceae Phylogeny and Trichome Evolution. *American Journal of Botany* 93(4): 607–619

Benjamini Y, Hochberg Y. 1995. Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. *J. R. Statist. Soc. B* 57, No. 1, pp. 289-300

Benovoy D, Kwan T, Majewski J. 2008. Effect of polymorphisms within probe–target sequences on olignonucleotide microarray experiments. *Nucleic Acids Research* 36(13): 4417–4423

Böcher TW. 1951. Cytological and embryological studies in the amphi-apomictic Arabis holboellii complex. *Det Kongelige Danske Videnskabernes Selskab* 6:1–59

Bolstad BM, Irizarry RA, Astrand M, Speed TP. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19(2):185-93

Bolstad BM, Irizarry RA, Gautier L, Wu Z. 2005. In: *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds), Springer, New York, 2005. p. 13-32

Carman JG. 1997. Asynchronous expression of duplicate genes in angiosperms may cause apomixis, bispory, tetraspory, and polyembryony. *Biological Journal of the Linnean Society 61*: 51-94.

Carman JG. 2004. Apomixis for crop production: Status of technology development and commercialization implications. *Willamette J Intern Law Dispute Resol* 12: 28-48

Carman JG. 2007. Do duplicate genes cause apomixis? In Hörandl E. Grossniklaus U. van Dijk P.J. Sharbel T.F. (ed) *Apomixis: evolution, mechanisms and perspectives*. A. R. G. Gantner Verlag K. G. p. 63-91.

Carman JG, Jamison M, Elliott E, Dwivedi KK, Naumova TN. 2011. Apospory appears to accelerate onset of meiosis and sexual embryo sac formation in sorghum ovules. *BMC Plant Biology* 11:9–22

Cervigni GDL, Paniego N, Pessino S, Selva JP, Díaz M, Spangenberg G, Echenique V. 2008. Gene expression in diplosporous and sexual Eragrostis curvula genotypes with differing ploidy levels. *Plant Mol Biol*. 67:11–23

Chain FJ, Ilieva D, Evans BJ. 2008. Duplicate gene evolution and expression in the wake of vertebrate allopolyploidization. *BMC Evolutionary Biology* 8: 43

Craigon DJ, James N, Okyere J, Higgins J, Jotham J, May S. 2004. NASCArrays: a repository for microarray data generated by NASC's transcriptomics service. *Nucleic Acids Research* 32: Database issue

Davey MW, Graham NS, Vanholme B, Swennen R, May ST, Keulemans J. 2009. Heterologous oligonucleotide microarrays for transcriptomics in a non-model species; a proof-of-concept study of drought stress in *Musa. BMC Genomics* 10(436)

Eden E, Navon R, Steinfeld I, Lipson D, Yakhini Z. 2009. GOrilla: A Tool For Discovery And Visualization of Enriched GO Terms in Ranked Gene Lists. *BMC Bioinformatics* 10: 48

Gene Ontology Consortium. 2000. Gene ontology: tool for the unification of biology. *Nat. Genet.* 25(1):25-9.

Graham NS, Broadley MR, Hammond JP, White PJ, May ST. 2007. Optimising the analysis of transcript data using high density oligonucleotide arrays and genomic DNA-based probe selection. *BMC Genomics* 8**:** 344

Graham NS, Clutterbuck AL, James N, Lea RG, Mobasheri A, Broadley MR, Maya ST. 2009. Equine transcriptome quantification using human GeneChip arrays can be improved using genomic DNA hybridisation and probe selection. *The Veterinary Journal* doi:10.1016/j.tvjl.2009.08.030

Grigoryev DN, Ma S, Simon BA, Irizarry RA, Ye SQ, Garcia JGN. 2005. *In vitro* identification and *in silico* utilization of interspecies sequence similarities using GeneChip® technology. *BMC Genomics* 6: 62

Gautier L, Cope L, Bolstad B, Irizarry RA. 2004. affy – analysis ofAffymetrix GeneChip data at the probe level. Bioinformatics 20(3): 307-15

Hammond JP, Bowen HC, White JP, Mills V, Pyke KA, Baker AJM, Whiting SN, May ST, Broadley MR. 2006. A comparison of the *Thlaspi caerulescens* and *Thlaspi arvense* shoot transcriptomes. *New Phytologist* 170:239–260

Hammond JP, Broadley MR, Craigon DJ, Higgins J, Emmerson ZF, Townsend HJ, White PJ, May ST. 2005. Using genomic DNA-based probe-selection to improve the sensitivity of high-density oligonucleotide arrays when applied to heterologous species. *Plant Methods* 1(10)

Hörandl E. 2009. A combinational theory for the maintenance of sex. Heredity 103: 445-457

Hudson ME, Bruggink T, Chang SH. 2007. Analysis of gene expression during *Brassica* seed germination using a cross species microarray platform. *Crop*

Science 47(S2): S96-S112

Irizarry RA, Gautier L, Huber W, Bolstad B. 2006. makecdfenv: CDF Environment Maker. R package version 1.26.0

Ji W, Zhou W, Gregg K, Yu N, Davis S, Davis S. 2004. A method for cross-species gene expression analysis with high-density oligonucleotide arrays. *Nucleic Acids Research* 32(11): e93

KBI: Kashiwa BioImaging. 2007. http://hasezawa.ib.k.u-tokyo.ac.jp/zp/Kbi/FrontPage

Knox RB, Heslop-Harrison J. 1963. Experimental control of aposporous apomixis in a grass of the Andropogoneae. *Bot Not*. 116:127-141.

Koltunow AM. 1993. Apomixis: embryo sacs and embryos formed without meiosis or fertilization in ovules. *Plant Cell* 5:1425-1437

Lamesch P, Berardini TZ, Li D, Swarbreck D, Wilks C, Sasidharan R, Muller R, Dreher K, Alexander DL, Garcia-Hernandez M, Karthikeyan AS, Lee CH, Nelson WD, Ploetz L, Singh S, Wensel A, Huala E. 2011. The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. *Nucl. Acids Res*. 40: D1202-10

Lampert KP. 2008. Facultative parthenogenesis in vertebrates: reproductive error or chance? *Sexual Development* 2: 290-301.

Laspina NV, Vega T, Seijo JG, González AM, Martelotto LG, Stein J, Podio M, Ortiz JPA, Echenique VC, Quarin CL, Pessino SC. 2008. Gene expression analysis at the onset of aposporous apomixis in Paspalum notatum. *Plant Mol. Biol.* 67:615–628

Lipschutz RJ, Fodor SPA, Gingeras TR, Lockhart DJ. 1999. High density synthetic oligonucleotide arrays. *Nature Genetics* 21: 20–24.

Mogie, M. 1992. The Evolution of Asexual Reproduction in Plants. *Chapman and Hall*, London

Morinaga S-I, Nagano AJ, Miyazaki S, Kubo M, Demura T, Fukuda H, Sakai S, Hasebe M. 2008. Ecogenomics of cleistogamous and chasmogamous flowering: genome-wide gene expression patterns from cross-species microarray analysis in *Cardamine kokaiensis* (Brassicaceae). *Journal of Ecology* 96: 1086–1097

Mutz K-O, Heilkenbrinker A, Lönne M, Walter J-G, Stahl F. 2012. Transcriptome analysis using next-generation sequencing. *Curr Opin Biotechnol*, http://dx.doi.org/10.1016/j.copbio.2012.09.004

NASCArrays:Xspecies. 2010. http://affymetrix.arabidopsis.info/xspecies/

Ochogavía AC, Seijo JG, González AM, Podio M, Silveira ED, Lacerda ALM, Cameiro VTC, Ortiz JPA, Pessino SC. 2011. Characterization of retrotransposon sequences expressed in inflorescences of apomictic and sexual Paspalum notatum plants. *Sex Plant Reprod.* 24:231–246

Ozias-Akins P, van Dijk PJ. 2007. Mendelian Genetics of Apomixis in Plants. *Annu. Rev. Genet*. 41:509–37

Polegri L, Calderini O, Arcioni S, Pupilli F. 2010. Specific expression of apomixis-linked alleles revealed by comparative transcriptomic analysis of sexual and apomictic Paspalum simplex Morong flowers. *Journal of Experimental Botany* 61(6): 1869–1883

Pollard KS, Dudoit S, van der Laan MJ. 2005. Multiple Testing Procedures: the *multtest* Package and Applications to Genomics. In: *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds), Springer, New York, 2005. p. 249-72

Rodrigues JCM, Luo M, Berger F, Koltunow AMG. 2010. Polycomb group gene function in sexual and asexual seed development in angiosperms. *Sexual Plant Reproduction* 23: 123-133.

Rushworth CA, Song B-A, Lee C-R, Mitchell-Olds T. 2011. Boechera, a model system for ecological genomics. *Molecular Ecology* 20: 4843–4857.

Sahu PP, Gupta S, Malaviya DR, Roy AK, Kaushal P, Prasad M. 2011. Transcriptome analysis of differentially expressed genes during embryo sac development in apomeiotic non-parthenogenetic interspecific hybrid of Pennisetum glaucum. *Mol Biotechnol*. 51(3): 262–271

Savidan, Y. 2000. In: *Plant Breeding Reviews,* vol. 18, p. 13-86.

Scholtens D, von Heydebreck A. 2005. Analysis of Differential Gene Expression Studies. In: *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W.

Huber (eds), Springer, New York, 2005.

Schranz ME, Dobes C, Koch MA, Mitchell-Olds T. 2005. Sexual Reproduction, Hybridization, Apomixis, and Polyploidization in the Genus *Boechera* (Brassicaceae). *American Journal of Botany* 92(11): 1797–1810

Schranz ME, Kantama L, de Jong H, Mitchell-Olds T. 2006. Asexual reproduction in a close relative of *Arabidopsis*: a genetic investigation of apomixis in *Boechera* (Brassicaceae). *New Phytologist* 171: 425–438

Sharbel TF, Voigt M-L, Corral JM, Thiel T, Varshney A, Kumlehn J, Vogel H, Rotter B. 2009. Molecular signatures of apomictic and sexual ovules in the Boechera holboellii complex. *The Plant Journal* 1365– 313X.2009.03826.x

Sharbel TF, Voigt M-L, Corral JM, Galla G, Kumlehn J, Klukas C, Schreiber F, Vogel H, Rotter B. 2010. Apomictic and sexual ovules of *Boechera* display heterochronic global gene expression patterns. *The Plant Cell* 22: 655–671.

Smyth GK. 2005. Limma: linear models for microarray data. In: *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds), Springer, New York, 2005. p. 397 -420

Spiewak Rinaudo JA, Gerin JL. 2004. Cross-species hybridization: characterization of gene expression in Woodchuck liver using human membrane arrays. *Journal of Medical Virology* 74: 300–313

Spillane C, Curtis MD, Grossniklaus U. 2004. Apomixis technology development – virgin births in farmers' fields? *Nature Biotechnology* 22: 687–691.

Stevens JR, Bell JL, Aston KI, White KL. 2010. A comparison of probe-level and probeset models for small-sample gene expression data. *BMC Bioinformatics* 11:281

Suomalainen E, Saura A, Lokki J. 1987. Cytology and evolution in parthenogenesis. CRC Press, Baca Raton, Florida

Tibshirani R, Chu G, Hastie T, Narasimhan B. 2010. samr: SAM: Significance Analysis of Microarrays. R package version 1.28. http://CRAN.R-project.org/package=samr

Toleno DM, Renaud G, Wolfsberg TG, Islam M, Wildman DE, Siegmund KD,

Hacia JG. 2009. Development and evaluation of new mask protocols for gene expression profiling in humans and chimpanzees. *BMC Bioinformatics* 10:77

Tucker MA, Koltunow AMG. 2009. Sexual and asexual (apomictic) seed development in flowering plants: molecular, morphological and evolutionary relationships. *Functional Plant Biology* 36: 490–504

Tusher VG, Tibshirani R, Chu G. 2001. Significance analysis of microarrays applied to the ionizing radiation response. Proceedings of the National Academy of Sciences 98(9): 5116-5121

Zeng Y, Conner J, Ozias-Akins P. 2011. Identification of ovule transcripts from the Apospory-Specific Genomic Region (ASGR)-carrier chromosome. *BMC Genomics* 12: 206–221

Zheng Q, Wang XJ. 2008. GOEAST: a web-based software toolkit for Gene Ontology enrichment analysis. *Nucleic Acids Res.* 36 (suppl 2): W358-W363.

Appendix

Code

Quality controls using R

```
#Look at images of residuals:
library(affyPLM)
Pset <- rmaPLM(abatch.raw)                    #abatch.raw = affy object of CEL files
par(mfrow=c(2,2))
image(Pset, type="sign.resids")

#Look at raw intensity histographs:
Tment = c(rep(0,8),rep(1,16))                 #sexuals = 0; apomicts = 1
hist(abatch.raw,col=Tment+1)
Tment = c(0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12)  #each stage/mode had own T-
level
hist(abatch.raw,col=Tment+1)

#Check covariance:
e.mat <- 2^exprs(all.eset)
gene.mean <- apply(e.mat,1,mean)
gene.sd <- apply(e.mat,1,sd)
gene.cv <- gene.sd/gene.mean
hist(gene.cv)
hist(log2(gene.mean))
hist(gene.cv, xlim=0:1)
eset.all = rma(abatch.raw)

#MA plot:
par(mfrow=c(3,3))
MAplot(abatch.raw[,(selected_arrays)], loess.col='white', cex=1, cex.main=0.5)
```

Normalization and DE tests examples using R

```
#Non-masked SAM:
#Lignifora stage 1 vs Stricta stage 1
library(affy)
abatch.raw = ReadAffy()
L1vsS1.exprs = exprs(rma(abatch.raw[,c(1,2,5,6)]))
T.cell = c(1,1,2,2)
gn <- rownames(L1vsS1.exprs)
data <- list(x= L1vsS1.exprs, y=T.cell,geneid=gn, genenames=gn, logged2=TRUE)
samr.obj <- samr(data, resp.type="Two class unpaired",nperms=1000, random.seed=42)
delta.table <- samr.compute.delta.table(samr.obj)
delta.table[,c(1,4,5)]
delta.table <- samr.compute.delta.table(samr.obj, dels = seq(.54,.7,by=.001))
delta.table[,c(1,4,5)]
#Best delta = 0.566      FDR = 0.04941315
SAM.tab <- samr.compute.siggenes.table(samr.obj, 0.566, data,delta.table)
gn.up <- SAM.tab$genes.up[,3]; gn.dn <- SAM.tab$genes.lo[,3]
```

```
gn.SAM <- c(gn.up,gn.dn)
length(gn.SAM)
SAM.tab$ngenes.up
SAM.tab$ngenes.lo

#Masking using SAM:
library(makecdfenv)
allLMF = make.cdf.env("allLMF.cdf", cdf.path = "~/Documents/2009-2011 Utah
State/Dissertation/Microarray Data/New Boechera Arrays/Work Folder")
abatch.raw@cdfName <- "allLMF"
#Lignifora vs. Microphylla stage 1:
library(affy)
abatch.raw = ReadAffy()
Lign1vMicro1.eset = exprs(rma(abatch.raw[,c(9:10,17:18)]))  #normalization
T.cell = c(1,1,2,2)
gn <- rownames(Lign1vMicro1.eset)
data <- list(x= Lign1vMicro1.eset, y=T.cell,geneid=gn, genenames=gn, logged2=TRUE)
samr.obj <- samr(data, resp.type="Two class unpaired",nperms=500, random.seed=42)
delta.table <- samr.compute.delta.table(samr.obj,)
delta.table[,c(1,4,5)]
#Best delta = 2.005705009        FDR =  0.0000000
SAM.tab <- samr.compute.siggenes.table(samr.obj, 2.005705009, data,delta.table)
gn.up <- SAM.tab$genes.up[,3]; gn.dn <- SAM.tab$genes.lo[,3]
gn.SAM <- c(gn.up,gn.dn)
length(gn.SAM)
SAM.tab$ngenes.up
SAM.tab$ngenes.lo

#NFM:
library(affy)
abatch.raw = ReadAffy()
library(nlme);library(perm)
use.t1 <- c(1, 2)
use.t2 = c(3,4)
use.abatch = abatch.raw[,c(7,8,23,24)]
set.seed(1234)
use.gn <- geneNames(use.abatch)
use.wd <- getwd()
use.filename <- "Form2vsLign2_NFM"
source("http://www.stat.usu.edu/~jrstevens/affyNFM.R")
affyNFM(abatch = use.abatch, t1 = use.t1, t2 = use.t2, gn = use.gn, wd = use.wd, filename =
use.filename)

#Masking using NFM:
library(makecdfenv)
allLMF = make.cdf.env("allLMF.cdf", cdf.path = "~/Documents/2009-2011 Utah
State/Dissertation/Microarray Data/New Boechera Arrays/Work Folder")
abatch.raw@cdfName <- "allLMF"
library(affy)
```

```
abatch.raw = ReadAffy()
library(nlme);library(perm)
use.t1 <- c(1, 2)
use.t2 = c(3,4)
use.abatch = abatch.raw[,c(7,8,35,36)]
set.seed(1234)
use.gn <- geneNames(use.abatch)
use.wd <- getwd()
use.filename <- "Form2vsMicro2_NFM_MASKED"
source("http://www.stat.usu.edu/~jrstevens/affyNFM.R")
affyNFM(abatch = use.abatch, t1 = use.t1, t2 = use.t2, gn = use.gn, wd = use.wd, filename =
use.filename)

#Organizing final results in NFM, masked and non-masked:
source("http://www.stat.usu.edu/~jrstevens/affyNFM.R")
library(value)
library(affy)
library(nlme)
library(perm)
use.wd = getwd()
use.filename = 'Form2vsLign2_NFM'
use.frame <- read.csv(paste(use.wd, "/", use.filename, ".csv", sep = ""))
pframe <- nfm.pvals(use.frame)
head(pframe)
use.abatch = abatch.raw[,c(7,8,23,24)]
gn.abatch = geneNames(use.abatch)
pframe$q <- qvalue(p = pframe$p)$q
conc.ctl <- as.numeric(pData(use.abatch)[1, ])
conc.trt <- as.numeric(pData(use.abatch)[4, ])
f <- data.frame(gn = gn.abatch, C = conc.ctl, T = conc.trt)39
g <- merge(f, pframe)
write.csv(g[g$p<=0.05,], file = "Form2vsLign2_NFM_RESULTS.csv")
```

Data organization using Python

```
#!/usr/bin/env python
#GOID_matcher.py
'''This code is used to find GOs that are in GOEAST outputs, and then organize
them into a table, showing which GOs are represented in more than one analysis'''

import csv

GOID_and_counts = {}
GOID_and_files = {}
GOID_and_term = {}
filenames = []
output = []

#Shell code to get FileNames.txt: find . -name '*GOs.txt' > GO_FileNames.txt
filenamefile = open('GO_FileNames.txt','r')
```

```
GOfile = open('GOs UpDown Reg Gene Counts Masked.csv','r')

for row in filenamefile:
    filenames.append(row.strip())

for row in GOfile:
    GOID_and_term[row.strip().split(',')[1]] = row.strip().split(',')[2]

def GOID_in_set_counter(GOID):
    '''Counts the number of times a GOID is found in GOEAST outputs'''
    if GOID in GOID_and_counts:
        GOID_and_counts[GOID] = GOID_and_counts[GOID] + 1
    else:
        GOID_and_counts[GOID] = 1

def GOID_in_set_names(GOID, containing_file):
    '''Attaches the names of the files for in which the GOID is contained'''
    if GOID in GOID_and_files:
        GOID_and_files[GOID] = GOID_and_files[GOID] + "; " + str(containing_file)
    else:
        GOID_and_files[GOID] = str(containing_file)


if __name__ == '__main__':
    for filename in filenames:
        print filename
        current_file = open(filename.strip(),'r')
        for row in current_file:
            GOID_in_set_counter(row.strip().split('¥t')[0])
            GOID_in_set_names(row.strip().split('¥t')[0],filename.strip().split('/')[-1])

    for key, val in GOID_and_counts.items():
        output.append([key,GOID_and_term[key],val,GOID_and_files[key]])

    print "Results saved as ¥'GOs in common.csv¥'"

    outfile = open('GOs in common.csv','wb')
    outwriter = csv.writer(outfile)
    outwriter.writerows(output)
    outfile.close()


#GO_Counter.py
'''This code matches and counts the number of GOIDs found in each given file,
and summarizes them into a table'''

import csv
GOs = {}
GOs_and_loci = []
```

```
GOs_and_terms = {}
last_GOID = "GO:00000000"
output = []

def loci_in_GOIDs(GOID, locus):
    if GOID in GOs:
        if locus in GOs_and_loci:
            print "already found"
        else:
            GOs[GOID] = GOs[GOID] + 1
            GOs_and_loci.append(locus)
    else:
        GOs[GOID] = 1
        GOs_and_loci.append(locus)


if __name__ == '__main__':
    datafile = open('1-3_SignificantGenes*_GOs_Masked.txt', 'r')
    for row in datafile:
        #print row
        GOID = row.strip().split('¥t')[4]
        term = row.strip().split('¥t')[3]
        locus = row.strip().split('¥t')[0]
        if len(locus) >= 10:
            print "Bad locus:" + locus
        current_GOID = GOID
        if current_GOID != last_GOID:
            GOs_and_loci = []
        loci_in_GOIDs(GOID, locus)
        last_GOID = current_GOID
        GOs_and_terms[GOID] = term

    for key, val in GOs.items():
        output.append([key,GOs_and_terms[key],val])

    print "Results can be found in ¥'Summarized GOs.csv¥'"

    outfile = open('Summarized GOs.csv','wb')
    outwriter = csv.writer(outfile)
    outwriter.writerows(output)
    outfile.close()


#!/usr/bin/env python
#Siggene_matcher.py
'''This code is used to find significant genes that are in multiple DE analyses
outputs, and then organize them into a table, showing which genes are
represented in more than one analysis'''
```

```
import csv

probes_and_counts = {}
probes_and_files = {}
probes_and_loci = {}
filenames = []
output = []

#Shell code to get FileNames.txt: find . -name *RESULTS.csv > FileNames.txt
filenamefile = open('FileNames.txt','r')
locusfile = open('AllArrayProbes_NoMasking_LocusFile.txt','r')

for row in filenamefile:
    filenames.append(row.strip())

for row in locusfile:
    probes_and_loci[row.strip().split('¥t')[0]] = row.strip().split('¥t')[1]

def gene_in_set_counter(gene):
    '''Counts the number of times a gene is found in siggene outputs'''
    if gene in probes_and_counts:
        probes_and_counts[gene] = probes_and_counts[gene] + 1
    else:
        probes_and_counts[gene] = 1

def gene_in_set_names(gene, containing_file):
    '''Attaches the names of the files in which the gene is contained'''
    if gene in probes_and_files:
        probes_and_files[gene] = probes_and_files[gene] + "; " + str(containing_file)
    else:
        probes_and_files[gene] = str(containing_file)

if __name__ == '__main__':
    for filename in filenames:
        print filename
        current_file = open(filename.strip(),'r')
        for row in current_file:
            #print len(row)
            gene_in_set_counter(row.strip().split(',')[1])
            gene_in_set_names(row.strip().split(',')[1],filename.strip().split('/')[-1])

    for key, val in probes_and_counts.items():
        output.append([key,probes_and_loci[key],val,probes_and_files[key]])

    outfile = open('Significant_Genes_in_Common.csv','wb')
    outwriter = csv.writer(outfile)
    outwriter.writerows(output)
    outfile.close()
```

```
    print "Output saved as Significant_Genes_in_Common.csv"


#!/usr/bin/env python
#GO_Result_Organizer.py
'''The purpose of this script is to read in the "plain text" GOEAST output and
then find the corresponding SAM output file, with the purpose of matching the
enriched GO categories to the actual significant genes that enriched them,
denoting the number of genes that were found to be up-regulated and down-
regulated in a final count.'''


from __future__ import division
import csv

def up_down_gene_counter(GOID, genesset, GO_up_reg, GO_down_reg, up_reg_genes, down_reg_genes):
    '''Takes the counts of up and down-regulated genes for all genes for a
    given GOID'''
    genes = []
    #print row[0]
    genes.append(genesset)
    #print row[7].strip().split("//")
    GO_up_reg[GOID] = 0
    GO_down_reg[GOID] = 0
    for geneset in genes:
        while (len(geneset)>0):
            probe = "¥""+geneset.pop().strip()+"¥""
            #print probe
            if probe in up_reg_genes:
                #print "upreg!"
                if GOID in GO_up_reg:
                    GO_up_reg[GOID] = GO_up_reg[GOID] + 1
                else:
                    GO_up_reg[GOID] = 1
            if probe in down_reg_genes:
                #print "downreg!"
                if GOID in GO_down_reg:
                    GO_down_reg[GOID] = GO_down_reg[GOID] + 1
                else:
                    GO_down_reg[GOID] = 1

    return GO_up_reg, GO_down_reg

def GOEAST_loader(open_file):
    '''Loads the GOEAST file into a list'''
    header = 0
    GOEAST_data_array = []
    datafile = open(open_file, 'r')
    for row in datafile:
```

```
        if header == 0:
            header = 1
        else:
            GOEAST_data_array.append(row.strip().split('¥t'))

    return GOEAST_data_array

def SAM_output_loader(open_file):
    '''loads corresponding SAM output file'''
    header = 0
    directories = {}
    up_reg_genes = []
    down_reg_genes = []
    CSV_name_file = open("../FileNames.txt")
    for directory in CSV_name_file:
        directories[directory.strip().split("/")[-1].split("_")[0]] = directory.strip()
    test_name = open_file.strip().split(".txt")[0].split("_")[0]
    print "Parsed: " + test_name

    datafile = open("../"+directories[test_name], 'r')
    for row in datafile:
        if header == 0:
            header = 1
        else:
            gene = row.strip().split(",")[1]
            score = float(row.strip().split(",")[2].strip("¥""))
            if score > 0:
                up_reg_genes.append(gene)
            elif score < 0:
                down_reg_genes.append(gene)

    return up_reg_genes, down_reg_genes


def count_up_reg(GO, gene):
    '''Adds to GO dictionary if the gene is found to be upregulated'''
    if gene in probes_and_counts:
        probes_and_counts[gene] = probes_and_counts[gene] + 1
    else:
        probes_and_counts[gene] = 1



if __name__ == '__main__':

    #Shell code to get FileNames.txt: find . -name *GOs.txt > FileNames.txt
    filenamefile = open('GO_FileNames.txt','r')

    output = [["Stage Comparion","GOID","Term","#Up","#Down","Total"]]
```

```
    for filename in filenamefile:
        print filename
        GO_up_reg = {}
        GO_down_reg = {}
        genes = []
        header = 0
        current_file_name = filename.strip().split('/')[-1]
        input_GOEAST_file = GOEAST_loader(current_file_name)
        up_reg_genes, down_reg_genes = SAM_output_loader(current_file_name)
        for row in input_GOEAST_file:
            GO_up_reg, GO_down_reg =
up_down_gene_counter(row[0],row[7].strip().split("//"),GO_up_reg,GO_down_reg,up_reg_genes,
down_reg_genes)

            output.append([current_file_name.strip().split(".txt")[0].split("_")[0], row[0],
row[2],GO_up_reg[row[0]], GO_down_reg[row[0]],row[3]])

    outfile = open('GOs UpDown Reg Gene Counts Masked.csv','wb')
    outwriter = csv.writer(outfile)
    outwriter.writerows(output)
    outfile.close()

    print "Output saved as GOs UpDown Reg Gene Counts.csv"

#!/usr/bin/env python
#genes_2_enrich_GOs.py
'''Counts the number of genes used to enrich GOs.'''

import os

def genes_2_enrich_counter(filename, gene_set):
    temp_list = gene_list_getter(filename)
    #print len(temp_list)
    for generow in temp_list:
        while (len(generow)>0):
            #print generow
            current_gene = generow.pop().strip()
            #print current_gene
            gene_set[current_gene] = 1

    return gene_set

def gene_list_getter(filename):
    temp_list = []
    datafile = open(filename,'r')
    header = 0
    for row in datafile:
        if header == 0:
```

```
            header = 1
        else:
            temp_list.append(row.strip().split("¥t")[8].strip().split("//"))

    return temp_list


if __name__ == '__main__':

    filenamefile = open('FileNames.txt','r')

    for filename in filenamefile:
        current_file_name = filename.strip().split('¥n')[0]
        #print current_file_name
        empty_gene_set = {}
        gene_set = genes_2_enrich_counter(current_file_name, empty_gene_set)

        print "For " + current_file_name + ": " +str(len(gene_set))
```

Sorghum analyses code

```
#!/usr/bin/env python
#TAIR_gene_searcher.py
'''Uses the gene search tool at the top of the TAIR website to query a list of
genes, and then returns the top hit for the search for each gene'''

import sys, os, time, urllib, optparse
import cStringIO, xml

import common, drv_fasta

def query(gene):
    #"seq -> htmlStr"
    cgiArgs = {}
    site =
"http://www.arabidopsis.org/servlets/Search?type=general&search_action=detail&method=1&show_obsol
ete=F&name="+gene+"&sub_type=gene"
    params = urllib.urlencode(cgiArgs)
    while True:
        try:
            f = urllib.urlopen(site, params)
        except IOError, a:
            print "network error: %s" % str(a)
            print "retry..."
            continue
        else:
            break

    return f.read()
```

```
datafile = open("/Users/Lamarck/Documents/2009-2011 Utah
State/Dissertation/Code/BioPython/genes_2_search.txt", 'r')

for g in datafile:
    g_gene = g.strip()
    htmlStr = query(g_gene)
    open('/Users/Lamarck/Documents/2009-2011 Utah
State/Dissertation/Code/BioPython/genesearch/'+g_gene+'.txt','w').write(htmlStr)
    print "finished: "+g_gene


#!/usr/bin/env python
#TAIRblasterParser.py
'''Module by Jonathan Cardwell that takes list of sequences and directory of
TAIR BLAST results to a CSV file with all BLASTed sequences alongside
the best TAIR blast locus result and e-value.  Creates 3 files: 1) TAIR BLAST
results; 2) text file with all TAIR loci to be used as the background file;
and 3) text file with loci of genes found to be significant.'''

if __name__ == '__main__':
    print "Does not run on its own.  Run TAIR_Blaster.py."

import csv

def tairBP(seqs, output, siggene_file):
    '''function for consolidating txt results files, parsing, and then
    creating the final results, CSV file, and the two files needed for
    AMIGO enrichemnts'''
    temp = []
    temp2 = temp[:]
    background_loci = []
    siggenes_loci = []
    siggenes = []
    for siggene in siggene_file:
        siggenes.append(siggene.strip())
    for x in seqs:
        current_seq_file = x+'-seq.txt'
        datafile = open(current_seq_file,'r')
        datareader = csv.reader(datafile)
        for row in datafile:
            temp.append(row.strip().split('.'))
            temp2.append(row.strip().split(' '))
        background_loci.append([temp[21][0]])
        if x in siggenes:
            siggenes_loci.append([temp[21][0]])
        if temp2[21][-1][0]=='e': #special case if eval has scientific notation
            output.append([x,temp[21][0],str('1'+temp2[21][-1])])
        else:
            output.append([x,temp[21][0],temp2[21][-1]])
```

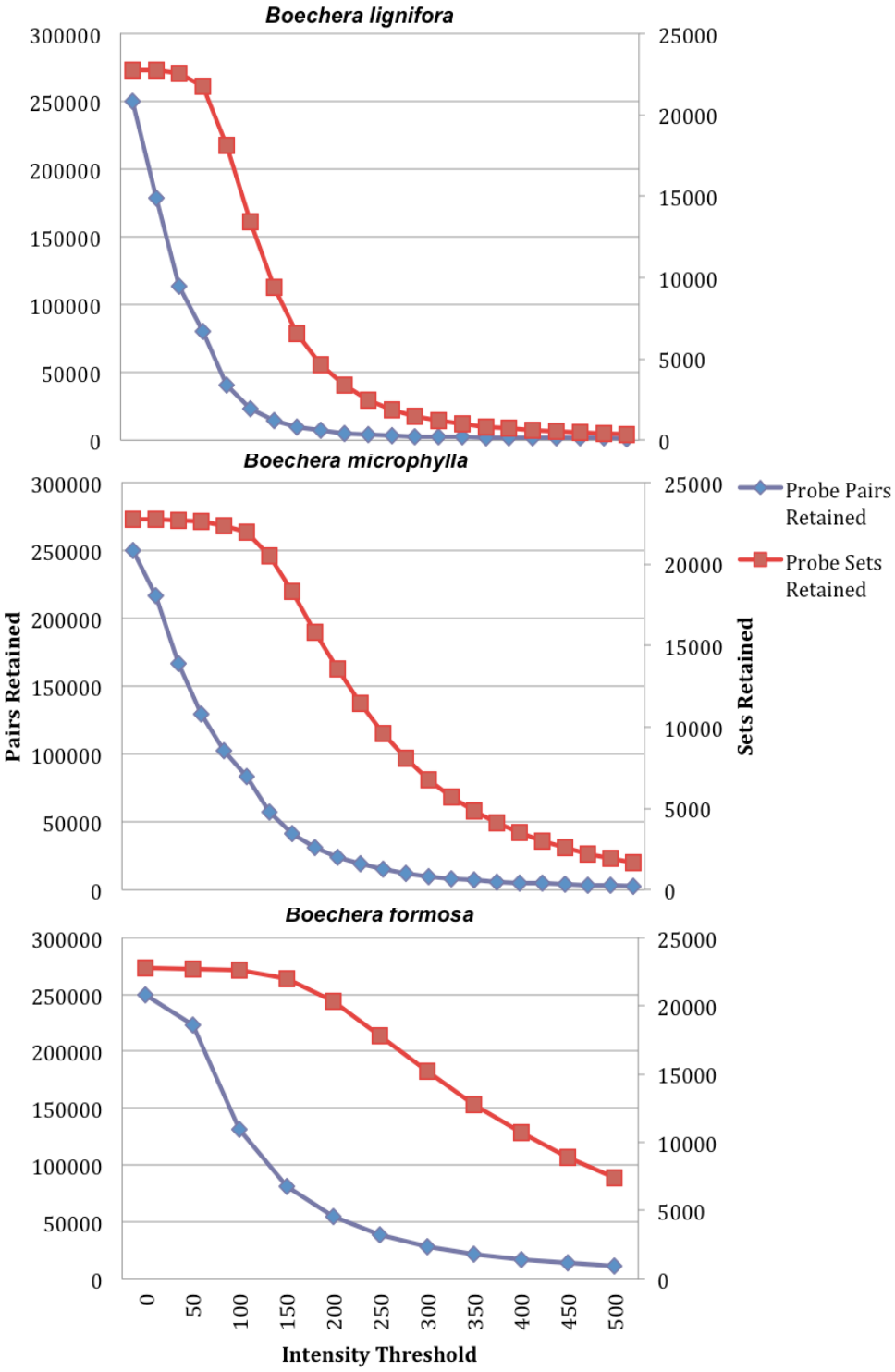
```
        temp = []
        temp2 = []
        datafile.close()

    make_files(output,siggenes_loci,background_loci)

def make_files(output, siggenes_loci, background_loci):
    output_file = open('TAIR_BLAST_Results.csv','wb')
    datawriter = csv.writer(output_file)
    datawriter.writerows(output)
    output_file.close()

    siggenes_loci_file = open('siggenes_loci.txt','wb')
    datawriter = csv.writer(siggenes_loci_file)
    datawriter.writerows(siggenes_loci)
    siggenes_loci_file.close()

    background_loci_file = open('background_loci.txt','wb')
    datawriter = csv.writer(background_loci_file)
    datawriter.writerows(background_loci)
    background_loci_file.close()

    print "Created ¥'TAIR_BLAST_Results.csv¥', ¥'siggenes_loci.txt¥', and ¥'background_loci.txt¥'"

'''Except from modified "tair_fasta.py" script written by KBI: Kashiwa Bioimaging.  Downloaded from
http://hasezawa.ib.k.u-tokyo.ac.jp/zp/Kbi/AtBlasts.  Script was modified to run my
TAIRblasterParser.py script.
    output = [["SeqID","TAIR_locus","evalue"]]
    siggene_file = open(siggeneFile,'r')
    TAIRblasterParser.tairBP(seqs,output,siggene_file)
    print "Sequences blasted: " + str(len(seqs))
    print "Begin at %s, End at %s" % (startTime, common.getIsoTime())
    print "Elapsed time: " + str(round(time.time() - newStartTime,2)) + " secs"
    return 0
```

Excerpt of "Xspecies" modification in Perl:

```
        # Replace old list with new list
        $probes->{$probename}->probe_pairs($newprobepairs);

        # If a probeset has no probes left, we have to remove the entire probeset

        $evenP = (scalar @{$probes->{$probename}->probe_pairs()});
        $evenP2 = $evenP/2;

        if (($evenP2==5)||($evenP2==4)||($evenP2==3)){
                $random_number = int(rand($evenP));
                splice(@{$probes->{$probename}->probe_pairs()},$random_number,1);
                $num_probes_removed++;
                $num_probes_kept--;
```

```
}

if ($evenP<5) {
    $num_probesets_removed++;
    delete $probes->{$probename};
} else {
    $num_probesets_kept++;
            }
```

Appendix figure 1 Plots used in determination of optimal intensity thresholds for masking by species (in bold). Optimization was considered the highest amount of probe set and probe pair retention. Combined CDF masking file was created from the joint usage of each species' optimal threshold. As no gDNA arrays were available for *B. stricta*, it is subsequently not represented here. The combined file of *lignifora, microphylla,* and *formosa* was used for all masking analyses. For instructions on creating a masked CDF, refer to NASCArrays:Xspecies, 2010.



Appendix Table 1 Tabular version of the determination of optimal probe masking CDF threshold. Rows in bold were the selected thresholds utilized for masking.

Threshold	Probe Pairs Retained	Probe Sets Retained	% Sets Kept	% Pairs Kept
Lignifora				
50	***178871***	***22717***	***99.87%***	***71.52%***
75	113820	22521	99.01%	45.51%
100	79973	21741	95.58%	31.98%
150	40366	18096	79.56%	16.14%
200	23428	13426	59.03%	9.37%
250	14672	9382	41.25%	5.87%
300	9841	6546	28.78%	3.93%
350	7011	4636	20.38%	2.80%
400	5243	3354	14.75%	2.10%
450	4117	2479	10.90%	1.65%
Microphylla				
50	216432	22739	99.97%	86.54%
75	166427	22703	99.81%	66.54%
100	***129170***	***22602***	***99.37%***	***51.65%***
125	102607	22381	98.40%	41.03%
150	83238	21947	96.49%	33.28%
200	57440	20517	90.20%	22.97%
250	41540	18293	80.42%	16.61%
300	31089	15821	69.56%	12.43%
350	23992	13575	59.68%	9.59%
400	18833	11449	50.33%	7.53%
450	15024	9570	42.07%	6.01%
500	12236	8053	35.40%	4.89%
Formosa				
50	223630	22739	99.97%	89.42%
100	***131253***	***22631***	***99.49%***	***52.48%***
150	81490	21953	96.51%	32.58%
200	54587	20305	89.27%	21.83%
250	38794	17799	78.25%	15.51%
300	28507	15194	66.80%	11.40%
350	21685	12774	56.16%	8.67%
400	16961	10678	46.94%	6.78%
450	13561	8914	39.19%	5.42%
500	10971	7409	32.57%	4.39%