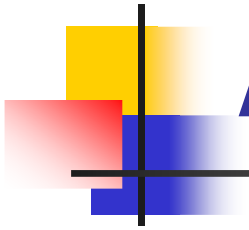
AMOS : A Modular Open Source Assembler

Michael Schatz

Center for Bioinformatics and Computational Biology
University of Maryland

August 13, 2006
University of Hawaii



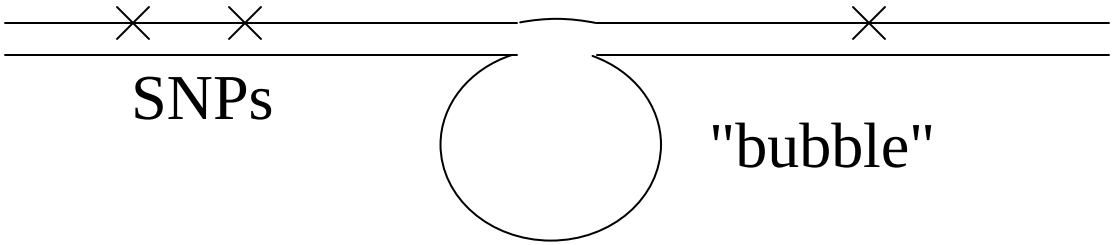
AMOS Goals

- Open Source Assembly Package
  - http://amos.sourceforge.net

- Modular design
- Well defined input/output formats
- Flexibility in building “pipelines” to attack next generation assembly challenges.
- General use: does not depend on databases, proprietary data formats, specialized hardware, etc.

Novel assembly challenges

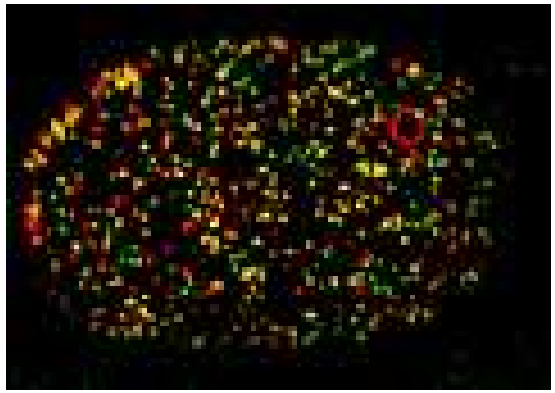
- New sequencing technologies require new assembly algorithms
- Complex genomes pose new challenges
  - High repeat content (e.g. Entamoeba hystolitica ~ 25% reads in exact repeats - thousands of copies each)
  - Large differences between homologous chromosomes (e.g. Ciona savigny ~ 20% divergence)



- Environmental samples (e.g. bacteria in human gut) - Lander-Waterman statistics no longer apply. Different representation of members of environment

The future of sequencing

Massively parallel sequencing

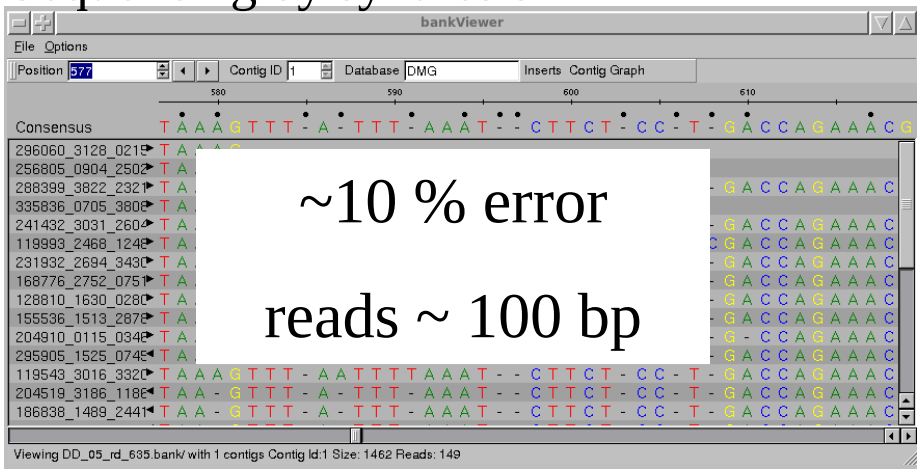


http://arep.med.harvard.edu/

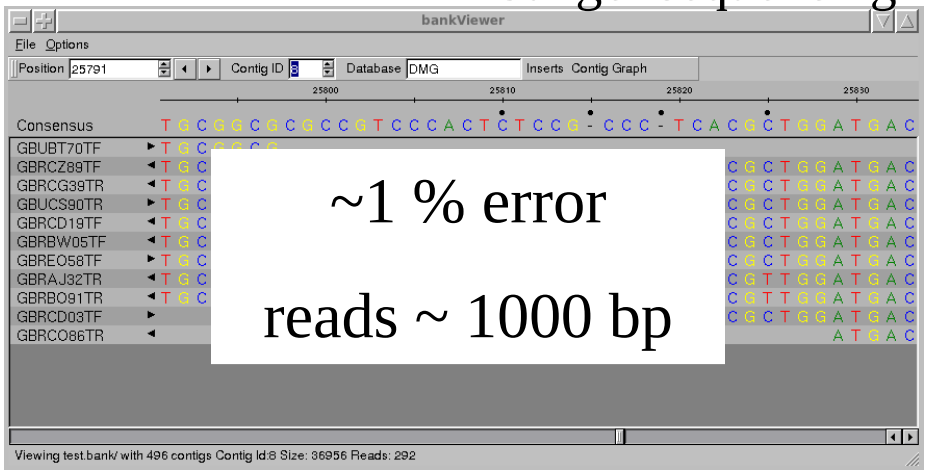
- each spot is a molecule or amplified from one molecule
- image processing used to track molecules during sequencing by synthesis
- often micro-fluidics/lab-on-a-chip used

Impact on assembly

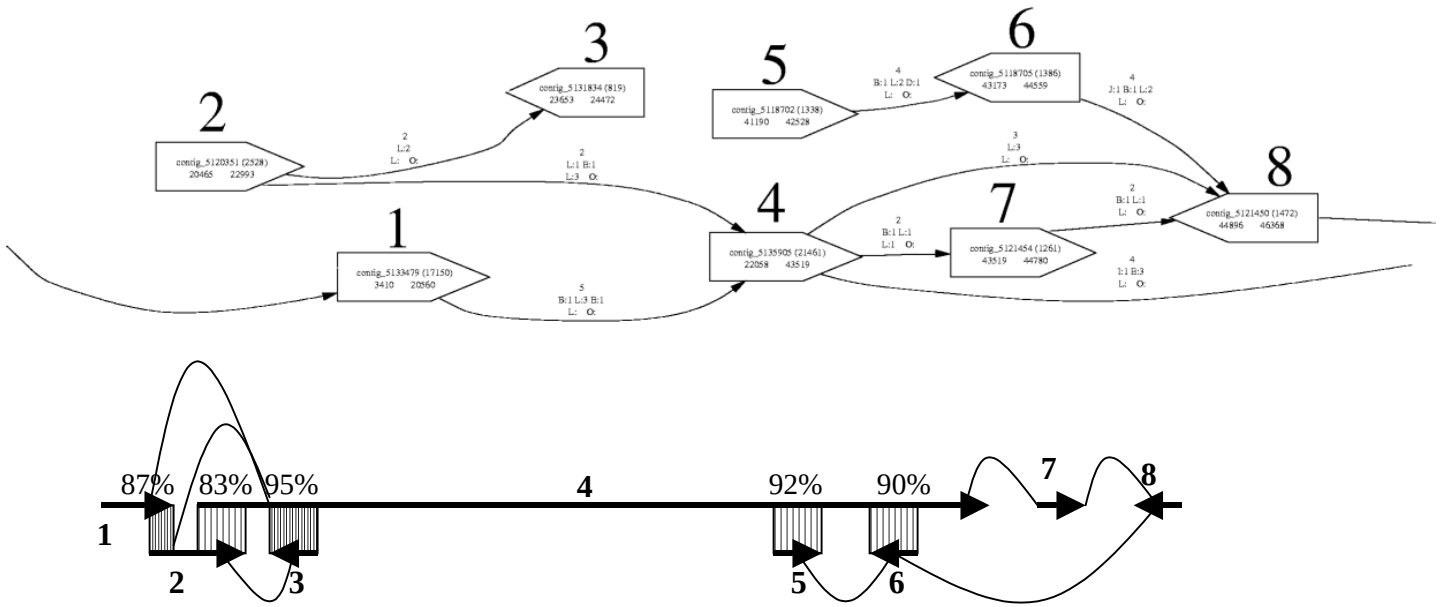
Sequencing by synthesis



Sanger sequencing

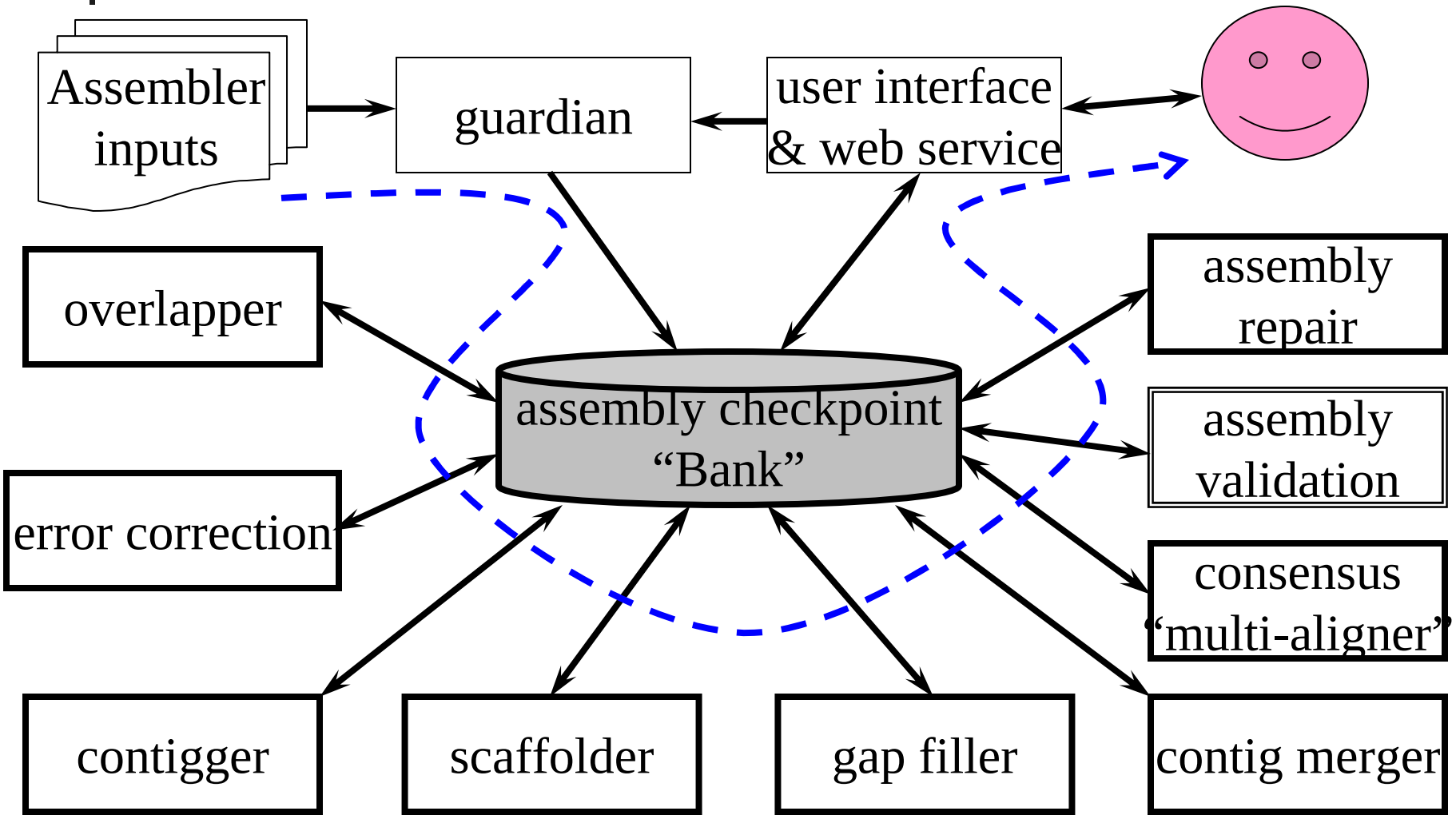


Haplotypes Difference

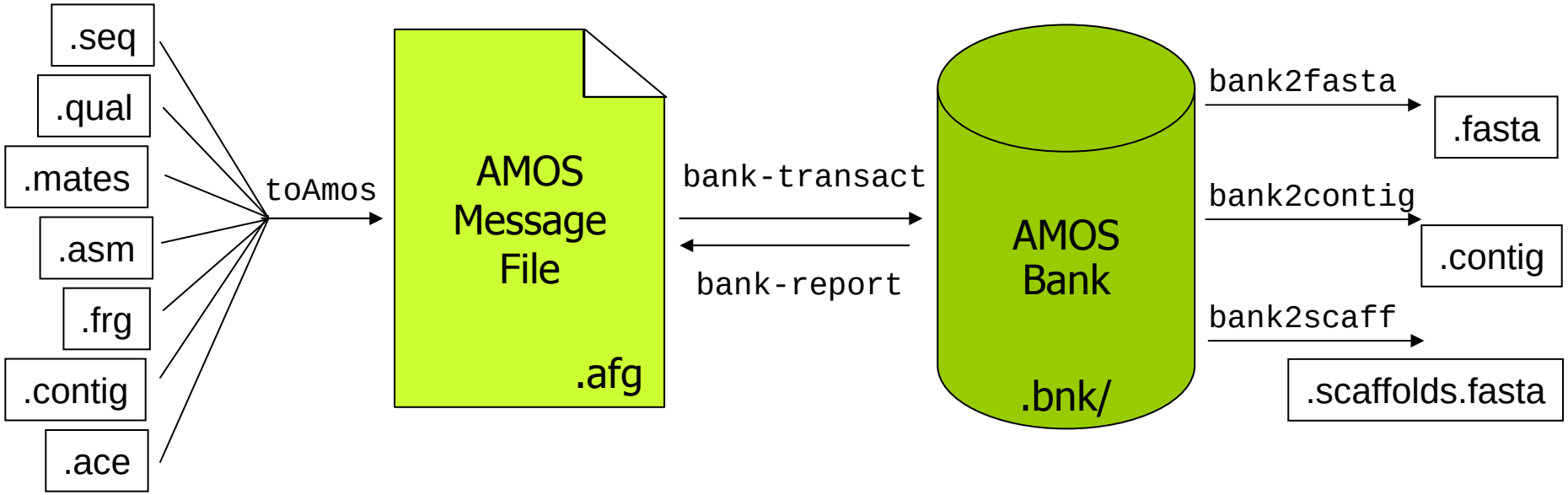


Question: How to represent such data? (e.g. for Blast)

AMOS modules



Assembly Data Conversions



CA Assembly w/ Surrogates to AMOS Message File (.asm, .frg)

```
$ toAmos –a prefix.asm –f prefix.frg –o prefix.afg –S
```

Finished Assembly to AMOS Message File (.contig, .frg)

```
$ toAmos –f prefix.frg –c prefix.contig –o prefix.afg
```

AMOS Message File to Bank

```
$ bank-transact –m prefix.afg –b prefix.bnk -c
```

An AMOS pipeline (AMOScmp)

```
#!runAmos -C

#-------------------------------------- USER DEFINED VALUES -----------------#
# allow input to be either <file>.afg or just <file>
REF = $(PREFIX).1con
TGT = $(strip .afg PREFIX).afg
#----------------------------------------------------------------------------#

BINDIR   = /usr/local/bin
NUCMER   = $(shell which nucmer)

SEQS     = $(PREFIX).seq
BANK     = $(PREFIX).bank
ALIGN    = $(PREFIX).delta
LAYOUT   = $(PREFIX).layout
CONFLICT = $(PREFIX).conflict
CONTIG   = $(PREFIX).contig
FASTA   = $(PREFIX).fasta

INPUTS   = $(TGT) $(REF)
OUTPUTS  = $(CONTIG) $(FASTA)


## Building AMOS bank
10: $(BINDIR)/bank-transact -c -z -b $(BANK) -m $(TGT)

## Collecting clear range sequences
20: $(BINDIR)/dumpreads $(BANK) > $(SEQS)

## Running nucmer
30: $(NUCMER) --maxmatch --prefix=$(PREFIX) $(REF) $(SEQS)

## Running layout
40: $(BINDIR)/layout-align -U $(LAYOUT) -C $(CONFLICT) -b $(BANK) $(ALIGN)

## Running consensus
50: $(BINDIR)/make-consensus -B -b $(BANK)

## Outputting contigs
60: $(BINDIR)/bank2contig $(BANK) > $(CONTIG)

## Converting to FastA file
70: $(BINDIR)/ctg2fasta < $(CONTIG) > $(FASTA)
```

AMOS interchange format

Based on Celera message format

3-letter object tag (RED= read)

single-line attribute (action: ADD)

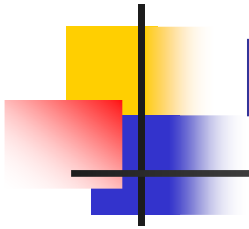
internal identifier (int32) (IID)

external identifier (EID)

multi-line attribute

```
{RED
act:A
iid:1
eid:nihaf5_10_a01.ab1
seq:
gggaattgctcgtttctggagccccgccagcgtctgcgctccgcctgtgcgcacagaaga
gaggtgtgagtaaagacagtgctgagtaccggcagaggagagagagaaacaacatcgccg
tcaggaagagtcgagataaagcgcggcgccgcatccagatgacccagcagagggcgctgc
agctgcaggatgagaatcaccggctgcaggtgcacatccagcgcctgctgcacgaggtgg
aggcgctcaggcattacctgtcccagcgtcacctgcaggacacatctgaggagcactgat
gagaatacacctggagaacacacacctgaagaaaaa
.
qlt:
7777777777777<?IMKD@988<?@C>>>HQQQUUUUXZhhhhhhhh[cXXXUUUUZZ_
ZUUUUUUXXXZZUSOPPSSZhhhhZZZXX]]ZZ\\\\\\h_hhhZZZ^^\ZZZUUU\\h\
h\\\\bbbbhh\\ZZZ[^Zhhhhhhhbbb\\_bbbb\\bZ[Z[^\hbbbbbhbhhhhhbh
hbbhhhZXXXXXZZZ[[ZZbbhhhhhbbbhhc[\ZZZb\ZZbbh\\\bbbb__\\\\\\h
\\hhhhhh\_____\\hhhhhhhhhhh[ZZZZZhhZXXXXXZ\\hhhhhhh[[ZXXXZZ
ZZZZZZZZZZhhZZZZZhhhhZUSSQOUULLAD998
.
frg:0
clr:14,333
}
```

C++ and Perl parsers are available



Bank Versions

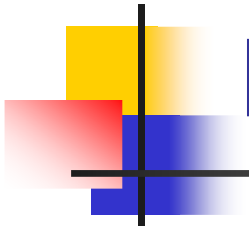
Banks are only compatible with version of AMOS that created them!

```
$ cat test.bank/RED.ifo
____RED BANK INFORMATION____
bank version = 2.8
bank type = 4474194
objects = 62229
indices = 62229
bytes/index = 55
partitions = 1
indices/partition = 1000000

locks =
```

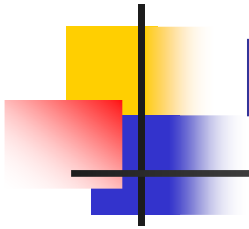
Updating to new bank version

1. bank-report-2.8 –b test.bank > test.afg
2. bank-transact –b test.bank –c –f –m test.afg (now version 2.9)
3. bank-transact –v (tells you version of bank it will write)



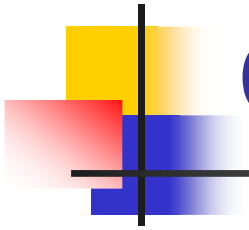
Existing Tools

- Data Conversions / Management
  - toAmos
  - bank-transact, bank-report
  - bank2contig, bank2fasta, bank2scaff
  - amos2frg, amos2ace, amos2mates
  - select-reads
- Pipelines
  - Minimus (hash-overlap, tigger, make-consenus)
  - AMOScmp (nucmer, casm-layout, make-consensus)
  - cavalidate



Existing Tools

- Validation / Repair Tools
  - Hawkeye
  - findMissingMates
  - stitchContigs
  - count-kmers
  - insert-sizes
  - analyze-snps
  - loadFeatures
  - resetFragLibrary

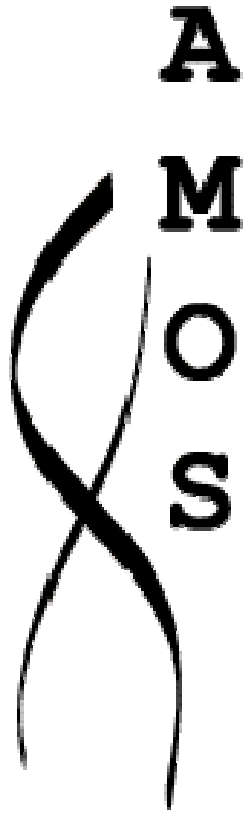


Creating your own tools

- Every class in core AMOS API is documented at Sourceforge.
- See AMOS/src/bank-tutorial.cc for an example of managing contigs and banks.
- Explore sourcecode to find something similar: alignment, contigs, bank, scaffolding, quality control…

More Information

- Contact AMOS
  - http://amos.sourceforge.net
  - amos-help [at] lists.sourceforge.net



- AMOS Team
  - Art Delcher
  - Adam Phillippy
  - Mihai Pop
  - Steven Salzberg
  - Michael Schatz
  - Dan Sommer