Pacific Symposium on Biocomputing 13:141-152(2008)

SGDI: SYSTEM FOR GENOMIC DATA INTEGRATION *

V. J. CAREY[†], J. GENTRY[‡], D. SARKAR[§], R. GENTLEMAN[¶], S. RAMASWAMY[‖]

Channing Laboratory, Brigham and Women's Hospital
Harvard Medical School
181 Longwood Avenue,
Boston, MA 02115, USA
E-mail: stvjc@channing.harvard.edu

This paper describes a framework for collecting, annotating, and archiving high-throughput assays from multiple experiments conducted on one or more series of samples. Specific applications include support for large-scale surveys of related transcriptional profiling studies, for investigations of the genetics of gene expression and for joint analysis of copy number variation and mRNA abundance. Our approach consists of data capture and modeling processes rooted in R/Bioconductor, sample annotation and sequence constituent ontology management based in R, secure data archiving in PostgreSQL, and browser-based workspace creation and management rooted in Zope. This effort has generated a completely transparent, extensible, and customizable interface to large archives of high-throughput assays. Sources and prototype interfaces are accessible at www.sgdi.org/software.

1. Introduction

It is becoming increasingly clear that biomarker and molecular target discovery in cancer, for example, will require the integrative analysis of multiple datasets generated in different centers, at different times, using different technology platforms. In fact, recent work suggests that integrative approaches can be highly useful for molecular target discovery [9, 11, 12], but there are still significant hurdles at the level of dataflow and data

*This work is supported in part by DFCI/HCC SPORE in Breast Cancer 2P50 CA89393-07.
[†]Channing Lab
[‡]Massachusetts General Hospital, Harvard Medical School
[§]Fred Hutchinson Cancer Research Center
[¶]Fred Hutchinson Cancer Research Center
[‖]Massachusetts General Hospital, Harvard Medical School

Pacific Symposium on Biocomputing 13:141-152(2008)

analysis workflow architecture, and deficiencies in software infrastructure, that retard progress in this research area. A very recent *Nature Reviews in Genetics* Perspectives report [8] discusses disparities between standard approaches to databasing genomic data and metadata and requirements of systems biology. Among the issues identified are deficiencies in meta-information necessary for resource discovery (by humans or by software), impoverishment of search predicate formulation options, unavailability of scalable/programmatic query resolution for queries with large payloads, non-robustness of client applications to alterations in central server data management patterns, resistance to adoption of XML markups (necessitating detailed non-generic parser development efforts), inappropriate conceptualizations (e.g., functions should be predicated of gene products, not genes, owing to splice variation) and a variety of difficulties related to communication, education, and licensing shortfalls.

To address some of these limitations, we have designed, developed, and deployed a software infrastructure for the storage and integrative analysis of biological data generated with high-thoughput tools in genomics and proteomics (www.sgdi.org/software). The proposed System for Genomic Data Integration (SGDI) is locally customizable. This is in contrast to read-only analysis-oriented repositories such as Oncomine [10], WebQTL [3], or SAGE-Genie [6], SGDI fills a critical gap in prevalent bioinformatics infrastructure, by permitting individual investigators to perform integrative analyses of unpublished data and to easily share unpublished data with colleagues, in a formally documented and auditable framework. In addition, researchers will be able to integrate their latest private data with a myriad of other publicly available data streams, thereby ensuring the greatest use of available resources. SGDI will enable integrative studies that are currently time-consuming and are difficult to standardize. It will facilitate data sharing and data reuse and will allow for data collected in one set of circumstances to be used to help test hypotheses in related areas. This system has been purpose-designed to enable sharing and analysis of private datasets that are generated either in single laboratories of through multi-investigator collaborations such as SPORE programs and program-project grants (PPGs).

While the ultimate objective of SGDI is an investigator-oriented browser-driven interface, we have adopted an approach that permits programmatic access to and manipulation of all data and metadata collected in the system. In this paper, we focus on elementary architecture and component functionalities. The first section details Bioconductor's approach to

Pacific Symposium on Biocomputing 13:141-152(2008)

coherent container design for multiple high-throughput assays applied to fixed series of samples. The second section describes the sample annotation problem and SGDI's ontoElicitor facilities for structuring and deploying regimented vocabularies for sample characteristics. The third section describes the reporter annotation problem and SGDI's reporter query facilities. The final section provides illustrations of the integrated framework and discusses future intentions of the project.

2. Integrative data structure design in Bioconductor

Consider the problem of representing the fully preprocessed and normalized data from an experiment in genetics of gene expression, as reported in Cheung et al[4]. Let G denote the number of mRNA reporters (e.g., the number of oligonucleotide probe sets in an Affymetrix(TM) microarray), let N denote the number of samples (e.g., the number, 58, of CEPH CEU founders studied by Cheung et al.), let S denote the number of SNPs genotyped on each of the N samples, and let r denote the number of clinical, demographic, and technical variables recorded on the N samples. mRNA abundance measures are recorded in a $G \times N$ table, genotype calls (unphased) are recorded in an $S \times 2N$ table, and clinical and demographic characteristics of the N individuals are recorded in an $N \times r$ table. For the analyses reported in Cheung et al., genotyping information is condensed into SNP-specific rare allele counts, where allele rarity is reckoned relative to the source population, necessitating only an $N \times S$ table.

Some basic premises of the Bioconductor approach to dealing with high-throughput data are now described. We use the symbol `X` to name a concrete container for experimental data; the term *phenodata* is used to refer to all information gathered on samples exclusive of the assay results.

Compact representation. All the information collected in a high-throughput experiment should be available in a single object.

Tight binding of phenodata to assay data. Sample-level information should be tightly bound to assay results and should be propagated through workflows along with assay results unless intentionally excluded.

Array-like selection; closure of container type under selection. The idiom `X[G, S]` in the R programming language can be used to derive a new instance of the container type of `X` restricted to data on reporters identified in the general predicate expression `G` and to samples identified in the predicate expression `S`.

Tightly bound metadata components available. Representations allow for

Pacific Symposium on Biocomputing 13:141-152(2008)

Table 1. Selected methods and operators for Bioconductor containers. Most of the infrastructure for managing sample-level data is defined for the `eSet` class and is inherited to specializations.

method example	purpose	replace?
`eSet` class		
`X$n`	obtain value for all samples	yes
`X[i,j]`	restrict to selection	yes
`abstract(X)`	return main publication abstract	no
`experimentData(X)`	return MIAME schema	yes
`featureData(X)`	return reporter metadata	yes
`phenoData(X)`	return sample-level data	yes
`varMetadata(X)`	return metadata on sample attributes	yes
`ExpressionSet` class		
`exprs(X)`	return matrix of assay results	yes
`makeDataPackage(X)`	create an installable R package	no
`racExSet` class		
`snps(X)`	return matrix of rare allele counts	yes
`snpNames(X)`	return SNP identifiers	yes
`cghExSet` class		
`cloneNames(X)`	return clone identifiers	no
`cloneMeta(X)`	return clone metadata	no
`logRatios(X)`	return CGH assay results	no

storage of additional (meta)data on the experiment (following the MIAME [1] schema) and definitions of attributes defining reporters or samples.

Exemplary published experiments should be instantiated for distribution as illustrations. See the Bioconductor packages *Neve2006* (CGH+expression, discussed below) and *GGtools* (whole genome SNP+expression).

Generic workflow operations. Methods development in Bioconductor consists primarily of defining parameterized methods `f()` that interrogate and transform experimental data to support biological inference through evaluations of `f(X, ...)`. Multiassay representations should inherit type information from the constituent container types so that generic operations continue to function for the extended container type.

The main abstract class used to define high-throughput containers is called `eSet`, defined in the *Biobase* package of Bioconductor. Expression microarray assay results and allied sample and metadata are stored in instances of the `ExpressionSet` class. Table 1 sketches some of the methods/operations defined for `eSet` and some of its descendants for expression and integrative experiments.

Pacific Symposium on Biocomputing 13:141-152(2008)

3. Sample annotation; ontoElicitor

Careful analysis of the relationship of genomic phenomena to phenotypic or clinical condition requires detailed description of phenotypic state of the sample assayed. The data from Neve's 2006 analysis of copy number and expression variation in breast tumor cell lines [7] are a good illustration of the sort of material published in this area. Here we excerpt two records from the sample annotation:

```
> library(Neve2006); data(neveExCGH)
> pData(neveExCGH)[1:4,]
       ind cellLine geneCluster ER  PR HER2 TP53
600MPE   1   600MPE          Lu  + [-] <NA>    -
AU565    2    AU565          Lu  - [-]    + <NA>
       Source tumorType Agey Ethnicity     cultMedia
600MPE   <NA>       IDC   NA      <NA>   DMEM,10%FBS
AU565      PE        AC   43         W RPMI, 10% FBS
              cultCond commonPt reductMamm
600MPE     37c, 5% CO2        0      FALSE
AU565      37c, 5% CO2        1      FALSE
> table(neveExCGH$Source)
  AF  CWN P.Br   PE   PF   Sk
   2    1   24   19    0    1
> varMetadata(neveExCGH)["Source",]
[1] "PE = pleural effusion, P.Br = primary breast,
     Sk = skin, CWN = chest wall nodule, AF = ascites fluid"
```

This illustrates Bioconductor facilitites for accessing and interpreting sample-level data. The `pData` method extracts the R data frame of attributes on samples, the `$` operator confers direct access to variable values, and the `varMetadata` method returns a subsettable data frame with definitions of symbols used.

When different nomenclatures are used for phenotype characterization in different experiments, a problem arises for users of public microarray archives who wish to perform synthetic analyses [5]. It becomes difficult to align samples across experiments. Figure 1 illustrates the situation in a collection of 25 breast cancer microarray experiments. Sample-level data available in public archives were reviewed. The union of the sets of terms employed for sample annotation was formed, and the subset of terms related to histopathology was selected. The left margin of Figure 1 lists all the

Pacific Symposium on Biocomputing 13:141-152(2008)

terms in this set, and the bottom margin lists the experiments. A dark square is plotted in cell (i, j) of the figure if term i is used in experiment j. It is clear that terms with similar meanings are not uniformly named, and that experimenters often do not report values of many relevant characteristics.

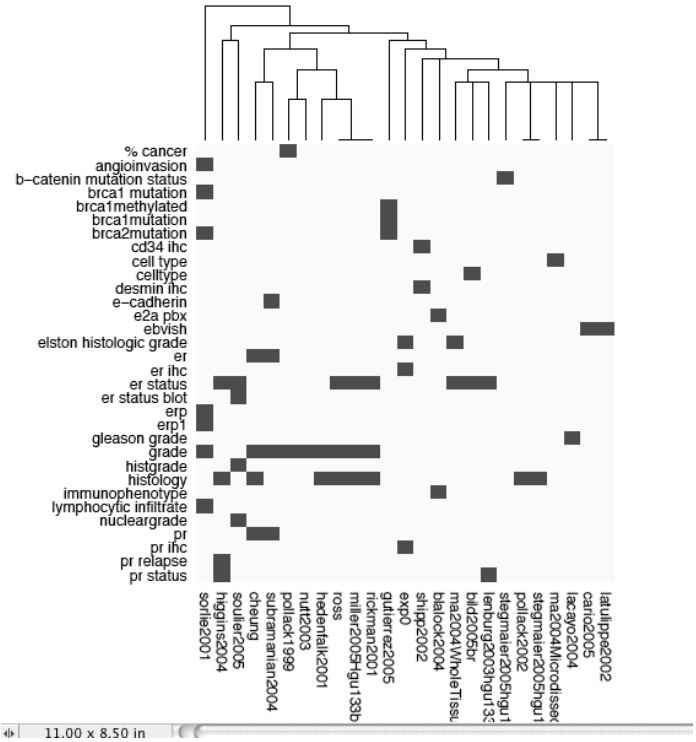


Figure 1. Rows: terms related to breast cancer histopathology. Columns: author-date tokens identifying 25 published breast cancer datasets. A dark square is plotted at location (i,j) if study i uses term j in characterizing its samples.

While Figure 1 indicates a problem with sparsity of shared annotation across independently performed experiments, it does not indicate another vulnerability: Even when experimenters do use a common term such as 'grade' in sample annotation, the values used for the term may not coincide.

SGDI has responded to this predicament with two novel tools. The first, ontoElicitor, is a simple framework for iteratively presenting and receiving feedback on a proposed structured vocabulary for sample annotation. Figure 2 illustrates a facet of the ontoElicitor for breast cancer samples.

Pacific Symposium on Biocomputing 13:141-152(2008)

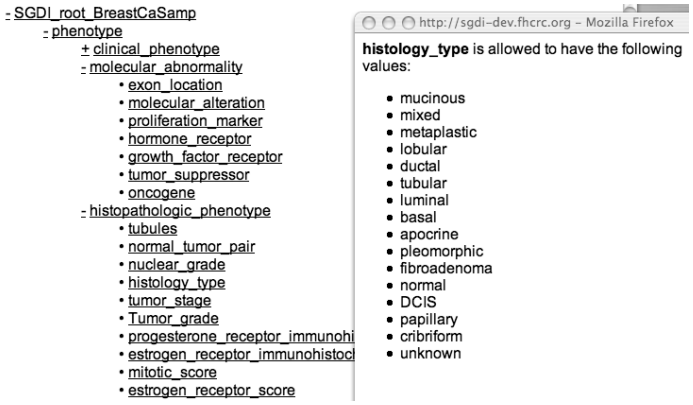


Figure 2. ontoElicitor facet for breast cancer, with expanded value set for histology type displayed.

Our *current* approach to vocabulary design and management eschews formal ontology engineering methodologies like OWL/RDF in favor of R graphs. The OWL concepts of class, property and individual are typically not familiar to experimentalists, and adaptation of OWL technology for *elicitation* and *revision* of vocabularies and valuations required in microarray archives does not seem cost-effective. We have found that practitioners are interested in working with tree-structured displays of terms, with enumerated valuations, and with valuation classes such as "numeric" or "string". Bioconductor graph structures can easily represent trees of nodes that represent terms as string literals. Because arbitrary node attributes can be attached, valuations and valuation classes can be bound directly to terms in the graph structures. These ontology graph structures, defined in the *ontoElicitor* package distributed with SGDI, can be serialized to HTML (for use in the ontoElicitor application) or CSV (for review in Excel by practitioners.) Note that we will support conversion between OWL/RDF ontology models and R ontology graphs upon adoption of a suitable RDF schematization for sample-level metadata. The *Rredland* package of Bioconductor exposes the `librdf.org` facilities for parsing, modeling, and archiving RDF.

The second tool of use in promoting adoption of uniform sample annotation is the phenoData editor application, with a demonstration instance at the SGDI portal. Given an ontoElicitor-derived ontology, the phenoData editor generates a page of fields with drop-down menus that are used to populate a sample attribute table with standardized values.

Pacific Symposium on Biocomputing 13:141-152(2008)

4. Reporter annotation and query facilities

Focused use of archives of high-throughput data is most convenient when genomic contexts and biological roles of reporters are easily established. In the case of SNP+expression experiments, it will be of interest to know relative locations of genotyped loci, assayed transcripts, and, e.g., locations of promoters for genes exhibiting differential expression; for CGH+expression, segmentation breakpoints need to be related to gene locations and phenotype. Substantial information on element locations is available through Bioconductor platform annotation packages and through translations of Entrez Gene and biomart-accessible annotation resources. It is frequently of interest to interrogate using higher-level concepts and gene collections. Figure 3 illustrates the interface for filtering reporters on the basis of membership in specific KEGG-catalogued pathways; GO categories and sets of HUGO symbols may be used as well. We also have recently introduced an R graph representing the KEGG orthology (a tree-structured hierarchy of KEGG pathways, package *keggorth*) and tree-based navigation of this structure will be supported.

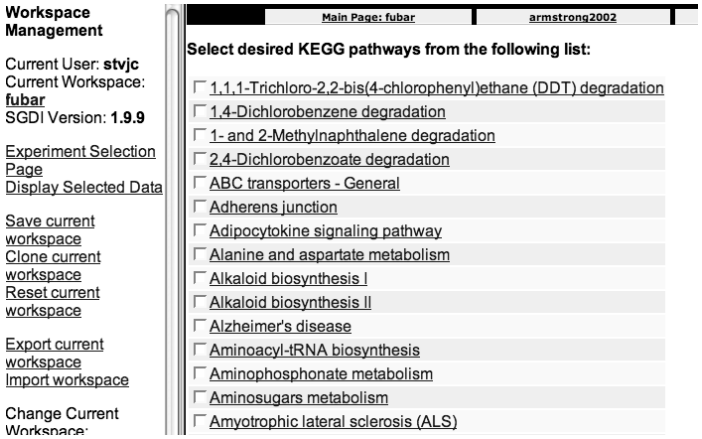


Figure 3. Selection of reporters using KEGG pathway catalog.

5. The integrated interface; use cases

The primary object that is manipulated in the SGDI framework is the workspace. This is an XML document that records all selections that have

Pacific Symposium on Biocomputing 13:141-152(2008)

occurred. Workspaces can be exported for sharing with colleagues, can be cloned so that multiple paths with common initial segments can be explored and saved, and can be revised through rollback or continuation. In general, a user will not be concerned with the contents or structure of the *workspace document*, but will work with the system to define a data extract that will be used for downstream analysis.

Figure 4 gives a view of the workspace obtained when three experiments are in scope. armstrong2002 and blalock2004 are classical breast cancer expression array experiments; testOGTES is a test instance of expression data (obtained on the u133x3p platform) and SNP data (obtained with the Affymetrix(TM) 500K Nsp+Sty platform). Expression assay results and standard errors of estimated expression are provided in two tables; enzyme-specific tables are provided for both the genotype calls and the call confidence as measured by the *crlmm* algorithm in development by Carvalho, Irizarry and colleagues [2].

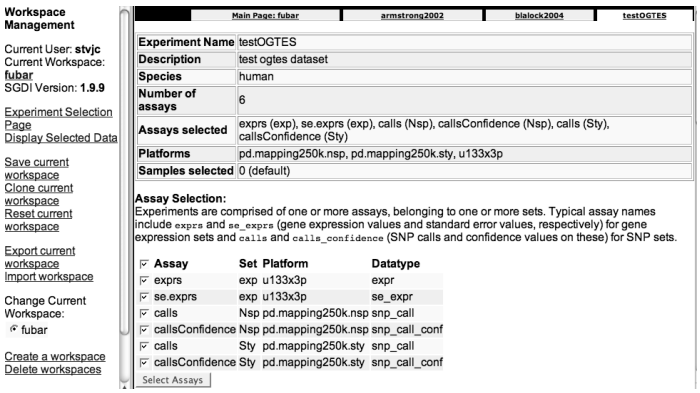


Figure 4. top level interface

Figure 5 depicts the interface to SNP selection using only physical coordinates on chromosomes. Additional facilities are available to employ annotation provided by Affymetrix detailing cytoband, harboring transcript, harboring gene, role of transcript in gene to form and condition queries. The exposition of these resources to simplify interrogation is complete for cytoband and gene relationships; more work is needed to take advantage of the detailed contextual vocabulary described in section 4 above.

Finally, a partial view of the HTML rendering of a workspace display for genotyping assays is given in Figure 6. Reporter metadata occupies the

Pacific Symposium on Biocomputing 13:141-152(2008)

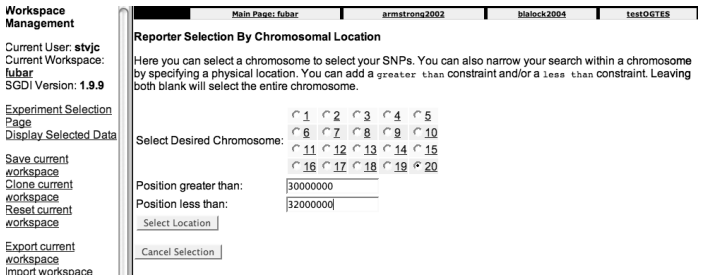
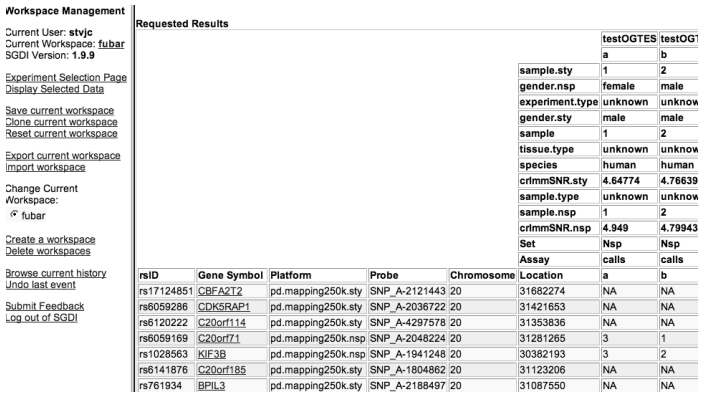


Figure 5. Selecting SNPs by location on chromosome.

first six columns, and sample characteristics occupy the first 13 rows. Some genotype calls are found at the lower right corner of the display.



						testOGTES	testOG
						a	b
					sample.sty	1	2
					gender.nsp	female	male
					experiment.type	unknown	unknow
					gender.sty	male	male
					sample	1	2
					tissue.type	unknown	unknow
					species	human	human
					crlmmSNR.sty	4.64774	4.76639
					sample.type	unknown	unknow
					sample.nsp	1	2
					crlmmSNR.nsp	4.949	4.79943
					Set	Nsp	Nsp
					Assay	calls	calls
rsID	Gene Symbol	Platform	Probe	Chromosome	Location	a	b
rs17124851	CBFA2T2	pd.mapping250k.sty	SNP_A-2121443	20	31682274	NA	NA
rs6059286	CDK5RAP1	pd.mapping250k.sty	SNP_A-2036722	20	31421653	NA	NA
rs6120222	C20orf114	pd.mapping250k.sty	SNP_A-4297578	20	31353836	NA	NA
rs6059169	C20orf71	pd.mapping250k.nsp	SNP_A-2048224	20	31281265	3	1
rs1028563	KIF3B	pd.mapping250k.nsp	SNP_A-1941248	20	30382193	3	2
rs6141876	C20orf185	pd.mapping250k.sty	SNP_A-1804862	20	31123206	NA	NA
rs761934	BPIL3	pd.mapping250k.sty	SNP_A-2188497	20	31087550	NA	NA

Figure 6. Reporting on selected SNPs.

6. Deployment; conclusions

One of the most significant problems tackled by SGDI is the challenge of providing fine-grained, investigator-friendly access to preprocessed and carefully annotated archives of high-throughput data. SGDI allows investigators to discover (using flexible but standardized query resolution) and extract (using a browser-based workflow) data on values of specific reporters associated with samples possessing specific phenotypic or experimental characteristics for their own local analysis. As the public instance of SGDI grows, this "read-only" facility will provide access to public datasets