Branching Networks I

Complex Networks
CSYS/MATH 303, Spring, 2011

Prof. Peter Dodds

Department of Mathematics & Statistics
Center for Complex Systems
Vermont Advanced Computing Center
University of Vermont

Licensed under the *Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License*.

Outline

Introduction
Definitions
Allometry
Laws
Stream Ordering
Horton's Laws
Tokunaga's Law
Nutshell
References

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

The University of Vermont

Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

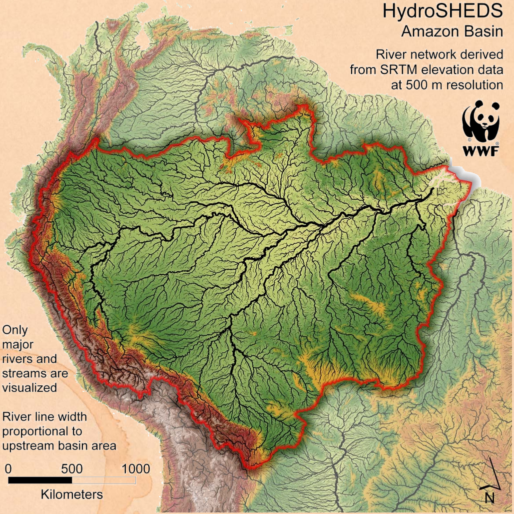
Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Branching networks are everywhere...

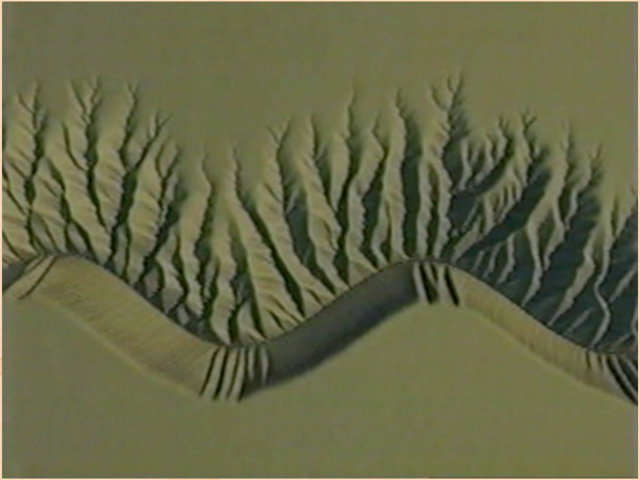


http://hydrosheds.cr.usgs.gov/ (⊞)

Branching networks are everywhere...

http://en.wikipedia.org/wiki/Image:Applebox.JPG (⊞)

A beautiful simulation of erosion:



Bruce Shaw (LDEO, Columbia) and Marcelo Magnasco (Rockefeller)

Geomorphological networks

Definitions

- Drainage basin for a point p is the complete region of land from which overland flow drains through p.
- Definition most sensible for a point in a stream.
- Recursive structure: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...

Geomorphological networks

Definitions

- Drainage basin for a point p is the complete region of land from which overland flow drains through p.
- Definition most sensible for a point in a stream.
- Recursive structure: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...

Geomorphological networks

Definitions

- Drainage basin for a point p is the complete region of land from which overland flow drains through p.
- Definition most sensible for a point in a stream.
- Recursive structure: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...

Geomorphological networks

Definitions

- Drainage basin for a point p is the complete region of land from which overland flow drains through p.
- Definition most sensible for a point in a stream.
- Recursive structure: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...

Geomorphological networks

Definitions

- Drainage basin for a point p is the complete region of land from which overland flow drains through p.
- Definition most sensible for a point in a stream.
- Recursive structure: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...

Geomorphological networks

Definitions

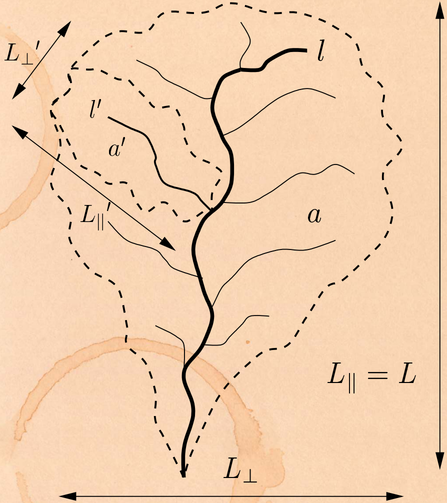
- Drainage basin for a point p is the complete region of land from which overland flow drains through p.
- Definition most sensible for a point in a stream.
- Recursive structure: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...

Geomorphological networks

Definitions

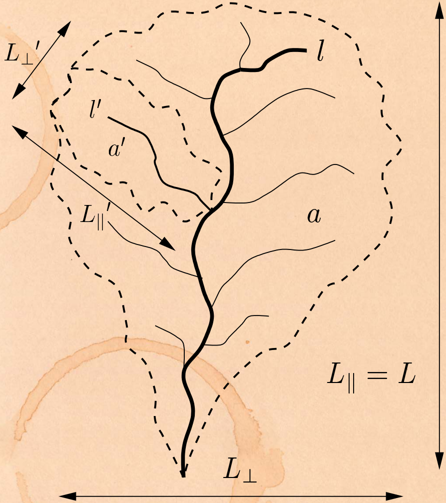
- Drainage basin for a point p is the complete region of land from which overland flow drains through p.
- Definition most sensible for a point in a stream.
- Recursive structure: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...

Basic basin quantities: a, l, $L_{\parallel}$, $L_{\perp}$:



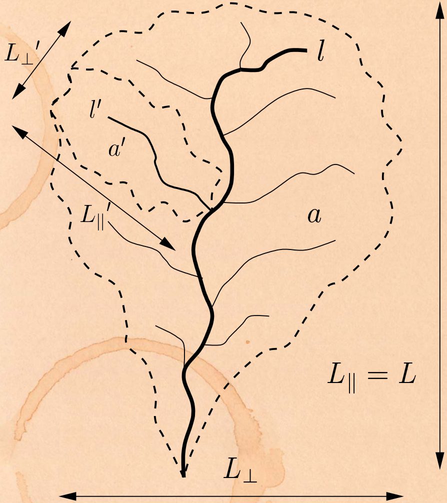
- a = drainage basin area
- ℓ = length of longest (main) stream (which may be fractal)
- $L = L_{\parallel}$ = longitudinal length of basin
- $L = L_{\perp}$ = width of basin

Basic basin quantities: a, l, $L_{\parallel}$, $L_{\perp}$:



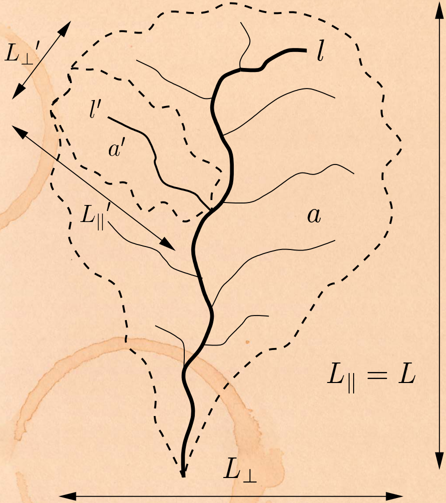
- a = drainage basin area
- ℓ = length of longest (main) stream (which may be fractal)
- $L = L_{\parallel}$ = longitudinal length of basin
- $L = L_{\perp}$ = width of basin

Basic basin quantities: a, l, $L_{\parallel}$, $L_{\perp}$:



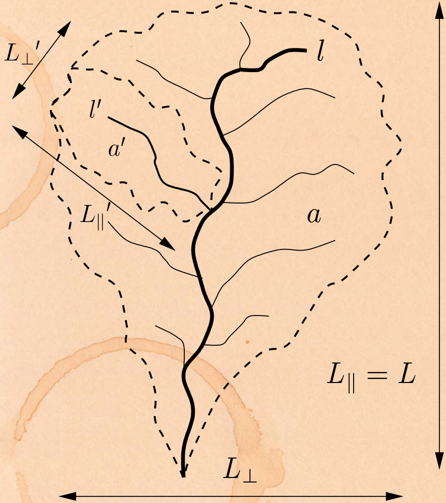
- a = drainage basin area
- ℓ = length of longest (main) stream (which may be fractal)
- $L = L_{\parallel}$ = longitudinal length of basin
- $L = L_{\perp}$ = width of basin

Basic basin quantities: a, l, $L_{\parallel}$, $L_{\perp}$:



- a = drainage basin area
- ℓ = length of longest (main) stream (which may be fractal)
- $L = L_{\parallel}$ = longitudinal length of basin
- $L = L_{\perp}$ = width of basin

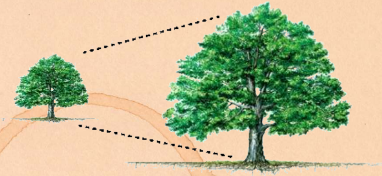
Basic basin quantities: a, l, $L_\parallel$, $L_\perp$:



- a = drainage basin area
- ℓ = length of longest (main) stream (which may be fractal)
- $L = L_\parallel$ = longitudinal length of basin
- $L = L_\perp$ = width of basin

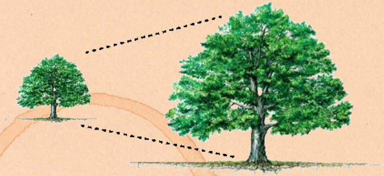
Allometry

- Isometry:
  dimensions scale linearly with each other.

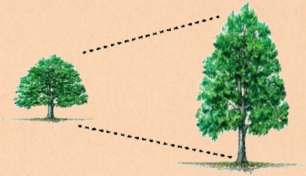


Allometry

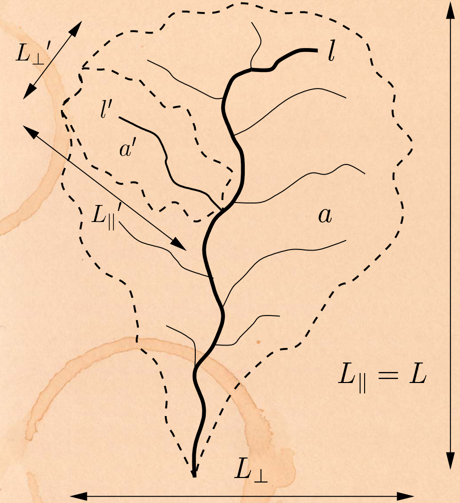
- Isometry:
  dimensions scale linearly with each other.



- Allometry:
  dimensions scale nonlinearly.



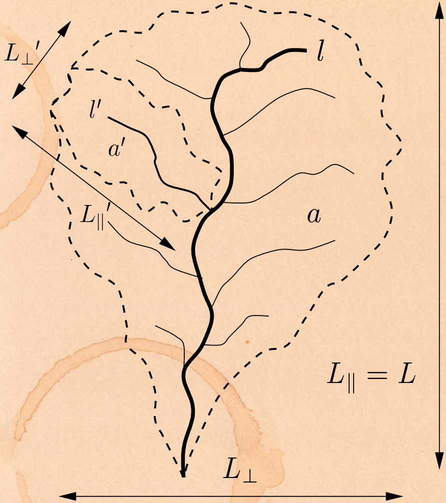
Basin allometry



Allometric relationships:

- $$\ell \propto a^h$$
- $$\ell \propto L^d$$
- Combine above:
$$a \propto L^{d/h} \equiv L^D$$

Basin allometry

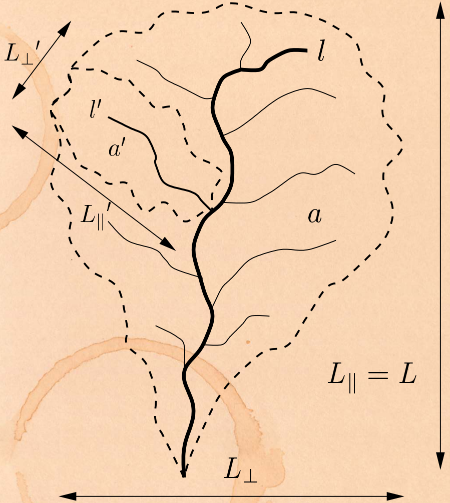


Allometric relationships:

- $\ell \propto a^h$
- $\ell \propto L^d$
- Combine above:

  $a \propto L^{d/h} \equiv L^D$

Basin allometry

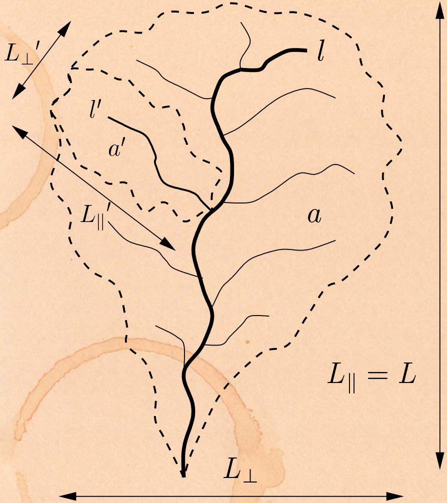


Allometric relationships:

- $\ell \propto a^h$
- $\ell \propto L^d$
- Combine above:

$$a \propto L^{d/h} = L^D$$

Basin allometry



Allometric relationships:

- $$\ell \propto a^h$$
- $$\ell \propto L^d$$
- Combine above:

$$a \propto L^{d/h} \equiv L^D$$

‘Laws’

- Hack’s law (1957) [2]:

$$\ell \propto a^h$$

reportedly $0.5 < h < 0.7$

- Scaling of main stream length with basin size:

$$\ell \propto L_{\parallel}^{d}$$

reportedly $1.0 < d < 1.1$

- Basin allometry:

$$L_{\parallel} \propto a^{h/d} \equiv a^{1/D}$$

$D < 2 \rightarrow$ basins elongate.

‘Laws’

- Hack’s law (1957) [2]:

$$\boxed{\ell \propto a^h}$$

reportedly $0.5 < h < 0.7$

- Scaling of main stream length with basin size:

$$\boxed{\ell \propto L_{\parallel}^{d}}$$

reportedly $1.0 < d < 1.1$

- Basin allometry:

$$\boxed{L_{\parallel} \propto a^{h/d} \equiv a^{1/D}}$$

$D < 2 \rightarrow$ basins elongate.

'Laws'

- Hack's law (1957) [2]:

$$\ell \propto a^h$$

reportedly $0.5 < h < 0.7$

- Scaling of main stream length with basin size:

$$\ell \propto L_{\parallel}^{d}$$

reportedly $1.0 < d < 1.1$

- Basin allometry:

$$L_{\parallel} \propto a^{h/d} \equiv a^{1/D}$$

$D < 2 \rightarrow$ basins elongate.

There are a few more 'laws':[1]

Relation:	Name or description:
$T_k = T_1(R_T)^k$	Tokunaga's law
$\ell \sim L^d$	self-affinity of single channels
$n_\omega / n_{\omega+1} = R_n$	Horton's law of stream numbers
$\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell$	Horton's law of main stream lengths
$\bar{a}_{\omega+1} / \bar{a}_\omega = R_a$	Horton's law of basin areas
$\bar{s}_{\omega+1} / \bar{s}_\omega = R_s$	Horton's law of stream segment lengths
$L_\perp \sim L^H$	scaling of basin widths
$P(a) \sim a^{-\tau}$	probability of basin areas
$P(\ell) \sim \ell^{-\gamma}$	probability of stream lengths
$\ell \sim a^h$	Hack's law
$a \sim L^D$	scaling of basin areas
$\Lambda \sim a^\beta$	Langbein's law
$\lambda \sim L^\varphi$	variation of Langbein's law

Reported parameter values: [1]

Parameter:	Real networks:
R_n	3.0–5.0
R_a	3.0–6.0
$R_\ell = R_T$	1.5–3.0
T_1	1.0–1.5
d	1.1 ± 0.01
D	1.8 ± 0.1
h	0.50–0.70
τ	1.43 ± 0.05
γ	1.8 ± 0.1
H	0.75–0.80
β	0.50–0.70
φ	1.05 ± 0.05

Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values

Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values

Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values

Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values

Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values

For (3): Many attempts: not yet sorted out...

Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Term: Horton-Strahler Stream Ordering [4]
- Can be seen as iterative trimming of a network.

Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Term: Horton-Strahler Stream Ordering [4]
- Can be seen as iterative trimming of a network.

Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Term: Horton-Strahler Stream Ordering [4]
- Can be seen as iterative trimming of a network.

Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Term: Horton-Strahler Stream Ordering [4]
- Can be seen as iterative trimming of a network.

Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Term: Horton-Strahler Stream Ordering [4]
- Can be seen as iterative trimming of a network.

Stream Ordering:

Some definitions:

- A channel head is a point in landscape where flow becomes focused enough to form a stream.
- A source stream is defined as the stream that reaches from a channel head to a junction with another stream.
- Roughly analogous to capillary vessels.
- Use symbol $\omega = 1, 2, 3, \ldots$ for stream order.

Stream Ordering:

Some definitions:

- A channel head is a point in landscape where flow becomes focused enough to form a stream.
- A source stream is defined as the stream that reaches from a channel head to a junction with another stream.
- Roughly analogous to capillary vessels.
- Use symbol $\omega = 1, 2, 3, \ldots$ for stream order.

Stream Ordering:

Some definitions:

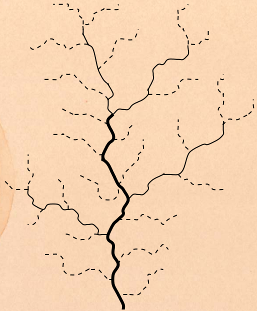
- A channel head is a point in landscape where flow becomes focused enough to form a stream.
- A source stream is defined as the stream that reaches from a channel head to a junction with another stream.
- Roughly analogous to capillary vessels.
- Use symbol $\omega = 1, 2, 3, \ldots$ for stream order.

Stream Ordering:

Some definitions:

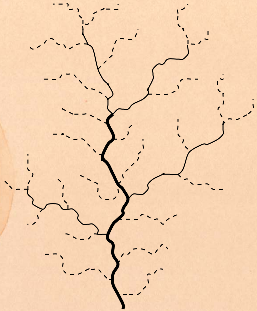
- A channel head is a point in landscape where flow becomes focused enough to form a stream.
- A source stream is defined as the stream that reaches from a channel head to a junction with another stream.
- Roughly analogous to capillary vessels.
- Use symbol $\omega = 1, 2, 3, ...$ for stream order.

Stream Ordering:



1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

Stream Ordering:



1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

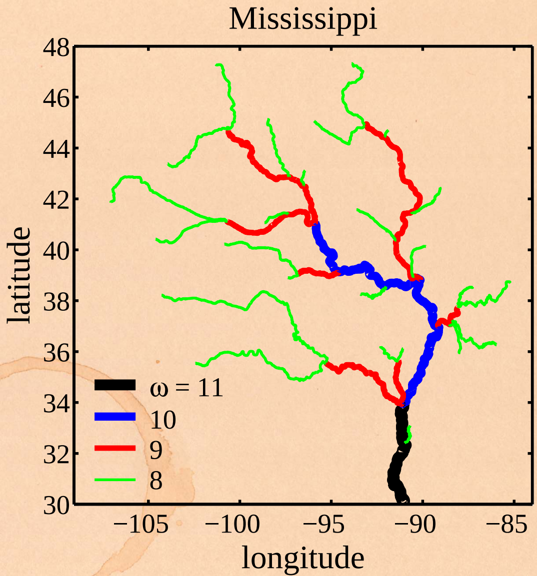
Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = Ω)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$.

Stream Ordering—A large example:



[source=/data6/dodds/work/rivers/dems/mississippi/figures/figorder_paths_mispi10.ps]

[21−Mar−2000 peter dodds]

Stream Ordering:

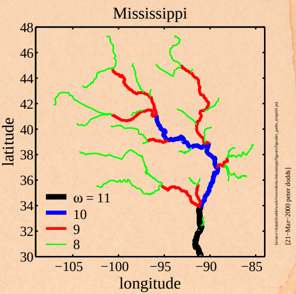
Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders ω_1 and ω_2 meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:

$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$

where δ is the Kronecker delta.



Stream Ordering:

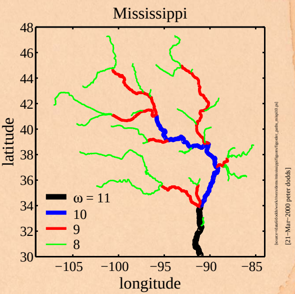
Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders ω_1 and ω_2 meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:

$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1,\omega_2}$$

where δ is the Kronecker delta.



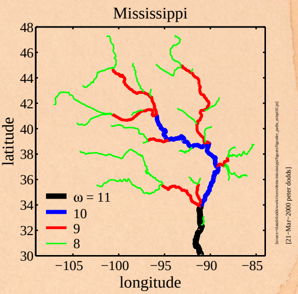
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
- If streams of different orders ω_1 and ω_2 meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:

$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$

where δ is the Kronecker delta.



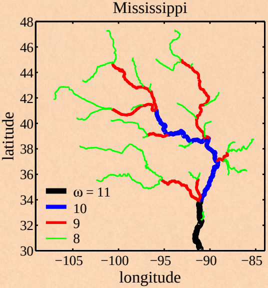
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
- If streams of different orders ω_1 and ω_2 meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:

$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$

where δ is the Kronecker delta.



Stream Ordering:

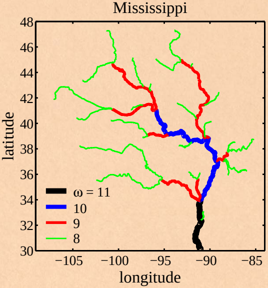
Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders ω_1 and ω_2 meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:

$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$

where δ is the Kronecker delta.



Stream Ordering:

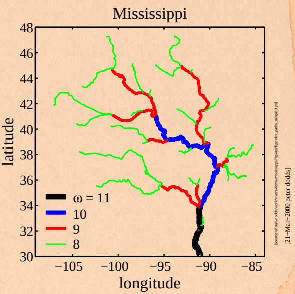
Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders ω_1 and ω_2 meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:

$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1,\omega_2}$$

where δ is the Kronecker delta.



Stream Ordering:

One problem:

- Resolution of data messes with ordering
- Micro-description changes (e.g., order of a basin may increase)
- ... but relationships based on ordering appear to be robust to resolution changes.

Stream Ordering:

One problem:

- Resolution of data messes with ordering
- Micro-description changes (e.g., order of a basin may increase)
- ... but relationships based on ordering appear to be robust to resolution changes.

Stream Ordering:

One problem:

- Resolution of data messes with ordering
- Micro-description changes (e.g., order of a basin may increase)
- ... but relationships based on ordering appear to be robust to resolution changes.

Stream Ordering:

Utility:

- Stream ordering helpfully discretizes a network.
- Goal: understand network architecture

Stream Ordering:

Utility:

- Stream ordering helpfully discretizes a network.
- Goal: understand network architecture

Stream Ordering:

Utility:

- Stream ordering helpfully discretizes a network.
- Goal: understand network architecture

Stream Ordering:

Resultant definitions:

- A basin of order Ω has n_ω streams (or sub-basins) of order ω.
  - $n_\omega > n_{\omega+1}$
- An order ω basin has area a_ω.
- An order ω basin has a main stream length ℓ_ω.
- An order ω basin has a stream segment length s_ω
  1. an order ω stream segment is only that part of the stream which is actually of order ω
  2. an order ω stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams

Stream Ordering:

Resultant definitions:

- A basin of order Ω has n_ω streams (or sub-basins) of order ω.
  - $n_\omega > n_{\omega+1}$
- An order ω basin has area a_ω.
- An order ω basin has a main stream length ℓ_ω.
- An order ω basin has a stream segment length s_ω
  1. an order ω stream segment is only that part of the stream which is actually of order ω
  2. an order ω stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams

Stream Ordering:

Resultant definitions:

- A basin of order Ω has n_ω streams (or sub-basins) of order ω.
  - $n_\omega > n_{\omega+1}$
- An order ω basin has area a_ω.
- An order ω basin has a main stream length ℓ_ω.
- An order ω basin has a stream segment length s_ω
  1. an order ω stream segment is only that part of the stream which is actually of order ω
  2. an order ω stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams

Stream Ordering:

Resultant definitions:

- A basin of order Ω has n_ω streams (or sub-basins) of order ω.
  - $n_\omega > n_{\omega+1}$
- An order ω basin has area a_ω.
- An order ω basin has a main stream length ℓ_ω.
- An order ω basin has a stream segment length s_ω
  1. an order ω stream segment is only that part of the stream which is actually of order ω
  2. an order ω stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams

Stream Ordering:

Resultant definitions:

- A basin of order Ω has n_ω streams (or sub-basins) of order ω.
  - $n_\omega > n_{\omega+1}$
- An order ω basin has area a_ω.
- An order ω basin has a main stream length ℓ_ω.
- An order ω basin has a stream segment length s_ω
  1. an order ω stream segment is only that part of the stream which is actually of order ω
  2. an order ω stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams

Stream Ordering:

Resultant definitions:

- A basin of order Ω has n_ω streams (or sub-basins) of order ω.
  - $n_\omega > n_{\omega+1}$
- An order ω basin has area a_ω.
- An order ω basin has a main stream length ℓ_ω.
- An order ω basin has a stream segment length s_ω
  1. an order ω stream segment is only that part of the stream which is actually of order ω
  2. an order ω stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams

Stream Ordering:

Resultant definitions:

- A basin of order Ω has n_ω streams (or sub-basins) of order ω.
  - $n_\omega > n_{\omega+1}$
- An order ω basin has area a_ω.
- An order ω basin has a main stream length ℓ_ω.
- An order ω basin has a stream segment length s_ω
  1. an order ω stream segment is only that part of the stream which is actually of order ω
  2. an order ω stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams

Horton's laws

Self-similarity of river networks

- First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton's law of stream numbers:

$$n_\omega / n_{\omega+1} = R_n > 1$$

- Horton's law of stream lengths:

$$\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell > 1$$

- Horton's law of basin areas:

$$\bar{a}_{\omega+1} / \bar{a}_\omega = R_a > 1$$

Horton's laws

Self-similarity of river networks

- First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton's law of stream numbers:

$$\boxed{n_\omega / n_{\omega+1} = R_n > 1}$$

- Horton's law of stream lengths:

$$\boxed{\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell > 1}$$

- Horton's law of basin areas:

$$\boxed{\bar{a}_{\omega+1} / \bar{a}_\omega = R_a > 1}$$

Horton's laws

Self-similarity of river networks

- First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton's law of stream numbers:

$$n_\omega / n_{\omega+1} = R_n > 1$$

- Horton's law of stream lengths:

$$\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell > 1$$

- Horton's law of basin areas:

$$\bar{a}_{\omega+1} / \bar{a}_\omega = R_a > 1$$

Horton's laws

Self-similarity of river networks

- First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton's law of stream numbers:

$$n_\omega / n_{\omega+1} = R_n > 1$$

- Horton's law of stream lengths:

$$\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell > 1$$

- Horton's law of basin areas:

$$\bar{a}_{\omega+1} / \bar{a}_\omega = R_a > 1$$

Horton's laws

Self-similarity of river networks

- First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton's law of stream numbers:

$$n_\omega / n_{\omega+1} = R_n > 1$$

- Horton's law of stream lengths:

$$\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell > 1$$

- Horton's law of basin areas:

$$\bar{a}_{\omega+1} / \bar{a}_\omega = R_a > 1$$

Horton's laws

Self-similarity of river networks

- First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton's law of stream numbers:

$$n_\omega / n_{\omega+1} = R_n > 1$$

- Horton's law of stream lengths:

$$\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell > 1$$

- Horton's law of basin areas:

$$\bar{a}_{\omega+1} / \bar{a}_\omega = R_a > 1$$

Horton's laws

Horton's Ratios:

- So... Horton's laws are defined by three ratios:

$$R_n,\ R_\ell,\ \text{and } R_a.$$

- Horton's laws describe exponential decay or growth:

$$\begin{aligned} n_\omega &= n_{\omega-1}/R_n \\ &= n_{\omega-2}/R_n^{\,2} \\ &\ \vdots \\ &= n_1/R_n^{\,\omega-1} \\ &= n_1 e^{-(\omega-1)\ln R_n} \end{aligned}$$

Horton's laws

Horton's Ratios:

- So... Horton's laws are defined by three ratios:

$$R_n,\ R_\ell,\ \text{and } R_a.$$

- Horton's laws describe exponential decay or growth:

$$\begin{aligned} n_\omega &= n_{\omega-1}/R_n \\ &= n_{\omega-2}/R_n^{\,2} \\ &\ \vdots \\ &= n_1/R_n^{\,\omega-1} \\ &= n_1 e^{-(\omega-1)\ln R_n} \end{aligned}$$

Horton's laws

Horton's Ratios:

- So... Horton's laws are defined by three ratios:

$$R_n, \ R_\ell, \text{ and } R_a.$$

- Horton's laws describe exponential decay or growth:

$$\begin{aligned} n_\omega &= n_{\omega-1}/R_n \\ &= n_{\omega-2}/R_n^2 \\ &\vdots \\ &= n_1/R_n^{\omega-1} \\ &= n_1 e^{-(\omega-1)\ln R_n} \end{aligned}$$

The University of Vermont

Horton's laws

Horton's Ratios:

- So... Horton's laws are defined by three ratios:

$$R_n, \; R_\ell, \text{ and } R_a.$$

- Horton's laws describe exponential decay or growth:

$$\begin{aligned} n_\omega &= n_{\omega-1}/R_n \\ &= n_{\omega-2}/R_n^2 \\ &\vdots \\ &= n_1/R_n^{\omega-1} \\ &= n_1 e^{-(\omega-1)\ln R_n} \end{aligned}$$

Horton's laws

Horton's Ratios:

- So... Horton's laws are defined by three ratios:

$$R_n, \ R_\ell, \text{ and } R_a.$$

- Horton's laws describe exponential decay or growth:

$$\begin{aligned} n_\omega &= n_{\omega-1}/R_n \\ &= n_{\omega-2}/R_n^2 \\ &\vdots \\ &= n_1/R_n^{\omega-1} \\ &= n_1 e^{-(\omega-1)\ln R_n} \end{aligned}$$

Horton's laws

Similar story for area and length:

- $$\bar{a}_\omega = \bar{a}_1 e^{(\omega-1)\ln R_a}$$
- $$\bar{\ell}_\omega = \bar{\ell}_1 e^{(\omega-1)\ln R_\ell}$$
- As stream order increases, number drops and area and length increase.

Horton's laws

Similar story for area and length:

- $$\bar{a}_\omega = \bar{a}_1 e^{(\omega-1)\ln R_a}$$
- $$\bar{\ell}_\omega = \bar{\ell}_1 e^{(\omega-1)\ln R_\ell}$$
- As stream order increases, number drops and area and length increase.

Horton's laws

Similar story for area and length:

- $$\bar{a}_\omega = \bar{a}_1 e^{(\omega-1)\ln R_a}$$
- $$\bar{\ell}_\omega = \bar{\ell}_1 e^{(\omega-1)\ln R_\ell}$$
- As stream order increases, number drops and area and length increase.

Horton's laws

A few more things:

- Horton's laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton's ratios go a long way to defining a branching network...
- But we need one other piece of information...

Horton's laws

A few more things:

- Horton's laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton's ratios go a long way to defining a branching network...
- But we need one other piece of information...

Horton's laws

A few more things:

- Horton's laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton's ratios go a long way to defining a branching network...
- But we need one other piece of information...

Horton's laws

A few more things:

- Horton's laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton's ratios go a long way to defining a branching network...
- But we need one other piece of information...

Horton's laws

A few more things:

- Horton's laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton's ratios go a long way to defining a branching network...
- But we need one other piece of information...

Horton's laws

A few more things:

- Horton's laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton's ratios go a long way to defining a branching network...
- But we need one other piece of information...

Horton's laws

A bonus law:

- Horton's law of stream segment lengths:

$$\bar{s}_{\omega+1}/\bar{s}_{\omega} = R_s > 1$$

- Can show that $R_s \equiv R_\ell$.
- Insert question 2, assignment 2 (⊞)

Horton's laws

A bonus law:

- Horton's law of stream segment lengths:

$$\bar{s}_{\omega+1}/\bar{s}_{\omega} = R_s > 1$$

- Can show that $R_s = R_\ell$.
- Insert question 2, assignment 2 (⊞)

Horton's laws

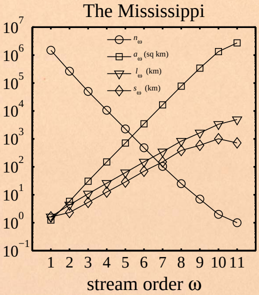
A bonus law:

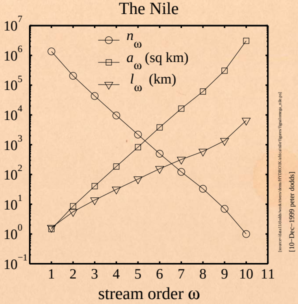
- Horton's law of stream segment lengths:

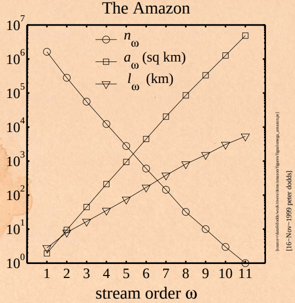
$$\boxed{\bar{s}_{\omega+1}/\bar{s}_{\omega} = R_s > 1}$$

- Can show that $R_s = R_\ell$.
- Insert question 2, assignment 2 (⊞)

Horton's laws in the real world:







Horton's laws-at-large

Blood networks:

- Horton's laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton's law.

Horton's laws-at-large

Blood networks:

- Horton's laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton's law.

Horton's laws-at-large

Blood networks:

- Horton's laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton's law.

Horton's laws-at-large

Blood networks:

- Horton's laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton's law.

Data from real blood networks

Network	R_n	R_r^{-1}	R_ℓ^{-1}	$-\frac{\ln R_r}{\ln R_n}$	$-\frac{\ln R_\ell}{\ln R_n}$	α
West *et al.*	–	–	–	1/2	1/3	3/4
rat (PAT)	2.76	1.58	1.60	0.45	0.46	0.73
cat (PAT) (Turcotte *et al.* [10])	3.67	1.71	1.78	0.41	0.44	0.79
dog (PAT)	3.69	1.67	1.52	0.39	0.32	0.90
pig (LCX)	3.57	1.89	2.20	0.50	0.62	0.62
pig (RCA)	3.50	1.81	2.12	0.47	0.60	0.65
pig (LAD)	3.51	1.84	2.02	0.49	0.56	0.65
human (PAT)	3.03	1.60	1.49	0.42	0.36	0.83
human (PAT)	3.36	1.56	1.49	0.37	0.33	0.94

Horton's laws

Observations:

- Horton's ratios vary:

R_n	3.0–5.0
R_a	3.0–6.0
R_ℓ	1.5–3.0

- No accepted explanation for these values.
- Horton's laws tell us how quantities vary from level to level ...
- ... but they don't explain how networks are structured.

Horton's laws

Observations:

- Horton's ratios vary:

R_n	3.0–5.0
R_a	3.0–6.0
R_ℓ	1.5–3.0

- No accepted explanation for these values.
- Horton's laws tell us how quantities vary from level to level ...
- ... but they don't explain how networks are structured.

Horton's laws

Observations:

- Horton's ratios vary:

R_n	3.0–5.0
R_a	3.0–6.0
R_ℓ	1.5–3.0

- No accepted explanation for these values.
- Horton's laws tell us how quantities vary from level to level ...
- ... but they don't explain how networks are structured.

Horton's laws

Observations:

- Horton's ratios vary:

R_n	3.0–5.0
R_a	3.0–6.0
R_ℓ	1.5–3.0

- No accepted explanation for these values.
- Horton's laws tell us how quantities vary from level to level ...
- ... but they don't explain how networks are structured.

Tokunaga's law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use stream ordering.
- Focus: describe how streams of different orders connect to each other.
- Tokunaga's law is also a law of averages.

Tokunaga's law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use stream ordering.
- Focus: describe how streams of different orders connect to each other.
- Tokunaga's law is also a law of averages.

Tokunaga's law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use stream ordering.
- Focus: describe how streams of different orders connect to each other.
- Tokunaga's law is also a law of averages.

Tokunaga's law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use stream ordering.
- Focus: describe how streams of different orders connect to each other.
- Tokunaga's law is also a law of averages.

Tokunaga's law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use stream ordering.
- Focus: describe how streams of different orders connect to each other.
- Tokunaga's law is also a law of averages.

Network Architecture

Definition:

- $T_{\mu,\nu}$ = the average number of side streams of order ν that enter as tributaries to streams of order μ
- $\mu, \nu = 1, 2, 3, \ldots$
- $\mu \geq \nu + 1$
- Recall each stream segment of order μ is 'generated' by two streams of order $\mu - 1$
- These generating streams are not considered side streams.

Network Architecture

Definition:

- $T_{\mu,\nu}$ = the average number of side streams of order ν that enter as tributaries to streams of order μ
- μ, ν = 1, 2, 3, ...
- $\mu \geq \nu + 1$
- Recall each stream segment of order μ is 'generated' by two streams of order $\mu - 1$
- These generating streams are not considered side streams.

Network Architecture

Definition:

- $T_{\mu,\nu}$ = the average number of side streams of order ν that enter as tributaries to streams of order μ
- μ, ν = 1, 2, 3, . . .
- $\mu \geq \nu + 1$
- Recall each stream segment of order μ is 'generated' by two streams of order $\mu - 1$
- These generating streams are not considered side streams.

Network Architecture

Definition:

- $T_{\mu,\nu}$ = the average number of side streams of order ν that enter as tributaries to streams of order μ
- μ, ν = 1, 2, 3, . . .
- $\mu \geq \nu + 1$
- Recall each stream segment of order μ is 'generated' by two streams of order $\mu - 1$
- These generating streams are not considered side streams.

Network Architecture

Definition:

- $T_{\mu,\nu}$ = the average number of side streams of order ν that enter as tributaries to streams of order μ
- μ, ν = 1, 2, 3, ...
- $\mu \geq \nu + 1$
- Recall each stream segment of order μ is 'generated' by two streams of order $\mu - 1$
- These generating streams are not considered side streams.

Network Architecture

Tokunaga's law

- Property 1: Scale independence—depends only on difference between orders:

$$T_{\mu,\nu} = T_{\mu-\nu}$$

- Property 2: Number of side streams grows exponentially with difference in orders:

$$T_{\mu,\nu} = T_1 (R_T)^{\mu-\nu-1}$$

- We usually write Tokunaga's law as:

$$\boxed{T_k = T_1 (R_T)^{k-1}} \quad \text{where } R_T \simeq 2$$

Network Architecture

Tokunaga's law

- Property 1: Scale independence—depends only on difference between orders:

$$T_{\mu,\nu} = T_{\mu-\nu}$$

- Property 2: Number of side streams grows exponentially with difference in orders:

$$T_{\mu,\nu} = T_1 (R_T)^{\mu-\nu-1}$$

- We usually write Tokunaga's law as:

$$\boxed{T_k = T_1 (R_T)^{k-1}} \quad \text{where } R_T \simeq 2$$

Network Architecture

Tokunaga's law

- Property 1: Scale independence—depends only on difference between orders:

$$T_{\mu,\nu} = T_{\mu-\nu}$$

- Property 2: Number of side streams grows exponentially with difference in orders:

$$T_{\mu,\nu} = T_1 (R_T)^{\mu-\nu-1}$$

- We usually write Tokunaga's law as:

$$\boxed{T_k = T_1 (R_T)^{k-1}} \quad \text{where } R_T \simeq 2$$

Network Architecture

Tokunaga's law

- Property 1: Scale independence—depends only on difference between orders:

$$T_{\mu,\nu} = T_{\mu-\nu}$$

- Property 2: Number of side streams grows exponentially with difference in orders:

$$T_{\mu,\nu} = T_1 (R_T)^{\mu-\nu-1}$$

- We usually write Tokunaga's law as:

$$\boxed{T_k = T_1 (R_T)^{k-1}} \quad \text{where } R_T \simeq 2$$

Network Architecture

Tokunaga's law

- Property 1: Scale independence—depends only on difference between orders:

$$T_{\mu,\nu} = T_{\mu-\nu}$$

- Property 2: Number of side streams grows exponentially with difference in orders:

$$T_{\mu,\nu} = T_1(R_T)^{\mu-\nu-1}$$

- We usually write Tokunaga's law as:

$$\boxed{T_k = T_1(R_T)^{k-1}} \quad \text{where } R_T \simeq 2$$

Network Architecture

Tokunaga's law

- Property 1: Scale independence—depends only on difference between orders:

$$T_{\mu,\nu} = T_{\mu-\nu}$$

- Property 2: Number of side streams grows exponentially with difference in orders:

$$T_{\mu,\nu} = T_1 (R_T)^{\mu-\nu-1}$$

- We usually write Tokunaga's law as:

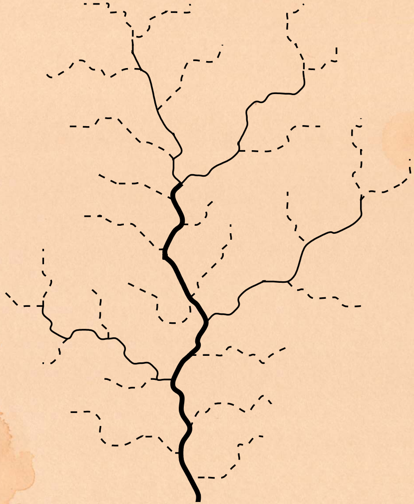
$$\boxed{T_k = T_1 (R_T)^{k-1}} \quad \text{where } R_T \simeq 2$$

The UNIVERSITY of VERMONT

Tokunaga's law—an example:

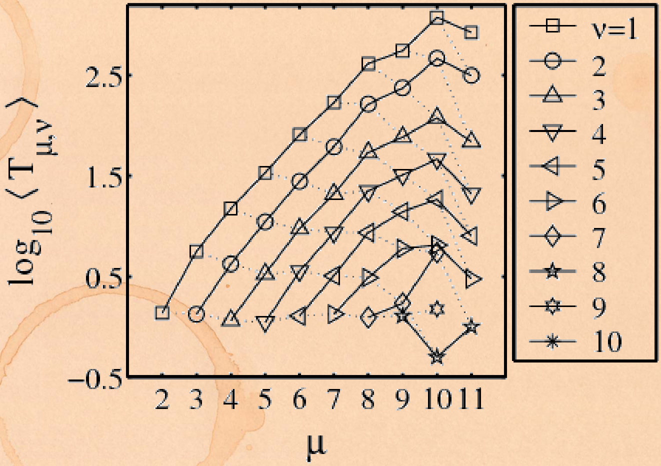
$T_1 \simeq 2$

$R_T \simeq 4$



The Mississippi

A Tokunaga graph:



Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

The UNIVERSITY of VERMONT

Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton's laws reveal self-similarity.
- Horton's laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga's laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).

References I

[1] P. S. Dodds and D. H. Rothman.
Unified view of scaling laws for river networks.
Physical Review E, 59(5):4865–4877, 1999. pdf (⊞)

[2] J. T. Hack.
Studies of longitudinal stream profiles in Virginia and Maryland.
United States Geological Survey Professional Paper, 294-B:45–97, 1957.

[3] R. E. Horton.
Erosional development of streams and their drainage basins; hydrophysical approach to quatitative morphology.
Bulletin of the Geological Society of America, 56(3):275–370, 1945.

References II

[4] I. Rodríguez-Iturbe and A. Rinaldo.
Fractal River Basins: Chance and Self-Organization.
Cambridge University Press, Cambrigde, UK, 1997.

[5] S. A. Schumm.
Evolution of drainage systems and slopes in badlands at Perth Amboy, New Jersey.
Bulletin of the Geological Society of America, 67:597–646, 1956.

[6] A. N. Strahler.
Hypsometric (area altitude) analysis of erosional topography.
Bulletin of the Geological Society of America, 63:1117–1142, 1952.

References III

[7] E. Tokunaga.
The composition of drainage network in Toyohira River Basin and the valuation of Horton's first law.
Geophysical Bulletin of Hokkaido University, 15:1–19, 1966.

[8] E. Tokunaga.
Consideration on the composition of drainage networks and their evolution.
Geographical Reports of Tokyo Metropolitan University, 13:G1–27, 1978.

[9] E. Tokunaga.
Ordering of divide segments and law of divide segment numbers.
Transactions of the Japanese Geomorphological Union, 5(2):71–77, 1984.

References IV

[10] D. L. Turcotte, J. D. Pelletier, and W. I. Newman.
Networks with side branching in biology.
Journal of Theoretical Biology, 193:577–592, 1998.
pdf (⊞)