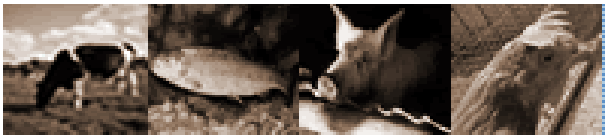
SURF

SeqUence Repository and Feature detection

USER GUIDE

Version 1.0.1

Eddie Iannuccelli



Revision table :

Version	*Date*	*Major modifications*	*Author(s)*
0.1	February 9th 2005	First release	C Dantec, E Iannuccelli
1.0	October 26 2005	DNA update	E Iannuccelli
1.0.1	16/02/2006	Instance creation description	E Iannuccelli

Acknowledgment :
I would like to thank the sigenae team members for their support and collaboration, special thanks to the local “Linux/Perl/etc. guru”, Patrice Dehais, how spent a lot of time to teach me most of the things I know about Linux, Perl and so on (my previous job was Windows development ...).

INDEX

1 Background

cDNA library sequencing has become a popular method to investigate genes through their transcripts, it also provide material for further expression and regulation studies (microarrays). Such libraries are usually sequenced using automated technique that produce huge amount of chromatogram data. These traces are to be processed before any exploitation and the resulting nucleotide sequences can be compared to the public nucleotide databases in order to extract new knowledge. As high flow sequencing can be done though dedicated robots, resulting data storage and processing is not always simple to solve.

2 Introduction

Pertinent nucleotide sequence production commonly involve several dedicated bioinformatic softwares (sequence base calling, vector detection, etc.) : SURF (*SeqUence Repository and Feature detection*) provide an integrated solution from chromatogram (or other popular format) data storage to cloned insert detection. SURF hosts sequences and manipulates them through two different concepts:

- *library*
  A library is the biological entity that hosts sequences, SURF uses libraries to describe features to detect (vector and adapters) from sequences.
- *Batch*
  A batch is the loading organizational entity which contains sequences that share a format and undergoes a common process (import into system , feature detection and statistic calculation). SURF supports 4 batch format types : Chromatogram, DbEST, EMBL and Fasta.

SURF is developed by SIGENAE team (AGENAE INRA project), SIGENAE extends no warranties of any kind, either expressed or implied.

3 User guide

3.1 Managing libraries

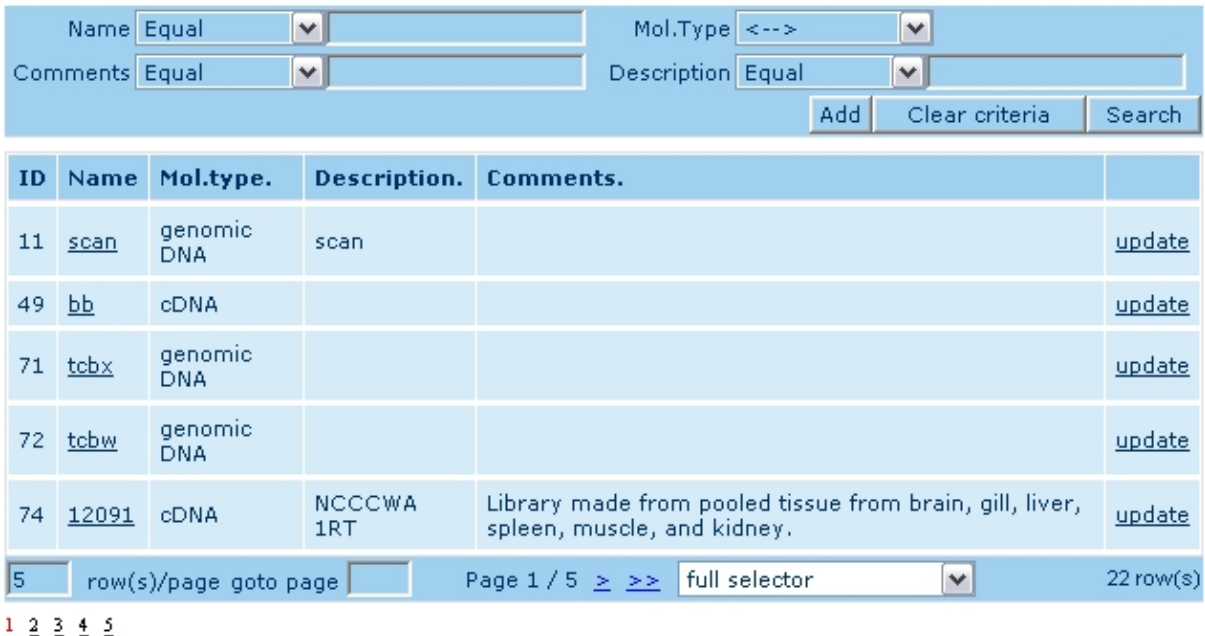
As the library drive the way the sequences are 'built', it is the best place to describe the expected features. Basic features are vector and adapters, but it is possible to add any kind of feature since a Fasta file is available. SURF uses *Crossmatch* software to detect features , this software needs two parameters (per feature) :

- *minmatch* : the minimum number of contiguous nucleotide to match before trying to extend match
- *minscore* : the minimum score to get in order to keep the hit

Feature can be freely named but some values are reserved for putative insert detection, we will call them '*construction features*' later in this document. The following names (case sensitive) are used to detect putative insert (other features are also used, see the Insert detection chapter for more details) :

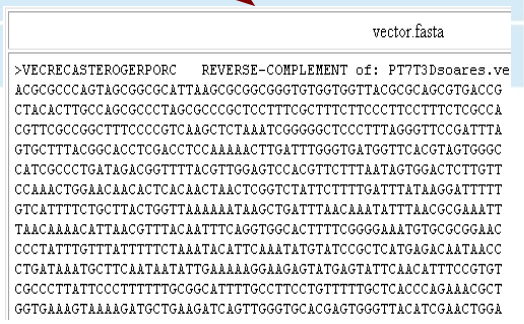
- 'vector' is used to designate the cloning vector feature.
- any feature starting by 'adapter' (ex : adapter5, adapter3)
- any feature ending by a '!' (ex : primer1!, primerA!)

Here is the library list screen, header provide criteria selection. In the list, the library name link provide a detailed library view page. The *update* link and *Add* button are reserved to system administrators.



One library detailed view

Name	scan
Code	scan
Molecule type	genomic DNA
description	scan
comments	
Features to detect	***Name*** ***Minmatch*** ***Minscore*** ***File*** ***SubFeatures*** vector 10 15 vector 2640-2660:E5;2667-2742:E3; adapter 3 9 adapter lucy 10 15 lucy
Custom properties	No data
Back	Library features combinations


vector.fasta

>VECRECASTEROGERPORC REVERSE-COMPLEMENT of: PT7T3Dsoares.ve
ACGCGCCCAGTAGCGGCGCATTAAGCGCGGCGGGTGTGGTGGTTACGCGCAGCGTGACCG
CTACACTTGCCAGCGCCCTAGCGCCCGCTCCTTTCGCTTTCTTCCCTTCCTTTCTCGCCA
CGTTCGCCGGCTTTCCCCGTCAAGCTCTAAATCGGGGGCTCCCTTTAGGGTTCCGATTTA
GTGCTTTACGGCACCTCGACCTCCAAAAACTTGATTTGGGTGATGGTTCACGTAGTGGGC
CATCGCCCTGATAGACGGTTTTACGTTGGAGTCCACGTTCTTTAATAGTGGACTCTTGTT
CCAAACTGGAACAACACTCACAACTAACTCGGTCTATTCTTTTGATTTATAAGGATTTTT
GTCATTTTCTGCTTACTGGTTAAAAAATAAGCTGATTTAACAAATATTTAACGCGAAATT
TAACAAAACATTAACGTTTACAATTTCAGGTGGCACTTTTCGGGGAAATGTGCGCGGAAC
CCCTATTTGTTTATTTTTCTAAATACATTCAAATATGTATCCGCTCATGAGACAATAACC
CTGATAAATGCTTCAATAATATTGAAAAAGGAAGAGTATGAGTATTCAACATTTCCGTGT
CGCCCTTATTCCCTTTTTTGCGGCATTTTGCCTTCCTGTTTTTGCTCACCCAGAAACGCT
GGTGAAAGTAAAAGATGCTGAAGATCAGTTGGGTGCACGAGTGGGTTACATCGAACTGGA

Available informations are :

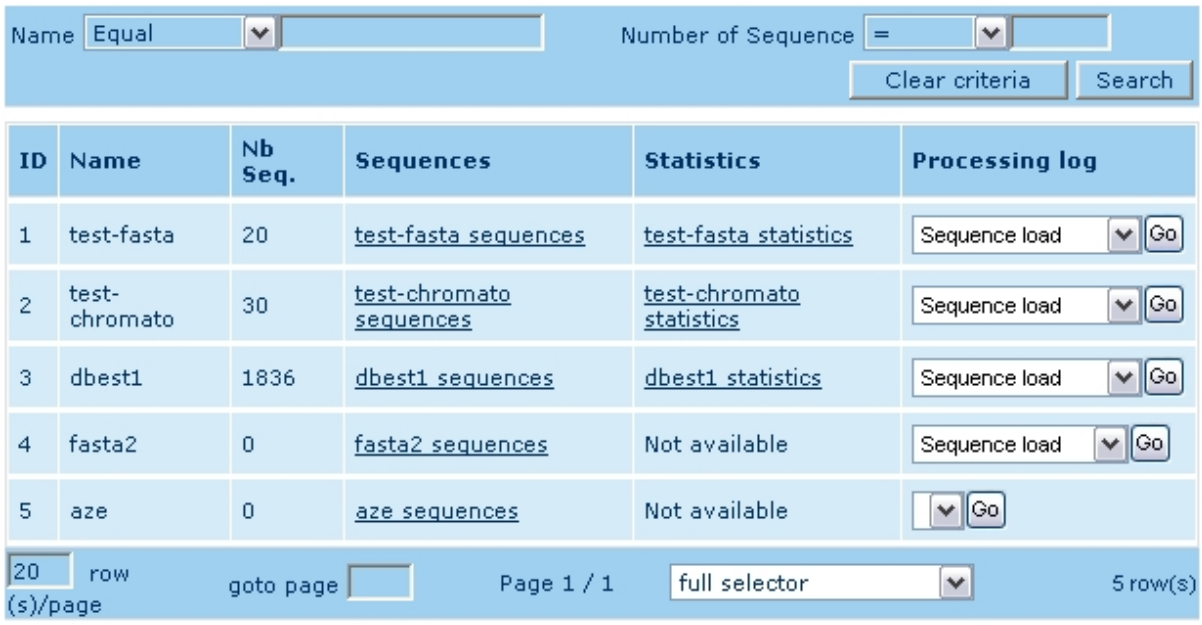
- *Name* : The library name
- *Code* : string used to connect a sequence to its library during data loading
- *Description* : any descriptive data
- *Comments* : any comment
- *Features* to detect list :
  - *Name*
  - *Minmatch* :crossmatch parameter
  - *Minscore* : crossmatch parameter
  - *File* : fasta file containing feature sequence
  - *Subfeatures* : sub features coordinates
    formatted as START-END:NAME ex : 10-15:EcoR1
- *Custom properties* : *tissue_type* and *organ* data only for imported DbEST libraries.
- *Library feature combinations* : link to library statistical page (also available from batch statistics, see *statistical calculation* chapter for details).

3.2 Managing batches

A batch contains sequences that shipped together during the primary data import from chromatogram, Dbest, Embl or fasta raw data. Batch sequences are also processed together for further processing (features detection and statistic calculation). The batch list screen provide following data :

- ID (SURF internal identifier)
- name
- number of sequences
- a link to see batch sequences : bring to sequence list screen
- a link to see batch statistics
- a pop up menu plus a 'Go' button to see processing logs (data loading, feature detection and statistic processing

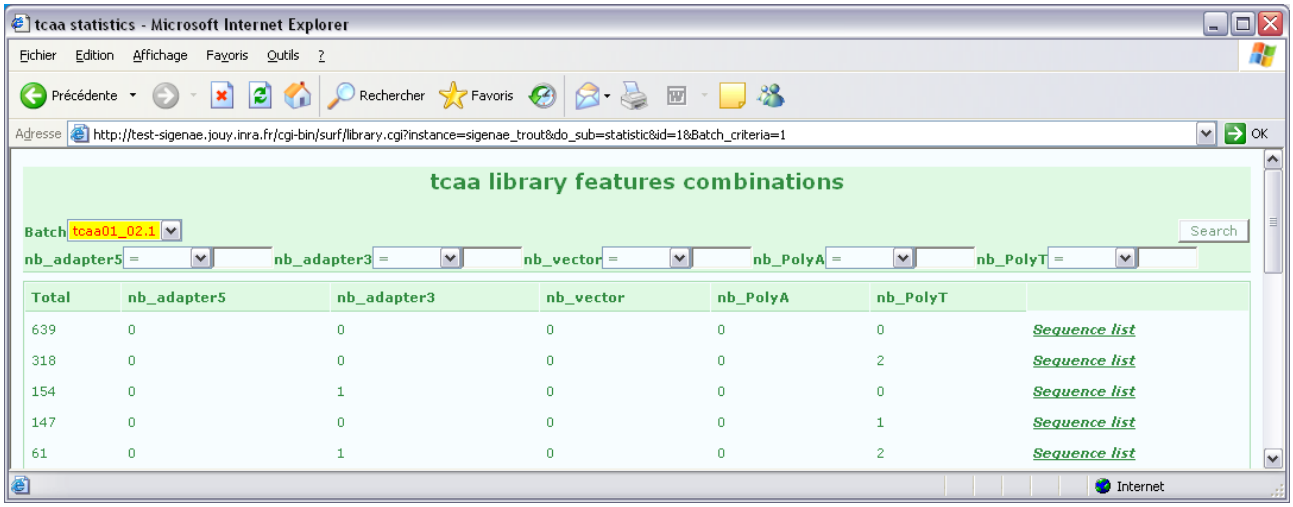
Here is a screen capture of the batch management tool.



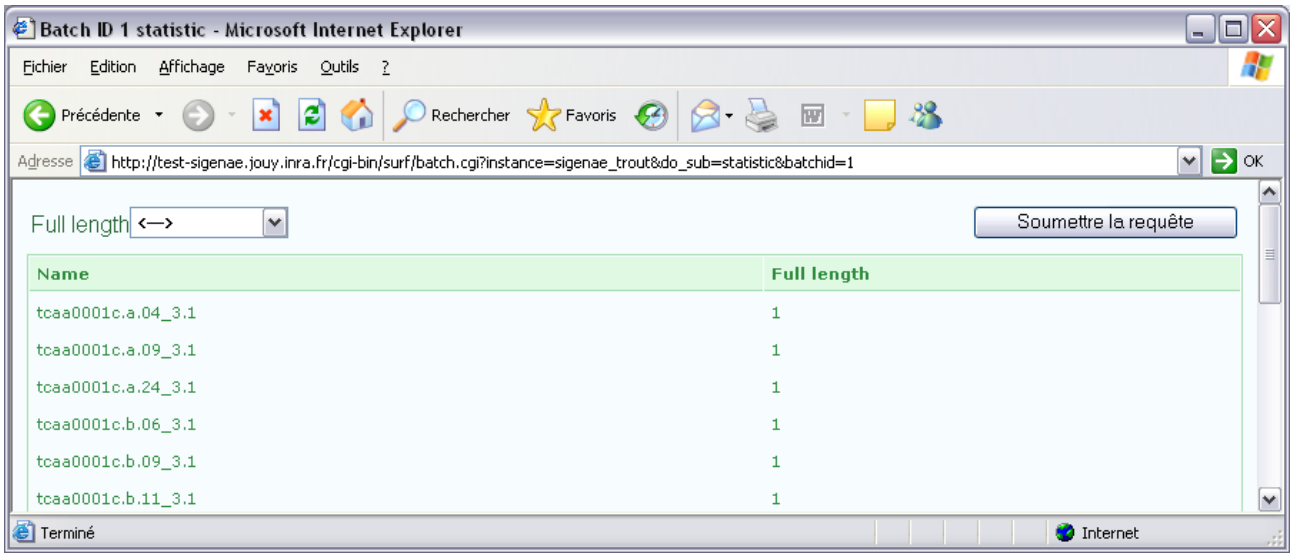
The batch statistics provide following informations :

- *Sequence number*
- *Null sequence count*
- *Count per status (see status chapter)*
- *Count per contaminant (see status chapter)*
- *Batch libraries features combinations*
- *Full length sequence list*
- *Sequence length histogram*
- *Sequence Nb good base histogram (see status chapter)*
- *Basecalling quality histogram*

Batch libraries features combinations

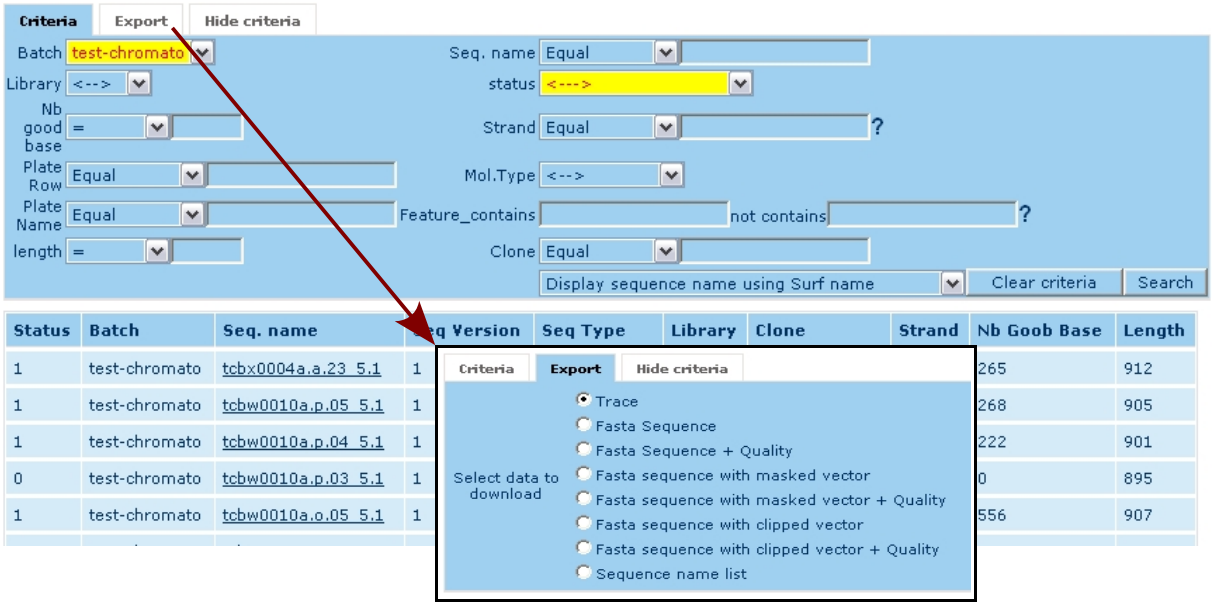


Full length sequence screen



3.3Managing sequences

Sequence is the SURF central entity, it is accessible through the following list screen which provide various selection criteria and export capabilities.



Sequence can be displayed using various names :

- *SURF name* (chromatogram file name, DbEST ID field, EMBL ID field or FASTA ID)
- *Accession number*
- *Genbank Gi*
- *Est name*

NB :

Selecting an alias in this screen will restrict results to sequence having such alias. By example, if you select 'Display sequence name using accession number', chromatogram sequences without accession number alias will simply be hidden.

One sequence details

3.4 Loading sequence

Sequences are not loaded into SURF one by one but all together through a batch. A batch is a set of sequences that share a common format and, most of the time, were created through the same production process (same automatic sequencer machine or/and run, same PCR reaction set, etc.). SURF does not use batch format only as a formatting specification, format also change the way SURF manages sequence versioning. Each batch format is loaded through dedicated command line programs, here are the various batch format SURF can load.

NB : command line programs are located in the SURF *bin* directory.

Programs will ask for common parameters :

- ***instance*** : SURF instance name
- ***batch name*** : batch name, note that SURF checks batch existence using batch name so if you want to override existing batch with another one, use the same name.
- ***library*** : existing library for batch sequence to link to (feature detection). Default value is 'autodetect' which tries to parse library name from batch source (file name or content).
- ***regular expression model name*** : corresponding *regexpmodel_*conf* section to use for parsing (see configuration files). Default is *default* section.
- ***Workflow step number to start*** : step rank to run , any previous step is ignored (useful to “debug” some huge batches). Default value is 'autodetect' which means that any previous failed step will be used as value.
- ***Address to send email notification when finished*** : e mail address to which success or failure notifications are sent.

3.4.1 Chromatogram batch

3.4.1.1 Launching the program

Use the *run_chromato.sh* to start loading batch, here are the specific parameters :

- ***zip file path*** : path to the zip file containing the chromatograms. This file is a zip archive, not a gzip archive. It is very important that archive DO NOT CONTAINS DIRECTORIES, all chromatogram files MUST be at the zip archive root level (this constraint force file name unicity). Any zip archive sub-directory will be simply ignored, SURF will load SCF, ABI, AB1, ABD extension zip files only (see surf.conf).
- ***default sequence type*** : default molecule type for standalone sequence (not linked to a library). Usually, when a sequence is linked to a library, the library molecule type is used as template for sequence molecule type. Default is cDNA.
- ***plate name*** : string to put as plate name, default is 'autodetect' which means that SURF will try to parse plate name from batch source (file name or content).
- ***strand*** : string to put as strand value, default is 'autodetect' which means that SURF will try to parse strand value from batch source (file name or content).
- ***clone name auto detection*** : tells SURF to parse batch source for clone name, default is 'yes'
- ***row and column values auto detection*** : tells SURF to parse batch source for row and col values, default is 'yes'.

3.4.1.2 Sequence name version rules

SURF tries to keep memory of sequence modifications so a version value is inserted in the sequence file name, before the file extension (ex : the file name of fourth version of *scac0001.a.04_5.scf* sequence wil be *scac0001.a.04_5.**4**.scf).* Any chromatogram sequence input is assumed to be a new version sequence (n+1) except when loading already existing batch (check based on the batch name). Typically, if you load a batch using an existing batch name, if sequence name are conflicting, SURF will overwrite existing batch sequences with new chromatogram data but will keep previous version values.

3.4.1.3 Libraries management:

SURF tries to link chromatogram sequences to their libraries at loading time. For each sequence, '*library*' regular expression is run onto chromatogram file name, resulting string is used for a library '*code*' field lookup. If a library match, the link is made, it will bring valuable informations at feature detection time.

3.4.2DbEST batch

3.4.2.1Launching the program

Use the *run_dbest.sh* to start loading a batch. The only specific parameter is ***gz file path ,***path to a tar.gz file containing DbEST files.

3.4.2.2Sequence name and versions

SURF uses *dbEST Id* tag as sequence name. Other DbEST identifiers (*EST name,GenBank Acc and GenBank gi*) are stored as sequence aliases. If a DbEST candidate sequence name already exists in SURF and if the two FASTA sequences are strictly equals, the candidate sequence is excluded from load. If a DbEST candidate sequence name corresponds to an existing chromatogram sequence alias, the candidate DbEST sequence is loaded ONLY when its FASTA nucleotidic sequence does not equal any existing chromatogram sequence '*publication*' feature FASTA. *Publication* feature FASTA sequence can be customized by a 'FASTA' tag stored in *publication* feature 'comment' field (Ex : *fasta (20..150,AATTT,200..210)*. This can be useful to avoid sequence duplication (private chromatogram version + public version return through DbEST batch). When 'publication' feature provides a FASTA comment, FASTA comment overrides 'publication' feature start and end positions. If a new sequence version is loaded, '.version' string is appended at the end of the sequence file name (Ex : XXX.2).

3.4.2.3Libraries management

Libraries are parsed from DbEST files and new libraries are added. If a library already exists in the system, the library *custom properties* are updated using DbEST library records *dev_stage* and *tissue_type* tags.

3.4.2.4Basecalling quality management emulation

Special '.type' file are also generated in order to emulate quality at sequence extraction time.
For each sequence, the program creates a file named *[sequence_name].fasta.type* and writes the *molecule_type*value on the first line. That file will be used later to generate quality on the fly during sequence quality extraction (see *get_sequence.pl* program).

3.4.2.5DbEST -> SURF fields mapping

DbEST file fields	*SURF*
DbEST Id	SURF sequence name
est_name	'est_name' type alias
genbank_acc	'genbank_acc' type alias
genbank_gi	'genbank_gi' type alias
Clone Id	Sequence clone
DNA type	Molecule type
SEQUENCE (1)	Sequence nucleotidic sequence
Plate Row Column	Sequence plate, row, and column
Entry Created	creation date
DbEST lib id	Library name
Lib Name	Library description
Description (1)	Library comments
Tissue type | Organ	**Library** “tissue_type” custom properties
Develop stage	**Library** “dev_stage” custom properties

(1) hard coded, not stored in *regexp_dbest.conf*

3.4.3EMBL batch

3.4.3.1Launching the program

Use the *run_embl.sh* script to start loading a batch. The only specific parameter is ***gz file path ,which is the*** path to a tar.gz file containing EMBL files.

3.4.3.2Sequence name and versions

SURF uses EMBL *ID* tag as sequence name. The First EMBL accession number is stored as an '*accession number*' alias type and others accession numbers are stored as '*secondary accession number*' alias type.
If an EMBL candidate sequence name already exists in SURF and if the two FASTA sequences are strictly equals, the candidate sequence is excluded from load. If an EMBL candidate sequence name corresponds to an existing chromatogram sequence alias, the candidate sequence is loaded ONLY when its FASTA nucleotidic sequence does not equal any existing chromatogram sequence '*publication*' feature FASTA. *Publication* feature FASTA sequence can be customized by a 'FASTA' tag stored in *publication* feature 'comment' field (Ex : *fasta (20..150,AATTT,200..210)*. This can be useful to avoid sequence duplication (private chromatogram version + public version return through EMBL batch). When *publication* feature provides a FASTA comment, FASTA comment overrides *publication* feature start and end positions. If a new sequence version is loaded, '.version' string is appended at the end of sequence file name (Ex : XXX.2).

3.4.3.3Libraries management:

As EMBL file format is generalist and do not give reliable structured library informations, SURF only parse *dev_stage* and *tissue_type* tags and place them into sequence custom properties (instead of library custom properties for DbEST).

3.4.3.4 Basecalling quality management emulation

Special '.type' file are also generated in order to emulate quality at sequence extraction time.
For each sequence, the program creates a file named *[sequence_name].fasta.type* and writes the *molecule_type*
value in first line. That file will be used later to generate quality on the fly during sequence quality extraction (see *get_sequence.pl* program).

3.4.3.5EMBL -> SURF fields mapping

EMBL file fields	*SURF*
ID	SURF sequence name
AC	First is “accession number” type alias, others (if applicable) are “accession number secondary” type aliases.
DE	Stored in FASTA file header
DT Rel. Created.	Creation date
SQ (multi-lines)	Sequence
FT /clone	Clone
FT /tissue_type	**Sequence** “tissue_type” custom properties
FT /dev_stage	**Sequence** “dev_stage” custom properties
FT /mol_type	Molecule type

3.4.3.6Molecule type EMBL specific rule

We assume that mRNA sequences lower than 1000 base pairs without 'complete' keyword in description field are not credible, so we change '*mol type*' field from “*mRNA*” to *cDNA* for such sequences.

3.4.4FASTA batch

3.4.4.1Launching the program

Use the *run_fasta.sh* script to start loading a batch,. Here are the specific parameters :

- ***gz file path*** : path to a gz archive containing the FASTA files.
- ***default sequence type*** : default molecule type for standalone sequence (not linked to a library). When a sequence is linked to a library, the library molecule type is used as a template for sequence molecule type. Default is cDNA.
- ***plate name*** : string to put as plate name. Default is 'autodetect' which means that SURF will try to parse plate name from batch source (file name or content).
- ***strand*** : string to put as strand value. Default is 'autodetect' which means that SURF will try to parse strand value from batch source (file name or content).
- ***clone name auto detection*** : tells SURF to parse batch source for clone name. Default is 'yes'
- ***row and column values auto detection*** : tells SURF to parse batch source for row and column values. Default is 'yes'.

3.4.4.2Sequence name and versions

If a FASTA candidate sequence name already exists in SURF and if the two FASTA sequences are strictly equals, the candidate sequence is excluded from load. The only exception is when a batch is reloaded (i.e. when using the same batch name). In that case, the new batch sequences always overwrite the previous ones. If a FASTA candidate sequence name corresponds to an existing chromatogram sequence alias, the candidate sequence is loaded ONLY when its FASTA nucleotidic sequence does not equal any existing chromatogram sequence '*publication*' feature FASTA. *Publication* feature FASTA sequence can be customized by a 'FASTA' tag stored in p*ublication* feature 'comment' field (Ex : *fasta (20..150,AATTT,200..210)*. This can be useful to avoid sequence duplication (private chromatogram version + public version return through FASTA batch). When *publication* feature provides a FASTA comment, FASTA comment overrides *publication* feature start and end positions.

3.4.4.3Libraries management

SURF tries to link FASTA sequences to their libraries at loading time. For each sequence, '*library*' regular expression is run onto FASTA file name. The resulting string is used for a library '*code*' field lookup. If a library matches, the link is made. It will bring valuable informations at feature detection time.

3.4.4.4 Basecalling quality management emulation

Special '.type' file are also generated in order to emulate quality at sequence extraction time.
For each sequence, the program creates a file named *[sequence_name].fasta.type* and writes the *molecule_type*
value in first line. That file will be used later to generate quality on the fly during sequence quality extraction (see *get_sequence.pl* program).

3.5 Features detection

Once sequences are load into SURF, only basic informations are available (name, sequences, quality, etc), no added value data (as vector, adapter, repeats) exist in the system. To get these data, you must start the feature detection process. This process will use various programs to characterize each sequence and will try to detect a putative insert. At the end, a status, reflecting sequence validity, is calculated.

NB : command line programs are located in the SURF */bin* directory.

3.5.1 Launching the program

Use the *run_feature.sh* script to start feature detection. Here are the parameters:

- ***instance*** : SURF instance name.
- ***batch ID***: batch ID.
- ***Workflow step number to start*** : step rank to run. Any previous steps are ignored (useful to "debug" some huge batches). Default value is 'autodetect' which means that any previous failed step will be used as value.
- ***Workflow step number to stop at*** : step rank to run as last step.
- ***Address to send email notification when finished*** : e mail address to which success or failure notifications are sent.

3.5.2 Feature detection steps

3.5.2.1 Library specific vector detection

The aim of this step is library specific construction detection using Crossmatch program. When a sequence is linked to a library which provides a 'vector' feature, the 'vector' feature FASTA file is used to detect cloning vector hits. If sequence is not linked to a library or if the library does not contain a 'vector' feature, UNIVEC database is used to detect cloning vector hits. Linked library can also provide other features (adapters, etc.), these features are also tested against each batch sequence in this step. Each feature can be described more finely using '*sub-features*' field. A sub-feature is a string containing one or more [*start-stop:label;*] descriptors (ex: *10-16:EcoR1;20-25:tag*). These sub-features are considered as features when they are fully included into parent feature hit.

3.5.2.2 Low complexity sequence detection

This step uses *RepeatMasker* program to detect low complexity sequences : LINE, SINE, (CAAAA)n#Simple_repeat, ()AGCTGTGGGGC()n#Satellite, etc. Note that SURF uses the open 3.0 version that can use *WU-Blast* as sequence comparison engine instead of *Crossmatch*.

3.5.2.3 Bad quality detection

This step uses *phred* quality files to produce bad quality features. Any *phred* value strictly lower than *minqual surf.conf* file key is considered as a bad base. Bad bases are clustered into bad quality features using a 10 base pairs maximum gap. Non chromatogram sequences bad quality features are emulated using *makequality_param_[moltype] surf.conf* keys. makequality_param is used when *makequality_param_moleculeType* key does not exist (ex makequality_param_mrna key will override makequality_param key for mRNA quality emulation).
These keys are formatted as : *makequality_param,= param1 param2* where:

- *param 1:* something like 'p1:v1,p2:v2,...,pn:vn'\nwhere pi<p(i+1), in fact a comma delimited list of couples pi:vi where pi is a position on the sequence and vi is the *phred* value that will be given to all bases located before the specified position. The list can be empty.
- *param 2*: the default *phred* value for other bases.

3.5.2.4 Single nucleotide repetition detection

Since *Crossmatch* does not always report this kind of repeat (by example when no other hit is detected) and since *RepeatMasker* does not always report small single nucleotide repetition, we decide to write a SURF specific program (*Polysmurtsh*) to detect single nucleotide repetition (15 bp minimum PolyN anchored by a 8 bp minimum with a maximum gap of 4 bp). For this reason, you can find some PolyN twice in sequences features, one found by *RepeatMasker*, one found by *Polysmurtsh*.

3.5.2.5 Initial database loading

?First feature database load, these features will serve as data source for the next insert detection step.?

3.5.2.6 Insert position detection

SURF detects insert position by using its own algorithm or using TIGR *lucy* program.

Using TIGR lucy

In order to use *lucy* as insert detection tool, you must update *surf.conf* file as following:

```
use_lucy=true
lucy_as_insert=true (if false, SURF will detect a useless lucy feature)
```

You also need to create a dedicated *lucy* feature which contains splicing sites (30 bp flanking regions minimum) for sens AND reverse so 4 FASTA entries (5S,5R,3S,3R) in one fasta file for EACH library.
NB: SURF will stop processing if a library does not provide the lucy feature.
Lucy also need a vector so if you do not add a library 'vector' feature UNIVEC will be used.

Using SURF algorithm

SURF tries to extend the vector from the sequence extremities using 'construction features' (described in the related library, usually vector and adapters plus polyA and polyT, see library).
Theses features are joined if they are :

- overlapping or
- separated by 20 bp maximum or
- separated by a minimum 60% bad quality zone.

SURF processes these features in a two process way : 5' and 3'

- 5' features are processed from the beginning to 20% of the sequence, they are joined as previously described.
- 3' features are processed from 60% of the sequence to the end or from any feature to the end since the feature has not been used by 5' side process and if bad quality between feature and sequence end is greater than 80%.

After putative insert has been detected, SURF calculates the number of *good bases.* The number of good bases is the count of sequence nucleotides that :

- belong to insert
- have a phred quality >= *minqual* surf.conf key (usualy 20)
- are not included in a repeat (only for cDNA and mRNA molecule types)

This value reflects the sequence pertinence regarding it quality and composition.

3.5.2.7 UNIVEC contamination detection

SURF uses UNIVEC database to detect any cloning vector contamination in putative insert. This method is an indirect way for chimera detection since a vector hit (using UNIVEC) located into putative insert often denote chimera presence. *Crossmatch* program is used with the following parameters : *minmatch* =10 and *minscore*=25. UNIVEC database is reached via a symbolic link, called *univec.fasta*, located in the instance *banks* directory, the targeted file is located into SURF dataroot *banks* directory. UNIVEC can be detected but not considered as a contaminant simply by removing its name from the *contaminants* surf.conf key.

3.5.2.8 E. Coli contamination detection

SURF uses E.Coli genome sequence database to detect any contamination in putative insert. *Crossmatch* program is used with the following parameters : *minmatch* =100 and *minscore*=150. E. Coli database is reached via a symbolic link, called coli.*fasta*, located in the instance *banks* directory, the targeted file is located into SURF dataroot *banks* directory. E.Coli can be detected but not considered as a contaminant simply by removing its name from the *contaminants* surf.conf key.

3.5.2.9 Yeast contamination detection

SURF uses yeast genome sequence database to detect any contamination in putative insert. *Crossmatch* program is used with the following parameters : *minmatch* =100 and *minscore*=150. Yeast database is reached via a symbolic link, called yeast.*fasta*, located in the instance *banks* directory, the targeted file is located into SURF dataroot *banks* directory. Yeast can be detected but not considered as a contaminant simply by removing its name from the *contaminants* surf.conf key.

3.5.2.10 Ribosome contamination detection

SURF uses dedicated species ribosome sequence database (should be set at instance creation) to detect any contamination in putative insert. *Crossmatch* program is used with the following parameters : *minmatch* =100 and *minscore*=150. Ribosome database is a file called *ribo.fasta* located in the instance *banks* directory. Ribosome can be detected but not considered as a contaminant simply by removing its name from the *contaminants* surf.conf key.

3.5.2.11 Mitochondry contamination detection

SURF uses dedicated species mitochondry sequence database (should be set at instance creation) to detect any contamination in putative insert. *Crossmatch* program is used with the following parameters : *minmatch* =100 and *minscore*=150. Mitochondry database is a file called *mito.fasta* located in the instance *banks* directory. Mitochondry can be detected but not considered as a contaminant simply by removing its name from the *contaminants* surf.conf key.

3.5.2.12 Poly-Adenylation signal detection

This step detects any poly-adenylation signal (AATAAA / ATTAAA) in the 30 base pairs before any polyA/polyT feature.

3.5.2.13 Final database loading

This step loads from *UNIVEC* step to *Poly-adenylation signal detection* step detected features into database.

3.5.2.14 Status update

In order to quickly sortgood and bad sequences, SURF provides a sequence status.
This status is strictly greater than 0 (1 or greater) when sequence is acceptable for further studies (clustering, assemblies, etc .) Status equal 0 when at least one of the following condition is true :

- *sequence contains contaminant* such as E. Coli, Yeast, Mitochondry, Ribosome or UNIVEC. UNIVEC is assumed as a contamination when it is found into putative insert (see UNIVEC contamination detection section), which means that sequence has a good chance to be a chimera.
- *Sequence having a good base number lower than 100 base pairs* are assumed to be too small to provide pertinent information.

Status is lower than 0 when sequence was correct but now have a newest active version

3.6 Batch statistics

After batch is loaded, SURF can calculate various statistics reflecting batch global quality.

3.6.1 Launching the program

Use the *run_stat.sh* script to start feature detection. Here are the parameters:

- ***instance*** : SURF instance name
- ***batch ID***: batch ID
- ***Workflow step number to start*** : step rank to run. Any previous steps are ignored (useful to "debug" some huge batches). Default value is 'autodetect' which means that any previous failed step will be used as value.
- ***Workflow step number to stop at*** : step rank to run as last step.
- ***Address to send email notification when finished*** : e mail address to which success or failure notifications are sent.

3.6.2 Statistics calculation steps

3.6.2.1 File system creation

The aim of this step is to create a *statistic.html* and a *statistic_files* directory under the batch *attach* folder. Some of the statistical data will be stored there.

3.6.2.2 General statistics calculation

This step calculates basic indicators such as sequence count, null sequence count, count per status and count per contaminant.

3.6.2.3 Full length sequence detection

This step counts and detects full length sequences.

3.6.2.4 Library feature combinations

This step compiles all feature combinations (including number of features) per batch library and count them, related sequence list is provided for each feature pattern.

3.6.2.5 Graphics production

This step uses *R* software to produce sequence length histogram, sequence number of good bases histogram, insert size histogram and basecalling quality / base pair position plot.

4 Licence

Version 1.2, November 2002

```
Copyright (C) 2000,2001,2002  Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies

to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one

passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Illustration Index