Homological Processing of Biomedical digital images: automation and certification*

Jónathan Heras, Gadea Mata, María Poza and Julio Rubio

Department of Mathematics and Computer Science
University of La Rioja
Spain

June 27, 2011

* Partially supported by Ministerio de Educación y Ciencia, project MTM2009-13842-C02-01, and by European Commission FP7, STREP project ForMath, n. 243847

Table of Contents

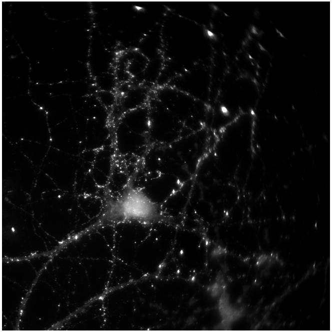
Table of Contents

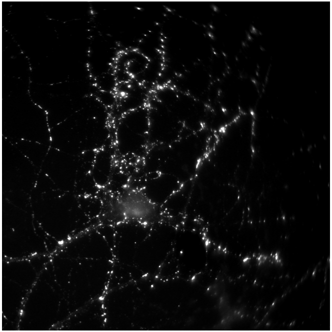
Motivation: Synapses counting

- *Synapses* are the points of connection between neurons
- *Relevance*: Computational capabilities of the brain
- The different number of synapses may be an important asset in the treatment of neurological diseases

Manual processing to count synapses

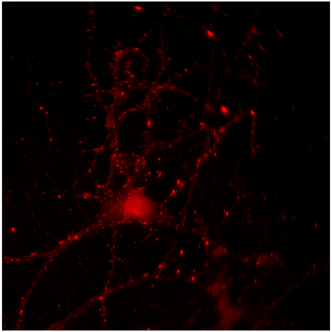
- Apply two different antibody markers, bassoon and synapsin

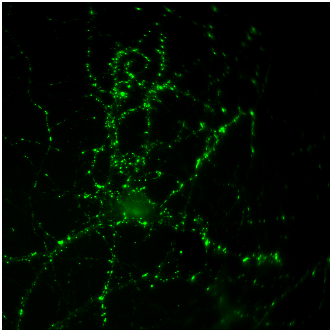




Manual processing to count synapses

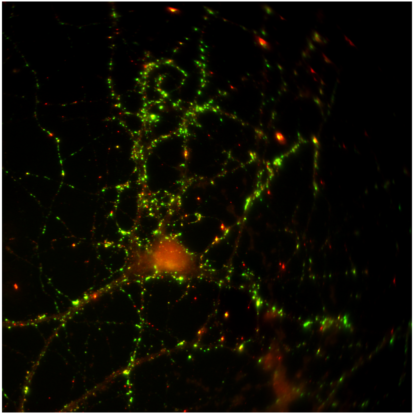
- Process the images in order to count the synapses *(ImageJ)*





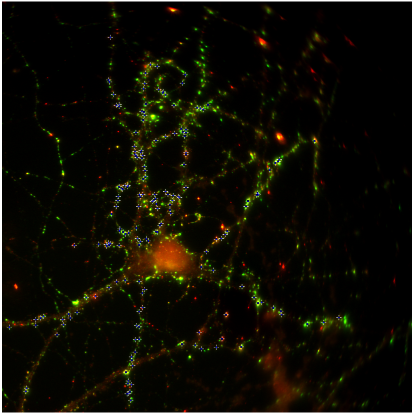
Manual processing to count synapses

- Overlap both images



Manual processing to count synapses

- The synapses are manually counted one by one



Problems and goal to count synapses

Problems

- Huge time investment
- This process is applied over a battery of neurons

Goal

Provide a reliable and automatic method for counting synapses in a neuron

Table of Contents

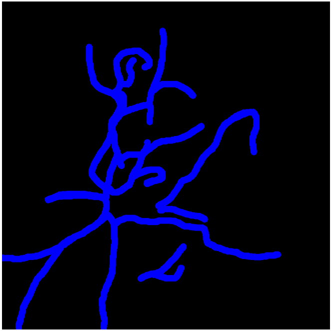
Table of Contents

Count synapses automatically

- New ImageJ plugin called *SynapCountJ*

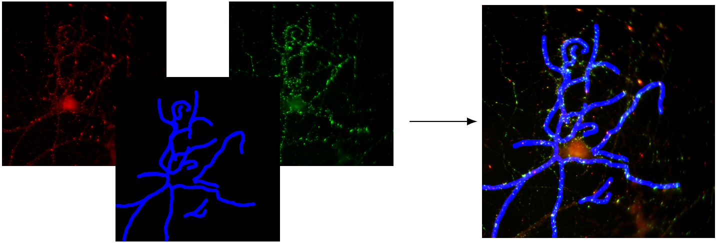
Count synapses automatically

- New ImageJ plugin called *SynapCountJ*
- Steps
  1. Determine the neuron morphology from one of those pictures *(NeuronJ plugin)*



Count synapses automatically

- New ImageJ plugin called *SynapCountJ*
- Steps
  1. Determine the neuron morphology from one of those pictures *(NeuronJ plugin)*
  2. Overlap the images with the two markers with the one with the structure *(SynapCountJ)*



Count synapses automatically

- New ImageJ plugin called *SynapCountJ*
- Steps
  1. Determine the neuron morphology from one of those pictures *(NeuronJ plugin)*
  2. Overlap the images with the two markers with the one with the structure *(SynapCountJ)*
  3. Invert the colors to show the synapses as black points

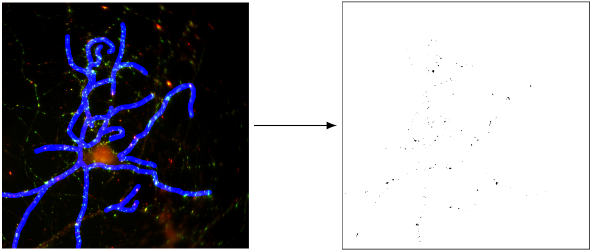
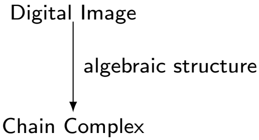


Table of Contents

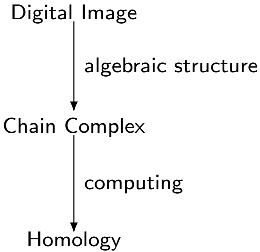
The method

Digital Image

The method



The method


Digital Image
algebraic structure
Chain Complex
computing
Homology

The method

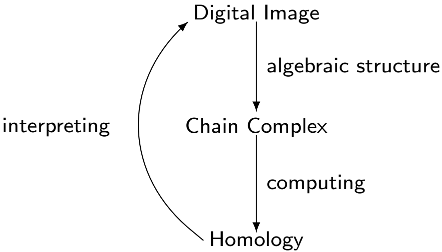

Digital Image
algebraic structure
Chain Complex
computing
Homology
interpreting

Image to Chain Complex

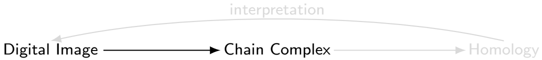
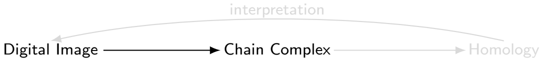


Image to Chain Complex



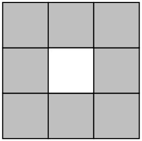
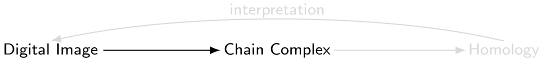


Image to Chain Complex



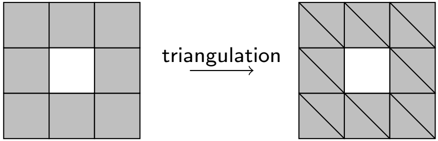
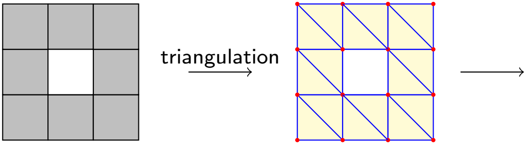


Image to Chain Complex

interpretation

Digital Image → Chain Complex → Homology

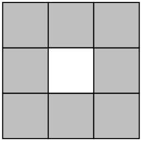


$$C_0 = \mathbb{Z}[vertices]$$
$$\uparrow d_1$$
$$C_1 = \mathbb{Z}[edges]$$
$$\uparrow d_2$$
$$C_2 = \mathbb{Z}[triangles]$$

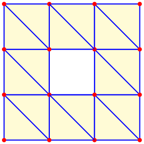
Image to Chain Complex

interpretation

Digital Image ⟶ Chain Complex ⟶ Homology



triangulation ⟶

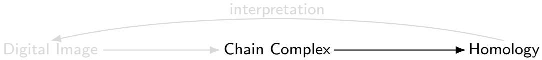


⟶

$$C_0 = \mathbb{Z}[vertices] \xleftarrow{d_1} C_1 = \mathbb{Z}[edges] \xleftarrow{d_2} C_2 = \mathbb{Z}[triangles]$$

$$0 \leftarrow \mathbb{Z}^{16} \xleftarrow{d_1} \mathbb{Z}^{32} \xleftarrow{d_2} \mathbb{Z}^{16} \leftarrow 0$$

Compute Homology groups



- Problem of diagonalizing matrices
- Compute the Smith Normal Form

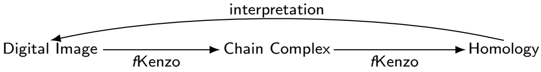
Properties from homology groups

- H_0 measures the number of connected components
- H_1 measures the number of holes

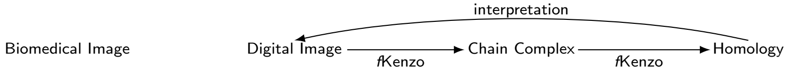
$$H_0(\textit{image with the points}) = \underbrace{\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}}_{\mathbb{Z}^{209}}$$

Table of Contents

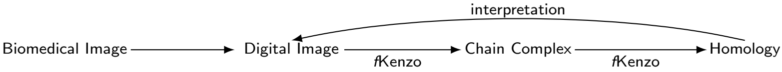
General method



General method



General method



General method

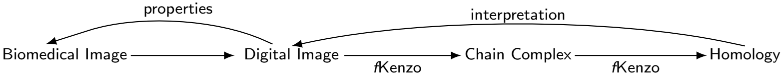


Table of Contents

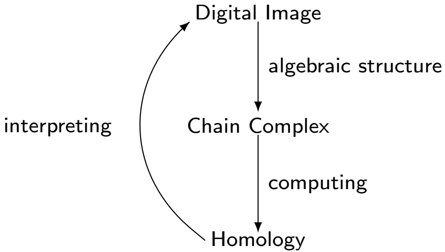
Problems

- Size of the images
- Correctness of the results

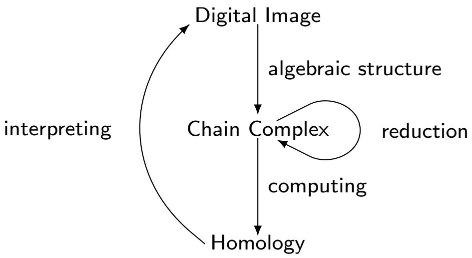
Problems

- Size of the images $\longrightarrow$ Discrete Morse theory
- Correctness of the results $\longrightarrow$ Certification of the programs

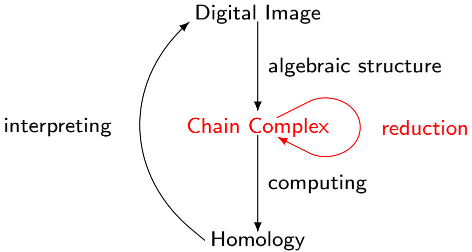
The method


Digital Image
algebraic structure
interpreting
Chain Complex
computing
Homology

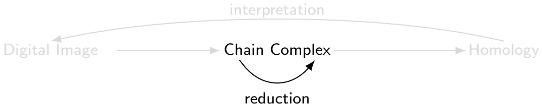
The method


Digital Image
algebraic structure
interpreting
Chain Complex
reduction
computing
Homology

The method

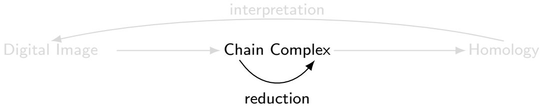

Digital Image
algebraic structure
interpreting
Chain Complex
reduction
computing
Homology

Reduction of chain complex

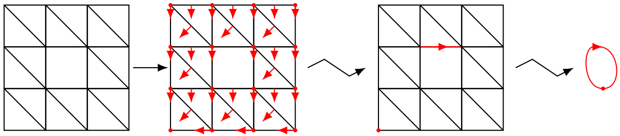


- Reduce information keeping the homological properties
- Discrete Morse Theory
    - Vector fields are a tool to cancel "useless" information

Reduction of chain complex

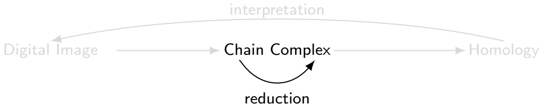


- Reduce information keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel "useless" information

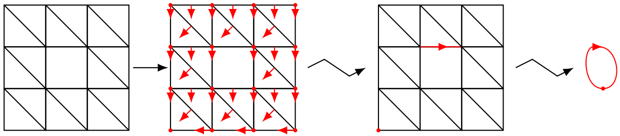


$$0 \leftarrow \mathbb{Z}^{16} \leftarrow \mathbb{Z}^{32} \leftarrow \mathbb{Z}^{16} \leftarrow 0$$

Reduction of chain complex



- Reduce information keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel "useless" information



$$0 \leftarrow \mathbb{Z} \leftarrow \mathbb{Z} \leftarrow 0 \leftarrow 0$$

Discrete Morse Theory

Definition

Let $C_* = (C_p, d_p)_{p \in \mathbb{Z}}$ be a free chain complex with distinguised $\mathbb{Z}-$basis $\beta_p \subset C_p$. A $(p-1)$-cell σ is a *face* of a p-cell τ if the coefficient of σ in $d\tau$ is non-null. It is a *regular face* if this coefficient is $+1$ or -1

Definition

A *discrete vector field* on C_* is a collection of pairs $V = \{(\sigma_i, \tau_i)\}_{i \in \beta}$ satisfying the conditions:

1. Every σ_i is some element of β_p, in which case the other corresponding component $\tau_i \in \beta_{p+1}$. The degree p depends on i and in general is not constant
2. Every component σ_i is a *regular face* of the corresponding component τ_i
3. A generator of C_* appears at most one time in V

Discrete Morse Theory

Definition

A *V-path of degree p* is a sequence $\pi = ((\sigma_{i_k}, \tau_{i_k}))_{0 \leq k < m}$ satisfying:

1. Every pair $((\sigma_{i_k}, \tau_{i_k}))$ is a component of V and the cell τ_{i_k} is a p-cell
2. For every $0 < k < m$, the component σ_{i_k} is a face of $\tau_{i_{k-1}}$, non necessarily regular, but different from $\sigma_{i_{k-1}}$

Definition

A *discrete vector field V* is admissible if for every $p \in \mathbb{Z}$, a function $\lambda_p : \beta_p \to \mathbb{Z}$ is provided satisfying the property: every V-path starting from $\sigma \in \beta_p$ has a length bounded by $\lambda_p(\sigma)$

Discrete Morse Theory

Definition

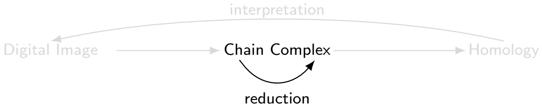
A cell χ which does not appear in a discrete vector field $V = \{(\sigma_i, \tau_i)\}_{i \in \beta}$ is called a *critical cell*

Vector-Field Reduction Theorem

Let $C_* = (C_p, d_p \beta_p)_p$ be a free chain complex and $V = \{(\sigma_i, \beta_i)\}_{i \in \beta}$ be an admissible discrete vector field on C_*. Then the vector field V defines a canonical reduction $\rho = (f, g, h) : (C_p, d_p) \Longrightarrow (C_p^c, d_p')$ where $C_p^c = \mathbb{Z}\left[\beta_p^c\right]$ is the free $\mathbb{Z}-$module generated by the critical p-cells

A. Romero and F. Sergeraert. Discrete Vector Fields and Fundamental Algebraic Topology, 2010. http://arxiv.org/abs/1005.5685v1.

Discrete vector field over matrices

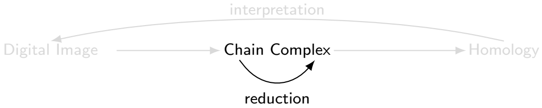


- 2D-images
- Chain complex associated with an image is finite

$$0 \leftarrow C_0 \overset{d_1}{\longleftarrow} C_1 \overset{d_2}{\longleftarrow} C_2 \leftarrow 0$$

- Differential maps can be represented by integer matrices
- Reduction chain complex $\longrightarrow$ Reduction matrices

Discrete vector field over matrices

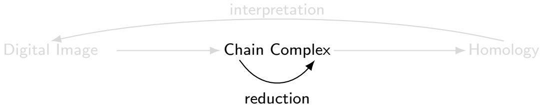


Definition

A *vector field* V for a matrix $M \in Mat_{m,n}(\mathbb{Z})$ is a set of integer pairs $\{(a_i, b_i)\}_i$ satisfying these conditions:

1. $1 \leq a_i \leq m$ and $1 \leq b_i \leq n$
2. The entry $M[a_i, b_i]$ is ± 1
3. The indices a_i (respectively b_i) are pairwise different

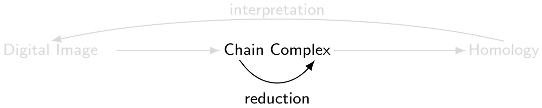
Reduction of Chain Complex



Goal

- Let M_n be a finite matrix which represents the differential map d_n of C_*
  - Compute an admissible discrete vector field V from M_n
  - Obtain a new matrix $\widehat{M}_n$ from M_n and V

Reduction of Chain Complex

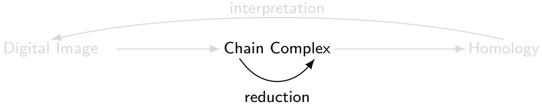


Goal

- Let M_n be a finite matrix which represents the differential map d_n of C_*
  - Compute an admissible discrete vector field V from M_n
  - Obtain a new matrix $\widehat{M}_n$ from M_n and V

In our case, we have to reduce two matrices M_1 and M_2. Compute the homology groups of C_* with $\widehat{M}_1$ and $\widehat{M}_2$ can be much faster.

Reduction of Chain Complex



Algorithm 1

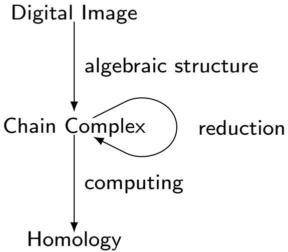
Input: an integer matrix M_n
Output: an admissible discrete vector field V

Algorithm 2

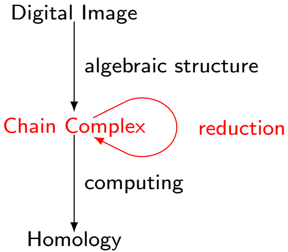
Input: an integer matrix M_n
Output: a reduced matrix $\widehat{M}_n$

- Implemented in Haskell

The method


Digital Image
algebraic structure
Chain Complex
reduction
computing
Homology

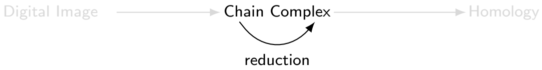
The method


Digital Image
algebraic structure
Chain Complex
reduction
computing
Homology

Coq/SSReflect

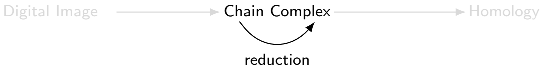
- Coq
  - Theorem Prover tool
  - High-order logic
- SSReflect
  - Extension of Coq
  - Introduce new tactics and libraries
  - Used to formalize of the Four Colour Theorem

Reduction of chain complex



- Steps
  1. Translate our *Haskell* code into the *Coq* language
  2. Define the test functions to specify the properties which our programs must satisfy
  3. State and prove the lemmas which ensure the correctness of our programs

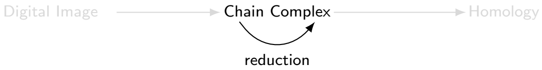
Reduction of chain complex



- Steps
  1. Translate our *Haskell* code into the *Coq* language
  2. Define the test functions to specify the properties which our programs must satisfy
  3. State and prove the lemmas which ensure the correctness of our programs

Example: Let M be an integer matrix, `(vectorCvd M)` builds an admissible discrete vector field

Reduction of chain complex



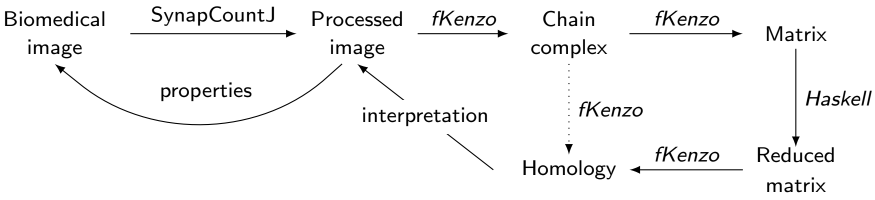
- Steps
  1. Translate our *Haskell* code into the *Coq* language
  2. Define the test functions to specify the properties which our programs must satisfy
  3. State and prove the lemmas which ensure the correctness of our programs

Example: Let M be an integer matrix, `(vectorCvd M)` builds an admissible discrete vector field

```
Lemma admissible-vf:
forall M, (int-matrix M) -> (admissible (vectorCvd M))
```

Table of Contents

Conclusions



- Methodology to study Biomedical images
- Programs partially verified with Theorem Prover tools
- Application to count synapses

Further work

- Verification of our *Haskell* programs by means of *Coq/SSReflect* is still an ongoing work
- Verification of Smith Normal Form of a matrix
- Find other applications of our homological tools in the Biomedical imaging context

Homological Processing of Biomedical digital images: automation and certification*

Jónathan Heras, Gadea Mata, María Poza and Julio Rubio

Department of Mathematics and Computer Science
University of La Rioja
Spain

June 27, 2011

* Partially supported by Ministerio de Educación y Ciencia, project MTM2009-13842-C02-01, and by European Commission FP7, STREP project ForMath, n. 243847