Towards a certified computation of homology groups for digital images*

J. Heras[1], M. Dénès[2], G. Mata[1], A. Mörtberg[3], M. Poza[1] and V. Siles[3]

[1]Department of Mathematics and Computer Science, University of La Rioja - [2]INRIA Sophia-Antipolis - [3]University of Gothenburg

4th International Workshop on Computational Topology in Image Context (CTIC 2012)

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

Motivation

R. Mikhailov and J. Wu. On homotopy groups of the suspended classifying spaces. Algebraic and Geometric Topology 10(2010), 565 – 625.

- Computable aim (without computers!)
- Theorem 5.4: Let A_4 be the 4-th alternating group. Then $\pi_4(\Sigma K(A_4, 1)) = \mathbb{Z}_4$

Motivation

R. Mikhailov and J. Wu. On homotopy groups of the suspended classifying spaces. Algebraic and Geometric Topology 10(2010), 565 – 625.

- Computable aim (without computers!)
- Theorem 5.4: Let A_4 be the 4-th alternating group. Then $\pi_4(\Sigma K(A_4, 1)) = \mathbb{Z}_4$

A. Romero and J. Rubio. Homotopy groups of suspended classifying spaces: an experimental approach. To appear in Mathematics of Computation.

- Kenzo computing homotopy groups
- $\pi_4(\Sigma K(A_4, 1)) = \mathbb{Z}_{12}$

Motivation

R. Mikhailov and J. Wu. On homotopy groups of the suspended classifying spaces. Algebraic and Geometric Topology 10(2010), 565 – 625.

- Computable aim (without computers!)
- Theorem 5.4: Let A_4 be the 4-th alternating group. Then $\pi_4(\Sigma K(A_4, 1)) = \mathbb{Z}_4$

A. Romero and J. Rubio. Homotopy groups of suspended classifying spaces: an experimental approach. To appear in Mathematics of Computation.

- Kenzo computing homotopy groups
- $\pi_4(\Sigma K(A_4, 1)) = \mathbb{Z}_{12}$

To make a long story short: Kenzo solution was the correct one

Motivation

R. Mikhailov and J. Wu. On homotopy groups of the suspended classifying spaces. Algebraic and Geometric Topology 10(2010), 565 – 625.

- Computable aim (without computers!)
- Theorem 5.4: Let A_4 be the 4-th alternating group. Then $\pi_4(\Sigma K(A_4, 1)) = \mathbb{Z}_4$

A. Romero and J. Rubio. Homotopy groups of suspended classifying spaces: an experimental approach. To appear in Mathematics of Computation.

- Kenzo computing homotopy groups
- $\pi_4(\Sigma K(A_4, 1)) = \mathbb{Z}_{12}$

To make a long story short: Kenzo solution was the correct one

In that situation

Analyze correctness of programs to ensure correctness of mathematical results

Goal

Goal

Certified computations of homology groups from biomedical digital images

Goal

Goal

Certified computations of homology groups from biomedical digital images

- Need of trustworthy tools

Goal

Goal

Certified computations of homology groups from biomedical digital images

- Need of trustworthy tools
- Formally verified using an interactive proof assistant

Interactive Proof Assistants

- What is an Interactive Proof Assistant?
  - Software tool for the development of formal proofs
  - Man-Machine collaboration:
    - Human: design the proofs
    - Machine: fill the gaps
  - Examples: Isabelle, Hol, ACL2, Coq, ...

Interactive Proof Assistants

- What is an Interactive Proof Assistant?
  - Software tool for the development of formal proofs
  - Man-Machine collaboration:
    - Human: design the proofs
    - Machine: fill the gaps
  - Examples: Isabelle, Hol, ACL2, Coq, . . .
- Applications:
  - Mathematical proofs:
    - Four Color Theorem
    - Fundamental Theorem of Algebra
    - Kepler conjecture
  - Software and Hardware verification:
    - C compiler
    - AMD5K86 microprocessor
    - . . .

Coq/SSReflect

- Coq:
  - An Interactive Proof Assistant
  - Based on Calculus of Inductive Constructions
  - Interesting feature: program extraction from a constructive proof

Coq/SSReflect

- Coq:
  - An Interactive Proof Assistant
  - Based on Calculus of Inductive Constructions
  - Interesting feature: program extraction from a constructive proof
- SSReflect:
  - Extension of Coq
  - Developed while formalizing the Four Color Theorem by G. Gonthier
  - Currently, it is used in the formalization of Feit-Thompson Theorem

Coq/SSReflect

- Coq:
  - An Interactive Proof Assistant
  - Based on Calculus of Inductive Constructions
  - Interesting feature: program extraction from a constructive proof
- SSReflect:
  - Extension of Coq
  - Developed while formalizing the Four Color Theorem by G. Gonthier
  - Currently, it is used in the formalization of Feit-Thompson Theorem
- Demo

The ForMath project

The ForMath project:

- European project
- ForMath: Formalization of Mathematics

The ForMath project

The ForMath project:

- European project
- ForMath: Formalization of Mathematics
- Develop libraries of formalized mathematics:
  - Algebra
  - Linear Algebra
  - Real number computation
  - Algebraic Topology

The ForMath project

The ForMath project:

- European project
- ForMath: Formalization of Mathematics
- Develop libraries of formalized mathematics:
  - Algebra
  - Linear Algebra
  - Real number computation
  - Algebraic Topology

General Goal

An Algebraic Topology library formalized in COQ/SSREFLECT

Table of Contents

Table of Contents

Verification in Coq/SSReflect

Algebraic Topology notions already formalized in Coq/SSReflect:

Verification in Coq/SSReflect

Algebraic Topology notions already formalized in Coq/SSReflect:

Simplicial Complexes:

```
(* V is the vertex set *)
Variable V : finType.

(* A simplex over V is a finite subset of V *)
Definition simplex := {set V}.

(* Simplicial Complex *)
Definition simplicial_complex (c : {set simplex}) :=
   forall x, x \in c -> forall y : simplex, y \subset x -> y \in c.

(* Generation of simplicial complex from a list of simplices *)

Definition create_sc (s : seq simplex) : {set simplex} :=
  \bigcup_(sp <- s) powerset sp.

Lemma create_sc_correct : forall s, simplicial_complex (create_sc s).
```

Verification in Coq/SSReflect

Algebraic Topology notions already formalized in Coq/SSReflect:

Boundary Matrices of Simplicial Complexes:

```
Lemma incidence_matrices_sc_product:
  forall (V:finType) (n:nat) (sc: {set (simplex V)}),
    simplicial_complex sc ->
       (incidence_mx_n sc n) *m (incidence_mx_n sc (n.+1)) = 0.
```

Verification in Coq/SSReflect

Algebraic Topology notions already formalized in Coq/SSReflect:

Boundary Matrices of Simplicial Complexes:

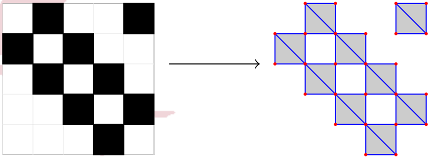
```
Lemma incidence_matrices_sc_product:
  forall (V:finType) (n:nat) (sc: {set (simplex V)}),
    simplicial_complex sc ->
        (incidence_mx_n sc n) *m (incidence_mx_n sc (n.+1)) = 0.
```

Homology:

```
Variable (K : fieldType) (V1 V2 V3 : vectType K)
         (f : linearApp V1 V2) (g : linearApp V2 V3).

Definition Homology := ((lker g) :\: (limg f)).

Definition dim_homology (mxf:'M[K]_(v1,v2)) (mxg:'M[K]_(v2,v3)) :=
  v2 - \rank mxg - \rank mxf.

Lemma dimHomologyrankE: mxf *m mxg = 0 ->
  \dim Homology (LinearApp mxf) (LinearApp mxg) = dim_homology mxf mxg.
```

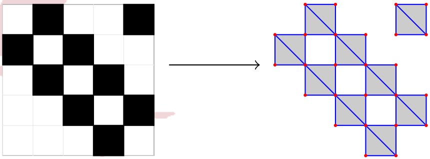
Verification in Coq/SSReflect

Connection with digital topology:



Verification in Coq/SSReflect

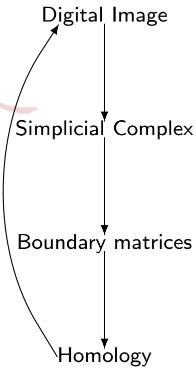
Connection with digital topology:



- An image is represented by means of a list of lists of booleans
- Function `createfacets` generates a list of simplexes from an image
- Correctness of `createfacets` has been proved

Verification in Coq/SSReflect

Putting all the pieces together:



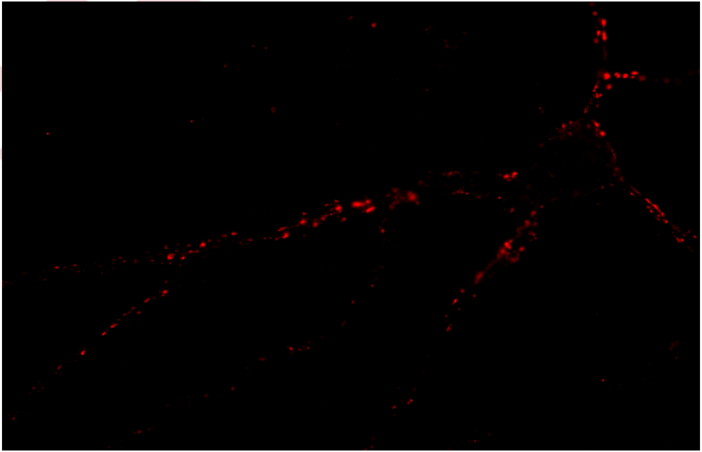
Achievement

Certified computation of Homology from digital images

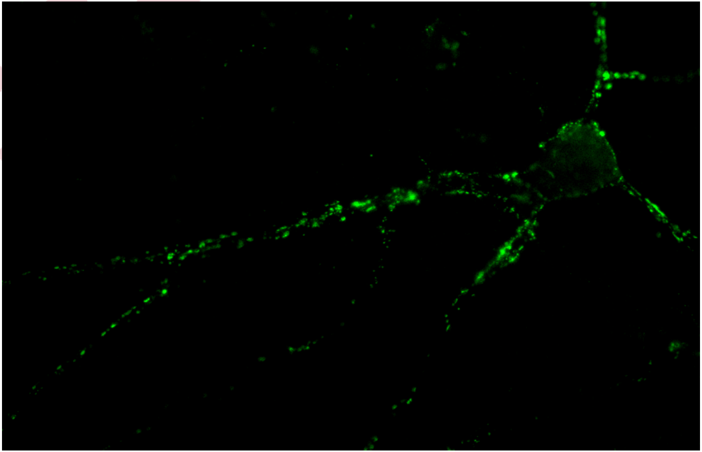
Testing our framework: Counting synapses

- Synapses are the points of connection between neurons
- Relevance: Computational capabilities of the brain
- Procedures to modify the synaptic density may be an important asset in the treatment of neurological diseases
- An automated and reliable method is necessary

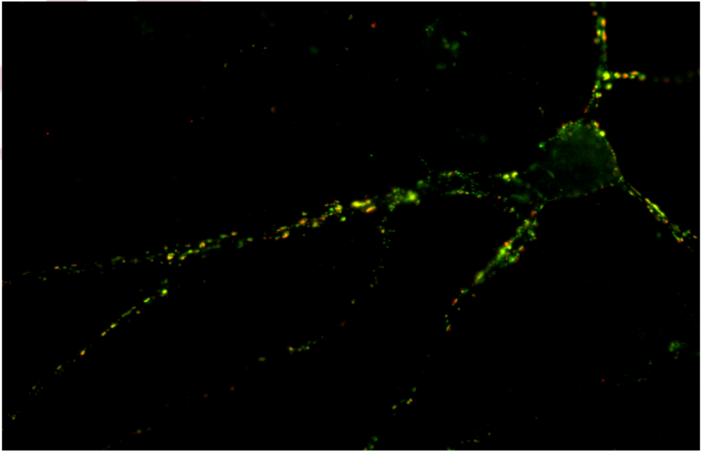
Testing our framework: Counting synapses



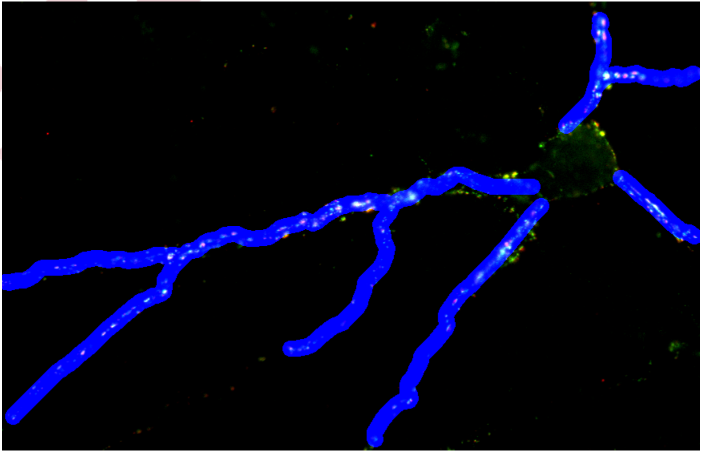
Testing our framework: Counting synapses



Testing our framework: Counting synapses



Testing our framework: Counting synapses



Testing our framework: Counting synapses

Testing our framework: Counting synapses

- Counting synapses:
  - Measure the number of connected components of the image
  - Good benchmark to test our framework: computation of H_0

Testing our framework: Counting synapses

- Counting synapses:
  - Measure the number of connected components of the image
  - Good benchmark to test our framework: computation of H_0
  - SynapCountJ: software to measure synaptic density evolution

Table of Contents

Demo

Demo

Computing inside Coq

- Coq is a Proof Assistant and not a Computer Algebra system

Computing inside Coq

- Coq is a Proof Assistant and not a Computer Algebra system
- Efficient implementations of mathematical algorithms inside Coq is an ongoing effort

Computing inside Coq

- Coq is a Proof Assistant and not a Computer Algebra system
- Efficient implementations of mathematical algorithms inside Coq is an ongoing effort
- Achieving a better efficiency:
  - Extraction mechanism
  - Internal compilation of Coq terms to OCaml
  - Sparse matrices
  - Better representations for images
  - Reduction of matrices keeping the same homological information

Computing inside Coq

- Coq is a Proof Assistant and not a Computer Algebra system
- Efficient implementations of mathematical algorithms inside Coq is an ongoing effort
- Achieving a better efficiency:
  - Extraction mechanism
  - Internal compilation of Coq terms to OCaml
  - Sparse matrices
  - Better representations for images
  - Reduction of matrices keeping the same homological information

Table of Contents

Discrete Vector Fields + Effective Homology

Reduction process:

- (Algebraic setting of) Discrete Morse Theory

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

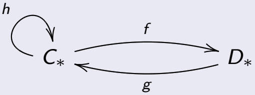
- Effective Homology

J. Rubio and F. Sergeraert. Constructive Algebraic Topology. Bulletin des Sciences Mathématiques, 2002, vol. 126, pp. 389-412

Effective Homology

Definition

A reduction ρ between two chain complexes C_* y D_* (denoted by $\rho : C_* \Rightarrow D_*$) is a triple $\rho = (f, g, h)$



satisfying the following relations:

1) $fg = \mathrm{Id}_{D_*}$;
2) $d_C h + h d_C = \mathrm{Id}_{C_*} - gf$;
3) $fh = 0; \quad hg = 0; \quad hh = 0.$

Theorem

If $C_* \Rightarrow D_*$, then $C_* \cong D_* \oplus A_*$, with A_* acyclic, which implies that $H_n(C_*) \cong H_n(D_*)$ for all n.

Discrete Vector Fields

Definition

Let $C_* = (C_p, d_p)_{p\in\mathbb{Z}}$ *a free chain complex with distinguished* $\mathbb{Z}$*-basis* $\beta_p \subset C_p$*. A discrete vector field* V *on* C_* *is a collection of pairs* $V = \{(\sigma_i; \tau_i)\}_{i\in I}$ *satisfying the conditions:*

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

Discrete Vector Fields

Definition

Let $C_* = (C_p, d_p)_{p \in \mathbb{Z}}$ a free chain complex with distinguished $\mathbb{Z}$-basis $\beta_p \subset C_p$. A discrete vector field V on C_* is a collection of pairs $V = \{(\sigma_i; \tau_i)\}_{i \in I}$ satisfying the conditions:

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

Definition

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

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

Discrete Vector Fields

Definition

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

Discrete Vector Fields

Definition

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

Definition

A cell σ which does not appear in a discrete vector field V is called a critical cell.

Discrete Vector Fields

Definition

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

Definition

A cell σ which does not appear in a discrete vector field V is called a critical cell.

Theorem

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

Vector fields and integer matrices

M is one of the boundary matrices of the chain complex:

$$\ldots \leftarrow \mathbb{Z}^m \stackrel{M}{\longleftarrow} \mathbb{Z}^n \leftarrow \ldots$$

Vector fields and integer matrices

M is one of the boundary matrices of the chain complex:

$$\ldots \leftarrow \mathbb{Z}^m \xleftarrow{M} \mathbb{Z}^n \leftarrow \ldots$$

Definition

An admissible vector field V for M is nothing but a set of integer pairs $\{(a_i, b_i)\}$ satisfying these conditions:

1. $1 \leq a_i \leq m$ *and* $1 \leq b_i \leq n$
2. *The entry* $M[a_i, b_i]$ *of the matrix is* ± 1
3. *The indices* a_i *(resp.* b_i*) are pairwise different*
4. *Non existence of loops*

Vector fields and integer matrices

M is one of the boundary matrices of the chain complex:

$$\ldots \leftarrow \mathbb{Z}^m \stackrel{M}{\leftarrow} \mathbb{Z}^n \leftarrow \ldots$$

Definition

An admissible vector field V for M is nothing but a set of integer pairs $\{(a_i, b_i)\}$ satisfying these conditions:

1. $1 \leq a_i \leq m$ *and* $1 \leq b_i \leq n$
2. *The entry* $M[a_i, b_i]$ *of the matrix is* ± 1
3. *The indices* a_i *(resp.* b_i*) are pairwise different*
4. *Non existence of loops*

Vector fields and integer matrices

M is one of the boundary matrices of the chain complex:

$$\ldots \leftarrow \mathbb{Z}^m \stackrel{M}{\longleftarrow} \mathbb{Z}^n \leftarrow \ldots$$

Definition

An admissible vector field V for M is nothing but a set of integer pairs $\{(a_i, b_i)\}$ satisfying these conditions:

1. $1 \leq a_i \leq m$ *and* $1 \leq b_i \leq n$
2. *The entry $M[a_i, b_i]$ of the matrix is* ± 1
3. *The indices a_i (resp. b_i) are pairwise different*
4. *Non existence of loops*

Vector fields and integer matrices

M is one of the boundary matrices of the chain complex:

$$\ldots \leftarrow \mathbb{Z}^m \stackrel{M}{\longleftarrow} \mathbb{Z}^n \leftarrow \ldots$$

Definition

An admissible vector field V for M is nothing but a set of integer pairs $\{(a_i, b_i)\}$ satisfying these conditions:

1. $1 \leq a_i \leq m$ *and* $1 \leq b_i \leq n$
2. *The entry* $M[a_i, b_i]$ *of the matrix is* ± 1
3. *The indices* a_i *(resp.* b_i*) are pairwise different*
4. *Non existence of loops*

```
Definition admissible_dvf (M: 'M['F_2]_(m,n)) (V: seq ('I_m * 'I_n)) (
    ords : simpl_rel 'I_m) :=
  all [pred p | M p.1 p.2 == 1] V &&
  uniq (map (@fst _ _) V) && uniq (map (@snd _ _) V) &&
  all [pred i | ~~ (connect ords i i)] (map (@fst _ _) V).
```

Romero-Sergeraert algorithms

Algorithm (Romero-Sergeraert 2010)

Input: A matrix M
Output: An admissible discrete vector field for M

- Implemented with the function `gen_adm_dvf`
- Verified in COQ/SSREFLECT

```
Lemma admissible_gen_adm_dvf m n (M : 'M['F_2]_(m,n)) :
  let (V,ords) := gen_adm_dvf M in admissible M V ord.
```

Romero-Sergeraert algorithms

Algorithm (Romero-Sergeraert 2010)

Input: A matrix M
Output: An admissible discrete vector field for M

- Implemented with the function `gen_adm_dvf`
- Verified in Coq/SSReflect

```
Lemma admissible_gen_adm_dvf m n (M : 'M['F_2]_(m,n)) :
  let (V,ords) := gen_adm_dvf M in admissible M V ord.
```

Algorithm (Romero-Sergeraert 2010)

Input: A matrix M and an admissible discrete vector field of M
Output: A reduced matrix M'

- Already implemented
- Verification in Coq/SSReflect is an ongoing work

Results

Some remarkable results have been obtained:

- 500 randomly generated images:
  - Initial size of matrices: $\sim 100 \times 300 \to \sim 12$ seconds
  - After reduction process: $\sim 5 \times 50 \to$ milliseconds
  - Most of reduced matrices were null

Results

Some remarkable results have been obtained:

- 500 randomly generated images:
  - Initial size of matrices: $\sim 100 \times 300 \rightarrow \sim 12$ seconds
  - After reduction process: $\sim 5 \times 50 \rightarrow$ milliseconds
  - Most of reduced matrices were null
- Biomedical images:
  - Initial size of matrices: $\sim 700 \times 1400 \rightarrow \infty$
  - After reduction process just 25 seconds

Table of Contents

Conclusions

- Towards an Algebraic Topology Formal library

Conclusions

- Towards an Algebraic Topology Formal library
- Certified computation of Homology from digital images

Conclusions

- Towards an Algebraic Topology Formal library
- Certified computation of Homology from digital images
- Discrete Vector Fields to deal with images

Conclusions

- Towards an Algebraic Topology Formal library
- Certified computation of Homology from digital images
- Discrete Vector Fields to deal with images
- Application to a biomedical problem: counting synapses

Further work

- Formalization aspects:
  - Correctness of reduction process
  - Integer homology computation

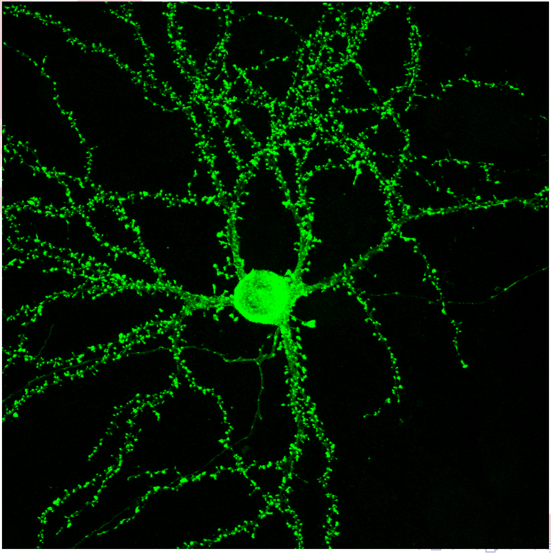
Further work

- Formalization aspects:
  - Correctness of reduction process
  - Integer homology computation
- Efficiency issues:
  - Better representations and more efficient algorithms

Further work

- Formalization aspects:
  - Correctness of reduction process
  - Integer homology computation
- Efficiency issues:
  - Better representations and more efficient algorithms
- Homology certified programs applied to more interesting biomedical cases:
  - Homology group in dimension 1 (structure detection)
  - Persistent Homology (denoising)

Further work



Towards a certified computation of homology groups for digital images

J. Heras[1], M. Dénès[2], G. Mata[1], A. Mörtberg[3], M. Poza[1] and V. Siles[3]

[1]Department of Mathematics and Computer Science, University of La Rioja - [2]INRIA Sophia-Antipolis - [3]University of Gothenburg

4th International Workshop on Computational Topology in Image Context (CTIC 2012)